

VŠB – Technická univerzita Ostrava
Fakulta strojní
Katedra automatizační techniky a řízení

Nasazení jednočipových počítačů pro sběr dat a
řízení

Deployment of Single-chip Computers for Data
Collection and Control

Student: Bc. Jiří Czebe
Vedoucí diplomové práce: Ing. Jaromír Škuta, Ph.D.

Ostrava 2015

Zadání diplomové práce

Student: **Bc. Jiří Czebe**

Studijní program: N2301 Strojní inženýrství

Studijní obor: 3902T004 Automatické řízení a inženýrská informatika

Téma: **Nasazení jednočipových počítačů pro sběr dat a řízení**
Deployment of Single-chip Computers for Data Collection and Control

Zásady pro vypracování:

1. Seznamte se s vybranými laboratorními úlohami dostupnými v laboratořích katedry a zvažte možnost nasazení jednočipových počítačů na místě řízení a monitorování.
2. Vyberte vhodný typ jednočipového počítače a seznamte se s prostředím pro jeho programování a s možnostmi jeho komunikace s nadřazenou úlohou.
3. Navrhněte a realizujte systém na bázi jednočipových počítačů realizující řízení a monitorování laboratorních úloh existujících na katedře. Pro monitorování využijte systém Control Web.
4. Zhodnoťte dosažené výsledky a navrhněte směr dalšího řešení.

Seznam doporučené odborné literatury:


- BALÁTĚ, J. *Automatické řízení*. Praha: Nakladatelství BEN, 2003, 654 s. ISBN 80-7300-020-2.
- BOYER, S. A. 1999. *SCADA: Supervisory control and data acquisition, 2nd edition*. NEW YORK (USA): ISA, 1999. 215 P. ISBN 1-55617-660-0.
- Diplomové práce realizované na katedře 352 v letech 2009 – 2014.
- JANEČEK, J. 1993. *Distribuované systémy*. Praha : Vydavatelství ČVUT, 1993.
- VACEK, V. *Učebnice programování PIC*. 1. vyd. Praha: BEN - technická literatura, 2000, 143 s. ISBN 80-860-5687-2.
- VLACH, J. *Počítačová rozhraní, přenos dat a řídicí systémy*. Praha, BEN-technická literatura, 1997, ISBN 80-85940-17-4.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Jaromír Škuta, Ph.D.**

Datum zadání: 13.12.2014

Datum odevzdání: 18.05.2015



doc. Ing. Renata Wagnerová, Ph.D.
vedoucí katedry



doc. Ing. Ivo Hlavatý, Ph.D.
děkan fakulty

Místopřísežné prohlášení studenta

Prohlašuji, že jsem celou diplomovou práci včetně příloh vypracoval samostatně pod vedením vedoucího diplomové práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě

.....

podpis studenta

Prohlašuji, že

- jsem byl seznámen s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo.
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen „VŠB-TUO“) má právo nevýdělečně ke své vnitřní potřebě diplomovou práci užít (§ 35 odst. 3).
- souhlasím s tím, že diplomová práce bude v elektronické podobě uložena v Ústřední knihovně VŠB-TUO k nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že údaje o kvalifikační práci budou zveřejněny v informačním systému VŠB-TUO.
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona.
- bylo sjednáno, že užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
- beru na vědomí, že odevzdáním své práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby.

V Ostravě

.....
podpis studenta

Jméno a příjmení autora práce: Jiří Czebe

Adresa trvalého pobytu autora práce: Jiráskova 30

58601 Jihlava

Anotace

Czebe, J. Nasazení jednočipových počítačů pro sběr dat a řízení : diplomová práce. Ostrava : VŠB – Technická univerzita Ostrava, Fakulta strojní, Katedra automatizační techniky a řízení, 2015, 72 s. Vedoucí práce: Škuta, J.

Diplomová práce se zabývá využití jednočipových počítačů pro sběr dat a řízení. Cílem je navrhnout obecný řídicí systém, který se bude snadno používat a rozšiřovat dle potřeby na další laboratorní úlohy. Výkonnost celého řídicího systému je limitovaná použitým MCU (PIC 16F873A) a jeho periferiemi (2xPWM, 5xA/D,...). Hlavním úkolem MCU je komunikovat s vyšší úrovní a implementovat požadované akce na řízenou soustavu, společně se sběrem dat. Komunikace mezi jednotlivými MCU (Slave) a PC (Master) je zajištěna sériovým rozhraním (RS232, RS422, RS485). Samotná úloha se skládá z obecné jednotky MCU a rozšiřující desky. Inicializace MCU proběhne dle potřeby (zvolené laboratorní úlohy) z řídicího systému (pro opětovnou inicializaci je třeba provést reset MCU). Vše je řízeno a monitorováno z SCADA/HMI systému napsaném v programu Control Web. Aplikace je dostupná přes webový prohlížeč, či v lokální síti laboratoře. Pro vybrané úlohy, je implementováno řízení z vyšší (PID či 2 polohová regulace), či z nižší úrovně (algoritmus PID v MCU). Je možné provést identifikaci vybraného systému (přechodová, frekvenční, statická charakteristika) a veškerá naměřená data jsou automaticky uložena v adresáři aplikace. Pro lepší orientaci v systému je pro studenta vytvořena nápověda k jednotlivým částím programu, spolu s odkazem na katedrální stránky s odpovídajícím obsahem teorie k dané problematice.

Klíčová slova

zpracování dat, programování v jazyce C, komunikační rozhraní, zpracování signálu, USART, SPI, I2C, USB, PIC, CORTEX, AVR, SCADA/HMI

Abstract

Czebe, J. Deployment of Single-chip Computers for Data Collection and Control : Master Thesis. Ostrava : VŠB – Technical University of Ostrava, Faculty of Mechanical Engineering, Department of Control Systems and Instrumentation, 2015, 72 p. Thesis head: Škuta, J.

The thesis deals with the usage of single-chip computers for the collection of data and control. The aim is to design a general control system, which is easy to use and easy to implement for other laboratory tasks. The performance of control system is limited by the performance of used MCU (PIC 16F873A) and its peripherals (2xPWM, 5xA/D...). The main task of MCU is communication with upper level and implementation of required control action together with data collection. Communication between control system (PC) and MCU is provided by serial interface. In the first step control system defines initialization of MCU (reset needed) for desired laboratory task then continues with the control application. Everything is controlled and monitored from SCADA/HMI system called Control Web. The application is accessible via Internet browser in the local network. Control is implemented from upper level (PID, on-off control) or from a lower level (algorithm of PID is in MCU unit). The student can identify the current system by step response, frequency response or create static characteristic. The application automatically stores every measured data into a local directory of the app. For better understanding of app is also include help with tips (how to use) and references (theory of current problematics).

Key words

data processing, programming in C, communication interface, signal processing, USART, SPI, I2C, USB, PIC, CORTEX, AVR, SCADA/HMI

Obsah

Seznam použitých značek a symbolů	9
Úvod.....	11
1 Laboratorní úlohy	12
1.1 Teplovzdušný model	12
1.2 Stejnoseměrný motor	15
1.3 Měření průhybu nosníku	19
2 Modul jednočipového počítače	22
2.1 Microchip PIC16F873A.....	22
2.2 Univerzální synchronní asynchronní přijímač/vysílač.....	24
2.3 Synchronní sériový port	24
2.4 Pulzně šířková modulace.....	26
2.5 Základní deska modulu MCU	30
3 Běžná komunikační rozhraní v MCU	32
3.1 Sériové komunikační rozhraní	32
3.1.1 Asynchronní mód.....	32
3.1.2 Synchronní mód Master-Slave.....	33
3.2 Synchronní sériový interfejs	36
3.3 I ² C - Vnitřní sériová linka	37
3.4 Univerzální sériová sběrnice USB	39
3.4.1 Vlastnosti USB	39
3.4.2 Fyzická vrstva	40
3.4.3 Způsob přenosu dat	40
4 Návrh řídicího systému (SCADA/HMI).....	41
4.1 Logická struktura systému	41
4.2 Komunikačního rozhraní.....	42
4.2.1 Point-to-Point.....	42

4.2.2	Průmyslová sběrnice RS422/RS485	43
5	Programovací prostředí mikroC.....	46
5.1.1	Rozblikání LED	47
5.1.2	Komunikace po USART	48
5.1.3	Rozsvícení LED pomocí PWM	48
5.1.4	A/D převod analogového signálu	48
6	Realizace algoritmů pro MCU v prostředí mikroC	48
6.1	Hlavní program	49
6.2	Inicializace MCU	49
6.3	Komunikace	50
6.4	Funkce MCU	52
6.5	Makra a sdílené proměnné	54
7	Vývojové prostředí Control Web.....	54
7.1	Textový editor	56
7.2	Datový inspektor	56
7.3	Grafický editor	57
8	Řídicí aplikace v prostředí Control Web	58
9	ZÁVĚR	62
	Seznam použité literatury	65
	Přílohy.....	70
1	Obsah přiloženého CD	70
2	Zlín - STOČ 2015	70
3	Kraków - 52 Konferencja Studenckich Kół Naukowych	70

Seznam použitých značek a symbolů

A/D	–	analogově digitální převodník
AHB	–	Advanced High-Performance Bus
ALU	–	aritmeticko-logická jednotka (arithmetic logic unit)
APB	–	Advanced Peripheral Bus
B	–	báze tranzistoru
BOR	–	reset při poklesu napětí (brown-out reset)
C	–	kolektor tranzistoru
CK	–	hodinový signál (clock)
CPU	–	centrální procesorová jednotka (central processor unit)
DC Motor	–	laboratorní úloha stejnosměrný motor
DMA	–	přímý přístup do paměti (direct memory access)
DSP	–	digitální signálový procesor (digital signal processor)
e	–	regulační odchylka
E	–	emitor tranzistoru
f_{PWM}	–	frekvence PWM signálu [Hz]
GUI	–	grafické uživatelské prostředí (graphic user interface)
CHA	–	kanál A (channel A)
CHB	–	kanál B (channel B)
I2C	–	vnitřní sériová linka (Inter-Integrated Circuit)
k_p	–	zesílení regulátoru [-]
Kb/s	–	přenosový rychlost v kilobytech za sekundu
LED	–	led dioda (Light-Emitting Diode)
LSB	–	nejméně významný bit (least significant bit)
ms	–	milisekunda
MCU	–	Řídící jednotka stroje (microcontroller unit)
MIPS	–	milion instrukcí za sekundu (Million Instruction Per Second)
MSB	–	nejvýznamnější bit (Most Significant Bit)
MSSP	–	hlavní synchronní sériový port (master synchronous serial port)
OS	–	operační systém
OST	–	zapnutí od časovače s oscilátorem (oscillator start-up timer)
PIC	–	obvodová řídicí jednotka (peripheral interface controller)

PID	–	Proportional-Integral-Derivative
POR	–	zapnutí resetem (power-on reset)
PR2	–	perioda čítače/časovače 2 (Period register 2)
PSD	–	Proportional-Sum-Derivative
PWM	–	pulzně šířková modulace (pulse width modulation)
PWRT	–	zapnutí časovačem (power-up timer)
RAM	–	paměť s náhodným přístupem (random access memory)
RTOS	–	system reálného času (real time operation system)
SCADA/HMI	–	supervisory control and data acquisition /human machine interface
SCI	–	sériové komunikační rozhraní (serial communications interface)
SPI	–	Serial peripheral interface
SSI	–	synchronní sériové rozhraní (synchronous serial interface)
SSP	–	synchronní sériový port (synchronous serial port)
t	–	(spojitý) čas [s]
tp	–	monoflop time
tp	–	pause time
T	–	perioda signálu (clock signal period) [s]
TENZO	–	laboratorní úloha tenzometrický můstek
T_D	–	derivační časová konstanta [s]
T_I	–	integrační časová konstanta [s]
TII	–	Texas Instruments Incorporated
T_{PWM}	–	perioda PWM signálu [s]
TMR2	–	registr čítače/časovače 2 (Timer 2 register)
TVM	–	laboratorní úloha teplovzdušný model
TWI	–	vnitřní sériová linka (2-wire Serial Interface)
u	–	akční zásah, vstupní veličina
U	–	napětí [V]
USART	–	Universal synchronous asynchronous receiver transmitter
USB	–	Universal serial bus
w	–	žádaná veličina
y	–	regulovaná veličina, výstupní veličina

Úvod

V dnešní době se pro řízení technologických procesů využívá počítačově řízená výroba. Obecně se jedná o řídicí systém na bázi mikroprocesoru. Součástí tohoto řídicího systému je blok SCADA/HMI a blok přímého řízení technologického procesu.

Přímé řízení technologického procesu dnes zajišťují převážně jednotky PLC. Tato vrstva je z hlediska řídicího systému nejnižší (nejblíže řízenému procesu). Řídicí jednotky mají přímé spojení s řízeným procesem (obvykle pomocí I/O signálů). Vzhledem k nutnosti zajištění maximální možné doby pro zpracování jednotlivých I/O operací, se používají systémy reálného času (RTOS).

SCADA/HMI systém má na starost vizualizaci řízeného systému tj. poskytuje rozhraní mezi člověkem (operátorem) a řídicím systémem. Komunikace mezi vrstvou přímého řízení a SCADA/HMI systémem zajišťuje především průmyslový Ethernet, vzhledem k většímu objemu přenášených dat.

Páteří celého řídicího systému je tedy komunikace mezi jednotlivými vrstvami systému. Užití běžného komunikačního rozhraní není vhodné vzhledem k velkému rušení z okolí či jeho stochastickému přístupu k přenosovému médium. Pro přenos dat (informace) mezi jednotlivými vrstvami řídicího systému je třeba užít průmyslového rozhraní (RS422, RS423, RS485, ...), resp. průmyslových standardů (protokol, rozhraní, zabezpečení přenosu, přístup k přenosovému médium, apod.).

Pro realizaci SCADA/HMI systému lze využít například program Control Web, který umožňuje vytvoření řídicího systému ve dvou základních režimech (událostním, realtime). Pokud operátor potřebuje mít kontrolu nad danou aplikací v čase, nebo čekat a obsloužit vzniklou událost, vyvolanou v systému.

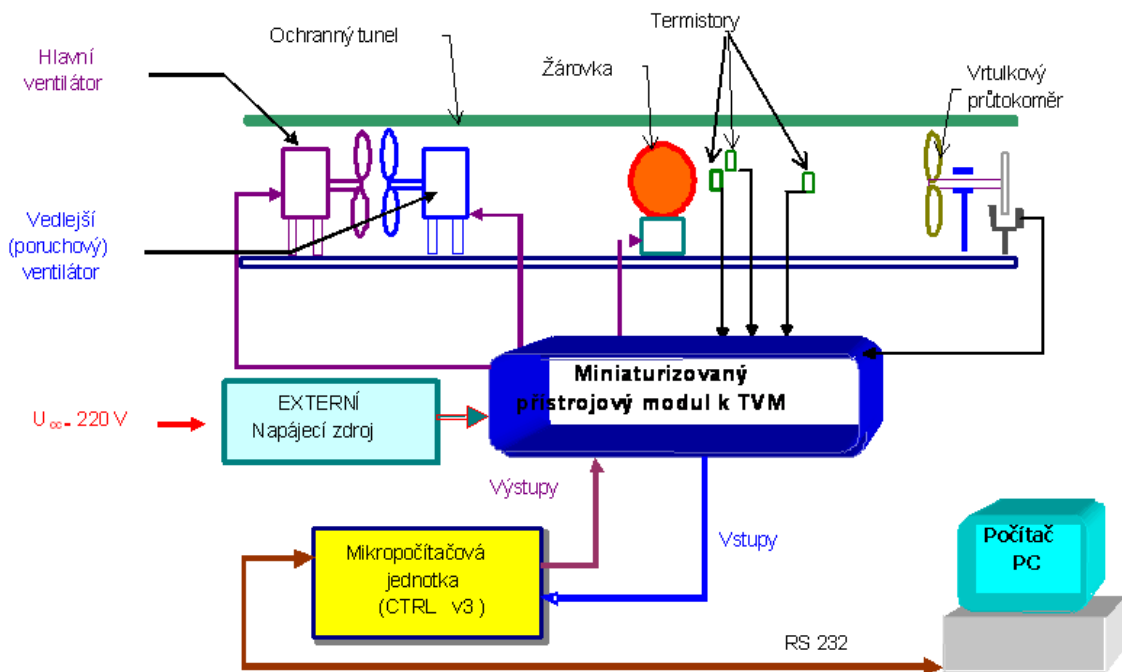
V oblasti přímého řízení se kromě PLC dále používají jednočipové počítače (MCU). Levnější a úplná náhrada, tam kde je možné provést distribuci řízení až k samotnému technologickému procesu. Mezi běžné vybavení MCU patří periferie pro nastavení (PWM, BCD,...), komunikaci (USART, SPI,...), získání dat (A/D, BCD,...), atpod.

1 Laboratorní úlohy

Kapitola obsahuje obecný popis vybraných laboratorních úloh, vhodných pro možné využití jednočipových počítačů v oblasti monitorování či řízení. Vybrané laboratorní úlohy teplovzdušný model (TVM) a stejnosměrný motor (DC Motor) jsou vhodné, jak pro monitorování průběhů veličin, tak i pro samotné řízení viz popis úlohy. Zatímco laboratorní úloha pro měření průhybu nosníku (TENZO), je vhodná pouze v rámci monitorování.

1.1 Teplovzdušný model

Model umožňuje jednorozměrné i vícerozměrné regulační úlohy. Na základě rozhodnutí programátora může být měřená (výstupní) veličina měřena v různých vzdálenostech od zdroje tepla (žárovka) nebo měřit hodnotu průtoku vzduchu v tunelu pomocí vrtulkového průtokoměru.



Obrázek 1 – Blokové schéma TVM s řídicí jednotkou a PC [SMUTNÝ, ŠKUTA, 2005]

Stávající jednotka CTRL v3 je využita pro proudové zesílení, napájení soustavy a úpravu signálu, která dále zesiluje signály z řídicí jednotky MCU. Základní modul MCU bude rozšířen o další rozšiřující desku pro výkonové zesílení signálu na požadovanou úroveň,

či snížení vstupního napětí na úroveň vhodnou pro zpracování řídicí jednotkou dle následující tabulky.

Tabulka 1 – Přehled I/O kanálů CTRL V3 [SMUTNÝ, ŠKUTA, 2005]

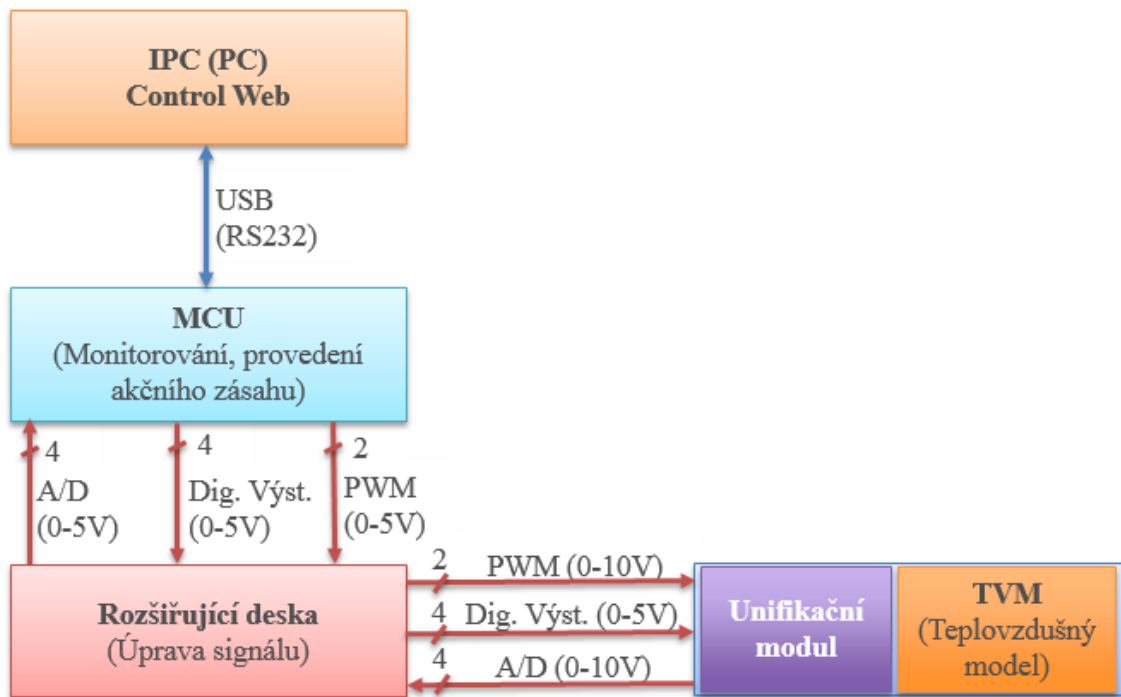
Vstupní kanál	Snímač	Výstupní kanál	Akční člen
Vstup 1	NR 354 20 KU (na baňce)	Výstup 1	Ovládací napětí na žárovce
Vstup 2	NR 354 20 KU (5 mm od baňky)	Výstup 2	Ovládací napětí na hlavním ventilátoru (řízení otáček)
Vstup 3	KTY 82 (5 mm od baňky)	Výstup 3 (BCD)	Ovládací napětí na ventilátoru realizujícím poruchu (řízení otáček)
Vstup 4	Vrtulkový průtokoměr		

U řídicí napětí ventilátoru a žárovky je nutné převést diskretní signál PWM na signál analogový (filtrace pomocí kondenzátoru). Výstupní napětí senzorů je nutné zeslabit na optimální úroveň napětí (pomocí napěťového děliče z 0-10 V na 0-5 V), pro dosažení co největší přesnosti A/D převodu, při použití referenčního (porovnávacího) napětí 5 V. Z rozšiřující desky je třeba vyvést odpovídající počet diskretních výstupu (4x), které definují svou kombinací (BCD) velikost napětí na ventilátoru realizujícím poruchu.

Tabulka 2 – Vstupní/výstupní kanály rozšiřující desky pro TVM

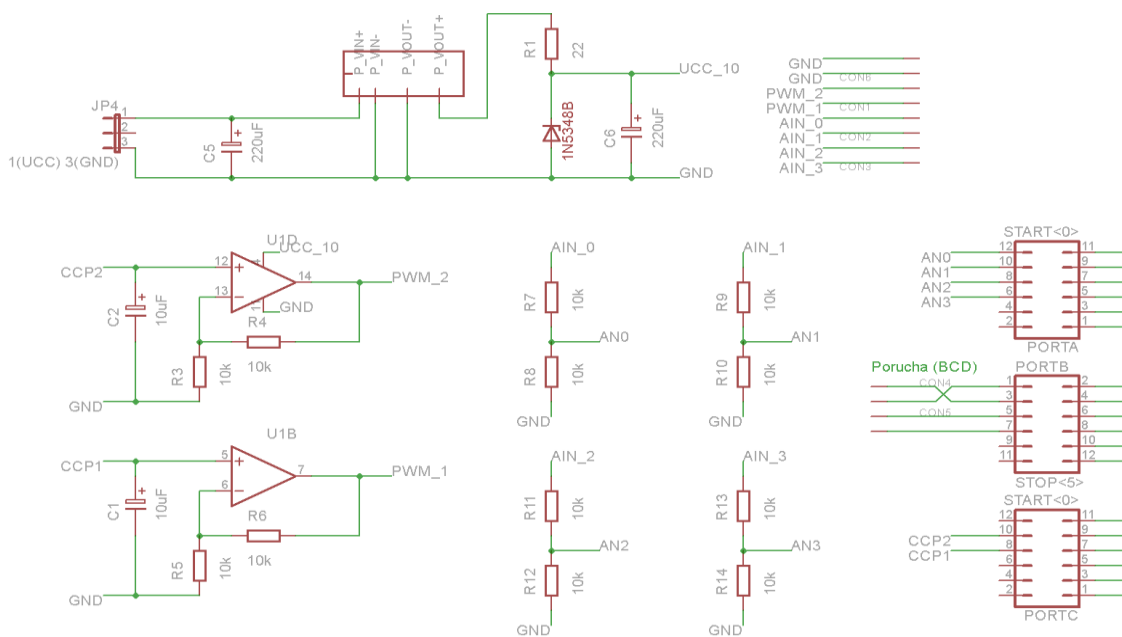
Typ I/O	Rozsah (modul MCU)
2x analogový výstup	0 - 10V (PWM)
4x analogový vstup	0 - 10V (A/D)
4x digitální výstup (BCD)	0 - 5V (I/O)

Blokové schéma takového řídicího systému, připojeného přímo k řídicí stanici (Point-to-Point) bude vypadat následovně.

