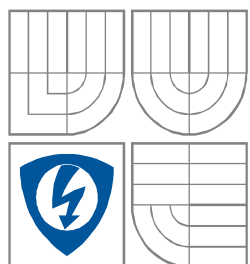


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

Komunikační modul pro řízení laboratorního zdroje

Communication module for power supplies controlling

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

Bc. Jiří Zachar

AUTHOR

VEDOUCÍ PRÁCE

Ing. Jiří Dřínovský, Ph.D.

SUPERVISOR

BRNO, 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Jiří Zachar
Ročník: 2

ID: 74931
Akademický rok: 2009/2010

NÁZEV TEMATU:

Komunikační modul pro řízení laboratorního zdroje

POKYNY PRO VYPRACOVÁNÍ:

Podrobně se seznámte s možností řízení laboratorního zdroje Manson SDP-2405. Pro tento zdroj navrhnete obvodové zapojení ovládacího modulu, kterým bude možné ovládat tento zdroj pomocí rozhraní USB. Příkazy, kterými bude možné tento laboratorní zdroj ovládat, musí být kompatibilní s protokolem SCPI (Standard Commands for Programmable Instruments).

Vámi navržený ovládací modul realizujte a ověřte jeho činnost. Ověřte též funkčnost jednotlivých vlastností zdroje, tak že jej budete ovládat pomocí vašeho modulu.

S pomocí vámi realizovaného modulu a laboratorního zdroje Manson SPD-2405 navrhnete jednoduchou laboratorní úlohu, která bude ovládána z programového prostředí VEE. Laboratorní úlohu koncipujte taky, aby byla jednoduchá a seznámila studenty s programovým prostředím Agilent VEE.

DOPORUČENÁ LITERATURA:

[1] KAINKA, B. Měření, řízení a regulace pomocí PC. Praha: BEN - technická literatura, 2003. 272 s.

[2] SCPI Consortium, Standard Commands for Programmable Instruments (SCPI) Volume 1: Syntax and Style [online], Versiton 1999.0, May 1999, [cit. dne 28-4-2008] dostupné na [www <http://www.scpiconsortium.org/SCPI-99.pdf>](http://www.scpiconsortium.org/SCPI-99.pdf).

Termín zadání: 8.2.2010

Termín odevzdání: 21.5.2010

Vedoucí práce: Ing. Jiří Dřinovský, Ph.D.

prof. Dr. Ing. Zbyněk Raida
Předseda oborové rady

LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Jiří Zachar
Bytem: Skupova 19, Olomouc, 779 00
Narozen/a (datum a místo): 6. července 1985 v Olomouci

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 53, Brno, 602 00
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika
(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- disertační práce
 - diplomová práce
 - bakalářská práce
 - jiná práce, jejíž druh je specifikován jako
- (dále jen VŠKP nebo dílo)

Název VŠKP: Komunikační modul pro řízení laboratorního zdroje

Vedoucí/ školitel VŠKP: Ing. Jiří Dřínovský, Ph.D

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP: _____

VŠKP odevzdal autor nabyvateli*:

- v tištěné formě – počet exemplářů: 2
- v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

* hodící se zaškrtněte

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 21. května 2010

.....

Nabyvatel

.....

Autor

ABSTRAKT

Cílem této práce je navrhnout komunikační modul pro laboratorní zdroj Manson. Tento napájecí zdroj komunikuje pouze prostřednictvím sériové linky RS-232. Moderní počítače obvykle používají pro připojení dalších periférií standard USB. Komunikační modul obsahuje převodník mezi sériovou linkou RS-232 a standardem USB. Napájecí zdroj se používá v automatizovaném měření, které je ovládáno z virtuálního vývojového prostředí od Agilent Technologies. Přímé příkazy pro laboratorní zdroj Manson nejsou kompatibilní se standardem SCPI (Standard Commands for Programmable Instruments). Takže, komunikační modul převádí příkazy používané pro zdroj Manson do formátu SCPI. Komunikace s laboratorním zdrojem Manson bude pro studenty mnohem názornější s využitím nového komunikačního modulu.

KLÍČOVÁ SLOVA

Komunikační modul, mikroprocesor, USB, RS-232, laboratorní zdroj, SCPI, VEE

ABSTRACT

The aim of this work is to design a communication module for laboratory source Manson. This power supply communicates only through a serial RS-232 link. Modern computers usually use for the connection with other peripherals mainly USB. Designed communication module contains converter between the RS-232 and USB. Power supply is used in the automated measurement setup, which is controlled from the Virtual Engineering Environment from the Agilent Technologies. Direct commands for laboratory source Manson are not compatible with the SCPI (Standard Commands for Programmable Instruments) standard commands. So, the communication module converts commands from the SCPI format to Manson supply instructions. The communication with the Manson supply will be obvious for students by using new communication module.

KEYWORDS

Communication module, microcontroller, USB, RS-232, Power supply, SCPI, VEE

ZACHAR, J. *Komunikační modul pro řízení laboratorního zdroje*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2010. 67 s., 10 s. příloh. Diplomová práce. Vedoucí práce: Ing. Jiří Dřínovský, Ph.D.

Prohlášení

Prohlašuji, že svoji diplomovou práci na téma Komunikační modul pro řízení laboratorního zdroje jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této 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 21. května 2010

.....

podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Jiřímu Dřínovskému, Ph.D za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne 21. května 2010

.....

podpis autora

OBSAH

Seznam obrázků

Seznam tabulek

Úvod	1
1 PC rozhraní	3
1.1 Sériové rozhraní RS-232	3
1.1.1 Princip sériové komunikace	3
1.1.2 Základní parametry RS-232	4
1.1.3 Základní zapojení konektoru a popis signálů	6
1.1.4 Připojení RS-232 na TTL logiku	7
1.2 Sériové rozhraní RS-485	7
1.3 Univerzální rozhraní USB	8
1.3.1 Parametry USB	8
1.3.2 Konektory a kabely	8
1.3.3 Verze USB a přenosové rychlosti	9
1.3.4 Přenos Dat	10
1.3.5 Rozpoznávání zařízení	11
2 Řízení laboratorního zdroje	12
2.1 Technické parametry zdroje	12
2.2 Příkazy laboratorního zdroje	14
2.3 Standard SCPI	16
2.4 Připojení zdroje k PC	17
3 Převod RS-232↔USB	20
3.1 RS232↔USB s obvody FTDI	20
3.2 RS-232↔USB s AT90USB	21
3.2.1 Základní vlastnosti	21
3.2.2 Návrh zapojení komunikačního modulu	24
3.2.3 Deska plošných spojů	29
4 Obslužný software	31
4.1 Projekt LUFA	31
4.2 Hlavní program	32
4.3 Úprava komunikace	35
4.3.1 Nastavení napěťové a proudové pojistky	38
4.3.2 Měření výstupního napětí a proudu	42
4.3.3 Aktivace výstupních svorek	45
4.3.4 Počáteční nastavení laboratorního zdroje	47
4.3.5 Časová prodleva	49
5 Laboratorní úloha	50
5.1 Seznámení s prostředím VEE Pro	50
5.2 Postup měření	55
5.3 Seznámení s prostředím Matlab	63
6 Závěr	65
Literatura	67
Příloha	68

SEZNAM OBRÁZKŮ

Obr. 1.1 Význam jednotlivých pinů konektoru DB-9/DE-9	6
Obr. 1.2 Konektory USB	9
Obr. 2.1 Laboratorní zdroj Manson SDP-2405	13
Obr. 2.2 Zadní část laboratorního zdroje Manson SDP-2405	14
Obr. 2.3 Připojení laboratorního zdroje k PC pomocí RS-232	18
Obr. 2.4 Připojení laboratorní zdroje pomocí komunikačního modulu	18
Obr. 2.5 Propojení laboratorního zdroje s PC pomocí RS-485	18
Obr. 2.6 Doporučené zapojení konektoru RS-232	19
Obr. 3.1 Blokové schéma obvodu FT232BM	21
Obr. 3.2 Mikroprocesor AT90USB v pouzdře TQFP64	24
Obr. 3.3 Blokové schéma komunikačního modulu pro laboratorní zdroj	25
Obr. 3.4 Schematické zapojení USB konektoru	26
Obr. 3.5 Zapojení převodníku úrovní MAX232 a jeho vnitřní struktura	26
Obr. 3.6 Doporučené zapojení ISP konektoru	28
Obr. 3.7 Schéma zapojení komunikačního modulu	29
Obr. 3.8 Vyrobena deska plošných spojů	30
Obr. 3.9 Konečná podoba komunikačního modulu	30
Obr. 4.1 Nastavení komunikačního portu	32
Obr. 4.2 Vývojové diagramy pro zpracování přijatých dat	34
Obr. 4.3 Vývojový diagram pro odeslání dat do laboratorního zdroje	36
Obr. 4.4 Vývojový diagram pro zpracování dat z laboratorního zdroje	37
Obr. 4.5 Vývojový diagram pro nastavení napěťové pojistky	40
Obr. 4.6 Vývojový diagram pro nastavení proudové pojistky	42
Obr. 4.7 Vývojový diagram pro měření výstupního napětí a proudu	44
Obr. 4.8 Vývojový diagram pro aktivaci a deaktivaci výstupních svorek	46
Obr. 4.9 Vývojový diagram pro reset a identifikaci laboratorního zdroje	48
Obr. 5.1 Připojený laboratorní zdroj v aplikaci Agilent Connection Expert	51
Obr. 5.2 Základní rozhraní programového prostředí Agilent VEE Pro	52
Obr. 5.3 Instrument manager	53
Obr. 5.4 Význam jednotlivých vstupů a výstupů objektu DIRECT I/O	54
Obr. 5.5 Nastavení komunikace	56
Obr. 5.6 Nastavení Direct I/O	56

Obr. 5.7 Cyklus FOR RANGE.....	58
Obr. 5.8 Objekt FORMULA	58
Obr. 5.9 Vlastnosti virtuálního vodiče	59
Obr. 5.10 Původní laboratorní úloha.....	60
Obr. 5.11 Nová laboratorní úloha.....	60
Obr. 5.12 Ampér – Voltová charakteristika žárovky	61
Obr. 5.13 Určení střední hodnoty odporu žárovky.....	62
Obr. 5.14 Postupné zhasnutí žárovky a určení výstupního výkonu	62
Obr. 5.15 Test&Measurement Tool	64

SEZNAM TABULEK

Tab. 1.1 Standardní používané napěťové úrovně v sériové komunikaci	4
Tab. 1.2 Maximální délka sériového vedení.	5
Tab. 1.3 Přenosové rychlosti přijímače a vysílače pro přenos 8 datových bitů.	5
Tab. 1.4 Označení a popis pinů	6
Tab. 1.5 Vývody USB konektoru.....	9
Tab. 1.6 Přenosové rychlosti standardu USB.....	9
Tab. 2.1 Základní parametry zdroje	13
Tab. 2.2 Přehled příkazů používaných při komunikaci s laboratorním zdrojem	16
Tab. 2.3 Přehled příkazů pro komunikační modul	17

ÚVOD

Cílem diplomové práce je návrh a realizace ovládacího modulu pro řízení laboratorního zdroje Manson SDP-2405. Tento zdroj komunikuje s osobním počítačem převážně pomocí sériové linky RS-232. Případně je možné využít rychlejší sériovou komunikaci pomocí průmyslové linky RS-485, která umožňuje připojit více zdrojů na jednu sériovou linku ovládanou jedním osobním počítačem. Tato sběrnice se převážně využívá v průmyslu. V dnešní době je komunikace po klasické sériové lince RS-232 na ústupu. U nových osobních počítačů již dokonce není sériová linka vyvedena na počítačové skříni. Takže pro většinu dnes běžně používaných počítačů je nutné použít různé převodníky mezi jednotlivými rozhraními. Od klasické sériové linky RS-232 se začalo postupně přecházet k modernějším a hlavně rychlejším počítačovým rozhraním. Za moderní sériový komunikační prostředek se dnes výhradně považuje rozhraní USB a to momentálně s nejrozšířenějším standardem USB 2.0. Přes sériové rozhraní USB se dnes ke stolním počítačům připojují téměř veškeré periferie od klasických klávesnic, tiskáren až po právě různé měřicí přístroje. Rozhraní USB tedy sjednocuje téměř všechny periferie připojované k osobním počítačům a výrazně usnadňuje komunikaci počítače s okolními zařízeními.

Jelikož laboratorní zdroj Manson SDP-2405 nemá možnost připojení přes rozhraní USB je prvním dílčím cílem projektu vytvořit komunikační modul, který bude komunikovat se zdrojem právě pomocí rozhraní USB. Pomocí vhodného mikroprocesoru, který bude umět obsluhovat obě použítá rozhraní, bude vytvořen komunikační modul, který výrazně usnadní komunikaci s laboratorním zdrojem. S vytvořeným modulem bude možné laboratorní zdroj připojit k dnes běžně používaným osobním počítačům a velmi pohodlně s ním komunikovat přes sériové rozhraní USB. V první části práce budou rozebrány základní parametry a principy sériových rozhraní. Hlavní pozornost bude upřena zejména na komunikaci pomocí sériové linky RS-232, a také samozřejmě na komunikaci pomocí rozhraní USB.

V další části projektu budou rozebrány možnosti převodníků mezi rozhraním RS-232 a standardem USB. V dnešní době existuje několik možností, jak realizovat tento převod. Na trhu je možné zakoupit již zhotovené převodníky realizující obousměrný převod jako koncové výrobky pro uživatele. Součástí této práce je navrhnutí obdobného převodníku s tím rozdílem, že bude možné komunikaci dle potřeby upravovat. Pro realizaci převodníku je možné použít dvě podobné řešení, přičemž pro konkrétní komunikační modul bude jedno řešení navrženo detailně. První možnost spočívá v zakoupení integrovaného obvodu od firmy FTDI dodávané na český trh firmou ASIX. Pomocí těchto obvodů se dnes realizuje mnoho USB aplikací, jejich největší výhodou je velmi nízká pořizovací cena a uživatelsky příjemná práce s nastavením těchto součástí. Přímo od výrobce jsou přizpůsobeny pro snadnou práci s USB rozhraním. Ovšem pomocí pouze těchto obvodů nelze realizovat komunikační modul pro laboratorní zdroj Manson. Do návrhu zapojení by bylo nutné vybrat vhodný mikroprocesor, který by byl schopný řídit obousměrnou komunikaci s požadovanými úpravami. Celý návrh by se tedy poměrně zkomplikoval. Druhou možností, jak navrhnout komunikační modul je výběr z bohaté nabídky různých typů mikroprocesorů od firmy Atmel, které mají v sobě přímo implementováno rozhraní USB a zároveň obsahují standardně používané periferie, mezi nimiž nechybí komunikace po sériové lince. Tento způsob realizace bude proveden v rámci diplomové práce. Vhodným zapojením těchto mikroprocesorů lze snadno dosáhnout převodu mezi USB standardem a sériovou linkou RS-232 a navíc je možné nevyužité části mikroprocesoru vhodně zapojit pro další potřeby a úpravy komunikace, které budou usnadňovat uživatelskou komunikaci s laboratorním zdrojem.

Pomocí vybraného mikroprocesoru AT90USB1287 bude zrealizován v rámci diplomové práce komunikační modul, který bude aplikovaný přímo na laboratorní zdroj Manson SDP-2405 používaný v laboratoři. Komunikační modul by měl studentům usnadnit práci při komunikaci s laboratorním zdrojem. Zdroj používá pro komunikaci definovanou sadu příkazů, které se běžně nepoužívají při automatizovaném měření pomocí grafických prostředí. Součástí komunikačního modulu tedy bude i patřičná úprava příkazů. Příkazy používané pro laboratorní zdroj Manson nejsou kompatibilní se standardem SCPI (Standard Commands for Programmable Instruments). Při komunikaci bude nutné vhodně implementovat převod příkazů, které standardně využívá laboratorní zdroj Manson na příkazy, které se obvykle používají v grafickém programovacím prostředí Agilent VEE Pro a zároveň splňují požadavky standardu SCPI. Tento převod bude realizovat vybraný mikroprocesor. Přijatá data přes rozhraní USB mikroprocesor zpracuje, tedy převede do vhodného tvaru a odešle na sériovou linku. To samé platí pro opačnou komunikaci, kdy přijatá data ze sériové linky RS-232 převede do vhodného tvaru a pošle na rozhraní USB. Jedná se o často používané příkazy pro nastavení výstupního napětí, proudu a pro zpětné vyčtení nastavených hodnot ze zdroje. To vše by mělo sloužit pro usnadnění práce s tímto laboratorním zdrojem, který je používán v laboratorním cvičení, kde má demonstrovat jednoduchou práci v grafickém prostředí Agilent VEE Pro.

Závěrečná část diplomové práce se věnuje návrhu laboratorní úlohy, ve které bude komunikační modul využíván a bude ovládán z programového prostředí Agilent VEE Pro. Účelem laboratorní úlohy má být názorné seznámení studentů s programovacím prostředím Agilent VEE Pro. Komunikační modul má být zařazen jako úvodní úloha do předmětu Radioelektronická měření, kde je využíván pro základní pochopení prostředí VEE. Studenti zde ovládají již zmiňovaná laboratorní zdroj Manson SDP-2405. Laboratorní zdroj je v této úloze již využíván, ale práci studentů komplikuje odlišnost používaných příkazů. Motivací této práce je tedy standardizovat komunikační příkazy pro všechny laboratorní úlohy a tím usnadnit studentům studium předkládané látky a zároveň umožnit připojení zdroje přes rozhraní USB. Součástí navrhované úlohy bude i podrobný laboratorní návod, pomocí kterého budou schopni studenti absolvovat měření ve vyhrazeném časovém úseku. Laboratorní návod by měl představit grafické prostředí Agilent VEE Pro a navést studenty k naprogramování zadané úlohy, která by měla reprezentovat snadné ovládání napájecího zdroje Manson SDP-2405 v tomto prostředí.

1 PC ROZHRAŇÍ

Dnešní osobní počítače využívají pro komunikaci s okolím několik druhů rozhraní. Rozhraní můžeme rozčlenit do několika velkých skupin. První velká skupina jsou paralelní rozhraní. Mezi paralelní rozhraní patří například standardní paralelní port pro připojení tiskárny. Další skupina jsou sériová rozhraní. Sériové rozhraní je vůbec nejstarším používaným rozhraním. Sériový přenos je pomalejší než paralelní komunikace, ale u pomalejších zařízení svojí rychlostí vyhovuje. Nejpoužívanějším rozhraním je univerzální USB rozhraní, přes které je možné připojit podstatnou část dnešních externích zařízení připojovaných k osobním počítačům. V této části bude podrobně přiblíženo sériové rozhraní RS-232 a USB rozhraní.

1.1 Sériové rozhraní RS-232

Sériová linka se využívá jako komunikační rozhraní osobních počítačů a další elektroniky včetně různých měřících přístrojů. Počátky této komunikace se datují někde kolem roku 1969. Původně bylo koncipováno pro připojení textových terminálů k jednoduchému modemu či přímo k blízkému serveru, posléze se však působnost jeho nasazení dosti podstatným způsobem rozšířila. Na sériový port se v minulosti připojovala například světelná pera, počítačové myši, dotykové LCD zobrazovače, perové i řezací plottery a některé tiskárny s typovým kolečkem, původní „pípající“ i moderní vysokorychlostní modemy, propojovaly se s ním počítače, budovaly jednoduché počítačové sítě apod. [1]. V současné době je sériové rozhraní na ústupu. Většina moderních osobních počítačů již nemá sériovou linku RS-232 vyvedenou ze základní desky počítače na zadní panel konektorů. Sériové rozhraní je nahrazováno modernějším, rychlejším a univerzálnějším USB rozhraním. U běžně dostupných externích zařízení se již sériové rozhraní RS-232 neobjevuje, nicméně v průmyslu je tento standard velice rozšířen a pro své specifické rysy tomu tak bude i nadále. Sériové rozhraní RS-232 na rozdíl od komplexnějšího USB pouze definuje, jak přenést určitou sekvenci bitů a nezabývá se už vyššími vrstvami komunikace. V referenčním modelu tedy představuje pouze fyzickou vrstvu [2].

1.1.1 Princip sériové komunikace

Komunikace pomocí sériové linky je založena na vysílání dat rozložených na jednotlivé bity. Vysílaná data se skládají samozřejmě z dat, které chceme přenést, ale pro správný přenos je třeba tyto data obohatit ještě o další bity, které většinou mají na starost synchronizaci, detekci chyb případně jejich korekci. Tok bitů musí být nějakým způsobem synchronizován, aby přijímací strana byla schopná rozeznat posílaná data a dále s nimi pracovat. Pro synchronizaci se používá kromě vlastního vodiče ještě vodič s hodinovým signálem (synchronní přenos) nebo je možné synchronizaci vhodně umístit do přenášené informace. Například je možné jedničkový bit reprezentovat přechodem z vysoké do nízké úrovně a bit nulový naopak přechodem z úrovně nízké do úrovně vysoké. Tím se však zvyšují nároky na kapacitu kanálu.

Konkrétně sériové rozhraní RS-232C využívá pro komunikaci UART (Universal Asynchronous Receiver and Transmitter), v překladu se tato komunikace označuje jako univerzální asynchronní přijímač a vysílač. Asynchronní komunikace se vyznačuje tím, že nepoužívá ani jeden zmíněný postup synchronizace uvedený v tomto textu. Tato komunikace používá synchronizační bity vkládané na začátek a konec posílaných dat. V praktických aplikacích se využívá 5, 6, 7 a nejvíce 8 bitová komunikace. Před začátkem komunikace je

potřeba přijímač i vysílač vhodně nakonfigurovat tak, aby komunikace probíhala plynule bez chyb a případných datových ztrát. Přijímač musí vědět jaká data má očekávat a jak často by měl vzorkovat datovou linku, neboli jaká bude použita přenosová rychlost komunikace. Obě zařízení musí mít stejně nastavenou délku přenášených dat, přenosovou rychlost uváděnou v bitech za sekundu (bitrate), délku stopbitu a v některých případech i to, zda a jakým způsobem se přenáší paritní bit, který představuje velmi primitivní podobu detekčního kódu ovšem u sériové komunikace často využívanou. Datový vodič se před zahájením komunikace nachází v klidové poloze, což v této komunikaci znamená vysokou úroveň napětí. Tato úroveň se může na datovém vodiči nacházet po libovolnou dobu do zahájení komunikace.

Zahájení komunikace se provede zasláním tzv. start bitu, který musí mít naopak nízkou úroveň napětí, což je logické, přijímač musí rozeznat zahájení komunikace od klidového stavu datové linky. Přijímací linka po přijetí start bitu očekává příchod předem nadefinovaných dat, která přijdou po datové lince s přesně definovanou přenosovou rychlostí. Přijímací strana tedy musí dostatečně rychle vzorkovat datovou linku, aby přijala všechna posílaná data. Na závěr se může poslat tzv. paritní bit, podle kterého přijímač odhalí, zda přijal data správně. Jedná se o velmi jednoduchou ochranu dat, která při více chybných bitech není schopna chyby odhalit. O uzavření komunikace se postará tzv. stop bit, který má vždy hodnotu vysoké úrovně napětí a tedy nastavuje datovou linku zpátky do klidového stavu. Přijímač čeká na znovuzahájení komunikace příchodem start bitu. Nejčastěji používaná délka stop bitu je jeden stop bit, lze použít i delší stop bity 1,5 až 2 násobek délky běžného bitu, aby měl přijímač dostatek času na zpracování přijatých dat.

Díky přítomnosti start a stop bitů je v praxi možné používat data obsahující samé vysoké nebo nízké logické úrovně příchodem start bitu dojde k opětovné synchronizaci. Kvalita čtení posílaných dat je závislá na kvalitě použitého UART zařízení. Většina těchto zařízení používá vyšší vzorkovací frekvenci než je rychlost přenosu dat. To v praxi znamená, že posílaná data jsou během doby bitu přečteny několikrát a následně vyhodnocena. Tímto se eliminuje vliv šumu a možného rušení z okolních zařízení. Samozřejmě vyšší vzorkovací frekvence zabezpečuje i přesnější určení start bitu a zahájení komunikace [1].

1.1.2 Základní parametry RS-232

Sériová komunikace RS-232 používá dvě napěťové úrovně. Logická 1 někdy označovaná jako *marking state* nebo také klidový stav komunikace. Logická 0 se označuje *space rate*. Důležitá informace je napěťová úroveň příslušející logické 1 respektive logické 0. Logická 1 odpovídá záporné napěťové úrovni a logická 0 odpovídá kladné napěťové úrovni. Dovolené napěťové úrovně pro přijímač a vysílač jsou uvedeny v tab. 1.1 [3].

Tab. 1.1 Standardní používané napěťové úrovně v sériové komunikaci

Datové signály		
Úroveň	Vysílač	Přijímač
Logická 0	+5 až +15 V	+3 až +25V
Logická 1	-5 až -15 V	-3 až -25V

Nejběžněji se pro generování napětí používá napěťový zdvojovač z 5 V a invertor. Tedy logické úrovně odpovídají hodnotám napětí + 10 V pro logickou 0 a -10 V pro logickou 1. Mezi další parametry může být zařazena maximální délka přenosového vedení. Standard

sériové komunikace uvádí jako maximální délku vedení 15 metrů anebo délku vodiče o kapacitě 2500 pF. Tedy při používání kvalitních vodičů je možné délku vedení protáhnout až na 50 metrů, taková délka vedení není ovšem v praxi většinou potřeba. Firma Texas Instruments prováděla laboratorní testování a zveřejnila následující parametry délky vedení v závislosti na přenosové rychlosti. Hodnoty uvedené v tab. 1.2 je nutné brát s rezervou. Chování sériové linky v laboratoři a v reálném rušeném prostředí se mohou značně lišit [3].

Tab. 1.2 Maximální délka sériového vedení

Přenosová rychlost [Bd]	Maximální délka vedení [m]
19200	15
9600	150
4800	300
2400	900

Přenosové rychlosti při sériovém přenosu dat jsou odvozeny od násobku 300 bitů za sekundu. V následující tab. 1.3 jsou uvedeny přenosové rychlosti platné při nastavení přijímače i vysílače na přenos osmi datových bitů, jednoho start bitu a stop bitu s délkou odpovídající délce běžného bitu [1].

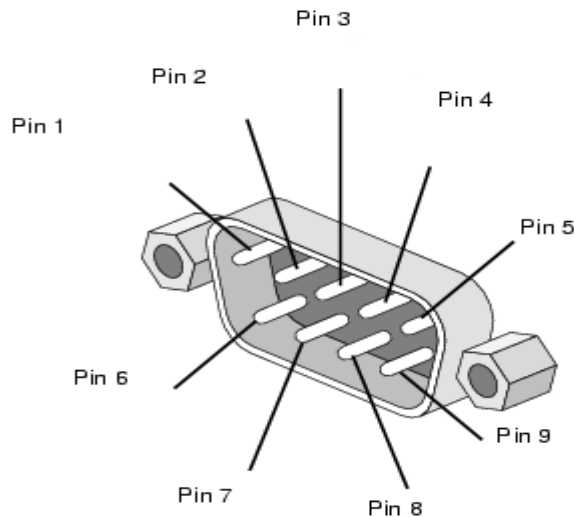
Tab. 1.3 Přenosové rychlosti přijímače a vysílače pro přenos 8 datových bitů

Přenosová rychlost [b/s]	Doba přenosu 1 bitu [μs]	Doba přenosu 1 bajtu [ms]
1200	833	8,33
2400	416	4,16
4800	208	2,08
9600	104	1,04
19200	52	0,520
38400	26	0,260
115200	8,6	0,086

Na přijímač i vysílač jsou kladeny velké nároky na přesné dodržení dob přenosů jednotlivých bitů. Odchylka pro přenos jednoho bitu by neměla překročit 2 %, jelikož pro každý bit se tyto odchylky sčítají, tak při přenosu datových bajtů se tato odchylka může přiblížit 20 %. Další navýšení ještě vzniká při identifikaci start bitu. Takže celková odchylka by se mohla vyšplhat až na nějakých 26 %, což není zanedbatelné číslo. Taková velká odchylka od původně nastavené přenosové rychlosti může vést k chybnému přenosu, někdy také označovaném *frame error*.

1.1.3 Základní zapojení konektoru a popis signálů

Pro sériovou komunikaci se nejčastěji používá 9 pinový konektor zobrazený na obr. 1.1 [1]. Na obrázku je ukázka konektoru v provedení samec, který nejčastěji nalezneme na panelu konektorů osobních počítačů. Pro většinu základních komunikací se ovšem využívají pouze tři piny a to *TxD*, *RxD* a *Ground*. Zbývající piny se využívají při spojení dvou počítačů nebo pro rychlou komunikaci s modemem. Jednotlivá zapojení pinů se liší podle aplikace, ve které je uskutečněna sériová komunikace. Například při komunikaci, při které nejsou očekávána žádná zpětná data, je možné zapojit pouze dva vodiče *TxD* a *Ground*. Ovšem nejčastější zapojení je obousměrná komunikace, při které je nutno připojit všechny tři vodiče *TxD*, *RxD* a *Ground*. Přehled označení všech pinů a jejich stručný popis je přehledně uveden v tab. 1.4 [3].



Obr. 1.1 Význam jednotlivých pinů konektoru DB-9/DE-9

Tab. 1.4 Označení a popis pinů

	Druh signálu	Popis signálu
Pin 1	DCD - Data Carrier Detect	Detekce nosné. Modem oznamuje terminálu, že na telefonní lince detekoval nosný kmitočet
Pin 2	RxD – Receive Data	Tok dat z modemu (DCE) do terminálu (DTE)
Pin 3	TxD – Transmit Data	Tok dat z terminálu (DTE) do modemu (DCE)
Pin 4	DTR – Data Terminal Ready	Terminál připraven na komunikaci
Pin 5	SGND – Signal Ground	Signálová zem
Pin 6	DSR – Data Set Ready	Modem připraven na komunikaci
Pin 7	RTS – Request to Send	Terminál oznamuje, že komunikační cesta je volná
Pin 8	CTS – Clear to Send	Modem oznamuje, že komunikační cesta je volná
Pin 9	RI – Ring Indicator	Indikátor zvonění. Modem detekoval signál zvonění

1.1.4 Připojení RS-232 na TTL logiku

Většina integrovaných obvodů pracuje s TTL logikou. TTL logika je standard používaný v digitální technice vycházející z použití bipolárních křemíkových tranzistorů. Napěťová úroveň odpovídající logické 1 je přibližně 5 V. Logická 0 je vyjádřena napětím 0 V. Obě tyto napětí se uvažují v určité toleranci. Rozmezí napětí 0 až 0,8 V je vyhodnoceno jako logická 0. Napěťovým hodnotám 2,0 až 5,0 V odpovídá logická 1. Napětí 0,8 až 2,0 V leží v tzv. zakázaném pásmu, ve kterém není přesně definované chování obvodových funkcí pracujících s TTL logikou. Zařízení kompatibilní s TTL logikou musí dále splňovat definované výstupní hodnoty napětí pro obě logické úrovně. Na výstupu zařízení musí být splněna tolerance napětí pohybující se mezi 2,7 až 5 V pro logickou 1. Logická 0 musí být na výstupu vyjádřena v rozmezí 0 až 0,3 V. V současnosti se napěťové hladiny snižují k hodnotám 3,3 až 1,2 V. Moderní trend snižování spotřeby zařízení a zvyšování integrace vede k těmto nižším napěťovým úrovním. Pro nízkonapěťové logické úrovně již neplatí žádný definovaný standard, který by vymezoval hodnoty pro logické úrovně.

Pro připojení sériového rozhraní RS-232 k nějakému obvodu, ve kterém by mělo docházet ke zpracování přijatých dat, je nutné nejprve převést napěťové úrovně do TTL logiky. Jelikož napěťové úrovně RS-232 nejsou slučitelné s žádnou logikou je tento převod nutný provést pro jakoukoliv používanou logiku. Pro převod se využívá převodník úrovní, který převádí napěťové úrovně sériového rozhraní RS-232 na napěťové úrovně nejčastěji TTL logiky.

Pro tento převod je nejpoužívanější převodník, který na trh dodává firma MAXIM a jedná se o jejich populární integrovaný obvod MAX232, který se dnes vyskytuje téměř ve všech komerčních zařízeních využívající sériovou linku. MAX232 je levný a velmi používaný převodník úrovní RS-232 na úrovně odpovídající TTL logice. Jeho nespornou výhodou je, že potřebuje pouze napájecí napětí +5 V oproti jiným převodníkům, které potřebují vyšší napájecí napětí, pohybující se většinou kolem ± 15 V. Firma MAXIM využila teorii spínaných nábojových měničů a vytvořila tak obvod, který je schopen i při napájení +5 V vytvořit patřičné napěťové úrovně potřebné pro sériovou linku. Zbývající napětí si tento obvod vytváří pomocí 4 externě zapojených kondenzátorů. Samozřejmě, že přesnost a stálost vytvářeného napětí záleží na kvalitě použitých kondenzátorů. Obvod konverguje logickou 0 na +3,15 V a logickou 1 na -3,15 V. Obsahuje 2 převodníky RS-232 na TTL a 2 převodníky TTL na RS-232 [3].

1.2 Sériové rozhraní RS-485

Komunikace pomocí sběrnice má své omezení ve vzdálenosti, na kterou lze data přenášet a také v nemožnosti větvení. Od zařízení běžně nebývá galvanicky oddělena, což znemožňuje její používání v oblasti průmyslových aplikací. Z těchto důvodů se používají v průmyslu spíše sériové linky RS-485 a nebo RS-482. Přenosový komunikační standard RS-485 byl definován v roce 1983 institucí EIA (Electronics Industries Association). RS-485 je specifikace dvoudrátového poloduplexního multibodového sériového spoje [8].

Sériová linka RS-485 může být vedena až na vzdálenost 1600 m a lze tyto linky větvit na více linek, ovšem maximální počet vysílačů a přijímačů je 32, dle použitého ovladače může být počet účastníků až 128. Maximální přenosová rychlost je 10 Mb/s, ovšem tuto rychlost podstatně ovlivňuje délka vedení, na kterém probíhá přenos dat. Sběrnice dat musí být ukončeny z každé strany impedancí 110 Ω . Výsledná impedance tedy bude 55 Ω , jelikož se jedná o paralelní zapojení. Každý ze signálů linky je přenášen po dvojici vodičů, nejlépe v

provedení kroucený pár. Vodiče označované A a B jsou vysílačem buzeny v protifázi a přijímač vyhodnocuje jejich napěťový rozdíl. Tímto principem se odstraní součtově (aditivní) rušení. Dalším rozdílem oproti sériové lince RS-232 jsou nižší používané napěťové úrovně, typický rozdíl mezi vodiči je 2 V. Přenos dat se uskutečňuje pomocí 7 nebo 8 bitových rámců se startbitem, 1 nebo více stopbity a případně i paritním bitem. Linka RS-485 používá jeden pár vodičů pro oba směry toku dat, je nutné tedy směr komunikace přepínat, což může způsobovat problémy. Většinou je nutné přepínání směru komunikace zařídit softwarově. Typicky používané přenosové rychlosti pro sériovou linku RS-485 jsou 9,6 kBd a 19,2 kBd. I když je spolehlivost přenosu po sériové lince RS-485 podstatně vyšší než u RS-232 nebude toto rozhraní v řešení této práce využito.

1.3 Univerzální rozhraní USB

Pomocí univerzálního rozhraní USB se dnes k počítači připojuje většina externích zařízení. Na vývoji USB rozhraní pracovalo několik firem (Compaq, DELL, IBM, Intel, Microsoft, NEC a Northern Telecom) a první oficiální dokument vznikl na konsorciu těchto firem v roce 1995. Tento dokument standardizoval základní parametry tohoto rozhraní. Mezi hlavní cíle patří několik faktorů, které standard USB vyřešil. Hlavní myšlenka byla zastavit vzrůstající počet různých konektorů na počítači, přes které se připojovalo mnoho různých periférií. Tento fakt vedl k zavedení jednoduchého rozhraní nenáročného z hlediska uživatele. Dále velmi snadné připojení zařízení k počítači, konektory běžně přístupné z vnějšku počítače. Přes stejný typ kabelu je možné připojit různé typy zařízení (myš, klávesnice, vnější disky, paměti flash). Připojená zařízení jsou napájena přímo přes kabel. Mezi další výrazné výhody patří možnost připojení zařízení při spuštění počítače a také z ekonomického hlediska je připojení pomocí USB levné. Existují tři verze specifikací: USB 1.0, USB 1.1, USB 2.0. V současnosti se zpracovává specifikace USB 3.0, která by měla v blízké době nahradit svého předchůdce USB 2.0 a přinést celou řadu nových vylepšení.

1.3.1 Parametry USB

Jedná se o sériové rozhraní dosahující přenosové rychlosti až 480 Mb/s (ver. 2.0). Umožňuje připojit zařízení na relativně velkou vzdálenost dosahující až 5 m. Při použití hubů je možné na sběrnici USB připojit velký počet zařízení až 127. Rozhraní podporuje Plug&Play, což znamená, že je možné zařízení připojit za běhu počítače aniž by došlo k poškození připojovaného zařízení nebo osobního počítače. Po připojení dochází k automatické identifikaci připojeného zařízení. Při prvním připojení dojde k nainstalování ovladačů potřebných pro správný chod připojeného zařízení. Při dalším připojení je zařízení přijato jako známý prvek a automaticky je připraveno k používání. Připojovaná zařízení je možné napájet přímo z USB konektoru běžně lze odebrat 100 mA pro speciální aplikace až 500 mA. Komunikace je poloduplexní, vždy hostitelem iniciovaná, nešifrovaná, chráněná cyklickou redundantní kontrolou, přenášena po metalickém diferenciálním vedení.

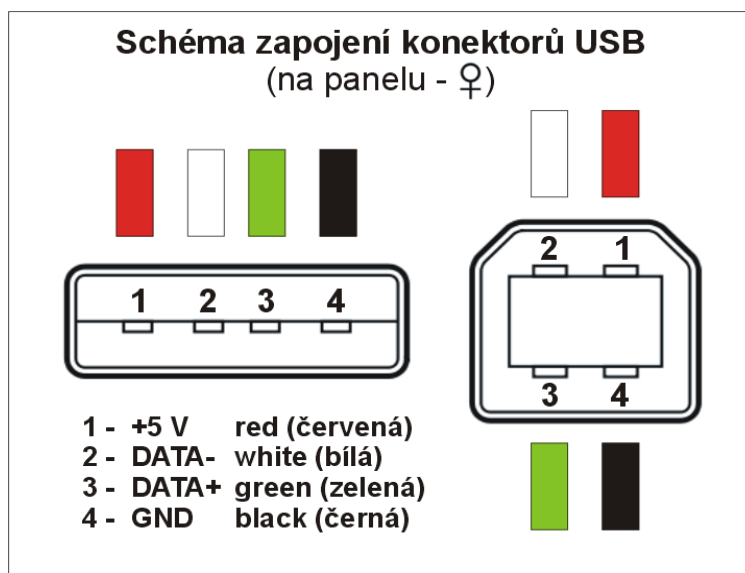
1.3.2 Konektory a kabely

USB je sériová sběrnice, data jsou tedy přenášena po jednotlivých bitech a to diferenčně pro snížení vlivu rušení. Datové vodiče přenáší vzájemně negované signály. Napěťové úrovně jsou v rozsahu od 0 V až 3,3 V. Význam vývodů konektoru USB je uveden v tab. 1.5 [4].

Tab. 1.5 Vývody USB konektoru

Číslo vývodu	Význam
1	+5 V (napájecí napětí)
2	Data + (přímá data)
3	Data – (negovaná data)
4	GND (zem)

USB využívá dva typy konektoru. Jeden je označován písmenem A, je plochý a zdířku na něj je možné nalézt na skříní osobního počítače. Druhý typ konektoru je označován písmenem B, je menší, téměř čtvercový a příslušná zdířka se nachází na perifériích jako je tiskárna, scanner atd. Konektory A i B jsou uvedeny na obr. 1.2. Nejpoužívanějším kabelem je tedy kabel typu A-B na kterém jsou oba konektory zástrčkami. Zvláštní typ kabelu je kabel typu A-A v konfiguraci zástrčka-zásuvka, který se používá k prodloužení sběrnice. Všeobecně se ovšem použití prodlužovacích kabelů nedoporučuje, jelikož dochází ke zhoršení přenosových vlastností [4].



Obr. 1.2 Konektory USB

1.3.3 Verze USB a přenosové rychlosti

USB zařízení pracují v různých verzích, které se od sebe liší hlavně přenosovými rychlostmi. Přehled přenosových rychlostí pro jednotlivé standardy je uveden v tab. 1.6 [4].

Tab. 1.6 Přenosové rychlosti standardu USB

Rychlost	Přenosový výkon [Mb/s]	Standard USB
Low Speed	1,5	USB 1.1/2.0
Full Speed	12	USB 1.1/2.0
High Speed	480	USB 2.0

Uvedené přenosové výkony je možné dosáhnout při použití jednoho zařízení. Při připojení více zařízení se tento výkon rozděluje mezi jednotlivé zařízení. Použitou přenosovou rychlost definují samotná zařízení dle zapojení rezistoru pull-up 1,5 kΩ. Při připojení pull-up rezistoru mezi vodiče D- a napětí 3,3 V se definuje zařízení s rychlostí low speed. Full speed zařízení se vyznačuje připojením pull-up rezistoru mezi datový vodič D+ a napětí 3,3 V. High speed se řeší stejným zapojením jako Full speed s tím rozdílem, že změna rychlosti se provádí programově. V případě, že připojíme zařízení s rozhraním USB 2.0 na rozhraní podporující pouze USB 1.1, dojde ke snížení přenosové rychlosti, tak aby zařízení mohlo fungovat s upozorněním, že zařízení může pracovat s větší přenosovou rychlostí [4].

