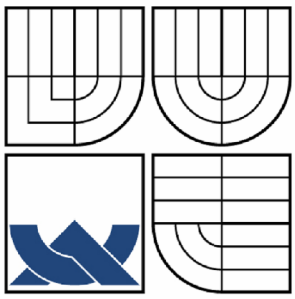
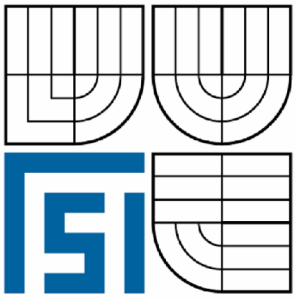


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

FACULTY OF MECHANICAL ENGINEERING

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

MULTIFUNKČNÍ ZÁZNAMOVÁ, MĚŘICÍ A ŘÍDICÍ JEDNOTKA

MULTIFUNCTION DATA ACQUISITION, MEASUREMENT AND CONTROL DEVICE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — IVO TRÁVNÍČEK

AUTHOR

VEDOUCÍ PRÁCE — Ing. ROBERT GREPL, Ph.D.

SUPERVISOR

BRNO 2010

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav mechaniky těles, mechatroniky a biomechaniky
Akademický rok: 2009/2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Ivo Trávníček

který/která studuje v **bakalářském studijním programu**

obor: **Mechatronika (3906R001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Multifunkční záznamová, měřicí a řídicí jednotka

v anglickém jazyce:

Multifunction data acquisition, measurement and control device

Stručná charakteristika problematiky úkolu:

Práce se bude zabývat návrhem, výrobou a testováním multifunkčního elektronického zařízení s následujícími vlastnostmi:
- digitální a analogové I/O
- řídicí mikrokontrolér
- předprogramované komponenty konfigurovatelné přes GUI z PC (ukládání dat do paměti, komunikace přes sériové rozhraní a regulace).

Cíle bakalářské práce:

1) Navrhněte strukturu HW s mikrokontrolérem Atmel AVR podle požadavků školitele (zejména specifikace počtu a typu I/O, specifikace provedení).
2) Vytvořte HW a SW prototypu s implementovanými základními vlastnostmi. Testujte vlastnosti vytvořeného prototypu.
3) Vytvořte SW s GUI pro konfiguraci na PC. Konfigurace zařízení je prováděna přes sériové rozhraní pomocí definovaných příkazů. Záznam dat nebo čtení dat z paměti lze provádět v reálném čase nebo dávkově.
4) Navrhněte DPS pro finální dvě varianty multifunkční jednotky.
5) Na několika konkrétních příkladech demonstrujte a detailně popište vlastnosti vyvinutého produktu (dlouhodobé měření analogových signálů, řízení DC motoru apod.).

Seznam odborné literatury:

- Dušek, F.: Matlab a Simulink, skriptum ČVUT
- Mann, B.: C pro mikrokontroléry, Nakladatelství BEN, 2003
- Váňa, V.: Mikrokontroléry ATMEL AVR - assembler, Nakladatelství BEN, 2003
- Herout, P.: Učebnice jazyka C

Vedoucí bakalářské práce: Ing. Robert Grepl, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2009/2010.

V Brně, dne 26.10.2009

L.S.

prof. Ing. Jindřich Petruška, CSc. Ředitel ústavu	doc. RNDr. Miroslav Doupovec, CSc. Děkan fakulty

Abstrakt
Práce se zabývá návrhem a testováním Multifunkční záznamové, měřicí a řídicí jednotky. Jednotka je založena na mikrokontroléru ATmega, který byl programován v jazyce C. Jednotka obsahuje funkce pro měření fyzikálních veličin, filtrování naměřených dat, regulaci dynamických soustav a komunikaci s PC. Konfigurace jednotky lze provádět v reálném čase ve speciálním software vytvořeném v jazyce Matlab nebo pomocí terminálu. Lze použít např. pro řízení DC motoru PID regulátorem, dlouhodobé snímání teploty nebo měření zrychlení pomocí akcelerometru.

Klíčová slova
MFU168, programátor, ASPUSB, ATmega, SD karta, microSD karta, PID, RS232, jazyk C, jazyk Matlab, Matlab GUI, Kalmanův filtr, akcelerometr, H-můstek, škrticí klapka

Abstract
The thesis deals with and tests the Multifunction data acquisition, measurement and control device. The unit is based on the microcontroller ATmega, which was programmed in the language C. The unit contains functions for the measurement of physical quantify, filtering record, regulation of the dynamic systems and communication with PC. Configuration of the unit is real-time in special software created in the language Matlab or by a terminal. The purpose of the unit is controlling a DC motor by the PID regulator, long-term measurement of temperature or measurement of acceleration by an accelerometer.

Keywords
MFU168, programmer, ASPUSB, ATmega, SD card, microSD card, PID, RS232, language C, language Matlab, Matlab GUI, Kalman filter, accelerometer, H-bridge, throttle

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce Ing. Robertu Greplovi Ph.D za odborné vedení, zapůjčení nástrojů potřebných k realizaci mé práce a trpělivost. Dále bych rád poděkoval rodině a přítelkyni za trpělivost a podporu při mé tvůrčí činnosti.

Čestné prohlášení

Prohlašuji, že jsem celou práci, včetně příloh, vypracoval samostatně a uvedl jsem všechny použité podklady a literaturu.

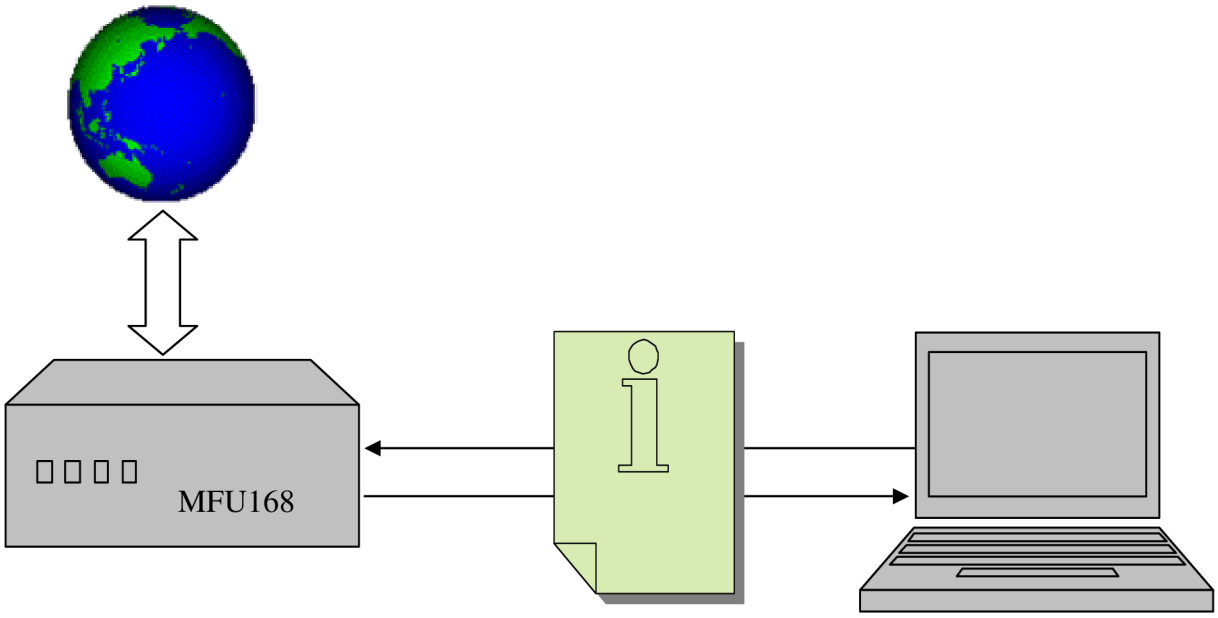
Trávníček Ivo, Brno, 2010

Obsah

1 Úvod

V moderní době je stále více využívána automatizace a inteligentní ovládání objektů počítačem. Spojení počítače a reálného objektu (např. motoru, senzoru,...) však není zcela triviální a je potřeba odborných znalostí a zařízení k uskutečnění námi požadovaného cíle. Touto problematikou se zabývá mnoho předmětů na Vysokém učení technickém v Brně. Jsou zde specificky vybavené laboratoře s odborným personálem. Pro výuku je jistě velmi přínosná ukázka měření a řízení reálných objektů. Tato zařízení a vybavení jsou v laboratořích, ale nelze je jednoduše předvádět na přednáškách nebo prezentacích, protože jejich přenositelnost je omezená. Jsou to často zařízení funkčně vázaná na laboratoř, příliš těžká, drahá nebo nepřenositelná z jiných důvodů.

Cílem této práce je navrhnout a realizovat přenosné zařízení (Multi Function Unit, zkráceně MFU168), které bude možné v těchto situacích použít, nebude vázané na laboratoř, bude snadno přenositelné, levné a použitelné pro výukové a prezentační účely. Použití MFU168 bude také v jiných projektech nebo v případě zájmu i komerční. Ovládání MFU168 bude probíhat přes PC. Pro snadné nastavení a zobrazení naměřených hodnot bude vytvořeno uživatelské rozhraní v prostředí MATLAB. Finální verze MFU168 budou dvě, *Řídicí a měřicí jednotka* a *Miniaturní měřicí jednotka*. Obě budou obsahovat stejný software, ale budou se lišit v provedení DPS (deska plošných spojů). *Řídicí a měřicí jednotka* je určena pro vývoj nových aplikací nebo pro výukové a prezentační účely. *Miniaturní měřicí jednotka* bude omezena velikostí DPS, je určena pro autonomní dlouhodobé snímání teploty a/nebo vibrací strojů a zařízení.



Obr. 1 Naivní blokové schéma MFU168

2 Formulace problému a cíle řešení

2.1 Obecný popis

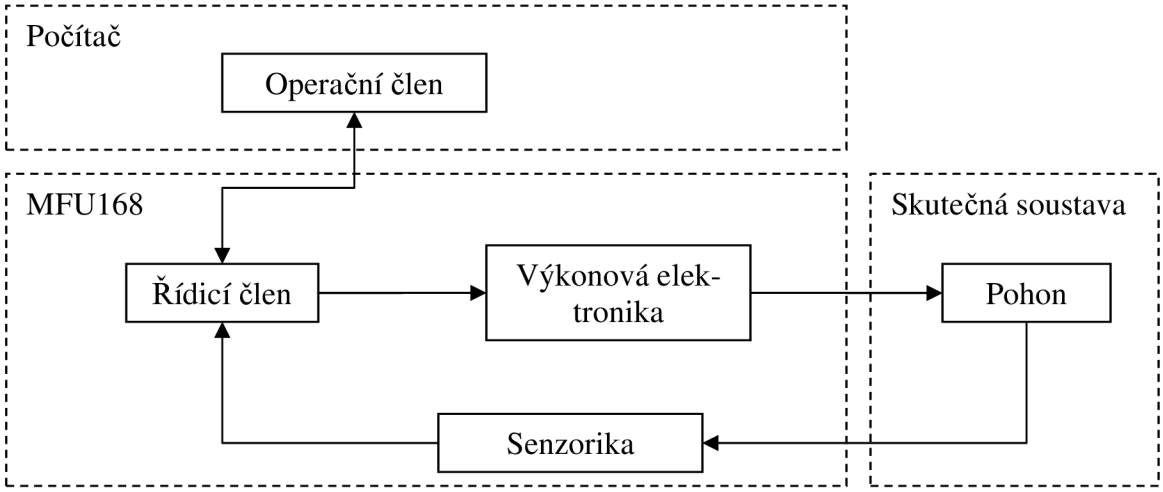
Jedná se o zařízení (Multi Function Unit, zkráceně MFU168), které funguje podobně jako vstupně/výstupní karta v PC. Cílem je měřit analogové a digitální signály (např. teplotu v místnosti) a tyto naměřené informace posílat do PC přes RS232 (sériová linka počítače). V PC se informace zpracují a přehledně zobrazí v uživatelském rozhraní. Předávání informací bude fungovat obousměrně. Na PC lze nastavit nebo zapnout řadu funkcí, které MFU168 podporuje. Příkladem jsou řídicí smyčky regulátoru, filtry nebo nastavení periferií připojených k MFU168. Výsledkem bude naprogramovaný software v PC, vytvořené DPS *Řídicí a měřicí jednotky* a naprogramovaný firmware.

Pro specifické použití bude vytvořena úprava této jednotky - *Miniaturní měřicí jednotka.* Bude používat totožný software i firmware, ale velikost DPS bude minimalizována (počítá se s použitím ve stísněných prostorech, např. rotor motoru). Bude sloužit pro dlouhodobé měření (teploty, vibrací strojů,...). Bude zcela autonomní, pro měření bude nutný pouze zdroj energie.

2.2 Příklady aplikací

2.2.1 Rychlostní nebo polohové řízení pohonu

Jednou ze zadaných aplikací je řízení DC motoru a to polohově nebo na určitou rychlost otáčení. Řízení servopohonu lze schématicky rozdělit na senzoriku, výkonovou elektroniku, řídicí člen (regulátor), operační člen a vlastní pohon. Toto rozdělení je znázorněno na Obr. 2.



Obr. 2 schéma DC motoru s regulátorem

Jednotlivé dílčí komponenty přesněji specifikujeme v následujícím výčtu:

Senzorika - typy měřených veličin

- natočení, posuv
- úhlová rychlost natočení, rychlost posuvu
- teplota
- proud
- napětí

Výkonová elektronika - výstupní výkonové zařízení

- Jednosměrné řízení výkonu
- Obousměrné řízení výkonu

Řídicí člen - vlastnosti regulátoru implementovaného v MFU168

- PID regulátor
- omezovač výstupního výkonu
- nastavení parametrů pomocí PC v uživatelském rozhraní
- zobrazení jednotlivých složek regulátoru na PC (výukové účely)

Operační člen - nastavení žádané polohy/rychlosti

- příkazem z PC
- ovladačem, kolečkem
- nastavením cyklu v MFU168 (skok, pila, sin,...)

2.2.2 Měření s ukládáním na SD kartu

Další možností využití je autonomní měřicí přístroj (*Miniaturní měřicí jednotka*), který bude konfigurovatelný přes PC. Po připojení k PC si budeme moci vybrat, kterou veličinu chceme měřit, s jakým vzorkováním, jaké senzory jsou připojené na vstupně/výstupní porty *Miniaturní měřicí jednotky*. *Miniaturní měřicí jednotka* bude založena na stejném hardware, firmware i software jako *Měřicí a řídící jednotka*, ale její fyzické provedení bude minimalizováno. Naměřené hodnoty budou ukládány na vestavěnou SD kartu. Po připojení *Miniaturní měřicí jednotky* k PC se naměřená data odešlou do PC, kde je lze dále zpracovávat.

Typy měřených veličin

- teplota
- zrychlení
- rychlost
- poloha

2.2.3 Měřicí jednotka - vizualizace měřených dat

Dalším využití *Měřicí a řídicí jednotky* bude podobné multifunkční měřicí kartě v PC. MFU168 bude připojená komunikačním kabelem s PC. Na PC budeme v uživatelském rozhraní provádět různá nastavení, vybírat připojené snímače, velikost vzorkování, filtry a citlivost měření. Naměřené hodnoty se budou posílat přes komunikační linku do PC, kde je můžeme sledovat v uživatelském rozhraní a dále je zpracovávat.

3 Analýza problému

3.1 Řízení pohonu

Pro řízení pohonu je nutné blíže specifikovat jednotlivé funkční bloky, které byly znázorněny na Obr. 2. V následujících podkapitolách podrobněji rozebereme použitelné součásti. Z rozboru dále vyplynou požadavky na MFU168, které musí být splněny pro testování na reálných úlohách.

3.1.1 Senzory pro měření regulované veličiny (Senzorika)

MFU168 bude postavena na bázi jednočipu. Ten může měřit pouze napětí, a to digitálně nebo analogově. Z tohoto důvodu je třeba všechny veličiny potřebné pro řízení pohonu převést na napětí měřitelné jednočipem.
Měření fyzikálních veličin uvedených v kap. 2.2.1 lze realizovat následujícími způsoby:

Měření polohy - potenciometr
Funguje jako napěťový dělič, po připojení na napájecí napětí MFU168 jej lze připojit přímo k analogovému vstupu jednočipu. Výstupní hodnota potenciometru udává jeho natočení (rotační) nebo posuv (lineární). Rozsah, proudový odběr nebo krajní polohy/dorazy se volí vhodným výběrem potenciometru. Při derivaci polohy podle času lze použít i k měření rychlosti.

Výhody:

- velmi levný
- výstupní napětí lze snímat přímo jednočipem bez použití převodníku
- při časové prodlevě v jednočipu nedochází ke ztrátě „správné polohy“

Nevýhody:

- trpí velkým šumem
- jedná se o mechaniku, která je velmi náchylná k poruchám

Požadavek na funkce MFU168:

- analogový napěťový vstup

Měření rychlosti - tachodynamo
Slouží pro měření úhlové rychlosti. Je to senzor, který funguje na podobném principu jako DC motor. Výstupní veličinou je napětí, které je úměrné rychlosti otáčení.

Výhody:

- výstupní napětí lze snímat přímo jednočipem nebo s použitím jednoduchého odporového děliče
- při časové prodlevě v jednočipu nedochází ke ztrátě „správné rychlosti“

Nevýhody:

- trpí velkým šumem
- jedná se o mechaniku, která je velmi náchylná k poruchám

Požadavek na funkce MFU168:

- analogový napěťový vstup

Měření polohy – inkrementální enkodér

Funguje na principu detekce kontrastních objektů, které jsou rozpoznávány a signalizují změnu polohy. Enkodéry jsou postaveny na různých fyzikálních veličinách, lze snímat změnu osvětlení (tmavé světlé pruhy), změnu elektrického pole anebo změnu magnetického pole. Důležitým faktem je, že výstupem jsou pulsy, udávající změnu polohy. Každý puls znamená posun o jeden krok. Při použití kvadraturního enkodéru lze měřit změnu polohy i směr otáčení. Derivací polohy podle času lze získat rychlost otáčení.