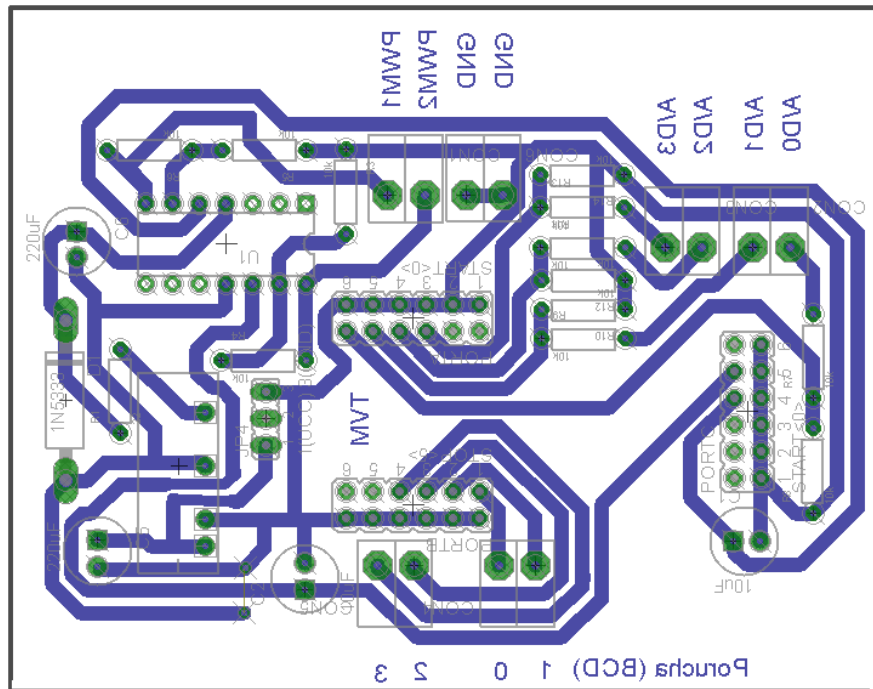


Obrázek 2 – Blokové schéma pro TVM (Point-to-Point)

Rozšiřující deska obsahuje DC/DC měnič z 5V na 12V, pro dosažení požadovaného analogového výstupu z jednotky (napájení OZ stabilizovaným napětím cca 10V). Přivedené řídicí napětí PWM na OZ je usměrněno a 2x násobně zesíleno. Vstupní napětí z TVM je přes napěťové děliče 2x násobně zeslabeno, pro následné zpracování A/D převodníkem.



Obrázek 3 – Rozšiřující deska pro TVM



Obrázek 4 – Rozložení součástek rozšiřující desky pro TVM

Tabulka 3 – Seznam použitých součástek pro rozšiřující desku TVM

Součástka	Katalogový název
DC/DC měnič	TRACO POWER TMH0512S
Port A, B, C	Dutinková lišta BL240G
Port pro napájení	Dutinková lišta BTK25G
Stabilizační dioda	Zenerova dioda 1N5348B
Svorkovnice	PTR AK550/2DS
Rezistor	RU 22R 0207 0,25W 5%
Rezistor 12x	RU 10K 0207 0,25W 5%
OZ	TLC274 DIP14 TEXAS INSTRUMENTS
Kondenzátor 2x	CE 10u/100V JAM-SK 6,3x11 RM2,5
Kondenzátor 2x	CE 220u/16VT JAM-- 6,3x11 RM2,5

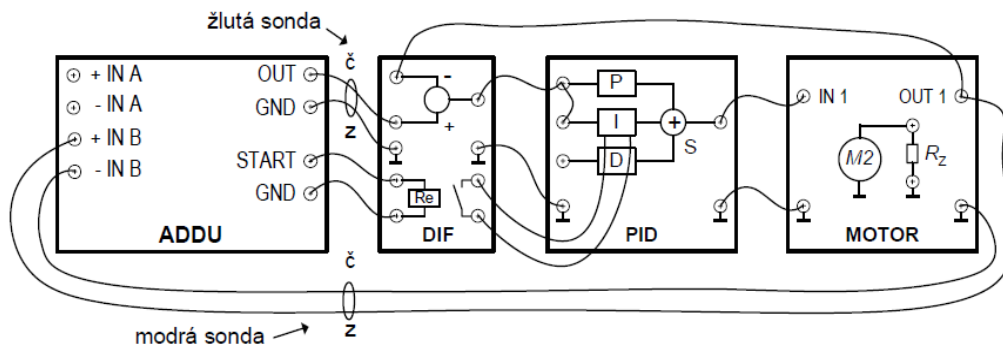
1.2 Stejnosměrný motor

Laboratorní úloha slouží pro identifikaci stejnosměrného motoru, kdy v programu od firmy RC Didactic lze nastavit průběh vstupního napětí a následně sledovat (měřit) průběh výstupního napětí v závislosti na vstupním napětí v čase tj. změřit přechodovou charakteristiku motoru a z ní určit typ i přenos soustavy.



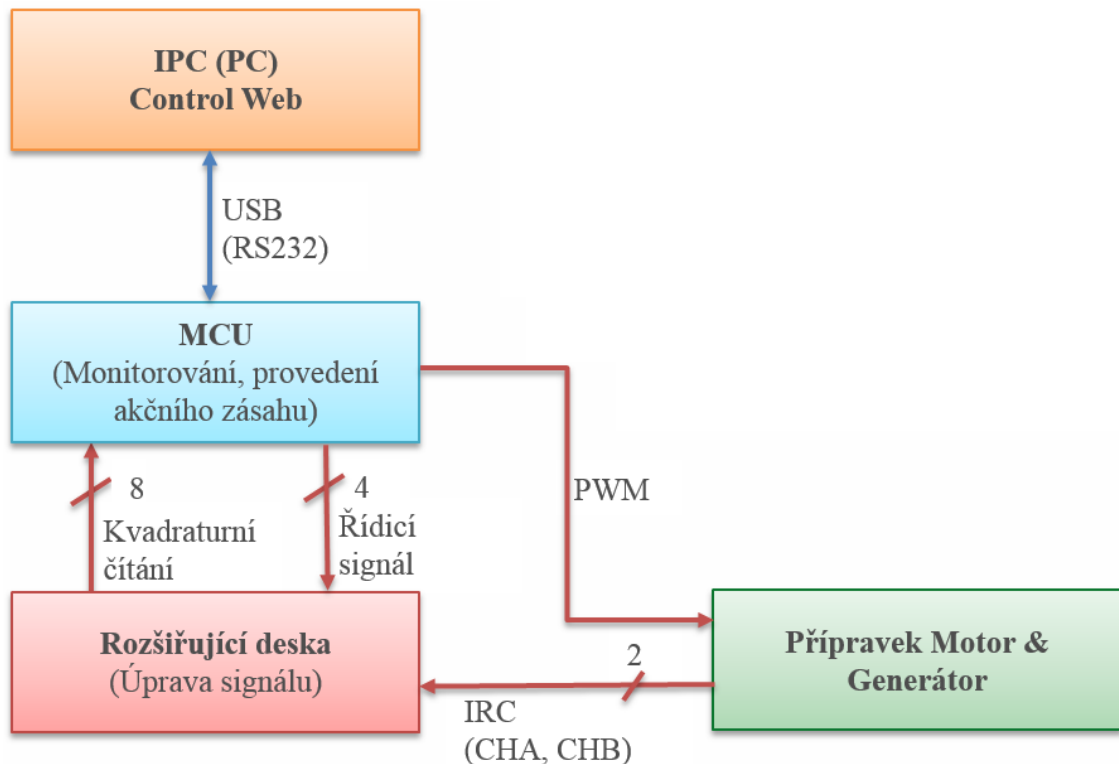
Obrázek 5 – Ukázka průběhu identifikace motoru [Výukový systém rc2000 – uLab]

Podle vypočteného přenosu soustavy zvolit vhodný typ regulátoru a určit jeho časové konstanty. Regulace je zde prováděna pomocí analogového PID regulátoru, který využívá pro zpětnou vazbu tach signál (analogový signál). Celý průběh regulace je možné archivovat.



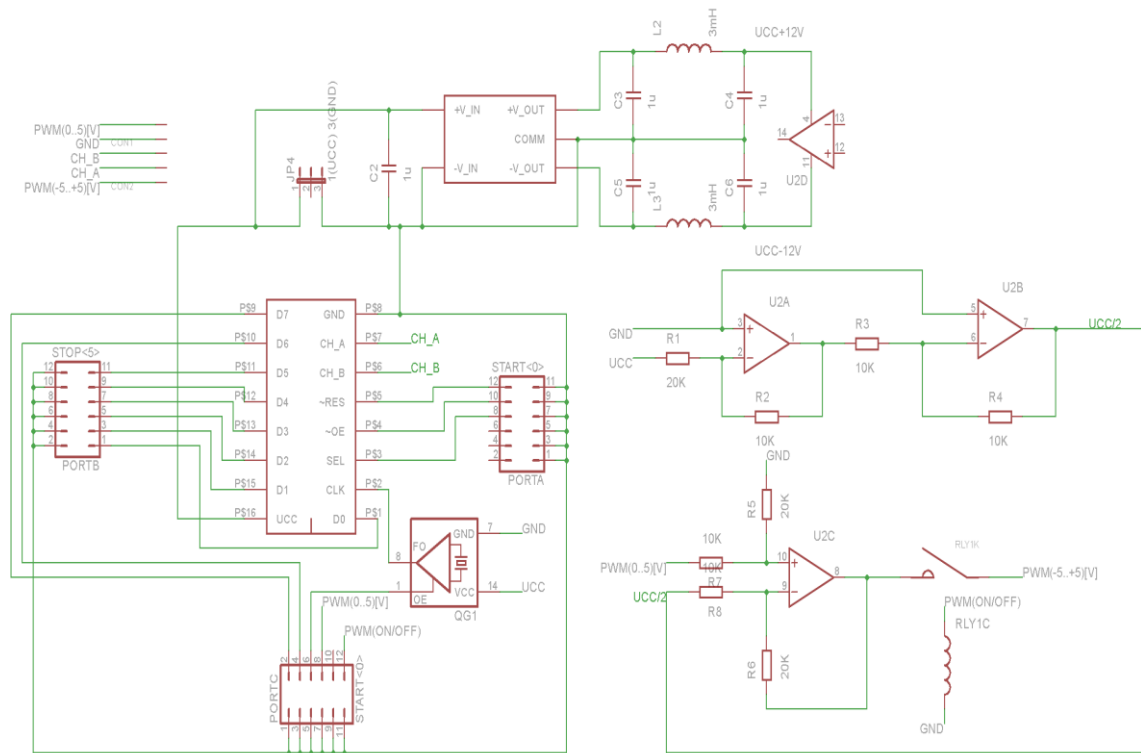
Obrázek 6 – Zapojení motoru s PI regulátorem [Výukový systém rc2000 – uLab]

Laboratorní úloha bude realizovat regulaci otáček stejnosměrného motoru pomocí MCU a nadřazené stanice IPC (PC). Řídicí algoritmus PID bude mít na starost IPC a nebo MCU jednotka. V případě volby regulace v samotném MCU, je možné z IPC nastavovat parametry regulátoru za běhu (žádaná hodnota, k_p , T_I , T_D , vypnutí/zapnutí regulátoru). Při řízení z IPC je MCU použito pouze, jako zprostředkovatel požadovaných dat (vyhodnocení relativní polohy motoru z enkodéru) a provedení žádaného akčního zásahu. IPC se stará o logování dat a výpočet akčního zásahu dle přijatých dat z MCU.

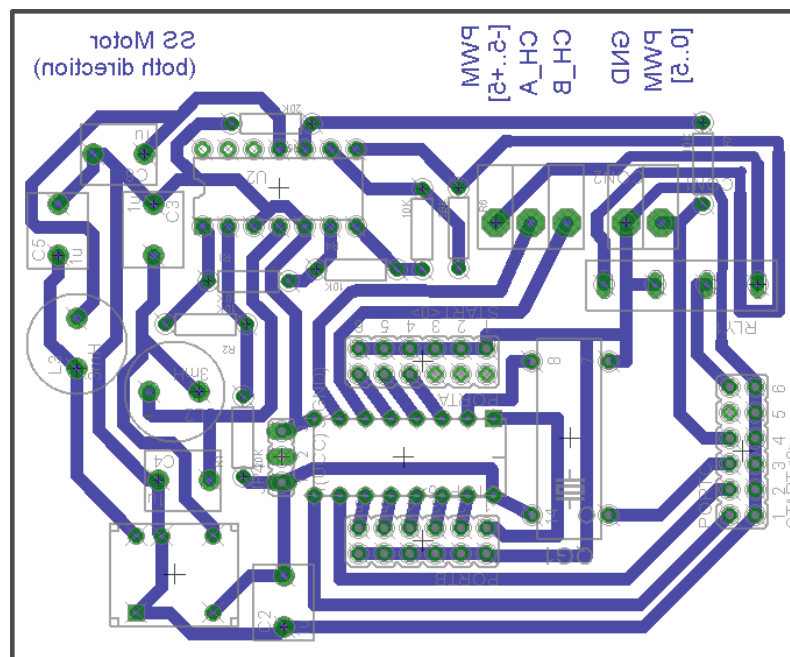


Obrázek 7 – Blokové schéma rozšiřující desky pro DC Motor

Rozšiřující deska obsahuje DC/DC měnič z 5 V na ± 12 V a jeho stabilizaci pro napájení OZ, který se stará o obousměrný chod na základě sečení vygenerovaného PWM napětí a zesíleného napájecího napětí. Výstup z PWM je tedy ve 2 možných variantách a to pro řízení stejnosměrného motoru v jednom směru v plném rozsahu PWM (0-1023), či v obou směrech (reverzní chod 0-512, dopředný chod 512-1023 hodnoty při $f_{PWM} = 5$ kHz). Vypnutí či zapnutí alternativního výstupu PWM je možné pomocí relé. Čítání impulsů z diskretního výstupu IRC má na starosti jednotka HCTL-2000, která vzájemně posunuté signály CHA a CHB vyhodnotí a do MCU, dle požadavku zasílá výsledky v daném pořadí (horní byte, dolní byte). Na desce je dále umístěn oscilátor s výstupní frekvencí 12 MHz, jakožto zdroj hodinového signálu CLK pro obvod HCTL-2000.



Obrázek 8 – Rozšiřující deska pro DC Motor



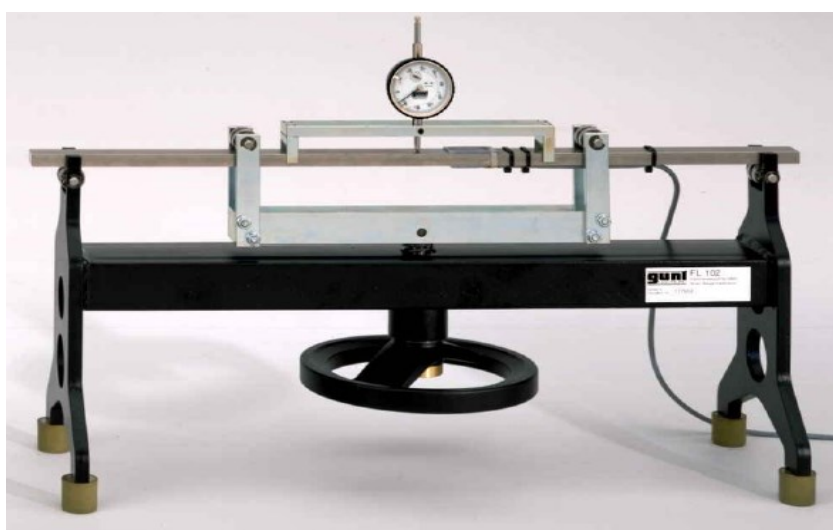
Obrázek 9 – Rozložení součástek rozšiřující desky pro DC Motor

Tabulka 4 – Seznam použitých součástek pro rozšiřující desku DC Motor

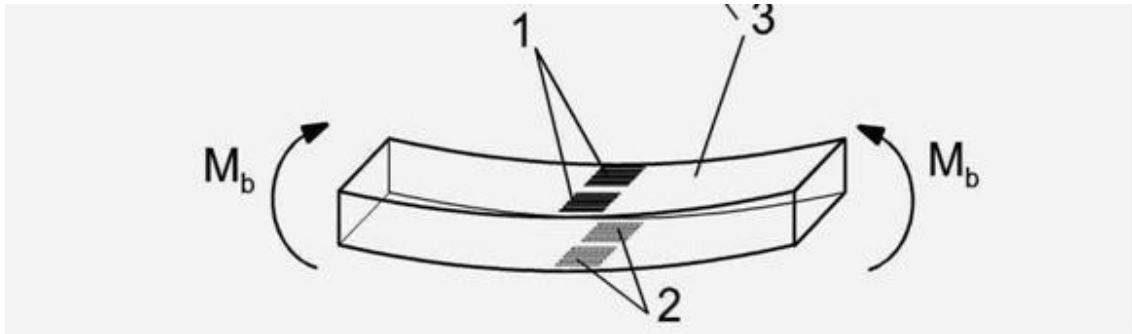
Součástka	Katalogový název
Kvadrurní čítač	HCTL – 2000 DIP16
DC/DC měnič	SIM1 – 512D
Port A, B, C	Dutinková lišta BL240G
Port pro napájení	Dutinková lišta BTK25G
Jazyčková relé	COSMO RELES1A 050 000
Oscilátor	SG51 – 12.000
Svorkovnice 2x	PTR AK550/2DS
Rezistor 5x	RU 10K 0207 0,25W 5%
Rezistor 3x	RM 20k 0207 0,6W 1%
OZ	TL084 DIP14 TEXAS INSTRUMENTS
Kondenzátor 5x	Fóliový kondenzátor CF1 – 1M0/J
Cívka 2x	Radiální tlumivka 09P – 332J

1.3 Měření průhybu nosníku

Laboratorní úloha pro měření deformace nosníku pomocí tenzometrů. Tenzometry jsou nalepeny na nosník z obou stran (kompresní, tahová), jejich zapojení do můstku brání jeho rozladění vlivem změny teploty. Měření napětí (při rozvážení můstku) probíhá pomocí přímého odečtení hodnoty z V-metru, či možným importem naměřených hodnot z V-metru do PC.

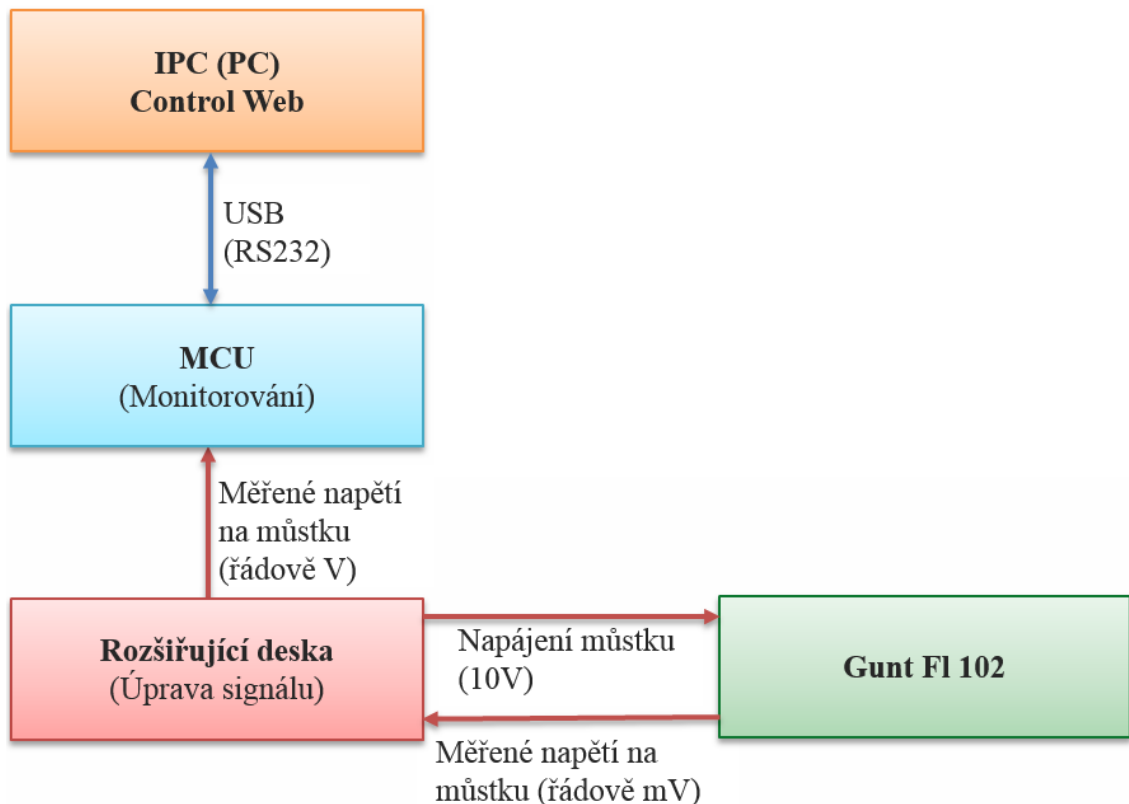


Obrázek 10 – Gunt FL 102 [FL 102, 2015]



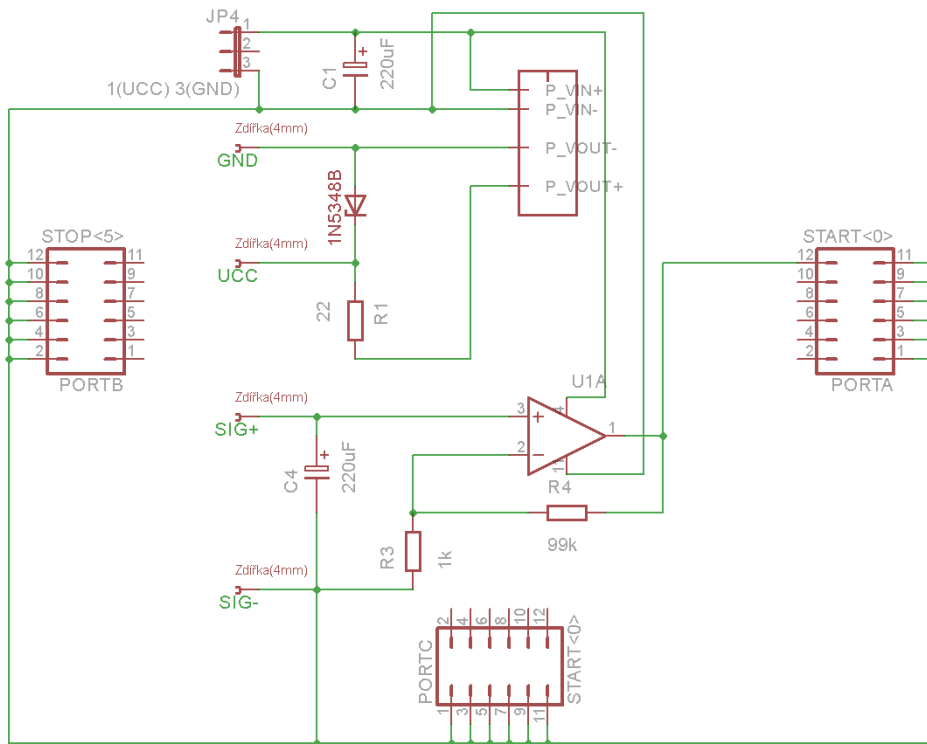
Obrázek 11 – Zapojení tenzometrů (průhyb nosníku) [FL 102, 2015]

V úloze je možné nahradit měřící V-metr, jehož činnost (monitorování měřené veličiny) převezme MCU. Vzhledem k vyššímu napájecímu napětí tenzometrů (cca 10 V) a velmi nízkému napětí na můstku tj. řádově desítky mV v závislosti na deformaci nosníku, je třeba doplnit základní modul MCU o rozšiřující desku, pro nutnou úpravu měřeného signálu (zesílení a napájení laboratorní úlohy stabilizovaným napětím).