Verze USB 1.1 přinesla přenosovou rychlost 12 Mb/s, což umožnilo připojovat přes USB rozhraní externí zařízení jako klávesnice a myši. Připojení pro vypalovací mechaniky nebo pro digitální fotoaparáty bylo na hranici použitelnosti. Z těchto důvodů se zavedl standard USB 2.0, který zachoval zpětnou kompatibilitu s USB 1.1 a navíc přinesl potřebnou přenosovou rychlost pro většinu současných zařízení. V současné době mají všechny moderní osobní počítače USB rozhraní v několika portech. USB rozhraní nachází uplatnění i v televizní technice, kde pomocí USB je možná komunikace s TV přijímačem či DVD přehrávačem. Vyskytuje se označení USB High-speed, které má stejný význam jako USB 2.0. Přínosem pro uživatele je podstatně větší šířka pásma, která umožňuje připojit více externích zařízení než v případě USB 1.1.

Jak již bylo zmíněno v úvodu této kapitoly, na svět přichází nový standard USB 3.0 označovaný také jako SuperSpeed USB. Tento standard by měl na trhu nahradit dosavadní dominantní postavení standardu USB 2.0. Hlavním přínosem má být více než desetinásobná rychlost oproti současnému stavu. Rychlost přenosu by se tedy měla pohybovat kolem 5 Gb/s rychlost, samozřejmě bude limitována výkonem mikroprocesorů, tedy v reálných aplikacích nebude možné zatím tuto rychlost naplno využívat. Opět by měla být zachována zpětná kompatibilita se standardy USB 1.1 a 2.0. Na trhu by se tento standard měl objevit v roce 2010 [5].

1.3.4 Přenos Dat

USB je tzv. jednomasterová sběrnice, což znamená, že všechny aktivity vycházejí z osobního počítače. Data se posílají v paketech, jejichž délka je 8 až 64 bajtů. Princip komunikace spočívá v tom, že počítač může požadovat data od připojeného zařízení, ale žádné zařízení nemá možnost začít vysílat data samo od sebe. Veškerý přenos dat se uskutečňuje v přesně daných časových rámcích o délce 1 ms. Uvnitř těchto rámců se zpracovávají data pro několik zařízení, dokonce se můžou v rámci objevit data s různými přenosovými rychlostmi. Při komunikaci počítače s více připojenými zařízeními zajišťuje rozdělení dat rozbočovač, který má také na starosti správné rozdělování dat na zařízení, které mohou pracovat s různými rychlostmi přenosu. Připojené zařízení se musí synchronizovat na datový tok, jelikož hodinový signál není přenášen po zvláštní lince, na začátku každého paketu je vyslán synchronizační vzorek, který má za úkol synchronizovat taktovací generátor přijímače. Synchronizační vzorek je tvořen sedmi nulami následovanými jedničkou v kódování NRZI (Non Return To Zero). Kódování NRZI má následující logiku: nuly v datech vedou ke změně úrovně, jedničky nechávají úroveň beze změny. Pro udržení synchronizace následuje-li po sobě šest jedniček v datech, přidá vysílač automaticky jednu 0 tzv. bit-stuffing, aby se vynutila změna úrovně. Tato operace je nutná pro obnovení hodinového kmitočtu z datového signálu. Přijímač poté musí tuto nadbytečnou nulu odstranit tzv. bit-unstuffing. Po odeslání synchronizačního vzorku následuje PID (Packet Identifier) udávající typ paketu, formát paketu a způsob detekce chyby aplikovaný v paketu. PID lze

rozdělit do čtyř skupin: token, data, potvrzení přenosu a speciální. Kódování a dekódování signálů je čistě hardwarová záležitost [4]. Zařízení USB obsahuje jednotku zvanou SIE (Serial Interface Engine), která přebírá vlastní práci. K výměně dat mezi SIE a zbytkem zařízení slouží vyrovnávací paměti FIFO (First In First Out). FIFO jsou vyrovnávací paměti, které mohou postupně přijímat a vydávat data podobně jako posuvné registry. Připojený mikrořadič tedy potřebuje jen přečíst data z FIFO a jiná data do FIFO zapsat. Všechno ostatní vyřídí SIE. Ve většině případů je SIE součástí mikrořadiče USB. Zařízení USB má obecně několik pamětí FIFO, jejichž prostřednictvím je možno přenášet data [9].

USB standard má definovány čtyři typy přenosu dat. Řídicí přenos tzv. Control transfer se využívá k řízení hardware. Používá automatické zabezpečení chyb a pracuje s vysokou přenosovou rychlostí. Další typ přenosu se označuje Interrupt-Transfer neboli přenos přes přerušení. Tento typ přenosu používají zařízení, která periodicky vysílají menší množství dat. Počítač se periodicky dotazuje na nová data. Tento typ přenosu je typický pro myši nebo klávesnice. Hromadný přenos (Bulk-Transfer) je dalším typem přenosu, který je vhodný pro přenos velkých množství dat se zabezpečením. Používá se u tiskáren, skenerů a externích ZIP. Poslední typ přenosu je izochronní přenos (Isochronous-Transfer). Používá se pro přenosy s velkým množstvím dat s definovanou rychlostí bez zabezpečení. S tímto typem je možné se setkat u externích zvukových karet.

1.3.5 Rozpoznávání zařízení

USB rozhraní podporuje Plug&Play. Plug and Play (v překladu „připoj a hraj“) je počítačová technologie umožňující jednodušší rozpoznávání a konfigurace hardware. Takže při připojení zařízení do USB konektoru musí být automaticky rozpoznáno operačním systémem. Po připojení se operační systém zeptá na parametry zařízení. Podle těchto informací poté proběhne instalace potřebných ovladačů pro správnou funkci zařízení. Po úspěšné instalaci ovladačů je zařízení ohlášeno jako připojené zařízení a obdrží svou sběrníkovou adresu. Celý tento postup by měl operační systém zabezpečit automaticky bez uživatelské asistence a bez nutnosti restartování počítače. V případě potíží s instalací systém požaduje příslušné ovladače zařízení.

Pro připojení několika zařízení k jednomu portu počítače se používá hub-rozbočovač. Existují dva typy hubů. Kořenový hub obsahuje každý osobní počítač, umožňuje mu realizaci dvou (i více) portů. Druhým typem je externí hub, který umožňuje připojit 4 i více portů pro spojení se zařízeními či dalšími huby. Každý hub ovšem zvyšuje zpoždění signálu, které nesmí překročit jistou maximální hodnotu. USB standard je tedy omezen počtem připojených zařízení, to je konkrétně 127 zařízení, které může rozdělovat až 7 hubů. Hub také zásobuje zařízení napájecím napětím [4].

2 ŘÍZENÍ LABORATORNÍHO ZDROJE

Tato kapitola je zaměřena na laboratorní zdroj Manson série SDP-2405 pro který bude navržen komunikační modul. Komunikace bude probíhat přes standard USB s osobním počítačem v grafickém programovém prostředí VEE Pro od firmy Agilent Technologies. Zdroje série SDP mají možnost úplného dálkového programování a zápisu dat. Funkce dálkového programování umožňuje plné ovládání funkcí a zpětné čtení dat. Zdroje jsou dodávány s potřebným softwarem pro zápis dat. Tento laboratorní zdroj standardně komunikuje přes rozhraní RS-232 nebo je možné použít průmyslové rozhraní RS-485. Pro komunikaci využívá sadu vlastních příkazů pomocí, kterých lze laboratorní zdroj ovládat s dodaným softwarem. Sada používaných příkazů je neslučitelná s obvykle používanými příkazy v prostředí VEE Pro a také není kompatibilní se standardem SCPI. Laboratorní zdroj Manson je také možné připojit k různým grafickým programovacím prostředím. Mezi tyto prostředí patří i program VEE Pro od firmy Agilent Technologies, do kterého je možné tento zdroj také implementovat s pomocí dodávaných ovladačů.

2.1 Technické parametry zdroje

Jak bylo zmíněno, komunikační modul bude navržen pro konkrétní zdroj od firmy Manson. V tab. 2.1 jsou uvedeny základní parametry napájecího zdroje důležité pro návrh modulu. Propojení napájecího zdroje s osobním počítačem pomocí RS-232 využívá data ve formátu ASCII (American Standard Code for Information Interchange), jde o kódovou tabulku, která definuje znaky anglické abecedy a znaky používané v informatice. Pro komunikaci se nepoužívá žádný paritní bit, tedy není použita žádná kontrola přijatých dat. Rámec dat obsahuje 8 datových bitů na závěr je použit 1 stop bit. Doporučuje se pro komunikaci s tímto zdrojem využívat přenosovou rychlost 9600 bps. Tento zdroj umožňuje využít i propojení pomocí průmyslového rozhraní RS-485. Pomocí RS-485 lze připojit až 31 zdrojů je ovšem potřeba zapojit adaptér RS232/485. Požadavky na systém jsou velmi malé. To je ideální v případě, kdy je potřeba různého nastavení výstupních parametrů a různých časů provozu při opakujících se testech s více zdroji. Tlačítka na čelním panelu umožňují veškeré programování a nastavení výstupu jako u samostatného zdroje. Dodávaný software umožňuje časové programování, zápis dat a nastavení napětí nebo proudu. Uvedené požadavky na osobní počítač jsou v dnešní době splněny u všech běžně dostupných osobních počítačů. Omezení pro software a také správnou komunikaci může zapříčinit použití nového operačního systému Windows Vista, který není ve výčtu podporovaných operačních systémů. Laboratorní zdroj má dále zabudovanou ochranu proti přepětí. Při překročení výstupního napětí o 10 % nastavené hodnoty dojde ke spuštění ochrany a dojde k odpojení výstupu přístroje. Na displeji se objeví varovné hlášení FAULT. Pro znovu uvedení do provozu je nutné zdroj odpojit od síťového napájení a posléze znovu připojit. Další zabudované zařízení ve zdroji je „bzučák“, který informuje o stavu přehřátí, přetížení anebo pokud sepne přepěťová ochrana. Na obr. 2.1 je zobrazený laboratorní zdroj, pro který bude navržen komunikační modul. Na obr. 2.2 je zobrazena zadní část laboratorního zdroje, z které je patrné rozmístění konektoru pro připojení do napájecí sítě a hlavně konektory pro dvě komunikační rozhraní RS-232 a RS-485 [10].

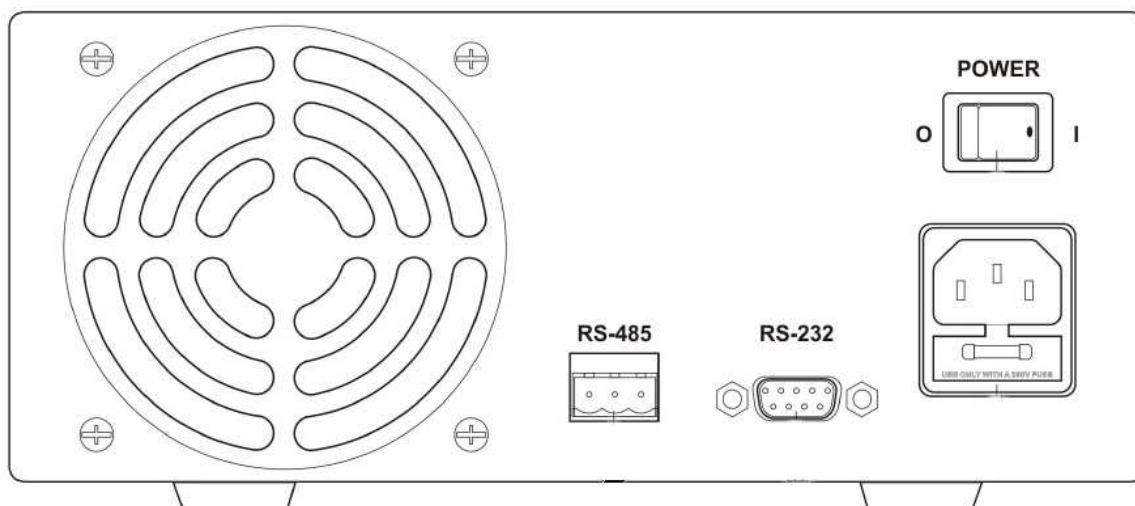
Tab. 2.1 Základní parametry zdroje

Parametr	Zdroj Manson SDP2405
Výstupní napětí	1-40* V _{ss}
Výstupní proud	0-5 A
Komunikační rozhraní	RS232 (pro jeden přístroj) RS485 (až pro 31 přístrojů)
Funkce dálkového programování	Plné ovládání funkcí a zpětné čtení dat
Zápis dat	Možný, s dodaným softwarem
Přenosová rychlost	9600 b/s
Zdroj napájení	100-240 V, 50/60 Hz
Pracovní teplota	0 - 40 °C
Učinnost	≥ 0,9
Zvlnění a šumy regulovaného napětí	≤ 30 mV _{p-p}
Jmenovitý výstupní výkon	200 W*
Krok regulace	10 mV

*Starší verze těchto zdrojů umožňují pouze 24 V a výkon 120 W



Obr. 2.1 Laboratorní zdroj Manson SDP-2405



Obr. 2.2 Zadní část laboratorního zdroje Manson SDP-2405

2.2 Příkazy laboratorního zdroje

Laboratorní zdroj pro komunikaci s počítačem má nadefinováno několik základních příkazů, podle kterých lze zdroj pohodlně ovládat přes grafické programovací prostředí. Tento zdroj používá příkazy dvou typů. První typ příkazů jsou příkazy, dle kterých lze nastavit hodnoty napětí, proudu nebo sepnout výstupní svorky, tedy příkazy, které nevrací žádná data zpět. Druhý typ příkazů má charakter dotazu. Tyto příkazy vrací data, která jsou po něm požadována, např. nastavená hodnota napětí či proudu. Grafické programové prostředí od firmy Agilent VEE Pro, ke kterému je zdroj Manson připojen, může použít ke komunikaci libovolný příkaz. Nicméně pro komunikaci s jinými laboratorními přístroji je vhodnější využívat jiné příkazy, které jsou obvykle definované standardem SCPI. Při práci v grafickém prostředí je pohodlnější používat stejné typy příkazů pro všechny používané typy měřících přístrojů. Komunikační modul, který bude navržen v této práci, bude provádět převod příkazů používaných v tomto programovacím prostředí tedy vycházející ze standardu SCPI na příkazy, které očekává laboratorní zdroj. Laboratorní zdroj reaguje na příkazy uvedené v tab. 2.2.

Po připojení zdroje k osobnímu počítači je nutné poslat příkaz `SESS00`, který otevře komunikaci mezi zdrojem a osobním počítačem. Po zaslání tohoto příkazu přestane laboratorní zdroj reagovat na tlačítka, která jsou umístěna na předním panelu zdroje. Zdroj lze tedy ovládat pouze pomocí vhodných příkazů viz tab. 2.2. Poté je možné nastavit napětíovou či proudovou výstupní pojistku. Tedy limitní hodnoty, které laboratorní zdroj bude dodávat do připojeného obvodu po sepnutí výstupních svorek. K nastavení těchto pojistek se musí použít příkazy ve tvaru `VOLT00XXX` pro nastavení napětí a pro nastavení proudu příkaz ve tvaru `CURR00YYY`. Napětí na laboratorním zdroji lze volit v rozmezí 0 až 40 V u novějších typů, u starších typů je to v rozmezí 0 až 24 V, přičemž napětí se udává ve tvaru `XX,X V` to znamená, že pokud má být na laboratorním zdroji nastaveno napětí 15 V je potřeba poslat příkaz ve tvaru `VOLT00150`. Pokud byla před posláním tohoto příkazu otevřena komunikace, zdroj nastaví výstupní napětíovou pojistku na 15 V, aby se požadované napětí objevilo na výstupních svorkách, je potřeba zaslat příkaz `SOUT000`, který slouží k sepnutí výstupních svorek. Po obdržení tohoto příkazu je na výstupních svorkách zdroje příslušné napětí 15 V s tolerancí danou zdrojem. Podobný postup příkazů platí pro nastavení výstupní proudové pojistky. Výstupní proud na laboratorním zdroji lze nastavit v rozmezí 0 až 5 A. Při nastavení

0 A a dostatečně velkého výstupního napětí přejde laboratorní zdroj do režimu zdroje konstantního proudu. Proud se zadává odlišně oproti napětí a to ve tvaru Y,YY A. Pokud se má na zdroji nastavit výstupní proudová pojistka 100 mA je potřeba do laboratorního zdroje poslat příkaz ve tvaru CURR00010. Platí obdobně jako u napětí, že pokud byla již otevřena komunikace mezi zdrojem a osobním počítačem dojde k nastavení výstupního proudu a po zaslání příkazu SOUT000 dojde k aktivaci výstupních svorek a na výstupu zdroje bude možné odebírat příslušný proud s danou tolerancí. Pro deaktivaci výstupních svorek je potřeba zaslat příkaz ve tvaru SOUT001. Zdroj odpojí výstupní svorky ovšem nastavení napěťové a proudové pojistky ponechá na původních hodnotách. Pro ukončení komunikace je potřeba do laboratorního zdroje zaslat příkaz ve tvaru ENDS00. Poté je možné opět ovládat laboratorní zdroj i pomocí tlačítek na předním panelu. Není nutné posílat příkazy pro zahájení a ukončení komunikace mezi osobním počítačem a laboratorním zdrojem, nicméně při vynechání těchto příkazů je možné se zdrojem libovolně manipulovat pomocí předního panelu, tedy měnit nastavené hodnoty napěťové a proudové pojistky.

Při práci s druhým typem příkazů, které vrací data zpět do grafického programovacího prostředí, se používají jiné typy příkazů. Při žádosti o přečtení neboli změření aktuální hodnoty výstupního napětí nebo proudu se používá příkaz GETD00, který vrátí aktuální hodnoty napětí a proudu na výstupních svorkách. Pokud je otevřená komunikace mezi laboratorním zdrojem a osobním počítačem, tak při zaslání příkazu GETD00 se mohou vrátit data ve dvou různých datových typech. Pokud použijeme pro vyčtení tohoto příkazu datový typ *string*, pak obdržíme data ve formátu 021903570, pokud použijeme datový formát *real 64*, obdržíme data ve tvaru 21,90357 M. V této číselné kombinaci je obsažena aktuální hodnota výstupního napětí a zároveň aktuální hodnota výstupního proudu. První čtveřice číslic reprezentuje aktuální hodnotu výstupního napětí. V uvedeném příkladu jsou data 0219, která reprezentují hodnotu napětí 2,19 V. Následující čtveřice čísel vyjadřuje aktuální hodnotu výstupního proudu. V uvedeném příkladu jsou tedy data 0357, což odpovídá hodnotě výstupního proudu 0,357 A. Poslední číslo nese informaci o tom, v jakém módu laboratorní zdroj pracuje. Pokud je poslední číslice 0, laboratorní zdroj pracuje jako zdroj konstantního napětí. Tedy na výstupních svorkách zdroj udržuje konstantní hodnotu napětí s rostoucím i klesajícím výstupním proudem. Druhá varianta je číslice 1, potom zdroj pracuje v módu konstantního zdroje proudu. Tedy do připojené zátěže by v tomto módu měl jít konstantní proud. Tento fakt je omezen velikostí vnitřního odporu laboratorního zdroje. Napěťový zdroj má ideální hodnotu vnitřního odporu rovnu 0Ω . Naopak proudový zdroj má ideální hodnotu vnitřního odporu zdroje rovnu $\infty \Omega$. Obdobný příkaz pro vrácení dat z laboratorního zdroje je GETS00. Tento příkaz vrací hodnoty nastavených napěťových a proudových pojistek. Formát posílaných dat ze zdroje je velmi podobný. Skládá se pouze z 6 číslic. První tři číslice obsahují hodnotu nastavené napěťové pojistky. Další tři číslice informují o nastavené hodnotě proudové pojistky. Při tomto dotazu laboratorní zdroj nevrací informaci o módu, ve kterém pracuje. Obě nastavené pojistky stanovují maximální výstupní výkon, který lze odebírat v daném okamžiku z laboratorního zdroje Manson. Poslední příkaz, který bude více rozebrán v této části je příkaz GPAL00. Tento příkaz vrací řetězec informací z displeje. Součástí řetězce jsou v podstatě kompletní informace zobrazované na displeji laboratorního zdroje, tedy aktuální hodnoty napětí, proudu, výkonu, stav výstupních svorek, nastavení napěťové či proudové pojistky, mód zdroje (V-konstantní, I-konstantní), zobrazení chyby FAULT, informace o stavu komunikace mezi zdrojem a osobním počítačem. Z těchto hodnot bude využita hlavně informace o stavu výstupních svorek.

Tab. 2.2 Přehled příkazů používaných při komunikaci s laboratorním zdrojem

Činnost	Command Code	Popis
Otevření komunikace	SESS <address>	Připraví zdroj na komunikaci
Uzavření komunikace	ENDS <address>	Uzavře komunikaci se zdrojem
Změna RS-232/RS-485	CCOM <address> <RS>	Změna mezi RS-232 (RS=0) a RS485 (RS=1)
Adresa RS-485	GCOM <address>	Získá adresu RS-485
Max. U a max. I	GMAX <address>	Získá max.U a max. I ze zdroje
Horní mez napětí	GOVP <address>	Získá horní mez napětí ze zdroje
Vyčtení hodnoty U a I	GETD<address>	Aktuální hodnota U a I na výstupních svorkách zdroje
Vyčtení nastavené hodnoty U a I	GETS <address>	Nastavená hodnota U a I
Všechny uložené hodnoty v paměti	GETM <address>	Vyčte z paměti všechny uložené hodnoty U a I
Všechny uložené časové programy	GETP <address>	Vyčte z paměti všechny časové programy
Display informace	GPAL <address>	Vrátí všechny informace zobrazené na displeji
Nastavení napětí	VOLT <address>XXX	Nastavení napětí ve tvaru XX,X V
Nastavení proudu	CURR <address>YYY	Nastavení napětí ve tvaru Y,YY A
Připojení výstupních svorek	SOUT <address>0	Aktivuje výstup zdroje
Odpojení svorek	SOUT <address>1	Deaktivuje výstup zdroje

2.3 Standard SCPI

Moderní měřicí přístroje či napájecí zdroje využívané pro automatizované měření jsou většinou ovládány sadou příkazů, které splňují standard SCPI. Tento standard specifikuje společnou syntaxi a formát dat, první verze byla vydaná v roce 1990. Standard zavádí obecné příkazy, které by měli být použitelné na libovolném zařízení, které má ve své specifikaci uvedeno kompatibilitu s SCPI. Příkazy jsou seskupeny do subsystému. Také je definováno několik tříd zařízení. Napájecí zdroje jsou zařazeny do třídy nazývané *DCPSUPPLY*. SCPI příkazy jsou ASCII textové řetězce, které jsou zasílány do přístrojů přes fyzickou vrstvu. Příkazy jsou série jednoho nebo více slov, které se dají dále rozdělit na povinnou a nepovinnou část příkazu, přičemž obě skupiny mohou být ve zkrácené nebo úplné verzi. Reakce na příkazy jsou opět většinou ASCII řetězce nebo se dá využít binární formát dat [13]. Laboratorní zdroj Manson má odlišnou sadu příkazů, která byla rozebrána v předchozí části.

Navrhovaný komunikační modul bude obsahovat kromě převodu na USB standard také převod nejdůležitějším příkazů, tak aby odpovídali požadavkům standardu SCPI. V tab. 2.3 jsou uvedeny vybrané příkazy ze standardu SCPI, které budou implementovány do komunikačního modulu, a bude k nim přiřazen odpovídající příkaz z instrukční sady laboratorního zdroje. Při zapojení laboratorního zdroje pomocí komunikačního modulu nebude možné zasílat jiné příkazy, než jsou uvedeny v tab. 2.3. Tím tedy bude omezena původní instrukční sada zdroje, ale nově využitelné příkazy budou naopak splňovat požadavky standardu SCPI. Povinná část příkazu je psána pomocí velkých písmen, rozšířené části příkazů jsou naopak zapsány pomocí malých písmen. Nepovinné části jsou vloženy do hranatých závorek. Číselné hodnoty jednotlivých příkazů jsou potom umístěny do složených závorek. Detailní syntaxe jednotlivých příkazů bude rozebrána v popisu obslužného programu pro komunikační modul.

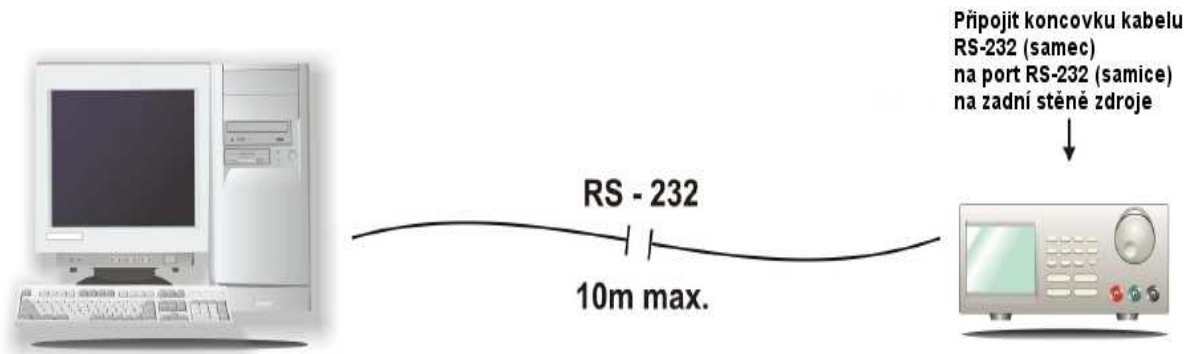
Tab. 2.3 Přehled příkazů pro komunikační modul

SCPI command	Manson command	Popis
*rst	-	tovární nastavení přístroje
*idn?	-	identifikace přístroje
MEASure:CURRent?	GETD00	aktuální hodnota proudu
MEASure:VOLTage?	GETD00	aktuální hodnota napětí
OUTPUT[:STATe] {OFF ON}	SOUT000 SOUT001	sepnutí výstupních svorek rozepnutí výstupních svorek
OUTPUT[:STATe]?	GPAL00	stav výstupních svorek
[SOURce:]CURRent {Y,YYA}	CURR00YYY	nastavení proudové pojistky
[SOURce:]VOLTage {XX,XV}	VOLT00XXX	nastavení napěťové pojistky
[SOURce:]CURRent?	GETS00	nastavená proudová pojistka
[SOURce:]VOLTage?	GETS00	nastavená napěťová pojistka

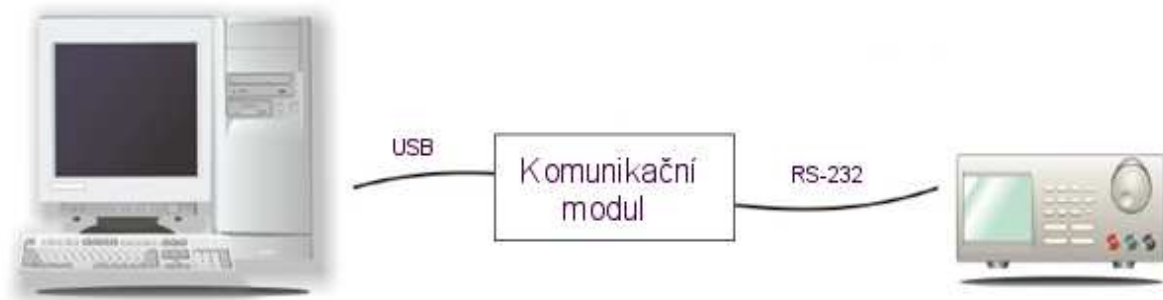
2.4 Připojení zdroje k PC

Laboratorní zdroj Manson lze k osobnímu počítači připojit pomocí dvou rozhraní a to pomocí sériového rozhraní RS-232 anebo pomocí průmyslového sériového rozhraní RS-485. Pro názornost je na obr. 2.3 zobrazeno připojení laboratorního zdroje Manson pomocí sériové linky RS-232. Maximální délka spojení je stanovena délkou kabeláže na 10 m. Na zadní stěně zdroje je sériový konektor typu samice. Tedy pro spojení s osobním počítačem se využívá standardní sériový kabel ukončený koncovkami v provedení samec – samice. Připojení laboratorního zdroje pomocí komunikačního modulu je zobrazeno na obr. 2.4. V obrázku jsou naznačena počítačová rozhraní, které budou využita při návrhu komunikačního modulu. Tedy s osobním počítačem bude modul spojený pomocí USB standardu a naopak s laboratorním zdrojem bude využito klasické spojení sériovým kabelem. Na obr. 2.5 je obdobné zapojení laboratorního zdroje jen pro případ průmyslového sériového rozhraní RS-485, které umožňuje připojení více laboratorních zdrojů Manson k jednomu počítači. Pro připojení je potřeba

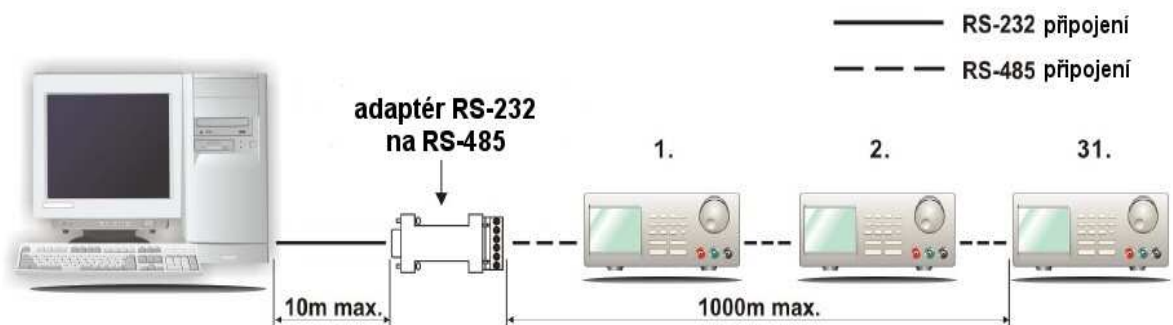
použít adaptér mezi RS-232 a RS-485, který je standardně dodáván s laboratorním zdrojem Manson. Zapojení z obr. 2.5 nebude v rámci této práce využito. Z obrázku je patrné, že pomocí sériového rozhraní RS-485 je možné připojit k jednomu osobnímu počítači až 31 laboratorních zdrojů Manson, přičemž vzdálenost mezi adaptérem a posledním připojeným zdrojem nesmí být větší než 1000 m. Připojení adaptéru k osobnímu počítači je opět realizováno pomocí sériového kabelu, kde platí opět limitní vzdálenost 10 m.



Obr. 2.3 Připojení laboratorního zdroje k PC pomocí RS-232

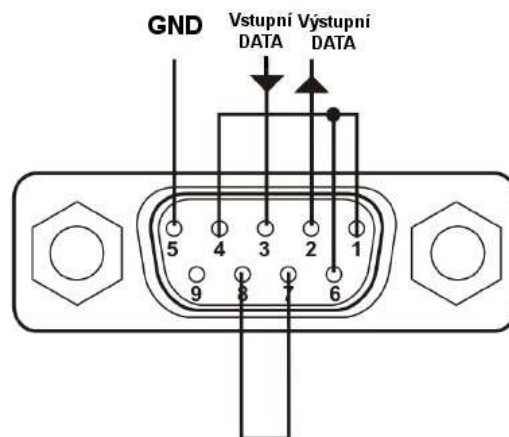


Obr. 2.4 Připojení laboratorní zdroje pomocí komunikačního modulu



Obr. 2.5 Propojení laboratorního zdroje s PC pomocí RS-485

Pro účely této práce je důležité hlavně zapojení laboratorního zdroje Manson pomocí sériové linky RS-232. Propojení s osobním počítačem je možné provést pomocí přímého sériového kabelu. Z toho plyne, že piny vyvedené na sériový konektor laboratorního zdroje odpovídají uspořádání pinů nacházejících se na osobních počítačích vybavených sériovým rozhraním RS-232. Na Obr. 2.6 je znázorněno doporučené zapojení sériového konektoru pro správnou komunikaci s laboratorním zdrojem. Zobrazené zapojení bude použito při návrhu komunikačního modulu. Laboratorní zdroj Manson zpracovává přijatá sériová data v rámcích, které obsahují 8 datových bitů. Řádková paměť zdroje má kapacitu 16 bytů. Sériový přenos probíhá asynchronně s přenosovou rychlostí optimálně 9600 baudů. Laboratorní zdroj je možné připojit i na paralelní rozhraní běžně používané pro připojení tiskáren. Je nutné dodržet správné připojení pinů, kdy všech 9 pinů nacházejících se na konektoru RS-232 má svůj odpovídající pin na paralelním konektoru umístěném v osobním počítači. Detailně je možné se s tímto zapojením seznámit v návodu k obsluze zdroje, který je dodáván s laboratorním zdrojem Manson [10]. Všechny uvedené informace o laboratorním zdroji Manson byly výhradně čerpány z návodu k obsluze. Komunikační modul, který má být navržen v této práci, nebude pracovat přes paralelní port, proto zde nebude toto zapojení rozebráno detailně.



Obr. 2.6 Doporučené zapojení konektoru RS-232

3 PŘEVOD RS-232↔USB

Cílem této kapitoly je rozebrat možnosti převodu mezi sériovým rozhraním RS-232 a standardem USB. V dnešní době je možné takový převodník přímo zakoupit jako hotové zařízení, nicméně v tomto projektu je potřeba samotný převod rozšířit o úpravu příkazů posílaných do laboratorního zdroje Manson. Z toho vyplývá, že zakoupení převodníku by nebylo elegantním řešením problému, jeho řešení by se mohlo stát zbytečně komplikovaným. Dále budou nastíněna dvě možná řešení univerzálního komunikačního modulu. Jedno řešení vede na použití obvodů FTDI pomocí kterých je možné zrealizovat převod RS-232↔USB. Druhá varianta vede na použití vhodně vybraného mikroprocesoru, který umí pracovat s oběma rozhraními. Pro praktickou realizaci bude použita druhá varianta, která bude navržena a zhotovena.

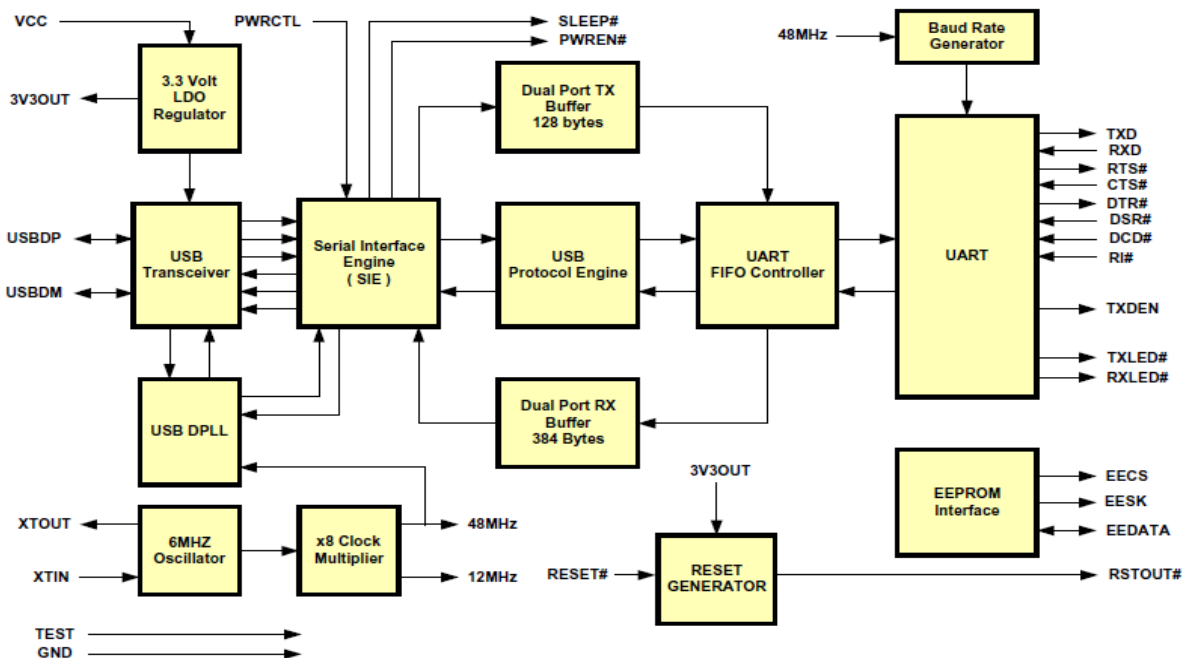
3.1 RS232↔USB s obvody FTDI

Firma FTDI Chip vyrábí obvody FT232BM a FT245BM pracující jako konvertory mezi rozhraními USB↔UART a USB↔FIFO. Pomocí těchto obvodů jsou zhotoveny již zařízení, které umí převádět data přijaté přes USB rozhraní a posílat je přes sériovou linku na připojené zařízení. Cenová relace těchto obvodů se pohybuje kolem 200 Kč za kus [4].

Tento jednočipový převodník USB↔UART podporuje plný handshake a plné rozhraní signálů modemu. To znamená, že je využit kompletní sériový přenos. Pro komunikaci využívá 7/8 bitový přenos užitečných dat používá 1 až 2 stop-bity a několik druhů parity (sudá, lichá, značená, mezerová, bez parity). Přenosová rychlost je nastavitelná v širokých mezích od 300 Bd až po 3 MBd. Jeho použití je tedy velmi univerzální, přenosová rychlost je nastavitelná dle požadavku připojovaného zařízení. Tento obvod je možné napájet přímo ze sběrnice USB, podporuje i napájení zařízení s vysokým odběrem. Obsahuje také integrovaný konvertor úrovní UART a řídicích signálů pro 5 V a 3,3 V logiku. Odpadá nutnost použití integrovaného obvodu MAX232, který provádí převod napěťových úrovní mezi TTL logikou a RS-232 rozhraním. Obvod také obsahuje integrovanou násobičku kmitočtu z 6 na 48 MHz, tedy na kmitočet používaným pro USB 2.0 rozhraní. Dále obsahuje integrovanou paměť typu E²PROM programovatelnou přímo v aplikaci přes USB. Do této paměti je možné uložit různé identifikační údaje jako třeba sériové číslo převodníku nebo jeho název. Je kompatibilní se standardy USB 1.1 a USB 2.0. Jeho uplatnění je tedy možné i u starších osobních počítačů, které ještě nevyužívali rychlejší standardy USB.

Na obr. 3.1 je blokové schéma obvodu FT232BM. Na základě blokového schématu lze stručně popsat funkce obvodu FT232BM. Na napájecím vstupu je připojen 3,3 V LDO regulátor, který má za úkol generovat referenční napětí pro buzení USB vysílače. Na vstupu obvodu je připojen USB přijímač/vysílač, který poskytuje fyzické rozhraní pro USB kabel. Na tento blok navazuje blok nazvaný USB DPLL, který provádí detekci hodinového a datového signálu z příchozího NRZI kódování používaného ve standardu USB. Na tento blok musí být připojen oscilátor, který generuje referenční hodinový kmitočet 6 MHz. Na tento oscilátor je napojena násobička kmitočtu, která vytváří referenční kmitočet 12 MHz pro bloky SIE, UPE a UART FIFO. Také generuje 48 MHz referenční hodiny pro USB DPLL. Zmíněný blok SIE (Serial Interface Engine) provádí paralelně-sériovou a opačnou konverzi USB dat. Dalším blokem je UPE (USB Protocol Engine), který spravuje datový proud z řídicího koncového bodu USB. Dvoubranový TX a RX buffer, kde se data ukládají a čekají, než budou vyjmuta patřičným blokem. Blok, který ovládá přenos dat mezi RX/TX buffery a vysílacím/přijímacím

registrem UART. Registr UART zajišťuje 7/8 bitovou paralelně-sériovou a sériově-paralelní konverzi dat na RS-232 rozhraní. Na tento registr je připojen generátor přenosové rychlosti, který umožňuje programovat přenosové rychlosti od 300 Bd do 3 Mbd. Blok generátor resetu slouží ke spolehlivému resetu při připojení napájení. Posledním blokem je EEPROM rozhraní, které slouží pro uložení informací o sériovém čísle nebo řetězci popisu výrobku [4]. Konkrétní popis vývodu součástky je detailně popsán v katalogovém listu [6]. Tento obvod pro realizaci komunikačního modulu nebude použit, proto zde nebude detailně součástka rozebrána a zobrazena.



Obr. 3.1 Blokové schéma obvodu FT232BM

3.2 RS-232↔USB s AT90USB

V této části bude představen mikroprocesor od firmy Atmel, který umí přímo pracovat s oběma rozhraními, které bude využívat komunikační modul. Jde o mikroprocesor s řady 8 bitových procesorů a to konkrétně varianta AT90USB647 a nebo AT90USB1287. S použitím tohoto mikroprocesoru se podstatně usnadní návrh komunikačního modulu. Odpadne problém s převáděním a následným zpracováním instrukcí. Vše bude probíhat přímo pomocí mikroprocesoru, který tedy bude zajišťovat oboustrannou komunikaci mezi osobním počítačem a laboratorním zdrojem Manson SDP-2405.