Výhody:

- malý šum
- digitální výstup lze připojit přímo k jednočipu

Nevýhody:

- při časové prodlevě jednočipu dochází ke ztrátě „správné polohy“
- zatížení jednočipu roste s počtem pulsů v čase

Požadavek na funkce MFU168:

- digitální vstupy pro kvadraturní enkodér
- rutina pro zpracování pulsů z enkodéru

Měření proudu

Princip funkce proudového snímače je nejčastěji měřením napětí na výkonovém odporu nebo hallovou sondou. Proudové senzory jsou většinou specializovanými integrovanými obvody s analogovým nebo digitálním výstupem. Některé výkonové moduly obsahují proudové senzory integrované v jednom pouzdře.

Požadavek na funkce MFU168:

- analogový napěťový vstup
- možnost rozšíření o digitální vstup se zpracováním protokolu proudového snímače

Měření napětí

Jednočip podporuje přímo měření analogových nebo digitálních napětí. Problémem může být rozsah, který se však dá upravit do správných mezí odporovým děličem nebo napěťovým zesilovačem.

Požadavek na funkce MFU168:

- analogový napěťový vstup
- digitální napěťový vstup

Měření teploty

Pro měření teploty je potřeba použít senzor teploty, který převede teplotu na napětí měřitelné jednočipem. Pro první fázi vývoje budeme počítat s jednoduchým analogovým teplotním senzorem. Vý-

stupem ze senzoru je napětí úměrné teplotě. Později lze přistoupit k sofistikovanějším a přesnějším senzorům, které však budou potřebovat širší podporu funkcí ze strany MFU168.

Požadavek na funkce MFU168:

- analogový napěťový vstup
- možnost rozšíření o další typy teplotních senzorů

3.1.2 Výstupní výkonové zařízení (Měnič)

Pro řízení motoru je potřeba zařízení (měnič), které převádí řídicí signál na výkon vstupující do soustavy. Řídicí výstupní signál z MFU168 má 5V a maximální proud je asi 10mA. To zdaleka nestačí pro chod motoru. Proto je třeba výstup posílit použitím výkonového prvku nebo integrovaného výkonového obvodu.

Jednosměrné řízení výkonu
Nejjednodušší řešení, lze použít jeden výkonový tranzistor nebo MOSFET.

Výhody:

- jednoduché zapojení
- málo součástek
- levné

Nevýhody:

- pouze jednostranný chod

Požadavek na funkce MFU168:

- digitální výstup
- PWM výstup

Obousměrné řízení výkonu
Pro řízení servopohonu je potřeba obousměrné řízení. To však nelze realizovat zapojením např. se dvěma tranzistory. Pro tento způsob řízení se používá tzv. můstkové zapojení. Při větších napětích lze sestavit z diskrétních součástek, avšak při nižším napětí je tato problematika značně složitá a nejlepší je použít nějaký hotový integrovaný obvod. Integrovaných obvodů existuje více typů a obsahují různé vstupy, je proto třeba v MFU168 uvažovat s přepínáním různých typů výkonových zařízení.

Výhody:

- oboustranné řízení

Nevýhody:

- složitější zapojení
- dražší
- problém s většími výkony

Požadavek na funkce MFU168:

- digitální výstupy
- možnost rozšíření pro různé typy

3.1.3 Regulátor implementovaný v MFU168 (Řídicí člen)

K řízení motoru je nutný regulátor, který zajišťuje dosažení a udržení námi požadovaného stavu. Pro řízení je důležitá rychlá odezva mezi senzory a výkonovou elektronikou. Při uvažování řídicí smyčky zahrnující spojení MFU168 s počítačem, který disponuje dostatečným výkonem, by se však projevily různé problémy s komunikací, buffery nebo dopravní zpoždění. Z těchto důvodů je řízení vhodné realizovat přímo v řídícím jednočipu – MFU168. Časově kritické řídicí algoritmy jsou tedy implementovány přímo do MFU168, ale protože bude jednočip omezen svým malým výpočetním výkonem, je dobré počítat i s přesunutím náročnějších operací, které nejsou kritické na čas, do osobního počítače. Proto zde vzniká požadavek na širokou komunikaci MFU168 s počítačem, možnosti nastavení různých parametrů a odesílání naměřených hodnot do počítače, kde je lze podrobněji analyzovat sofistikovanějšími nástroji. Po analýze dat v PC můžeme cíleně upravit parametry regulátoru v MFU168.

Požadavek na funkce MFU168:

- PID regulátor
- omezovač výstupního výkonu
- nastavení parametrů regulátoru z PC
- zobrazení jednotlivých složek regulátoru na PC (výukové účely)

3.1.4 Nastavení žádaného natočení/rychlosti

Při tvorbě servopohonu potřebujeme ovládat natočení nebo rychlost otáčení (podle typu regulace). Předání informace lze uskutečnit různými způsoby.

Manuálním ovladačem
Otáčením kolečka nebo posuvným ovladačem budeme měnit žádanou hodnotu natočení/rychlosti otáčení. Nejjednodušší realizace je použití otočného nebo posuvného potenciometru.

Požadavek na funkce MFU168:

- analogový napěťový vstup

Příkazem z PC
Na PC budeme dostupnými ovládacími prvky (klávesnice, myš,..) měnit žádanou hodnotu natočení/rychlosti otáčení. Informace bude odeslána do MFU168 pomocí komunikační linky.

Požadavek na funkce MFU168:

- komunikační linka
- rutina pro příjem požadované hodnoty

Nastavení cyklu v MFU168
Při variantě nastavování žádané hodnoty natočení/rychlosti otáčení pomocí PC můžeme narazit na problém s rychlostí komunikace a dopravním zpožděním. Pokud budeme chtít opakující se prů-

běh žádaných hodnot, bude zbytečné jej stále posílat po komunikační lince a vhodnější volbou bude nastavovat žádanou hodnotu natočení/rychlosti otáčení přímo v MFU168. Jako základní typy cyklů, které by měla MFU168 podporovat interně, můžou být skok, pila nebo harmonický cyklus.

Požadavek na funkce MFU168:

- komunikační linka
- funkce pro nastavení cyklů a jeho parametrů

3.2 Měření s ukládáním na SD kartu

Provedení *Miniaturní měřicí jednotky* je požadováno co možná nejmenší. Z toho plyne mnoho omezení. Jako záznamové médium bude použita MicroSD karta, která je v dnešní době nejmenší dostupnou velkokapacitní paměťovou kartou. Je levná a lze ji bez problémů přečíst běžně dostupnými čtečkami karet.

Kvůli univerzálnímu použití je třeba počítat s řadou vstupně/výstupních portů, ke kterým budeme připojovat senzory. Před samotným měřením se *Miniaturní měřicí jednotka* nastaví přes PC a toto nastavení se uloží na MicroSD kartu. Po zapnutí měření (autonomní provoz bez připojeného PC) bude konfigurace znovu načtena z MicroSD karty a spuštěno měření.

Podle pokynů vedoucího BP *Miniaturní měřicí jednotka* bude obsahovat vestavěný akcelerometr, tzn. že senzor bude součástí DPS.

Požadavek na funkce MFU168:

- slot na MicroSD kartu
- rutiny pro načtení/uložení konfigurace na MicroSD kartu
- vstupně/výstupní porty pro senzory a rutiny pro jejich ovládání a měření
- rutiny pro ukládání naměřených hodnot do MicroSD karty
- rutiny pro odesílání naměřených dat do PC
- akcelerometr součástí DPS *Miniaturní měřicí jednotky*

3.3 Měřicí jednotka - vizualizace měřených dat

Měření veličin

Pro měření budeme potřebovat různé typy vstupů. MFU168 je postavena na jednočipu, který může měřit pouze napětí, digitálně nebo analogově. Prvním typem podporovaných veličin je měření napětí. Při rozsahu napětí, které nebude odpovídat rozsahu jednočipu, bude třeba v případě většího napětí použít odporový dělič, nebo v případě menšího napětí zesilovač. Pro měření jiných fyzikálních veličin je nutné použít senzor, který převede fyzikální veličinu na napětí. Pro různé typy senzorů jsou zapotřebí různé vstupy v MFU168, které budeme moci přepínat a rutiny, které budou měřená data správně interpretovat. Některé typy senzorů mají digitální výstup, naměřená data jsou posílána pomocí vlastního protokolu. Protože protokoly senzorů se značně liší, bude zapotřebí, aby byly funkce MFU168 snadno rozšiřitelné a upravitelné. Tím zajistíme, že MFU168 nebude omezená pouze na úzký okruh aplikací.

Naměřená data a komunikace

Dejme tomu, že máme naměřená data. MFU168 bude snímat různé typy dat, s různou přesností a nelze je tedy ukládat a pracovat s nimi stejně. Musíme zajistit ovládání a rutiny, které budou manipulovat s daty podle jejich typu a správně je odesílat do PC. Na PC nastává stejný problém, že data

nebudou stejného formátu. MFU168 a PC se musí synchronizovat a předat si vzájemně nastavení, aby se zajistila správná interpretace všech naměřených dat.

V PC se počítá s uživatelským rozhraním, které bude disponovat příkazy, tlačítky a jinými ovládacími prvky, které usnadní a urychlí správné nastavení MFU168.

Ovládací rozhraní

Bylo již zmíněno, že se počítá s rozšiřitelností a úpravou funkcí v MFU168. Pro nové funkce by bylo nutné vždy upravovat i software na PC, aby je podporoval a umožnil jejich využití. Odbourání tohoto problému se dá částečně vyřešit adaptabilním uživatelským prostředím, které bude nabízet obecné funkce a po každém připojení MFU168 se synchronizuje. Funkce, které MFU168 již nenabízí, zruší a nové naopak přidá. Zajistíme tak rychlý vývoj a testování nových funkcí MFU168. Z všestrannosti vyplývají nedostatky. Prostředí bude oplývat mnoha nástroji, tlačítky a tím se stane nepřehledným a pro neodbornou obsluhu značně složitým. Je dobré tedy uvažovat i o speciálních uživatelských prostředích, které budou určeny pouze pro specifické úkoly. Jejich vzhled bude uzpůsoben dané úloze, budou obsahovat pouze nezbytné ovládací prvky a zobrazení naměřených dat bude uzpůsobeno danému typu úlohy. Po zapnutí se sami synchronizují s MFU168 a nebudou obtěžovat uživatele nastaveními, které se pro danou úlohu nemění.

4 Návrh řešení

4.1 Schéma zapojení a moduly

Při analýze problému v kap. 3 jsme vyjádřili požadavky, které by měla MFU168 splňovat. Je jich celá řada a tyto požadavky zasahují nejen do hardwaru, ale taky softwaru v MFU168 a v PC. Pro další postup bylo vytvořeno blokové schéma (Obr. 3), které rozdělí problém na dílčí části.

V dnešní době jde vývoj techniky stále kupředu, hardware se mění velice rychle a software ještě rychleji. Aby mohl tento projekt držet krok s vývojem, je nutné vytvořit flexibilní a modulární strukturu. Např. při dostupnosti rychlejšího jednočipu, který bude zvládat větší nároky na výpočetní výkon a rychlost, neupravovat celý systém, ale pouze dílčí část, která bude kompatibilní s již hotovým uživatelským rozhraním, vytvořenými programy atd. Dalším příkladem může být přechod na jiný vývojový systém. Máme hotový hardware, komunikační spojení s PC, drivery, uživatelské rozhraní, ale chceme vytvořit nové uživatelské rozhraní, které není závislé na komerčním softwaru. Cílem je tedy vytvořit pouze nové uživatelské rozhraní, které bude schopné komunikovat s ostatními dílčími moduly beze změny (původní komerční verze bude nadále funkční).

Prvním problémem k dosažení našeho cíle je komunikace. Pro předávání dat v různých typech systémů se dnes používá např. jazyk XML. Tento jazyk by byl vhodný pro naše potřeby, ale protože jsme omezeni paměťovou kapacitou a výpočetním výkonem jednočipu, bylo nutné použít něco úspornějšího. V kap. 4.2 je popsán komunikační protokol, který funguje na bázi textových příkazů. Jde o univerzální protokol, který může fungovat ve všech operačních systémech i jednočipech.
V dalším textu budou podrobněji popsány dílčí moduly z blokového schématu na Obr. 3.

MFU168
MFU168 je hardware, který funguje jako vstupně/výstupní karta, obsahuje komunikační linku a firmware, který reaguje na textové příkazy přicházející z bloku Komunikace a umí je vykonat. Vzhledem k použitému protokolu z kap. 4.2 může být použit jakýkoliv jednočip, který má dostatečnou kapacitu pro zpracování textových příkazů a obsahuje nutné periferie, k jejich vykonání. V této práci se počítá s použitím jednočipu řady ATmega. Při vývoji bude však brán zřetel pro možné nahrazení jiným typem jednočipu.

Komunikace
Jedná se o mezičlánek mezi vyvíjenou MFU168 a softwarem na PC, kterým budeme MFU168 ovládat. Tento blok můžeme chápat pouze jako prostředek, který doručí data mezi MFU168 a aplikačním softwarem. Zpočátku se předpokládala komunikace po RS232 lince kvůli její jednoduchosti. Díky absenci sériových portů na dnešních laptopech je však nutné použít převodníky na USB sběrnici. Tímto značně roste variabilita použitých zařízení v bloku Komunikace. Výsledným požadavkem pro náš další postup při návrhu bude zcela oddělit tento blok jak od bloku MFU168, tak od bloku Aplikace (viz. Obr. 3).

Při vývoji byly střídavě využívány různé převodníky (viz kap. 5.3.1, kap. 5.3.2), z toho plyne používání různých driverů. Pro ovládání MFU168 byly používány také různé typy aplikací. Pro zvýšení efektivity při programování bylo tedy nutné vytvořit blok Komunikace a tento blok udělat co nejvíce transparentním, abychom mohli komunikovat mezi blokem Aplikace a MFU168 bez uvažování, jakým způsobem jsou spojeny.

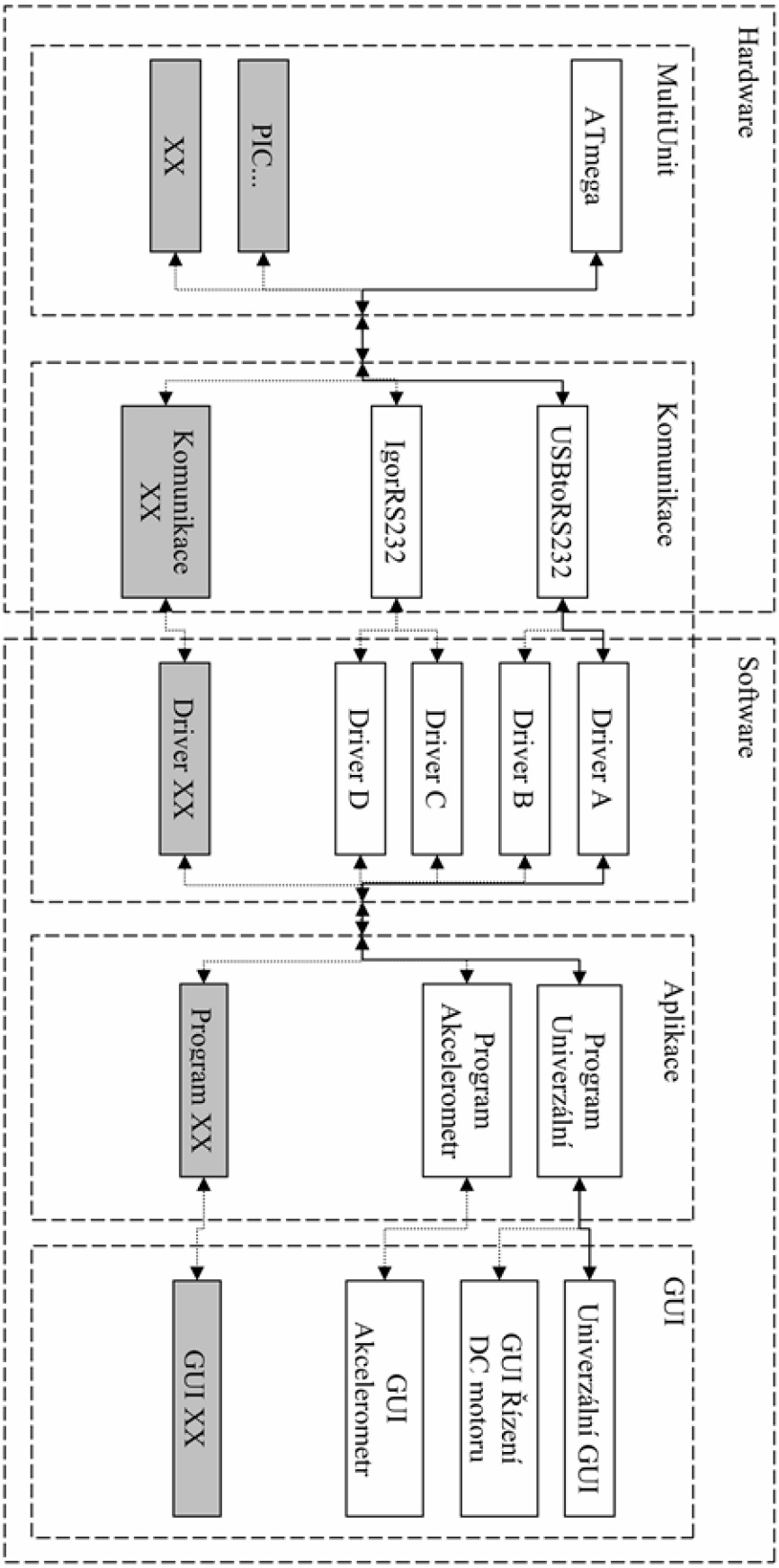
Za tímto účelem byl navržen dvojbran, branou mezi blokem MFU168 a blokem Komunikace byl zvolen komunikační port RS232, druhou branou mezi blokem Aplikace a blokem Komunikace bude tvořit buffer v paměti PC, do kterého může zapisovat a číst Aplikace.