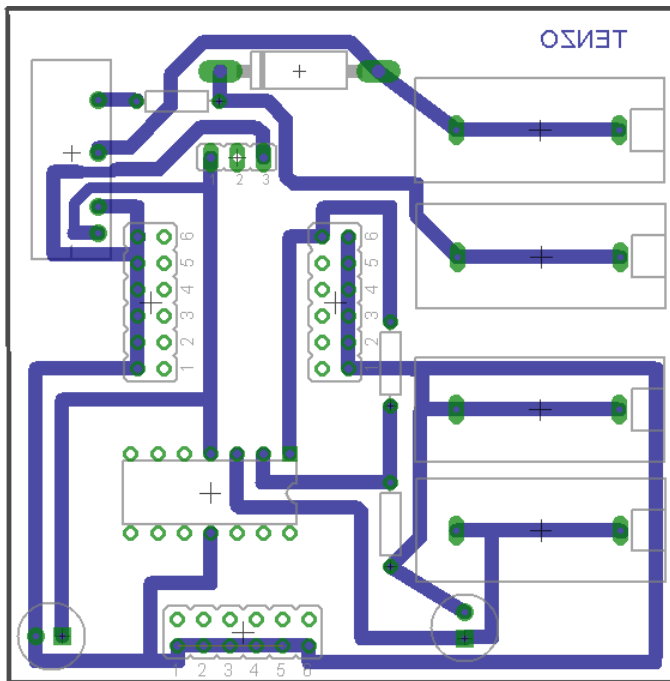


Obrázek 12 – Blokové schéma rozšiřující desky pro TENZO

Deska je osazena DC/DC měničem z 5V na 12V, jehož výsledné napětí je usměrněno pomocí Zenerovy diody. Výsledné napětí z můstku (řádově desítky mV) je zesíleno OZ, do vhodného rozsahu (citlivosti) $0 \div 4,7$ V pro A/D převodník v MCU.



Obrázek 13 – Rozšiřující deska pro TENZO



Obrázek 14 – Rozložení součástek rozšiřující desky pro TENZO

Tabulka 5 – Seznam použitých součástek pro rozšiřující desku TENZO

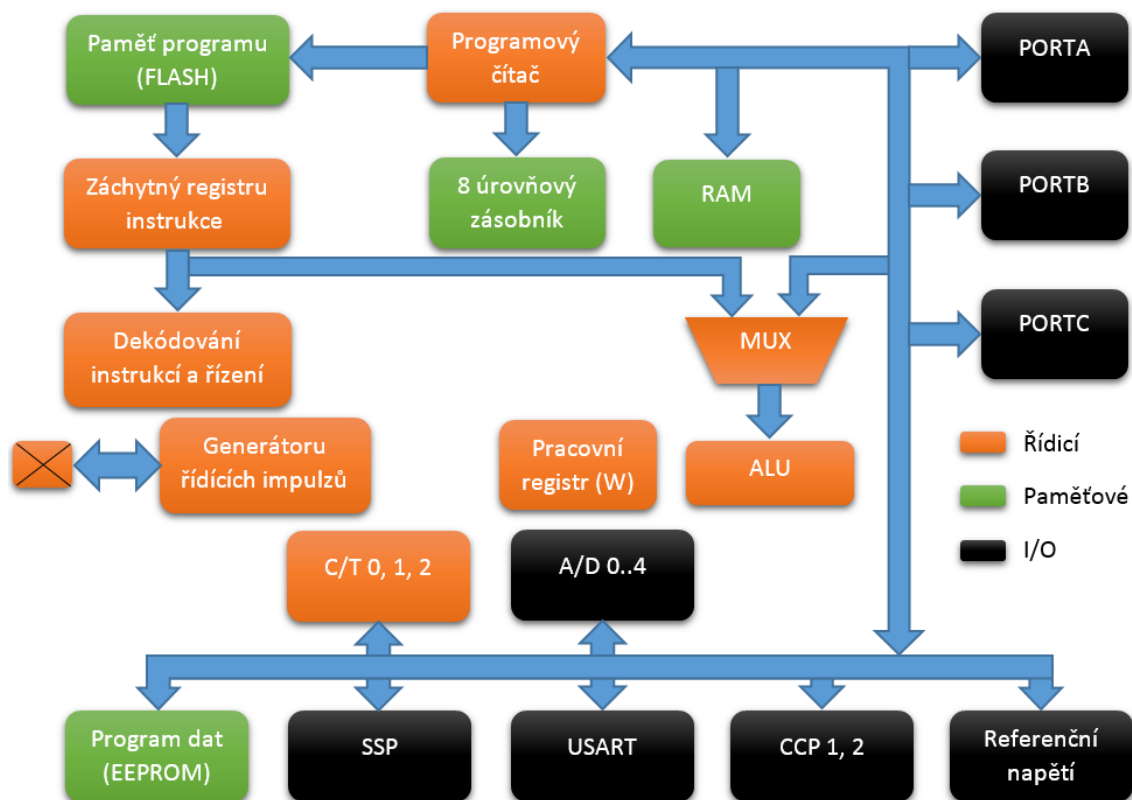
Součástka	Katalogový název
DC/DC měnič	TRACO POWER TMH0512S
Port A, B, C	Dutinková lišta BL240G
Port pro napájení	Dutinková lišta BTK25G
Stabilizační dioda	Zenerova dioda 1N5348B
Rezistor	RU 22R 0207 0,25W 5%
Rezistor	RM 1K 0207 0,6W 1%
Rezistor	RM 100K 0207 0,6W 1%
OZ	TLC274 DIP14 TEXAS INSTRUMENTS
Kondenzátor 2x	CE 220u/16VT JAM-- 6,3x11 RM2,5
Přístrojová zdířka	4mm, černá
Přístrojová zdířka	4mm, červená

2 Modul jednočipového počítače

Obecný modul pro vybrané laboratorní úlohy obsahuje MCU značky PIC, konkrétně typ 16F873A, který svou vybaveností (I/O, komunikačním rozhraním,...) a rychlostí je postačující pro jejich realizaci. Popis samotného MCU a jeho vybraných částí je součástí této kapitoly, včetně návrh základní desky v programu Eagle.

2.1 Microchip PIC16F873A

Výkonná výpočetní jednotka (zpracování jedné instrukce 200 ns) postavená na 8 bitové architektuře založené na flešových paměti typu CMOS, která obsahuje redukovanou sadu instrukcí procesoru (RISC, 35 instrukcí). Architektura zakomponovaná do 28 pinového pouzdra a se zpětně kompatibilní uspořádáním pinů pro vybrané procesory předchozí řady. MCU nabízí dostatečný počet vstupně/výstupních pinů, komunikačních rozhraní, čítačů, CCP modulů, dostatečnou velikost paměti programu a pro data, apod.



Obrázek 15 – Architektura PIC16F873A [MICROCHIP, 2013]

PIC16F873A má 128 bytovou paměť typu EEPROM, vlastní možnost programování LCD displeje, dva komparátory, pěti kanálový 10 bitový A/D převodník, synchronní sériový port, který může být nakonfigurován, jako 3 žilové sériové rozhraní (SPI) nebo 2 žilová vnitřní sériová linka I²C a v neposlední řadě obsahuje univerzální asynchronní vysílač/přijímač (USART). Všechny tyto funkce jej činí ideální pro pokročilou úroveň nasazení, jako A/D převodník, v automobilovém průmyslu, spotřebičích, průmyslu a spotřební elektronice. Vzhledem k využití MCU k řešení daných úloh je popis rozšířen o další jeho periferie.

Tabulka 6 – Parametry procesoru PIC16F873A [MICROCHIP, 2013]

Frekvence procesoru	DC - 20 MHz
Reset (a zpoždění)	POR, BOR (PWRT, OST)
Paměť programu (14 bit. slovo)	4K
Paměť dat	192
Paměť dat (EEPROM)	128
Přerušeni	13
V/V porty	Porty A,B,C
Čítače/Časovače	3
Záznam/Komparátor/PWM modul	2
Sériová komunikace	MSSP, USART
Paralelní komunikace	—
10 bit. A/D modul	5 vstupních kanálů
Instrukční set	35 instrukcí

2.2 Univerzální synchronní asynchronní přijímač/vysílač

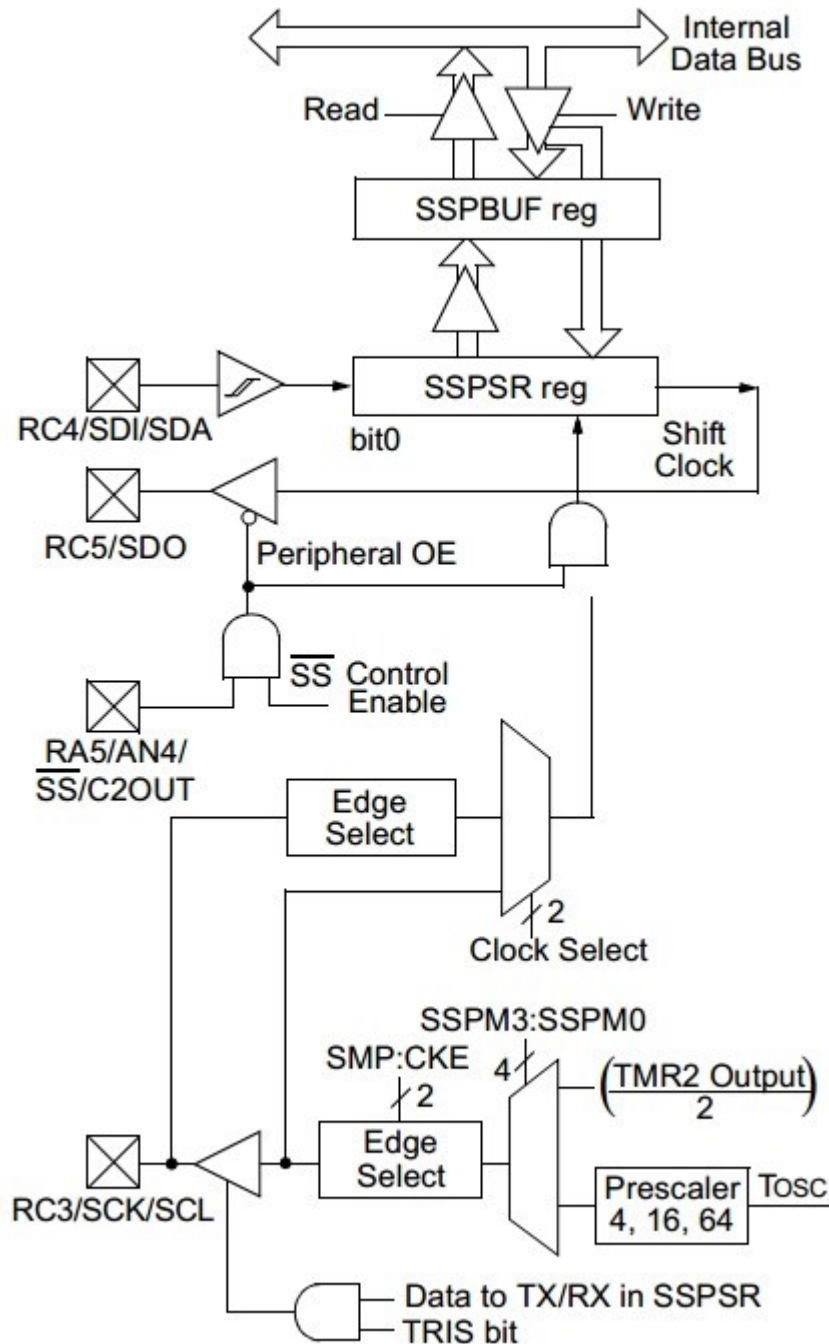
Jeden ze dvou vstupně/výstupních sériových modulů MCU (USART též znám pod názvem SCI). Může být nastaven jako plně duplexní asynchronní systém, který může komunikovat s periferií jakou je CRT terminál, PC nebo může být nastaven, jako synchronní systém s polovičním duplexem pro komunikaci s A/D, D/A integrovaným obvodem, sériovou pamětí typu EEPROM atd. USART je možné nastavit ve 3 režimech: asynchronní (plný duplex), synchronní-Master (poloviční duplex), synchronní-Slave (poloviční duplex). Pomocí bitu SPEN (7 bit v registru RCSTA) a 6,7 bitu v registru TRISC určit správný směr pinu, dále je třeba nakonfigurovat přenosovou rychlost, apod. pro použití pinu 6,7 na portu C ve funkci USART. Modul má možnost komunikovat s více procesory při použití 9 bitové adresní detekce.

2.3 Synchronní sériový port

Modul sériového synchronního portu se používá pro komunikaci s ostatními periferiemi nebo mikrokontroléry. Těmito periferiemi mohou být sériové paměti EEPROM, posuvné registry, zobrazovače, A/D převodník, atd.

Sériové rozhraní

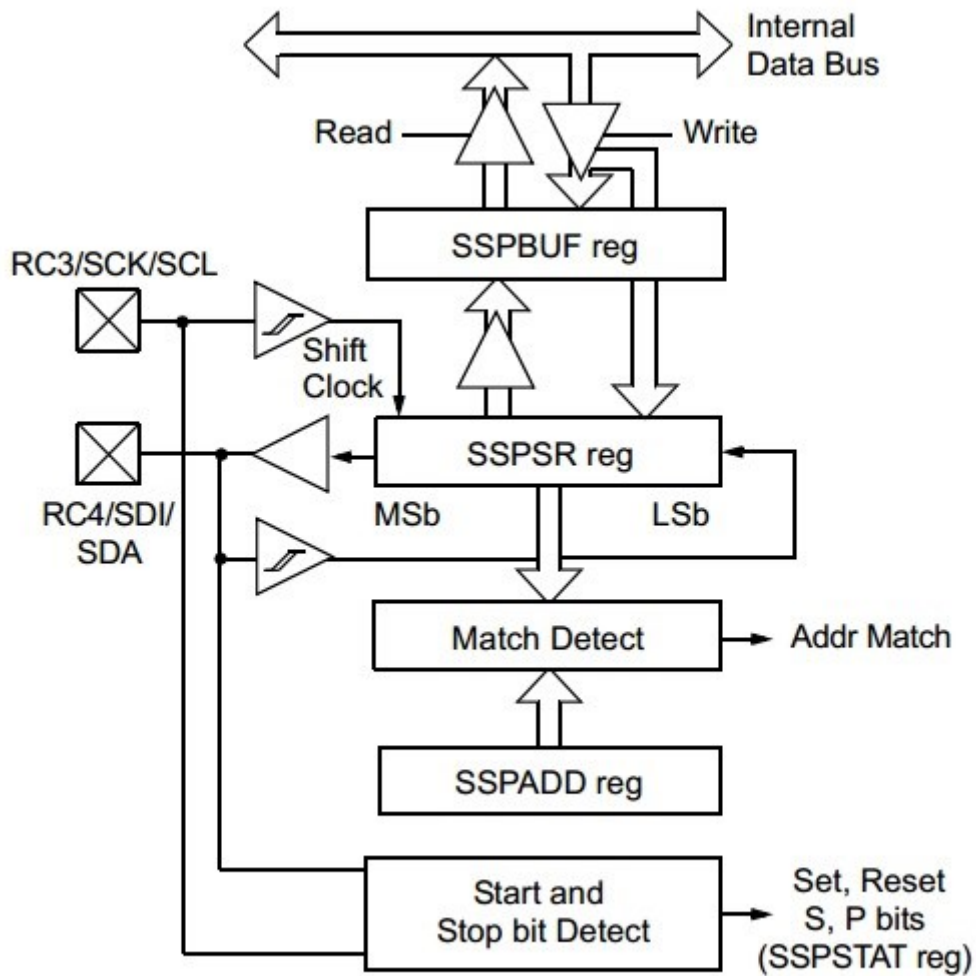
SPI mód umožňuje synchronní přenos 8 bitových rámců oběma směry (full duplex). Podporuje všechny čtyři módy SPI. Pro uskutečnění komunikace je typicky použito tři žilové zapojení (SDO – RC5, SDI – RC4, SCK – RC3). Pro využití Slave módu (4 žilové komunikace), je užít pin SS – RA5.



Obrázek 16 – Blokové schéma SPI [MICROCHIP, 2013]

Vnitřní sériová linka

MSSP modul v módu I²C plně implementuje všechny funkce Master a Slave. Poskytuje hardwarové přerušení na Start a Stop bit, pro detekci volné sběrnice (Multi-Master). Modul implementuje standardní specifikaci módu, stejně jako 7 bitovou a 10 bitovou adresaci. Pro komunikaci jsou použity 2 piny (SCL – RC3, SDA – RC4). Je třeba nastavit tyto piny jako vstupní nebo výstupní přes registr TRISC<4:3>.



Obrázek 17 – Blokové schéma I²C [MICROCHIP, 2013]

2.4 Pulzně šířková modulace

PWM je modulační technika užívaná v komunikačních systémech pro zašifrování amplitudy signálu do šířky (trvání) pulsu jiného signálu. Převážně je tato technika užívaná pro řízení zátěže (motor, žárovka,...).

Obecně o modulu CCP:

MCU je vybaven tzv. CCP modulem (Capture/Compare/PWM) obsahujícím 16 bitový registr, který může operovat v jednom ze tří módů.

- 16 bitový záchytný registr (Capture).
- 16 bitový porovnávací registr (Compare).
- 10 bitový PWM Master/Slave registr (cyklický režim).

MCU obsahuje celkem 2 moduly (CCP1, CCP2), které jsou identické v operacích s výjimkou provádění speciální spouštěcích událostí. Oba moduly mají stejnou frekvenci a obnovovací rychlost (přerušení od TMR2).

CCP1 modul:

- Hlavní registr CCPR1 (Capture/Compare/PWM register 1) je složen ze dvou 8 bitových (CCPR1L – dolních 8 bitů, CCPR1H – horních 8 bitů).
- Řídicí registr CCP1CON řídí operace modulu.
- Speciální spouštěcí událost je vygenerována shodou (dočítáním do nastavené úrovně) a následným resetem čítače 1 (Timer1).

CCP2 modul:

- Analogie s CCP1 modulem (CCPR2, CCP2CON)
- Navíc po resetu Timer1 zahájí A/D převod, tedy pokud je A/D modul povolen.

Konfigurace modulů CCP 1, 2 (CCPRxCON):

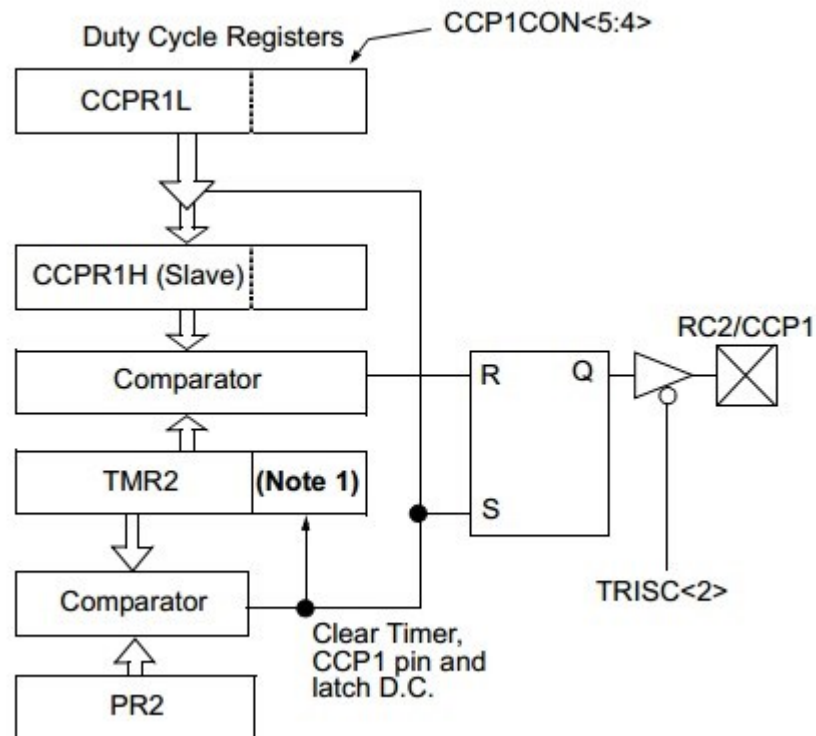
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							bit 0

Obrázek 18 – Registr CCPRxCON [MICROCHIP, 2013]

- Bity <7:6> nejsou implementovány.
- Bity <5:4> dva nejméně významové (LSbs) bity PWM pracovního cyklu. Osm více významných (MSbs) lze nalézt v registru CCPRxL.
- Bity <3:0> slouží pro nastavení pracovního módu modulu (Capture/Compare/PWM), pro nastavení PWM módu je třeba nastavit bity CCPxM3 a CCPxM2 na log. 1. Zbylé spodní 2 bity pro PWM nemají význam (ignoruje se jejich obsah)

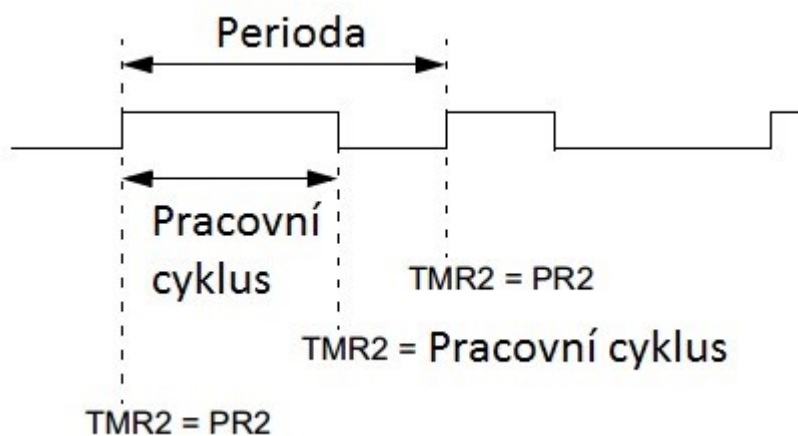
PWM mód

PWM výstup z modulu CCPx dosahuje až 10 bitového rozlišení. Výstupní pin modulu CCP1 je multiplexován s portem C (TRISC<2>). Pro uvolnění datového zámku, je třeba nastavit příslušný pin, jako výstupní.