3.2.1 Základní vlastnosti

Mikroprocesor AT90USB1287 je 8 bitový procesor s redukovanou instrukční sadou tedy procesor typu RISC. RISC označuje jednu z architektur procesorů. Jedná se tedy o procesor, který pracuje s redukovanou sadou příkazů, což v praxi znamená, že složitější příkazy nahrazuje posloupností jednodušších příkazů a tím se stává práce a následná komunikace procesoru efektivnější a podstatně rychlejší. Komunikace procesoru s pamětí probíhá přes sběrnici. Sada používaných instrukcí má jednotnou délku vykonávání a to jeden strojový cyklus. V dnešní době téměř každý procesor pracuje s redukovanou instrukční sadou obohacenou o náročnější instrukce. Mikroprocesor AT90USB1287 dosahuje při frekvenci 16 MHz výkonnosti až 16 MIPS (jednotka pro výkonnost instrukcí vyjadřující počet instrukcí

vykonaných za sekundu). Optimální frekvence, na které mikrokontrolér pracuje, se pohybuje v rozmezí od 8 MHz do 16 MHz. Většina aplikací používá pracovní frekvenci 8 MHz.

Tento procesor má paměťový prostor tzv. non-volatile pro data i program. To znamená, že paměť si svůj obsah dokáže podržet i po dobu nepřítomnosti napájecího napětí. Hlavní paměť je typu flash. Elektricky programovatelná paměť s libovolným přístupem. Vnitřní struktura paměti je bloková podobně jako paměť typu EEPROM, ale na rozdíl od paměti EEPROM je možné paměť flash programovat po jednotlivých blocích, přičemž obsah ostatních bloků je zachován. Data jsou ukládána v poli tranzistorů, zvaných „buňky“ každá z nich ukládá 1 bit informace. Mezi největší výhody patří možnost několikanásobného zápisu, vydrží tisíce přepsání. Paměti používané v mikroprocesorech AVR mají velikost 64/128 kB. Pouze velikostí pamětí se od sebe liší mikroprocesory AT90USB647 a AT90USB1287. Větší paměťový prostor má mikroprocesor AT90USB1287. Pro programování pamětí flash je používáno několik metod programování mezi které patří i programování přes rozhraní JTAG (Joint Test Action Group), které mikrokontrolér podporuje. JTAG je standard pro testování plošných spojů a programování pamětí flash. Dále mikrokontrolér obsahuje standardně používané paměti EEPROM (Electrically Erasable Programmable Read-Only Memory) a SRAM (Static Random Access Memory). Paměť typu EEPROM je elektricky mazatelná a má oproti paměti flash omezenější počet zápisů a hlavně před novým programováním je nutné smazat elektrickým signálem celý obsah. V současnosti se upouští od používání těchto pamětí. Paměť typu SRAM je označena pro polovodičovou paměť, která k uchování svých dat nepotřebuje jejich periodickou obnovu. Používají se pro dočasné ukládání programu či dat. Při odpojení napájení dochází ke ztrátě uložených dat. V porovnání s pamětí typu DRAM (Dynamic Random Access Memory) jsou rychlejší a v klidovém stavu mají menší odběr. Nicméně paměti DRAM jsou dnes používanější z důvodu, že lze dosáhnout při výrobě větších kapacit než u pamětí SRAM.

Mikroprocesor obsahuje zařízení podporující full speed i low speed mód standardu USB 2.0. Rychlost přenosu dat až do 12 Mbit/s. Je tedy možné plnohodnotně využít USB standard pro běžné aplikace, jako připojení USB klíčenky nebo i pro připojení laboratorního zdroje Manson. Mikroprocesor je možné zapojit v obou používaných režimech, tedy jako zařízení anebo jako host. USB mikroprocesor lze nastavit jako klasické USB zařízení anebo jako simulace sériového portu nazývané „Virtuální COM“.

Dále má mikroprocesor dva 8 bitové čítače/časovače, dva 16 bitové čítače/časovače. Čítač/časovač je periférie, která odečítá čas anebo načítá vstupní pulsy. To se nejčastěji využívá pro měření krátkých intervalů, pro periodické spouštění funkce, lze to využít i pro jednoduché frekvenční čítače. Obě funkce využívají registry, jejichž obsah se inkrementuje pomocí hodinového případně externího signálu, který je odvozen od pracovního kmitočtu. Dále je možné využít čtyři 8 bitové PWM (PWM – Pulse Width Modulation – pulsně šířková modulace) kanály. Informace je PWM signálem přenášena proměnnou střídou obdélníkového signálu. Princip generování PWM signálu je založen na přetečení časovače.

Další periférie, kterou obsahuje popisovaný mikroprocesor je 10 bitový AD převodník. AD převodník převádí velikost vstupního analogového napětí na n-bitové digitální číslo. Z důvodu nižšího odběru bývá AD převodník defaultně vypnut. Pro použití je ho tedy nutné prve povolit v příslušném registru. Součástí této periférie bývá i funkce analogového komparátoru, který porovnává dva signály přiváděné na vstup. Při rovnosti těchto signálů dojde k překlopení komparátoru a opětovnému porovnávání.

Pro sériovou komunikaci s okolím je možno využít programovatelnou sériovou jednotku USART (Addressable Universal Synchronous Asynchronous Receiver Transmitter). Synchronní a asynchronní sériové rozhraní USART je také někdy nazýváno SCI (Serial

Communications Interface). Jednotka USART může pracovat ve dvou různých módech. První mód je nazývaný asynchronní plně duplexní, pomocí kterého může mikroprocesor komunikovat se zařízeními jako je hyperterminál (nástroj pro konfiguraci síťových zařízení připojených přes sériový COM port) nebo osobní počítač. Druhý možný mód je označován synchronní polo duplexní, pomocí kterého může mikrokontrolér komunikovat s perifériemi např. AD převodník, sériová paměť EEPROM. Tento synchronní mód je možné nastavit do režimu Master nebo Slave. Synchronní mód používá pro komunikaci hodinovou a datovou linku na rozdíl od asynchronního módu, který pro komunikaci potřebuje pouze linku datovou. Jeden pin je využívám pro příjem dat a druhý pro vysílání dat, oba piny mohou pracovat nezávisle na sobě a dokonce mohou pracovat současně. Proto je tento mód plně duplexní. Nejčastější použití jednotky USART v asynchronním módu je u komunikace s osobním počítačem prostřednictvím sériového portu a protokolu RS-232.

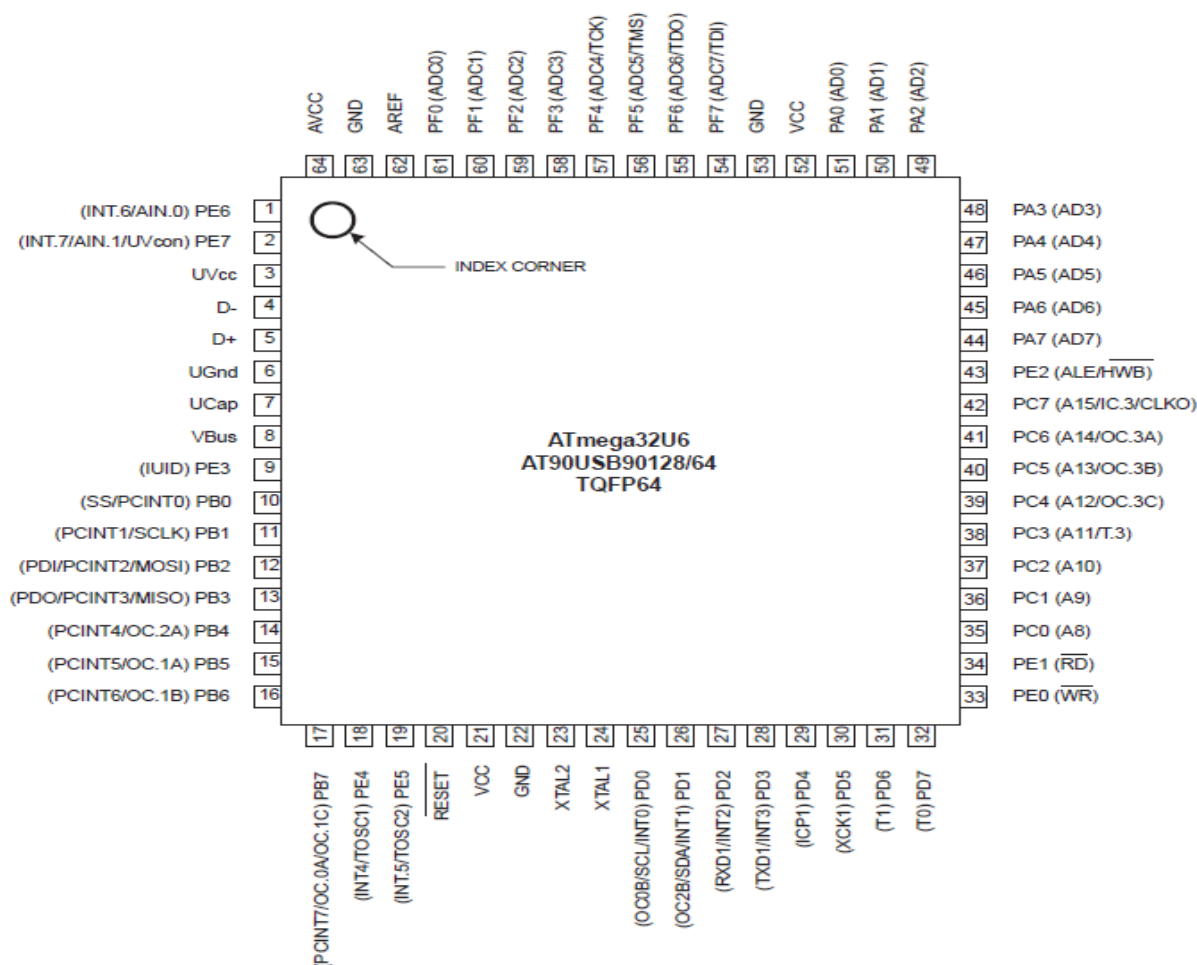
Mezi další možnosti komunikace patří komunikace po sběrnici SPI (Serial Peripheral Interface). Tohle sériové periferní rozhraní se využívá pro komunikaci mezi řídicími mikroprocesory a ostatními integrovanými obvody např. paměť EEPROM, AD převodník, LCD display. Komunikace je realizována pomocí společné sběrnice, na které jsou zařízení rozdělena do dvou typů. První typ je zařízení nazývané Master, které řídí komunikaci pomocí hodinového signálu a určuje, s kterým zařízením na sběrnici bude komunikovat pomocí signálu SS neboli Slave Select. Druhý typ zařízení je nazýván Slave, které na základě aktivního signálu SS zahájí komunikaci s nadřazeným zařízením Master.

Součástí dnešních mikroprocesorů bývá i obvod nazývaný programovatelný watchdog. V českém překladu lze obvod watchdog vyjádřit slovním spojením „hlídací pes“ nebo odborně řečeno „bezpečnostní časovač“. Význam obvodu watchdog spočívá ve vyvolání resetu mikroprocesoru v případě, že se procesor „ztratí“ nebo přestane vykonávat funkce, které by měl. K resetu mikroprocesoru dojde po přetečení watchdog časovače. Po resetu začne procesor vykonávat instrukce od začátku programu. Funkce watchdog je defaultně vypnuta a používá se většinou v aplikacích, které nejsou snadno dostupné, tedy není možné při poruše resetovat mikroprocesor manuálně.

Mikroprocesor má celkem 38 přerušovacích vektorů, mimo jiné může externí přerušení vyvolat příjem dat na USB řadiči nebo na sériové jednotce USART. Je možné použít až 48 programovatelných vstupně/výstupních portů pro připojení různých externích prvků. Pro programování lze využít tzv. Bootloader neboli zavaděč. Bootloader je část kódu umístěná v horní části programové paměti, která má za úkol přijímat kód a zapisovat jej do dolní části programové paměti. Umožňuje tedy programování přes USB rozhraní. Velmi pohodlně lze tedy programovat tento mikroprocesor přímo z vývojového prostředí AVR studio. Pokud je potřeba nastavit USB rozhraní lze využít zavaděč až po provedení aktivace pomocí jiného způsobu programování např. pomocí ISP (In-System Programming) programování. Podrobněji bude ISP programování mikroprocesoru rozebráno v další části textu, jelikož tento způsob programování bude využíván k nahrávání obslužného programu do komunikačního modulu.

AT90USB647 respektive AT90USB1287 se vyrábí v pouzdře TQFP, které má rozměry včetně vývodů 16 x 16 mm. Mikroprocesor má celkem 64 vývodů, přičemž některé piny reprezentují dvě i více možných funkcí, které se nastavují softwarově pro konkrétní aplikace. Provozní napětí mikroprocesoru může být v rozmezí 2,7 až 5,5 V. Při zapojení v režimu device mode je mikroprocesor napájen ze sběrnice USB a po rozpoznání zařízení je mu přiřazeno napětí 5 V. Provozní teplota v průmyslovém prostředí by se měla pohybovat v rozpětí od -40 °C až do +85 °C. Na obr. 3.2 je mikroprocesor AT90USB647 respektive AT90USB1287 a popis jednotlivých pinů osazených v pouzdře TQFP64, které bude použito

pro realizaci komunikačního modulu pro laboratorní zdroj Manson SDP-2405 [7].



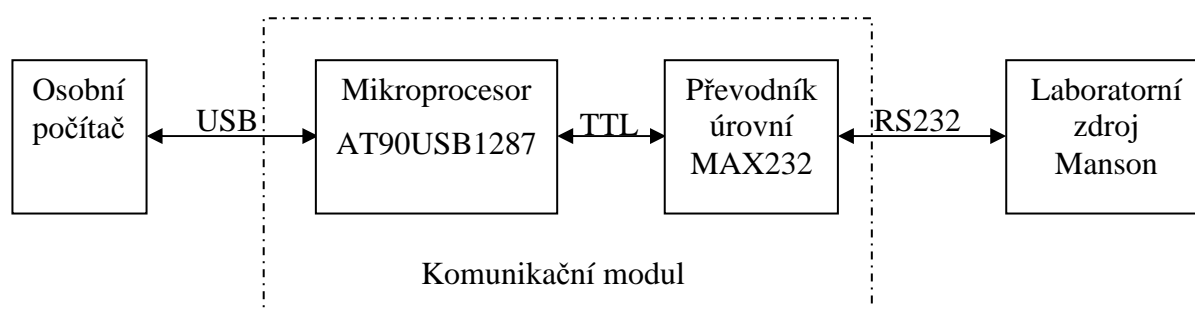
Obr. 3.2 Mikroprocesor AT90USB v pouzdře TQFP64

3.2.2 Návrh zapojení komunikačního modulu

Pro řešení komunikačního modulu je možné použít mikroprocesory od firmy Atmel a to konkrétně AT90USB647 anebo AT90USB1287. Některé parametry těchto mikroprocesorů a možnosti využití byly popsány v předchozím textu. Pro finální řešení komunikačního modulu je nejvíce podstatný fakt, že tyto mikroprocesory umí pracovat plně duplexně s USB rozhraním a zároveň obsahují sériovou jednotku USART, která zabezpečuje synchronní i asynchronní plně duplexní sériovou komunikaci s okolními periferiemi v tomto konkrétním případě s laboratorním zdrojem Manson SDP-2405.

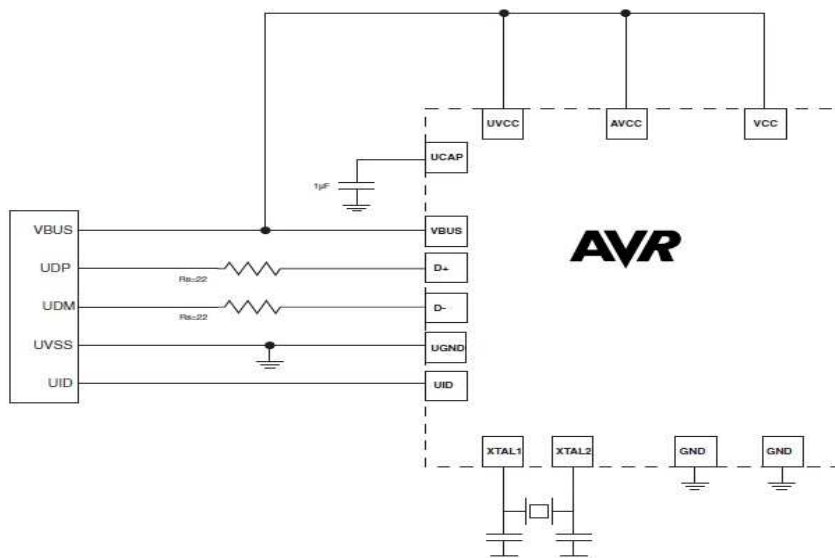
Na obr. 3.3 je blokový návrh komunikačního modulu pro laboratorní zdroj. Na vstupu přípravku je uvažován klasický USB konektor typu A, pro připojení ke standardnímu konektoru používaného v osobních počítačích. Dále je již zmiňovaný mikroprocesor, který bude sloužit jako jádro celého komunikačního modulu. Pomocí vytvořeného programu bude řídit obousměrnou komunikaci mezi osobním počítačem a laboratorním zdrojem. Bude tedy přijímat data po sběrnici USB a tato data poté předá po sériové lince laboratornímu zdroji. V rámci komunikace bude převádět některé příkazy, které standardně vyžaduje laboratorní zdroj Manson na příkazy, které jsou pro uživatele přívětivější při komunikaci s tímto zdrojem. Mikroprocesor běžně pracuje s TTL logikou, což ovšem neplatí pro sériovou komunikaci pomocí sériové linky RS-232, která pracuje při jiných napěťových úrovních. Z tohoto důvodu je nutné v blokovém schématu uvažovat i převodník napěťových úrovní MAX232, který tedy

provede příslušnou konverzi napěťových úrovní. Pro komunikaci s laboratorním zdrojem je poté již použita klasická sériová linka RS-232. Napájecí napětí komunikačního modulu je realizováno ze sběrnice USB, která je napájena ze zdroje umístěného přímo v osobním počítači a je tedy schopna poskytnout napájecí napětí připojovaným zařízením. Celá komunikace by měla fungovat v obou směrech, kdy bude možné provádět nastavení různých parametrů laboratorního zdroje jako je nastavení výstupní napěťové pojistky, omezení velikosti výstupního proudu nebo sepnutí respektive rozepnutí výstupních svorek, ale také bude možné zpětně přečíst informace, které poskytuje laboratorní zdroj a zobrazit je v požadovaném formátu ve zmíněném grafickém prostředí Agilent VEE Pro. Mezi informace, které lze získat z laboratorního zdroje patří například měření výstupního napětí či proudu nebo informace o stavu výstupních svorek.



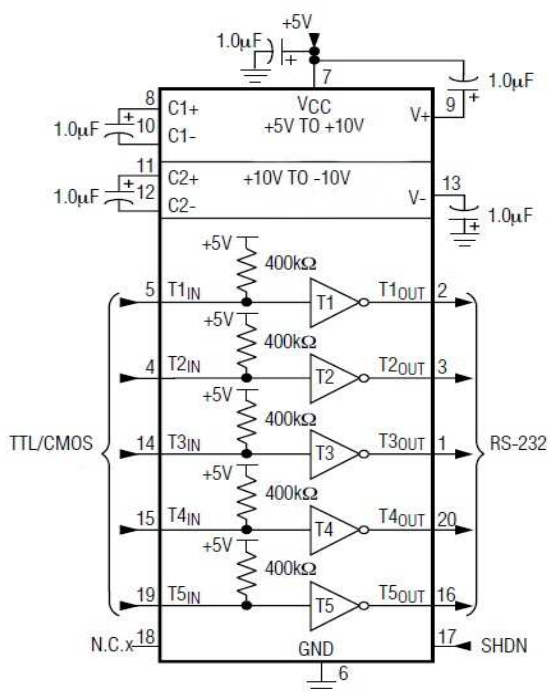
Obr. 3.3 Blokové schéma komunikačního modulu pro laboratorní zdroj

Při návrhu komunikačního modulu bylo třeba prostudovat detailně možnosti zapojení jednotlivých částí mikroprocesoru AT90USB647 respektive AT90USB1287. Návrh zapojení lze tedy rozdělit do několika částí. První část schématu je správné zapojení USB konektoru. Je několik možností jak zapojit USB konektor. Základní rozdělení se odlišuje v typu zapojení napájecího napětí. Jsou dva možné způsoby napájení mikroprocesoru. První způsob vyžaduje externí napájecí zdroj. Tato varianta zbytečně komplikuje napájení komunikačního modulu, který neklade vysoké nároky na stabilitu a přesnost napájecího napětí. Komunikační modul bude zapojen tak, aby využil druhého způsobu napájení a to přes USB sběrnici, tedy napájení bude zajišťovat napájecí zdroj osobního počítače, ke kterému bude komunikační modul připojen. Tento způsob napájení je nejčastěji používán pro USB zařízení, která jsou připojována k osobním počítačům. Poté je možné volit velikost napájecího napětí. První úspornější varianta uvažuje napájecí napětí 3 V, jelikož komunikační modul bude komunikovat i přes sériovou linku RS-232, je vhodnější, zvolit vyšší napájecí napětí 5 V. Poslední možná volba při návrhu zapojení USB konektoru spočívá ve výběru režimu, ve kterém bude USB rozhraní pracovat. Opět jsou na výběr dvě možnosti nazývané „Device mode“ a „Host/OTG mode“. Častěji využívanou možností je „Device mode“, ve kterém pracuje většina zařízení připojovaných přes USB rozhraní. Pro zapojení komunikačního modulu bylo vybráno z dokumentace zapojení, které je uvedeno na obr. 3.4 je to zapojení v režimu „Device mode“ s napájecím napětím 5 V, které je napájené ze zdroje osobního počítače, tedy po sběrnici USB [7].



Obr. 3.4 Schematické zapojení USB konektoru

Zapojení sériové linky RS-232 není náročné, jedná se pouze o vyvedení pinů označených TxD a RxD vysílací respektive přijímací datová linka. Jediná komplikace, která nastává při zapojení sériové linky RS-232 je rozdílná velikost napěťových úrovní vyskytujících se na výstupu mikroprocesoru a očekávanou napěťovou úrovní, která musí být přítomna na příslušných pinech sériového konektoru. Z tohoto důvodu je nutné do zapojení uvažovat zapojení převodníku napěťových úrovní označovaným MAX232. Doporučené zapojení převodníku napěťových úrovní je uvedeno na obr. 3.5 včetně vnitřní struktury obvodu [14]. Samotný konektor RS-232 je možné zvolit buď v provedení označovaném „male“ nebo „female“, jelikož laboratorní zdroj používá sériový konektor typu „female“ bude na komunikačním modulu použit sériový konektor typu „male“, aby pro propojení s laboratorním zdrojem mohl být používán běžně dostupný sériový kabel zakončený koncovkami typu „male“ a „female“.



Obr. 3.5 Zapojení převodníku úrovní MAX232 a jeho vnitřní struktura

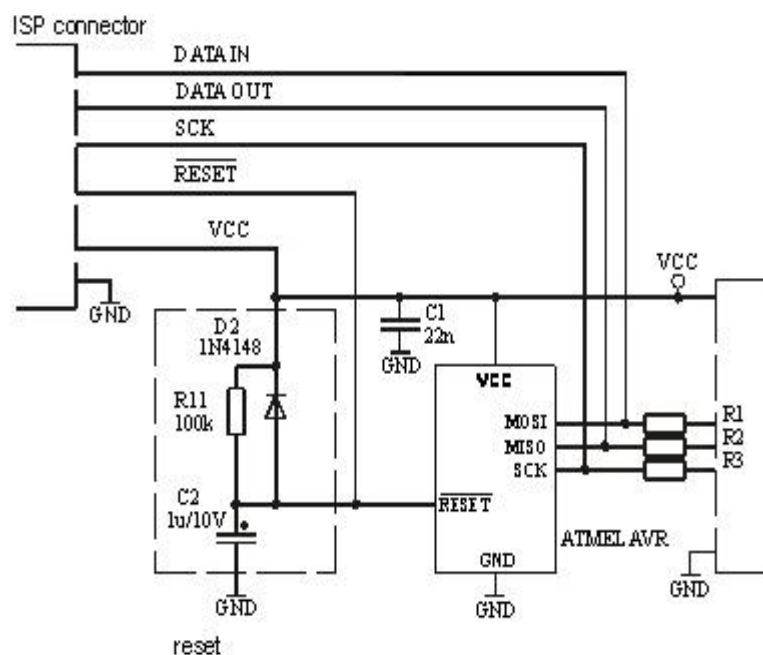
Při návrhu zapojení je nutné uvážit, jakým způsobem bude osazený mikroprocesor naprogramován. Programování mikroprocesorů znamená uložení programu, nejčastěji ve formátu Intel Hex, do flash paměti mikroprocesoru. Mezi základní způsoby programování moderních AVR procesorů patří několik způsobů naprogramování.

Pro paralelní programování je nutný velký počet vodičů. Programování je celkem nepohodlné z důvodu nutnosti při každém přeprogramování vyjmout mikrokontrolér ze systému. Ovšem nesporná výhoda je ve velké rychlosti programování. Je možné programovat všechny paměti (EEPROM, flash, SRAM) i nastavení zámkových bitů, či programovacích propojek. Paralelní programování využívá napájecí napětí 12 V.

Další možnost programování je pomocí rozhraní JTAG. Největší výhodou při programování přes rozhraní JTAG je skutečnost, že je možné komunikovat s mikroprocesorem během ladění aplikace. Je možné tedy sledovat stav periférií mikroprocesoru a případně zasahovat do jeho nastavení za chodu programu. Možnost krokovat program přímo na čipu velmi usnadňuje ladění výsledné aplikace. Konkrétně AVR mikroprocesory lze takto ovládat přímo z AVR studia, které je nejčastěji používané pro tvorbu aplikací.

Jako nejvhodnější a zároveň snadno dostupné programování pro tento projekt se jeví sériové programování v systému často označované zkratkou ISP. Není nutné mikrokontrolér při každém přeprogramování vyjmout ze systému, ovšem ani neposkytuje možnost ladění aplikace za chodu programu jako je to v případě použití rozhraní JTAG. ISP programování využívá komunikaci po sériovém rozhraní SPI. SPI je duplexní, synchronní přenos, který využívá ke komunikaci čtyři vodiče, které se dnes nachází na všech typech mikroprocesorů. Při programování ISP je tedy nutné zapojit tyto čtyři vodiče: pin označovaný SCK, který reprezentuje hodinový signál, pin MOSI vodič, který řídí komunikaci od nadřazeného zařízení k podřazenému, poté opačný směr komunikace zajišťuje vodič označovaný MISO, poslední pin označovaný \overline{SS} , pomocí kterého se vybírá podřazené zařízení, které bude komunikovat s nadřazeným. Programování tedy probíhá stylem, že nadřazené zařízení označované Master programuje podřazené zařízení označené Slave. Využívá se dvou posuvných registrů a generátoru hodin od nadřazeného zařízení. Komunikace při programování je zahájena nastavením signálu \overline{SS} do logické nuly. Připravená data k přenosu se umístí do posuvných registrů. Samotný přenos je poté řízen hodinovým signálem SCK generovaným od Mastera. Po vodiči označovaném MOSI poté probíhá komunikace a přenos dat směrem k programovanému zařízení. Data se posílají zpravidla od nejvíce významného bitu MSB až po nejméně významný bit LSB. Po odeslání všech dat je signál \overline{SS} opět nastaven do klidové polohy, tedy logické jedničky.

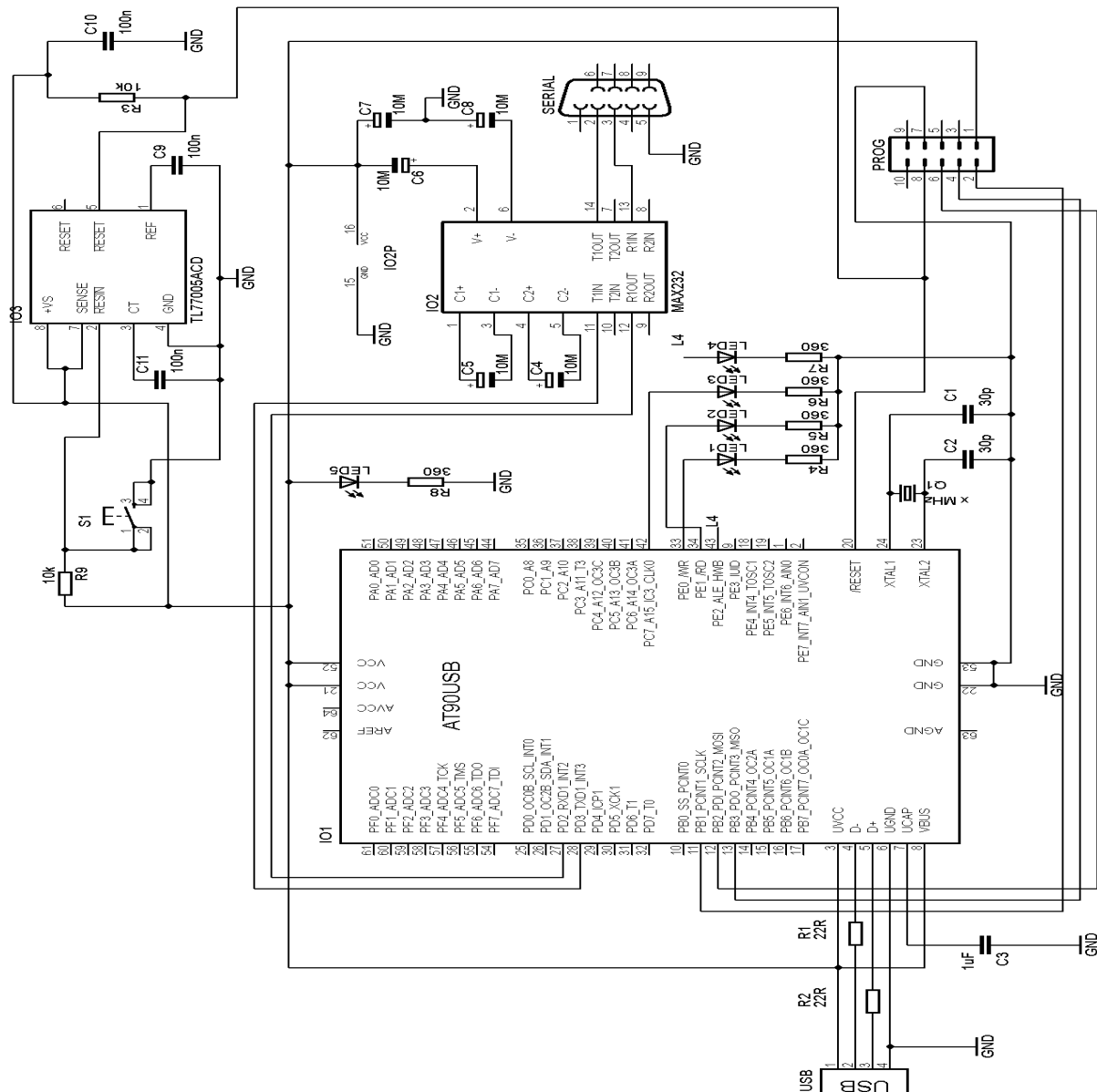
Pro programování komunikačního modulu bude používán programátor od firmy ELNEC s označením T51prog2. Je to moderní univerzální programátor, který umožňuje programovat klasické patkové mikroprocesory, ale má vyveden i ISP konektor, pro které je doporučené zapojení uvedeno na obr. 3.6. Dle tohoto zapojení bude na komunikačním modulu vyveden ISP konektor pro programování AT90USB647 respektive AT90USB1287. Na obr. 3.6 je uvedeno i zapojení resetovacího obvodu, které nebude využito. Účelem rezistorů R_1 , R_2 , R_3 , je oddělení programovaného obvodu od zbytku zařízení, jelikož komunikační modul není potřeba od něčeho oddělovat, nebudou tyto rezistory v zapojení uvažovány.



Obr. 3.6 Doporučené zapojení ISP konektoru

Poslední část zapojení komunikačního modulu je resetovací obvod, který se dá vyřešit jednoduchým zapojením uvedeným na obr. 3.6 anebo lze použít složitější zapojení resetovacího obvodu, které bude použito pro komunikační modul [15]. V zapojení figuruje integrovaný obvod, který řídí funkci resetu podle zapojení uváděného v dokumentaci. Výhoda zapojení spočívá v tom, že obvod získá velmi přesný impuls resetu, nenachází se v něm žádné překmity nebo menší napěťové úrovně, které by mohli vést k neurčitým stavům. Reset je vyveden na tlačítko, které při stisku vyvolá resetovací signál, který je přiveden na příslušný vývod mikroprocesoru a provede jeho reset.

Na obr. 3.7 je uvedeno výsledné schéma zapojení, které obsahuje všechny popsané části zapojení a jako celek tedy tvoří zapojení pro komunikační modul, který bude vytvořen a bude sloužit pro komunikaci s laboratorním zdrojem Manson. Zapojení obsahuje i několik LED diod, které budou sloužit pro signalizaci zapnutého napájecího napětí, ale také budou signalizovat chybový stav komunikačního modulu, který nastane při chybně zadaném příkazu v grafickém prostředí VEE Pro a bude tedy signalizovat uživateli, že zadaný příkaz není správně zadán a očekává se opravená verze odesílaného příkazu. Zapojení obsahuje i zmiňovaný integrovaný obvod, který obstarává reset systému. Reset systému v tomto případě znamená odpojení od operačního systému Windows. Po stisknutí tlačítka reset bude nutné komunikační modul odpojit od osobního počítače a znovu připojit. Z funkce resetu je tedy patrné, že resetovací tlačítko je používáno jen ve stavu, kdy komunikační modul přestane vykonávat svou funkci a zůstane v neurčitém stavu zpracování informací. Mikroprocesor nepoužívá periférii watchdog, proto může nastat stav, kdy bude potřeba využít právě tlačítko reset k obnovení správné funkce komunikačního modulu.

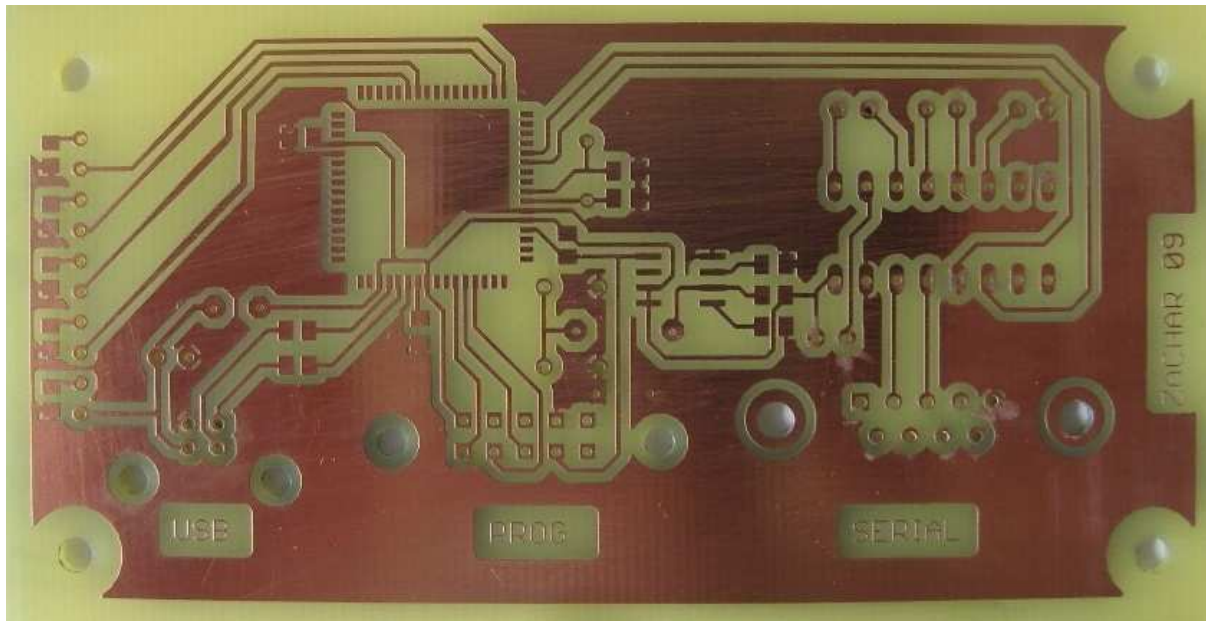


Obr. 3.7 Schéma zapojení komunikačního modulu

3.2.3 Deska plošných spojů

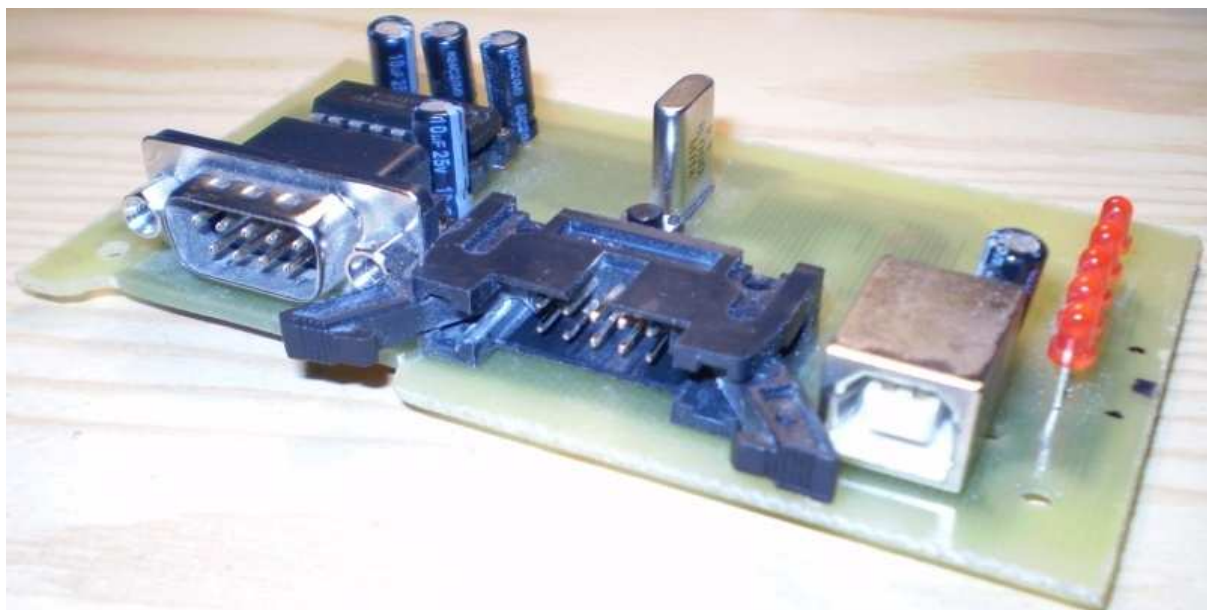
Navrhnuté schematické zapojení komunikačního modulu je nutné pomocí vhodného programového vybavení převést na návrh desky plošných spojů, která bude následně zhotovena a osazena příslušnými součástkami. Pro návrh desky plošných spojů byla použita studentská verze programu Eagle verze 5.6.0 Light. Do programu bylo nutné nahrát potřebné knihovny, aby mohly být uvažovány reálné velikosti součástek. Nejdůležitější část zapojení mikroprocesor AT90USB1287 je miniaturní součástka v provedení SMD, která má 64 vývodů a pro osazení do desky plošných spojů musí být předloha velmi kvalitní a přesná. Při návrhu je potřeba uvážit vhodné rozmístění konektorů, aby bylo možné všechny konektory pohodlně využívat a nehrozilo vytrhnutí konektoru z desky. Dále je nutné zvážit vhodnou volbu provedení součástek. V dnešní době lze použít klasické velikosti součástek anebo jejich ekvivalenty v provedení SMD. Kromě zapojení převodníku napětíových úrovní MAX232 je na desce použito převážně součástek v provedení SMD, tím se samozřejmě zmenšuje velikost desky. Integrovaný obvod MAX232 bude osazen v příslušné patici, jelikož pracuje s větším

napětím, než zbytek zapojení reálně může nastat stav poškození a nutnost vyměnit tento obvod. Velikost navrhované desky je limitována použitím tří poměrně prostorově náročných konektorů pro připojení USB kabelu, sériového kabelu RS-232 a také konektoru pro ISP programování. Napájecí napětí je vyřešeno vhodným zapojením mikroprocesoru, který je napájen z USB sběrnice, tedy přímo ze zdroje počítače, ke kterému je komunikační modul připojen. Komunikační modul má tedy napájecí napětí 5 V. Navrhnutá deska plošných spojů je znázorněna na obr. 3.8.