Aplikace
Jedná se o software pro ovládání MFU168, odesílá do MFU168 příkazy a zpracovává přijaté data. Pro různé typy úloh bude potřeba vytvořit různé aplikace, které budou zpracovávat přijaté data nej-

vhodnějším způsobem pro daný typ úlohy. Díky stejnému komunikačnímu protokolu s MFU168 bude možné použít jednu aplikaci i pro více tříd úloh, které mají podobný charakter. Např. při měření teploty nebo měření osvětlení získáváme z MFU168 v obou případech sérii hodnot, které pouze jinak interpretujeme. Tato interpretace lze uskutečnit v dalším bloku GUI. Můžeme používat tedy stejnou aplikaci na více úlohách a pouze měnit zobrazení (GUI), jakým budeme měřená data interpretovat. Snahou je co nejvíce minimalizovat psaní nového programového kódu.

GUI

Uživatelské rozhraní, které obsahuje ovládací prvky jako tlačítka, posuvníky atd. a které zobrazuje přehledně naměřená data. Pro jeden typ úlohy je počítáno i s více uživatelskými rozhraními, které se liší dostupnými ovládacími nástroji a možnostmi nastavení, avšak využívá stejné Aplikace. Není vyloučeno ani použití uživatelského rozhraní s více podobnými aplikacemi, cílem je tedy vytvořit modulární komunikační systém. Tento systém bude jednotný a budou jej využívat všechny aplikace a uživatelská rozhraní ke komunikaci mezi sebou. Protokol mezi GUI a Aplikací je plánován na podobném principu jako v kap. 4.2. Dosáhneme tím rychlejšího vývoje a jednotného uspořádání, které usnadní pozdější editaci softwaru.



Obr. 3 Vývojové blokové schéma pro MFU168 a její ovládání z PC

4.2 Komunikační protokol

V kap. 4.1 jsme zjistili, že je třeba vytvořit komunikační protokol nezávislý na platformě, kde je provozován. Zároveň musí být protokol natolik jednoduchý, aby byl snadno zpracovatelný jednočipem. Pro komunikaci mezi MFU168 a aplikací (viz schéma na Obr. 3) byla navržena následující konstrukce.

Příkaz bude složen z ASCII znaků, každý příkaz je ukončen znakem „;“,např. „funkce;“. Pokud je nutno zadat příkaz i s parametrem, oddělovací znak je mezera „ “, forma příkazu „funkce parametr;“. V případě více parametrů se parametry oddělují čárkou „,“, např. „funkce parametr1,parametr2;“.

Pro komunikaci mezi Aplikací a GUI (viz schéma na Obr. 3) bude použit textový řetězec k identifikaci funkce. Tento řetězec bude předán jako první parametr funkci s názvem *pokyn*:

```
pokyn('merit');
```

Tato funkce bude obsažena v každé Aplikaci nebo GUI a bude se starat o zpracování příchozích příkazů. Pokud je potřeba předávat s příkazem i parametr, zadáme jej jako druhý parametr ve funkci *pokyn*:

```
pokyn('merit',3);
```

V případě předávání více parametrů použijeme jako druhý parametr funkce pokyn strukturu, která bude obsahovat všechny parametry

```
pokyn('merit',struct{3,'teplota',...,0.01});
```

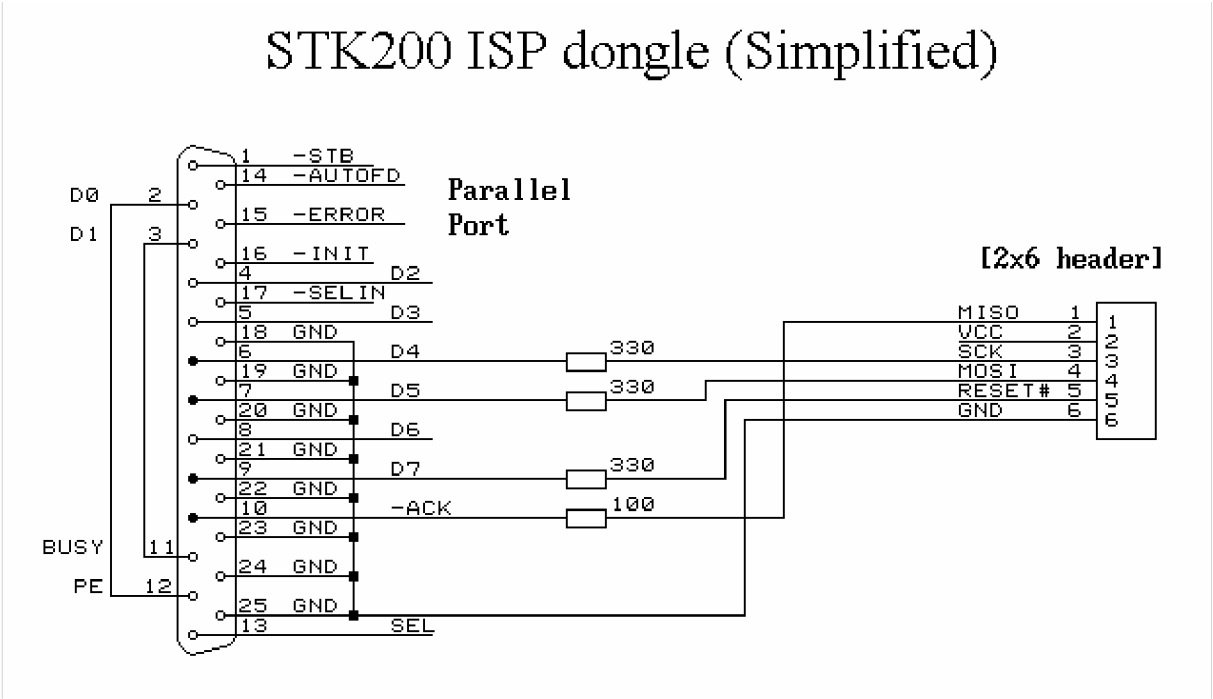
Tímto je zajištěna jednotná komunikace mezi všemi funkčními bloky a po vytvoření prvního funkčního programu můžeme kód pro komunikaci jen kopírovat. Všechny aplikace s tímto protokolem můžou spolu komunikovat a není nutné měnit kód starých aplikací pro podporu nově vytvořených aplikací.

4.3 Jednočip

Pro první seznámení byl vybrán jednočip ATmega8. Na internetu jsou spousty návodů, v diskuzích je často probírán a existuje celá řada tutoriálů. Je běžně dostupný a cena je v řádech desetikorun. Nevýhodou je však jeho malá kapacita programovatelné paměti FLASH, pouhých 8kB. Později bude nutné přejít na jednočip s větší kapacitou. Pinově kompatibilní s ATmega8 je ATmega168, která má větší programovatelnou kapacitu (16kB). Při ještě větších požadavcích lze přejít na vyšší řady, největší z této rodiny je ATmega128, která má 128kB programovatelné FLASH paměti a více jak 2x tolik I/O pinů oproti ATmega8.

5 Realizace procesu řešení

5.1 Programátor STK200



Obr. 4 Schéma zapojení programovacího kabelu STK200 ISP

[3]

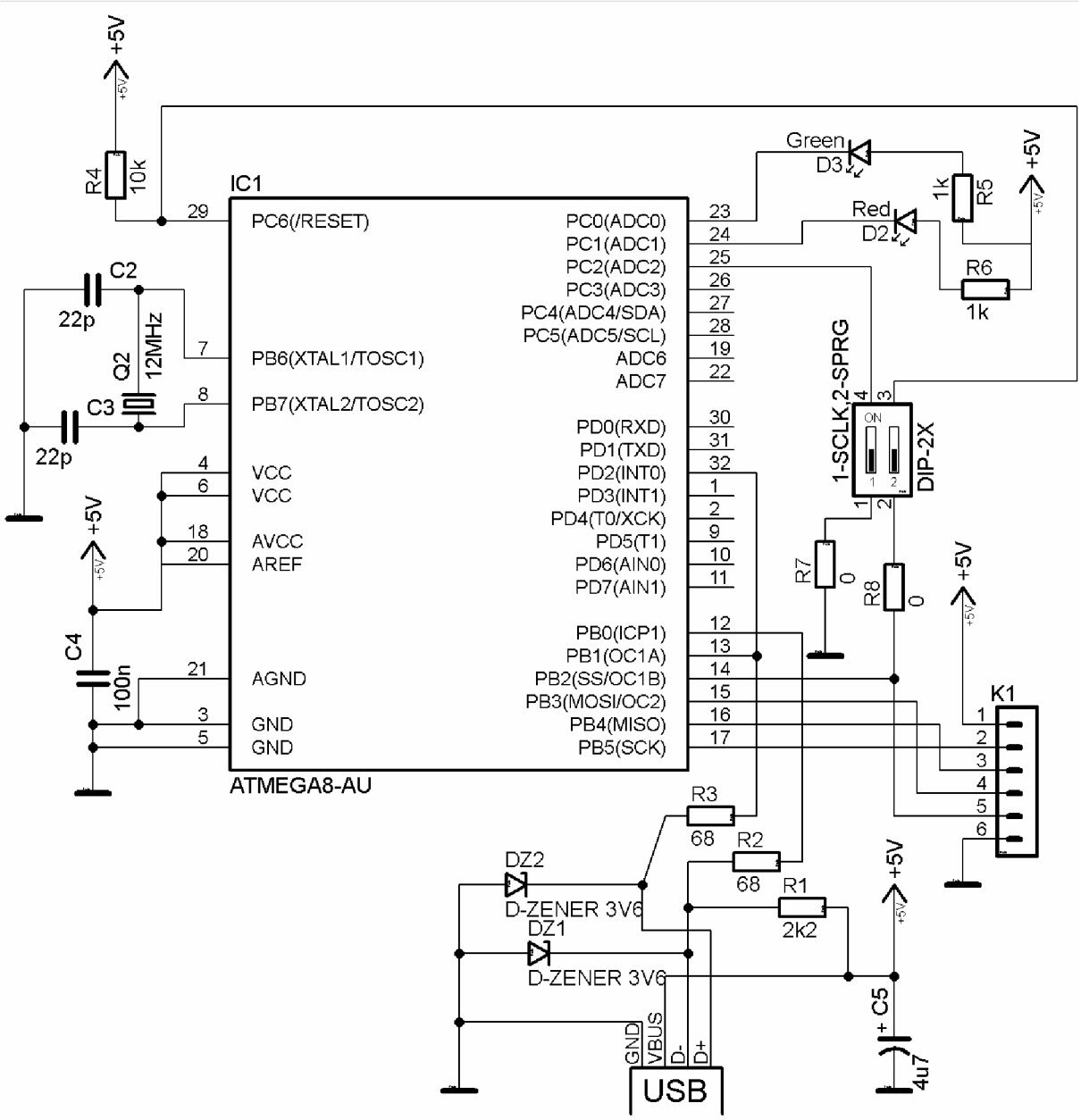
Pro první programování byl použit programátor využívající paralelní port počítače. Na Obr. 4 je schéma zapojení. Je velmi jednoduchý, používá minimum součástek a pro první seznámení s mikročipy je na stránkách [3] detailně napsaný návod. Jeho nevýhodou je nutnost mít paralelní port v počítači, který se dnes na laptopech již nevyskytuje.

5.2 Programátor ASPUSB

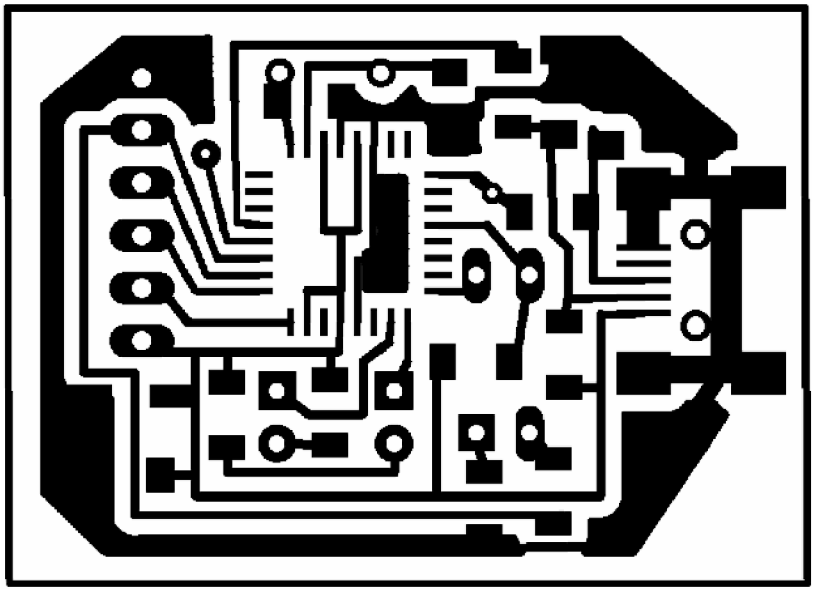
Kvůli absenci paralelního portu v novějších laptopech bylo nutné použít programátor využívající sběrnici USB. Je sice možné využít jednodušších programátorů viz. kap. 5.1 a nějakého převodníku na USB sběrnici, ale při pokusech nebylo dosaženo uspokojivých výsledků. Při variantě s programátorem využívající paralelní port a převodníku na USB se přenos vůbec neuskutečnil. Při další variantě použít programátor využívající sériovou linku a převodníku na USB byl sice přenos uskutečněn, ale při naprogramování krátkého kódu trvala operace téměř hodinu, což je nepřijatelné.

V [2] lze nalézt programátor využívající přímo sběrnici USB. Je založen na mikročipu ATmega8, USB rozhraní je implementováno softwarově. Je to levný programátor, který byl prvotně sestaven na nepájivém kontaktním poli. K jeho naprogramování byl využit programátor z kap. 5.1. Při testování byl menší problém s knihovnami, v případě nefunkčnosti programátoru je třeba vyzkoušet starší verze firmware. Po odzkoušení na nepájivém kontaktním poli bylo navrženo schéma a předlohy DPS v programu EAGLE. Spousta schémat a předloh DPS je uvedena v [2], avšak cílem bylo mimo hotového programátoru lépe se seznámit s prostředím EAGLE a výrobou plošných spojů. Na Obr. 5 lze shlédnout hotové schéma zapojení programátoru USBASP. Oproti schématům uvedených v [2] bylo provedeno několik kosmetických úprav (barvy LED, spínače,...). Na Obr. 6 je výsledná předloha DPS, je to jednostranná předloha, je použita jedna drátová propojka. Rozměry jsou

39x28mm. Na Obr. 8 a Obr. 9 je osazený programátor USBASP. Bylo použito pouze SMD prvků a celá výroba probíhala v domácích podmínkách.

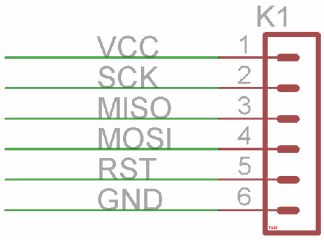


Obr. 5 Schéma zapojení programátoru USBASP

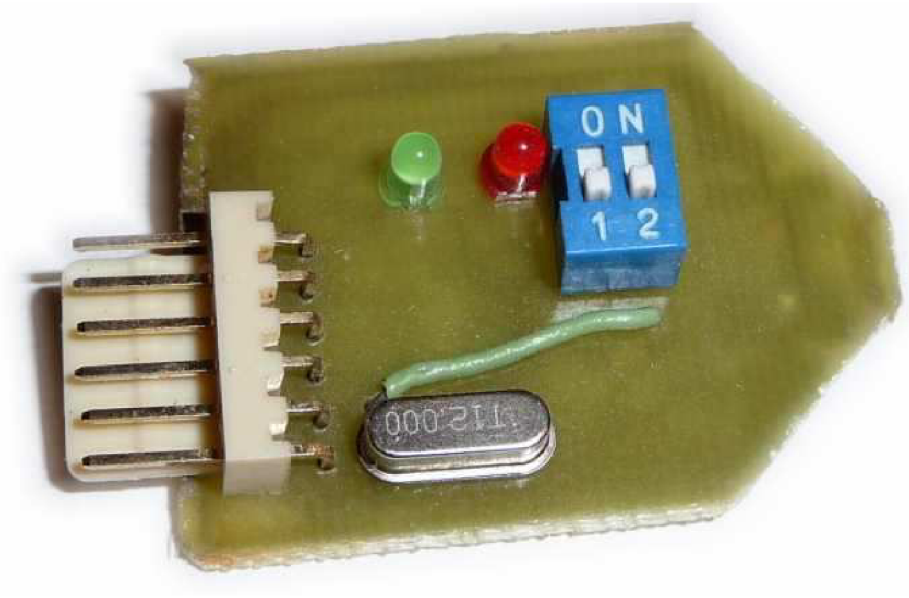


Obr. 6 Předloha DPS pro programátor USBASP

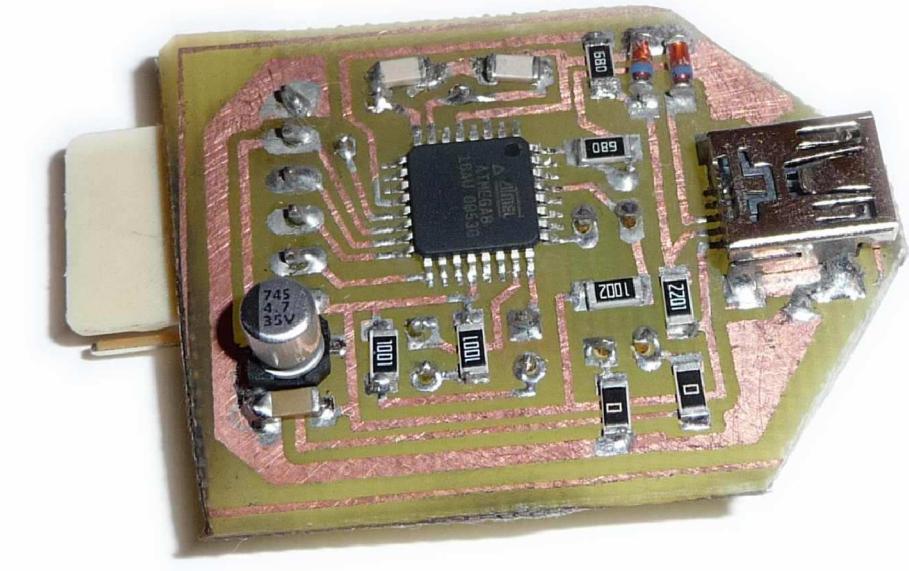
Programátor se k mikročipu připojuje přes šestipinovou lištu se zámkem - Obr. 8. vlevo. Na svrchní straně jsou dva spínače - Obr. 8. vpravo. Spínač S1 slouží k přepínání programovací rychlosti. Pro programování mikročipů s taktem pod 1,5MHz je nutné přepnout na nižší programovací rychlost zapnutím spínače S1 do polohy on. Nejvyšší rychlost programování je autorem [2] udávaná 5kB/sec. Pro vyšší programovací rychlost spínač S1 vrátíme do polohy off.
Spínač S2 je určen pro update firmwaru v programátoru.