Obrázek 19 – Blokové schéma popisující princip PWM modulu [MICROCHIP, 2013]

PWM výstup má časovou základnu (periodu) a pracovní oblast (čas), ve kterém je výstup nastaven na vysokou úroveň (log. 1). Frekvence PWM signálu odpovídá převrácené hodnotě periody $f = \frac{1}{T}$.



Obrázek 20 – Časový průběh signálu PWM [MICROCHIP, 2013]

Perioda (frekvence) PWM

Hodnota periody je lze nastavit zápisem do registru PR2. Výpočet její hodnoty je uveden pomocí vzorce:

$$T_{PWM} = [(PR2) + 1] * 4 * T_{OSC} \quad (2.1)$$

* (hodnota předěličky TMR2)

[MICROCHIP, 2013]

Při shodě hodnoty registru TMR2 a PR2, budou provedeny následující tři události (start nového cyklu resp. periody):

- Registr TMR2 bude resetován
- Pin modulu CCP1 je nastaven na vysokou úroveň (log. 1). Výjimka nastane v případě, kdy pracovní cyklus PWM má být 0%, pin CCP1 nebude nastaven.
- Hodnota PWM pracovního cyklu je načtena z registru CCPR1L a uložena do CCPR1H.

Pracovní cyklus PWM

Pracovní cyklus je specifikován zapsáním hodnoty do registru CCPR1L a nastavením dvou bitů na v registru CCP1CON<5:4>. Maximální rozlišení 10 bitů je dosaženo spojením hodnoty registrů CCPR1L(8 bitů vyšší úrovně) a CCP1CON<5:4>(nižší úrovně). Následné vyjádření popisuje výpočet pracovního cyklu PWM:

$$\begin{aligned} \text{pracovní cyklus PWM} \\ = (CCPR1L: CCP1CON < 5:4 >) * T_{OSC} \quad (2.2) \\ * (\text{hodnota předěličky TMR2}) \end{aligned}$$

[MICROCHIP, 2013]

Hodnota pracovního cyklu může být zapsána kdykoliv do registrů CCPR1L, CCP1CON<5:4>, ale jejich hodnota bude načtena do registru CCPR1H až při novém cyklu (periodě) neboli při shodě hodnot registrů PR2 a TMR2. V režimu PWM je registr CCPR1H pouze pro čtení.

Registr CCPR1H a interní 2 bity paměti jsou použity ve funkci dvojitého vyrovnávací paměti (double-buffer) pracovního cyklu PWM. Vyrovnávací paměť je důležitá pro bezproblémový chod PWM.

Při shodě hodnoty (CCPR1H -> 8 bitů a vyrovnávací paměti -> 2 bitů) s registrem TMR2, zřetěžených s vnitřní 2 bitovými hodinami (Q) nebo dvěma bity předděličky TMR2, bude CCP1 pin vynulován.

Maximální rozlišení PWM (v bitech) pro danou frekvenci PWM je dáno vztahem:

$$\text{Rozlišení} = \frac{\log\left(\frac{F_{osc}}{F_{PWM}}\right)}{\log(2)} [\text{bit}] \quad (2.3)$$

[MICROCHIP, 2013]

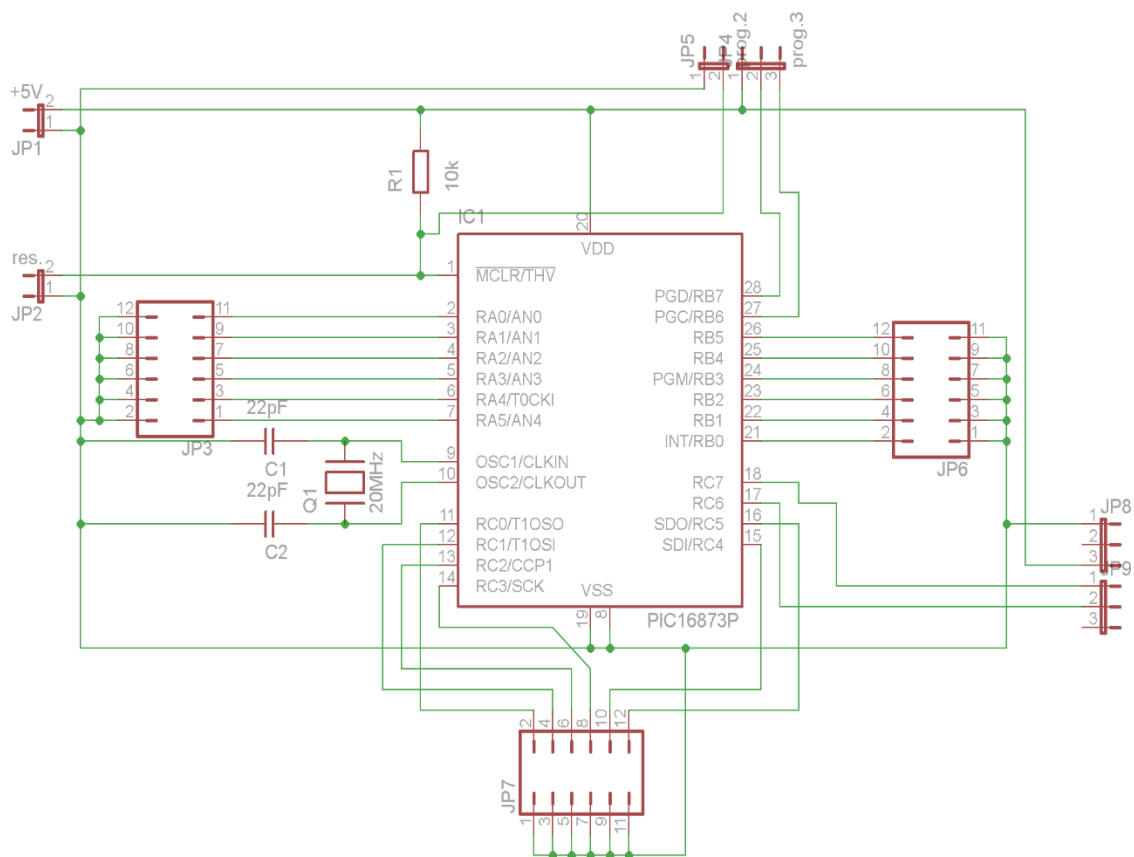
Pozn.: Pokud je hodnota pracovního cyklu delší než perioda PWM cyklu, bit CCP1 nebude nastaven na log. 0.

2.5 Základní deska modulu MCU

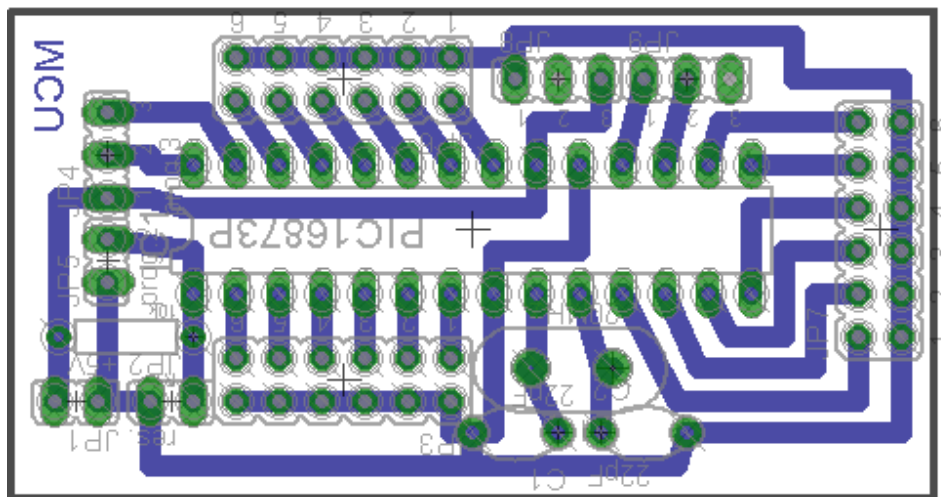
Hlavní integrovaný obvod pro MCU byl navržen v programu Eagle, kde bylo nutné navrhnout propojení MCU a ostatních součástek nutných pro realizaci komunikace s PC či vyvést jednotlivé piny MCU pro další možné užití. Výsledkem je dostupnost portu A, B a C, kdy jejich užití je všeobecné a záleží pouze na jejich nastavení (6x3 I/O, 5x A/D, 2x PWM, ...).

Tabulka 7 – Seznam použitých součástek pro základní desku MCU

Součástka	Katalogový název
MCU	PIC16F873A-I/SP SDIP28 DIL28 MICROCHIP
Port A, B, C	Dutinková lišta BL240G
Port pro programátor, reset, napájení	Dutinková lišta BTK25G
Pull-up rezistor R1	RRU 10K
Krystal	Krystal HC49/US QM 20.000MHZ
Enkodér	HCTL-2000 DIP16
Kondenzátor 2x	CK 22p/50V NPO GYM RM5,08 +/-0,5 pF



Obrázek 21 – Základní deska pro MCU



Obrázek 22 – Rozložení součástek MCU

3 Běžná komunikační rozhraní v MCU

V kapitole jsou uvedeny základní komunikační rozhraní, které se běžně používají pro komunikaci s prvky obvodu (A/D převodník, IRC enkodér, ...), nadřazeným systémem (PC, PLC, IPC,...), či mezi samotnými mikrokontroléry.

3.1 Sériové komunikační rozhraní

Sériové komunikační rozhraní lze nazvat dvěma způsoby a to zkráceně USART nebo UART. USART ,jak plyne z názvu, podporuje oba módy přenosu, tedy synchronní a asynchronní přenos. Port USART může být použit ke komunikaci s počítačem, A/D převodníkem, D/A převodníkem a sériovými paměťmi typu EEPROM.

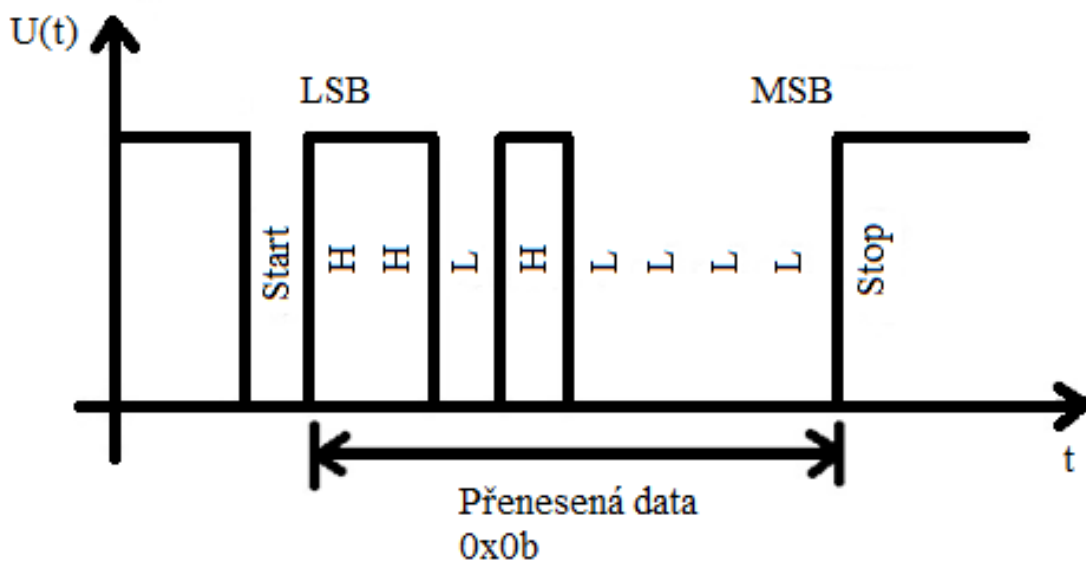
[HUANG, 2004]

Operační módy USART:

- Asynchronní mód (plně duplexní)
- Synchronní – Master (poloviční duplex)
- Synchronní – Slave (poloviční duplex)

3.1.1 Asynchronní mód

Datový formát obsahuje jeden start bit, 8 až 9 datových bitů a jeden stop bit. Jako první datový bit je přenesen nejméně významný (bit 0). Vysílač a přijímač jsou na sobě nezávislé, ale musí používat stejný datový formát a přenosovou rychlost.



Obrázek 23 – Průběh přenosu 1 bytu (1 start bit, 8 bitů data, 1 stop bit, bez parity)

3.1.2 Synchronní mód Master-Slave

Tento mód je také označován, jako SPI a velmi často se používá, pokud zařízení trpí nedostatkem vstupně/výstupních bran (pinů). SPI je synchronizační protokol, který umožňuje jednotce master zahájit komunikaci s jednotkou/jednotkami Slave. Rychlost sériové komunikace mezi dvěma či více zařízeními dosahuje vysokých rychlostí a je snadná na implementaci.

Synchronizační protokol:

- Hodinový signál poskytuje jednotka master pro umožnění synchronizace. Hodinový signál určuje, kdy je možno data číst nebo je měnit.
- Protože se jedná o synchronní komunikaci, dochází k přenosu dat po dobu trvání hodinového signálu.
- Hodinový signál se může měnit během přenosu a přitom nedojde k narušení dat. Rychlost přenosu dat se jednoduše mění spolu se změnami rychlosti hodinového signálu. SPI je tedy ideální volbou komunikace pro MCU s nepřesným taktem (např. jako zdroj hodin slouží RC oscilátor).

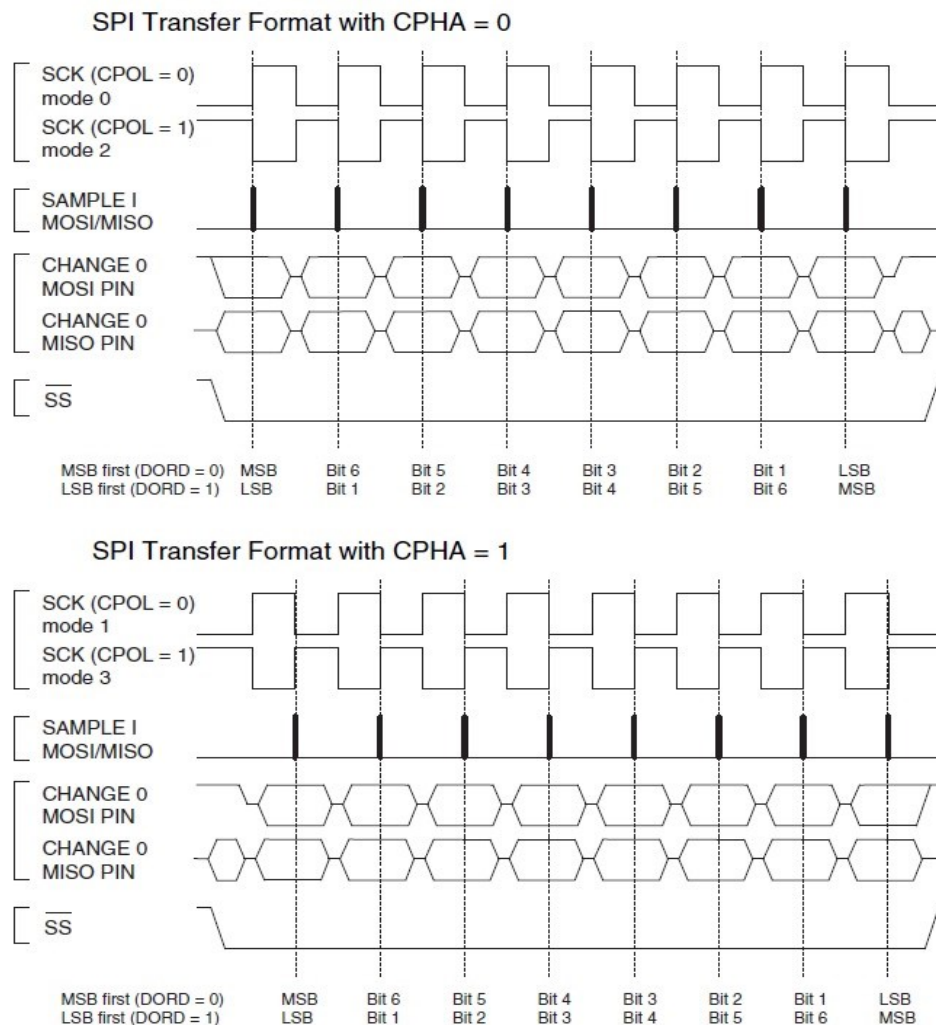
Master-Slave protokol:

- Zařízení Master řídí hodinový signál (SCK).
- K přenosu dat nedochází, pokud není hodinový signál přítomen (konstantní v čase).
- Všechny jednotky Slave jsou řízené hodinovým signálem Master jednotky.
- Způsob jakým bude jednotka Slave odpovídat je uložen v konfiguračním registru SSP.

Výměna dat:

- Jednotka Master zahájí komunikaci s vybranou jednotkou Slave tím, že nastaví u této jednotky vstup SS na log. 0 ($\sim SS = 1$, pouze tato jednotka je aktivní, ostatní zařízení typu Slave mají MISO ve vysoko impedanční úrovni). Dále Master jednotka nastaví výstup SS (občas je označen CS) na log. 1 ($\sim SS = 0$) a na této hodnotě je, dokud nedojde k uskutečnění přenosu všech dat. Samotný přenos dat začne až s generováním hodinového signálu SCK.
- Přenos proběhne oběma směry Master->Slave, Slave->Master
- Pokud jednotka Master chce poslat/přijmout další data od jednotky Slave, nepřestává vysílat hodinový signál SCK a hodnotu SS nemění.

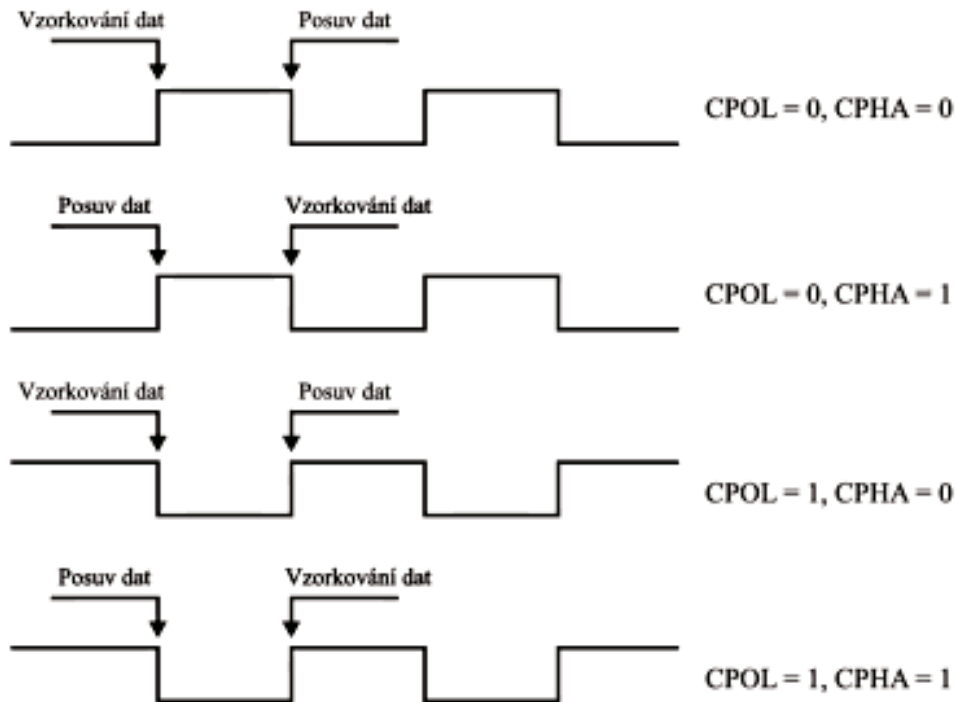
- Pro ukončení komunikace jednotka Master ukončí vysílání hodinových pulsů SCK a nastaví výstup SS zpět do log. 1 u této jednotky.



Obrázek 24 – Ukázka komunikace Master-Slave [Atmel Microcontroller, 2013]

Vzorkování dat během přenosu

- SPI pomocí specializovaného řadiče
 - Parametry CPOL a CPHA určují svou kombinací, v jakém časovém okamžiku dojde k přečtení přenášených dat, tak aby nedošlo k chybnému čtení.
- SPI programové řešení
 - Programátor musí zajistit vhodný okamžik čtení přenášených dat.



Obrázek 25 – Vzorkování dat (CPOL, CPHA) [Dudáček, 2002]

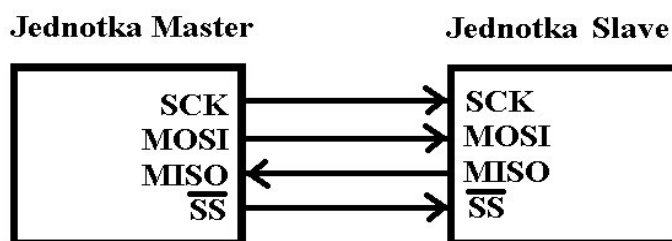
Popis V/V jednotek Master a Slave:

Master:

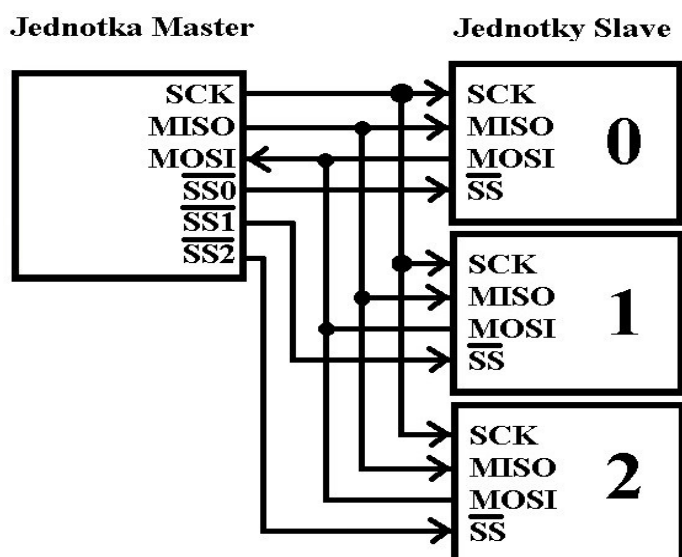
- SS – Slave select tj. výběr jednotky Slave (někdy pod názvem CS – chip select).
- MISO – Master in Slave out – příjem dat z jednotky Slave do jednotky Master.
- MOSI – Master out Slave in – přenos dat z jednotky Master do jednotky Slave.
- SCK – generovaný hodinový signál ovládající průběh komunikace.

Slave:

- SS – podle logické hodnoty vykazuje vstup buď vysokou impedanci, anebo se jednotka připraví na zahájení komunikace.
- MISO – přenos dat do řídicí jednotky.
- MOSI – příjem dat z řídicí jednotky.
- SCK – synchronizační synchronní signál pro příjem/odeslání dat.



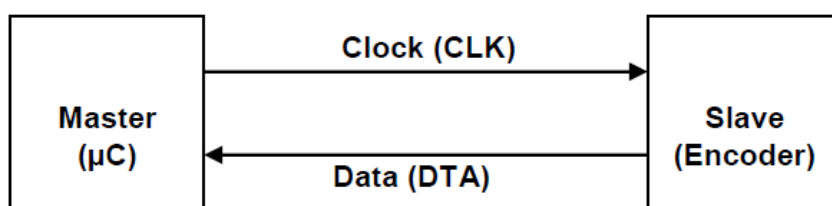
Obrázek 26 – Koncepce SPI Master-Slave [MICROCHIP, 2011]



Obrázek 27 – Koncepce SPI Master-Slave 3x [MICROCHIP, 2011]

3.2 Synchronní sériový interfejs

Jednoduchý komunikační protokol navrhnutý pro datový přenos mezi jednotkou Master (PC či uživatelským terminálem) a jednotkou Slave (tzv. smart senzory). Komunikační protokol je založený na standardech RS-422. Jedná se o připojení typu point-to-point do Mastera k Slave jednotce. Průmyslový standard TIA/EIA-422-B specifikuje úroveň rozdílového napětí řízeného přenosem obvodu. Výhoda v principu přenosu dat pomocí rozdílu mezi napětím mezi dvěma vodiči, spočívá v odolnosti proti rušení před elektromagnetickým rušením v provozu a při větší délce přenosového segmentu.