Obr. 3.8 Vyrobena deska plošných spojů

Pro osazení desky plošných spojů byla použita pájecí stanice od firmy Metcal, která má ve vybavení pájecí hroty přizpůsobené pro pájení velmi malých integrovaných obvodů. Takže při použití vhodných pomůcek lze integrovaný obvod se 64 vývody osadit velmi pohodlně a v krátkém čase. Výsledná podoba osazené desky plošných spojů je na obr. 3.9



Obr. 3.9 Konečná podoba komunikačního modulu

4 OBSLUŽNÝ SOFTWARE

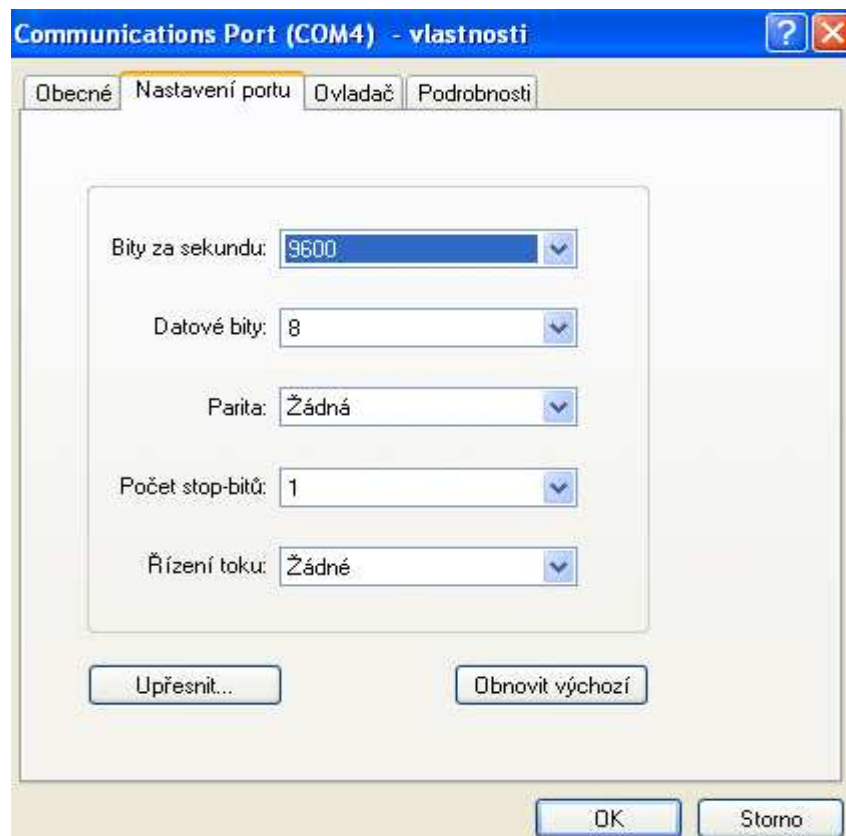
Navržený a zrealizovaný komunikační modul je potřeba doplnit obslužným programem, který bude splňovat požadavky zadání projektu. Obslužný software zabezpečí obousměrnou komunikaci mezi USB rozhraním a sériovou linkou využívanou u laboratorního zdroje. Součástí obousměrné komunikace bude vhodná úprava používaných příkazů. Pro komunikaci s mikroprocesorem bude použit projekt nazývaný „LUFA“, který obsahuje potřebné ovladače pro práci s USB rozhraním pro mikroprocesory řady AT90USB. Tento projekt obsahuje kromě ovladačů i rozpracované funkce pro snadnou komunikaci přes USB nebo sériovou linku RS-232. Na základě projektu LUFA bude vytvořen obslužný program pro navržený komunikační modul.

4.1 Projekt LUFA

Tento projekt přináší ovladače pro AVR mikroprocesory, které již obsahují USB rozhraní. Komunikace přes USB rozhraní není programově lehce řešitelná, proto začal vznikat projekt, který usnadňuje komunikaci přes rozhraní USB a umožňuje uživatelské přizpůsobení. LUFA je tzv. open-source knihovna pro USB AVR mikroprocesory. Open-source nebo také open-source software je počítačový software s otevřeným zdrojovým kódem. Otevřenost zde znamená jak technickou dostupnost kódu, tak legální dostupnost - licenci software, která umožňuje, při dodržení jistých podmínek, uživatelům zdrojový kód využívat, například prohlížet a upravovat [11]. LUFA je psána ve volně šiřitelném kompilátoru AVR-GCC, který je možno implementovat do AVR studia. Je tedy možné tento projekt volně využít pro vytvářené aplikace s využitím USB rozhraní. AVR studio je doporučeno pro práci s AVR mikroprocesory a je možné ho stáhnout přímo ze stránek firmy Atmel [7], která tyto mikroprocesory vyrábí. Součástí knihovny jsou demonstrační aplikace, které je možné použít pro načerpání informací o tom, jak vlastně celá knihovna pracuje. Jednotlivé demoverze jsou ukázkou, jak vhodně implementovat projekt LUFA do vlastních aplikací. Pro komunikační modul je nejvhodnější demo aplikace s názvem USBtoSerial, které bylo součástí starších verzí LUFA knihovny (do verze 090924). V současné verzi bylo demo USBtoSerial nahrazeno aplikací s názvem VirtualSerial, která pracuje na velmi obdobném principu. USBtoSerial demo řeší přímo převod mezi standardem USB a sériovou linkou RS-232. Tedy přijatá data z rozhraní USB předá sériové jednotce USART, která je pošle na sériovou linku RS-232. Data přicházející přes sériovou linku RS-232, vyvolají externí přerušování mikroprocesoru, jsou uložena a přeposlána na USB rozhraní. Knihovna je v současnosti dostupná ve stabilní verzi umožňující začlenění do uživatelských projektů. Součástí projektu je i podrobně zpracovaná dokumentace, která obsahuje detailní popis jednotlivých funkcí. Tyto funkce jsou součástí všech demo aplikací a jsou připraveny k různým kombinacím. Lze sloučit několik těchto aplikací do jednoho projektu a vytvořit si aplikaci, která bude splňovat požadavky zadavatele. Dokumentace obsahuje i návod, jak začít používat tento projekt. Dále je tam uveden soupis všech podporovaných mikroprocesorů. Celý projekt je neustále vyvíjen, s každou novou verzí je v dokumentaci detailně popsáno, které části projektu jsou nové a jaké změny byly provedeny ve stávajících aplikacích. Použití je možné při dodržení licenčních podmínek uváděných autorem knihovny a zároveň uvedení pevně definované modifikované licence MIT ve vlastním vytvářeném projektu [12].

4.2 Hlavní program

Po seznámení s možnostmi, které umožňuje projekt LUFA, byl vytvořen hlavní obslužný program pro vytváření komunikačního modulu. V první části programování bylo nutné vytvořit ovladače pro komunikaci s operačním systémem Windows XP. Tyto ovladače jsou součástí popisované knihovny LUFA. Komunikace s operačním systémem je realizována pomocí simulace virtuální sériové linky. K vytvoření virtuální sériové linky dochází při prvním připojení komunikačního modulu do sběrnice USB. Zařízení se inicializuje klasickým způsobem. Při prvním připojení operační systém nalezne zařízení s názvem „USB-RS-232 Modul“, jelikož operační systém Windows XP nemá v sobě zabudované příslušné ovladače, je nutné tedy instalaci zařízení dokončit pomocí instalace vhodných ovladačů, které obsahuje projekt LUFA. Ovladače se nacházejí přímo v kořenové složce konkrétních demoverzí. Pro komunikační modul je tedy potřebný ovladač ve složce USBtoSerial demo. Po úspěšné instalaci proběhne vytvoření tzv. virtuálního sériového portu, který simuluje funkci sériové linky. Zařízení je tedy detekováno v ovládacích panelech v kategorii porty COM a LPT. Při instalaci operační systém přidělí komunikačnímu modulu číslo portu, na kterém posléze bude s operačním systémem komunikovat. Obvykle je možné komunikační modul nalézt pod označením „Communications Port (COM 4)“. Číslo portu uvedené v závorce se liší dle jednotlivých osobních počítačů, konkrétně podle obsazenosti jednotlivých dostupných portů. Pro správný chod komunikačního modulu je dobré zkontrolovat nastavení vytvořeného virtuálního sériového portu. Ve vlastnostech přiděleného portu je možné provést základní nastavení. Na obr. 4.1 je uvedeno, jak má vypadat správné nastavení sériové komunikace pro realizovaný komunikační modul. Nejedná se tedy o plnohodnotný režim komunikace přes rozhraní USB. Nevýhoda této komunikace spočívá v nutnosti nainstalovat patřičný ovladač a nadefinování konkrétního USB jako virtuálního sériového portu.

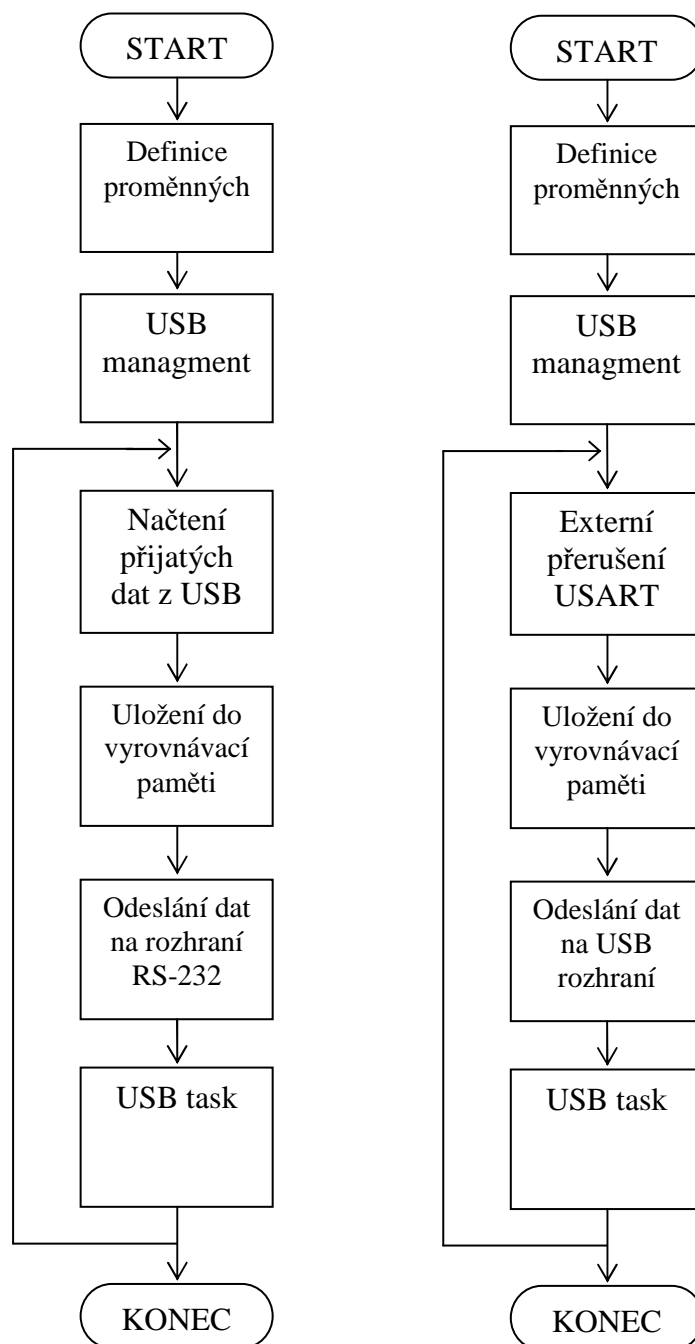


Obr. 4.1 Nastavení komunikačního portu

Struktura projektu se skládá ze tří částí. Popisovaný průběh instalace mají za úkol tzv. deskriptory. Pokud je USB zařízení v device mode, tak pomocí vhodně nastavených deskriptorů dojde k rozeznání zařízení a je umožněna následná instalace zařízení. V deskriptorech je možné definovat vlastnosti zařízení. Tedy informace o sériovém čísle, názvu zařízení, struktuře paměti a další parametry. Další součástí projektu je vytvoření kruhových vyrovnávacích pamětí tzv. RingBuff, které budou využity pro zpracování přijatých dat. Hlavní část projektu zpracovává komunikaci mezi USB standardem a sériovou linkou RS-232. Hlavní program lze rozdělit do několika částí, které jsou potřeba pro správnou funkci výsledného komunikačního modulu.

V první části hlavního programu jsou nadefinována jednorozměrná znaková pole, která budou využívána pro zpracování uložených dat. Dále potom jsou zde načteny kruhové vyrovnávací paměti, které udržují data před odesláním na USB rozhraní nebo na sériovou linku RS-232. Pomocí funkce *USB_ClassInfo_CDC_Device_t* jsou nakonfigurovány požadované parametry pro správnou komunikaci s USB rozhraním dále je možné pomocí této funkce zjistit informace o stavu připojení. Tato část programu je nazývána jako „USB management“. Druhá část programu začíná řešit problematiku komunikace mezi sériovým rozhraním RS-232 a USB rozhraním. Komunikace musí probíhat po celou dobu připojení napájecího napětí, proto je hlavní program uzavřen do nekonečné smyčky, která se opakuje po celou dobu komunikace. Tato smyčka neustále kontroluje obousměrnou komunikaci v komunikačním kanále. V okamžik, kdy se na některém rozhraní objeví data, dochází k jejich zpracování a odeslání na příslušné rozhraní.

Mezi hlavní úkoly nekonečné smyčky patří přečtení dat z USB Endpointu a uložení do vyrovnávací paměti vysílače USART. Endpoint je vyrovnávací paměť typu FIFO (first in first out), která pracuje obdobným způsobem jako posuvný registr, tedy postupně přijímá a vydává data. Při komunikaci přes USB rozhraní se využívá právě těchto Endpointů, kdy připojený mikroprocesor si z nich přečte přijatá data anebo naopak do nich data uloží. Další úkol je načtení dat z vyrovnávací paměti vysílače USART a odeslání na sériovou linku RS-232. Po odeslání dat dochází ke kontrole připojení k USB rozhraní. Tato část kódu je nazývána „USB task“. Po kontrole připojení se program vrací do stavu čekání na příchod dalších dat ke zpracování. Poslední úkol je přečtení a uložení přijatých dat ze sériové linky RS-232. Pokud dorazí data na sériovou jednotku USART dojde k vyvolání externího přerušení. Při identifikaci přijatých dat se využívá externí přerušení od sériové linky, kdy při příchodím bytu dojde k žádosti o přerušení. Mikroprocesor vyskočí z nekonečné smyčky, zpracuje přijatá data, která následně odesílá do USB IN endpoint. Dochází k opačné komunikaci, přijatá data jsou přítomna na USB rozhraní. Po dokončení zpracování dat dochází opět ke kontrole připojení. Pokud nenastala chyba, mikroprocesor očekává příchod dalších dat. Na obr. 4.2 je zobrazen vývojový diagram nekonečné smyčky aplikované v hlavní části programu pro komunikační modul. Vývojový diagram je rozdělen na dvě části. První část umístěna na obr. 4.2 vlevo se věnuje okamžiku, kdy jsou přijatá data na USB rozhraní. Postupně jsou vykonány uvedené bloky, přičemž první dva bloky jsou provedeny pouze při připojení komunikačního modulu k osobnímu počítači. Zbývající bloky jsou vykonávány neustále ve stejné posloupnosti. Druhá část umístěna na obr. 4.2 vpravo zobrazuje vývoj programu, pokud jsou přijatá data na sériovém rozhraní RS-232. Začátek vývojového diagramu je totožný s předchozím diagramem. Rozdíl nastává ve zpracování dat, které probíhá odlišným způsobem. Zpracování dat je založeno na vyvolání požadavku na externí přerušení od sériové linky.



Obr. 4.2 Vývojové diagramy pro zpracování přijatých dat

Přečtení dat z USB endpointu je provedeno pomocí funkce *CDC_Device_BytesReceived*, která určuje počet přijatých bytů, které se nachází na rozhraní USB. Tedy informuje mikroprocesor o počtu přijatých bytů, které je nutné poté uložit funkcí *Buffer_StoreElement* do vyrovnávací paměti vysílače USART.

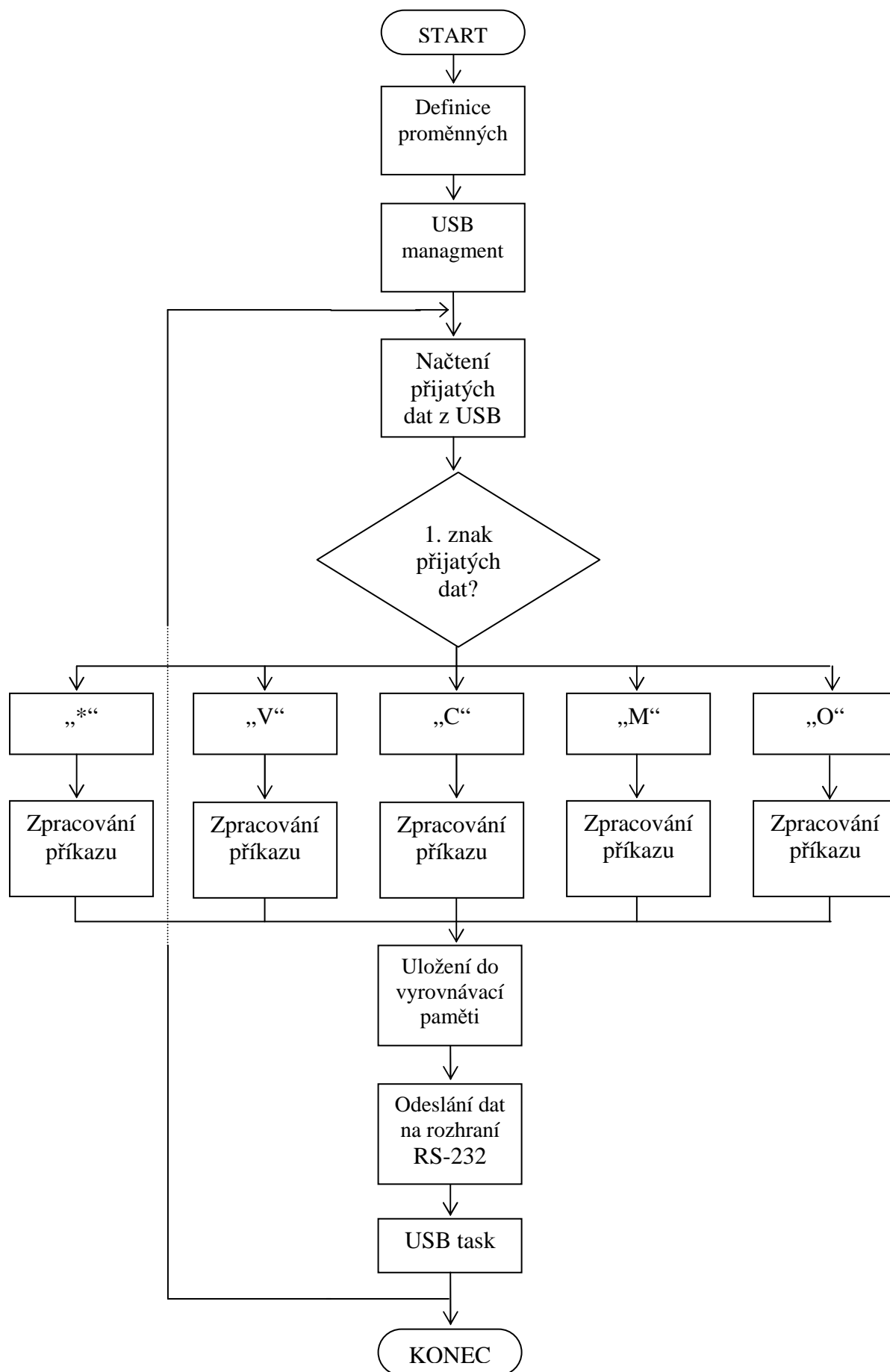
V okamžiku, kdy jsou přijata nějaká data a jsou uložena ve vyrovnávací paměti vysílače USART jsou pomocí funkce *Serial_TxByte* odeslána na sériovou linku RS-232 k příjemci tedy konkrétně k laboratornímu zdroji Manson SDP-2405. Po odeslání dat se mikroprocesor vrací do stavu, kdy čeká na nová přijatá data.

Pokud dorazí do přijímače USART data, které posílá laboratorní zdroj Manson, je vyvoláno externí přerušení a opět pomocí funkce *Buffer_StoreElement* jsou přijatá data uložena do vyrovnávací paměti a pomocí funkce *CDC_Device_SendByte* odeslána na připojeného USB hostitele, pokud není hostitel aktivní, jsou přijatá data zahozena. Po úspěšném odeslání se mikroprocesor vrací do stavu, kdy čeká na nová přijatá data.

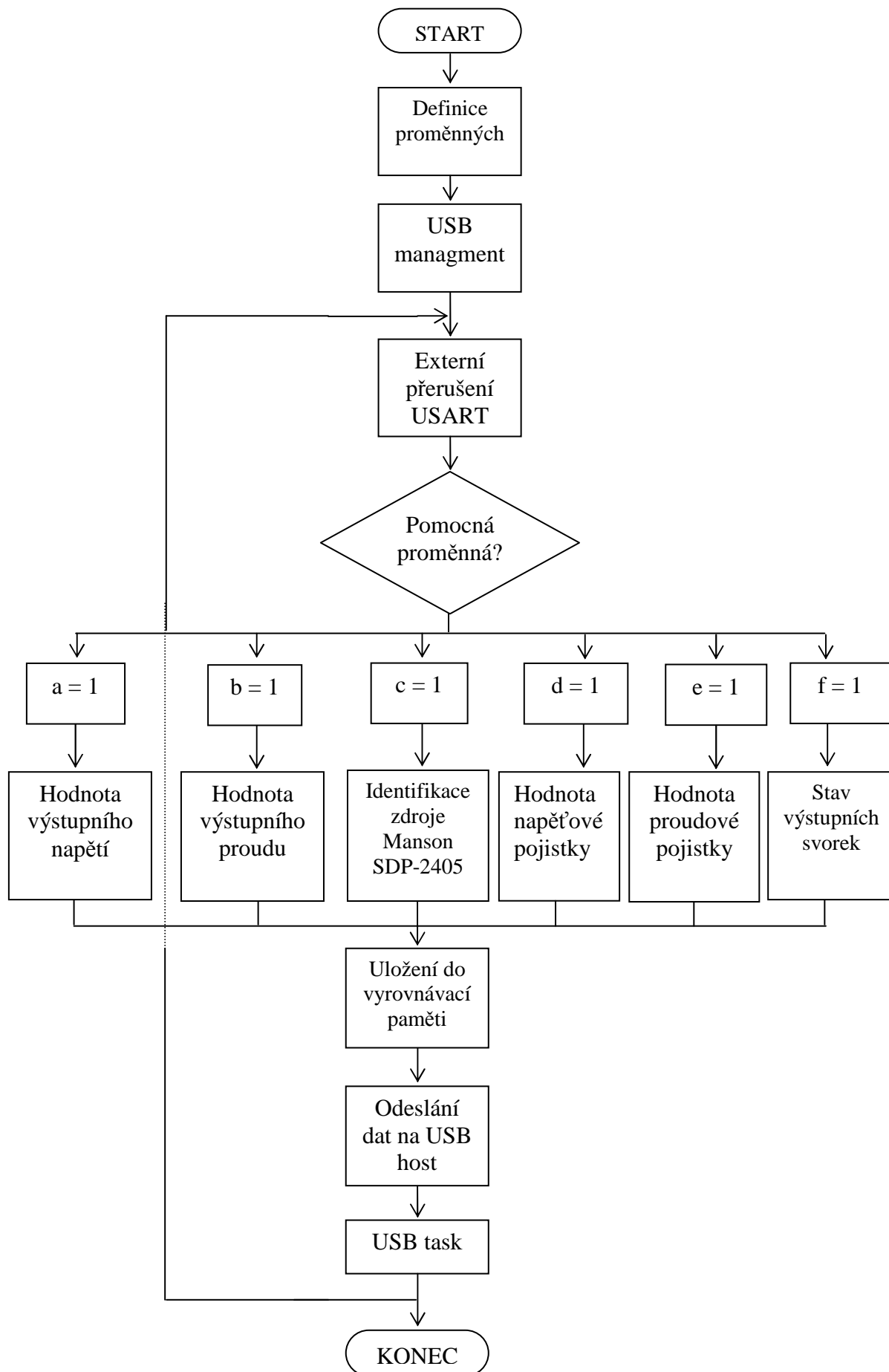
Poslední dvě funkce, které obsahuje nekonečná smyčka, jsou *CDC_Device_USBTask* a *USB_USBTask*, které jsou nutné pro správnou komunikaci s operačním systémem Windows XP. Dochází ke kontrole správné funkce USB rozhraní a neustálé kontrole připojení.

4.3 Úprava komunikace

Obousměrná komunikace mezi standardem USB a sériovou linkou RS-232 není finální řešení obslužného programu. Komunikační modul dále bude vhodně upravovat komunikaci mezi osobním počítačem a laboratorním zdrojem Manson. Výsledný obslužný software by měl zkvalitnit komunikaci s tímto napájecím zdrojem v grafickém prostředí VEE Pro. Instrukční sada dodávaná s laboratorním zdrojem tedy bude nahrazena novou instrukční sadou, která navíc bude splňovat požadavky SCPI standardu. Úprava komunikace probíhá také obousměrně. Zpracování přijatých dat je tedy doplněno o úpravu, která provádí příslušný převod příkazů. Princip hlavního programu je zachován. Ke komunikaci jsou využívány stejné postupy, které byly uvedeny výše. Úprava komunikace probíhá v okamžiku, kdy dojde k přečtení kompletního přijatého příkazu. Přijatý příkaz je vyhodnocen podle prvního přijatého znaku. Hlavní program se tedy větví do několika částí. V každé části dochází ke zpracování příslušného příkazu. Po zpracování příkazu dochází k uložení a odeslání dat příslušným směrem. Na obr. 4.3 je uveden vývojový diagram pro zpracování dat, které přicházejí z USB rozhraní. Vývojový diagram je doplněn o část, která se zabývá rozhodováním a zpracováním příkazů. Rozhodování probíhá podle prvního přijatého znaku. Tedy pokud přijde písmeno V, mikroprocesor vybere vhodnou část programu, ve které tento příkaz dále zpracovává. Zpracování příkazu je detailně popsáno v další části textu, ve které jsou uvedeny zkrácené vývojové diagramy pro jednotlivé přijaté příkazy a jejich možné podoby. Po zpracování uvedených znaků dochází k uložení a odeslání dat na sériovou linku RS-232. Na závěr je otestováno USB připojení. Na obr. 4.4 je uvedena podoba vývojového diagramu pro zpracování přijatých dat ze sériového rozhraní RS-232. Mikroprocesor obdrží požadavek na externí přerušení od jednotky USART. Dojde ke zpracování dat na základě nastavené pomocné proměnné. Tedy pokud uživatel požadoval změření výstupního napětí je nastavená příslušná pomocná proměnná. Na základě pomocné proměnné mikroprocesor vyhodnotí a zpracuje přijatá data ze sériové linky RS-232. V uvedeném příkladě uloží hodnotu výstupního napětí do vyrovnávací paměti, která je poté odeslána na rozhraní USB. Na závěr procesu je zkontrolováno USB připojení. Podoba příkazů pro získání hodnot, uvedené na obr. 4.4, je blíže popsána v dalším textu, který rozebírá detailně postup vyhodnocení a následného zpracování celé instrukční sady pro laboratorní zdroj Manson. Každá část instrukční sady je doplněna o zkrácenou podobu vývojového diagramu. Všechny dále uvedené vývojové diagramy jsou rozšířením pro vývojový diagram uvedený na obr. 4.3. Dále jsou uvedeny případy, kdy dochází k nastavení pomocných proměnných, které jsou vyhodnocovány po přijetí dat ze sériové linky RS-232 a odpovídá jim vývojový diagram uvedený na obr. 4.4.



Obr. 4.3 Vývojový diagram pro odeslání dat do laboratorního zdroje



Obr. 4.4 Vývojový diagram pro zpracování dat z laboratorního zdroje

4.3.1 Nastavení napěťové a proudové pojistky

Úprava komunikace spočívá ve vhodném uložení vysílaných nebo přijímaných dat. Uložené data je potřeba upravit, uložit a odeslat správným směrem. V první části šlo o úpravu dat, které jsou vysílány z prostředí VEE Pro do laboratorního zdroje Manson. Jedná se hlavně o základní příkazy pro nastavení maximálního výstupního napětí a výstupního proudu a o požadavek navrácení těchto hodnot zpět do grafického prostředí VEE Pro.

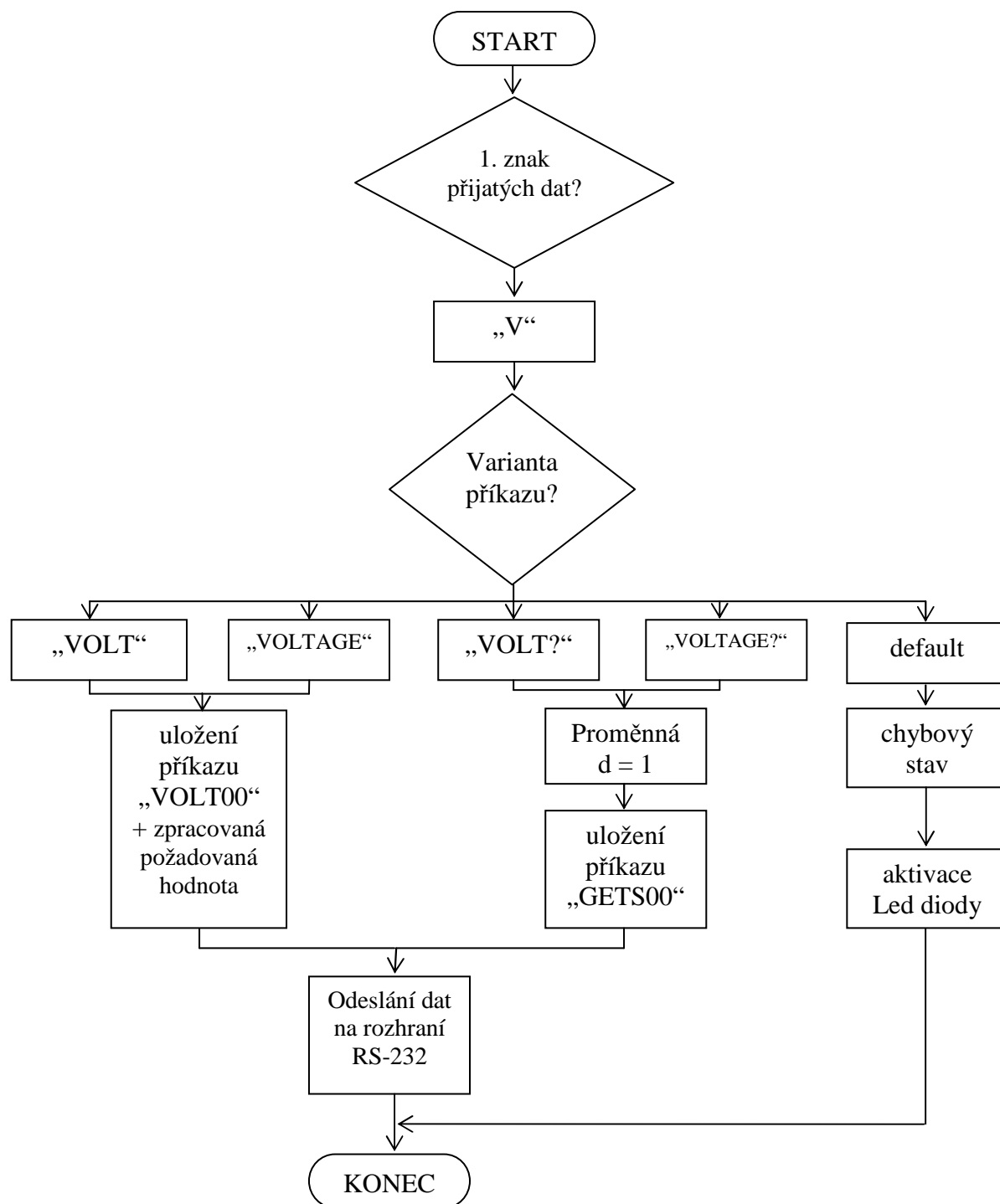
Nekonečnou smyčku programu je potřeba upravit. V okamžiku, kdy se na rozhraní USB objeví data, je nutné použít funkci pro uložení `CDC_Device_ReceiveByte`. Tato funkce přečte byte přijatých dat z osobního počítače, pokud data nejsou anebo USB hostitel není připojený, dochází k navrácení hodnoty 0. Pro jednoduchou úpravu dat je vhodné přijatá data uložit do jednorozměrného pole a dále s nimi pracovat pomocí dotazů na jednotlivé prvky pole. Úpravu lze potom provést pomocí délky jednorozměrného pole anebo čekáním na znak ukončení řádku „\n“. Ukončení řádku znakem „\n“ signalizuje příchod úplných dat k úpravě, tedy jeden celý příkaz. Příkazy pro ovládání laboratorního zdroje pomocí komunikačního modulu jsou uvedeny v tab. 2.3. V této tabulce jsou příkazy odpovídající standardu SCPI a k nim přiřazeny ekvivalentní příkazy, které používá laboratorní zdroj Manson standardně. Takže příklad přicházejících dat z osobního počítače pro nastavení napěťové či proudové pojistky vypadá následovně `VOLTage XX.XV, CURRent YY.YA`. Velkými písmeny je uvedena povinná část příkazu. Malými písmeny je nepovinná část, která může, ale nemusí být součástí příkazu. Příkaz `VOLT XX.XV` je používán pro nastavení hodnoty napěťové pojistky na laboratorním zdroji, kde `XX.X` je konkrétní hodnota napětí v rozsahu 0 až 24 V. Podoba příkazu, který je schopen zpracovat laboratorní zdroj musí být ve tvaru `VOLT00XXX`, kde `XXX` vyjadřuje požadovanou hodnotu napěťové pojistky. Takže je nutné data vhodně uložit a zpracovat dle správného zařazení. Jedna z možností, která umožňuje přijatá data roztřídit je využití příkazu „*switch-case*“. Příkaz „*switch*“ se používá pro členění programu podle proměnné, která může nabývat různých hodnot. Příkaz „*switch*“ zjišťuje hodnotu a porovnává ji se seznamem „*case*“, který k ní náleží. Při souhlasu s konstantou umístěnou mezi příkaz „*case*“ a dvojtečkou se zpracuje příkaz této větve „*case*“, dokud se nenarazí na klíčové slovo „*break*“. Nežadá-li se klíčové slovo „*break*“, budou se provádět i příkazy následující větve „*case*“.

Po příchodu znaku pro ukončení řádku „\n“ a následném uložení dat do jednorozměrného pole je poté využít příkaz „*switch-case*“, který kontroluje příchod prvních znaků v uloženém jednorozměrném poli. Přijatá data jsou tedy zkoumána a zpracovávána podle prvního znaku pole, zda se jedná o příkaz obsahující znak písmeno V, tedy požadavek o nastavení napěťové pojistky anebo je na první pozici znak písmeno C potom se jedná o nastavení proudové pojistky. V dalším textu jsou rozebrány další varianty, které se mohou objevit jako první znak přijatého příkazu. Pokud je první znak v jednorozměrném poli písmeno V, mikroprocesor dále testuje další znaky, které jsou uloženy. Pokud zjistí, že v jednorozměrném poli je uloženo klíčové slovo `VOLT` anebo případně jeho doplňková varianta `VOLTAGE` očekává po mezeře příchod čísel z daného rozsahu. Tyto čísla udávají požadovanou hodnotu napěťové pojistky. Pokud neodpovídá některá část příkazu, mikroprocesor ukončí úpravu dat a zahlásí chybu. Chyba je signalizována pomocí led diody, která je zapojena na jeden z portů mikroprocesoru. Naopak správně přijatá data odpovídající zadanému příkazu, jsou vhodně přeskupena a dále zpracována. Pokud přijaté číslo náleží do povoleného intervalu, tedy od hodnoty 1 V až po 24 V, dojde k uložení do vyrovnávací paměti. Číselné hodnoty lze zadávat i ve formátu s jedním desetinným místem. Ovšem hodnota napěťové pojistky 24.1 V je už vyhodnocena jako chybně zadané číslo. Příkazy je možné zadávat i malými písmeny. Hodnota napěťové pojistky je potřeba ukončit odpovídající jednotkou, tedy pro nastavení napěťové jednotky je to písmeno V. Laboratorní zdroj Manson při nastavení napěťové pojistky očekává příkaz ve tvaru `VOLT00XXX` a navíc ukončení

řádku „\r“. Tedy vyrovnávací paměť se naplní pevnou částí příkazu VOLT00 a poté se k ní vhodně doplní příchozí hodnota požadované napěťové pojistky. Na závěr je nutné doplnit celý příkaz znakem „\r“. Vyrovnávací paměť je potom uložena do vysílače sériové jednotky USART. Pomocí sériové linky je celý příkaz odeslán do laboratorního zdroje Manson. Na závěr úpravy komunikace je nutné vyprázdnit jednorozměrné pole a čekat na příchod dalšího příkazu ukončeného znakem „\n“.

Příkaz VOLT může být místo hodnoty napěťové pojistky ukončen znakem „?“ . Pokud je příkaz ukončen znakem „?“ uživatel požaduje navrácení již nastavené hodnoty napěťové pojistky. Příkaz může opět nabývat i úplné verze ve tvaru VOLTAGE?. V okamžiku, kdy mikroprocesor zjistí, že obdržel podobu příkazu ukončenou znakem „?“ , nastaví pomocnou proměnou. Při zjištění neúplnosti či chybně zadaného příkazu je opět ohlášena chyba pomocí led diody. Pomocná proměnná v programu slouží pro správnou identifikaci přijatých dat od laboratorního zdroje Manson. Dále je naplněna vyrovnávací paměť příkazem GETS00, který je opět doplněn znakem „\r“. Poté je odeslán obsah vyrovnávací paměti do laboratorního zdroje Manson pomocí sériové linky RS-232. Na závěr úpravy komunikace je nutné vyprázdnit jednorozměrné pole a čekat na příchod dalšího příkazu ukončeného znakem „\n“.

Laboratorní zdroj Manson po zpracování příkazu odpoví a to tak, že pošle hodnotu nastavených pojistek. Tedy posílá data, která obsahují nastavené hodnoty napěťové i proudové pojistky. Při příchodu dat z laboratorního zdroje je vyvolán požadavek na externí přerušení. Mikroprocesor zahájí obsluhu externího přerušení. Přijatá data jsou uložena do jednorozměrného pole. Mikroprocesor čeká, dokud jednorozměrné pole nemá kompletní obsah, tedy musí být ukončeno znakem „\r“. Pomocí nastavené proměnné mikroprocesor vyhodnotí, že uživatel požadoval hodnotu napěťové pojistky. Na základě této informace vybere z jednorozměrného pole správné hodnoty informující o nastavené napěťové pojistce. Zpracovaná data jsou uložena do vyrovnávací paměti. Pokud je komunikační modul správně připojen k USB rozhraní, je možné provést odeslání přijatých dat. Hodnota nastavené proudové pojistky je při tomto požadavku zanedbána. Mikroprocesor ji odstraní z vyrovnávací paměti. Při správném nastavení grafického prostředí VEE Pro, se hodnota napěťové pojistky zobrazí uživateli v patřičném zobrazovacím bloku. Popisovaný průběh zpracování dat je zobrazen pomocí vývojového diagramu na obr. 4.5. Jedná se o zkrácenou variantu hlavního vývojového diagramu, který je možné vidět na obr. 4.3. Mikroprocesor po zjištění, že první znak přijatých dat obsahuje písmeno V, zkoumá dále variantu přijatého příkazu. Blok označený default je aktivován v okamžiku, kdy mikroprocesor narazí na chybu. Při vyhodnocení chybně zadaného příkazu je aktivována led dioda a zpracování je ukončeno. Mikroprocesor vymaže vyrovnávací paměť a přechází do klidového stavu, kdy očekává příchod dalších dat. Zpracování se dělí dle požadavku uživatele na část, kdy dochází k nastavení napěťové pojistky. Ve druhé části uživatel požaduje informaci o již nastavené napěťové pojistce. Po dokončení zpracování pokračuje hlavní program stejným způsobem, data uloží a odešle na příslušné rozhraní.