Obr. 7 Schéma zapojení zásuvky pro programovací kabel



Obr. 8 Horní strana DPS osazeného programátoru USBASP



Obr. 9. Spodní strana DPS osazeného programátoru USBASP

5.3 Převodníky USB pro komunikační linku RS232

5.3.1 USB2RS232 převodník

Převodník z USB na sériový port (COM), viz Obr. 10. Přenosová rychlost až 1,5Mbaud. Převodník je běžně k dostání v prodeji, je založen na FT232BM (jednočipový převodník USB ↔ RS232). [4]

Obr. 10. Převodník USB to RS232 Cable

Výhody:

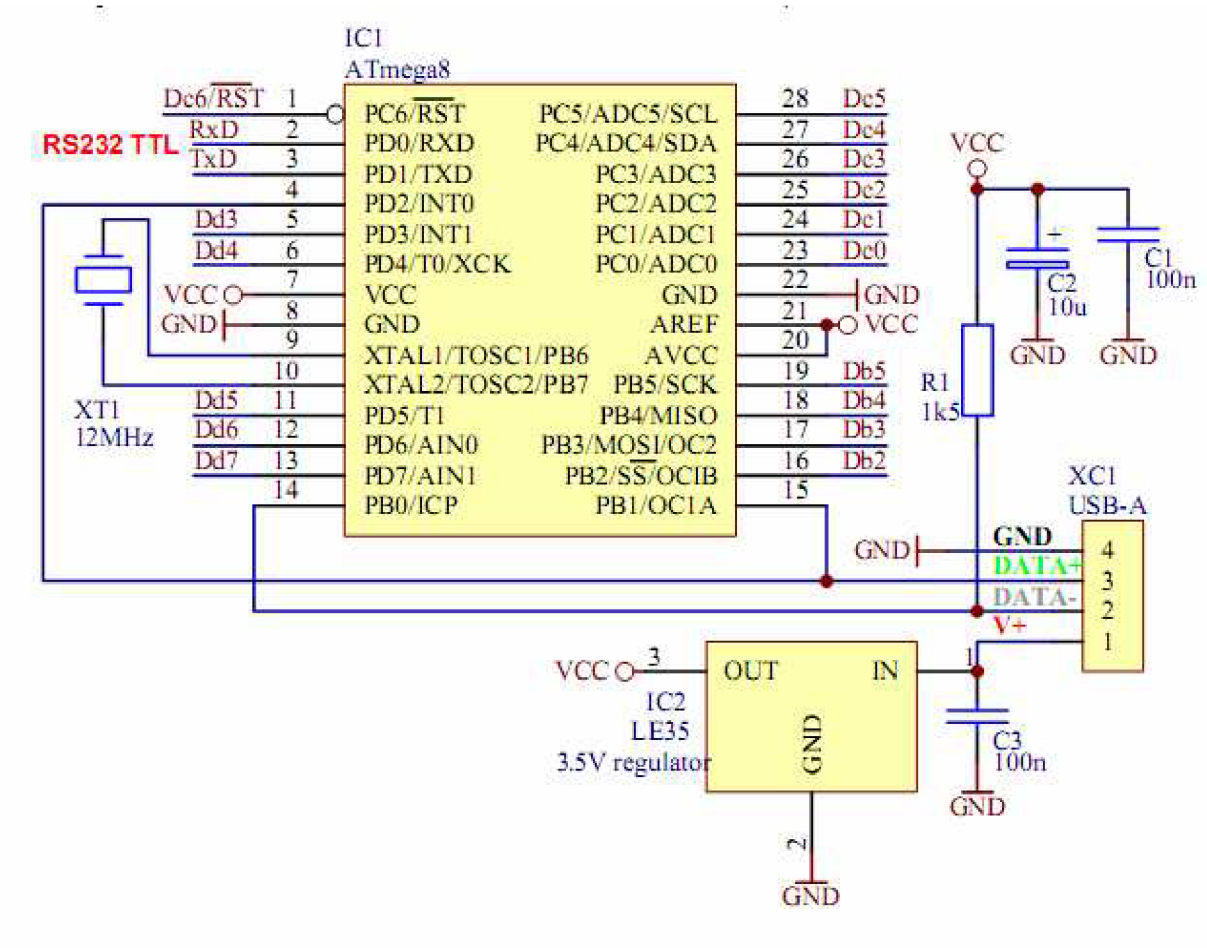
- hotové řešení, není potřeba odborných znalostí
- uspokojivá rychlost přenosu

Nevýhody:

- nelze zvýšit rychlost přenosu
- malý interní buffer
- je potřeba invertovat napěťové úrovně pro připojení mikročipu

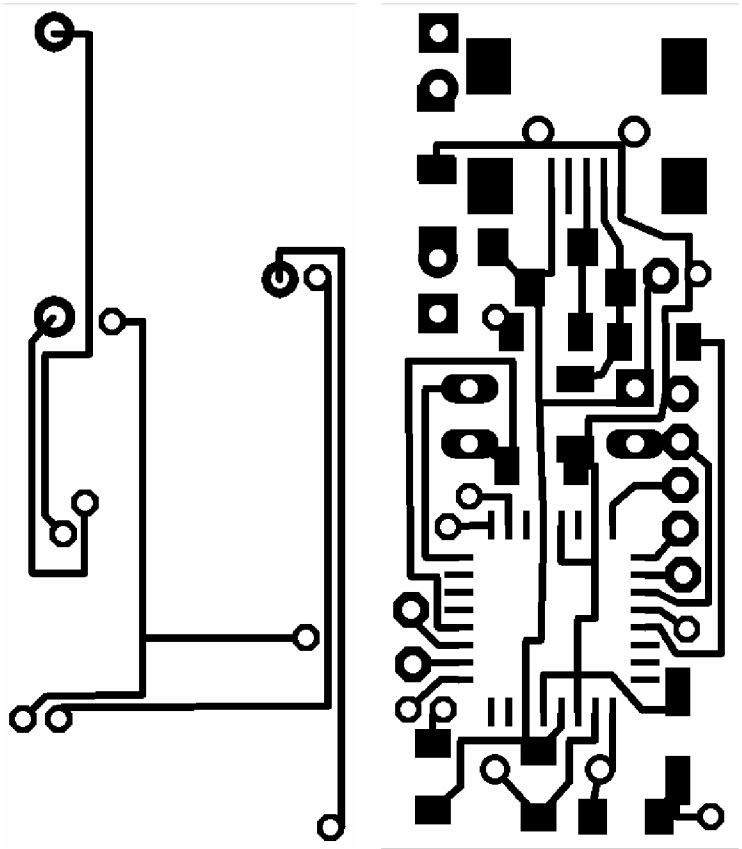
5.3.2 IgorRS232toUSB

Autorem převodníku je Igor Česko, viz. [1]. Tento převodník je postaven na mikrokontroléru ATmega8. Schéma zapojení je velmi podobné zapojení programátoru ASPUSB použitého v kap. 5.2, rozdíl je pouze ve výstupu pro sériovou linku. Cenově je IgorRS232toUSB nejlevnější řešení pro převod sériové linky na sběrnici USB, ale je potřeba odborných znalostí k jeho zapojení a spuštění. Autor napsal ovládací program, ve kterém lze vše vyzkoušet. Později lze převodník využít v našem software pomocí ovládací knihovny, která je součástí ovladačů k IgorRS232toUSB. Převodník umožňuje mimo sériové komunikace zapínání a vypínání ostatních nepoužitých pinů na mikročipu ovládacími příkazy po USB sběrnici, lze tedy využít i ostatní piny např. na indikaci komunikace nebo přepínání příjemce.



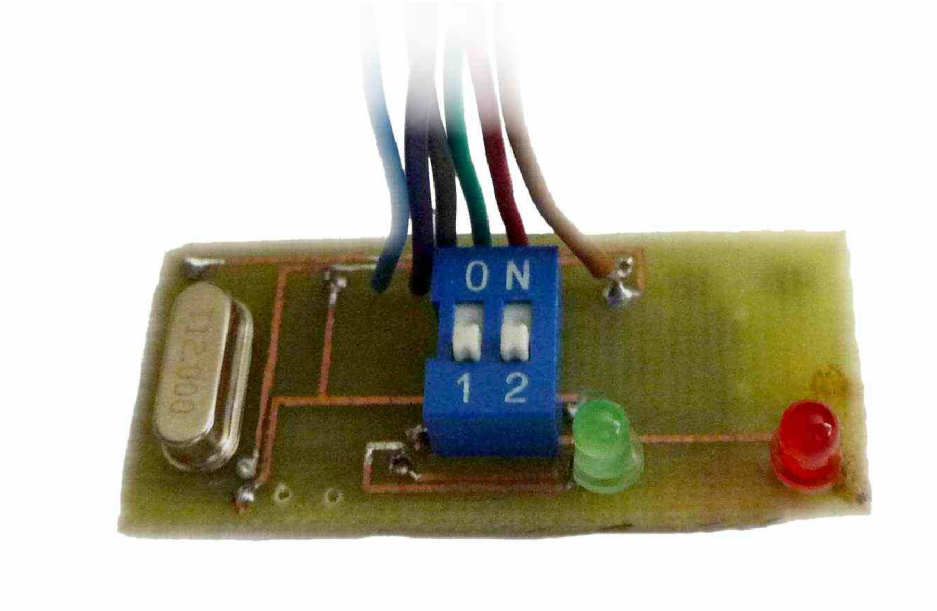
Obr. 11. Schéma zapojení převodníku IgorRS232toUSB [1]

Vzhledem k podobnosti s programátorem ASPUSB je nejrychlejší cestou k funkčnímu prototypu IgorRS232toUSB využít dřívější schéma z Obr. 5 a mírně jej upravit. Paradoxně však vznikl zcela nový obvod a to z důvodu lepšího seznámení s tvorbou menších elektrických obvodů. V zadání je požadavek na miniaturní provedení měřicí jednotky a převodník IgorRS232toUSB se v této fázi návrhu jevil jako dobrá volba pro zkoušení miniaturizace obvodu díky dřívějším zkušenostem s podobným zapojením u programátoru ASPUSB.

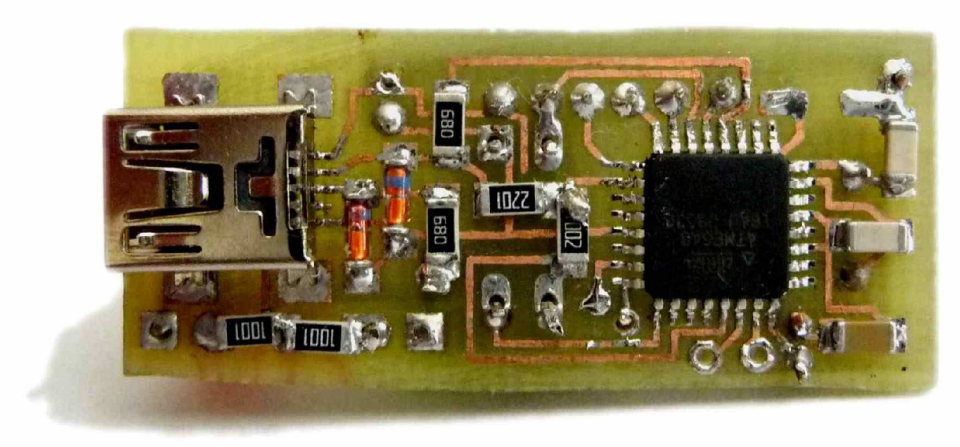


Obr. 12. Předloha DPS IgorRS232toUSB

Hotová předloha DPS je na Obr. 12, je patrné, že na plošném spoji téměř není nevyužité místo. Rozměry DPS jsou pouhých 40x15mm. Hotový a osazený převodník je na Obr. 13 a Obr. 14. Pro připojení k PC je použita miniUSB zásuvka v SMD provedení. Vstup na programování převodníku byl navržen pouze z přívodních kabelů (viz. Obr. 13 nahoře), protože při použití konektoru by se rapidně zvětšila velikost výsledného DPS.



Obr. 13. Osazené DPS IgorRS232toUSB horní strana



Obr. 14. Osazené DPS IgorRS232toUSB dolní strana

Osazení IgorRS232toUSB bylo značně složitější než např. programátor z kap. 5.2, bylo nutné dbát na pořadí osazování součástek a velmi pečlivý přístup. Při chybném pořadí osazení není vyloučeno ani odstraňování správně osazených součástek, aby bylo možné se s pájecím hrotem fyzicky dostat k pájecím ploškám.

Výhody:

- nízká cena
- snadná dostupnost
- mirkočip lze propojit přímo s převodníkem bez změny napěťových úrovní

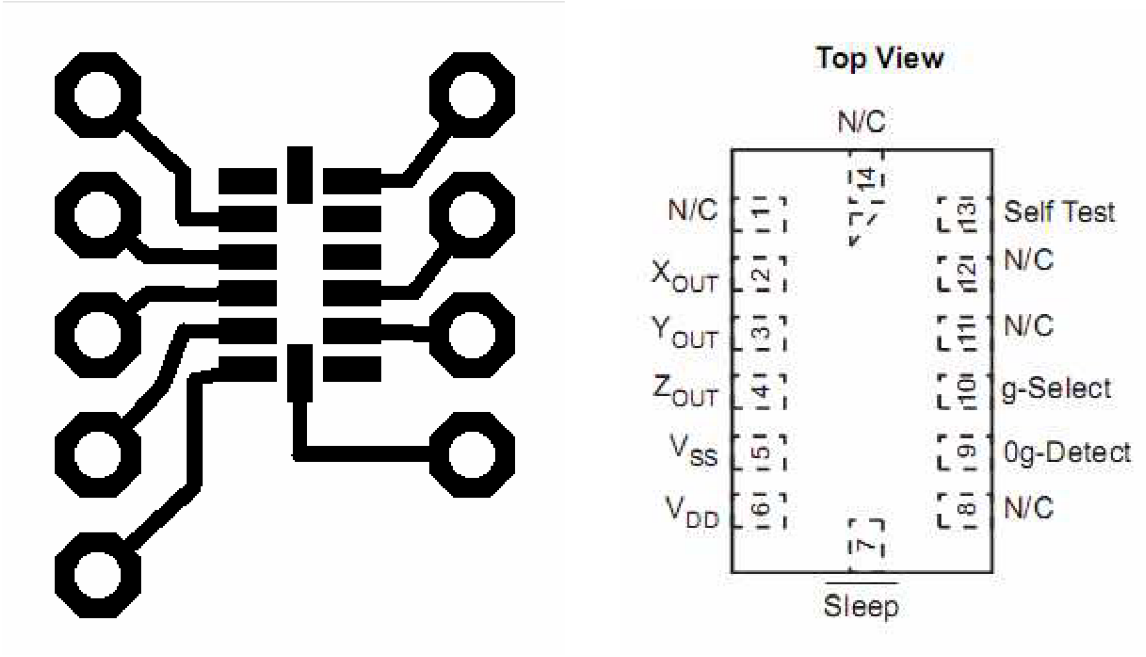
Nevýhody:

- malá rychlost přenosu
- potřeba odborných znalostí pro sestavení

5.4 Akcelerometr

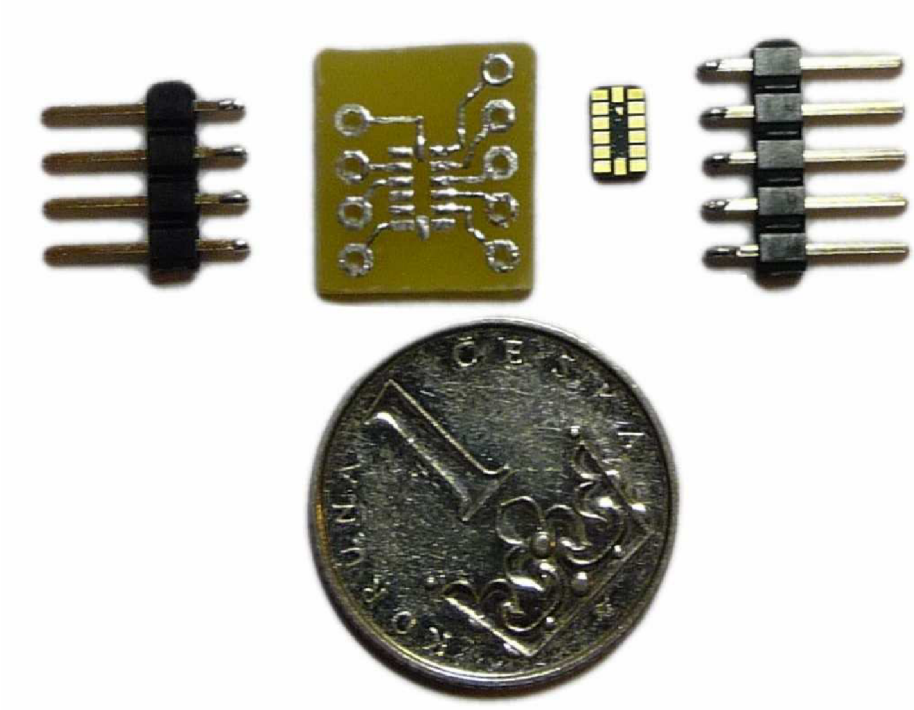
Součástí miniaturní měřicí jednotky je akcelerometr, senzor pro měření zrychlení. V dnešní době je tento typ senzoru vyráběn pouze v pouzdrech pro SMT, pouzdra typu LGA, LCC, QFN16 (pájecí plošky na spodní straně pouzdra). To je značně problematické z hlediska práce v domácích podmínkách, protože tento typ pouzdra je velmi malých rozměrů a osazení mikročipu je také problematické.