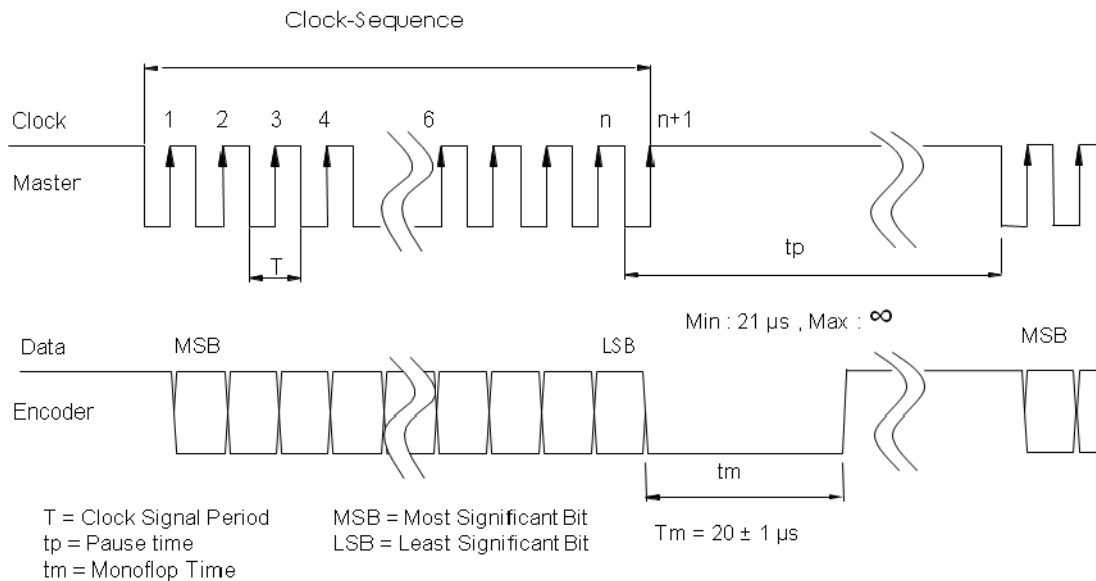


Obrázek 28 – Zjednodušené blokové schéma SSI [POSITAL GMBH, 2013]

Přenos dat po sběrnici (Master-Slave)

Data jsou vysílána, pokud jednotka Slave přijme posloupnost pulsů od jednotky Master (CLK). Posloupnost bytů je od MSB k LSB a je třeba nastavit u jednotky Master délku přenášených dat (počet bitů). Po přenosu nejméně významného bitu, senzor drží datový výstup na konstantní hodnotě po určitý čas t_p . Po uplynutí tohoto času, jsou připravena nová data k přenosu (např. poloha snímače, teplota, atd...). Při nedodržení doby odstavení

tp jsou z jednotky Slave opět přijata stejná data, jako při posledním zdařilém čtení. Datová linka během doby tp je převedena na nízkou úroveň až do uplynutí potřebné doby. Následně je nastavena na vysokou úroveň.

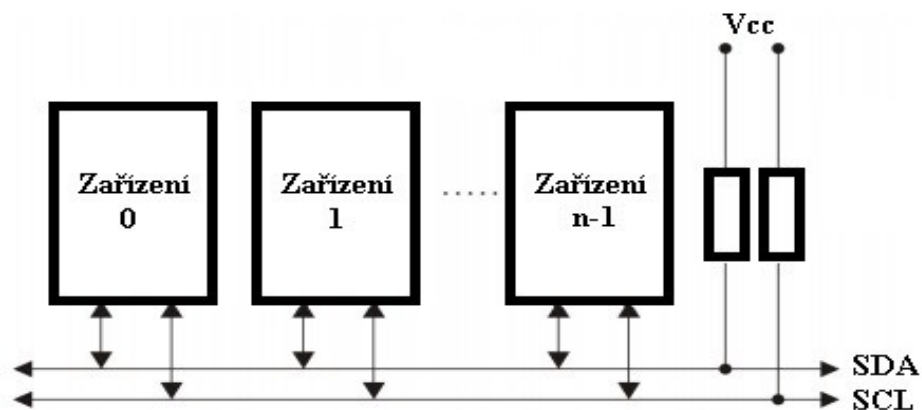


Obrázek 29 – Ukázka komunikace SSI [POSITAL GMBH, 2013]

- tm reprezentuje přenosový výpadek (monoflop time). Minimální čas potřebný pro jednotkou Slave si uvědomit, že nastal konec přenosu. Po uplynutí času tm Slave nastaví datový výstup do stavu nečinnosti a začne aktualizovat svá data v posuvném registru
- tp neboli přestávka (pause time). Zpoždění mezi dvěma po sobě jdoucími hodinovými sekvencí od jednotky Master.
- T představuje šířku (velikost) každého hodinového cyklu. Čas mezi dvěma nástupnými hranami nebo mezi dvěma sestupnými v nepřetržité hodinové sekvenci.

3.3 I²C - Vnitřní sériová linka

Byla vyvinuta za účelem komunikace mezi dílčími bloky jednoho zařízení po dvou obousměrných vodičích (SDA – datový kanál, SCL – hodinový signál). Komunikační protokol umožňuje takto propojit až 128 zařízení. Pro zajištění vysoké impedance signálu v klidovém stavu obsahuje sběrnice pull-up rezistor.



Obrázek 30 – Koncepte I2C sběrnice [HUANG, 2004]

Vlastnosti jednotek

Zařízení připojená do sběrnice musí mít přidělenou jedinečnou 7 bitovou adresu, pro jednoznačné určení adresy zařízení v rámci sběrnice. Včetně samotné podpory zařízení komunikovat po sběrnici I²C. Výstup zařízení je třístavový či s otevřeným kolektorem. Pokud jedno nebo více zařízení nastaví výstup do log. 0, na sběrnici dojde ke generování nízké úrovně.

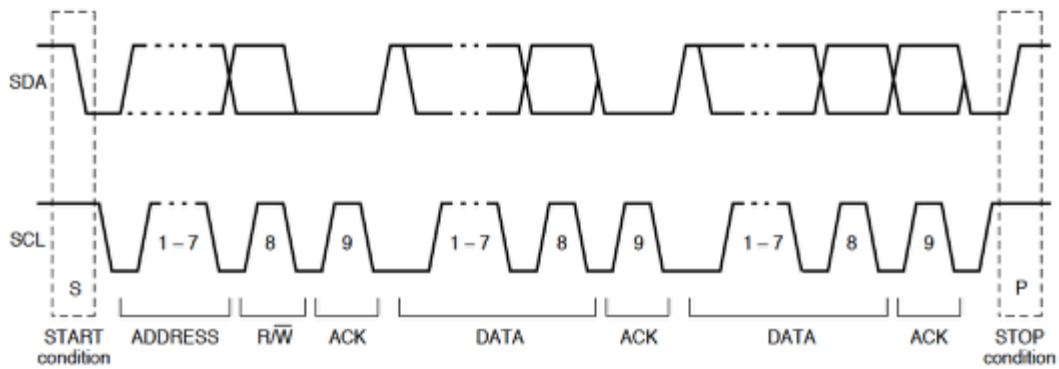
Rozdělení činností jednotek z pohledu Master a Slave.

Master:

- Zahájení komunikace (Start Condition).
- Ukončení komunikace (Stop Condition).
- Generování hodinového signálu SCL.
- Odesílání požadavků na sběrnici (datový rámeček).
- Výběr podřízeného zařízení na sběrnici (adresní rámeček).

Slave:

- Komunikační proces je řízen hodinovým signálem Master jednotky SCL.
- Jednotka je podřízené zařízení a tzv. čeká na požadavky Master jednotky.



Obrázek 31 – Komunikační rámec I²C [Prashant Jaiswar, 2013]

S každým hodinovým pulsem SCL je přenášen právě jeden bit. Logická úroveň datové linky (SDA) se během vysoké logické úrovně SCL nemění. Jedinou výjimkou je právě podmínka zahájení a ukončení komunikace, kdy změna SDA musí proběhnout v okamžiku vysoké logické úrovně SCL.

Adresní rámec je složen ze 7 bitové adresy (zařízení Slave), řídicího bitu R/~W (čtení/zápis), ~ACK (potvrzení o přijetí výzvy ke komunikaci). Pokud zařízení Slave rozpozná svou adresu během 9 cyklu SCL, nastaví SDA do log. 0, pokud tak neučiní jednotka Master ukončí komunikaci ukončovací podmínkou. První zasláný bit je nejvýznamnější.

Datový rámec obsahuje 8 bitů dat a jeden potvrzovací bit (~ACK), funkce je obdobná jako u adresního rámce.

3.4 Univerzální sériová sběrnice USB

USB je externí komunikační sběrnice, kde na jednom kabelu jsou připojena pouze 2 zařízení. Avšak komunikace může být s více zařízeními v rámci jedné USB sběrnice propojené huby tj. uzly, které zajišťují větvení. Připojení, či odpojení zařízení probíhá pomocí plug and play, tj. zapoj a hraj bez nutnosti instalace ovladačů nebo restartování zařízení. Dále přivádí napájení do zařízení o velikosti 5V.

3.4.1 Vlastnosti USB

Schéma USB je založeno na hvězdicové topologii, a proto ani neobsahuje žádný cyklus, tzn. zařízení nelze zapojit do kruhu. Řídicím prvkem je zařízení v režimu Master a k němu

může být připojeno až 127 zařízení typu Slave. K zařízení v režimu Master lze připojit rozbočovače, které zajistí připojení dalších zařízení typu Slave, ale není možné zapojit 2 zařízení v režimu Master.

3.4.2 Fyzická vrstva

Používají se dva základní typy konektorů. Typ A, který se používá na straně hostitelského zařízení (Master) a typ B. Tento typ se využívá většinou na straně připojených zařízení. Kabel, který se používá na propojení mezi zařízeními, se skládá ze 4 vodičů, z toho jsou 2 vodiče určené k napájení a další 2 na datovou část.

3.4.3 Způsob přenosu dat

Data se vysílají v krátkých paketech o 8 bajtech a delších paketech o délce až 256 bajtů. PC může požadovat data od zařízení, naopak žádné zařízení nemůže vysílat data samo od sebe.

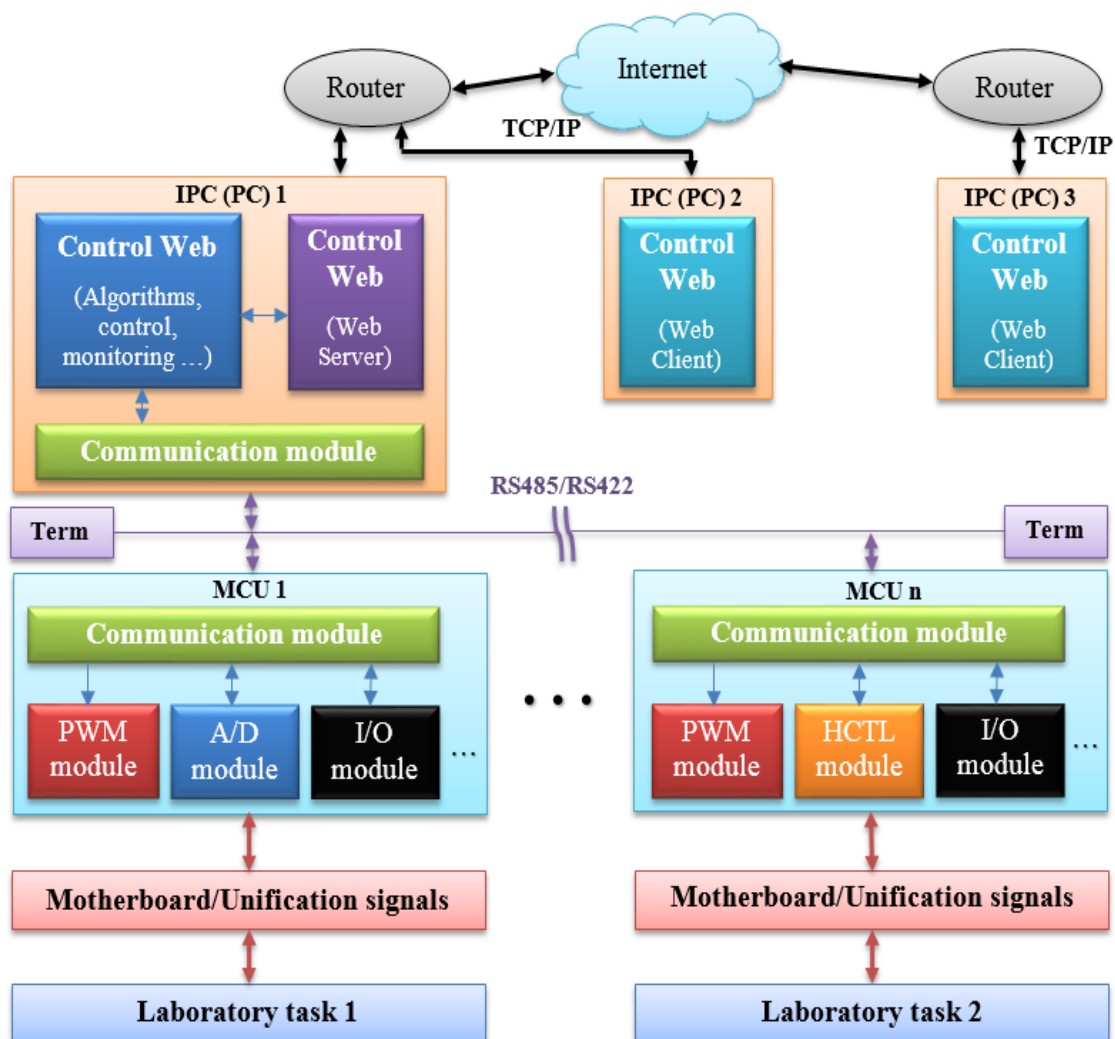
Veškerý přenos dat se uskutečňuje v tzv. rámcích, které trvají přesně 1 milisekundu. Uvnitř jednoho rámce mohou být postupně zpracovávány pakety pro několik zařízení. Přitom se mohou spolu vyskytovat pomalé (low-speed) i rychlé (full-speed) pakety. Obrací-li se PC na více zařízení, zajišťuje jejich rozdělení, jako rozdělovač sběrnice (hub). Zabraňuje také, aby signály s plnou rychlostí (full-speed) byly vedeny na pomalá zařízení. Časový průběh přenosu informace je předepisován výhradně Masterem. Zařízení typu Slave se musí synchronizovat s datovým tokem.

4 Návrh řídicího systému (SCADA/HMI)

Řídicí systému na bázi jednočipových počítačů, pro monitorování a řízení již existujících laboratorních úloh na katedře, je třeba přizpůsobit jednotlivým úlohám. Pro optimální řešení různorodosti laboratorních úloh, bylo třeba systém navrhnout s maximální důkladem na obecnost systému a jeho snadnou rozšiřitelnost. Z tohoto důvodu je nutné oddělit výkonovou část (MCU) od části unifikace (přizpůsobení vstupního/výstupního signálu potřebám daných úloh). Spolu s vyřešením komunikace a možností, danou úlohu připojit bez nutnosti připojení zařízení na společnou sběrnici (např. pro nahrání nového firmwaru, testování úlohy, atd.).

4.1 Logická struktura systému

Návrh distribuovaného řídicího systému, lze rozdělit na základní dílčí části a to řízený/monitorovaný proces (laboratorní úloha), řídicí podsystém (základní a rozšiřující deska), komunikační rozhraní, řídicí systém, webový server a klient. Laboratorní úlohy a jejich popis byl již zmíněn, spolu s návrhem unifikace/rozšiřující desky a MCU. Jednotlivé řídicí podsystémy (Slave) komunikují s řídicím systémem (Master) po sériovém rozhraní RS422/RS485, které je tedy v prvním případě schopno komunikovat obousměrně (full duplex) či v druhém provedení jednosměrně (half duplex). Je tedy při návrhu komunikace myslet na řízení toku komunikace mezi zařízeními Master a zařízeními Slave. Při přechodu mezi jednotlivými úlohami detekovat, zdali požadovaný systém je dosažitelný (z inicializovaný, připojený ke sběrnici, atd.). Řídicí systém spolu s webovým serverem se nachází na jednom PC, ve kterém běží plná verze řídicího programu, jehož vygenerovaná (ořezaná verze) je spuštěna na webovém serveru. Rozdíl mezi přímým přístupem a vzdáleným je tedy v počtu zobrazených (generovaných) komponent a množství zasílaných dat, tak aby nebyla příliš zatěžována síť. Přístup klienta je tedy pomocí webového serveru omezen (zadávaní parametrů regulace, které by mohly mít destruktivní následky na samotnou úlohu).



Obrázek 32 – Blokové schéma řídicího systému

4.2 Komunikačního rozhraní

Komunikační rozhraní zajišťuje samotný přenos informace mezi řídicím systémem a řízeným systémem, proto je velmi důležité vhodně zvolit druh komunikace spolu se zabezpečením přenášené informace (proti rušení, poškození přenášených dat, ...). Vzhledem k předpokládané topologii SCADA/HMI systému je vhodné zvolit sériové komunikační rozhraní RS232 pro přenos dat mezi řídicím systémem (Master) a řízeným systémem (Slave).

4.2.1 Point-to-Point

Pro vývoj, testování či nahrání nového firmwaru bylo třeba zajistit snadnou komunikaci mezi IPC (PC) a MCU, tak aby k provedení těchto úkonů nebylo třeba externího napájení

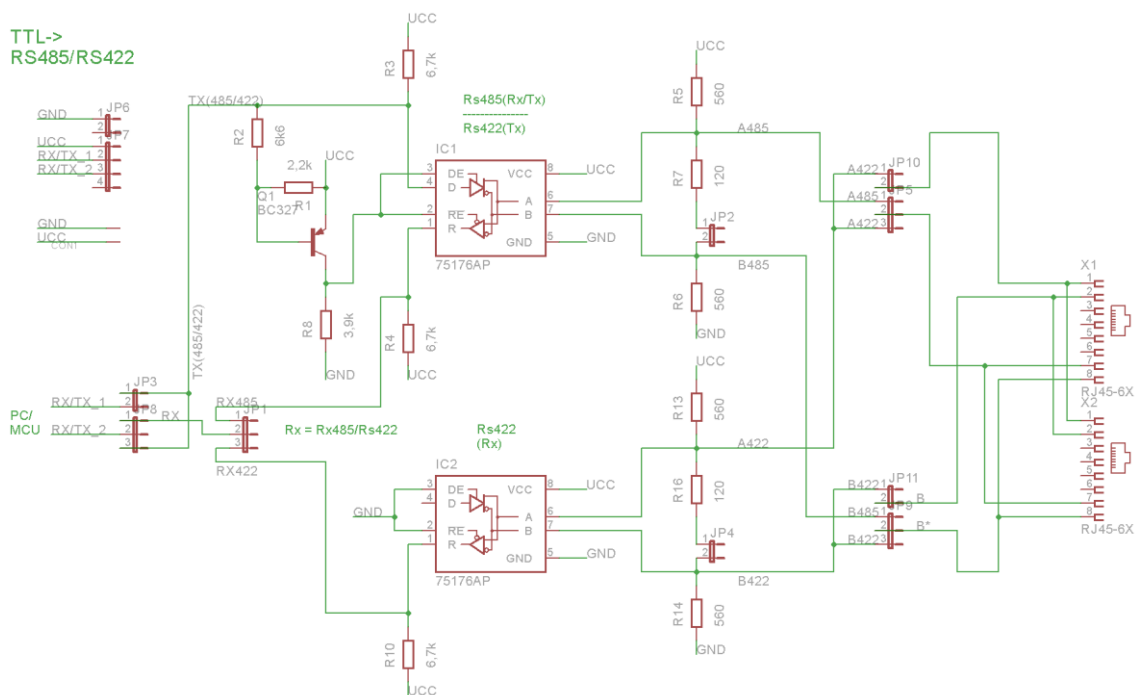
řídící jednotky či trvalého připojení ke komunikační sběrnici (imobilita). Každý z navržených modulů je možné připojit přímo k PC přes přípravek ARDUINO (USB to Serial). Z USB portu je napájen přípravek ARDUINO spolu s připojeným řídicím systémem (základní deska MCU a rozšiřující deska).



Obrázek 33 – Arduino USB [USB Serial Light Adapter, 2015]

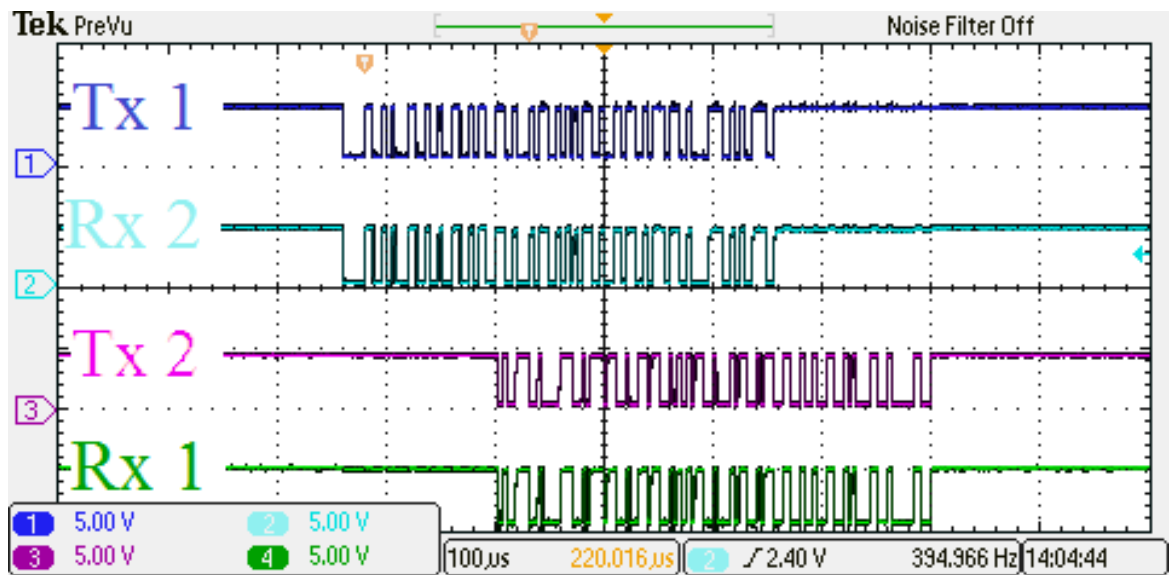
4.2.2 Průmyslová sběrnice RS422/RS485

Rozhraní RS232 je vhodné pro komunikaci na krátkou vzdálenost a původně bylo navrženo pouze pro komunikaci Point-to-Point. Během vývoje byly představeny i možné variace rozhraní pro více zařízení na sběrnici, ale pro možnost většího rozšíření počtu zařízení či zajištění odolnosti proti rušení bylo vhodné komunikační rozhraní změnit. O přenos dat (TTL úrovně RS232) mezi komunikačními deskami (Master, Slave) se postará rozhraní RS485/RS422, které fyzicky přenese požadovaná data po sběrnici.