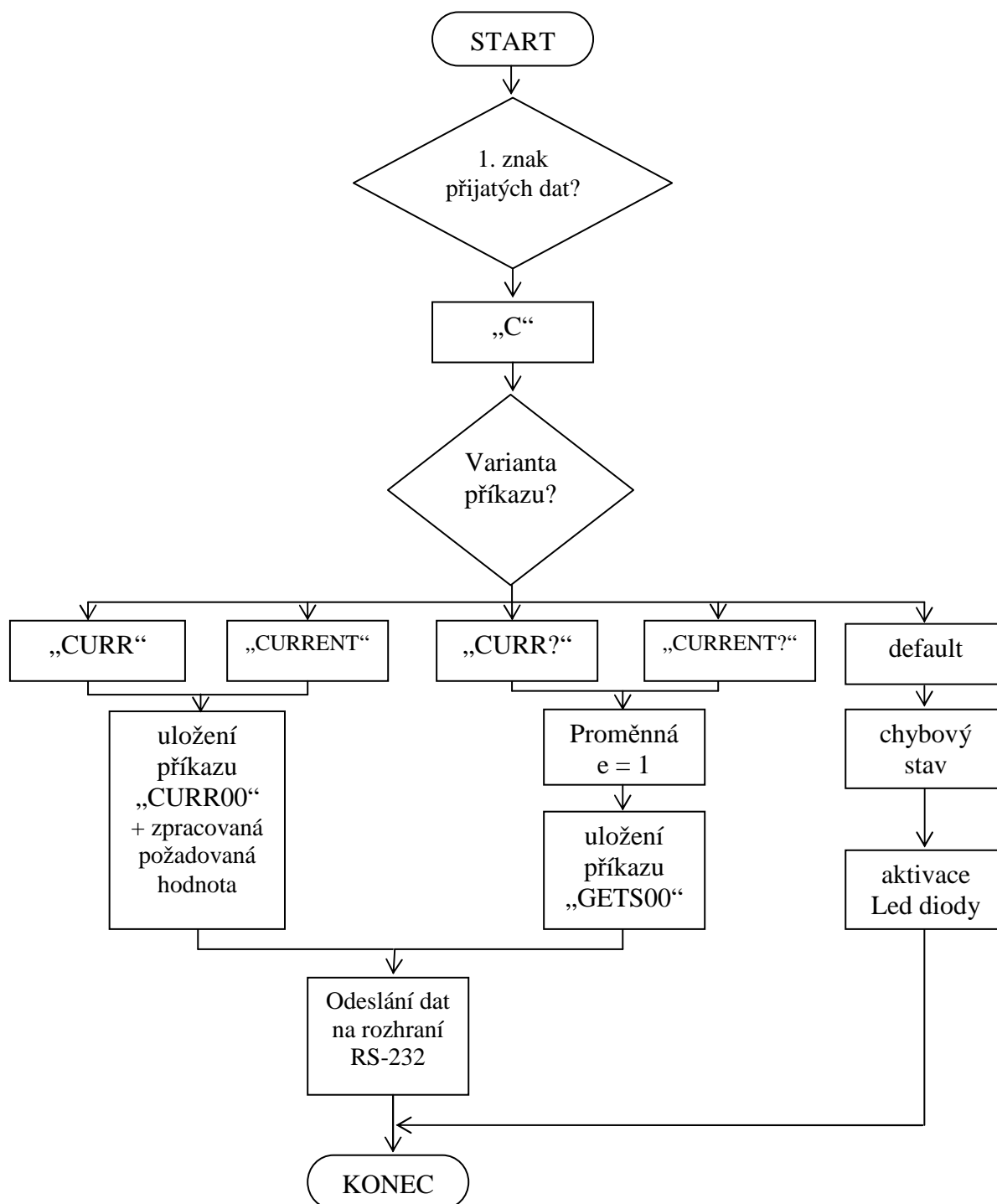


Obr. 4.5 Vývojový diagram pro nastavení napěťové pojistky

Pokud první znak odpovídá znaku písmena C, potom se jedná o práci s proudovou pojistkou. Mikroprocesor ověří, zda došel příkaz ve formátu CURR anebo CURRENT. Příkazy je možné posílat i malými písmeny. Pokud některý z přijatých prvků neodpovídá očekávané hodnotě je zhlášen chybový stav a mikroprocesor přestane dále pracovat na přijatém příkazu. Dojde k vymazání přijatého příkazu a rozsvícení chybové led diody. Textová část příkazu je ukončena číselnou hodnotou, kterou uživatel požaduje nastavit, jako proudovou pojistku. Pokud číselná hodnota spadá do intervalu od 0 A až do 5 A dojde

k dalšímu zpracování. Číselné hodnoty je možné zadávat s přesností na dvě desetinná místa. Ovšem pozor hodnota proudové pojistky ve tvaru 5.01 A je vyhodnocena mikroprocesorem jako chyba. Podobně jako u nastavení napěťové pojistky je potřeba hodnoty proudové pojistky ukončit příslušnou jednotkou, tedy pro proudovou pojistku je to písmeno A. Do vyrovnávací paměti je uložena pevná část příkazu CURR00. Poté dochází k vyhodnocení přijatého čísla. Zpracovaná hodnota proudové pojistky je doplněna do vyrovnávací paměti ve správném formátu. Na závěr je opět nutné celý příkaz doplnit ukončovacím znakem „\r“. Takto připravená data jsou v dalším kroku programu odeslána na sériovou linku RS-232. Po odeslání mikroprocesor vymaže vyrovnávací paměť. Po ukončení celé operace zpracování přijatých dat se mikroprocesor vrací do klidové polohy a očekává příchod dalších dat, která budou zpracována stejným způsobem.

Podobně jako příkaz VOLT pro nastavení napěťové pojistky může být i příkaz CURR pro nastavení proudové pojistky ukončen znakem „?““. Potom uživatel požaduje hodnotu již nastavené proudové pojistky. Opět je možné zadat příkaz i ve variantě CURRENT?. Pokud mikroprocesor rozezná jednu variantu z těchto příkazů, nastaví příslušnou proměnnou, která je určena pro odečtení nastavené hodnoty proudové pojistky. Do vyrovnávací paměti je uložen příkaz GETS00, který musí být ukončen znakem „\r“. Vyrovnávací paměť je odeslána pomocí sériové linky RS-232 do laboratorního zdroje Manson. Tento příkaz je vyhodnocen. Následuje reakce laboratorního zdroje Manson. Opět jsou odeslána data, která obsahují nastavení obou pojistek. Příchod těchto dat vyvolá v mikroprocesoru požadavek na externí přerušení. Mikroprocesor zahájí obsluhu externího přerušení vyvolaného od jednotky USART, která identifikovala příchozí data ze sériové linky RS-232. Dojde k uložení přijatých dat do jednorozměrného pole. Úplnost dat je opět rozeznána pomocí ukončovacího znaku „\r“. Mikroprocesor vybere pomocí nastavené proměnné data, které odpovídají hodnotám proudové pojistky. Uloží je do příslušné vyrovnávací paměti. Pokud je komunikační modul správně připojen do operačního systému Windows XP, mohou být zpracovaná data odeslána na USB rozhraní. Data obsahující nastavenou hodnotu napěťové pojistky jsou mikroprocesorem vymazána. Po odeslání dat je vyrovnávací paměť vymazána. Mikroprocesor se vrací do klidového stavu do doby, než přijde nový požadavek na zpracování dat. Při správném nastavení grafického prostředí VEE Pro, se hodnota proudové pojistky zobrazí uživateli v patřičném zobrazovacím bloku. Popisovaný průběh zpracování dat je zobrazen pomocí vývojového diagramu na obr. 4.6. Jedná se o zkrácenou verzi vývojového diagramu, která se zaměřuje na zpracování příkazů začínajících písmenem C. Blok nazývaný default je aktivní v okamžik, kdy mikroprocesor vyhodnotí chybu. Dojde k rozsvícení led diody a je přerušeno zpracování příkazu. Mikroprocesor vymaže přijatý příkaz a očekává příchod dalších dat. Dále je zpracování děleno podle požadavku uživatele. První část zpracování je aktivní, pokud uživatel požaduje nastavení proudové pojistky. Pokud uživatel potřebuje zjistit již nastavenou hodnotu proudové pojistky je aktivována druhá část zpracování. Po dokončení zpracování pokračuje hlavní program stejným způsobem, data uloží a odešle na příslušné rozhraní.



Obr. 4.6 Vývojový diagram pro nastavení proudové pojistky

4.3.2 Měření výstupního napětí a proudu

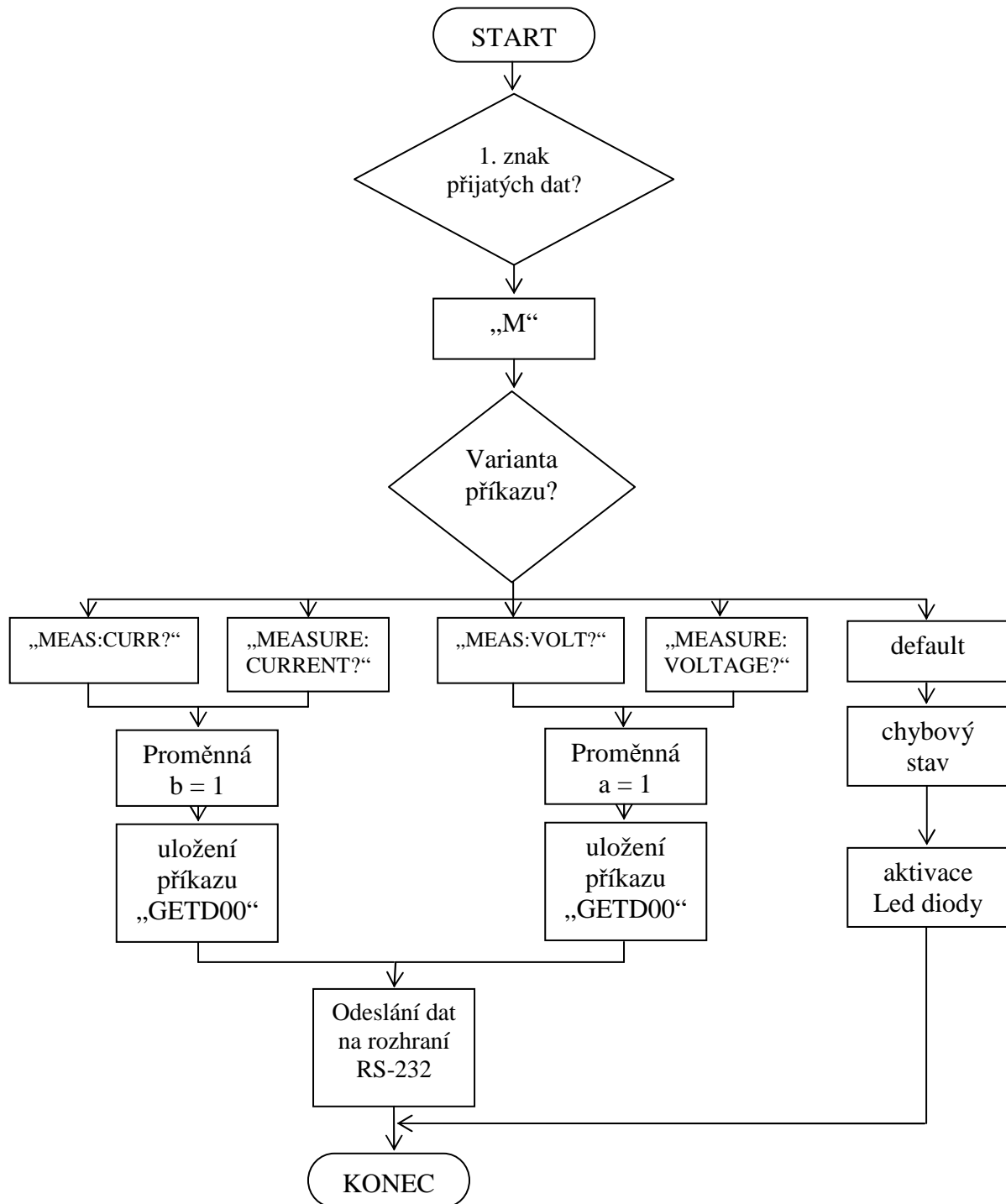
Velmi důležitá součástí instrukční sady jsou příkazy pro měření aktuálního výstupního napětí nebo proudu. Laboratorní zdroj Manson má pro tuto operaci definovaný příkaz GETD00, který opět musí být zakončen ukončovacím znakem „\r“. Na tento příkaz laboratorní zdroj Manson zareaguje tak, že pošle zpět aktuální hodnoty výstupního napětí i proudu, které protékají výstupními svorkami. Laboratorní zdroj posílá data ve formátu 021903570“\r“.

První část zprávy obsahuje aktuální hodnotu výstupního napětí, tedy 2.19 V. Ve druhé části této zprávy je informace o hodnotě výstupního proudu, tedy 0.357 A. Úkolem obslužného software je tedy oddělit informaci o nastaveném napětí či proudu. Poslední číslo před ukončovacím znakem je informace o režimu, ve kterém laboratorní zdroj pracuje. Pokud je poslední číslo 0 jedná se o zdroj konstantního napětí, naopak pokud je poslední číslo 1 jedná se o režim zdroje konstantního proudu.

Pro změření aktuální hodnoty výstupního napětí lze využít příkaz `MEASure:VOLTage?`, přičemž povinná část příkazu je velkými písmeny a nepovinná část je malými písmeny. Pokud první znak přijatého příkazu je písmeno M malé nebo velké, mikroprocesor ověří následující přijaté znaky. Pokud odpovídají jedné z variant uvedeného příkazu, nastaví pomocnou proměnnou. Pomocná proměnná signalizuje požadavek na zaslání aktuální hodnoty výstupního napětí. Špatně zadaný příkaz vede na rozsvícení chybové led diody a ukončení zpracování mikroprocesorem, který přechází do klidového stavu a očekává nově příchozí data. Vyhovující příkaz je převeden mikroprocesorem do vyrovnávací paměti v podobě odpovídající příkazu `GETD00`, který je ukončený znakem „\r“. Tento příkaz je odeslán na sériovou linku RS-232. Laboratorní zdroj Manson zpracuje příkaz a odešle zpět odpovídající hodnoty výstupního napětí a proudu. Při příchodu dat obdrží mikroprocesor požadavek na vyvolání externího přerušení od sériové linky. Mikroprocesor přeruší vykonávání nekonečné smyčky a přesune se do oblasti obsluhy externího přerušení vyvolané jednotkou USART. Dojde opět k uložení přijatých dat do jednorozměrného pole. Potom je nutné rozeznat, co uživatel požadoval. Pomocná proměnná určená pro měření výstupního napětí je nastavena. Podle této proměnné mikroprocesor vybírá vhodná data, které požaduje uživatel. Při požadavku o hodnotu napětí je přijatý řetězec zkrácen jen na hodnoty informující o nastavení výstupního napětí. Řetězec je doplněn ukončovacím znakem „\n“. Hodnota výstupního napětí je uložena do vyrovnávací paměti, která je následně odeslána na USB rozhraní. Pokud by nebyl USB host připojen, dojde k přerušení obsluhy externího přerušení a přijatá data by byla ztracena. V opačném případě dojde k zobrazení požadovaných dat v grafickém prostředí VEE Pro. Mikroprocesor po odeslání dat přechází do klidového stavu, ve kterém očekává příchod nových dat.

Pro určení aktuální hodnoty výstupního proudu je nutné zadat příkaz ve tvaru `MEASure:CURRent?`, přičemž povinná část příkazu je velkými písmeny a nepovinná část je malými písmeny. Je tedy možné příkaz využívat v několika podobách. Každá podoba vrací aktuální hodnotu výstupního proudu. Pokud první znak přijatého řetězce je malé nebo velké písmeno M, mikroprocesor ověří další příchozí znaky. Pokud je vyhodnotí jako příkaz pro měření aktuální hodnoty výstupního proudu, nastaví pomocnou proměnnou. Pomocná proměnná je použita pro správné vyhodnocení přijatých dat z laboratorního zdroje Manson. Odlišnost přijatého příkazu aktivuje chybový stav, který ukončí zpracování a rozsvítí chybovou led diodu. Komunikační modul do laboratorního zdroje opět zasílá data v podobě `GETD00` ukončené znakem „\r“. Reakce laboratorního zdroje Manson je obdobná jako v případě měření výstupního napětí. Opět jsou zaslány hodnoty výstupního napětí společně s hodnotami výstupního proudu. Příchod dat vyvolá externí přerušení a následně uložení příchozích dat. Mikroprocesor na základě nastavené proměnné pro měření výstupního proudu vybere z těchto dat jen hodnoty odpovídající výstupnímu proudu, vhodně je přeskládá a uloží do vyrovnávací paměti. Hodnoty obsahující informaci o výstupním napětí jsou při tomto požadavku vymazány. Na konec je k hodnotě výstupního proudu přidán ukončovacím znak „\n“. Vyrovnávací paměť je potom odeslána na USB rozhraní, pokud není komunikační modul správně připojen, dojde ke smazání přijatých dat. V opačném případě jsou odeslaná data zpracovaná na USB rozhraní a vyhodnocena v grafickém prostředí VEE Pro. Při vhodném nastavení se uživateli zobrazí aktuální hodnota výstupního proudu. Po úspěšném dokončení se

mikroprocesor vrací do klidového stavu, ve kterém očekává příchod nových dat. Popsaný průběh zpracování dat je zobrazen pomocí vývojového diagramu na obr. 4.7. Zobrazený vývojový diagram je zkrácená verze hlavního vývojového diagramu. Pozornost je zaměřena na zpracování dat po rozeznání prvního znaku v podobě písmene M. Blok označený default je aktivován v případě chybného stavu, který může nastat během zpracování přijatého příkazu. Zpracování se v tomto případě dělí na dvě části. První část řeší požadavek na měření výstupního proudu a druhá část se zabývá měřením výstupního napětí.



Obr. 4.7 Vývojový diagram pro měření výstupního napětí a proudu

4.3.3 Aktivace výstupních svorek

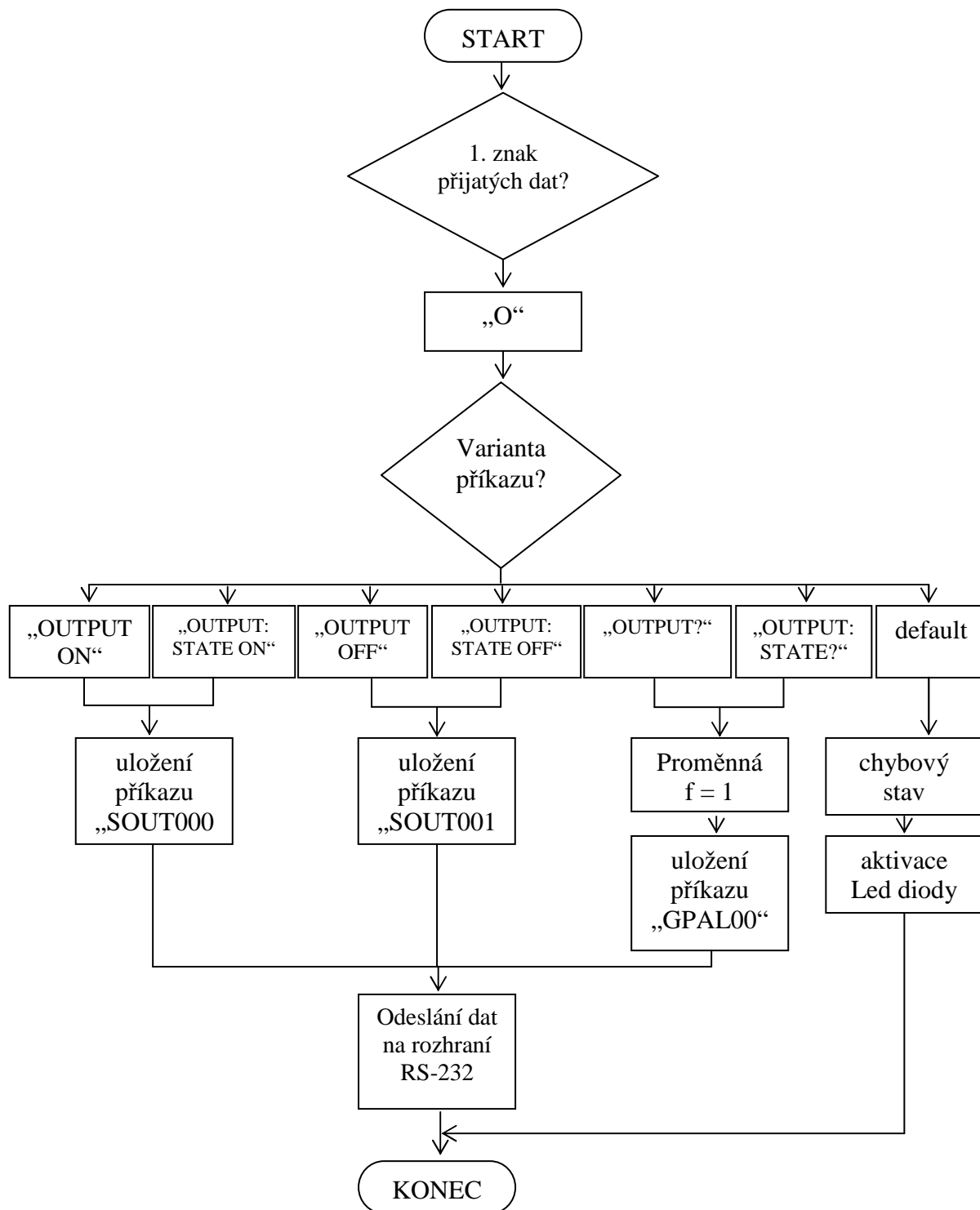
Mezi další důležité příkazy pro práci s laboratorním zdrojem Manson patří příkazy pro aktivaci a deaktivaci výstupních svorek. Aktivní výstupní svorky jsou tehdy, pokud je možné z laboratorního zdroje odebírat příslušný proud pod určitým napětím. Pokud naopak výstupní svorky nejsou aktivní, není možné, aby do zátěže procházel proud. Na displeji laboratorního zdroje je symbol označující sepnuté či naopak rozepnuté výstupní svorky.

Pro sepnutí respektive rozepnutí výstupních svorek je možné využít příkaz ve tvaru `OUTPUT[:STATE] {OFF|ON}`, přičemž v hranaté závorce je uvedena nepovinná část příkazu. Ve složených závorkách jsou uvedeny parametry příkazu pro rozepnutí (OFF) nebo sepnutí (ON). Tyto hodnoty je možné nahradit i čísly reprezentující logické úrovně. Logická 0 odpovídá rozepnutí výstupních svorek, naopak logická 1 odpovídá sepnutí výstupních svorek. Pokud mikroprocesor obdrží ukončený příkaz začínající malým nebo velkým písmenem O, otestuje následující znaky. Při dodržení jednoho z možných formátů příkazu pro rozepnutí výstupních svorek naplní vyrovnávací paměť příkazem `SOUT001`, který ukončí znakem „\r“. Tento příkaz je pak dále odeslán na sériovou linku RS-232 a předán laboratornímu zdroji Manson. Tento příkaz je vyhodnocen a následuje rozpojení výstupních svorek. Po odeslání tohoto příkazu není možné z laboratorního zdroje odebírat žádný výstupní výkon. Display zobrazuje symbol rozepnutých svorek. Pro zpětné sepnutí je nutné použít opačný příkaz, tedy v příslušném formátu, jak bylo uvedeno výše. Proces mikroprocesoru je obdobný jen s tím rozdílem, že na konci přijatého příkazu je hodnota ON nebo 1. Tedy žádost o sepnutí svorek. Při vyhodnocení takového příkazu je vyrovnávací paměť komunikačního modulu naplněna příkazem `SOUT000`, který je ukončen znakem „\r“. Po odeslání příkazu do laboratorního zdroje Manson dojde k sepnutí výstupních svorek. Na výstupu zdroje je možné odebírat výstupní výkon odpovídající nastaveným pojistkám. Tedy maximální výstupní výkon je stanovený hodnotami napěťové a proudové pojistky. Symbol rozepnutých svorek se změní na označení sepnutých svorek. Obdobně jako v předchozích příkazech pokud dojde příkaz, který neodpovídá přesně definovaným podmínkám je vyhodnocen jako chybný. Mikroprocesor přeruší zpracování příkazu, vymaže příkaz z přijímací paměti a rozsvítí chybovou led diodu. Nakonec přechází do režimu, kdy očekává příchod dalšího příkazu.

Pro ověření stavu výstupních svorek je určen příkaz ve tvaru `OUTPUT[:STATE]?`, přičemž v hranatých závorkách je uvedena nepovinná část příkazu. Pokud mikroprocesor vyhodnotí na správné pozici znak „?“ dojde k nastavení pomocné proměnné pro signalizaci stavu výstupních svorek. Do vyrovnávací paměti je uložen příkaz `GPAL00` ukončen znakem „\r“. Příkaz je následně odeslán pomocí sériové linky RS-232 do laboratorního zdroje Manson. Tento příkaz vrací hodnoty obsahující kompletní informace zobrazované na displeji laboratorního zdroje Manson. Tedy aktuální hodnoty výstupního proudu, napětí a výkonu nebo také hodnoty nastavené proudové či napěťové pojistky. Mikroprocesor ovšem zajímá pouze informace o stavu výstupních svorek. Z celých dat se tedy zaměřuje na jeden konkrétní údaj, který nabývá dvou stavů. Pokud údaj nabývá hodnoty logická 1, znamená to, že jsou výstupní svorky sepnuty, naopak pokud je údaj v logické 0, jsou výstupní svorky rozepnuty. Pokud tedy je nastavená příslušná pomocná proměnná a laboratorní zdroj odešle požadované hodnoty, dojde k vyvolání externího přerušení. Mikroprocesor přechází do režimu obsluhy externího přerušení vyvolaného jednotkou USART. V rámci této obsluhy dochází ke zpracování přijaté informace. Vyhodnocená data jsou poté již obvyklým způsobem uložena do vyrovnávací paměti a odeslána na USB rozhraní, kde při správném nastavení grafického prostředí VEE Pro jsou zobrazena uživateli. Po úspěšném dokončení je vynulována vyrovnávací paměť a mikroprocesor očekává příchod dalšího příkazu.

Popisovaný průběh zpracování je zobrazen pomocí vývojového diagramu na obr. 4.8.

Je to opět zkrácená verze hlavního vývojového diagramu se zaměřením na zpracování dat následující po přijetí prvního znaku v podobě písmena O. Blok označený default se stává aktivní při vyhodnocení jakékoliv chyby při zpracování dat. Dále je nutné rozlišovat požadavky na sepnutí, rozepnutí a stav výstupních svorek. Každá část využívá odlišný příkaz, který je uložen do vyrovnávací paměti a odeslán do laboratorního zdroje Manson.



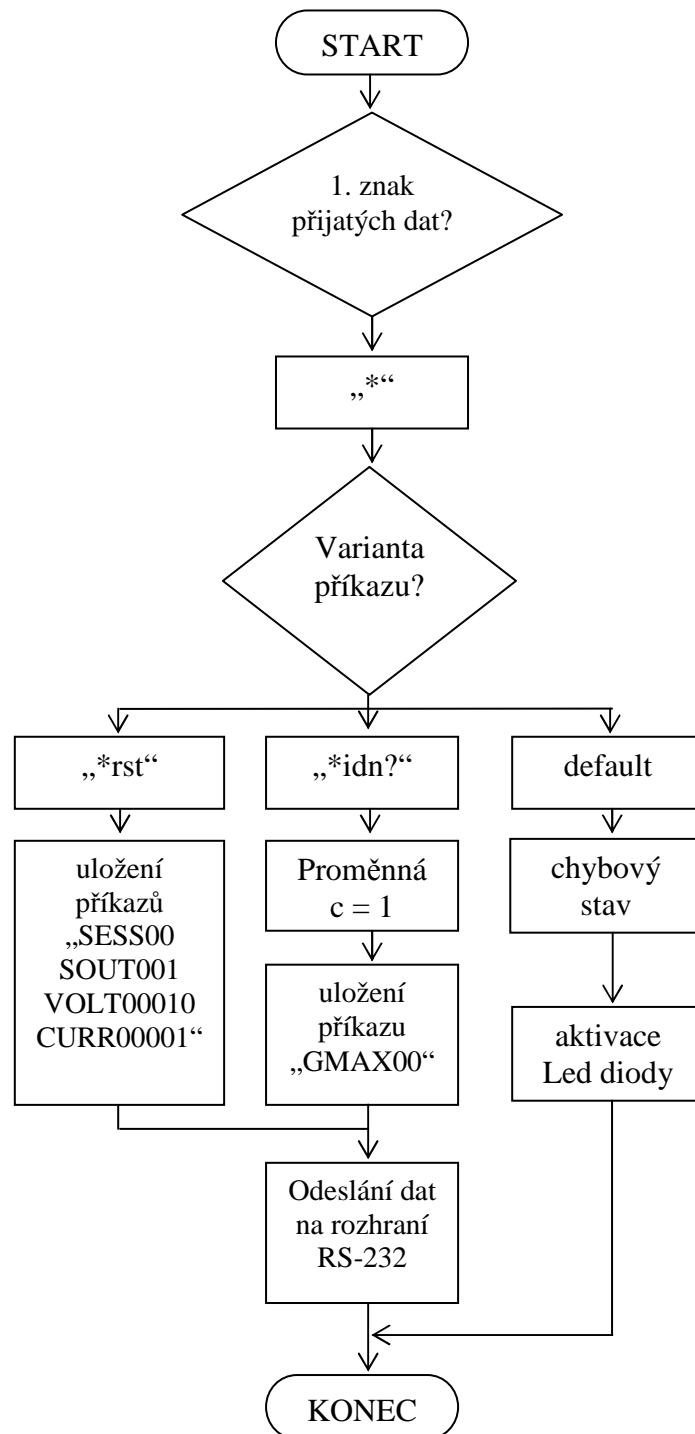
Obr. 4.8 Vývojový diagram pro aktivaci a deaktivaci výstupních svorek

4.3.4 Počáteční nastavení laboratorního zdroje

Laboratorní zdroj Manson neobsahuje příkaz, který by nastavil laboratorní zdroj do počátečního nastavení. Tedy výchozí nastavení, které většinou měřicí přístroje nastaví po zaslání příkazu reset, který je možné zapsat formou příkazu ve tvaru *rst. Po odeslání resetovacího příkazu se většinou měřicí přístroj nastaví na nejmenší nastavitelné výstupní parametry. V případě napájecích zdrojů se většinou jedná o nastavení minimálních pojistek. Pro úplnou bezpečnost jsou odpojeny výstupní svorky. Komunikační modul pro laboratorní zdroj Manson doplňuje instrukční sadu příkazů o právě takový resetovací příkaz. Dojde k nastavení minimálních hodnot pojistek, rozpojení výstupních svorek a navázání komunikace s osobním počítačem. Pokud tedy mikroprocesor rozezná příchozí příkaz ve tvaru *rst dojde k naplnění vyrovnávací paměti následující sadou příkazů. Jako první je odeslán příkaz SESS00 tedy příkaz pro navázání komunikace mezi osobním počítačem a laboratorním zdrojem Manson. Po odeslání uvedeného příkazu již není možné ovládat laboratorní zdroj pomocí předního panelu s tlačítky. Jako další příkaz je odeslán pokyn k rozepnutí výstupních svorek, tedy příkaz ve tvaru SOUT001. Laboratorní zdroj rozepne výstupní svorky, potom již není možné ze zdroje odebírat žádný výstupní výkon. Na závěr jsou odeslány příkazy pro nastavení napěťové a proudové pojistky. Hodnoty pojistek jsou zvoleny jako nejmenší možné, tedy pro napěťovou pojistku je to 1 V a pro proudovou pojistku je to 0,01 A. Příkazy jsou odeslány ve tvaru VOLT00010 pro nastavení napěťové pojistky a pro nastavení proudové pojistky CURR00001. Všechny použité příkazy musí být ukončeny znakem „\r“. Poslední činností příkazu je zhasnutí chybové led diody, která indikuje, že při předchozím zpracovávání informací došlo k nějaké chybě. Tedy po vykonání těchto příkazů je laboratorní zdroj Manson připraven k použití v základním nastavení. I pro tento příkaz platí, že při špatném zadání dojde k rozsvícení chybové led diody a přerušení zpracovávání příkazu. Mikroprocesor vymaže vyrovnávací paměť a přechází do klidového režimu, kdy očekává příchod dalších dat.

Mezi běžně používané příkazy lze zařadit i příkaz pro identifikaci měřicího přístroje. Většinou se jedná o příkaz ve tvaru *idn?. Na tento příkaz by měl měřicí přístroj odeslat identifikační údaje. Většinou se jedná o název přístroje, sériové číslo a případně typ rozhraní, přes které komunikuje. Komunikační modul vytváří identifikační příkaz pro laboratorní zdroj Manson. Pokud mikroprocesor identifikuje správně zadaný příkaz v uvedeném formátu, dojde k uložení příkazu GMAX00 do vyrovnávací paměti. Příkaz musí být ukončen znakem „\r“. Následně je vyrovnávací paměť odeslána do laboratorního zdroje. Po zpracování příkazu dojde k odeslání informace o maximálních hodnotách, které je schopen laboratorní zdroj nastavit. Tedy přijde odpověď obsahující maximální hodnotu napěťové pojistky respektive maximální hodnotu proudové pojistky. Pro správné vyhodnocení je opět použita příslušná pomocná proměnná, která je nastavena při příchodu příkazu *idn?. Pokud mikroprocesor během zpracování dostane odlišné znaky identifikačního příkazu, dojde opět k rozsvícení chybové led diody a přestane se zpracovávat přijatý příkaz, který se odstraní z paměti a očekává se příchod dalšího příkazu. Po vyvolání externího přerušení mikroprocesor opět přechází k obsluze přerušení, ve kterém opět vyhodnotí přijatá data. Na základě vyhodnocení pomocné proměnné procesor nepoužije získaná data o maximálních hodnotách, ale vyrovnávací paměť naplní řetězcem „Manson,SDP-2405,0,ver.1.01“. Navíc je vyrovnávací paměť doplněna o jedinečné číslo, podle kterého dojde k identifikaci laboratorního zdroje z rozsahu čísel 0 až 9. Každý laboratorní zdroj tedy bude odlišen pomocí koncového čísla identifikačního řetězce. Poté je vyrovnávací paměť doplněna o ukončovací znak „\n“ a takto připravená data jsou odeslána na USB rozhraní, kde jsou opět zpracována v grafickém prostředí VEE Pro. Pokud není správně nastavené USB rozhraní, dochází opět k vymazání dat a přechodu mikroprocesoru do klidového režimu, ve kterém očekává příchod nových dat.

Popisované zpracování resetovacího a identifikačního příkazu je zobrazeno pomocí vývojového diagramu na obr. 4.9. Jedná se opět o zkrácenou verzi hlavního vývojového diagramu, který se zabývá zpracováním dat po identifikaci prvního znaku v podobě *. K aktivaci bloku default dochází při chybových stavech, které odhalí mikroprocesor v průběhu zpracování přijatého příkazu. Po identifikaci prvního znaku pole dochází k rozhodování o variantě příkazu. Z toho plynou dvě různé cesty zpracování. Jedna cesta se zabývá zpracováním resetovacího příkazu. Druhá cesta zpracování popisuje práci s identifikačním příkazem laboratorního zdroje Manson.



Obr. 4.9 Vývojový diagram pro reset a identifikaci laboratorního zdroje

4.3.5 Časová prodleva

Laboratorní zdroj Manson není schopen zpracovávat příkazy, které přicházejí jeden za druhým bez časové prodlevy. Pro správné vyhodnocení a splnění požadavku potřebuje mezi jednotlivými příkazy určitou časovou prodlevu, ve které nepřicházejí žádná data. Tento časový úsek je možné nastavit v grafickém prostředí VEE Pro mezi jednotlivými příkazy je vložena čekací pauza definovaná uživatelem. Druhá varianta je vytvoření časové prodlevy po zpracování přijatého příkazu přímo v komunikačním modulu. Pomocí vhodného upravení hlavní smyčky lze vytvořit požadovanou časovou prodlevu mezi odesílanými příkazy. Dojde ke kompletnímu odeslání vytvořeného příkazu, po kterém následuje časová prodleva, ve které nejsou posílána žádná data na sériovou linku RS-232. Po dobu časové prodlevy ovšem komunikace musí pokračovat přes rozhraní USB, kdy dochází ke zpracování dalších přijatých příkazů. Laboratorní zdroj potřebuje minimální časovou prodlevu 30 ms pro správné vyhodnocení přicházejících příkazů. Tento čas byl zjištěn testováním v grafickém prostředí VEE Pro. Zasláním série příkazů bylo zjištěno, že laboratorní zdroj Manson je schopný přijímat více příkazů po sobě, pokud se nejedná o příkazy, které nastavují nebo požadují návrat nastavené nebo změřené hodnoty. Lze tedy poslat bez časové prodlevy například příkaz pro navázání komunikace s laboratorním zdrojem společně s příkazem pro nastavení napěťové pojistky. Ovšem v případě, že laboratorní zdroj dostane bez časové prodlevy příkazy pro nastavení napěťové a proudové pojistky v krátkém čase po sobě, nestihne je správně vyhodnotit. Nastaví prvně přichodzí příkaz správně, ovšem druhý příkaz rozezná tak, že pozná pouze, o jaký parametr se jedná a ten nastaví chybně. Chybné nastavení je většinou reprezentováno vysokou hodnotou nastavovaného parametru. Z tohoto důvodu je do obslužného programu pro komunikační modul vložena časová prodleva mezi jednotlivé odesílané příkazy. K vytvoření časové prodlevy byla použita standardní knihovna pro avr studio, ve kterém je celý obslužný software vytvářen. Knihovna je známá pod označením <util/delay.h>. Pomocí této knihovny lze vytvářet téměř libovolné časové prodlevy. Časové zpoždění je odvozeno od pracovní frekvence externího krystalu. Komunikační modul využívá pracovní frekvenci 8 MHz. Na základě této frekvence je potom pomocí knihovny zavoláno časové zpoždění `_delay_ms(1)`. Pomocí cyklu `for` (opakování určité části kódu, počet opakování je definováno pomocí proměnné) je vytvořena potřebná časová prodleva. Tento cyklus je nutné zopakovat 30 krát, aby byla vytvořena časová prodleva 30 ms. Cyklus `for` je implementován do hlavního programu. Nekonečná smyčka je doplněna o časovou prodlevu v okamžik, kdy dochází k odeslání dat na sériovou linku RS-232 a dochází k ověřování připojení USB rozhraní pomocí funkce `USB_Task`. Takto upravený hlavní program hlídá, aby nedošlo k nechtěnému odeslání dat před vypršením časové prodlevy 30 ms. Pomocí vytvořené časové prodlevy je možné garantovat zpracování všech odeslaných příkazů do laboratorního zdroje Manson. Pro uživatele odpadá nutnost zadávat časovou prodlevu v grafickém prostředí VEE Pro. Práce s tímto laboratorním zdrojem se stává pohodlnější a názornější. Není nutné přemýšlet, kdy je potřeba časová prodleva a kdy naopak je možné časovou prodlevu vynechat úplně. Jediná nevýhoda spočívá ve zdržení komunikace s laboratorním zdrojem. Ovšem běžný uživatel většinou časové zpoždění nepostřehne.

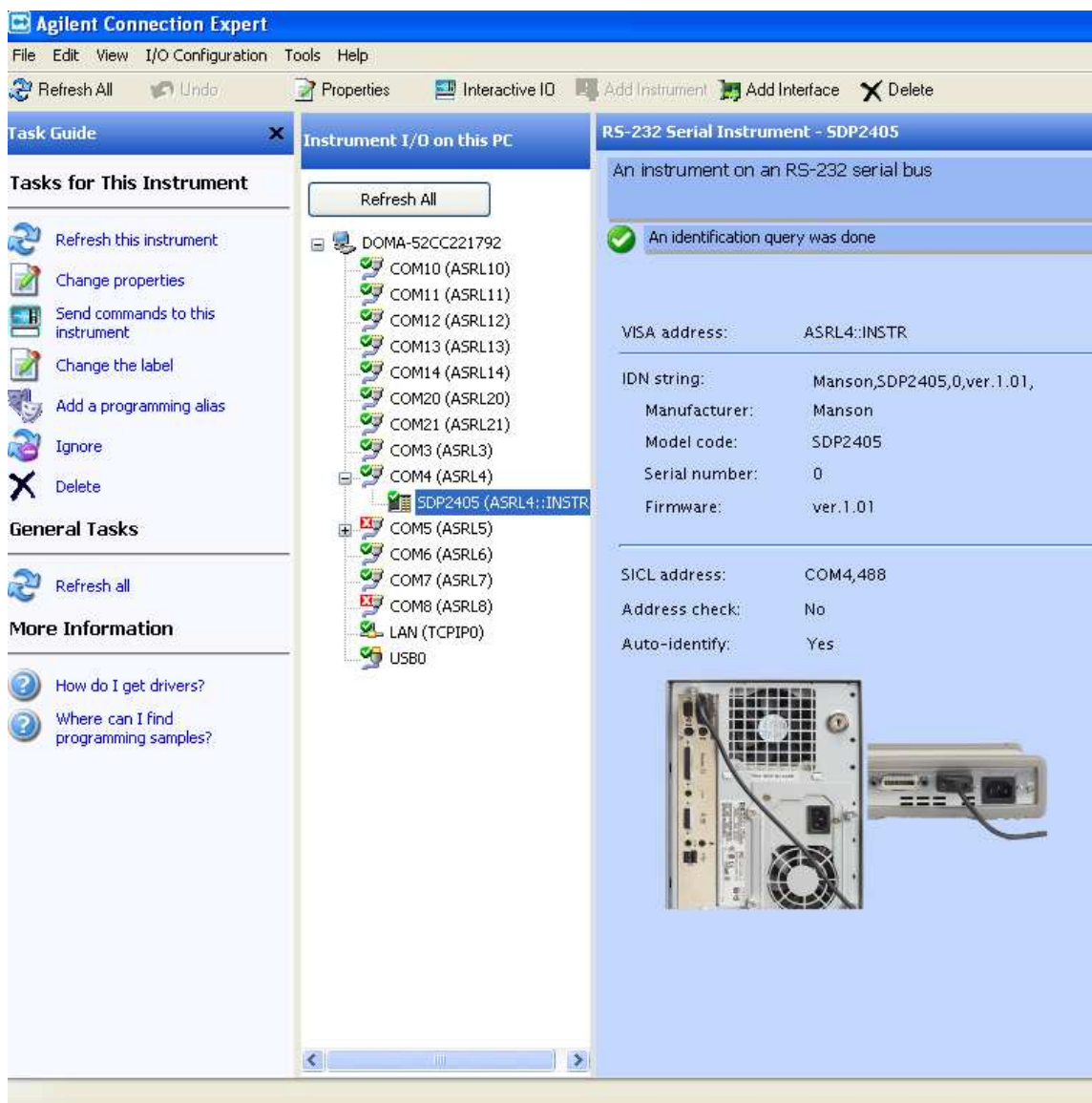
5 LABORATORNÍ ÚLOHA

Poslední část diplomové práce je věnovaná laboratorní úloze, ve které je používán laboratorní zdroj Manson SDP-2405. Komunikační modul bude připojený k laboratornímu zdroji. Na základě této kombinace bude vytvořena nultá laboratorní úloha v předmětu Radioelektronická měření. Laboratorní úloha má sloužit k základnímu seznámení s grafickým prostředím VEE Pro od firmy Agilent Technologies. Svou jednoduchostí a názorností je vhodná pro úvodní laboratorní cvičení. Cílem laboratorní úlohy je stručné seznámení s grafickým prostředím. Na základě seznámení si student osvojí komunikaci s měřicími přístroji, tedy v této úloze se naučí ovládat laboratorní zdroj Manson SDP-2405, který bude spojen s komunikačním modulem. Na závěr úlohy student doplní vytvořený program tak, aby splnil jednotlivé požadované úkoly. Laboratorní zdroj Manson je v nulté laboratorní úloze již využíván. Nicméně pro studenty není práce s instrukční sadou příkazů, kterou standardně používá laboratorní zdroj, příliš názorná. Nově vytvořený komunikační modul by měl studentům výrazně usnadnit práci s tímto laboratorním zdrojem. Pozornost studentů se může zaměřit spíše na pochopení základních principů ovládní grafického prostředí VEE Pro.