Pro *Miniaturní měřicí jednotku* byl vybrán akcelerometr MMA7361L. Jedná se o analogový 3-osý akcelerometr, s volitelnou citlivostí 1,5g nebo 6g. Jako první testovací verze byla navržena redukce, která má výstupní piny kompatibilní s nepájivým kontaktním polem (Obr. 15).



Obr. 15 Předloha DPS redukce akcelerometru MMA7361L, popis pinů [6]

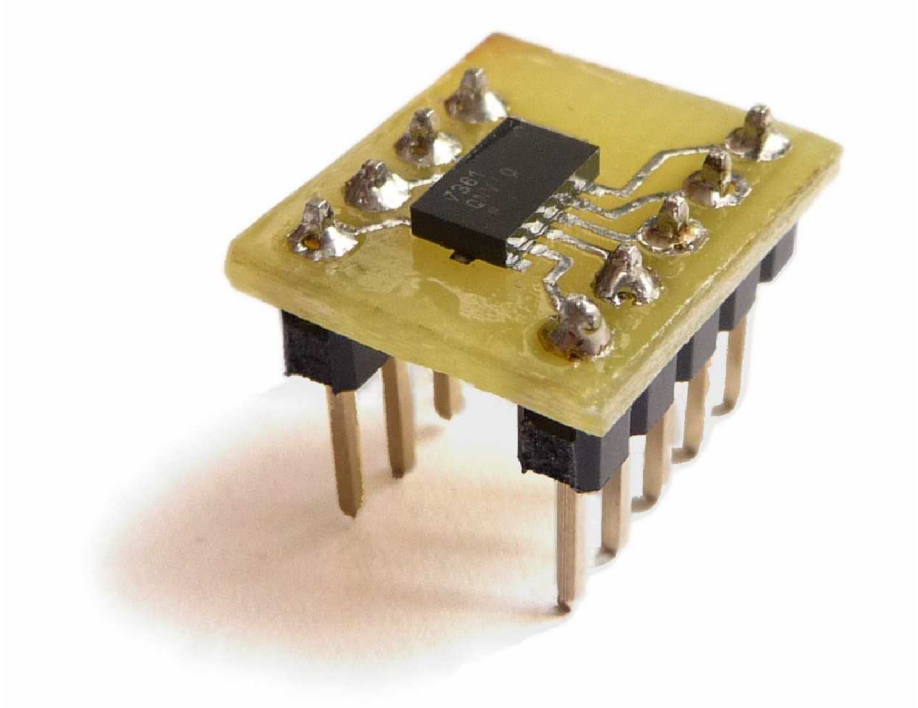
Při návrhu DPS bylo čerpáno z doporučení pro pájení pouzder LGA, viz [7]. Cesty vycházející z pájecích plošek musí být tloušťky 2x menší, než je pájecí ploška, z důvodu přesného usazení součástky a prohřátí cínu. V okolí součástky se nesmí nacházet žádné díry nebo koncentrátory napětí, aby nedošlo k poškození akcelerometru během provozu. Po dokončení návrhu byl vytvořen plošný spoj, na který byl osazen akcelerometr a pinové lišty (Obr. 16).



Obr. 16 DPS redukce akcelerometru MMA7361L

Osazená redukce je na Obr. 17, podle [6] bylo provedeno základní zapojení v nepájivém kontaktním poli a odzkoušena funkčnost. Akcelerometr byl funkční, avšak výstupní hodnoty byly velmi

zarušené, proto byl přidán do obvodu RC filtr. Ten byl následně použit i při návrhu DPS *Miniaturní měřicí jednotky* v příloze IV.2.



Obr. 17 Osazená redukce akcelerometru MMA7361L

5.5 Funkce MFU168

V kap. 3 byly analyzovány požadavky na funkce, které by měla MFU168 obsahovat. V rámci univerzálního použití bylo při vývoji dbáno na tvorbu funkcí, které mezi s sebou nebudou kolidovat a bude možné s jedním firmwarem v MFU168 pouze přepínáním příkazů měnit využití schopností MFU168. Během vývoje firmwaru byl mnohokrát kompletně předělán princip zpracování příkazů odeslaných z PC a většina původních funkcí, které tyto příkazy spouštěly, byla nahrazena novějšími, které jsou obecnější a nekolidují mezi s sebou navzájem. Výsledkem celého vývoje je řada příkazů/funkcí, které jsou popsány v příloze II.1. V případě tvorby nových funkcí, filtrů nebo regulátorů je vhodné kopírovat již hotové funkce a pouze je upravovat. Ve většině funkcí jsou použita makra, která velmi urychlují programování a zpřehledňují zápis kódu.

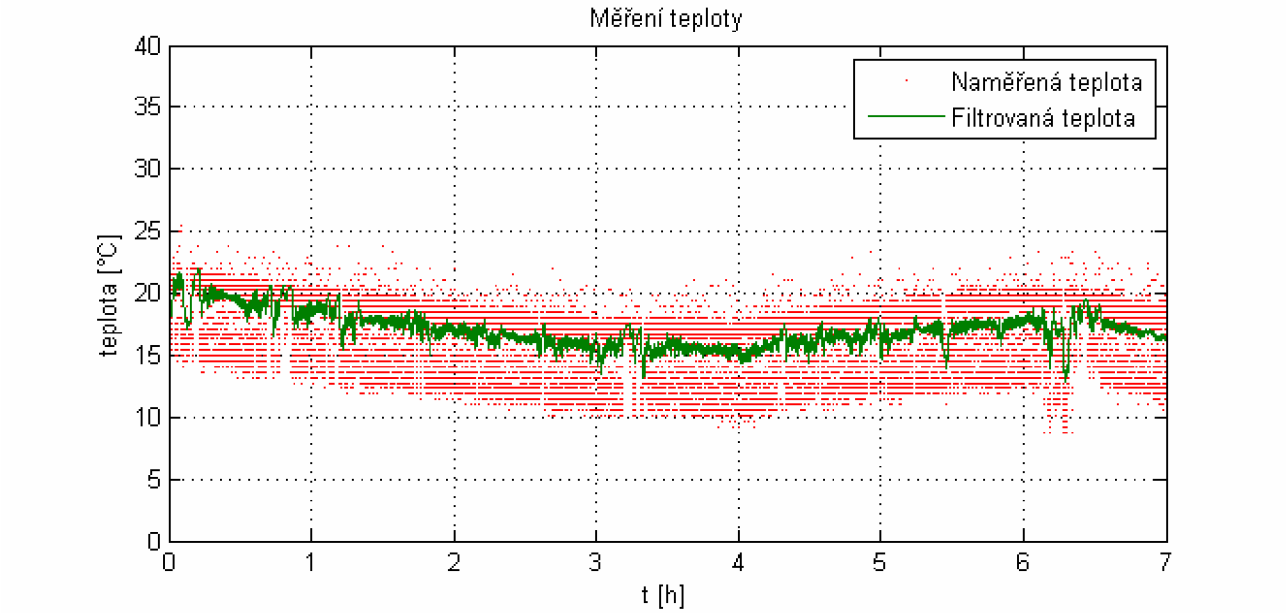
V hlavičkovém souboru *prikazy.h* jsou uložena klíčová slova pro spuštění příkazů MFU168, krátká nápověda pro každý příkaz a odkazy na rutiny, které se spouští po vyvolání příkazu. Rutiny jsou uloženy v souboru *prikazy.c*.

6 Prezentace výsledků řešení problému

6.1 Měření teploty

Pro otestování funkčnosti *Miniaturní měřicí jednotky* (MMJ) provedeme měření teploty, které bude probíhat v řádu hodin a měření bude uloženo na MicroSD kartu. MMJ připojíme k PC a odešleme příkazy pro nastavení měření teploty a zápis na MicroSD kartu.
Příkazy pro měření teploty a ukládání na MicroSD kartu:
`logSD;`
zapne logování příkazů do SDkarty, dále se budou ukládat všechna nastavení do připojené MicroSD karty
`meritAD0;`
zapne analogové měření napětí na pinu AD0, kde je připojený teplotní senzor
`kodeslani AD0;`
hodnota AD0 bude odesílána komunikační linkou do PC
`zapisSD;`
hodnoty budou místo odesílání do PC ukládány na MicroSD kartu
`vzorkovani 1000;`
měření bude probíhat každých 1000ms, tzn. každou vteřinu
`povolit;`
spustí měření
`svit;`
rozsvítí signalizační LED, indikuje zapnuté měření
`logSD;`
vypne logování příkazů na MicroSD kartu

Tímto je v MicroSD kartě uloženo nastavení pro měření teploty, při zapnutí MMJ a po zapnutí konfiguračního vypínače je načteno nastavení z MicroSD karty a spuštěno měření. Při vypnutí konfiguračního vypínače je měření ukončeno a uloženo na MicroSD kartu. Poté můžeme naměřená data stáhnout např. v GUI z přílohy II.2 do PC. Ukázka měření pokojové teploty v nočních hodinách po dobu 7 hodin je na Obr. 18. Měření trpí velkým šumem, to je způsobeno levným teplotním senzorem, který je určen pro rozsah 0-100°C a nedává přesné výsledky v malém teplotním rozsahu.

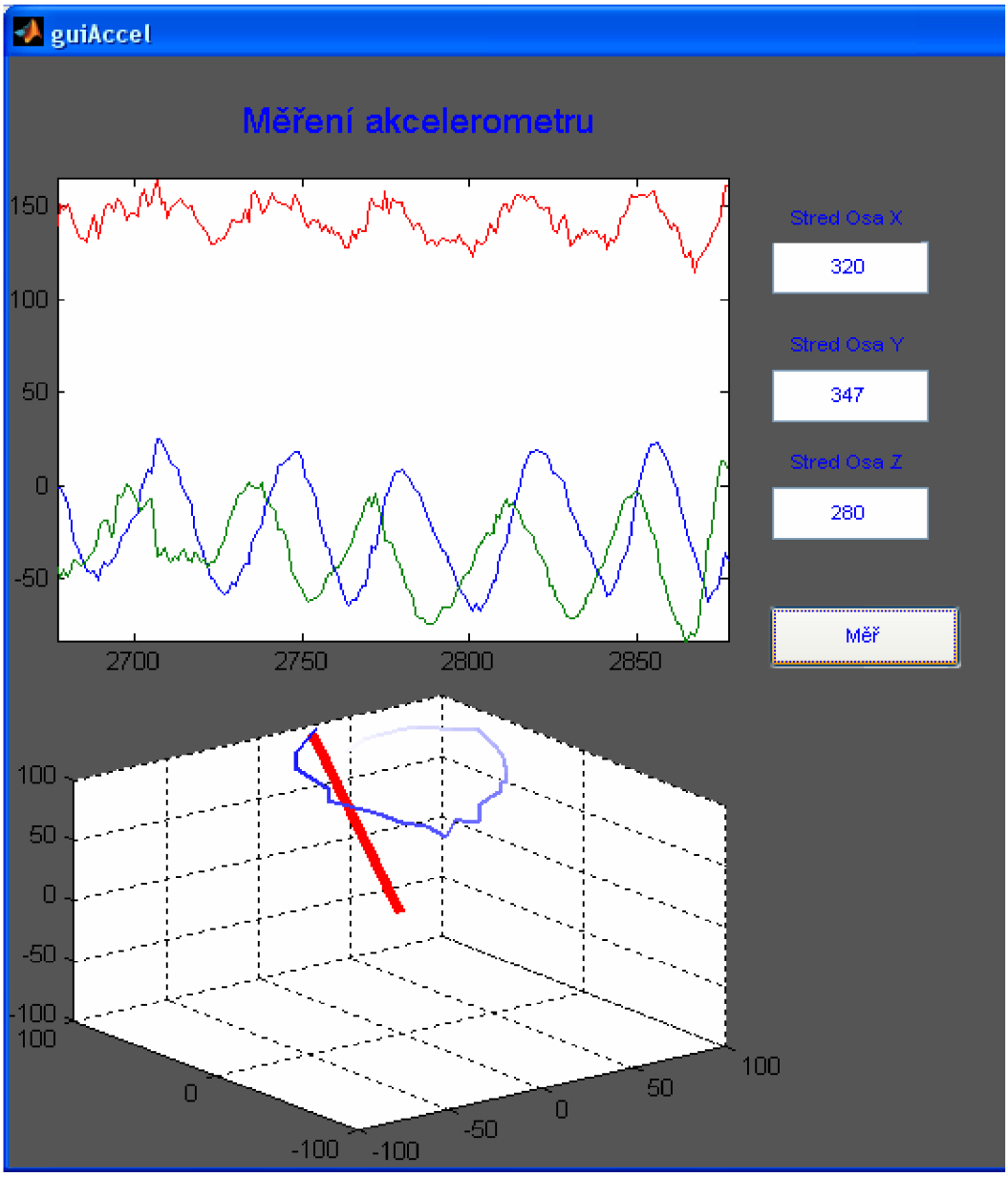


Obr. 18 Měření teploty

6.2 Měření akcelerometru

K MFU168 je připojen 3-osý akcelerometr z kap. 5.4. V programu MATLAB bylo vytvořeno uživatelské rozhraní (viz. Obr. 19), které je uzpůsobeno pro zobrazení hodnot z akcelerometru. Obsahuje tlačítko pro spuštění/zastavení měření a tři editovatelné pole pro nastavení středních hodnot. V horním grafu jsou vidět průběhy zrychlení všech tří os v čase. Měřítko je variabilní podle použitého akcelerometru a zvolené citlivosti měření. V dolním grafu jsou údaje o zrychlení složeny do 3D-grafu. Pokud je akcelerometr v klidu, díky statickému gravitačnímu zrychlení ukazuje dolní graf natočení akcelerometru v prostoru.

MFU168 je v zapnutém stavu vyresetovaná, proto je nutné před samotným měřením akcelerometru odeslat nastavovací příkazy. Odeslání příkazů je implementováno v inicializačním souboru uživatelského rozhraní, takže uživatel se jimi nezatěžuje, ale při tvorbě nového GUI je třeba s nimi počítat.



Obr. 19. Uživatelské rozhraní pro vizualizaci dat naměřených z akcelerometru

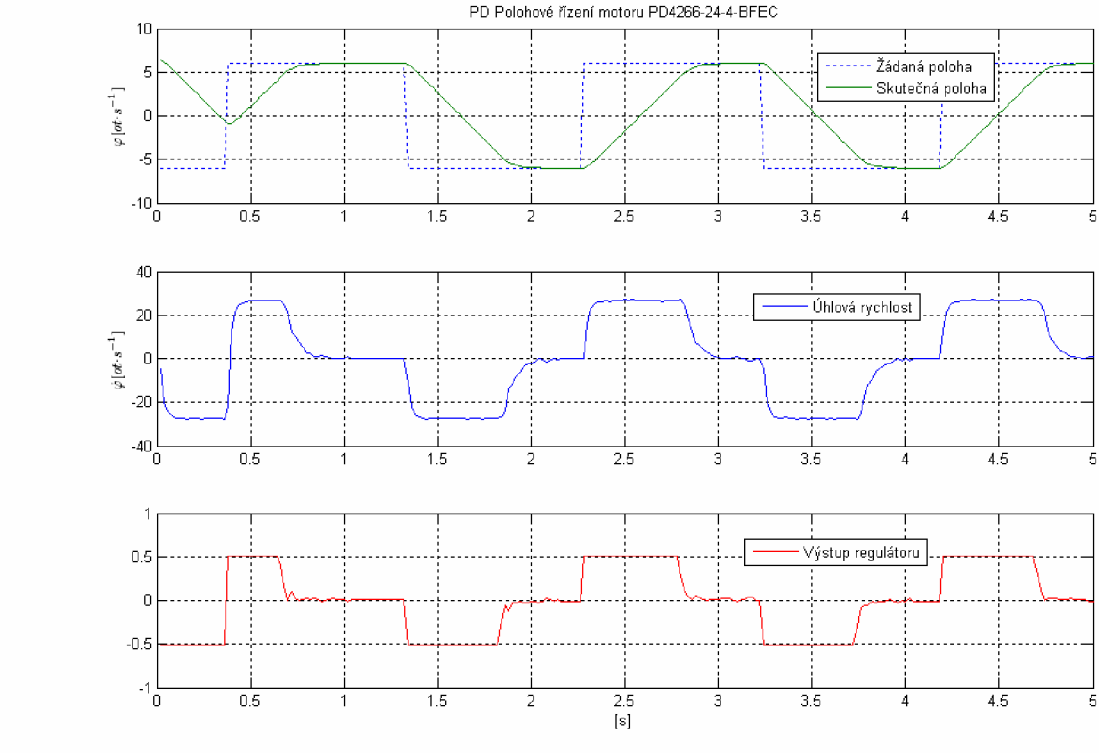
6.3 Regulace motoru PD4266-24-4-BFEC

MFU168 byla testována při řízení s motorem PD4266-24-4-BFEC. Jedná se o DC motor s planetovou převodovkou od firmy Transmotec. Bylo vyzkoušeno polohové i rychlostní řízení motoru. Motor obsahuje vestavěný enkodér, který byl využit při regulaci. Jako výkonové zařízení byl využit H-můstek z přílohy III.2.