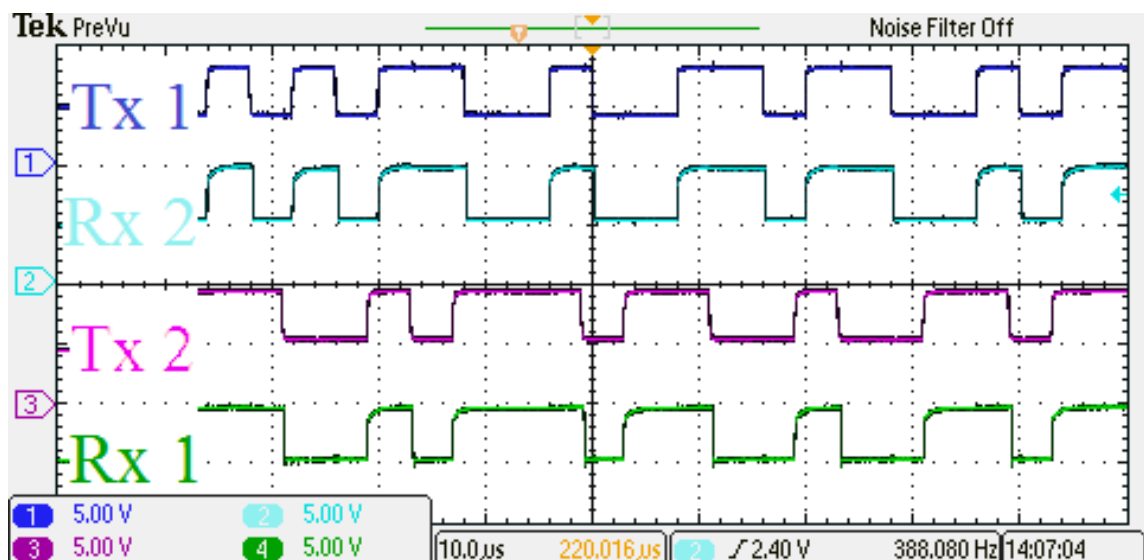
Obrázek 34 – Komunikační deska Master/Slave

(posílání dat) Tx z 5V na 0V dojde k sepnutí tranzistoru mezi B-E a na přechodu mezi B-C bude vygenerováno plné napájecí napětí (tj. $E \approx C$).



Obrázek 36 – Ukázka obousměrné komunikace při rychlosti 256 Kb/s (RS232(TTL)<->RS422<->RS232(TTL))

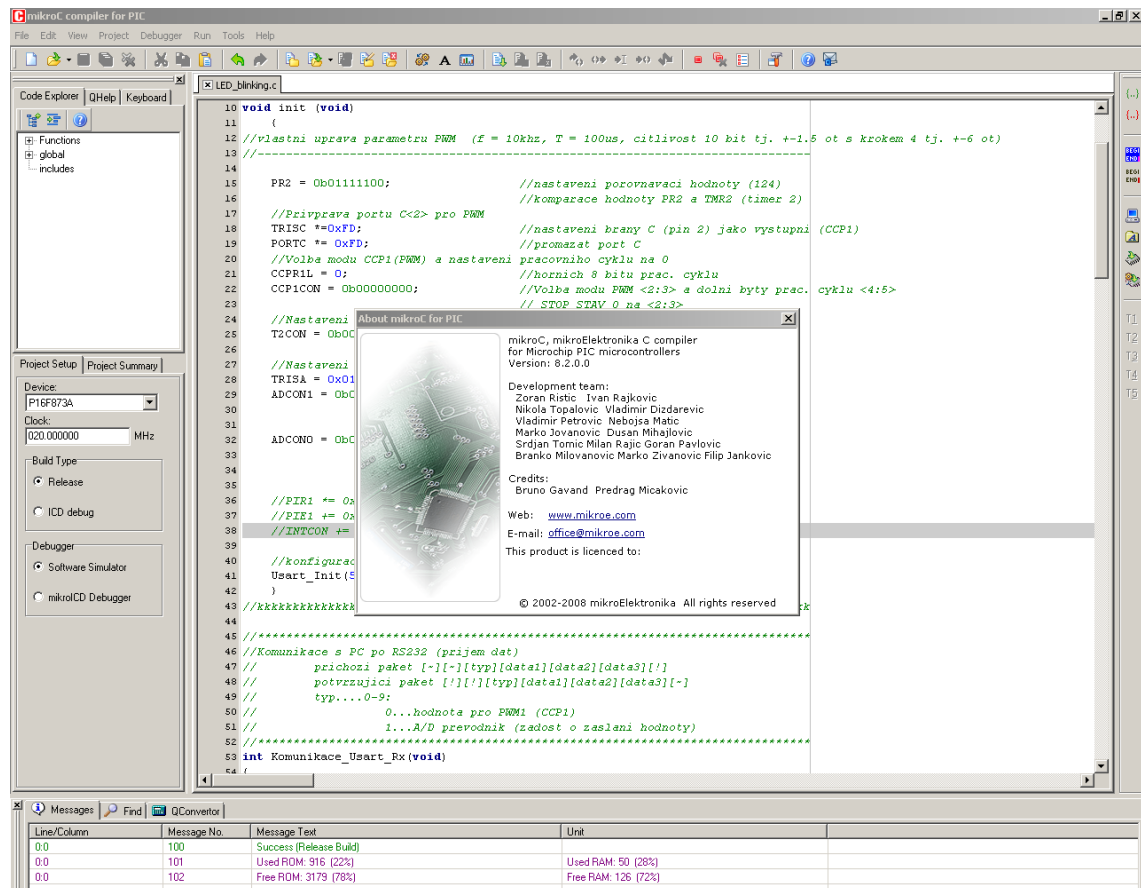
Navržená komunikační deska zvládá maximální možnou přenosovou rychlost 256 Kb/s komunikačního rozhraní RS232 velmi dobře, tj. rozdíl ve tvaru průběhu mezi vyslanou a přijatou zprávou (úrovni TTL) mezi komunikujícími stanicemi, nedochází k výrazné deformaci signálu. V průběhu signálu (na přijímací straně) lze vidět spínání (doba náběhu) tranzistoru a jeho rozpínání (doba doběhu) tranzistoru.



Obrázek 37 – Ukázka (přiblížení) obousměrné komunikace při rychlosti 256Kb/s (RS232(TTL)<->RS422<->RS232(TTL))

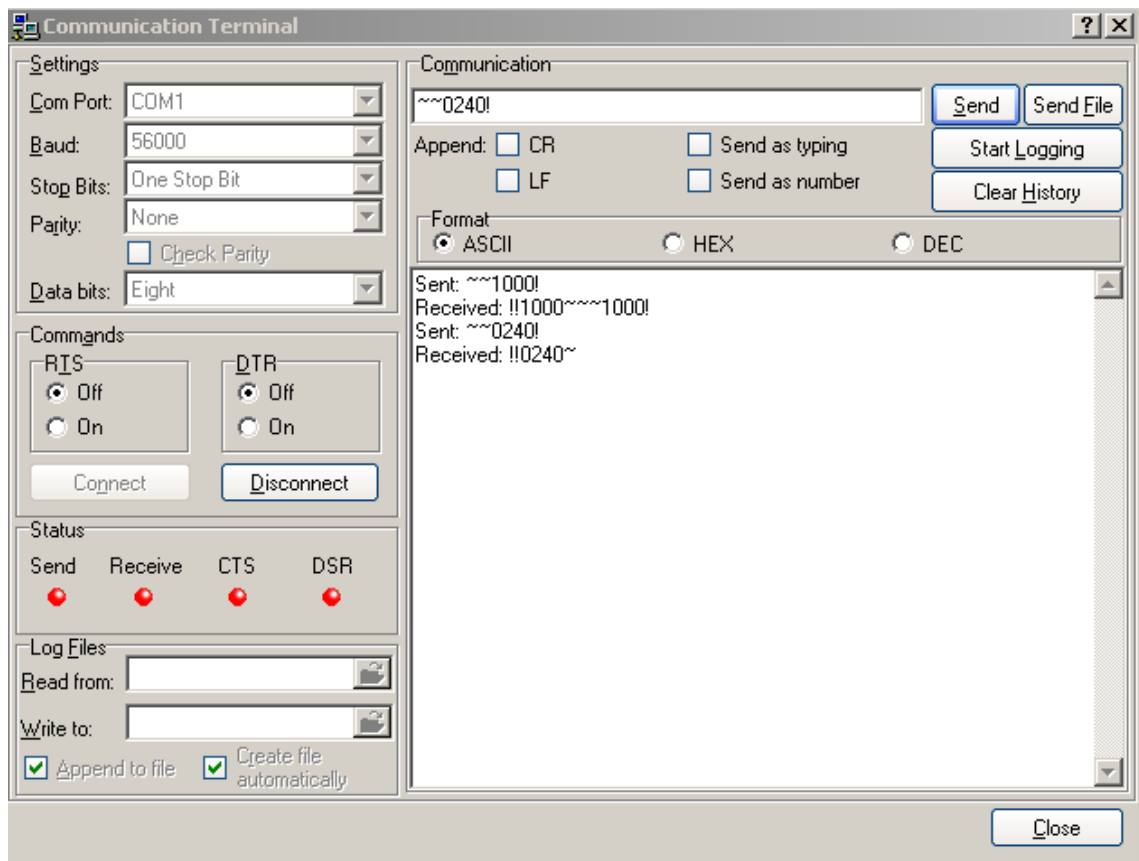
5 Programovací prostředí mikroC

MikroC je plnohodnotný ANSI C kompilátor pro 5 různých architektur MCU. Nabízí velkou škálu hardwarových a softwarových knihoven a dalších nástrojů, které usnadňují práci se samotným zařízením. Každý kompilátor obsahuje rozsáhlou nápovědu a spousty dalších příkladů užití (kódové ukázky). Licence kompilátoru zahrnuje zdarma update a doživotní technickou podporou.



Obrázek 38 – Ukázka programovacího prostředí

Mezi užitečné nástroje programu patří Terminál, či zobrazení ASCII tabulky pro rychlou orientaci ve znakové sadě či komunikaci s PC pomocí sériového rozhraní s možností nastavení veškerých potřebných parametrů komunikace (rychlost, parita, port, atd.).



Obrázek 39 – Terminál v prostředí mikorC

Demo aplikace

Ukázka krátkých programů pro seznámení se stylem programování MCU (PIC16F873A). Kód programu lze obecně rozdělit na 2 základní části:

Inicializační blok:

Oblast kódu, kde probíhá inicializace dílčích součástí MCU použitých v programu (portu I/O, periférie, Č/Č, ...).

Hlavní blok:

Na rozdíl od klasického stylu programování programů pod libovolným OS, je chod hlavního programu MCU cyklicky opakován tj. uzavřen v nekonečném cyklu (obvykle while, do...while).

5.1.1 Rozblikání LED

V inicializačním bloku je třeba nastavit směr daného portu (I/O) a v hlavním bloku provádět zápis log. 1 (LED...ON) či log. 0 (LED...OFF) na příslušný pin portu. Protože provádění 1 instrukce zabere velmi krátký čas (cca 200 μ s), je třeba zahrnout i zpoždění mezi změnami, aby bylo vidět přepínání logické úrovně (blikání LED).

5.1.2 Komunikace po USART

Komunikace po USART v našem případě probíhá mezi MCU a PC, pro komunikaci lze použít již zmíněný program terminál. Před komunikací je třeba nastavit komunikační rychlost i na PC a to ve stejné formě, jako u MCU (rychlost 9600 baudu/s,...).

5.1.3 Rozsvícení LED pomocí PWM

Pozvolné rozsvícení LED pomocí 10 bitového (se změnou horních 8 bitů) PWM signálu. LED je třeba umístit na port C <2>, kde se nachází výstup CPP1 modulu použitého v příkladu. Po inicializaci bude LED postupně rozsvěcována s krokem 1 (reálně s krokem 4) po cca 30 ms do plného svitu (cyklicky 0-255).

5.1.4 A/D převod analogového signálu

Vyhodnocení analogové veličiny (signálu) pomocí A/D převodníku na portu A<0>. Před spuštěním převodníku je třeba nastavit jeho rychlost, kanál, formát uložení, atd. Výsledek převodu je zobrazen na portu B a C<7:6> respektive horních 8 bitů převodu 10 bitového A/D převodníku.

Pozn.: Zdrojový kód uvedených programů se nachází v příloze.

6 Realizace algoritmů pro MCU v prostředí mikroC

Inicializace MCU a jeho periférii je provedeno na základě požadavku z vyšší úrovně. Do té doby je MCU ve stavu klidu (v provozu je pouze komunikace s IPC). Nahraný program dle stanoveného typu úlohy (definováno před samotnou kompilací uživatelem v prostředí mikro C), již obsahuje obecné funkce, které jsou definovány pro vybranou úlohu. Pomocí direktiv pro preprocesor, je dosažena redukce kompilovaného programu, čímž vzniká více místa pro programátora na danou úlohu. V MCU je naprogramován obecný přístup dle přidělené adresy k požadované periférii, parametru, atd. Komunikace je zabezpečena vlastním algoritmem (výpočet kontrolního součtu, bitového posuvu dat, se zajištěním nenulového paketu, atd.). Program je podle předem přidělené adresy pro laboratorní úlohu rozdělen na samostatné bloky (nekonečné smyčky). Celý program je jakožto celek rozdělen do několika souborů (oddělený překlad), které svým obsahem i názvem zcela

definují jejich význam (Ini_Soubory.h, Komunikace.h, ...). Jednotlivé soubory budou v následujících kapitolách popsány.

6.1 Hlavní program

Hlavní (spouštěcí) soubor „Main.c“ obsahuje deklaraci a definici lokálních (globálních) proměnných použitých v programu, spolu s hlavní funkcí. Při spuštění programu (obvykle po resetu, či po zapnutí napájení) se provede z bezpečnostních důvodů promazání hodnot na všech portech. Následně dochází k inicializaci modulu USART, který zajišťuje komunikaci s MCU. Parametry komunikace jsou nastaveny následovně.

Tabulka 8 – Parametry komunikace USART modulu

Rychlost:	256 Kb/s	Stop bit (počet):	1
Parita:	Není využita	Data (počet bitů):	8
Start bit (počet):	1		

Při úspěšné inicializaci komunikace bude MCU v pasivním stavu čekat na potvrzení své inicializace (přeložené a nahané dle jeho adresy viz direktivy pro preprocesor). Při příjmu dat z PC, zkontroluje všechny náležitosti přenosu (adresa MCU, zabezpečení paketu, atd.) a podle přijatých dat, dojde ke spuštění nekonečné smyčky pro příslušnou úlohu. Zde se již program začíná členit na bloky, které odpovídají svým složením potřebám konkrétní úlohy, ale funkce použité v něm jsou opět obecné (nastavení parametrů regulace, frekvence PWM, atd.). Pokud je použita regulace přímo v MCU, je tento blok doplněn o kontrolu (přetečení) časovače. Např. úloha DC Motor potřebuje dodržet vzorkování 5 ms, pro vyčtení odpovídajícího počtu naměřených otáček motoru ve stanoveném časovém limitu. Proto je nutné zkontrolovat zbývající čas, před vykonáním kontroly, zdali nepřišly nová data z vyšší úrovně (zpoždění cca 1 ms). Celkové zpoždění by vedlo k nesprávnému výsledku načítaných impulsů za časovou jednotku tj. nesoulad mezi vypočtenou a reálnou rychlostí.

6.2 Inicializace MCU

Soubor „Ini_Soubory.c“ uchovává inicializace modulu rozdělených do jednotlivých bloků, které jsou vytvořeny dle požadavků pro jednotlivé laboratorní úlohy (DC Motor, TVM, TENZO). Funkce cyklicky kontroluje komunikačního rozhraní, jestli nedošly nová

data a pokud ano, provede celkovou kontrolu přijatého paketu. Když má adresa periferie příslušnou hodnotu („Z“...inicializace), dojde k provedení inicializačního bloku, který je požadován (dle přijaté hodnoty). Na konci každé inicializace dojde k zaslání potvrzení od jednotky MCU (přeпоšle zpět přijatá dat od PC).

Pro správnou funkčnost řídicí jednotky v režimu úlohy DC Motor je třeba správně nastavit modul CCP1 (PWM1), port (A, B, C), časovač 1 (vzorkovací čas 5 ms) s ohledem na zapojení jednotlivých pinů (I/O).

6.3 Komunikace

Soubor „Komunikace.h“ obsahuje funkce pro zajištění komunikace mezi nadřazeným systémem a řízeným systémem. Funkce pro příjem „Komunikace_Usart_Rx“, provádí kontrolu přijatých dat podle definovaných pravidel (kontrolní součet, definované rozsahy, atpod.), tak aby nedocházelo k možným problémům (zpracování poškozených dat).

Pokud jsou data připravena (byla provedena detekce) funkce může načíst jeden znak (významově adresu MCU). Ihned po přijetí zkontroluje, zdali se nejedná o shodnou adresu daného zařízení (při neshodě dojde k ukončení čtení). Při shodě dochází k čekání, dokud se aktuálně zpracovaný znak nenahradí novým (kontrola příznaku) ve stanoveném přípustném čase, odpovídajícímu době přijetí nového znaku při stanovené rychlosti komunikace (funkce „Cekej“). Pro nový znak (adresa periferie/fce) je stanovený rozsah platného znaku [65-90] a pokud jej splňuje, pokračuje dále s načítáním dalšího znaku (kontrola příznaku, čekání). Následující dva bajty obsahující přenášená data, přijme bez kontroly jejich obsahu (hlídají se pouze zakázané meze, tj. znak 0 a 255). Během celého příjmu, již zmíněných znaků, se provádí výpočet kontrolního součtu, který při příchodu nadcházejícího znaku bude porovnán s hodnotou přijatého znaku (kontrolní součet zasláný z PC). Vypočtený kontrolní součet je 7 bitový a podle výsledku jejich komparace se čtení ukončí, nebo se vyčká na přijetí poslední zprávy tj. terminačním znaku (255).

Tabulka 9 – Ukázka komunikačního paketu Rx

ADR_MCU	ADR_PER	DATA_H	DATA_L	CH_SUM	TERM
----------------	----------------	---------------	---------------	---------------	-------------

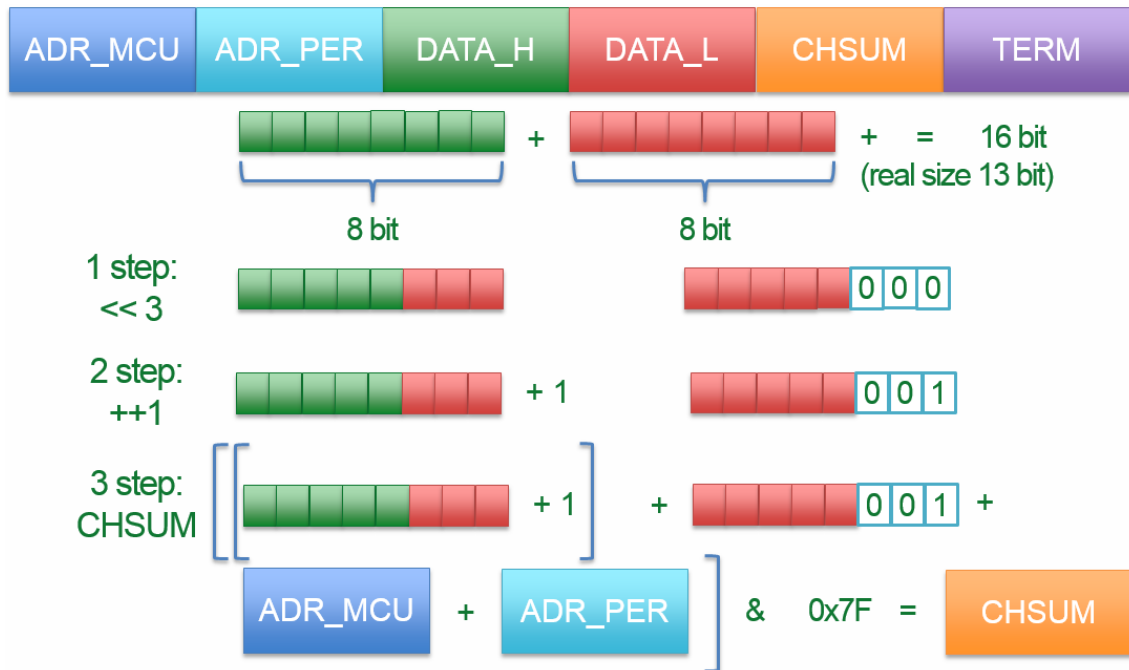
Zatím co výstupní komunikaci má na starost „Komunikace_Usart_Tx“, která podle přijatých parametrů (adresa periferie a počet přenášených dat) provede kódování dat

připravených k odeslání. Paket pro příjem i odeslání mají stejný formát, jen při odeslání dat ze všech čtyř A/D převodů se provede jejich odeslání najednou, tj. délka paketu je prodloužena o dalších 6 bajtů.

Tabulka 10 – Ukázka komunikačního paketu Tx

ADR_ MCU	ADR_ PER	DATA_ H0	DATA_ L0	...	DATA_ H3	DATA_ L3	CH_ SUM	TERM
---------------------------	---------------------------	---------------------------	---------------------------	-----	---------------------------	---------------------------	--------------------------	-------------

Funkce je nutná pro výpočet kontrolního součtu a kódování přenášených dat. Data pro odeslání musí projít bitovou rotací o 3 pozice vlevo (přenos mezi spodním a horním bajtem) a přičtením hodnoty 1, ke spodnímu i hornímu bajtu. Následuje 7 bitový výpočet kontrolního součtu. V posledním kroku dochází k odeslání dat a připojení terminačního znaku na konec zprávy.



Obrázek 40 – Zabezpečení paketu

Poslední funkcí je „Rychla_Odpoved“, která slouží k rychlému přeposlání přijatých dat od PC. V rámci co nejrychlejší odpovědi bez nutnosti upravovat přijatá data, dojde k jejich odeslání v původním stavu, v co nejkratší době. Funkce je využita až po vykonání požadovaných kroků (operací), tj. pokud je zpráva přijata celá a v pořádku, řídicí jednotka provede požadovanou sekvenci příkazů, na jehož konci bude vyslána rychlá odpověď.

6.4 Funkce MCU

Soubor „FunkceMCU.h“ zajišťuje hlavní funkce řídicí jednotky pro zpracování dat ze zpětné vazby, reset, detekce přetečení čítače, PID regulaci, úpravu dat (dekódování přijaté zprávy) a nastavení frekvence PWM.

Obsluhu pro přerušení od čítače 1 má na starost funkce „Interrupt“, která smaže příznak přerušení a opětovně nastaví požadovanou hodnotu pro nové čítání, včetně nastavení globální proměnné pro vyčtení hodnoty ze zpětné vazby, resp. provedení regulace (pokud je využita). Pro úlohu DC Motor je nutné dodržet vzorkovací periodu co nejpřesněji, proto je vyčtení hodnoty aktuálního stavu otáček zcela nezávislé na požadavku od nadřazené úrovně a jeho výsledek je ukládán do vyrovnávací paměti (proměnné). Stáří každého vzorku je tedy přibližně 5 ms (odchylka je způsobena nepřesností použitého krystalu cca 1%).

O úpravu přijatých dat se stará funkce „Uprav_Data“, která po přijetí dat z nadřazené stanice provede jejich dekódování (bitová rotace, odečtení hodnoty, atd.). Dekódovaná data uloží do připraveného pole, pro další využití.