5.1 Seznámení s prostředím VEE pro

Grafické prostředí VEE Pro od firmy Agilent Technologies je vytvořeno pro práci s měřicími přístroji, které podporují komunikaci přes standardizované rozhraní. Pomocí následujících rozhraní je možné komunikovat: LAN, GPIB, RS-232 a standard USB. Poslední dvě zmíněná sériová rozhraní jsou detailně popsány v úvodních kapitolách této diplomové práce. LAN (Local Area Network) je sériové rozhraní používané pro vytvoření počítačových sítí. Počítačové sítě LAN jsou nejrozšířenější, používají se po celém světě. Přenosová rychlost těchto sítí se nejčastěji pohybuje kolem 100 Mbit/s. Je možné narazit na sítě, které dosahují přenosové rychlosti až 1 Gbit/s. Moderní měřicí přístroje většinou umožňují připojení do sítí LAN. Paralelní rozhraní GPIB (General Purpose Interface Bus) umožňuje připojení až 15 měřicích přístrojů. Obsahuje 24 pinů a je možné využívat délku sběrnice až 25 m. Přenos dat je asynchronní a řízený hardwarově. Maximální přenosová rychlost je 8 Mb/s, která ovšem bývá typicky nižší. Při komunikaci je použita negativní TTL logika. Logická 0 je reprezentována napětím $> 2,1$ V. Naopak logická 1 je reprezentována napětím $< 0,8$ V. Pro účely nulté laboratorní úlohy jsou potřeba pouze sériová rozhraní RS-232 a USB. Komunikace v grafickém prostředí VEE Pro je řízena knihovnou VISA, která je součástí programového balíčku Agilent IO Libraries Suite. Jedná se o knihovny, pomocí kterých dochází ke komunikaci s měřicími přístroji. Díky této vytvořené knihovně je možné měřicí přístroje ovládat i pomocí různých programových prostředí. Součástí zmíněného programového balíčku je i aplikace s názvem Agilent Connection Expert. Po spuštění osobního počítače je tento program spuštěn. Uživatel si může zkontrolovat stav připojených měřicích přístrojů. Pokud měřicí přístroj má ve své instrukční sadě příkaz pro identifikaci dojde při spuštění aplikace Agilent Connection Expert k navázání komunikace. Po odpovědi je přístroj považován za aktivní a je uveden v seznamu připojených zařízení. Přímou pomocí aplikace Agilent Connection Expert je možné do měřicího přístroje posílat příkazy. Laboratorní zdroj Manson ve své instrukční sadě standardně nemá identifikační příkaz. Při inicializaci dojde k chybě a aplikace Agilent Connection Expert považuje laboratorní zdroj za nepřipojený a není možné s ním komunikovat. Při připojení vytvořeného komunikačního modulu je možné identifikovat laboratorní zdroj Manson standardním průběhem, který

vyžaduje aplikace Agilent Connection Expert. Při inicializaci komunikační modul zareaguje na výzvu o identifikaci a zašle řetězec obsahující název zdroje, typové označení, sériové číslo komunikačního modulu a na závěr aktuální verzi obslužného softwaru. Na obr. 5.1 je uvedený případ, kdy je laboratorní zdroj Manson přihlášen do aplikace Agilent Connection Expert pomocí identifikace komunikačního modulu.

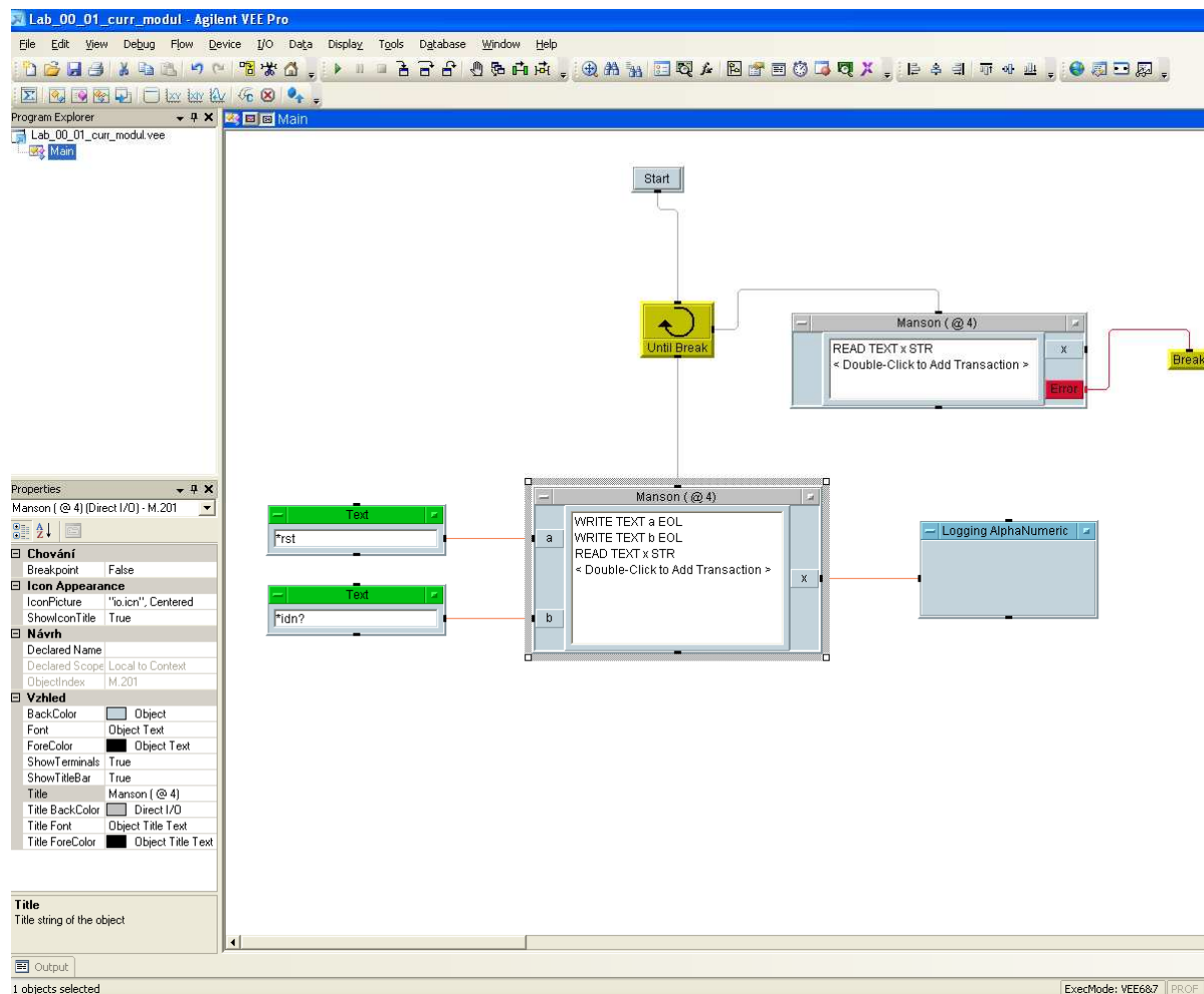


Obr. 5.1 Připojený laboratorní zdroj v aplikaci Agilent Connection Expert

Přehledné prostředí aplikace nabízí celou řadu funkcí, pomocí kterých je možné otestovat měřicí přístroje a změnit jejich nastavení. Jsou zde uvedeny detailní informace, které o sobě měřicí přístroj zasílá při navazování spojení. Přístroje jsou rozděleny podle typu rozhraní, pomocí kterého je příslušný měřicí přístroj zapojen. V případě, že je připojen pomocí sériové linky RS-232 je zde uvedeno číslo přiděleného portu. I v případě komunikačního modulu, který se do operačního systému připojí jako simulace virtuální sériové linky, je zde uvedeno číslo komunikačního portu, které bylo přiděleno při úvodní instalaci. Dále je možné pomocí této aplikace přímo posílat příkazy do jednotlivých měřicích přístrojů a sledovat odezvu v podobě příchozích dat. Mezi další zajímavé funkce patří možnost monitorování komunikační sběrnice. Uživatel může odeslat příkaz do měřicího přístroje a sledovat data, které proběhly po sběrnici, tedy je možné pozorovat detailní průběh komunikace. Je možné zjistit veškeré informace, které si vymění měřicí přístroj s aplikací

Agilent Connection Expert. Bez této aplikace není možné využívat grafické prostředí VEE Pro, které je propojeno s touto aplikací. Přebírá od této aplikace informace o připojených zařízeních.

Laboratorní cvičení předmětu Radioelektronická měření jsou zaměřena na představení programového prostředí Agilent VEE Pro. Programování v tomto prostředí je založeno na sestavování blokových diagramů z definovaných objektů a jejich vzájemného propojení. Na obr. 5.2 je uvedeno základní rozhraní programového prostředí VEE Pro s ukázkou velmi jednoduchého programu, který obsahuje příkazy pro základní nastavení laboratorního zdroje Manson SDP-2405 připojeného pomocí nově vytvořeného komunikačního modulu.

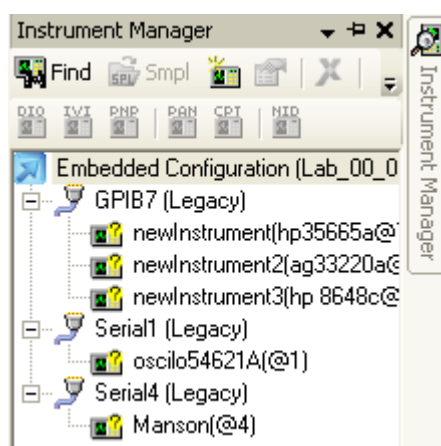


Obr. 5.2 Základní rozhraní programového prostředí Agilent VEE Pro

Rozhraní programového prostředí VEE Pro je rozděleno do tří oken. V levém horním rohu je okno s názvem Program Explorer, ve kterém je možné najít jednotlivé části, které přímo souvisí s vytvářeným programem nebo projektem. V okně nazvaném Properties, které je umístěno přímo pod oknem s názvem Program Explorer, je možné nastavovat vlastnosti vybraných objektů. Je možné měnit například barvu, velikost či název zvoleného objektu. Na uvedeném obr. 5.2 je vybrán objekt s názvem Direct I/O, který slouží pro komunikaci s laboratorním zdrojem a pomocí okna Properties je možné měnit základní parametry tohoto bloku. Hlavní okno projektu s názvem Main je určeno pro tvorbu vlastního ovládacího programu. Konkrétně na obr. 5.2 je vytvořený krátký program pro zaslání dvou příkazů. Jako první je do laboratorního zdroje zaslán příkaz *rst, pomocí kterého je laboratorní zdroj uveden do počátečního nastavení. Druhý příkaz, který je zaslán pomocí uvedeného programu je

příkaz *idn?. Na tento příkaz odpovídá laboratorní zdroj Manson odesláním identifikačního řetězce. Tento řetězec obsahuje informaci o konkrétním laboratorním zdroji. Pomocí přečtení přijatých dat a vhodné zobrazovací jednotky je zobrazen řetězec, který obsahuje informaci o názvu, typu, sériovém čísle a verzi obslužného softwaru používaného laboratorního zdroje.

Celý projekt začíná objektem, který má název START a je možné ho nalézt v položce menu Flow/START. Tento objekt slouží ke spuštění celého vytvořeného programu, který je pospojován pomocí virtuálních vodičů. V tomto případě vede virtuální vodič do objektu, který má název UNTIL BREAK. Objekt se nachází v položce menu Flow/Repeat – UNTIL BREAK. Jedná se o cyklus, který je vykonáván neustále, pokud nedojde k aktivaci bloku s názvem BREAK. K tomuto objektu je na pravé straně připojen objekt, který má název DIRECT I/O. Tento objekt je možné nalézt v položce menu I/O – Instrument manager. Po kliknutí se objeví okno, které je zobrazeno na obr. 5.3. V tomto okně jsou přehledně zobrazeny všechny připojené měřicí přístroje, které jsou rozřazeny dle použitých komunikačních rozhraní.



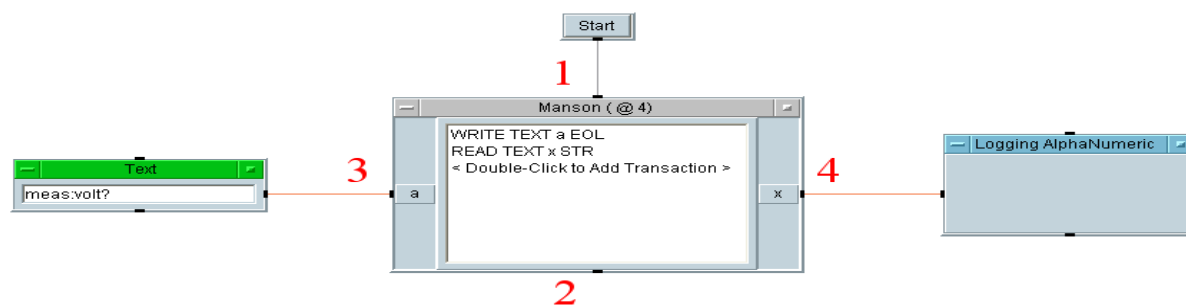
Obr. 5.3 Instrument manager

Objekt DIRECT I/O je možné vložit do projektu kliknutím na příslušný měřicí přístroj. Uprostřed okna je uvedený nápis <Double-Click to Add Transaction>, který je možné vidět na obr. 5.2. Pomocí této volby je možné vytvářet již konkrétní příkazy, které budou odesílány do laboratorního zdroje Manson. Příkazy jsou různého typu např. zápis, čtení, časová prodleva. V tomto konkrétním případě uvedeném na obr.5.2. je provedeno vyčítání dat z laboratorního zdroje Manson. Příkaz uvedený v objektu DIRECT I/O má následující tvar: READ TEXT x STR. Doslova je možné tento příkaz popsat jako čti text z měřicího přístroje a výsledek ve formátu string zapiš na výstup x. String je datový typ, který představuje řetězec znaků. Druhý výstup z tohoto objektu je nazývaný ERROR. Tento výstup je aktivován v případě, že během této operace dojde k výskytu nějaké chyby nebo vyprší čas, po který se čeká na příchozí data z měřicího přístroje. Tedy dojde k překročení času, který je označován jako timeout. Každý měřicí přístroj může mít odlišný timeout. Pokud tedy dojde k situaci, která je vyhodnocena jako ERROR dojde k přerušení vykonávání cyklu UNTIL BREAK pomocí aktivace tlačítka BREAK. Tento objekt je možné nalézt v položce menu Flow/Repeat – BREAK. Výše popsaný blok má za úkol vyprázdnit vyrovnávací paměť osobního počítače, ke kterému je připojený laboratorní zdroj Manson. Někdy může nastat situace, kdy v této paměti zůstanou nezpracovaná data, a může tím docházet k posuvu vyčtených dat. To může vést k porušení časové synchronizace měření.

Dále je na obr. 5.2 uvedena ukázka zápisu jednoduchých příkazů do měřicího přístroje. Tento blok je aktivován po dokončení cyklu UNTIL BREAK, po vyprázdnění

přijímací vyrovnávací paměti. V objektu DIRECT I/O, s názvem Manson (@4) je zapsána posloupnost příkazů, která se provede po aktivování bloku. Relace pro zápis příkazu má tvar WRITE TEXT a EOL. Po aktivaci objektu DIRECT I/O dojde k zapsání příkazu ze vstupu a do laboratorního zdroje Manson. Celá komunikace je ukončena znakem EOL (End of Line). Grafické prostředí VEE Pro použije k ukončení příkazu nastavený ukončovací znak. Každý měřicí přístroj využívá jiný typ ukončovacího znaku. Kde je možné nastavit ukončovací znak, bude zmíněno v dalším textu. Data obsahující příkazy se zadávají dvěma způsoby. První způsob je možný pomocí objektu TEXT, který je možné nalézt v položce menu Data/Constant – TEXT. Do objektu se zapíše příkaz např. *rst, *idn? a pomocí virtuálního vodiče se spojí objekt TEXT s objektem DIRECT I/O, jak je uvedeno na obr. 5.2. Každý objekt TEXT umožňuje zápis pouze jednoho příkazu. Pro další příkazy je nutné použít nové objekty. Pro správnou funkci objektu DIRECT I/O je nutné doplnit příslušný počet vstupních proměnných. V uvedeném příkladu figurují dvě proměnné. Každá proměnná musí mít svůj vlastní vstupní terminál, který lze vytvořit klávesovou zkratkou CTRL + A, přičemž kurzor myši musí být na správné straně objektu DIRECT I/O. To v praxi znamená, že pokud uživatel chce vytvořit vstupní terminál, musí podržet kurzor myši na levé straně objektu DIRECT I/O, pokud chce vytvořit výstupní terminál, musí být kurzor myši na opačné straně objektu. Vstupní terminál je označen podle proměnné, která je poté součástí vkládaného příkazu. Příkaz je možné zapsat do objektu DIRECT I/O i přímo. Tedy zápis poté bude vypadat následovně WRITE TEXT “*rst” EOL. Požadovaný příkaz je nutné zapsat do uvozovek. Takto je možné pod sebe vkládat několik příkazů, které budou postupně zasílány do laboratorního zdroje Manson. Pokud je příkaz dotazovacího typu je potřebné dále nastavit v objektu DIRECT I/O požadavek na čtení dat. Podobným způsobem se vytvoří příkaz, který je zapsán jako READ TEXT x STR. Tento příkaz je nutné umístit pod požadavek na vrácení nějaké změřené či identifikační hodnoty. Objekt DIRECT I/O očekává příchod dat ve formátu String, které budou dostupné na výstupním terminálu x. Výstup x je poté nutné umístit do vhodného zobrazovače. Nejčastěji se používá objekt nazývaný Logging Alpha Numeric, který slouží pro zobrazení různých formátů dat. Poslední typ příkazů vkládá časovou prodlevu mezi jednotlivé příkazy, kterou jsou sekvenčně vykonávané. Příkaz je zapsán jako WAIT 1s. DIRECT I/O vykoná zápis příkazu, po úspěšném zapsání vytvoří časovou prodlevu 1 s. Poté se vrátí ke zpracování dalších příkazů.

Pro práci v grafickém prostředí je nutné přiblížit řízení toku ve vytvořeném programu. Obvykle používané programovací prostředí C++ používá sekvenční vykonávání příkazů, tedy zpracování příkazů po jednotlivých řádcích. VEE Pro používá obdobný styl zpracování, ovšem s omezením na jednotlivé objekty. Objekt DIRECT I/O vykonává příkazy řádek po řádku až do posledního příkazu. Zpracování mimo objekty je řízeno pomocí spojení virtuálními vodiči. Na obr 5.4 je uveden význam jednotlivých vstupů a výstupů z objektu DIRECT I/O. Objekt DIRECT I/O je možné rozdělit na čtyři části, které vykonávají odlišnou funkci.



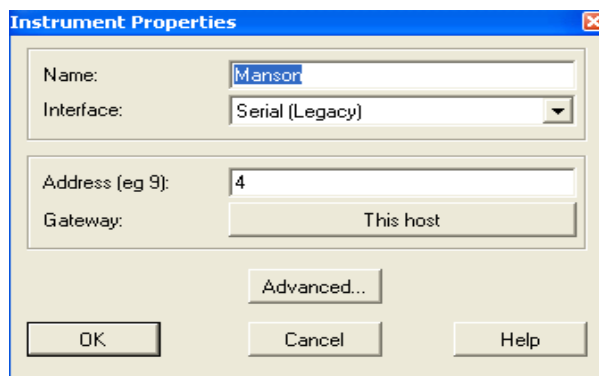
Obr. 5.4 Význam jednotlivých vstupů a výstupů objektu DIRECT I/O

První část je označena na obr. 5.4 číslem 1. Tento vstup lze označit za aktivační. K aktivaci dochází v okamžiku, kdy se na tomto vstupu objeví signál. Po aktivaci dochází k postupnému vykonávání příkazů, které obsahuje objekt DIRECT I/O. Podle uvedeného obr. 5.4 dojde k aktivování objektu po stisknutí tlačítka START. Druhá část je označena na obr. 5.4 číslem 2. Tento výstup je aktivován po dokončení všech příkazů, které obsahuje objekt DIRECT I/O. Tímto způsobem je předáváno řízení toku programu do dalšího navazujícího bloku. Třetí část je označena na obr. 5.4 číslem 3. Tento datový vstup je přiveden do objektu DIRECT I/O a následně je zpracován. Těchto datových vstupů může být na levé straně objektu několik. Čtvrtá část je na obr. 5.4 označena číslem 4. Pomocí tohoto výstupu je možné číst výsledek zpracovaných dat. Opět může být několik výstupních terminálů umístěných na pravé straně objektu DIRECT I/O. Virtuální vodiče je možné spojovat kliknutím vždy na malý obdélníček na libovolné hraně objektu. Jednotlivé virtuální vodiče mají různé barvy podle jejich významu, díky této vlastnosti je vlastní programování jednodušší a mnohem přehlednější. Smazání virtuálního vodiče je možné provést po stisknutí kláves CTRL+SHIFT a kliknutím na příslušný vodič.

5.2 Postup měření

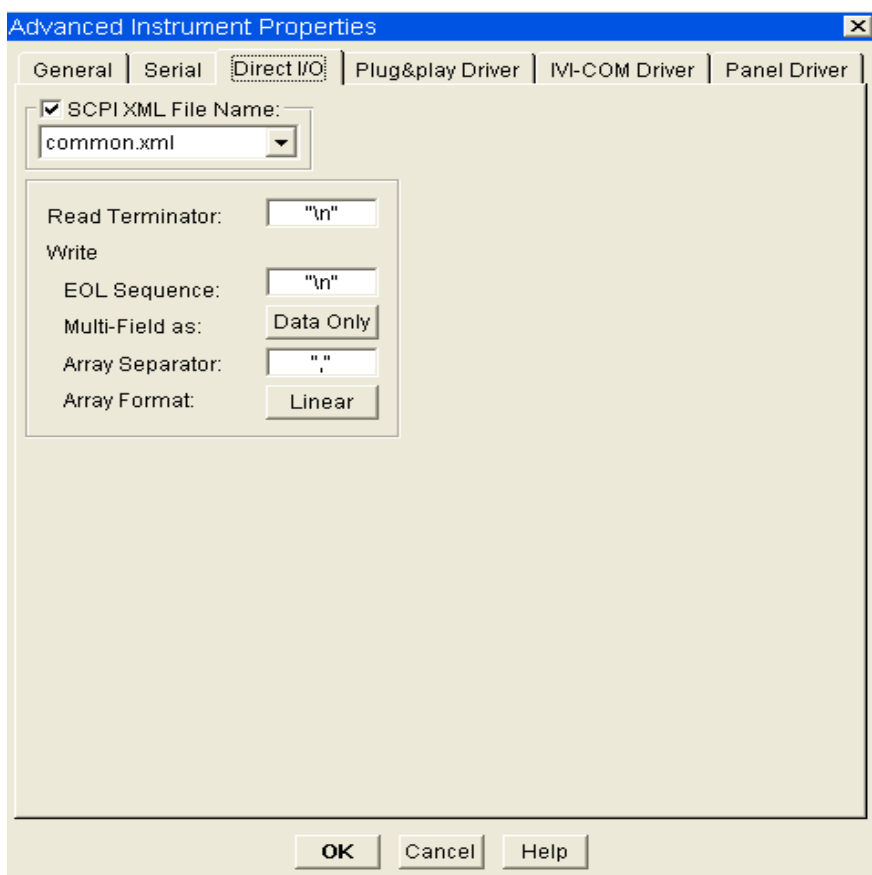
Tato část kapitoly je zaměřena na postup měření popisované laboratorní úlohy. Součástí postupu měření bude i přiblížení odlišností, které do laboratorní úlohy vnese nově vytvořený komunikační modul. Některé části laboratorní úlohy zůstanou velmi podobné, ovšem řešení bude s novým komunikačním modulem snadnější. Z tohoto důvodu bude laboratorní úloha rozšířena a doplněna o nové úkoly, které by měli studentům více přiblížit grafické prostředí VEE Pro.

Před zahájením tvorby vlastního projektu je nutné nastavit parametry komunikace s laboratorním zdrojem Manson SDP-2405. Komunikace probíhala pomocí klasické sériové linky RS-232, kterou standardně používá laboratorní zdroj. Specifikace laboratorního zdroje uvádí následující parametry komunikace: datový formát ASCII, žádný paritní bit, 8 datových bitů, 1 stop bit a doporučená přenosová rychlost je 9600 baudů. Za každým příkazem, který má zpracovat laboratorní zdroj Manson, musí být ukončovací znak reprezentovaný dvojznakem „\r“. Všechny uvedené parametry je nutné nastavit před zahájením komunikace s laboratorním zdrojem. Komunikační modul přináší jednu změnu uvedených parametrů. Každý příkaz, který má být zpracován a následně odeslán do laboratorního zdroje musí být ukončen znakem „\n“. Tato změna je z důvodu, že většina dnes používaných měřicích přístrojů používá právě tento ukončovací znak. Zbývající parametry komunikace zůstávají beze změny. Komunikační modul je sice připojen k osobnímu počítači pomocí USB rozhraní, nicméně v operačním systému je reprezentován jako virtuální sériová linka. Nastavení komunikace je možné nastavit v nabídce I/O-Instrument manager. Okno, které se otevře po vyvolání této nabídky, je uvedeno na obr. 5.3. V tomto okně je nutné vybrat měřicí přístroj, u kterého chceme provést nastavení komunikace. Poté vybrat možnost Properties. Objeví se okno, které je uvedeno na obr. 5.5.



Obr. 5.5 Nastavení komunikace

V tomto okně je možné nastavit název měřicího přístroje a rozhraní, přes které bude měřicí přístroj komunikovat. Pro komunikační modul připojený k laboratornímu zdroji je nutné vybrat možnost Serial (Legacy). Poslední položku, kterou uživatel musí nastavit, je položka Address. Do ní je potřeba vložit správné číslo komunikačního portu, které bylo přiděleno komunikačnímu modulu při prvním připojení do operačního systému. Pro další nastavení je nutné zvolit položku Advanced. Okno, které se objeví po vyvolání této položky, je uvedeno na obr. 5.6.



Obr. 5.6 Nastavení Direct I/O

Okno je rozděleno do několika částí, v každé části uživatel doplňuje parametry komunikace. V okně General se nastaví položka Timeout (sec), která nastavuje časovou prodlevu, po kterou program čeká na odezvu z měřicího přístroje. Standardně je přednastavena tato hodnota na 5 sekund. Další záložka označená Serial umožňuje nastavení

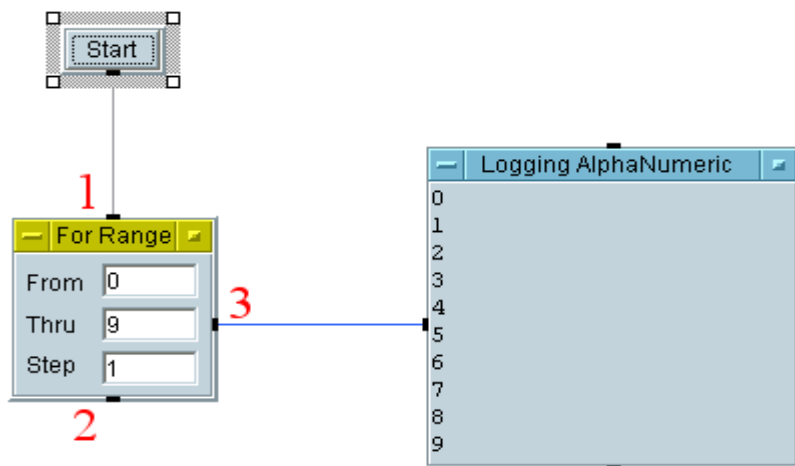
sériové komunikace. Přenosová rychlost se doplní do položky Baud Rate. Počet datových bitů se doplňuje do položky Character Size. Počet stop bitů je nastaveno pomocí volby Stop Bits. Pokud uživatel nepoužívá při sériové komunikaci paritní bit, je nutné nastavit do položky Parity None. Další záložka je označena Direct I/O, která je uvedena na obr. 5.6. V této položce je nutné nastavit zmiňovaný ukončovací znak, který bude doplňován k příkazům, které budou posílány do laboratorního zdroje. Samozřejmě je nutné ukončovací znak doplnit i do položky pro čtení dat z laboratorního zdroje. Obě položky je nutné nastavit na ukončovací znak „\n“. Na tento znak reaguje komunikační modul, který po zpracování příkazu doplňuje ukončovací znak „\r“, na který právě reaguje laboratorní zdroj Manson SDP-2405. Po nastavení a potvrzení všech položek je možné zahájit komunikaci s laboratorním zdrojem, tedy začít vytvářet ovládací program pro splnění laboratorní úlohy.

Při řešení této laboratorní úlohy studenti původně používali instrukční sadu, kterou standardně využívá laboratorní zdroj Manson. Pomocí příkazů museli řešit několik komplikací, které podstatnou měrou odstraňuje nová instrukční sada vytvořená pro komunikační modul. V dalších úlohách, kdy již studenti pracují s lepšími přístroji je nutné tyto špatné návyky z nulté laboratorní úlohy odstraňovat. Z tohoto hlavního důvodu je navržen tento komunikační modul, aby měli studenti k dispozici měřicí přístroj se standardizovanou instrukční sadou již od první laboratorní úlohy. Pro řešení této úlohy je nutné použít dvě skupiny příkazů. První skupinu tvoří příkazy, které nastavují určité parametry laboratorního zdroje a nemají žádnou odezvu, kterou by muselo vyhodnotit grafické prostředí VEE Pro. Do této skupiny patří příkazy pro nastavení napěťové a proudové pojistky, tedy příkaz VOLT XXV respektive příkaz CURR YYA. Dále potom příkaz OUTPUT pro sepnutí výstupních svorek (ON nebo 1) nebo rozepnutí výstupních svorek (OFF nebo 0). Do této skupiny je možné zařadit i příkaz *RST pro počáteční nastavení laboratorního zdroje. Druhou skupinu tvoří příkazy, které se dotazují na hodnotu určitého parametru. Do této skupiny je možné zařadit příkazy pro zjištění nastavené hodnoty napěťové pojistky VOLT? respektive proudové pojistky CURR?. Dále příkaz OUTPUT?, který vrací hodnotu stavu výstupních svorek. Nejdůležitějšími příkazy této části jsou příkazy pro měření napětí MEAS:VOLT? anebo pro měření proudu MEAS:CURR?.

Před samotným vytvářením jakéhokoliv programu je vhodné si ověřit funkce jednotlivých příkazů, které se uživatel chystá používat. Pro ověření příkazů je možné použít program, který je uveden na obr. 5.2. Do pole text je možné zapisovat jiné příkazy, anebo je možné přidat nové objekty text a upravit stávající program pro využití více příkazů. Jen je nutné dodržet správnou posloupnost příkazů a to hlavně v případě, kdy je zasílán dotaz a očekává se na něj odpověď. V tomto případě je nutné pod zápis příkazu vložit příkaz READ k němu doplnit vhodný formát dat. Vytvořit výstupní terminál, který je poté nutné připojit k vhodnému zobrazovači, nejčastěji objekt s názvem Logging Alpha Numeric. Ukázka programu pro správné vyčtení dat z laboratorního zdroje je uvedena na obr. 5.2. Pomocí těchto příkazů a jednoduchého programu si studenti mohou vyzkoušet základní komunikaci s laboratorním zdrojem Manson SDP-2405. Odeslané příkazy mohou kontrolovat vizuálně pomocí displeje laboratorního zdroje či připojené zátěže v podobě žárovky.

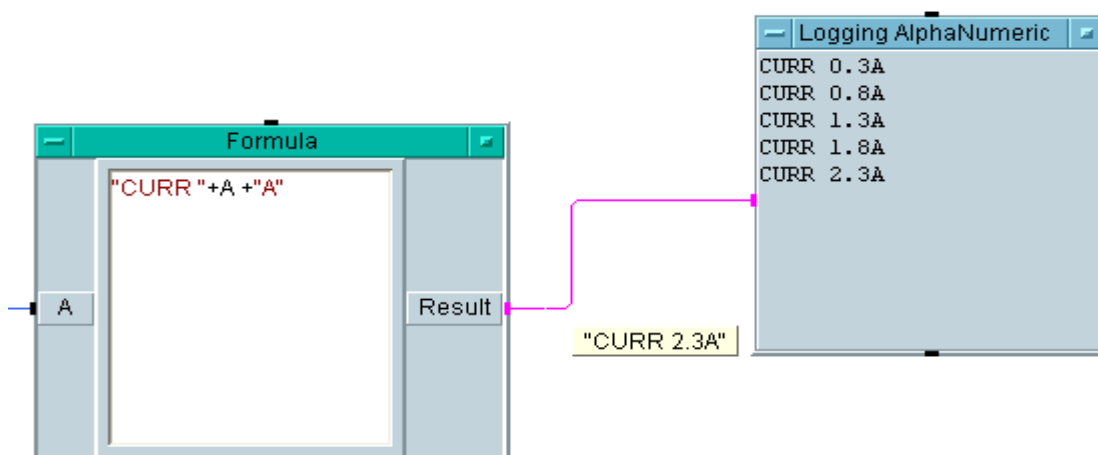
V další části úlohy by měli studenti vytvořit program v grafickém prostředí VEE Pro tak, aby byl schopen proměřit Ampér-Voltovou charakteristiku připojené žárovky. Pro vytvoření programu, který je schopen automaticky měřit A-V charakteristiku je třeba ještě objasnit funkci některých dosud nezmiňovaných objektů. Jedná se zejména o objekty X vs Y PLOT, FORMULA a cyklus FOR RANGE. Objekt X vs Y PLOT patří do skupiny objektů pro zobrazování naměřených dat, je možné ho najít v položce menu Display. Tento objekt je vhodný zejména při zobrazování různých charakteristik. Umožňuje zobrazení více os, tedy více průběhů, přičemž jedna osa je společná pro oba měřené parametry. Dále je možné

změřená data vynášet v logaritmické ose. Pomocí lupy je možné průběhy zkoumat detailně. Další objekt, který je nutné použít pro změření A-V charakteristiky, je cyklus FOR RANGE. Tento objekt je možné přirovnat k cyklu for, který je běžně používán v programovacích prostředích Matlab nebo C++. Cyklus FOR RANGE je zobrazen na obr. 5.7, z kterého je patrné, že objekt má jeden vstup a dva výstupy.



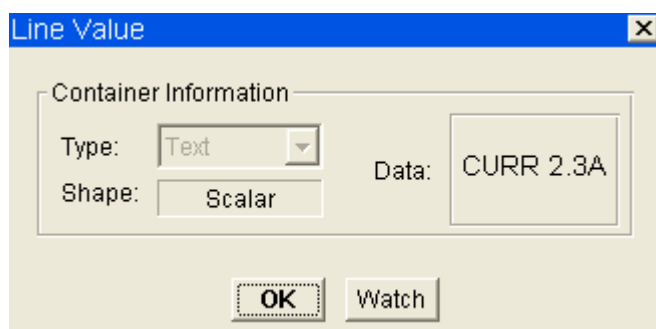
Obr. 5.7 Cyklus FOR RANGE

Číslo 1 označuje aktivační vstup, na kterém když se objeví signál, je zahájen chod cyklu. Po dokončení je aktivní výstup označený číslem 2. Pomocí tohoto výstupu je aktivována další část programu, která je připojena na tento výstup. Při samotném běhu cyklu je aktivován výstup označený číslem 3. Na něj jsou postupně odesílána data ve formátu REAL 64 a to v rozmezí, které si uživatel sám zvolí. Do položky From je vložena počáteční hodnota, do položky Thru je vložena konečná hodnota a do položky Step je vložen krok mezi jednotlivými hodnotami. Tímto způsobem je možné provádět opakovaná měření, tedy i změřit V-A charakteristiku žárovky, která je připojena na výstupní svorky laboratorního zdroje. Ovšem samotná vygenerovaná čísla laboratorní zdroj nedokážou zpracovat, proto je nutné použít objekt FORMULA. Objekt FORMULA je v menu Device. Tento objekt slouží k jednoduchým výpočtům, ale je ho možné využít i pro formátování odesílaných příkazů. Na obr. 5.8 je uveden jednoduchý příklad použití objektu FORMULA.



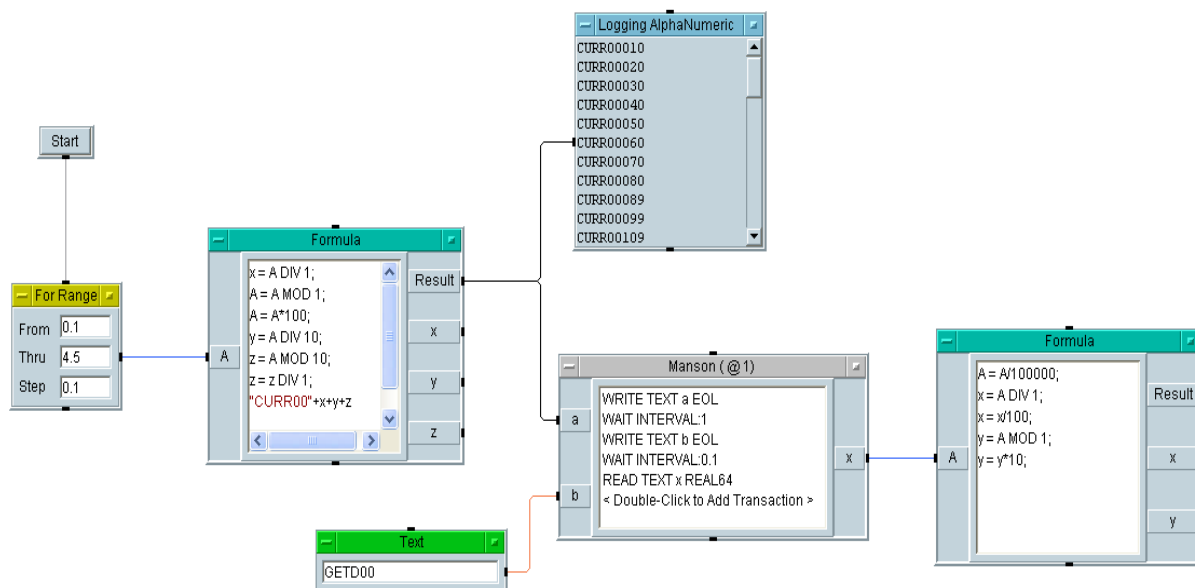
Obr. 5.8 Objekt FORMULA

Jedná se zde o složení příkazu pro nastavení proudové pojistky na laboratorním zdroji Manson, ke kterému musí být připojen vytvořený komunikační modul. Do objektu je odesláno číslo ve tvaru REAL 64. Objekt FORMULA připojí k tomuto číslu uvedený text, který musí být vložen do uvozovek. Číslo, které má být připojené musí být vloženo do objektu pomocí znaku +. Výsledná relace má tedy následující tvar: "CURR"+A+"A". Písmeno A reprezentuje hodnotu čísla, které má být doplněno do příkazu a odesláno do laboratorního zdroje. Výsledný generovaný text je poté zobrazen pomocí objektu Logging Alpha Numeric. Tento způsob vytváření příkazů se používá velmi často v cyklech. Na obr. 5.8 je jeden z virtuálních vodičů zobrazen růžovou barvou a je u něj nápis CURR 2.3A. Tohoto pomocného zobrazení je možné dosáhnout pomocí kurzoru myši. Pokud ukazatel myši přidržíme na jakémkoliv virtuálním vodiči, zobrazí se nám naposledy přenesená hodnota přes tento virtuální vodič. Při kliknutí na virtuální vodič se zobrazí okno, ve kterém je uvedeno více informací o vlastnostech virtuálního vodiče, jak je uvedeno na obr. 5.9.