Pro polohovou regulaci motoru PD4266-24-4-BFEC byly použity následující příkazy:

Tab. 1: Polohová regulace motoru PD4266-24-4-BFEC

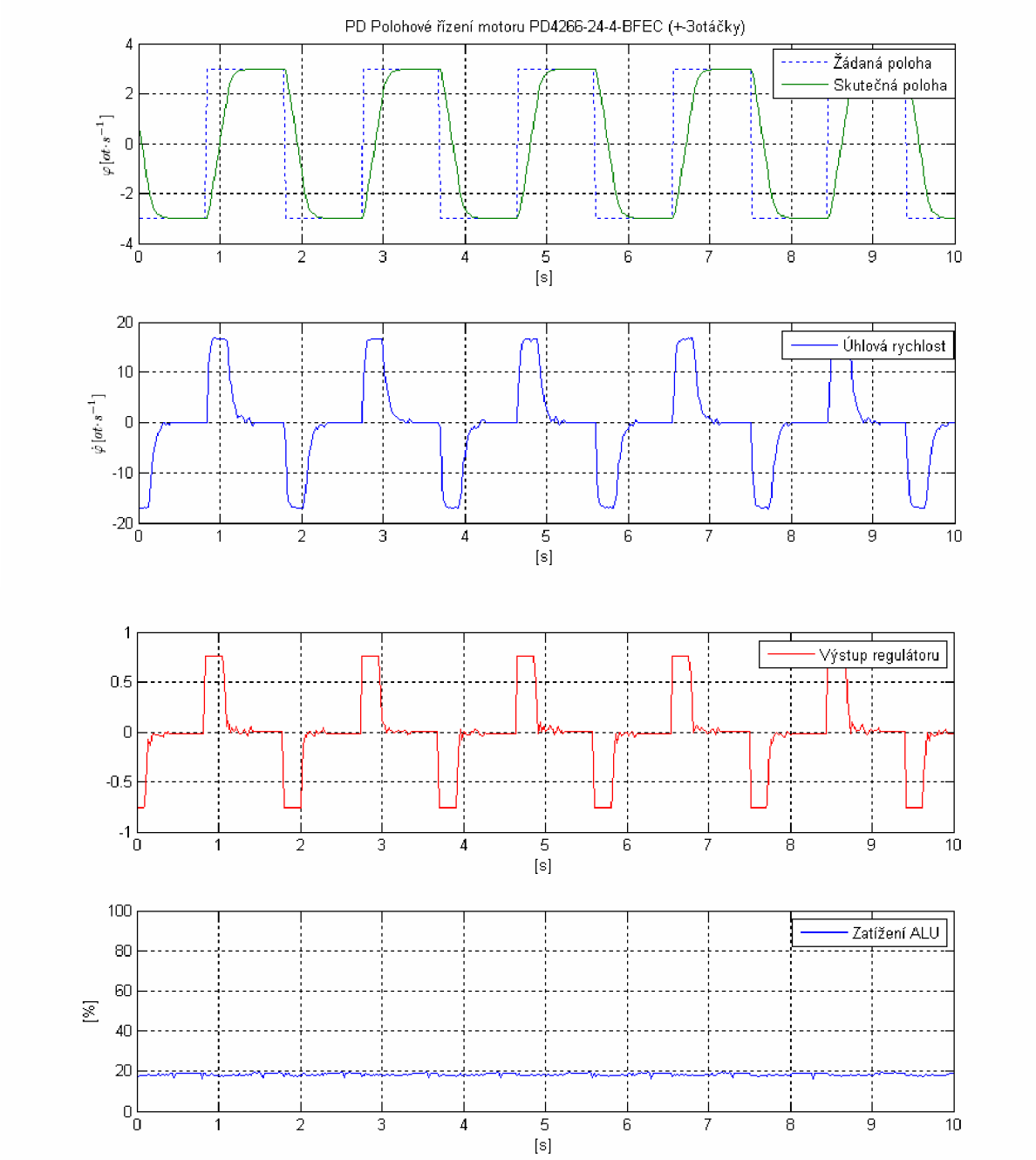
Příkaz	Popis
vzorkovani 1	Řídicí smyčka-perioda=1ms
K1 0,34	Zesílení pro kalmanův filtr
K2 67.6	Zesílení pro kalmanův filtr
gp 1000	Přepínání obdélníkové referenční hodnoty, změna každou 1s
gh 1000	Horní poloha obdélníkové referenční hodnoty
gd -1000	Dolní poloha obdélníkové referenční hodnoty
prp 2	Zesílení proporcionální složky regulátoru
der -0.1	Zesílení derivační složky regulátoru



Obr. 20 PD Polohové řízení motoru PD4266-24-4 BFEC (6 otáček)

První regulací bylo polohové řízení, záznam z provedeného řízení je na Obr. 20. Jako referenční poloha byl nastaven obdélníkový cyklus, který každou sekundu změní požadovanou polohu výstupní hřídele motoru o 12 otáček. Pro odstranění šumu z hodnot naměřených enkodérem a hlavně pro získání časové derivace polohy byl použit kalmanův filtr, který je přímo integrován v MFU168. Časová derivace polohy je vstupem pro derivační složku regulace, která byla použita u polohového řízení. Její vliv je patrný např. před časem 3s na Obr. 20, kde můžeme pozorovat zmenšení úhlové rychlosti po přiblížení skutečné polohy k poloze referenční.

Na Obr. 21 lze pozorovat, že změna referenční polohy o 6 otáček nemá žádný vliv na kvalitu regulace. Zatížení ALU v MFU168 je 20%, tzn. že máme ještě rezervu ve výpočetním výkonu.



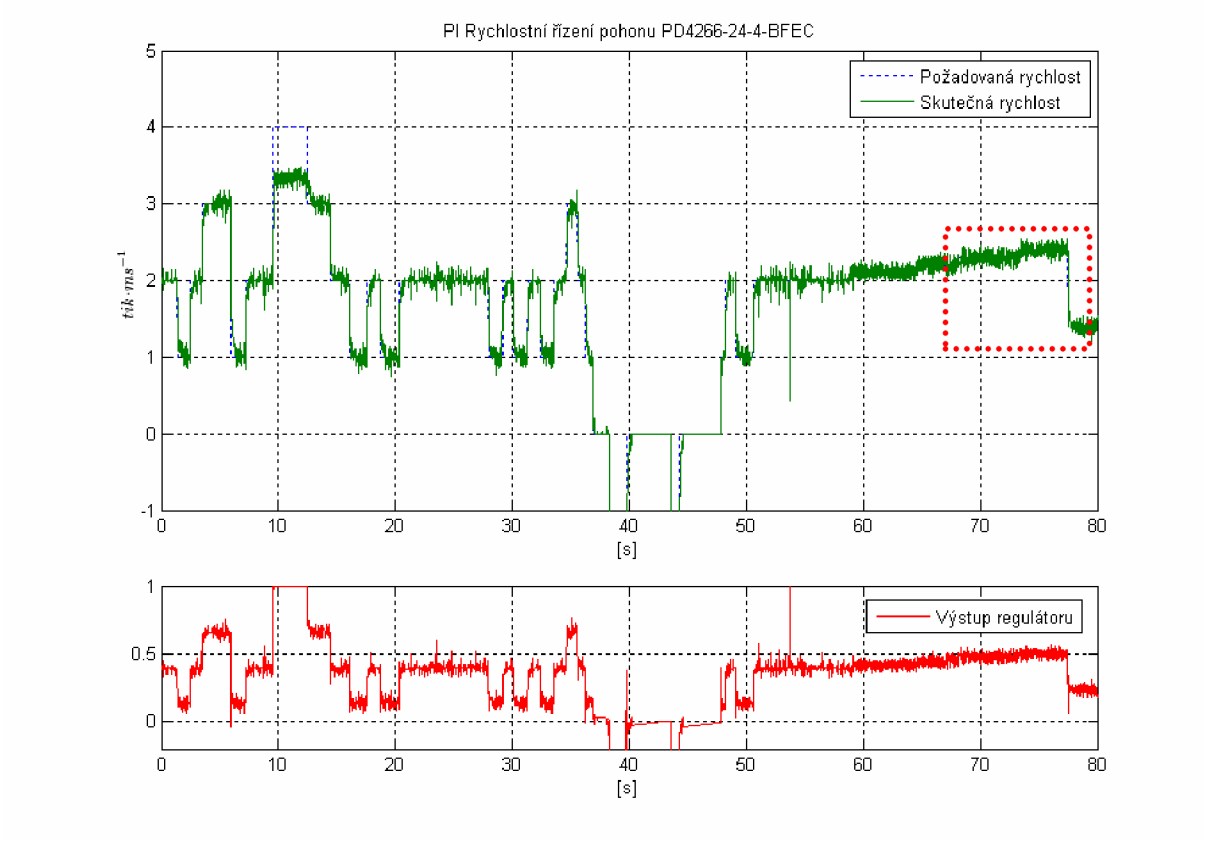
Obr. 21 PD Polohové řízení motoru PD4266-24-4 BFEC (3 otáčky)

Dalším typem je rychlostní řízení. Protože použitý senzor je polohový, k rychlostnímu řízení je nezbytné užít časovou derivaci polohy. Vzhledem k jinému typu řízení se mění parametry regulátoru a použité složky regulátoru. Pro realizaci derivační složky regulátoru je nutná druhá derivace polohy podle času. Protože při tomto rychlostním řízení není derivační složka významná a obecně derivace zvyšuje šum, nebyla derivační složka pro řízení použita. Při rychlostním řízení se však značně projevuje ustálená regulační odchylka, byla proto přidána integrační složka, která chybu odstraní. Ukázka rychlostního řízení je na Obr. 22. V čase 10s lze pozorovat, že skutečná rychlost se neblíží k referenční rychlosti, protože referenční rychlost je zde větší než maximální rychlost motoru.

Pro rychlostní regulaci motoru PD4266-24-4-BFEC byly použity následující příkazy:

Tab. 2: Rychlostní regulace motoru PD4266-24-4-BFEC

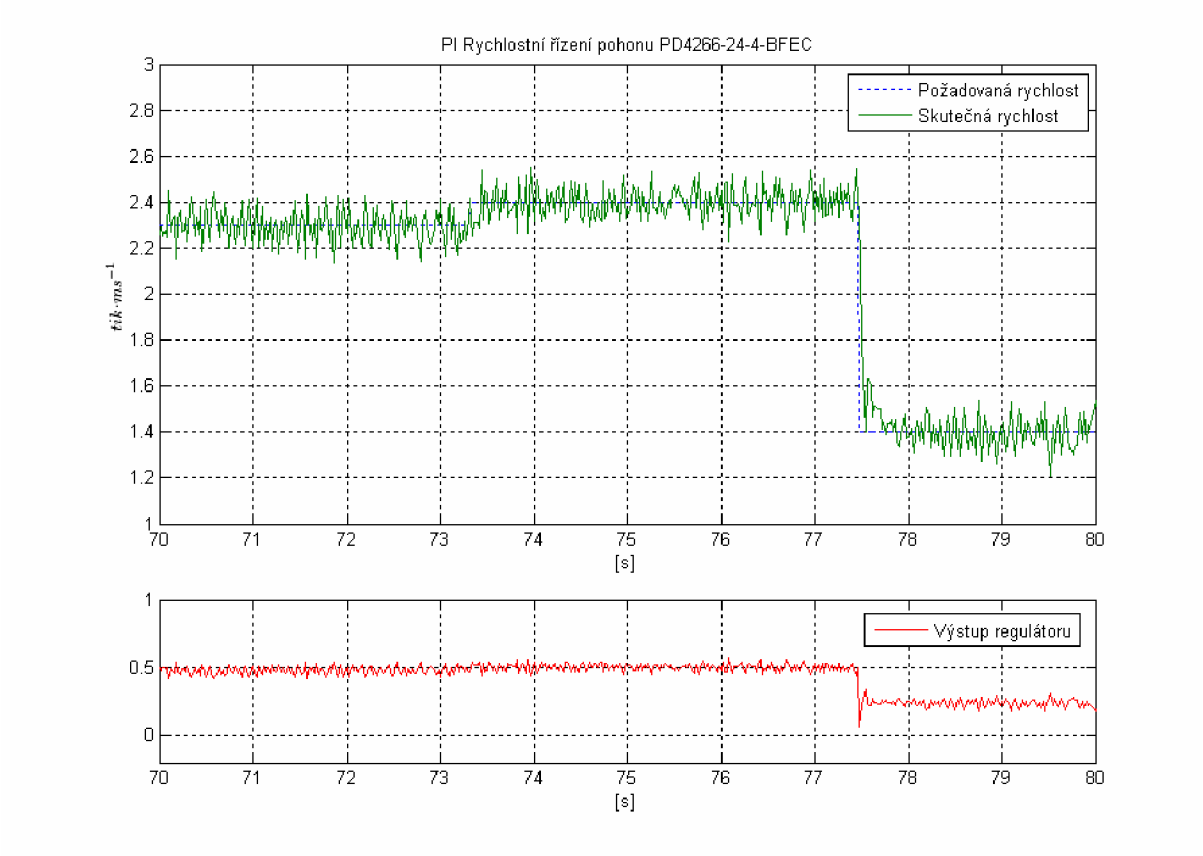
Příkaz	Popis
K1 0.1878	Zesílení pro kalmanův filtr
K2 0	Zesílení pro kalmanův filtr
vzorkovani 1	Řídicí smyčka-perioda=1ms
itg 1	Zesílení integrační složky regulátoru
prp 100	Zesílení proporcionální složky regulátoru
fderenky	Derivace polohy podle času



Obr. 22 PI Rychlostní řízení motoru PD4266-24-4 BFEC

Na Obr. 23 je zobrazen náhled rychlostního řízení z Obr. 22. Lze pozorovat, že skutečná rychlost osciluje kolem referenční rychlosti a regulační odchylka je díky použité integrační složce nulová. Zajímavé je, že výstupem ze senzoru jsou pouze celočíselné hodnoty pootočení, které určují,

o kolik bodů se hřídel motoru pootočila. Na Obr. 23 jsou to celá čísla na vertikální ose. To mimo jiné znamená, že při derivaci polohy podle času získáme také celočíselné hodnoty rychlosti a nebudeme schopni řídit rychlost motoru na jiné než tyto hodnoty. Použitím kalmanova filtru jsme však schopni naměřené hodnoty rychlosti odfiltrovat a určovat i na setiny. Tím jsme schopni řídit rychlost otáčení motoru daleko přesněji. To je patrné i z Obr. 23, kde vidíme, že skutečná hodnota neosciluje jen kolem rychlostí, která je násobkem celých čísel.



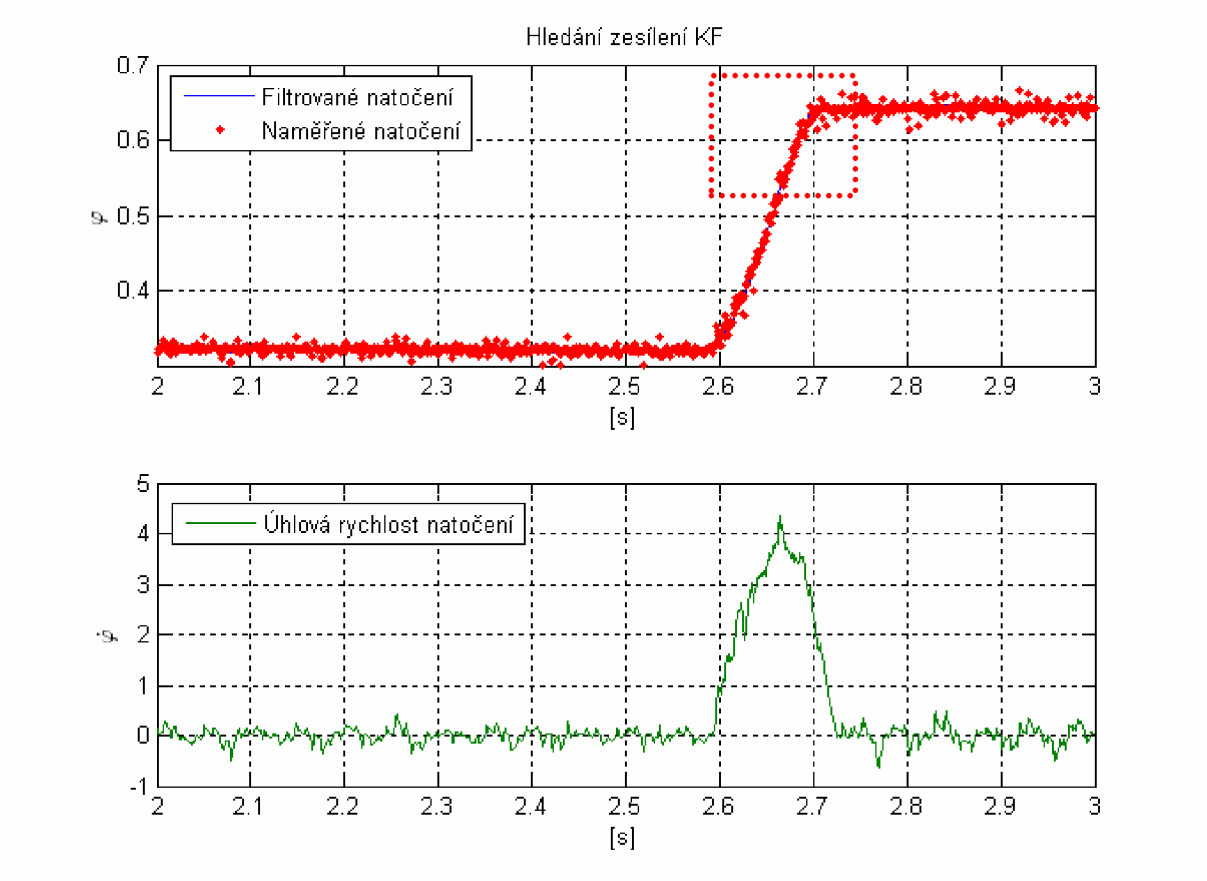
Obr. 23 PI Rychlostní řízení motoru PD4266-24-4 BFEC (detail)

Polohová a rychlostní regulace motoru PD4266-24-4 BFEC pomocí *Měřicí a řídicí jednotky* proběhla úspěšně. Podle předchozích grafů lze usoudit, že regulační odchylka byla ve všech případech v přijatelných mezích. S přihlédnutím na akční veličinu můžeme usoudit, že regulátor nemá tendence ke zbytečným nebo neuváženým regulačním zásahům a z hlediska životnosti motoru je regulace i po této stránce přijatelná. Výpočetní výkon MFU168 při regulaci a komunikaci s PC dosahoval okolo 20%, což znamená, že jsme nevyužili plný potenciál MFU168 a máme ještě rezervy pro případné použití složitějších konstrukcí regulátoru nebo podrobnější sledování stavů a proměnných v MFU168.