Výčet aktuální hodnoty z A/D převodu zajišťuje funkce „A_D_Read“, které je předán parametrem číslo kanálu (vstupu), jehož vstupní napětí má převést na odpovídající 10 bitový ekvivalent, při komparaci s referenčním napětím v hodnotě 5 V. Výsledek převodu uloží do pomocného pole pro další využití.

Získání aktuálního stavu otáček (relativního) má na starost funkce „HCTL_2000“, která provede definovaný postup pro vyčtení horního a po sléze dolního bytu. Po vyčtení hodnot je kvadrurní převodník uveden do resetu a pokračuje v čítání. Vyčtená data jsou upravena do požadovaného tvaru dvou bajtů a následně porovnána s maximální možnou hodnotou čítání (vychází z maximálních otáček motoru a času mezi vzorky). Pokud hodnota není dosažena, motor se otáčí v kladném směru. Při přetečení je proveden bitový doplněk (převedeno do kladného směru otáčení) a nastaven příznak, který signalizuje, že motor je v reverzním režimu či kanály CHA, CHB jsou prohozeny.

Nastavení frekvence PWM „PWM_Frekvence“, je provedeno na základě přijatého parametru s požadovanou frekvencí v jednotkách Hertz (f_{PWM}).

$$T_{PWM} = (PR2 + 1) * 4 * T_{OSC} * TMR2 [s] \quad (6.1)$$

[MICROCHIP, 2013]

Při dosazení $T = \frac{1}{f}$ a substituci součinu $[(PR2 + 1) * TMR2] = Výsledek$, bude výsledný tvar následovný:

$$Výsledek = \frac{f_{osc}}{f_{PWM} * 4} \quad (6.2)$$

Časovač 2, který generuje stejnou časovou základnu pro moduly CCP1, CCP2 má k dispozici 3 děličky v hodnotách 1, 4, 16. Postupným vydělením výsledku děličkami od nejnižší po nejvyšší a jeho komparací s maximální velikostí registru PR2 (1 B...255), se dojde k výslednému nastavení registrů TMR2 (dělička) a PR2 (hodnota čítání).

PID regulace „PID_Controller“ je volána v příslušný časový okamžik (dle úlohy), kdy před jeho zavoláním dojde k načtení hodnoty ze zpětné vazby a výsledek je upraven dle potřeby na daný rozměr (např. ot/m). Následuje výpočet aktuální odchylky od žádané hodnoty.

$$e = w - y \quad (6.3)$$

Použitý algoritmus je přírůstkový (rekurzivní), pro menší paměťové nároky (polohový provádí sumaci odchylek od počátku algoritmu, tj. vysoká možnost přetečení datového typu). Algoritmus PID regulace je shodný s použitým v nadřazené úrovni a má následující tvar:

$$u_{k+1} = u_k + k_p * [e_{k+1} - e_k + \frac{T_{VZ}}{T_i} e_{k+1} + \frac{T_d}{T_{VZ}} (e_{k+1} - 2 * e_k + e_{k-1})] \quad (6.4)$$

[Ščevík, 2008]

kde:

u_{k+1} ...aktuální akční zásah	e_{k+1} ...aktuální odchylka
u_k ...předešlý akční zásah	e_k ...předchozí odchylka
k_p ...zesílení	e_{k-1} ...před předchozí odchylka
T_i ...integrační časová konstanta	T_{VZ} ...vzorkovací perioda
T_d ...derivační časová konstanta	

Algoritmus tedy musí uchovávat předešlý akční zásah a poslední dvě vypočtené odchylky. Vzhledem k reálnému omezení velikosti akčního zásahu na 10 bitový rozsah, je třeba omezit jeho velikost v obou směrech (přetečení, podtečení rozsahu). Neboli aplikovat tzv. AntiWindup pro omezení sumace akčního zásahu, např. při přidržení motoru či zapojení zátěže na maximální možnou hranici akčního zásahu. Po ošetření vypočteného zásahu dojde k jeho okamžitému provedení a odeslání jeho velikosti spolu s hodnotou ze zpětné vazby (např. DC Motor – naměřené otáčky).

6.5 Makra a sdílené proměnné

Poslední částí programu je soubor „Makra_Globalni.h“, ve kterém jsou definovány globální proměnné a makra. Kdy za pomoci maker je definována adresa zařízení, vzorkovací perioda, maximální akční zásah, atpod. Spolu se sdílenými (globálními) proměnnými dosažitelnými pro jednotlivé funkce v rámci celého programu.

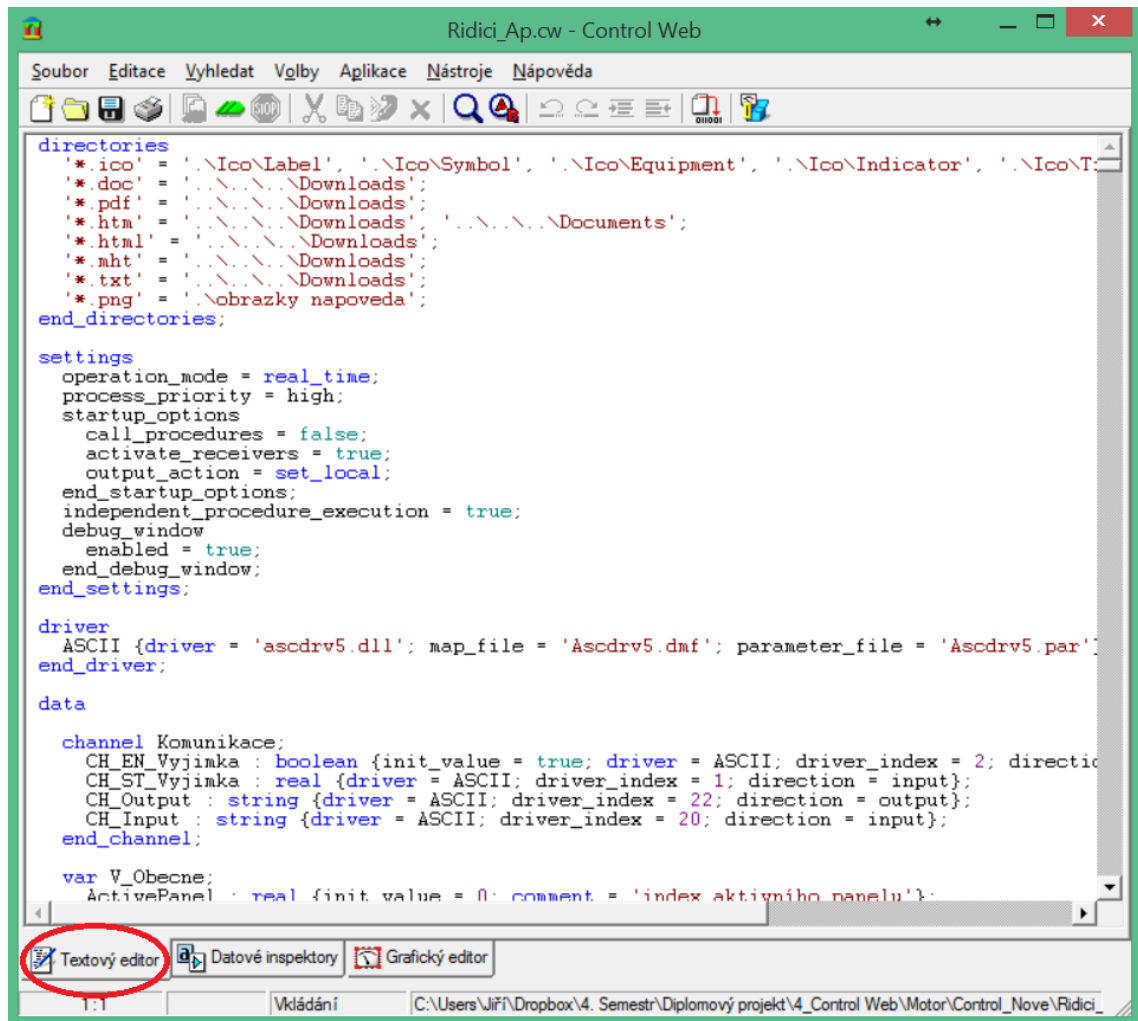
Při nahrání firmwaru do MCU je třeba správně nastavit hodnotu makra „CHOOSE_DEVICE“, která definuje adresu MCU i výslednou velikost sestaveného kódu programu, podle předdefinované hodnoty adresy zařízení resp. laboratorní úlohy.

7 Vývojové prostředí Control Web

Vizualizační program s komponentovou a objektově orientovanou koncepcí pro návrh SCADA programů. Vhodný pro řízení technologických procesů v reálném čase, zpracování dat v počítačových sítích, v oblasti simulace a modelování, atd. Aplikace je dostupná v několika verzích produktu, ale v principu ji lze rozdělit na vývojovou a runtime verzi. Runtime verze produktu si musí zakoupit zákazník, pro spouštění na zakázku vytvořené aplikace. Zatímco vývojová verze je určena pro samotný návrh

7.1 Textový editor

Vygenerovaný kód při založení projektu či přidání nové komponenty (přístroje, panelu, programu,...) dochází k automatickému vložení celého kódu do textového editoru. Při velkých zkušenostech programátor nemusí používat přiložených nástrojů pro definici objektu, ale sám si ho nadefinovat (jen velmi zkušený uživatel). Právě pro celkovou správu aplikace je vhodné používat následně záložky datový inspektor a grafický editor, kde uživatel definuje potřebné objekty a nastaví jejich vlastnosti dle svých potřeb.

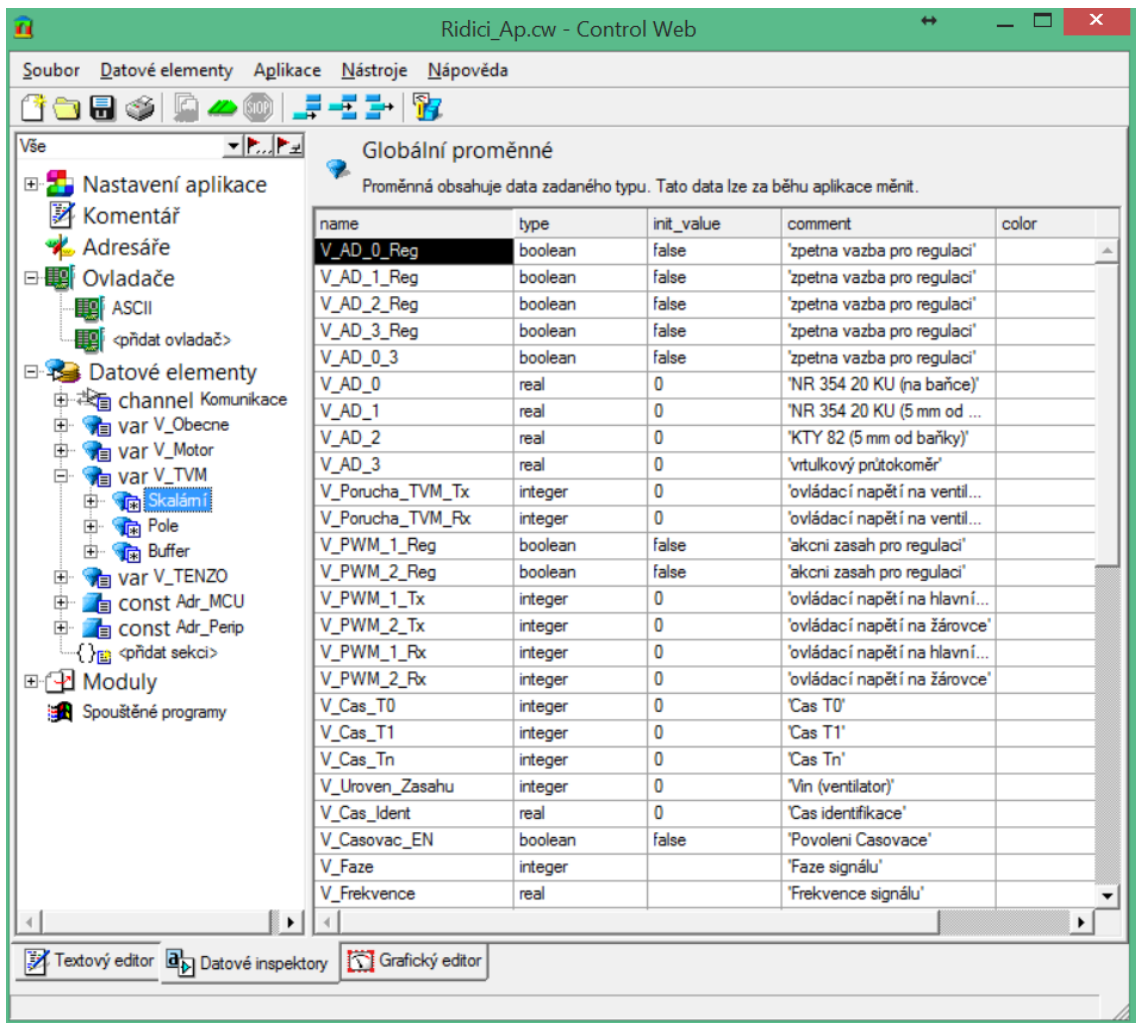


Obrázek 42 – Ukázka textového editoru

7.2 Datový inspektor

Slouží pro vytvoření datových elementů, vytvoření ovladačů, komentářů, nastavení aplikace, apod. Pokud se zaměříme například na možnosti datových elementů. Sekce je rozdělena z hlediska použití jednotlivých elementů na globální konstanty, globální proměnné, kanály, výrazy, atd. Jednotlivé podsekce lze dále dělit z hlediska jejich

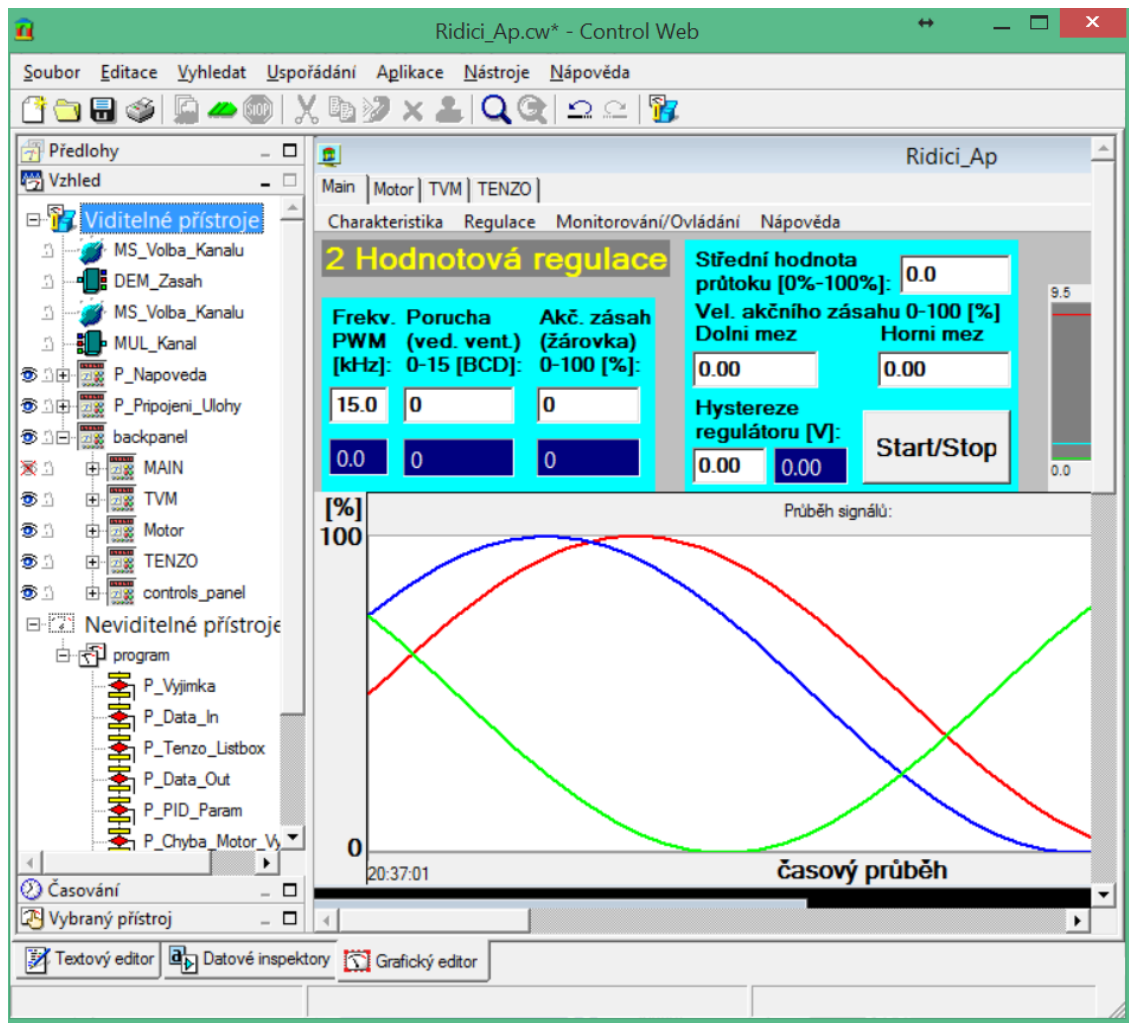
pojmenování a tím celou aplikaci zpřehlednit. Následuje rozdělení datového elementu podle typu proměnné skalární, pole, buffer.



Obrázek 43 – Ukázka datového inspektoru

7.3 Grafický editor

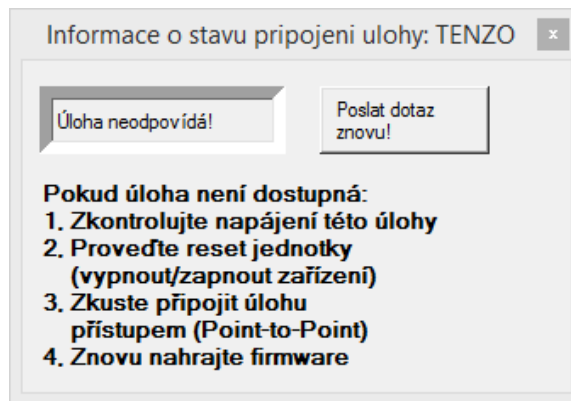
Editor lze rozdělit na pracovní plochu (hlavní panel tzv. background) a přehled vložených přístrojů z hlediska vzhledu, časování, vybraný přístroj a předlohy. Vzhled přístroje je dále rozdělen na neviditelné (programy,...) a viditelné (panely, přístroje,...). V oblasti časování opět na časované přístroje a nečasované (tj. událostně řízené). Přehled příjemců či řízených přístrojů daným přístrojem a jiné důležité vlastností lze zobrazit v záložce vybraný přístroj. Vkládání objektů (přístrojů, programů, časovačů, apod.) se dělá pomocí palety přístrojů pomocí přetažení na pracovní plochu (viditelné přístroje) či přetažením do neviditelných přístrojů (programy, časovače, atd.).



Obrázek 44 – Ukázka grafického editoru

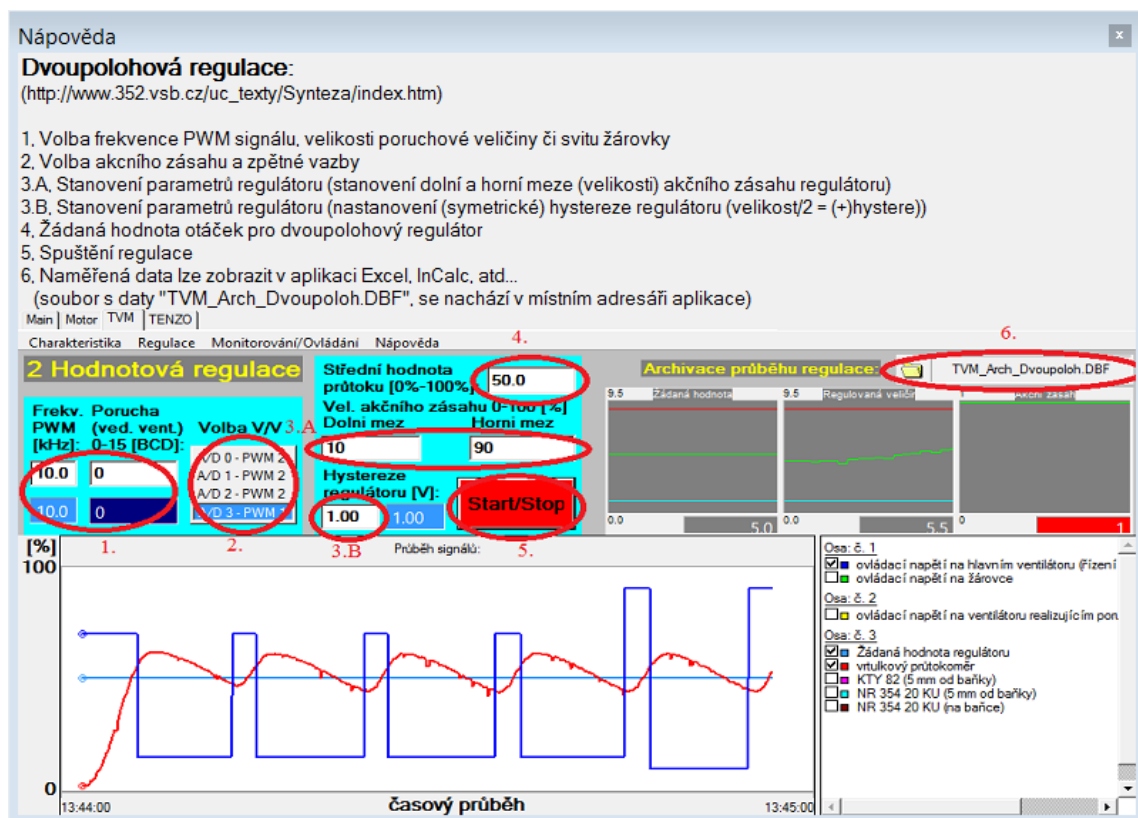
8 Řídicí aplikace v prostředí Control Web

V řídicí aplikaci je vytvořen příjem a odeslání dat se zmíněným zabezpečením viz kapitola věnovaná komunikaci (realizace alg. v MCU). Pro ověření, zdali je dotyčná laboratorní úloha správně připojena k síti, dochází při každém přepínání oken ke kontrole jejich stavu. Řídicí stanice pošle požadavek na zvolenou řídicí jednotku, která při přijetí výzvy odpoví. Po přijetí paketu v řídicím systému proběhne kontrola příchozího paketu, pokud je vše v pořádku, dochází k přepnutí okna. V opačném k přepnutí nedojde, ale uživatel má možnost zaslat jej opětovně pomocí zobrazeného okna. Dokud uživatel neodstraní příčinu, není možné se na konkrétní panel s laboratorní úlohou přepnout.



Obrázek 45 - Opětné připojení úlohy (úloha neodpověděla)

GUI aplikace umožňuje velmi snadno a rychle přepínat mezi jednotlivými úlohami. Záložky se specifickými názvy jsou po celou dobu zobrazeny v horním levém rohu aplikace. Každá úloha dále obsahuje menu s nápovědou programu, kde se nachází odkaz na teoretické podklady (katedrální stránky) k vybranému tématu spolu s ukázkou postupu měření/regulace.