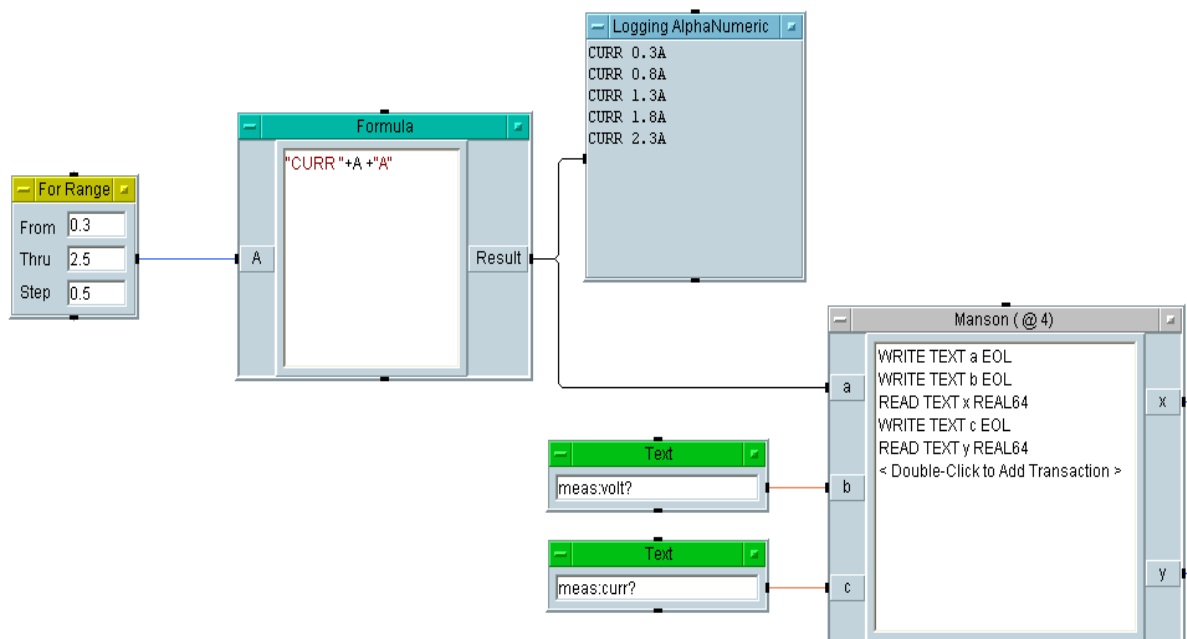
Obr. 5.9 Vlastnosti virtuálního vodiče

Při řešení laboratorní úlohy bez využití vytvořeného komunikačního modulu museli studenti využívat instrukční sadu laboratorního zdroje Manson SDP-2405. Pro splnění zadání bylo nutné vytvořit vhodný cyklus, který bude posílat do laboratorního zdroje postupně zvyšující se hodnoty proudové pojistky a zároveň odečítat hodnoty výstupního napětí a proudu. Změřené hodnoty poté pomocí vhodného objektu zobrazit. Výsledkem tedy je správně zobrazený průběh Ampér – Voltové charakteristiky připojené žárovky. Při řešení bylo nutné správně upravit příkaz pro nastavení proudové pojistky. Pomocí objektu FORMULA a vhodných matematických úprav bylo nutné připravit vhodný příkaz pro zaslání do laboratorního zdroje Manson. Při zpracování příkazu bylo nutné použít matematické operace DIV a MOD. Operace DIV je celočíselné dělení a operace MOD je zbytek po celočíselném dělení. Cyklus FOR RANGE tedy do objektu FORMULA postupně posílal hodnoty z nastaveného rozsahu proudových pojistek, které poté byly upraveny tak, aby výsledný formát příkazu byl CURR00YYY, kde YYY vyjadřuje hodnotu požadované proudové pojistky ve tvaru Y.YY A. Takto upravený příkaz byl odeslán do laboratorního zdroje. Po odeslání a nastavení proudové pojistky byl odeslán příkaz GETD00, který vrací hodnoty výstupního napětí a proudu. Přijatá data bylo nutné opět vhodně upravit pomocí matematických operací DIV a MOD, aby bylo možné zobrazit výsledný průběh Ampér – Voltové charakteristiky. K zobrazení byl využit objekt pro zobrazování s názvem X vs Y PLOT. Tento objekt potřebuje na vstup přivést odděleně hodnoty pro osu y a pro osu x, proto bylo nutné přijatá data rozdělit. Mezi jednotlivé příkazy bylo nutné vkládat časové prodlevy, na které student nesměl zapomenout. Po dokončení měření měl student rozpojit výstupní svorky a ukončit komunikaci s laboratorním zdrojem. Na obr. 5.10 je zobrazena hlavní část laboratorní úlohy, kterou musel student vytvořit pro absolvování laboratorního cvičení. Z popisovaných částí není na obrázku zachycen objekt pro zobrazování Ampér – Voltové charakteristiky s názvem X vs Y PLOT, který nebylo nutné speciálně upravovat.



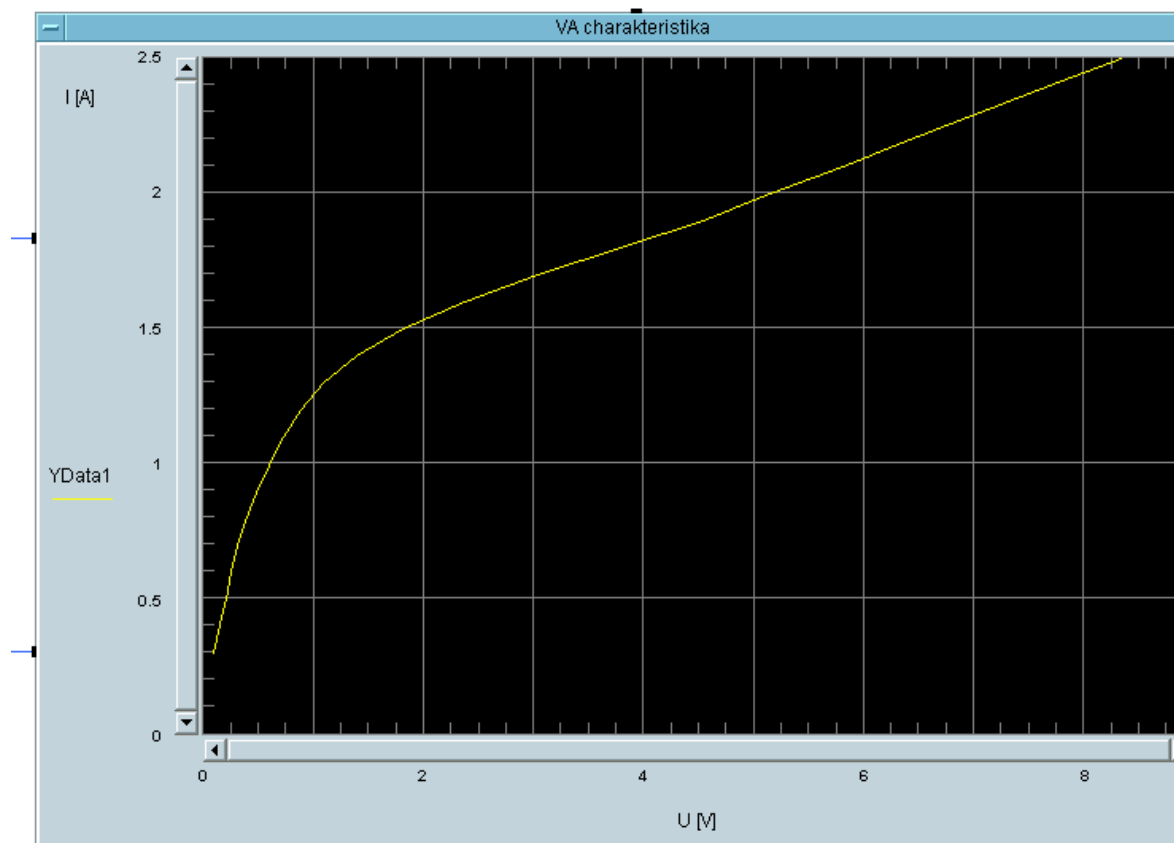
Obr. 5.10 Původní laboratorní úloha

Naopak při řešení laboratorní úlohy s vytvořeným komunikačním modulem dojde k výraznému usnadnění. Není nutné používat matematické úpravy pro získání vhodného tvaru příkazu, ani není potřeba rozdělovat změřené hodnoty výstupního napětí a proudu. Dokonce není potřeba vkládat mezi jednotlivé příkazy časovou prodlevu, která je potřebná pro správné vyhodnocení příkazů. Pro řešení úlohy je využíváno stejných objektů jako při předchozím řešení. Na obr. 5.11 je uvedena část programu nové laboratorní úlohy, která řeší opět zobrazení Ampér – Voltové charakteristiky připojené žárovky. Z obrázku je patrné, že není nutné používat matematické úpravy pro nastavení vhodného příkazu a dále také není potřeba rozdělovat přijatá data výstupního napětí či proudu. Na objekt Manson (@4) je možné připojit zobrazovací jednotku X vs Y PLOT, která již zobrazí změřený průběh charakteristiky.



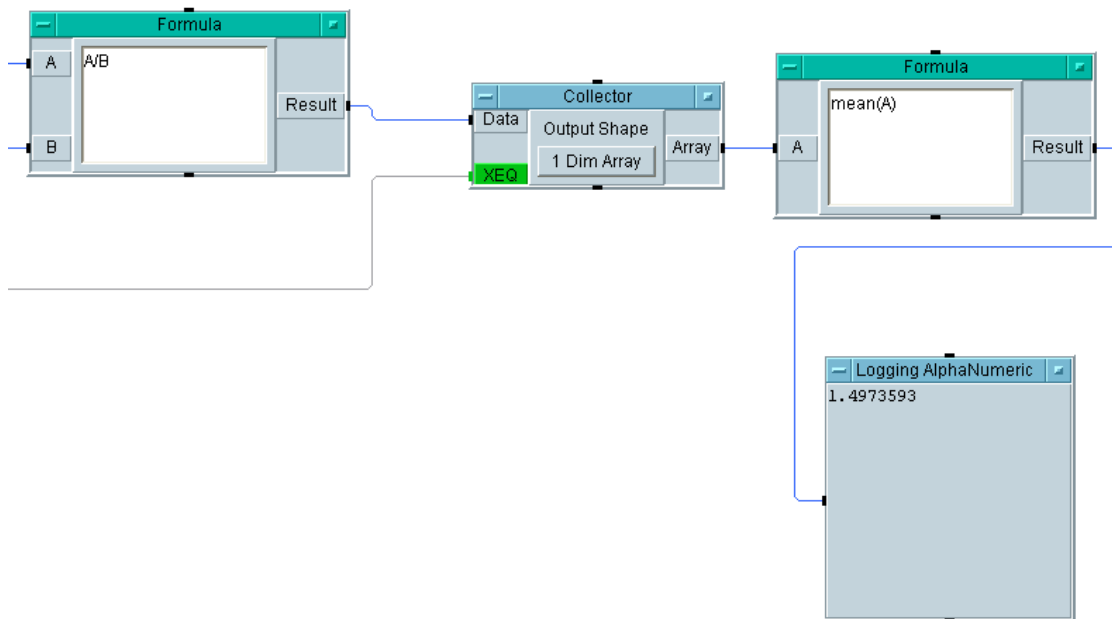
Obr. 5.11 Nová laboratorní úloha

Ukázka změřeného průběhu Ampér – Voltové charakteristiky je uvedena na obr. 5. 12. Na vstupy objektu X vs Y PLOT jsou přivedeny změřené hodnoty výstupního napětí a proudu.



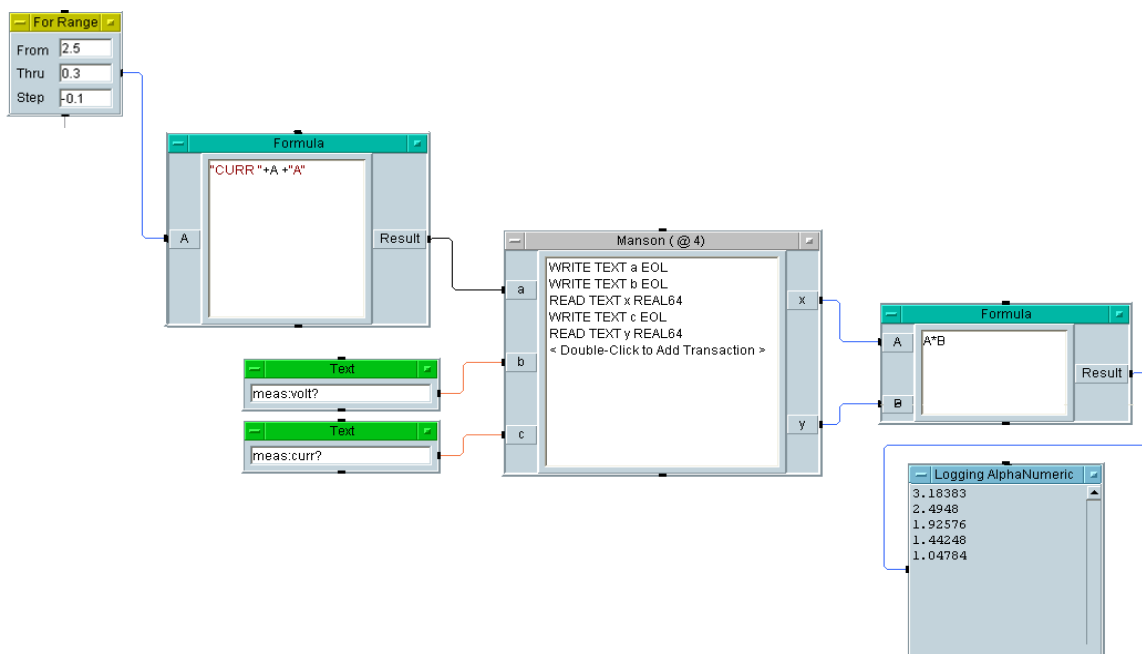
Obr. 5.12 Ampér – Voltová charakteristika žárovky

Nová laboratorní úloha je výrazně jednodušší než stávající laboratorní úloha. Z tohoto důvodu bude nová laboratorní úloha rozšířena o další části. Při měření Ampér – Voltové charakteristiky dochází k postupnému nárůstu proudové pojistky. V průběhu cyklu FOR RANGE žárovka svítí se stále vzrůstající intenzitou. Během této změny se mění i odpor žárovky. Do laboratorní úlohy bude tedy doplněn výpočet odporu žárovky během měření Ampér – Voltové charakteristiky. Výpočet může být realizovaný například v objektu FORMULA, do kterého vstupují změřené hodnoty výstupního napětí a proudu. Vypočítaný odpor bude přiveden do objektu COLLECTOR, pomocí kterého dojde k vyhodnocení střední hodnoty odporu během měření. Objekt COLLECTOR je možné najít v nabídce DATA. Tato komponenta vytvoří pole z postupně načítaných hodnot ze vstupu. Pole je ukončeno aktivací řídicího signálu XEQ. Získané pole je možné upravit v objektu FORMULA, kde pomocí funkce mean(x) je možné vyhledat střední hodnotu měřeného odporu žárovky. Získaná hodnota se potom zobrazí pomocí vhodného objektu. Na obr. 5.12 je uvedena část programu, pomocí které je možné určit střední hodnotu výstupního odporu připojené žárovky. Na vstup objektu FORMULA jsou přivedeny změřené hodnoty výstupního napětí a proudu. Objekt COLLECTOR je ukončen po dokončení cyklu FOR RANGE, který postupně nastavuje hodnoty proudových pojistek. Po dokončení je tedy z jednorozměrného pole vypočítána střední hodnota, která je poté zobrazena v objektu Logging Alpha Numeric.



Obr. 5.13 Určení střední hodnoty odporu žárovky

Po dokončení měření zůstává žárovka svítit. Jednoduchou úpravou úlohy v dalším cyklu FOR RANGE student provede postupné zmenšování proudové pojistky až do stavu, kdy žárovka úplně zhasne. Cyklus FOR RANGE je nutné nastavit od maximální hodnoty proudové pojistky až po nejmenší hodnotu. Krok cyklu je záporný. Během zmenšování proudové pojistky bude dále student určovat výstupní výkon dodávaný do zátěže. Vypočítaný výstupní výkon je možné vizuálně ověřit přímo na displeji laboratorního zdroje. Výpočet výstupního výkonu je proveden opět pomocí objektu FORMULA, do kterého vstupují změřené hodnoty výstupního napětí a proudu. Vypočítané hodnoty je možné zobrazovat ve vhodném objektu, nejčastěji LOGGING ALPHANUMERIC. Na Obr. 5.13 je uvedena část programu, která má za úkol postupně zhasnout žárovku a zároveň určovat výstupní výkon pomocí změřených hodnot na výstupních svorkách zdroje.

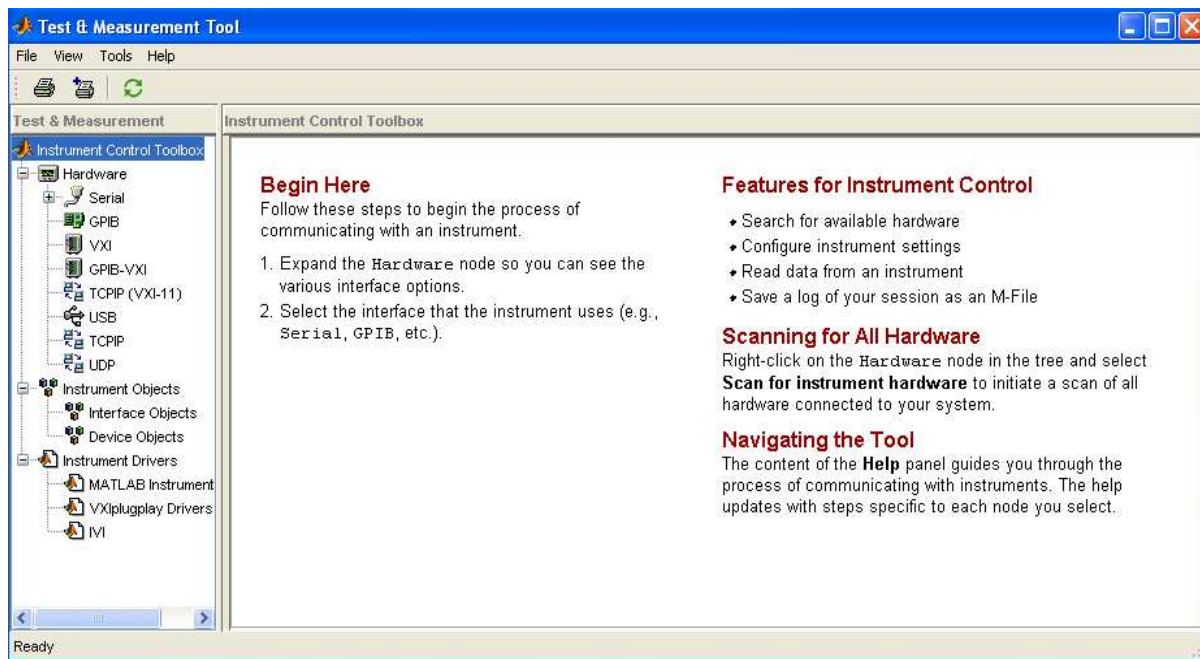


Obr. 5.14 Postupné zhasnutí žárovky a určení výstupního výkonu

Na závěr celé úlohy je vhodné odpojit výstupní svorky od zátěže pomocí příkazu OUTPUT OFF, který je poslán do laboratorního zdroje po dokončení cyklu FOR RANGE. Tímto krokem je dokončena seznamovací laboratorní úloha předmětu Radioelektronické měření. Cílem úlohy je pochopení funkce jednotlivých komponent grafického prostředí Agilent VEE Pro a dále také osvojení si základních postupů při realizaci automatizovaných měření. Po absolvování této úlohy by měl student zvládnout samostatně vypracovat následující úlohy, které obsahuje tento předmět.

5.3 Seznámení s prostředím Matlab

V programovém prostředí Matlab je možné ovládat měřicí přístroje velmi podobným způsobem jako je tomu v grafickém prostředí Agilent VEE Pro. Pro otevření komunikace je potřeba nejprve zjistit přesnou adresu měřicího přístroje. Adresa měřicího přístroje vychází z nastavení připojení v operačním systému. Tedy laboratorní zdroj Manson využívá adresu, kterou dostane přidělenou komunikační modul při prvním připojení a instalaci ovladačů. Nejjednodušším způsobem, jak zjistit přesnou adresu měřicího přístroje, je otevřít si již představovanou aplikaci Agilent Connection Expert. V seznamu připojených měřicích přístrojů je nutné vybrat přístroj, se kterým chceme komunikovat. Laboratorní zdroj Manson je možné nalézt mezi souhrnem sériových portů s označením COM. Po výběru správného portu se v pravé části aplikace zobrazí parametry vybraného měřicího přístroje. Mezi zobrazenými parametry je položka VISA address, za kterou následuje kompletní adresa laboratorního zdroje Manson SDP-2405. Uvedená adresa měřicího přístroje je potřebná pro navázání komunikace v prostředí Matlab. Měřicí přístroj je tedy potřeba nadefinovat pomocí vhodného příkazu a zjištěné kompletní adresy. Definici je možné provést buď přímo v okně Command Window anebo v nově vytvořeném souboru označovaném m-file a to následujícím způsobem `Manson = visa('agilent','ASRL4::INSTR')`. Po této definici je nutné používat při komunikaci s laboratorním zdrojem název *Manson*. Pokud je přístroj nadefinován je možné jeho parametry zjistit v aplikaci Test&Measurement Tool. Tato aplikace obsahuje přehled nadefinovaných měřicích přístrojů je svým obsahem velmi podobná aplikaci Agilent Connection Expert. Pro vyvolání aplikace je nutné do okna Command Window v prostředí programu Matlab napsat příkaz `tmtool`. Na obr. 5.15 je uvedeno okno, které se zobrazí po vyvolání aplikace. Aplikace je rozdělena na dvě části. V levé části okna je Instrument Control Toolbox, ve kterém jsou vypsána všechna možná rozhraní, přes které můžou být připojeny měřicí přístroje. Pokud je nalezen měřicí přístroj na některém s prohledávaných rozhraní, je vypsán v příslušné kolonce pod označením Hardware. V pravé části okna je poté zobrazeno okno, ve kterém má uživatel možnost připojit, nebo odpojit měřicí přístroj, pomocí volby Connection. Pokud je správně připojený měřicí přístroj je dále možné odeslat příkaz v položce Sending Data. V této části okna je nutné nadefinovat datový typ a ukončovací znak `dat`. Pro komunikaci s laboratorním zdrojem Manson je potřeba nastavit datový typ ASCII a ukončovací znak „\n“. Poté je možné v položce Data to Write zvolit příkaz, který chceme odeslat a na závěr potvrdit možností Write. Naopak pokud chceme odeslat dotazovací typ příkazu, je nutné přecházet přijatá data. To můžeme provést v pravé části okna, kde je opět nutné nastavit datový typ ASCII a ukončovací znak „\n“. Na závěr zvolit položku Read, která přečte přijatá data. V položce Instrument Objects, která se nachází hned pod soupisem připojených měřicích přístrojů, je vypsáno zařízení, které bylo nadefinováno pomocí výše uvedeného způsobu. Aplikace umožňuje prohledávání sběrnice pomocí uživatelem nadefinovaného příkazu. Toho se dá využít v případě, že měřicí přístroj nemá ve své instrukční sadě zavedený standardní identifikační příkaz `*idn?`, který odpovídá standardu měřicích příkazů SCPI.



Obr. 5.15 Test&Measurement Tool

Pokud je přístroj správně nadefinován je možné začít komunikovat s měřicím přístrojem. Prvním úkolem je otevření komunikace pomocí příkazu *fopen(Manson)*. Opět je možné tento příkaz zadávat přímo v okně Command Window anebo využít nového souboru pro vytvoření celého programu tzv. m-file. Po úspěšném odeslání tohoto příkazu je programové prostředí Matlab připraveno komunikovat s laboratorním zdrojem Manson, který je připojen pomocí komunikačního modulu. Odeslat příkaz do laboratorního zdroje je možné udělat pomocí příkazu *fprintf*. Pokud chceme například odeslat příkaz pro nastavení počátečních parametrů, bude příkaz vypadat následovně: *fprintf(Manson, '*rst');*. Po odeslání dojde k nastavení počátečních parametrů zdroje a chod ovládacího programu v prostředí Matlab bude ukončeno. Pokud je příkaz dotazovacího typu například příkaz pro identifikaci laboratorního zdroje **idn?*, je nutné po odeslání příkazu použít příkaz pro přečtení přijaté hodnoty od laboratorního zdroje. Příkaz bude vypadat následovně *idn = fscanf(Manson)*. Po odeslání tohoto příkazu se objeví v okně Command Window položka „*idn =*“, do které budou doplněna přijatá data z laboratorního zdroje Manson. Důležité je zmínit, že pokud uživatel chce vidět přijatá data ve zmíněném okně Command Window nesmí příkaz ukončit znakem „;“ v opačném případě zůstanou data nezobrazena, nicméně uložena v proměnné „*idn*“, kterou lze poté nalézt v záložce Workspace. V této záložce jsou uloženy všechny proměnné, s kterými uživatel pracoval a je možné ji najít v levé části pracovního okna programového prostředí Matlab. Obsahem zmiňované proměnné „*idn*“ budou přijatá data z laboratorního zdroje Manson, která budou obsahovat identifikační údaje o laboratorním zdroji. Další příkazy z instrukční sady laboratorního zdroje je možné posílat obdobným způsobem. Syntaxe běžně používaných příkazů v programovém prostředí Matlab je zachována. Pro vytvoření opakovaného měřicího cyklu je možné použít cyklus for, který vytvoří obdobný cyklus FOR RANGE z grafického prostředí Agilent VEE Pro a studenti ho mohou využít pro změření Ampér – Voltové charakteristiky, která je požadována v zadání laboratorní úlohy. V případě potřeby vytvoření časové prodlevy je možné využít příkaz *pause()*, do závorky je nutné vložit časovou prodlevu v sekundách. Na závěr měření je nutné komunikaci s laboratorním zdrojem ukončit pomocí příkazu *fclose(Manson)*. Pokud není odeslán tento příkaz, není možné komunikovat s žádným měřicím přístrojem a programové prostředí Matlab je nutné znovu otevřít a zahájit komunikaci popisovaným způsobem.

6 ZÁVĚR

Cílem a motivací diplomové práce bylo vytvořit komunikační modul pro laboratorní zdroj Manson SDP-2405, který bude usnadňovat komunikaci s tímto zdrojem a bude pro studenty názorněji ovládan. V zadání je dále požadavek, aby bylo možné s laboratorním zdrojem komunikovat pomocí moderního univerzálního rozhraní USB a nově vytvářená instrukční sada pro laboratorní zdroj má odpovídat požadavkům standardu měřicích příkazů SCPI. S pomocí vytvořeného komunikačního modulu byla potřeba také navrhnout jednoduchou laboratorní úlohu, která by měla sloužit pro seznámení studentů s grafickým prostředím Agilent VEE Pro, případně s programovým prostředím Matlab. Komunikační modul by měl studentům názorně ukázat, jak je snadné komunikovat s měřicími přístroji. Navrhovaná laboratorní úloha by měla být velmi názorná a jednoduchá, jelikož by měla sloužit jako úvodní laboratorní úloha předmětu Radioelektronická měření, která slouží pro seznámení studentů s automatizovaným měřením.

V první části práce bylo přiblíženo sériové rozhraní RS-232 pomocí kterého laboratorní zdroj standardně komunikuje s okolím. Také bylo zmíněno průmyslové sériové rozhraní RS-485, které má řadu výhod, nicméně pro komunikaci se využívá spíše v průmyslových aplikacích. Dále bylo přiblíženo univerzální sériové rozhraní USB, které bude použito ke komunikaci s osobním počítačem. USB standard je dnes asi nejrozšířenější rozhraní, které je využíváno ve všech oblastech elektronických zařízení. Laboratorní zdroj Manson není specifický pouze sériovou komunikací pomocí linky RS-232, která je v dnešní době téměř vytlačena a přestává se používat, ale také používá ke komunikaci charakteristickou instrukční sadu příkazů, na které reaguje. Tyto příkazy jsou jedinečné svým charakterem a nejsou běžně používané v grafickém prostředí Agilent VEE Pro. Moderní měřicí přístroje využívají většinou instrukční sadu, která odpovídá standardu pro měřicí příkazy SCPI, který udává možné podoby základních ovládacích příkazů pro jednotlivé měřicí přístroje spadající do stejných klasifikačních tříd.

Ve druhé části práce byl navržen a realizován komunikační modul pro laboratorní zdroj Manson. Pro komunikaci s osobním počítačem byl vybrán vhodný mikroprocesor, který má mimo jiné periferie pro ovládání sériové linky RS-232 a univerzálního rozhraní USB. Byl vybrán mikroprocesor s typovým označením AT90USB1287 od firmy Atmel. Vhodné zapojení vycházející s doporučení, které bylo uvedeno v katalogovém listu součástky, a bylo doplněno o převodník úrovně MAX232. Výsledný komunikační modul obsahuje konektory pro USB připojení, sériové připojení linky RS-232 a také konektor pro ISP programování mikroprocesoru. Napájení komunikačního modulu má za úkol sběrnice USB, která poskytuje připojeným zařízením dostatečné napájecí napětí. Zhotovený komunikační modul po oživení bylo nutné doplnit obslužným programem. Obslužný program má za úkol tři funkce. První obsahem nejnáročnější funkce jsou ovladače pro USB rozhraní. Tuto část obslužného softwaru vyřešil projekt s názvem „LUFÁ“, který obsahuje potřebné ovladače pro vybrané mikroprocesory s implementovaným USB rozhraním. Pomocí získaných ovladačů je možné komunikační modul připojit k operačnímu systému Windows XP, kde po první instalaci je komunikační modul považován za simulaci virtuální sériové linky a je s ním možno komunikovat. Vytvořené ovladače bylo nutné doplnit o řízení obousměrné komunikace mezi osobním počítačem a laboratorním zdrojem. Jedná se tedy o převodník sériového rozhraní RS-232 a USB rozhraním. Pro komunikaci byly použity funkce, které jsou součástí projektu „LUFÁ“. Jednoduchá struktura hlavního programu umožňuje pohodlné doplnění úpravy komunikace. Samotná úprava komunikace je poslední funkcí obslužného programu. Má za úkol přijatý příkaz z USB rozhraní vhodně uložit pomocí jednorozměrného pole a dále upravit do podoby, na kterou reaguje laboratorní zdroj Manson a takto upravený příkaz odeslat na

sériovou linku. Pokud se jedná o dotazovací typ příkazu je opět vhodně zpracován a odeslán. Laboratorní zdroj odesílá reakci na příkaz a v komunikačním modulu je vyvoláno externí přerušení od sériové jednotky USART, kde opět dojde k vhodnému uložení příkazu a jeho úpravě. Upravený příkaz je poté odeslán na rozhraní USB, kde je ve vhodném prostředí zobrazen. Pokud je příkaz zadán chybně, dochází k přerušení zpracování a je ohlášen chybový stav komunikačního modulu, který je indikován pomocí připojených LED diod. Pokud je příkaz opraven a znovu odeslán dojde k opětovnému zpracování a odeslání do laboratorního zdroje Manson. Komunikační modul zpracovává všechny běžně používané příkazy pro ovládání laboratorního zdroje (nastavení napěťové a proudové pojistky, měření výstupního napětí a proudu, sepnutí respektive rozepnutí výstupních svorek, nastavení počátečních parametrů, identifikace laboratorního zdroje, zjištění stavu výstupních svorek, zjištění hodnot nastavených pojistek). Nově vytvořená instrukční sada vychází ze standardu měřicích příkazů SCPI a to ve zkrácené i v plné podobě doporučených příkazů.

Poslední část diplomové práce se zabývá zařazením komunikačního modulu do úvodní laboratorní úlohy předmětu Radioelektronická měření. Laboratorní úloha má sloužit k seznámení studentů s grafickým prostředím Agilent VEE Pro, případně s programovým prostředím Matlab, ve kterém je také možné ovládat měřicí přístroje. Hlavními cíli laboratorní úlohy jsou názornost a jednoduchost řešení. Pomocí nové instrukční sady je možné si ověřit jednoduchou komunikaci s měřicím přístrojem. Zasláním jednotlivých příkazů do laboratorního zdroje a jejich vizuální kontrolou na displeji je možné si ověřit základní ovládání laboratorního zdroje. Potom pomocí představení jednotlivých objektů využívaných v grafickém prostředí Agilent VEE Pro je úkolem vytvořit jednoduchý program pro změření Ampér – Voltové charakteristiky připojené součástky. Pro zvýšení obtížnosti je úloha doplněna o několik výpočtů při měření výstupních parametrů. Pomocí vhodných příkazů je požadováno změření střední hodnoty výstupního odporu a určení výstupního výkonu ze změřených hodnot. Pomocí vhodného zobrazovacího displeje je vynesena požadována Ampér – Voltová charakteristika připojené součástky. Na závěr seznámení je program doplněn o postupné snižování výstupního proudu, při kterém je počítán výstupní výkon dodávaný do připojené součástky. Poslední část navrhované laboratorní úlohy se zabývá seznámením studentů se základním ovládáním programového prostředí Matlab. Pomocí elementárních příkazů je představena komunikace pomocí tohoto prostředí, které má velmi univerzální využití ve všech oblastech simulování a matematických operací. Vytvořená laboratorní úloha je svým obsahem názorná a vhodná pro pochopení základních principů ovládání měřicích přístrojů při automatizovaném měření. Student po absolvování nulté laboratorní úlohy zaměřené na ovládání laboratorního zdroje Manson by měl mít dostatek informací pro splnění následujících laboratorních úloh, které obsahuje předmět Radioelektronická měření. Při použití vytvořeného komunikačního modulu má student možnost seznámení se s instrukční sadou příkazů, která je využívána u většiny moderních měřicích přístrojů. Původní instrukční sada laboratorního zdroje Manson SDP-2405 velmi komplikovala názornost a jednoduchost ovládání měřicích přístrojů. Jedinečné příkazy laboratorního zdroje Manson přinášeli řadu komplikací při řešení úvodní laboratorní hodiny. Student neměl možnost názorného seznámení s grafickým prostředím Agilent VEE Pro jeho pozornost byla zaměřena na pochopení struktury původních příkazů laboratorního zdroje Manson. S vytvořeným komunikačním modulem bude studentovi práce s ovládacími příkazy laboratorního zdroje výrazně usnadněna. Nová laboratorní úloha s využitím komunikačního modulu by měla mít pro studenta daleko větší přínos v názornosti a srozumitelnosti při práci s měřicími přístroji, které jsou využívány ve všech laboratorních úlohách předmětu Radioelektronická měření.

LITERATURA

- [1] TIŠNOVSKÝ, P. *Sériový port RS-232C*. [online], listopad 2008, [cit. 18. 4. 2009], Dostupné z WWW: <<http://www.root.cz/clanky/seriovy-port-rs-232c/>>
- [2] *RS-232* [online], poslední aktualizace 9.11.2009 14:34 [cit. 18. 4. 2009], Wikipedie. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/RS-232>>
- [3] OLMR, V. *HW server představuje – Sériová linka RS-232* [online], prosinec 2005, [cit. 18. 4. 2009], Dostupné z WWW: <<http://hw.cz/rs-232>>
- [4] MATOUŠEK, D. *USB prakticky s obvody FTDI*. Praha: BEN - technická literatura, 2003. 272 s + CD ROM. ISBN:80-7300-103-9
- [5] MATUŠKA, P. *Nová generace USB rozhraní za dveřmi*. [online], listopad 2008, [cit. 18. 4. 2009], Dostupné z WWW: <<http://pcworld.cz/hardware/nova-generace-usb-rozhrani-za-dvermi-3487>>
- [6] FTDI CHIP. *Datasheet FT232BM*. [online], Version 1.8, 25s, 2005 [cit. 18. 4. 2009], Dostupné z WWW: <http://www.ftdichip.com/Documents/DataSheets/DS_FT232BM.pdf>
- [7] ATMEL. *Datasheet AT90USB1287*. [online], Version 7593J-AVR-03/09, 463s, 2009 [cit. 18. 4. 2009], Dostupné z WWW: <http://www.atmel.com/dyn/resources/prod_documents/doc7593.pdf>
- [8] *RS-485* [online], poslední aktualizace 15. října 2009 16:16 [cit. 10. 12. 2009], Wikipedie. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/RS-485>>
- [9] *Universal serial bus* [online], poslední aktualizace 21. prosince 2009 9:55 [cit. 21. 12. 2009], Wikipedie. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/USB>>
- [10] MANSON ENGINEERING INDUSTRIAL LTD. *Downloadable Files* [online]. 2003 [cit. 16. 12. 2009]. Dostupné z WWW: <http://www.manson.com.hk/downloads/4/SDP_User_Manual.pdf>.
- [11] *Open source software* [online], poslední aktualizace 24. listopadu 2009 18:39 [cit. 10. 12. 2009], Wikipedie. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Open_source>
- [12] CAMERA, Dean. *LUFA (Formerly MyUSB) (2009)* [online]. 2009 [cit. 2009-12-10]. Dostupné z WWW: <<http://www.fourwalledcubicle.com/LUFA.php>>.
- [13] *Standard Commands for Programmable Instruments (SCPI)* [online], poslední aktualizace 15. duben 2010 16:15 [cit. 10.3. 2010], Wikipedie. Dostupné z WWW: <http://en.wikipedia.org/wiki/Standard_Commands_for_Programmable_Instrumentation>
- [14] MAXIM – Dallas Semikonduktor. *MAX232 Datasheet*. [online] Version 19-4323; Rev 11; 2/03 [cit. 18. 5. 2010]. Dostupné z WWW: <<http://www.datasheetcatalog.org/datasheet/maxim/MAX220-MAX249.pdf>>
- [15] ELNEC. *User's Manual for T51prog2*. [online] Latest Version Jan. 21, 2010 [cit. 18. 5. 2010]. Dostupné z WWW: <<http://www.elnec.com/download/#manuals>>

PŘÍLOHA

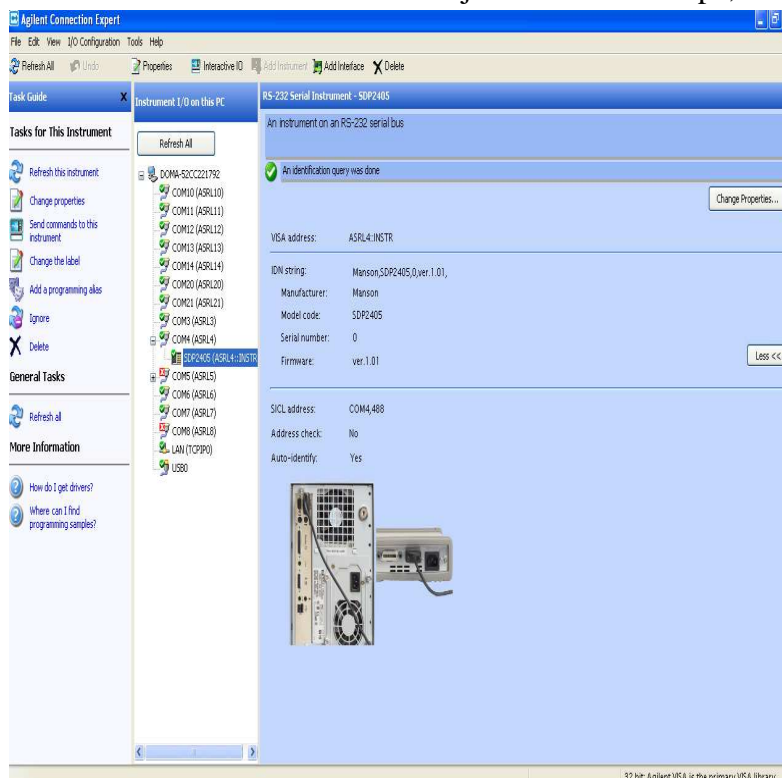
Seznámení s grafickým programovým prostředím Agilent VEE a Matlab

Zadání laboratorní úlohy

- 1) Seznamte se s ovládáním grafického programového prostředí Agilent VEE Pro 9.0. Seznamte se s nastavením komunikačního protokolu pro komunikování přes sériovou linku (RS-232).
- 2) Vyzkoušejte si základní komunikaci s laboratorním zdrojem Manson SDP-2405 (identifikace zdroje, čtení dat, zápis dat do laboratorního zdroje, atd.). Vše kontrolujte vizuálně přímo na laboratorním zdroji.
- 3) Na základě vašich znalostí doplňte vytvořený program v prostředí VEE Pro tak, aby byl schopen proměřit Ampér-Voltovou (dále jen A-V) charakteristiku připojené součástky. Stanovte střední hodnotu odporu součástky a velikost výstupního výkonu. Při realizaci využijte vlastností laboratorního zdroje a vlastností programového prostředí VEE Pro.
- 4) Seznamte se s ovládáním programového prostředí Matlab. Vyzkoušejte si základní komunikaci s laboratorním zdrojem Manson SDP-2405 (identifikace zdroje, čtení dat, zápis dat do laboratorního zdroje). Vše kontrolujte vizuálně přímo na laboratorním zdroji.