6.4 Regulace škrticí klapky

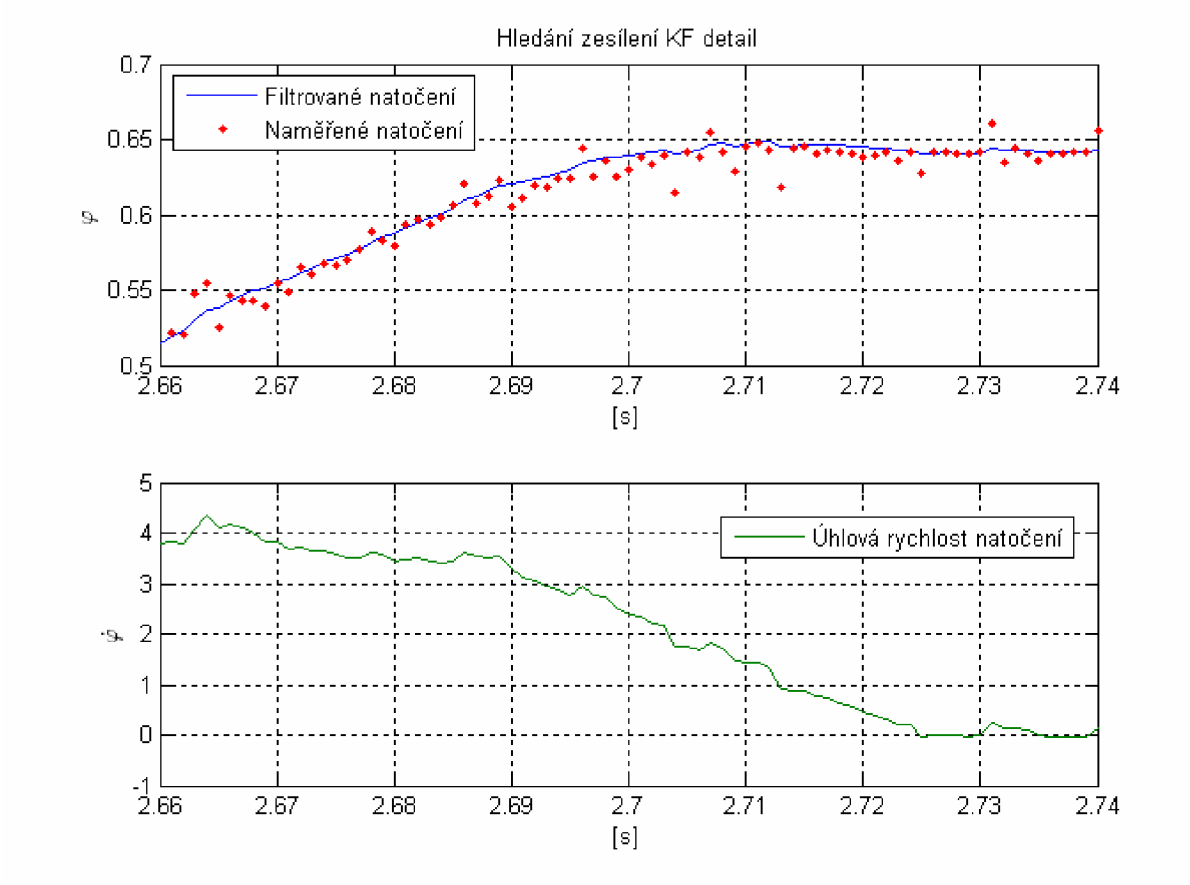
Pro regulaci byla použita škrticí klapka Magneti Marelli C146. Tato škrticí klapka obsahovala řídicí elektroniku, ale ta nebyla použita. Pro otáčení byly použity dva vodiče připojené přímo na motor ve škrticí klapce, které byly připojeny k H-můstku z přílohy III.2. Pro snímání aktuální polohy natočení byl využit jeden ze dvou potenciometrů, které jsou součástí škrticí klapky. Protože výstupní napětí na potenciometru trpí velkým šumem, bylo zapotřebí velmi pečlivě analyzovat data a určit koeficienty kalmanova filtru, který je součástí MFU168 a později z velké části šum odfiltruje.

Pro nalezení koeficientů KF budeme potřebovat vstupní data, která můžeme analyzovat. V první fázi tedy připojíme potenciometr k MFU168, se stejným vzorkováním, jaké bude použito při regulaci, budeme měřit polohu potenciometru a zároveň budeme ručně měnit polohu natočení klapky. Tím byla získána výchozí data k analýze. Pro snazší hledání koeficientů KF využijeme uživatelské rozhraní z přílohy II.1. Výchozí data jsou na Obr. 24. Z obrázku je patrné, že naměřené natočení trpí velkým šumem.



Obr. 24 Hledání koeficientů KF pro škrticí klapku

Pro bližší přiblížení byl vytvořen detail Obr. 24 v čase od 2,6s až 2,8s. V tomto detailu na Obr. 25 je jasně patrné, že filtrace kalmanovým filtrem velmi zlepšuje přesnost naměřené polohy natočení škrticí klapky. Průběh je plynulejší, což má příznivý vliv na regulaci, která nebude vnášet do soustavy zbytečné rázy, které by zkracovali životnost mechanických částí. Díky použitému filtru jsme schopni určit i aktuální úhlovou rychlost, kterou později použijeme v derivační složce PID regulátoru.



Obr. 25 Hledání koeficientů KF pro škrticí klapku (detail)

Po úspěšném nalezení koeficientů KF můžeme přistoupit k regulaci polohy škrticí klapky. K regulaci bude použit PID regulátor. Referenční poloha bude zpočátku ovládána ručně pomocí otočného ovladače, tato část bude simulovat plynulé analogové řízení. V druhé fázi bude zapnut obdélníkový referenční signál, který ověří regulaci při rychlých změnách referenční polohy.

Pro polohovou regulaci škrticí klapky byly použity následující příkazy:

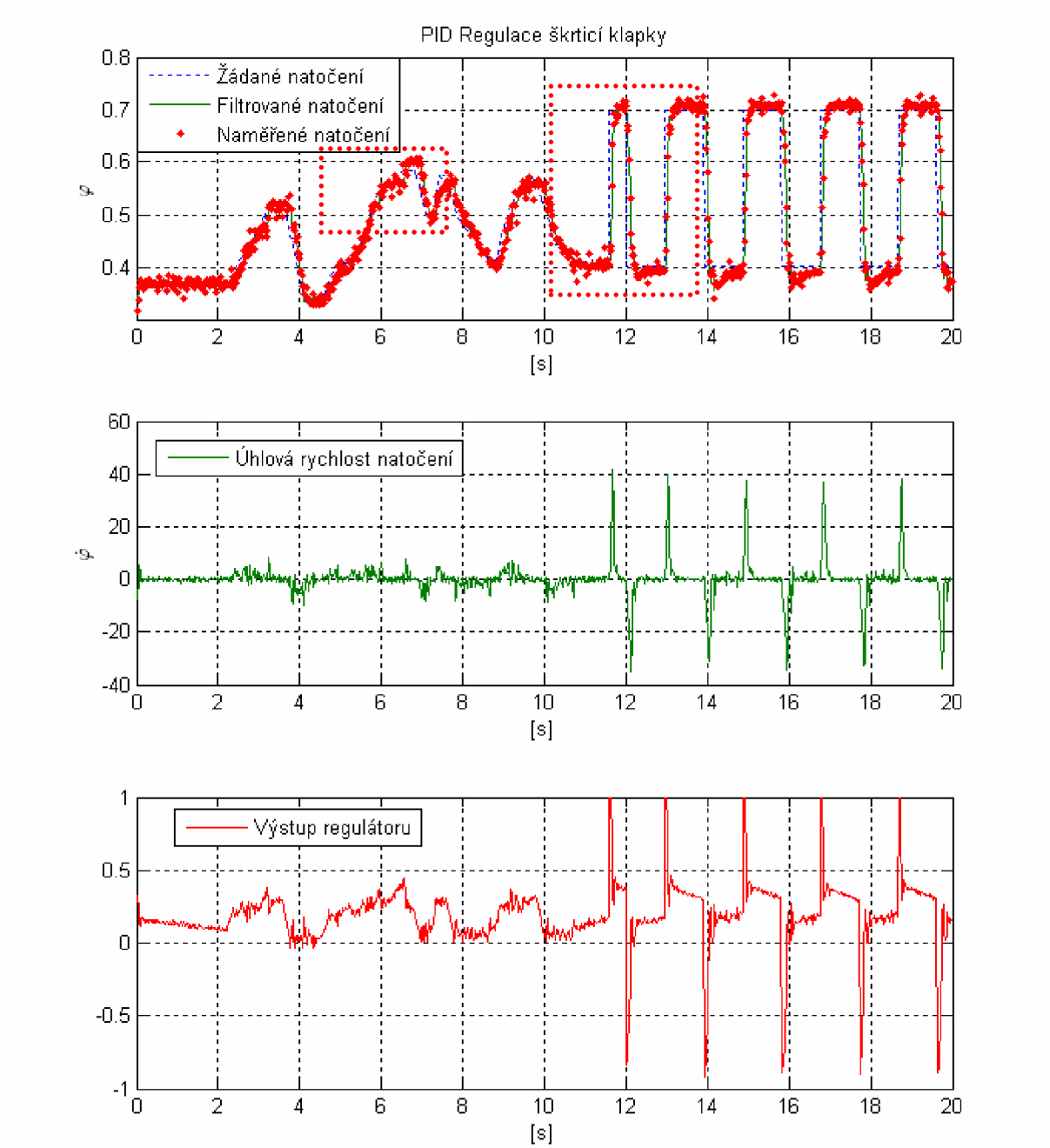
Tab. 3: Rychlostní regulace motoru PD4266-24-4-BFEC

Příkaz	Popis
prp 1000	Zesílení proporcionální složky regulátoru
itg 3	Zesílení integrační složky regulátoru
der -20	Zesílení derivační složky regulátoru
K1 0.0977	Zesílení pro kalmanův filtr
K2 5.013	Zesílení pro kalmanův filtr
gp 1000	Přepínání obdélníkové referenční hodnoty, změna každou 1s
gh 0.8	Horní poloha obdélníkové referenční hodnoty
gd 0.4	Dolní poloha obdélníkové referenční hodnoty

Sledování a nastavování regulace škrticí klapky probíhalo v uživatelském rozhraní z přílohy II.2. Záznam průběhu regulace je uveden na Obr. 26. Kolem času 11s byla referenční poloha přepnuta na obdélníkový signál. Osa y vyjadřuje natočení potenciometru škrticí klapky,

v normovaném rozsahu 0÷1. Skutečné pootočení je však díky dorazům v rozmezí 0,3÷0,75, kde 0,3 odpovídá zavřenému stavu a 0,75 otevřenému stavu. V poloze okolo hodnoty 0,42 je oblast tzv. LimpHome, kde zůstává klapka mírně otevřená v případě výpadku proudu. Pokud chceme škrticí klapku více otevřít nebo zavřít, musíme překonat sílu pružiny, která je součástí škrticí klapky.

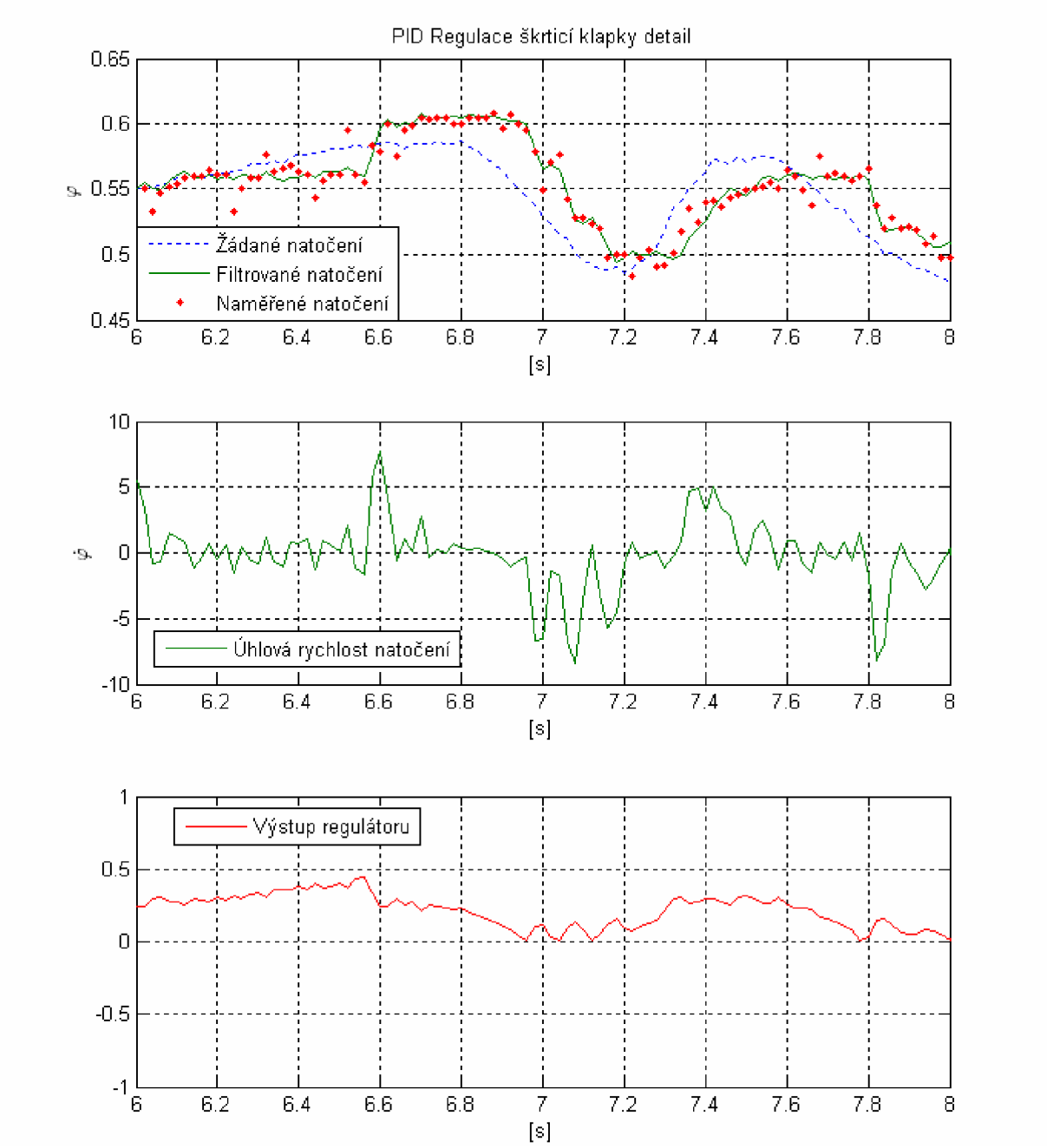
Z Obr. 26 je patrné, že regulace globálně zajišťuje sledování referenční polohy. Pro lepší představu byly vytvořeny detaily, první v rozmezí 6÷8s, druhý v rozmezí 12÷14s.



Obr. 26 PID regulace škrticí klapky

Při pohledu na detail na Obr. 27 můžeme pozorovat rozptyl naměřených hodnot pootočení škrticí klapky, ze kterého můžeme usoudit, že bez filtrace by byla regulace řádově horší. Soustava trpí suchým třením, které znatelně zhoršuje regulaci. Jeho projevy lze vidět např. v čase 6,6s na Obr. 27, kde skutečná poloha „poskočí“ prudce nahoru. Další projevy jsou kolem času 7s nebo 7,8s. S přihlédnutím k těmto vlastnostem degradujícím řízení můžeme však stále prohlásit regulaci

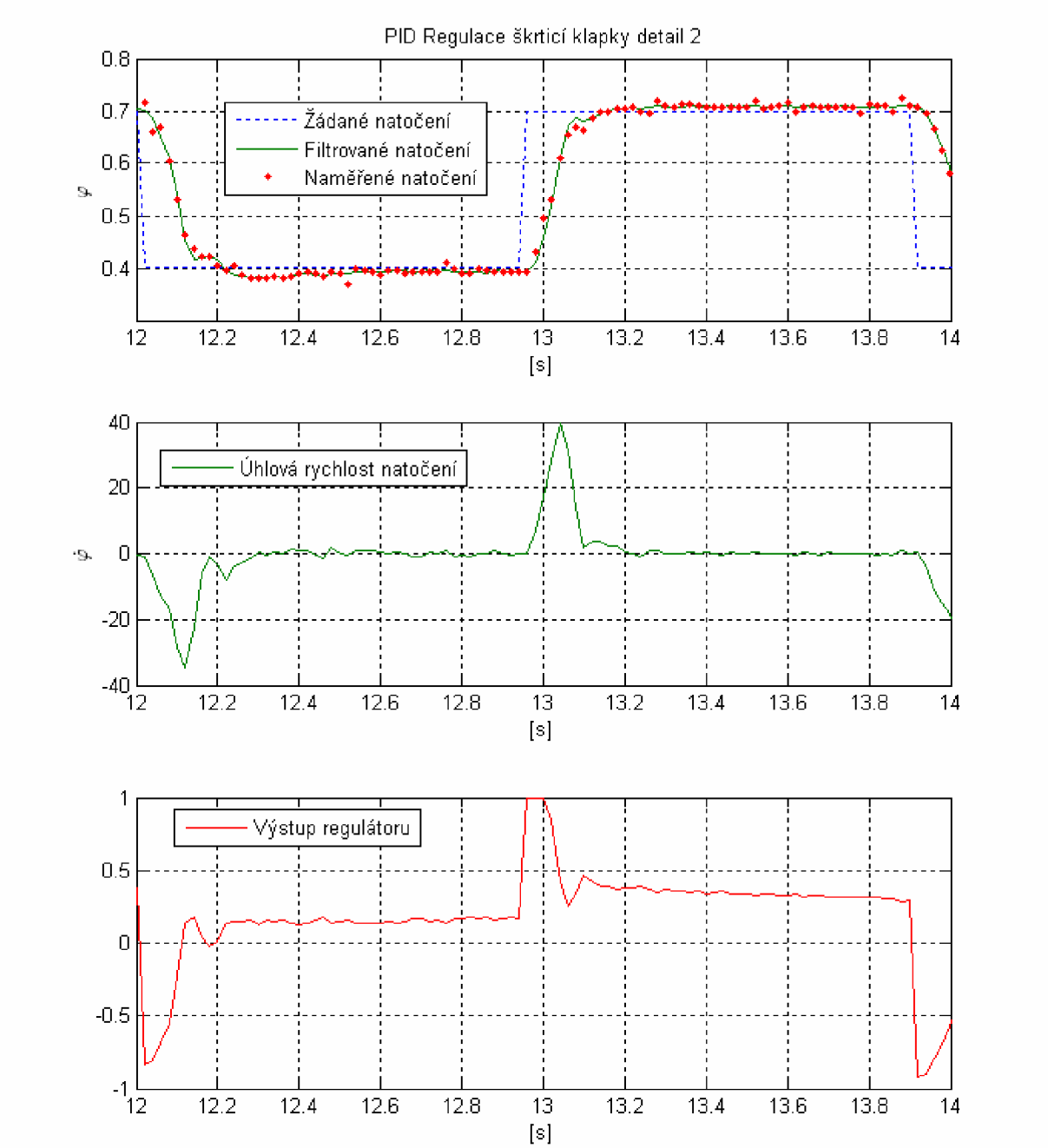
za uspokojivou, regulační odchylka je minimalizována a zpoždění skutečné polohy za referenční polohou je cca 0,1s.