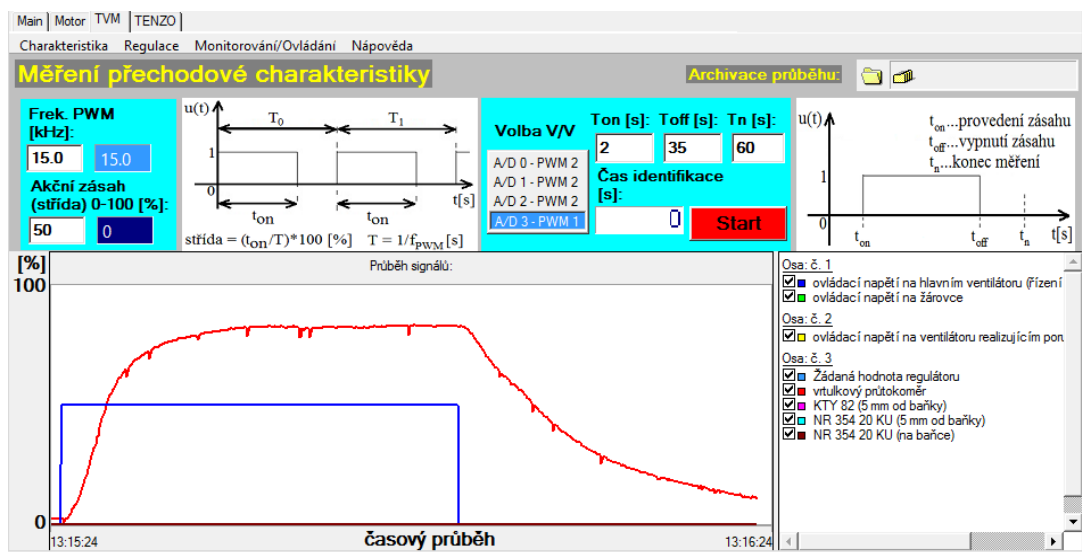


Obrázek 46 – Ukázka nápovědy programu

V aplikaci je možné provádět měření charakteristik přechodové, frekvenční, statické a tím určit dynamické a statické vlastnosti sledovaného systému. Z naměřených dat určit přenos soustavy a vypočítat např. parametry PID regulátoru. Pro vybrané úlohy má

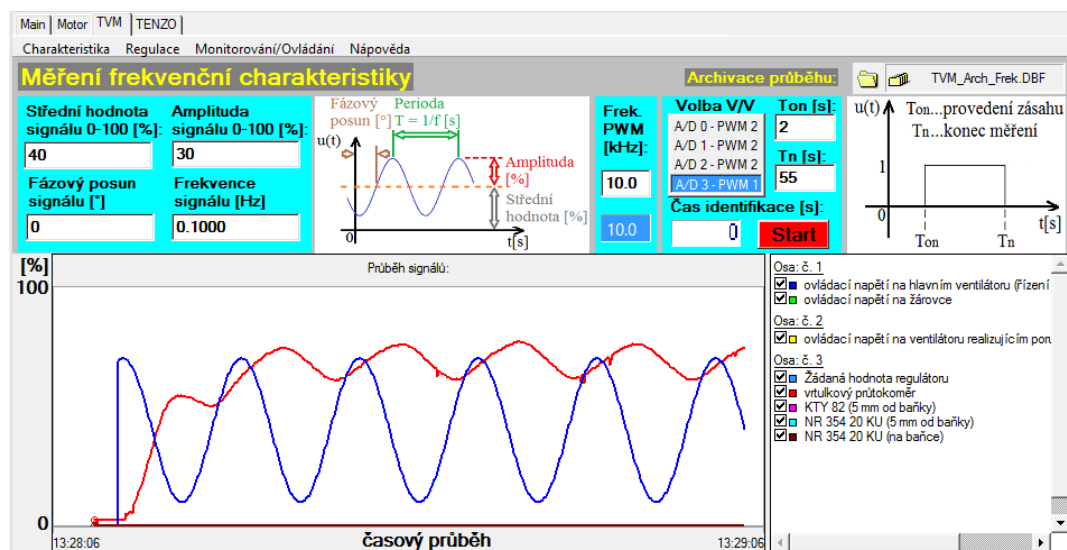
k dispozici PID regulátor, 2 hodnotovou regulaci či PID regulátor implementovaný v samotné řídicí jednotce. Pro seznámení s dotyčnou úlohou je možné daný model ovládat. Vzniklé průběhy jsou uživateli ihned zobrazeny pomocí komponenty „Data Viewer“, spolu s jejich automatickou archivací (kromě režimu ovládání či ručně naměřených dat v případě laboratorní úlohy TENZO).

V oblasti identifikace modelu může uživatel měnit základní frekvenci řídicího signálu PWM, spolu s jeho velikostí a délkou jeho trvání (přechodová charakteristika).



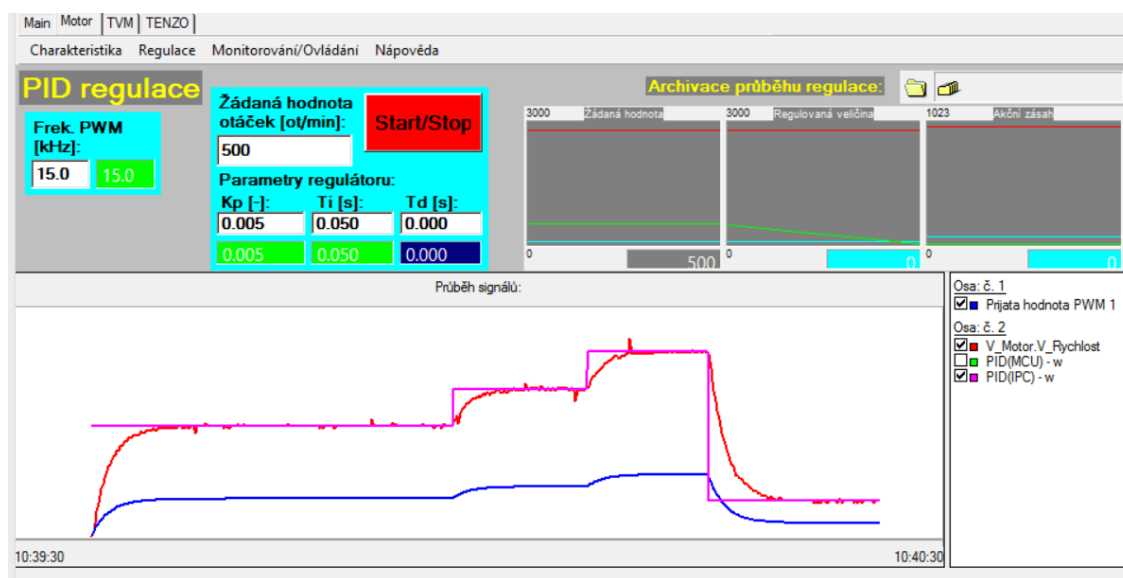
Obrázek 47 – Ukázka identifikace systému (TVM)

Pro identifikaci ve frekvenční oblasti je možné kromě uvedených společných rysů upravovat následující vlastnosti vstupního signálu, a to amplitudu, střední hodnotu, fázový posun a frekvenci.



Obrázek 48 – Ukázka identifikace systému ve frekvenční oblasti

Algoritmus PID (PSD) regulace realizovaný v MCU je přírůstkový (rekurzivní). V průběhu regulace lze měnit jeho parametry a žádanou hodnotu, či v případě stejnosměrného motoru i nosnou frekvenci PWM signálu. Vzorkovací perioda regulátoru je stanovena na 5 ms, kdy MCU stačí vypočítat novou hodnotu akčního zásahu (provést ji) a odeslat vypočtené a změřené údaje do IPC. Ve zbylém čase cyklicky ověřuje, zdali nedošla od IPC nová data (změna parametru, žádané hodnoty, vypnutí/zapnutí regulátoru,...).



Obrázek 49 – Ukázka vzhledu aplikace (PID regulace v MCU s různými parametry regulátoru)

V případě dvoupolohové a PID regulace z nadřazeného systému, byl dosažen velmi dobrý průběh regulace až při optimalizaci komunikace mezi MCU a IPC. Kdy po přijetí informace z MCU (zpětné vazby) dochází k okamžitému odeslání dat z jednotky Master (IPC) do příslušné jednotky Slave (MCU), pouze v případě, že se požadovaná hodnota liší od naposledy přijaté. Vzhledem k velmi nízké vzorkovací periodě, kterou není řídicí aplikace schopná dodržet. Bylo zapotřebí zajistit pevné vzorkování v nižší úrovni. V případě úlohy DC motor je přímo v jednotce MCU vytvořena dočasná paměť, která uchovává aktuální rychlost dosaženou v časovém úseku 5 ms, k této hodnotě může přistoupit IPC tzv. požadavkem (zasláním dotazu o aktuálním stavu otáček).

9 ZÁVĚR

Cílem práce bylo implementovat MCU pro monitorování a řízení technologických procesů. Pro jejich využití jsem navrhl a vytvořil SCADA/HMI systém, který umožňuje zmíněné úlohy řídit/monitorovat/identifikovat. Vybraný MCU jsem zasadil do vrstvy instrumentace/přímého řízení, pro přímý kontakt s technologickým procesem za účelem sběru dat nebo jeho samotného řízení.

Pro již existující vybrané laboratorní úlohy TENZO, DC Motor a TVM, které byly vhodné pro jejich možné nasazení do SCADA/HMI systému a to z hlediska monitorování či řízení. Zmíněné laboratorní úlohy jsem teoreticky popsal a pro stávající systém vytvořil nový způsob jejich využití upravením nebo rozšířením.

Řízení laboratorní úlohy teplovzdušného modelu jsem upravil pro jedno rozměrové řízení dle otáček či teploty na žárovce s možností přepínání zpětnovazebního signálu mezi třemi termistory. Vytvořený systém umožňuje identifikaci modelu v časové i frekvenční oblasti s nastavení požadované velikosti akčního zásahu a jeho frekvence pro žárovku, a hlavní ventilátor.

Laboratorní úlohu stejnosměrný motor jsem rozšířil o možnost diskrétního řízení z řídicí jednotky (nižší úrovně) nebo z řídicího systému (vyšší úrovně). Spolu s možností identifikace systému diskrétním signálem PWM v časové a frekvenční oblasti.

Model pro měření průhybu nosníku jsem doplnil o napájení můstku stabilizovaným napětím z řídicího MCU. Napětí na můstku je možné sledovat online, kdy naměřené hodnoty spolu se zapsaným průhybem mohou být vykresleny do grafu a následně uloženy do souboru pro další zpracování.

Vytvořil jsem SCADA/HMI systém, ve kterém jsem jakožto technologický proces použil již stávající laboratorní úlohy. Jakožto instrumentaci/přímé řízení jsem využil MCU, spolu s modifikovanou částí (rozšiřující deskou) pro úpravu signálu na potřebnou úroveň. Jako komunikační rozhraní mezi jednotlivými řídicími jednotkami a řídicím systémem jsem zvolil RS232, která svými vlastnostmi zcela vyhověla daným požadavkům o možnost připojení zařízení point-to-point či rozšířením na průmyslovou sběrnici. Řídicí systém jsem zprovoznil na stolním počítači standardu PC s operačním systémem Windows 8.

Pro implementaci jednotlivých úloh jsem vytvořil tři rozšiřující desky, které upravují vstupující a vystupující signály ze soustavy do řídicí jednotky na stanovené úrovni. Každá rozšiřující deska je vytvořena jako protikus řídicí desce (úspora místa), kdy

základní deska MCU sdílí napájení s rozšiřující deskou a v případě přímého zapojení (point-to-point) je tato soustava napájena z přípravku Arduino (USB portu).

Rozšíření stávajícího rozhraní RS232 na průmyslové RS422/RS485 resp. vytvoření komunikační sběrnice pro SCADA/HMI systém jsem dosáhl vytvořením univerzální komunikační karty pro jednotku Master i Slave. Komunikační karta též obsahuje možnost externího napájení, kdy distribuuje své napájení pro řídicí jednotku a její rozšiřující desku. Komunikační desku lze provozovat ve dvou průmyslových standardech a módech (Master, Slave), pro přehlednost a rychlé nastavení jsem komunikační desku rozšířil o nápovědu (zapojení jednotlivých propojek).

Jednotlivé panely řídicího systému jsem pro přehlednost doplnil o nápovědu programu, kde je vysvětlen celý postup pro úspěšné zvládnutí měření/identifikace/regulace. Popis jsem doplnil o odkazy na katedrální stránky se související teorií k danému tématu, včetně ukázky naměřeného průběhu a možnostmi se zobrazením výstupních dat ze systému.

Během návrh vyhodnocení otáček ze stejnosměrného motoru jsem se seznámil s kvadrurním čítačem a jeho funkcí od firmy HP HCTL-2000(2016). Kdy pro vyhodnocení dvou vzájemně posunutých signálů z IRC snímače bylo nutné zajistit jejich bezchybné čtení (odstranit zákmity při rozjedu motoru). Pro tento rozšiřující obvod jsem navrhnul rozšiřující desku, kde jako zdroj hodin jsem zvolil krystalový oscilátor. Nadefinoval jsem řídicí a datové piny a seznámil se s řídicím algoritmem pro vyčtení a reset vnitřní logiky.

- **Poděkování**

- *Na tomto místě bych rád poděkoval svému vedoucímu práce Ing. Jaromíru Škutovi, Ph.D. za poskytnutí tématu a možnost vytvářet ho pod jeho vedením. Dále bych rád poděkoval Ing. Miroslavu Mahdalovi, Ph.D. za věcné rady a připomínky.*

Seznam použité literatury

ATMEL. 2013. *8-bit Atmel Microcontroller with 16K/32K/64K Bytes In-System Programmable Flash: ATmega164P/V ATmega324P/V ATmega644P/V* [online]. [cit. 2014-11-09]. Dostupné z: http://www.atmel.com/Images/Atmel-8011-8-bit-AVR-Microcontroller-ATmega164P-324P-644P_datasheet.pdf

ARM. 2010. Cortex-M series processors [online]. [cit. 2014-10-05]. Dostupné z: <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0552a/index.html>

BALÁTĚ, J. *Automatické řízení*. Praha: Nakladatelství BEN, 2003, 654 s. ISBN 80-7300-020-2.

BOYER, S. A. 1999. *SCADA: Supervisory control and data acquisition, 2nd edition*. NEW YORK (USA): ISA, 1999. 215 P. ISBN 1-55617-660-0.

DUDÁČEK, K. Sériová rozhraní SPI, Microwire, I2C a CAN [online]. 2002 [cit. 2014-11-13]. Dostupné z: http://home.zcu.cz/~dudacek/NMS/Seriova_rozhrani.pdf

ECKHARD, Kamil. Univerzální sériová sběrnice. [online]. 2001 [cit. 2014-11-13]. Dostupné z: <http://home.zcu.cz/~eckhardt/popis.html>

Elektronická podpora výuky pravděpodobnosti a statistiky. In: Pantech Solutions [online]. 2013 [cit. 2014-10-11]. Dostupné z: <https://www.pantechsolutions.net/microcontroller-boards/getting-started-with-pic16f-architecture>

Elektrorevue: Sběrnice CAN. POLÁK, Karel. VUT FEKT BRNO. Elektrorevue [online]. 2003 [cit. 2014-11-13]. Dostupné z: <http://www.elektrorevue.cz/clanky/03021/index.html>

HUANG, Han-Way. PIC microcontroller: an introduction to software. Clifton Park [u.a.]: Delmar Thompson Learning, 2004. ISBN 14-018-3967-3.

I2C BUS COMMUNICATION. Prashant Jaiswar [online]. 2013 [cit. 2014-10-11]. Dostupné z: <http://www.prashantjaiswar.com/uncategorized/i2c-bus-communication/>

JANEČEK, J. 1993. *Distribuované systémy*. Praha : Vydavatelství ČVUT, 1993.
MICROCHIP TECHNOLOGY INC. Serial Peripheral Interface (SPI) [online]. 2011 [cit. 2014-11-24]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/70206D.pdf>

MICROCHIP. PIC16F87XA Datasheet [online]. 2013 [cit. 2014-10-11]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/39582C.pdf>

MORAVSKÉ PŘÍSTROJE A.S. *Moravské přístroje* [online]. 2012 [cit. 2015-05-13]. Dostupné z: <http://www.mii.cz/>

POSITAL GMBH. IMPLEMENTATION OF SSI MASTER INTERFACE [online]. 2013 [cit. 2014-11-15]. Dostupné z: https://www.posital.com/media/posital_media/documents/AbsoluteEncoders_Context_Technology_SSI_AppNote.pdf

Ščevík, P. Zpracování výukových textů z oblasti číslicové regulace v prostředí Internetu. Ostrava: katedra automatizační techniky a řízení, VŠB-TU Ostrava, 2008. 116 stran. Diplomová práce, vedoucí: Vítečková, M.

TEXAS INSTRUMENTS INCORPORATED. Stellaris® LM4F120H5QR Microcontroller [online]. 2013 [cit. 2014-10-05]. Dostupné z: <http://www.mouser.com/ds/2/405/lm4f120h5qr-124014.pdf>

VACEK, V. *Učebnice programování PIC*. 1. vyd. Praha: BEN - technická literatura, 2000, 143 s. ISBN 80-860-5687-2.

VLACH, J. *Počítačová rozhraní, přenos dat a řídicí systémy*. Praha, BEN-technická literatura, 1997, ISBN 80-85940-17-4.

Seznam obrázků

Obrázek 1 – Blokové schéma TVM s řídicí jednotkou a PC [SMUTNÝ, ŠKUTA, 2005].....	12
Obrázek 2 – Blokové schéma pro TVM (Point-to-Point).....	14
Obrázek 3 – Rozšiřující deska pro TVM	14
Obrázek 4 – Rozložení součástek rozšiřující desky pro TVM	15
Obrázek 5 – Ukázka průběhu identifikace motoru [Výukový systém rc2000 – uLab] ..	16
Obrázek 6 – Zapojení motoru s PI regulátorem [Výukový systém rc2000 – uLab].....	16
Obrázek 7 – Blokové schéma rozšiřující desky pro DC Motor	17
Obrázek 8 – Rozšiřující deska pro DC Motor	18
Obrázek 9 – Rozložení součástek rozšiřující desky pro DC Motor.....	18
Obrázek 10 – Gunt Fl 102 [FL 102, 2015]	19
Obrázek 11 – Zapojení tenzometrů (průhyb nosníku) [FL 102, 2015].....	20
Obrázek 12 – Blokové schéma rozšiřující desky pro TENZO	20
Obrázek 13 – Rozšiřující deska pro TENZO	21
Obrázek 14 – Rozložení součástek rozšiřující desky pro TENZO	21
Obrázek 15 – Architektura PIC16F873A [MICROCHIP, 2013]	23
Obrázek 16 – Blokové schéma SPI [MICROCHIP, 2013].....	25
Obrázek 17 – Blokové schéma I ² C [MICROCHIP, 2013]	26
Obrázek 18 – Registr CCPRxCON [MICROCHIP, 2013].....	27
Obrázek 19 – Blokové schéma popisující princip PWM modulu [MICROCHIP, 2013].....	28
Obrázek 20 – Časový průběh signálu PWM [MICROCHIP, 2013].....	28
Obrázek 21 – Základní deska pro MCU	31
Obrázek 22 – Rozložení součástek MCU	31
Obrázek 23 – Průběh přenosu 1 bytu (1 start bit, 8 bitů data, 1 stop bit, bez parity)	32
Obrázek 24 – Ukázka komunikace Master-Slave [Atmel Microcontroller, 2013].....	34
Obrázek 25 – Vzorkování dat (CPOL, CPHA) [Dudáček, 2002].....	35
Obrázek 26 – Koncepce SPI Master-Slave [MICROCHIP, 2011].....	35
Obrázek 27 – Koncepce SPI Master-Slave 3x [MICROCHIP, 2011]	36
Obrázek 28 – Zjednodušené blokové schéma SSI [POSITAL GMBH, 2013].....	36
Obrázek 29 – Ukázka komunikace SSI [POSITAL GMBH, 2013]	37
Obrázek 30 – Koncepce I2C sběrnice [HUANG, 2004]	38

Obrázek 31 – Komunikační rámec I ² C [Prashant Jaiswar, 2013].....	39
Obrázek 32 – Blokové schéma řídicího systému.....	42
Obrázek 33 – Arduino USB [USB Serial Light Adapter, 2015]	43
Obrázek 34 – Komunikační deska Master/Slave.....	43
Obrázek 35 – Rozložení součástek komunikační desky Master/Slave.....	44
Obrázek 36 – Ukázka obousměrné komunikace při rychlosti 256 Kb/s (RS232(TTL)<->RS422<->RS232(TTL))	45
Obrázek 37 – Ukázka (přiblížení) obousměrné komunikace při rychlosti 256Kb/s (RS232(TTL)<->RS422<->RS232(TTL)).....	45
Obrázek 38 – Ukázka programovacího prostředí	46
Obrázek 39 – Terminál v prostředí mikorC.....	47
Obrázek 40 – Zabezpečení paketu	51
Obrázek 41 – Ukázka vývojového prostředí Control Web.....	55
Obrázek 42 – Ukázka textového editoru.....	56
Obrázek 43 – Ukázka datového inspektoru	57
Obrázek 44 – Ukázka grafického editoru	58
Obrázek 45 - Opětovné připojení úlohy (úloha neodpověděla).....	59
Obrázek 46 – Ukázka nápovědy programu.....	59
Obrázek 47 – Ukázka identifikace systému (TVM)	60
Obrázek 48 – Ukázka identifikace systému ve frekvenční oblasti	60
Obrázek 49 – Ukázka vzhledu aplikace (PID regulace v MCU s různými parametry regulátoru)	61

Seznam tabulek

Tabulka 1 – Přehled I/O kanálů CTRL V3 [SMUTNÝ, ŠKUTA, 2005]	13
Tabulka 2 – Vstupní/výstupní kanály rozšiřující desky pro TVM	13
Tabulka 3 – Seznam použitých součástek pro rozšiřující desku TVM.....	15
Tabulka 4 – Seznam použitých součástek pro rozšiřující desku DC Motor	19
Tabulka 5 – Seznam použitých součástek pro rozšiřující desku TENZO	22
Tabulka 6 – Parametry procesoru PIC16F873A [MICROCHIP, 2013].....	24
Tabulka 7 – Seznam použitých součástek pro základní desku MCU	30
Tabulka 8 – Parametry komunikace USART modulu	49
Tabulka 9 – Ukázka komunikačního paketu Rx	50
Tabulka 10 – Ukázka komunikačního paketu Tx	51

Přílohy

1 Obsah přiloženého CD

Na přiloženém CD se v kořenovém adresáři nachází tato diplomová práce ve formátu *diplomová_prace.pdf* spolu s programy uvedenými v diplomové práci.

- Rozblikání LED.c
- Rozsvícení LED pomocí PWM.c
- Komunikace po USART.c
- AD převod analogového signálu.c

2 Zlín - STOČ 2015

3 Kraków - 52 Konferencja Studenckich Kół Naukowych



 Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

uděluje

DIPLOM

za 3. místo

Jiřímu Czebe

v soutěži **STUDENTSKÁ TVŮRČÍ A ODBORNÁ ČINNOST**

v sekci **S4 HW A SW APLIKACE**



prof. Ing. Vladimír Vašek, CSc.
předseda sekce

23. dubna 2015



doc. Mgr. Milan Adámek, Ph.D.
děkan FAI



AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE

DYPLOM UCZESTNICTWA

W 52 KONFERENCJI
STUDENCKICH KÓŁ NAUKOWYCH
PIONU HUTNICZEGO

Sekcja Automatyki i Robotyki
podsekcja 1

Jiří Czebe

Temat referatu:
USAGE OF SINGLE-CHIP COMPUTERS FOR DATA COLLECTION AND CONTROL

Przewodniczący Jury

Pełnomocnik Rektora
ds. Kół Naukowych

dr inż. Leszek Kurcz

Kraków 2015