Teoretický úvod

Programové prostředí VEE Pro od firmy Agilent Technologies je navrženo pro rychlou tvorbu automatizovaných měřicích pracovišť. Vlastní programování v tomto vývojovém prostředí je velmi podobné prostředí LabView. Prostor VEE Pro komunikuje nebo možná lépe, umí komunikovat s měřicími přístroji pomocí následujících sběrnic: RS-232, LAN, GPIB a USB. Veškerá komunikace je prováděna pomocí knihovny VISA, která je součástí programového balíčku Agilent IO Libraries Suite. Tento balíček je standardně dodáván se všemi produkty Agilent Technologies, které mají nějaký komunikační port. Podobné ovladače a knihovny dodává také např. společnost National Instruments atd. Pomocí knihoven VISA je možné velmi pohodlně ovládat měřicí přístroje i z jiných programových prostředí např. C++, Matlab. V tomto prostředí se využívá hlavičkového souboru visa.h.



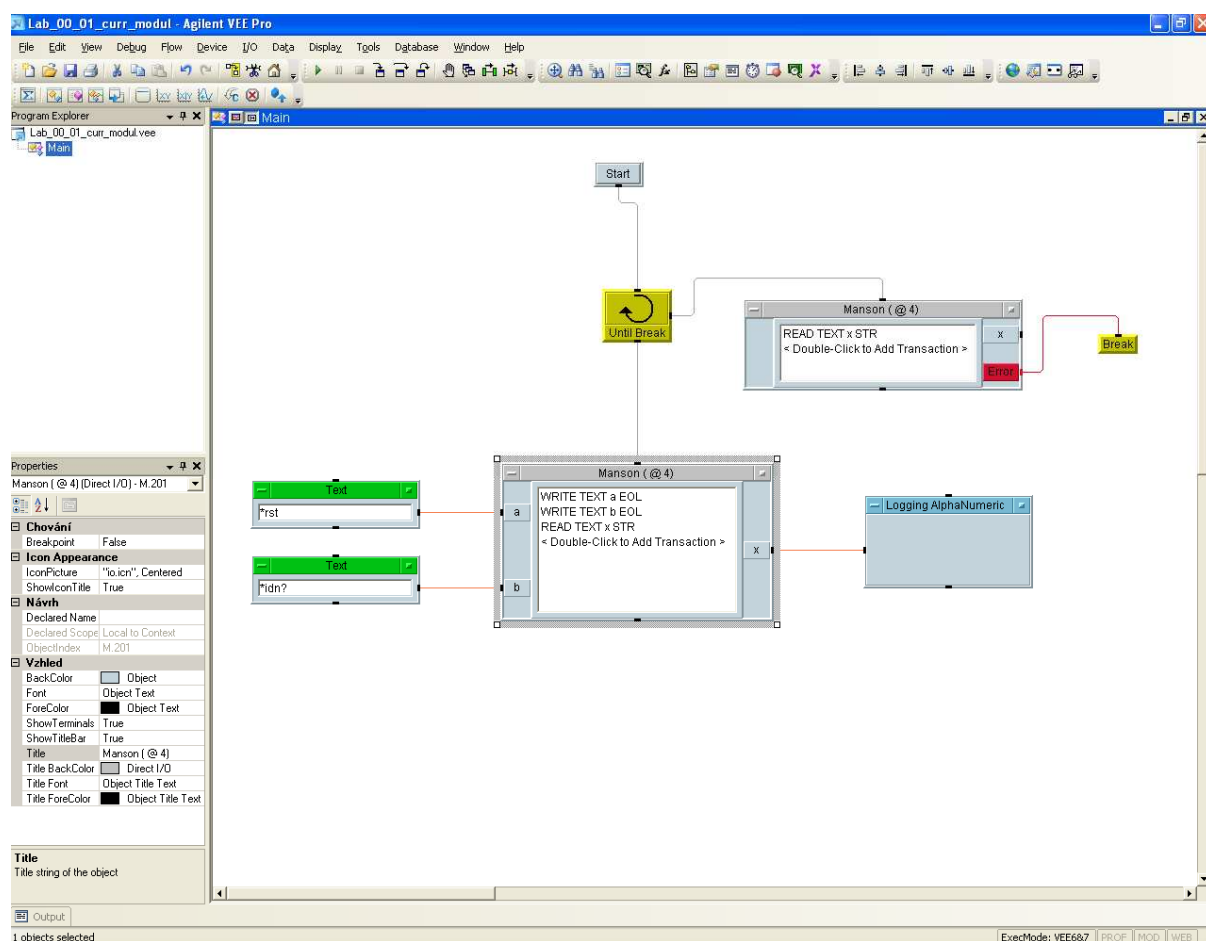
Obr.1 Agilent Connection Expert

Pro snadnou orientaci v připojených zařízeních je po spuštění osobního počítače


spuštěn program Agilent Connection Expert (obr. 1). Tento na ovládání jednoduchý program přehledně informuje o připojených měřicích zařízeních včetně jejich identifikačních řetězců. Pro správnou komunikaci jednotlivých měřicích přístrojů např. pomocí sběrnice GPIB je nutné, aby každé zařízení mělo svoji identifikační adresu. Adresa může nabývat hodnot od 0 do cca 30. Programový balíček dodávaný pod hlavičkou Agilent IO Libraries Suite je vybaven také monitorem této sběrnice, pro snadnou kontrolu při ladění programů.

Laboratorní cvičení předmětu Radioelektronická měření jsou zaměřena na představení programového prostředí Agilent VEE Pro. Zejména jsou zaměřena na tvorbu jednoduchých automatizovaných měřicích úloh a kladou si za cíl poskytnout studentům potřebný prostor pro pochopení problematiky automatizovaných měření a získání zkušeností s touto problematikou.

Na obr. 2 je zobrazeno základní rozhraní programového prostředí VEE s ukázkou velmi jednoduchého programu, který obsahuje pouze příkazy pro počáteční nastavení a identifikaci laboratorního zdroje Manson SDP-2405.



Obr. 2 Základní rozhraní programovacího prostředí VEE Pro

Rozhraní programového prostředí VEE Pro, které spustíte kliknutím na ikonu , je rozděleno do tří oken, která jsou označena modrými lištami. V levém horním rohu je okno s názvem Program Explorer. V tomto okně jsou uvedeny jednotlivé části, které obsahuje vytvářený program nebo projekt. V tomto okně nazvaném Properties je možné nastavovat vlastnosti vybraných objektů, např. barvu, velikost, jméno apod. V tomto konkrétním případě jsou zobrazeny vlastnosti objektu Direct I/O, laboratorního zdroje. Hlavní okno projektu s názvem Main je určeno pro tvorbu vlastního ovládacího programu. V tomto okně (obr. 2) je

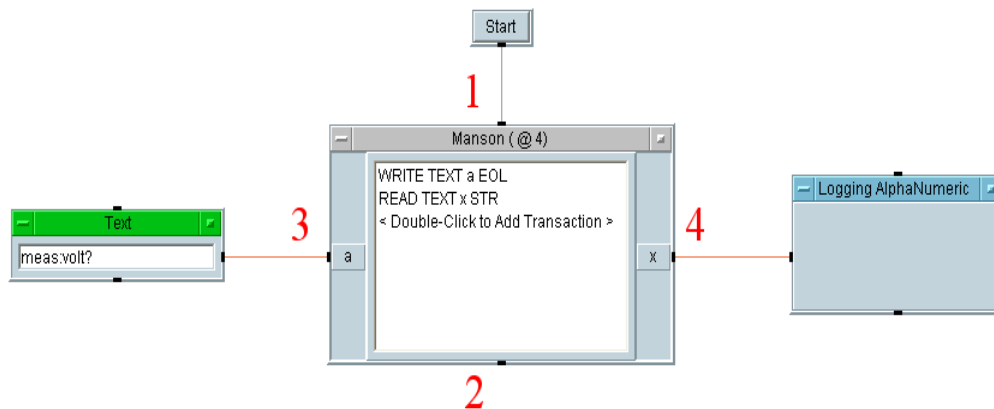
uveden příklad pro zaslání základních ovládacích příkazů pro laboratorní zdroj Manson.

Celý projekt začíná objektem **Start**, který se nachází v položce **Flow/Start**. Tímto objektem se spouští vždy program k němu připojený pomocí „virtuálního“ vodiče. V tomto případě vede „virtuální“ vodič, kterým je předáváno řízení programu do objektu popsaného **Until Break**. Objekt se nachází v **Flow/Repeat-Until Break**, jedná se o cyklus repeat until. K tomuto objektu je na pravé straně připojen objekt **Direct I/O**, který slouží pro komunikaci s měřicími přístroji. Objekt lze nalézt v I/O-Instrument manager..., po kliknutí se objeví okno, které je zobrazeno na obr. 4. V tomto okně jsou přehledně zobrazeny všechny připojené měřicí přístroje. Vše je přiřazeno k příslušným komunikačním sběrnicím. Objekt **Direct I/O** vybereme kliknutím na položku **Direct I/O** vpravo. Uprostřed okna Direct I/O na obr. 2 je napsáno <Double-Click to Add Transaction>. Tento řádek slouží pro přidání příkazu (např. čtení, zápis, čekání, atd.). V tomto konkrétním případě (obr. 2) je provedeno vyčítání dat z laboratorního zdroje. Relace zde uvedená má tvar: READ TEXT x STR, jejíž význam je: čti text z měřicího přístroje a výsledek ve formátu string zapiš na výstup x. Druhým výstupem je z tohoto objektu výstup nazvaný **Error**. Tento výstup je aktivován v případě, že během této operace dojde k výskytu nějaké chyby nebo vyprší čas, po který se čeká na data z měřicího přístroje (**Timeout**). Pokud dojde k výskytu této události pak je ukončen cyklus **Until Break** aktivováním objektu **Break**. Tento objekt lze nalézt v **Flow/Repeat-Break**. Celý tento dosud popsaný blok má za úkol vyprázdnit buffer osobního počítače, který se používá pro komunikaci pomocí sériové linky RS-232. Někdy se totiž stává, že v něm zůstanou nepoužitá data a ta pak zapříčiní jejich posuv a dochází k tomu, že čteme výsledek operace, která již měla být dávno vyčtena a je tím porušena časová synchronizace měření.

Na obr. 2 je dále uvedena ukázka zápisu příkazů do měřicího přístroje. Je uvedena v bloku, který je napojen na dolní hranu objektu **Until Break**. Tento blok je aktivován po skončení cyklu **Until Break**, po vyprázdnění přijímacího bufferu na portu COM. V objektu **Direct I/O**, s názvem Manson SDP2405(@4) je zapsána následující relace WRITE TEXT a EOL. Tato relace provede, při aktivaci tohoto bloku, zápis textu (příkazu) ze vstupu **a** do laboratorního zdroje Manson a celou komunikaci zakončí znakem EOL (*End Of Line*). Prozatím ponechme bez vysvětlení, jaký znak se pro konec řádku používá a jak je možné jej nastavit. Tyto podrobnosti budou vysvětleny dále. Data pro vykonání příkazu vstupují do objektu **Direct I/O** ze vstupu **a** a to ve formě textu. V tomto případě je zde uveden příkaz *rst, který je zapsán v objektu **Text Constant**. Tento objekt je snadno k dispozici v záložce **Data/Constant-Text**. V tomto příkladě (obr. 2) jsou uvedeny příkazy laboratorního zdroje Manson SDP2405. Jde o příkaz *rst, který má za úkol nastavit laboratorní zdroj do počátečního nastavení. Další příkaz *idn? odesílá do laboratorního zdroje dotaz na identifikaci přístroje (typ, označení a výrobní číslo laboratorního zdroje). Uvedené příkazy jsou standardní podle specifikace IEEE 488.2 a doporučení SCPI, nicméně laboratorní zdroj Manson používá z výroby jinou sadu příkazů, kterou upravuje připojený komunikační modul.

Pro pochopení vytváření vlastních ovládacích programů ve VEE Pro je nutné ještě pochopit řízení toku ve vytvářeném programu. V normálním programovacím prostředí např. C++ jsou uživatelé zvyklí, že se program vykonává sekvenčně po jednotlivých řádcích. Toto platí i v grafickém programovém prostředí VEE Pro, ale s omezením pouze na jednotlivé objekty, ve kterých se jednotlivé příkazy vykonávají sekvenčně, tak jak jsou v tomto objektu zapsány. Mimo objekty jsou vazby mezi jednotlivými objekty provedeny „virtuálními“ vodiči, kterými se objekty spojují. Ukázka komunikačního objektu Direct I/O s jednotlivými vstupy a výstupy je uvedena na obr. 3. Význam jednotlivých vstupů a výstupů z objektu Direct I/O je následující:

1 tento vstup lze označit za aktivační (horní) vstup. Pokud se na tomto vstupu objeví signál, dojde k aktivování bloku **Direct I/O**, a začne jeho provádění. V konkrétním případě na obr. 3 je po stisknutí tlačítka **Start** aktivován tento blok a budou se sekvenčně provádět zde zadané relace;



Obr. 3 Význam jednotlivých vstupů a výstupů v prostředí VEE

- 2 po vykonání všech relací v právě aktivním objektu je aktivován výstup 2 (vždy dolní). Tímto způsobem je předáváno řízení toku programu do dalšího navazujícího bloku;
- 3 vstup označený číslem 3 je vstupem datovým. Těchto datových vstupů může být na levé straně objektu více. Data přivedená na tento vstup jsou pak objektem zpracována;
- 4 výsledek zpracování je pak možné číst z výstupu na pravé straně.

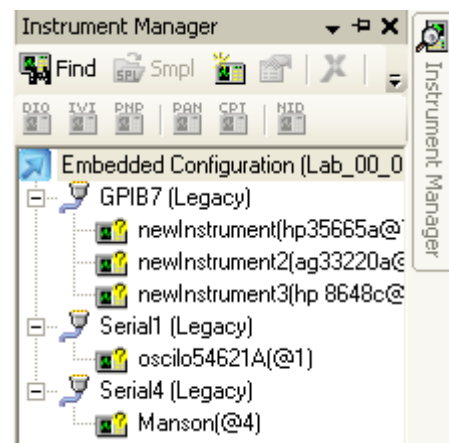
„Virtuální“ vodiče je možné spojovat kliknutím vždy na malý obdélníček na libovolné hraně objektů. Jednotlivé „virtuální“ vodiče mají různé barvy podle jejich významu, díky této vlastnosti je vlastní programování jednodušší a mnohem přehlednější. Smazání „virtuálního“ vodiče je možné po současném stisku kláves CTRL+SHIFT a kliknutím na příslušný vodič.

Postup měření

1) Před zahájením samotné tvorby vlastního projektu, který bude automaticky ovládat požadované měření, je nutné nastavit podmínky samotné komunikace. V této laboratorní úloze probíhá komunikace pomocí rozhraní USB připojeného komunikačního modulu, nicméně se jedná o simulaci virtuální sériové linky RS-232 a jedná se tedy o sériový přenos dat.

Ve specifikaci laboratorního zdroje Manson SDP 2405 je uvedeno následující nastavení komunikace pomocí sériového rozhraní RS-232: datový formát ASCII, parity bit žádný, 8 datových bitů, 1 stop bit a doporučená přenosová rychlost je 9600 baudů.

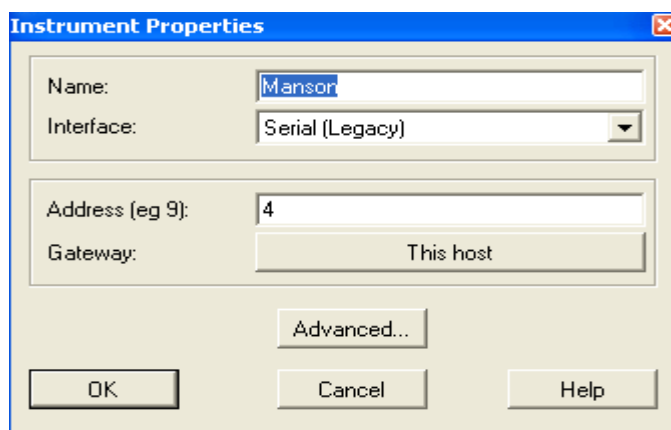
Pro ukončení příkazu je nutné ještě za každý příkaz přidat znak CR (*Carriage Return*), tento znak je v prostředí VEE reprezentován dvojnázkem `\r`. Připojený komunikační modul pracuje s obdobnými parametry komunikace s výjimkou ukončovacího znaku, který požaduje reprezentovat dvojnázkem `\n`. Tyto všechny parametry je nutné před samotným komunikováním s měřicím přístrojem nastavit. Nastavení se provádí v nabídce **I/O-Instrument manager...** (obr. 4). V tomto okně je nutné zvolit zařízení, u kterého chcete nastavit komunikační parametry a pak zvolit položku **Properties...**. Objeví se okno, které je uvedeno na obr. 5. V tomto okně je možné zvolit či modifikovat název zařízení a



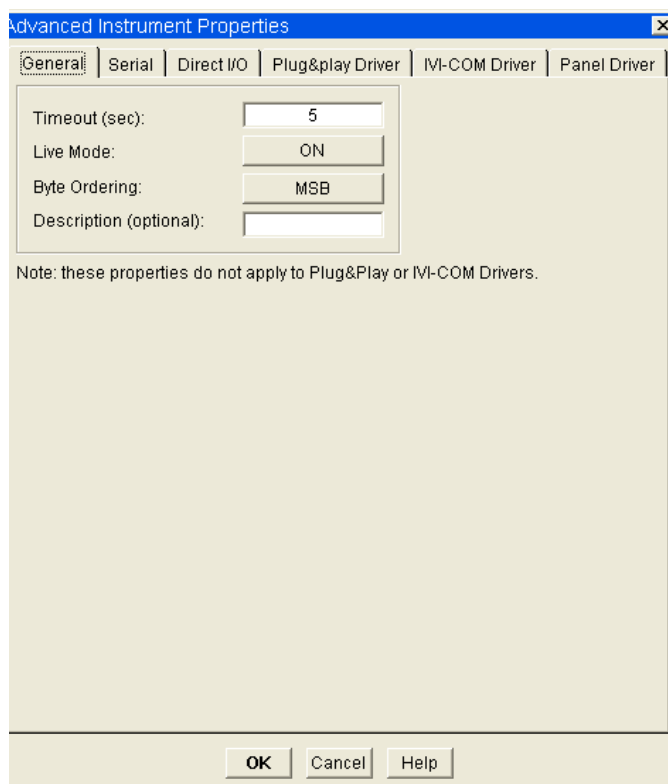
Obr. 4 Instrument manager

komunikační rozhraní, ke kterému je měřicí zařízení připojeno. Připojení je již vybráno a jedná se o typ **Serial (Legacy)**.

Pro další nastavení je nutné se dále ponořit do menu a to pomocí volby **Advanced...**. Po vybrání této volby se objeví okno, které je zobrazeno na obr. 6. Zde je možné v políčku **Timeout (sec)** zvolit časový interval, po který program čeká na odezvu z měřicího přístroje. Standardně je zde nastavena prodleva 5 sekund, lze ji však libovolně měnit dle vlastní potřeby.

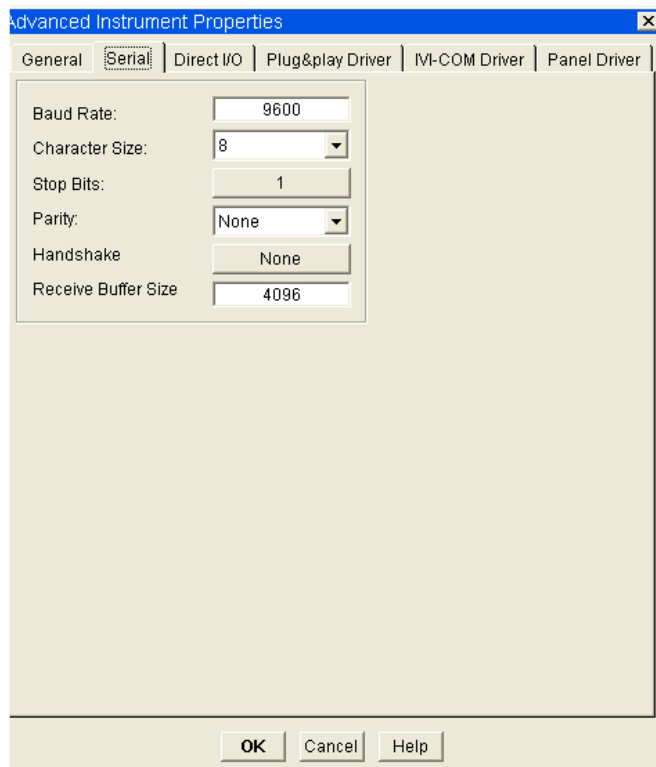


O
Obr. 5 Properties



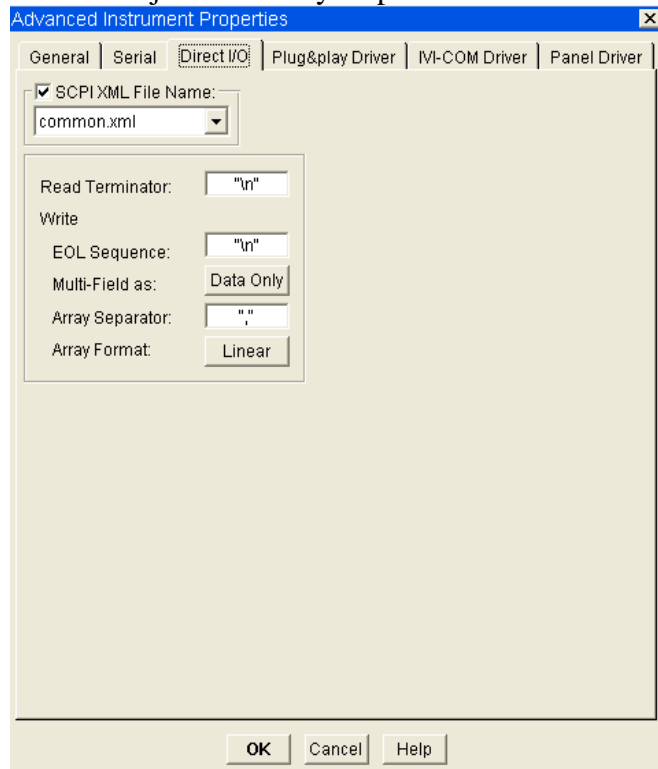
Obr. 6 General properties

Na další záložce označené **Serial** (obr. 7) je možné nakonfigurovat vlastnosti samotné komunikace po sériové lince. **Baud Rate** je přenosová rychlost, jakou se bude komunikovat. Zde je tedy nutné doplnit hodnotu 9600 Baudů (bit/s). Počet datových bitů se volí v položce **Character Size**. A konečně počet stop bitů lze volit ve třetím řádku. V této záložce musí hodnoty odpovídat parametrům komunikace uvedených na začátku této kapitoly. V případě odlišností může komunikace fungovat nestabilně, případně nebude probíhat vůbec.



Obr. 7 Nastavení vlastností komunikace po sériové lince

V další záložce označené **Direct I/O** (obr. 8) je nutné nastavit znak, který je vysílán do laboratorního zdroje a má funkci **EOL** (*End of Line*). Ve specifikaci laboratorního zdroje Manson SDP-2405 je to, jak již bylo zmíněno, znak CR neboli dvojznak \r. Tento znak posílá do laboratorního zdroje komunikační modul. V prostředí VEE je nutné nastavit ukončovací znak jako kombinaci znaků \n. Tyto znaky je nutné doplnit do položek **Read Terminator** a také **Write / EOL Sequence** a dvojznak musí být zapsán do uvozovek.



Obr. 8 Nastavení ukončovacího znaku

2) Při řešení této laboratorní úlohy, bude třeba použít následujících komunikačních příkazů, které lze rozdělit do dvou skupin. V první skupině lze uvést následující příkazy: *rst, output on/off (1/0), volt YY.YV, curr X.XXA. Jedná se o příkazy, které nastavují určité parametry laboratorního zdroje a nemají žádnou odezvu, to znamená, že nepošílají žádná data zpět. Naopak do druhé skupiny lze zařadit následující příkazy: *idn?, meas:volt?, meas:curr?, output?, volt?, curr?. Tyto příkazy mají charakter dotazu na hodnotu nastavenou na laboratorním zdroji.

Před samotným vytvářením jakéhokoliv programu pro automatizovaná měření, je velmi vhodné si ověřit funkce jednotlivých příkazů. Pro ověření první skupiny, která byla definována v předchozím odstavci lze použít část programu uvedenou na obr. 2. Postup vytvoření této části programu je uveden na str. 4. Do objektu **Text** pak zapisujete postupně jednotlivé příkazy. Po stisknutí tlačítka **Start** je nejprve vyčtena komunikační paměť sériové linky, následně je zapsaný příkaz odeslán do laboratorního zdroje. Reakci zdroje lze vizuálně kontrolovat přímo na zdroji a ověřit tak správnost nastavení komunikačního protokolu. Jednotlivé příkazy zahrnuté do první skupiny mají následující význam:

- ***rst** – počáteční nastavení přístroje (navázání komunikace, rozepnutí výstupních svorek, nastavení nejmenší napěťové a proudové pojistky)
- **output on/off (1/0)** – sepnutí respektive rozepnutí výstupních svorek laboratorního zdroje. Po sepnutí výstupních svorek dochází k aktivaci výstupu laboratorního zdroje, z kterého je možné odebírat požadovaný výstupní výkon.
- **volt YY.YV** – nastavení napěťové pojistky v rozmezí 1 až 24 V. Příkaz se skládá ze tří částí: volt – uvozuje nastavení napětí, YY.Y – definuje hodnotu požadované napěťové pojistky, V – zakončuje příkaz jednotkou napětí. Zasláný příkaz tedy musí mít tvar volt 15.5V nebo je možné využít zkrácené verze volt 15V. Je nutné hodnotu ukončit znakem „V“, který reprezentuje jednotku napětí. Pokud je příkaz zadán v jiném formátu dojde k chybě a požadovaný parametr nebude nastaven na laboratorním zdroji.
- **curr X.XXA** – nastavení proudové pojistky v rozmezí 0.01 až 5 A. Pro tento příkaz platí obdobná pravidla jako u předchozího. Příkaz se skládá ze tří částí: curr – uvozuje nastavení proudu, X.XX – definuje hodnotu požadované proudové pojistky, A – příkaz je ukončen jednotkou proudu. Zasláný příkaz tedy musí mít tvar curr 3.55A anebo je možné využít zkrácený tvar curr 3A. Pokud nebude příkaz poslán ve správném tvaru, dojde opět k chybě a požadovaný parametr nebude nastaven na laboratorním zdroji.

Pro ověření funkčnosti druhé skupiny příkazů je třeba použít jinou část programu, jejíž příklad je uveden na obr. 2. V tomto případě je nutné, pro ověření funkčnosti příkazu, také načíst odpověď z laboratorního zdroje. Pro zobrazení odpovědi je vhodné využít např. objekt **Logging AlphaNumeric**, který se nachází v menu **Display**. Do objektu s názvem **Direct I/O** (obr. 2) je nutné ještě přidat další relaci, která zajistí vyčtení dat z laboratorního zdroje.

- ***idn?** – identifikace přístroje. Při identifikaci jsou obdrženy údaje o názvu laboratorního zdroje, typovém označení, výrobním čísle a verzi obslužného softwaru. Pro správné vyčtení je nutné zvolit datový formát string.
- **meas:volt?** – změření aktuální hodnoty výstupního napětí. Pro správné vyčtení hodnoty je potřeba uvést datový formát real64.
- **meas:curr?** – změření aktuální hodnoty výstupního proudu. Pro správné vyčtení je opět vhodné použít datový formát real64.
- **output?** – dotaz na stav výstupních svorek laboratorního zdroje. Pokud obdržíme hodnotu 0, znamená to, že výstupní svorky jsou rozepnuty, pokud je naopak vyčtená

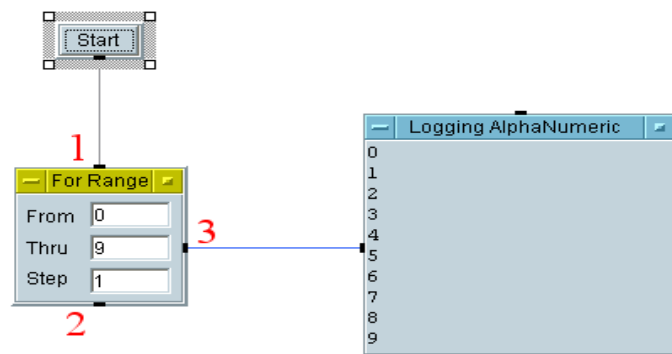
hodnota rovna 1, znamená to, že výstupní svorky jsou sepnuty. Pro správné vyčtení je vhodné použít datový formát int.

- **volt?** – zjištění nastavené hodnoty napěťové pojistky. Pro správné vyčtení je opět vhodné nastavit datový formát real64.
- **curr?** – zjištění nastavené hodnoty proudové pojistky. Pro správné vyčtení je opět vhodné nastavit datový formát real64.

3) Pro vytvoření programu, který je schopen automaticky měřit A-V charakteristiku je třeba ještě vysvětlit funkci některých objektů, které mohou být použity. Jedná se zejména o objekty **X vs Y plot**, **Formula**, **For Range** a **Collector**.

Objekt **X vs Y plot** slouží jako jednoduchý graf pro zobrazování výsledků naměřených dat. Tento objekt se nachází v menu **Display**.

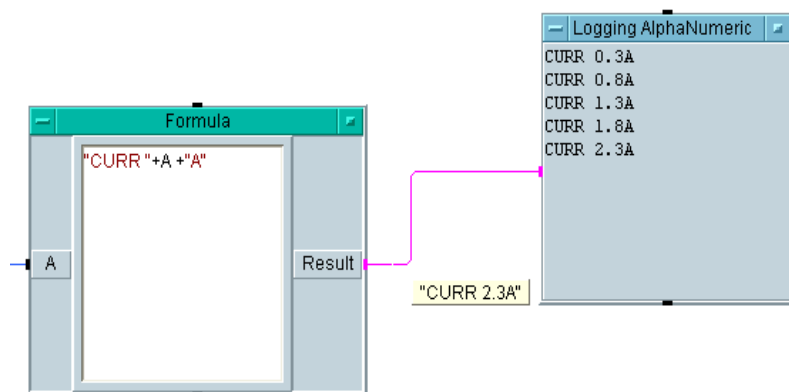
Objekt **For Range**, lze bez nadsázky přirovnat např. k cyklu for z prostředí Matlab® nebo C++. Objekt tohoto cyklu je zobrazen na obr. 9. Samotný objekt lze nalézt v nabídce **Flow/Repeat-For Range**. Tento cyklus má jeden vstup a dva výstupy, jež jsou označeny čísly na obr. 9. Vstup označený číslem 1 je vstupem aktivacím. Po dokončení cyklu je aktivován výstup označený



Obr. 9 Cyklus For Range

číslem 2, tímto výstupem je aktivována další část programu, která je k němu připojena. Při samotném běhu cyklu je vždy aktivován výstup č. 3 a je na něj odesláno číslo ve formátu REAL 64 (obr. 9). Tímto způsobem lze např. provádět opakovaná měření. Před odesláním generovaného čísla do měřicího přístroje je nutné jej ještě obohatit o příkaz samotný. Tuto operaci lze provést např. pomocí objektu **Formula**.

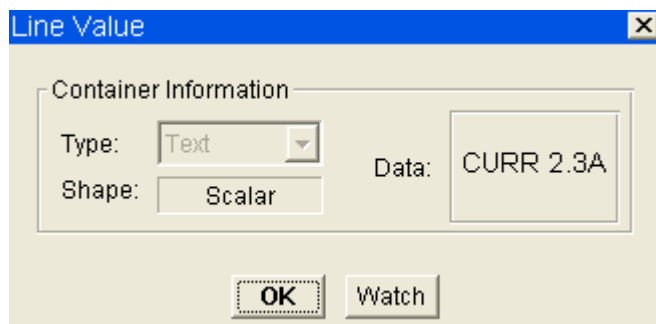
Objekt **Formula** naleznete v menu **Device/Formula**. Tento objekt slouží k různým jednoduchým výpočtům a lze ho také využít například k formátování příkazů, jejichž parametr je něčím automaticky generován. Na obr. 10 je uveden jednoduchý příklad použití objektu **Formula**. Jedná se zde o složení příkazu pro nastavení proudu na



Obr. 10 Objekt Formula

laboratorním zdroji Manson SDP-2405. Z objektu je číslo odesláno ve tvaru **Real64** např. číslo 5.2 do objektu **Formula**. V tomto objektu je k tomuto číslu ještě přidán příkaz, který je následně odeslán do měřicího přístroje. V objektu **Formula** je zapsána následující relace: "CURR "+A+"A". V „uvozovkách“ je vždy zapsán text, pak je k němu pomocí znaků **+A** připočteno číslo ze vstupu **A** a celá relace je dokončena přičtením jednotek ve formě textu. Výsledný generovaný text je pak zobrazen v objektu **Logging AlphaNumeric**. Tento způsob tvoření příkazů se velmi často využívá v různých cyklech apod.

Na obr. 10 je jeden z virtuálních vodičů zobrazen růžovou barvou a je u něj nápis "CURR 2.3A". Tohoto zobrazení lze dosáhnout nastavením ukazatele myši na virtuální vodič, tento vodič se obarví růžově a zobrazí se informace o naposledy přenášeném signálu. Jedná se vždy o poslední informaci, kterou šířil. Pokud je tedy vodič používán opakovaně, vždy se



Obr. 11 Informace o virtuálním vodiči

zobrazí poslední přenášená informace. Pokud však na vodič klikneme, pak se zobrazí okno, které je uvedeno na obr. 11. Zde je o vlastnostech „virtuálního“ vodiče uvedeno ještě více informací.

Pro určení střední hodnoty odporu součástky použijte komponentu **Collector** z nabídky **Data**, do které přiveďte postupně vypočtené hodnoty odporu z naměřených hodnot výstupního napětí a proudu. Tato komponenta je zobrazena na obr. 12. Pomocí této komponenty se vytvoří datové pole (zvolte jednorozměrné pole 1 Dim Array) z načítaných (sesbíraných) hodnot ze vstupu. Pole ukončíte a odešlete na výstup aktivací řídicího vstupu **XEQ**. Pole z výstupu pak zpracujte v objektu **Formula**, kde pomocí funkce **mean(x)** vyhledejte průměrnou hodnotu v poli, kterou vhodně zobrazte.

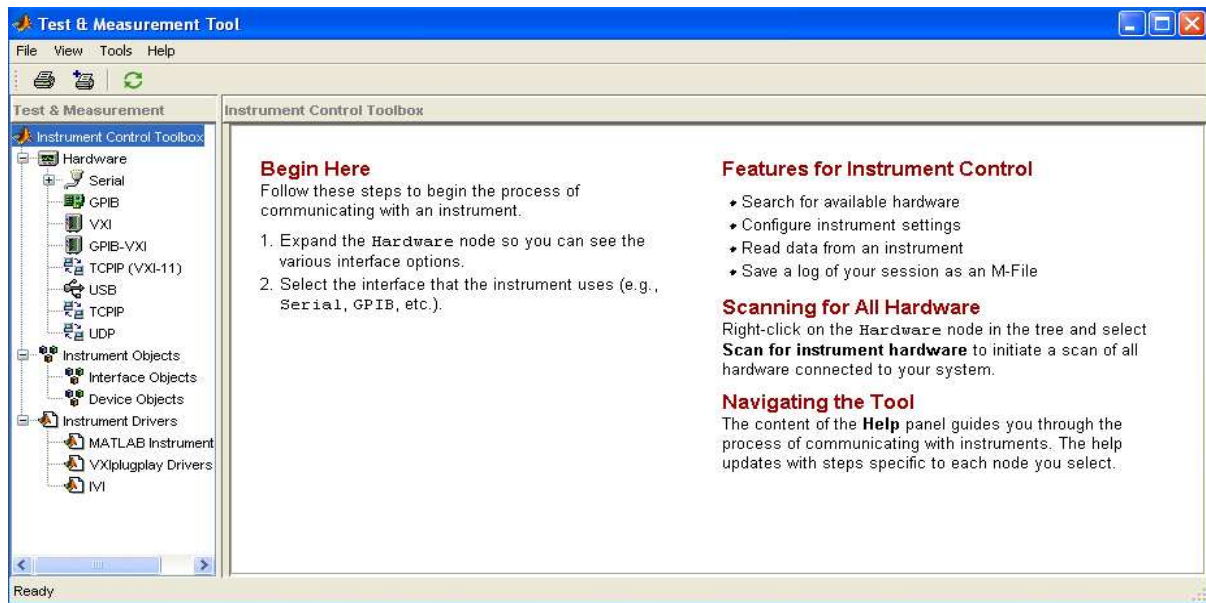


Obr. 12 Objekt Collector

V zadání laboratorní úlohy bylo požadováno vytvoření ovládacího programu pro měření A-V charakteristiky připojené součástky. Požadovaný program může mít následující vývojový diagram:

- 1 vyčtení zbylých dat z vyrovnávací paměti virtuální sériové linky;
- 2 počáteční nastavení a identifikace laboratorního zdroje;
- 3 nastavení počátečního napětí a proudového omezení;
- 4 sepnutí výstupních svorek;
- 5 pomocí cyklu (**For Range**) provést vlastní měření A-V charakteristiky. Toto měření lze provádět buďto změnou napětí nebo proudu (změna nastavení proudu se jeví jako vhodnější, protože zdroj neumožňuje nastavit výstupní napětí menší než 1 V), přičemž druhý parametr musí být dostatečně předem nastaven (celkové proudové nebo napětíové omezení). Následně je nutné provést odečet aktuálních hodnot napětí a proudu, které budou sloužit pro výpočet odporu součástky a zároveň pro zobrazení A-V charakteristiky. Pro určení střední hodnoty odporu je vhodné využít objekt Collector, který bude ukončen po dokončení cyklu. Pro výpočet střední hodnoty odporu je potřeba použít objekt Formula.
- 6 pomocí cyklu (**For Range**) postupně zmenšovat hodnotu výstupního proudu a zároveň určovat matematicky hodnotu výstupního výkonu.
- 7 po skončení měření je vhodné odpojit laboratorní zdroj od zátěže.

4) Pro splnění posledního bodu této úlohy je potřeba si spustit program Matlab. V programovém prostředí Matlab je možné ovládat měřicí přístroje velmi podobným způsobem jako v grafickém prostředí Agilent VEE Pro. Pro otevření komunikace je potřeba nejprve zjistit přesnou adresu měřicího přístroje. Nejjednodušším způsobem je otevřít si již představovanou aplikaci Agilent Connection Expert. V seznamu připojených měřicích přístrojů je nutné vybrat přístroj, s kterým chceme komunikovat. Mezi zobrazenými parametry je položka VISA address. Uvedená adresa měřicího přístroje je potřebná pro navázání komunikace v prostředí Matlab. Měřicí přístroj je tedy potřeba nadefinovat a to následujícím způsobem $Manson = visa('agilent','ASRL4::INSTR');$. Pokud je přístroj nadefinován je možné jeho parametry zjistit v aplikaci Test&Measurement Tool. Pro vyvolání aplikace je nutné do okna Command Window napsat příkaz *tmtool*. Na obr. 13 je uvedeno okno, které se zobrazí po vyvolání aplikace.



Obr. 13 Test&Measurement Tool

Pokud je přístroj správně nadefinován je potřeba otevřít komunikaci pomocí příkazu *fopen(Manson)*. Po úspěšném odeslání tohoto příkazu je programové prostředí Matlab připraveno komunikovat s měřicím přístrojem. Odeslat příkaz do laboratorního zdroje je možné udělat pomocí příkazu *fprintf*. Pokud chceme například odeslat příkaz pro nastavení počátečních parametrů, bude příkaz vypadat následovně: *fprintf(Manson, '*rst');*. Pokud je příkaz dotazovacího typu například příkaz pro identifikaci laboratorního zdroje **idn?*, je nutné po odeslání příkazu použít příkaz pro přečtení přijaté hodnoty. Příkaz bude vypadat následovně *idn = fscanf(Manson)*. Po odeslání tohoto příkazu se objeví v okně Command Window položka „idn =“, do které budou doplněna přijatá data z laboratorního zdroje Manson. Syntaxe běžně používaných příkazů v prostředí Matlab je zachována. Pro vytvoření opakovaného měřicího cyklu je možné použít cyklus *for*. Na závěr měření je nutné komunikaci s laboratorním zdrojem ukončit pomocí příkazu *fclose(Manson)*.

Literatura

- [1] Agilent Technologies, *VEE Pro Help (součást programového prostředí VEE 8.)*.
- [2] Manson, *Laboratory Grade, Remote Programming Switching Mode DC regulated Power Supplies, SDP Series*. 2005, 7673-2405-0007.
- [3] The MathWorks, *Matlab and Simulink for Technical Computing Help (součást programového vybavení Matlab 7.1)*