Obr. 27 PID regulace škrticí klapky (detail 1)

V druhé fázi jsme testovali odezvu na skokovou změnu referenční polohy, detail odezvy je na Obr. 28. Při konstantní hodnotě referenční polohy se k ní velmi blíží skutečná poloha, odchylka je způsobena vysokým třením a k jejímu odstranění by bylo nutné použít sofistikovanější typ regulátoru. Při skokovém průběhu referenční polohy je dobře zřetelná derivační složka regulátoru, která začne brzdit pohyb po přiblížení skutečného natočení k referenčnímu natočení a dochází k plynulému pohybu. Tato vlastnost je vidět také v průběhu akčního zásahu (výstup regulátoru), který působí chvíli před „správnou“ polohou natočení proti původnímu pohybu. Výše bylo uvedeno, že škrticí klapka obsahuje pružinu, která působí proti otevření/zavření škrticí klapky. Její projevy jsou vidět v průběhu akčního zásahu, kde je nenulová hodnota akčního zásahu i pokud je škrticí

klapka v klidu. Tento nenulový akční zásah je úměrný natočení škrticí klapky, můžeme jej porovnat podle dvou konstantních poloh natočení na Obr. 28.



Obr. 28 PID regulace škrticí klapky (detail 2)

7 Výsledky a závěr

Cílem této práce bylo vytvořit Multifunkční záznamovou, měřicí a řídicí jednotku (Multi Function Unit, zkráceně MFU168) na bázi jednoduchého mikrokontroléru. Dle pokynů vedoucího BP bylo nutné splnit několik požadavků. Při realizaci MFU168 byl vytvořen funkční hardware (příloha IV), programátor (kap. 5.1, 5.2), komunikace s PC (kap. 5.3) a ovládací rozhraní (příloha II). MFU168 byla testována při různých aplikacích, avšak software i hardware je pro všechny totožný. Variabilita použití je ošetřena hardwarově i softwarově, pro každou úlohu je třeba zadat počáteční podmínky a nastavení, které MFU168 přizpůsobí pro danou úlohu. Toto nastavení lze provést pomocí PC nebo jej můžeme načíst z microSD karty. Tím je dosaženo širokého využití při minimálním objemu hardwaru i softwaru. Úprava i rozšíření funkcí MFU168 je také efektivnější, protože stačí upravovat pouze jeden hardware/software.

V kap. 6.3 byla testována regulace DC motoru. MFU168 obsahuje regulátor PID s některými vylepšeními. Všechny parametry regulátoru i nastavení periferií lze měnit v reálném čase a naměřené hodnoty můžeme sledovat v GUI na PC, kde jsou přehledně zobrazeny. Výběr měřených hodnot lze také provádět v reálném čase. Další aplikací bylo řízení škrticí klapky, průběhy regulace a zhodnocení je v kap 6.4.

Úpravou MFU168 je *Miniaturní měřicí jednotka*, viz. příloha IV.2. Ta obsahuje stejný software jako MFU168, ale velikost DPS je minimalizována. Jednotka obsahuje vestavěný akcelerometr. Ukázka měření akcelerometru je v kap. 5.4. Pro autonomní dlouhodobé měření byla využita MicroSD karta, záznam měření teploty po dobu 8h je v kap. 6.1.

K ovládání MFU168 byla vytvořena řada GUI, speciální GUI pro měření akcelerometru je v kap. 5.4, další specializované GUI pro hledání koeficientů Kalmanova filtru, který je integrován v MFU168, je v příloze II.1. Univerzální GUI pro nastavení MFU168 je v příloze II.2, v tomto GUI lze nastavit veškeré funkce, které MFU168 podporuje. Pro větší efektivitu práce je GUI adaptivní, po připojení MFU168 k PC se GUI synchronizuje a načte aktuální seznam funkcí, filtrů a proměnných dostupných v MFU168, viz příloha V. Zobrazení naměřenýh hodnot přijatých z MFU168 můžeme měnit řadou nástrojů, které univerzální GUI obsahuje.

Všechny body zadání práce byly splněny, byl vytvořen funkční hardware i software, jeho funkčnost byla úspěšně otestována a demonstrována na konkrétních příkladech.

8 Použitá literatura a další zdroje

[1] Implementation USB into microcontroller Igor Atmel-USB device: USB-RS232 converter + USB-8/16bit converter + USB-EEPROM scratch pad based on cheap AVR microcontroller.
Dostupný z WWW: <http://www.cesko.host.sk/IgorPlugUSB_RS232/IgorPlug-USB%20%28AVR%29%20RS232_eng.htm>.

[2] USBasp - USB programmer for Atmel AVR controllers
Dostupný z WWW: < http://www.fischl.de/usbasp/>.

[3] Blikáme LEDkou, Zbyněk Winkler a Martin Dlouhý, 2005-07-24, první praktické seznámení s jednočipem a elektronikou
Dostupný z WWW: < http://robotika.cz/guide/blink/cs >.

[4] Převodník USB2RS232 Cable
Dostupný z WWW: < http://www.gme.cz/cz/usb2rs232-p752-587.html >.

[5] ST VNH2SP30 30A Motor Driver
Dostupný z WWW: < http://www.pololu.com/catalog/product/537>.

[6] Akcelerometr MMA7361L DSH, datasheet

[7] Soldering and Mounting Guidelines for the LGA Accelerometer Sensor to a PC Board

Přílohy

I Základní popis MFU168

I.1 Úvod

Jednotka je schopna měřit analogové a digitální signály. Tyto naměřené hodnoty lze v jednotce dále filtrovat, nastavit je jako vstup zpětnovazební řídící smyčky nebo je odesílat do PC, kde je můžeme přehledně zobrazovat.

Po zapnutí jednotky je vše vypnuté. Činnost, kterou má jednotka vykonávat, je nutné nastavit z počítače. Komunikace PC s jednotkou probíhá formou textových příkazů. Nejjednodušším příkazem je „help;“, po jeho odeslání do jednotky se na PC vypíše seznam všech příkazů, kterými můžeme jednotku nastavovat a ovládat.

Všechna měření a nastavení jsou v jednotce uložena v proměnných. Do těchto proměnných se ukládají naměřené veličiny a další hodnoty potřebné k nastavení jednotky. Pokud chceme sledovat měřenou veličinu, musíme zapnout odesílání proměnné, do které se tato veličina ukládá. Pro nastavení parametrů, např. PID regulátoru, jednoduše nastavíme proměnné „P“ „I“ a „D“. Regulátor pak s těmito proměnnými automaticky počítá při regulaci. Tyto změny lze provádět i během regulace nebo měření, není nutné nijak přerušovat nebo pozastavovat probíhající činnost.

Textové příkazy poslané do jednotky v ní spouštějí funkce. Tyto funkce můžou vykonávat libovolnou činnost od nastavování proměnných až po zapínání periferií.

I.2 Počáteční stav

Po zapnutí napájení je jednotka plně vyresetovaná, většina nastavení je vypnutá a stavové proměnné jsou vynulované.

I.3 Komunikace s MFU168

Komunikace probíhá přes asynchronní sériovou linku. Jednotka přijímá textové příkazy, které musí být ukončeny znakem „;“,např. „funkce;“. Pokud je nutno zadat příkaz i s parametrem, oddělovací znak je mezera „ “ , forma příkazu je „funkce parametr;“. V případě více parametrů se parametry oddělují čárkou „,“, např. „funkce parametr1,parametr2;“.

I.4 Základní funkce

Jednotka funguje v nekonečné smyčce, která v každém kroku vykonává jednotlivé funkční bloky (měření, filtrace, regulace,...). Perioda je nastavena proměnou ‚vzorkovani‘, nejmenší perioda je 1ms a odpovídá ‚vzorkovani=1‘. Každý z těchto bloků lze zapnout/vypnout.

I.5 Bloky v MFU168

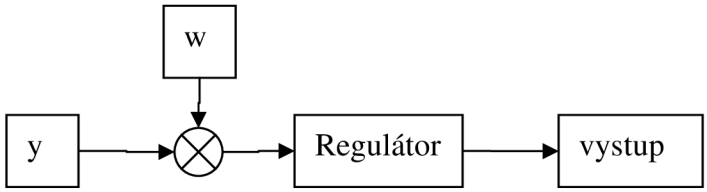
Měření

Jednotka je schopna měřit analogový signál, výstup z enkodéru nebo digitální signál. Všechny způsoby měření lze zavolat příkazem z PC, který jej v jednotce aktivuje. Přesnost měření, místa uložení hodnot apod. jsou popsány dále v popisu příkazů.

Regulace

Regulátor porovnává hodnotu uloženou v proměnné ‚y‘ a žádanou hodnotu uloženou v proměnné ‚w‘. Po provedení vlastní regulace nastaví výstupní hodnotu do proměnné ‚vystup‘. Regulátor lze

nastavovat (např. parametry P,I,D), nebo můžeme nastavit jiný regulátor. Regulátor lze zapnout např. příkazem „rPID;“.



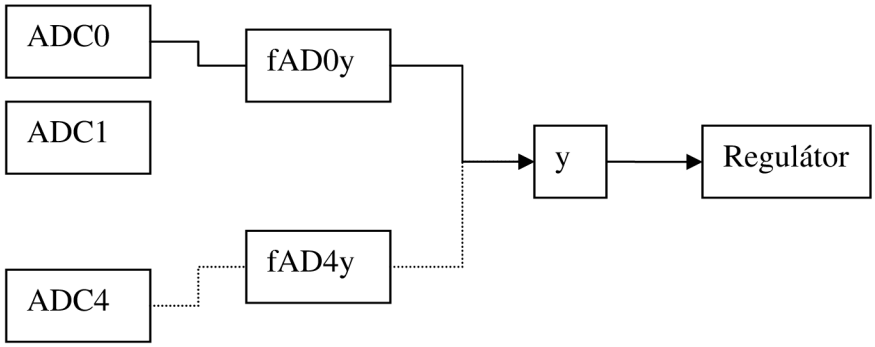
Filtrace

Filtrace je použita pro manipulaci s daty. Jednotka obsahuje sadu proměnných, se kterými můžeme manipulovat skrze filtry.

Příklad:

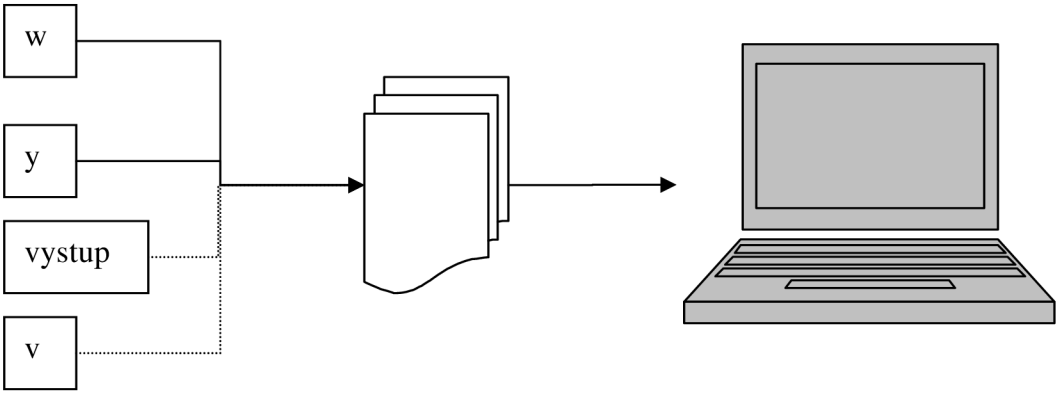
Měříme analogovou hodnotu (třeba napětí na potenciometru zapojené na vstupní pin ADC0) a chceme ji použít jako vstup do regulátoru. Naměřená hodnota se uloží do proměnné ‚AD0‘. Pokud ji chceme použít jako vstup regulátoru, musíme ji propojit se vstupní proměnou regulátoru, která má název ‚y‘. Propojení obou proměnných můžeme dosáhnout zapnutím filtru „fAD0y;“. Ten zkopíruje hodnotu v proměnné ‚AD0‘ a uloží ji do proměnné ‚y‘. Proč je nutné naměřenou hodnotu takto propojovat přes filtr a neukládá se rovnou do proměnné ‚y‘? Pokud bychom zapojili potenciometr na vstupní pin ADC4 místo jako v předchozím případě na ADC0, stačí pak zapnout jiný filtr „fAD4y“ a vše bude fungovat správně. Tímto způsobem si můžeme vybírat, odkud se bude číst vstupní hodnota regulátoru.

Další možností je zapnout jiný filtr, který odstraní šum z měření. Tzn. že např. na naměřenou hodnotu v proměnné ‚y‘ aplikujeme filtraci klouzavým průměrem. Možnosti jsou velmi rozsáhlé.



Odesílání dat

Pro sledování nějaké veličiny je zde blok odesílání dat. Vybraná veličina se neustále odesílá do PC v nastaveném intervalu. V PC tak lze sledovat průběh například napětí na motoru, nebo pootočení hřídele. Lze odesílat libovolné množství veličin. Např. pro sledování požadované výchylky a skutečné výchylky je třeba odeslat příkaz „odesilatmsk w,y;“ (w – název proměnné pro žádanou výchylku, y – název proměnné pro skutečnou výchylku).



Výstup

Jsou různé typy výkonových zařízení a každé má specifické požadavky. V bloku výstup se podle zvoleného režimu projeví hodnota uložená v proměnné „vystup“ na výstupních pinech jednotky. Např. pro použití H-můstku se vstupy PWM a DIR zadáme v PC příkaz „PWMDIR;“. Ten v jednotce aktivuje funkci, která nastaví potřebné periferie a na výstupní pin PWM ukládá absolutní hodnotu z proměnné ‚vystup‘, na výstupní pin DIR ukládá znaménko z proměnné ‚vystup‘, tzn. směr.

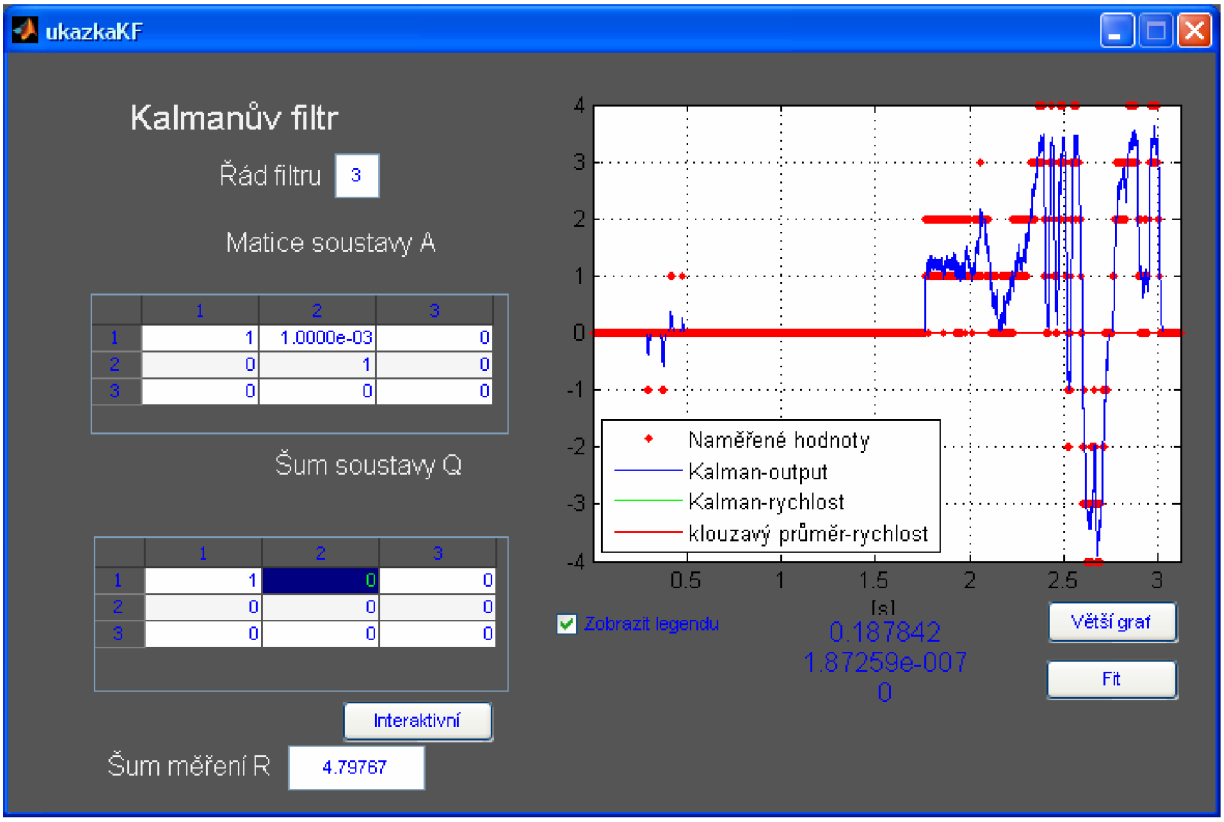
II Uživatelské rozhranní

II.1 Uživatelské rozhraní pro nalezení koeficientů KF

Obvykle trpní naměřená data velkým šumem. V MFU168 je implementován kalmanův filtr, který je uzpůsoben pro odfiltrování šumu a popř. numerickou derivaci naměřených dat. Problém však nastává v určení správných koeficientů filtru. Pro snazší hledání koeficientů kalmanova filtru bylo vytvořeno GUI (Obr. 29), které umožňuje měnit parametry filtru a v reálném čase zobrazovat výsledný průběh filtrovaných dat. Spuštění aplikace lze provést z příkazové řádky MATLABu:

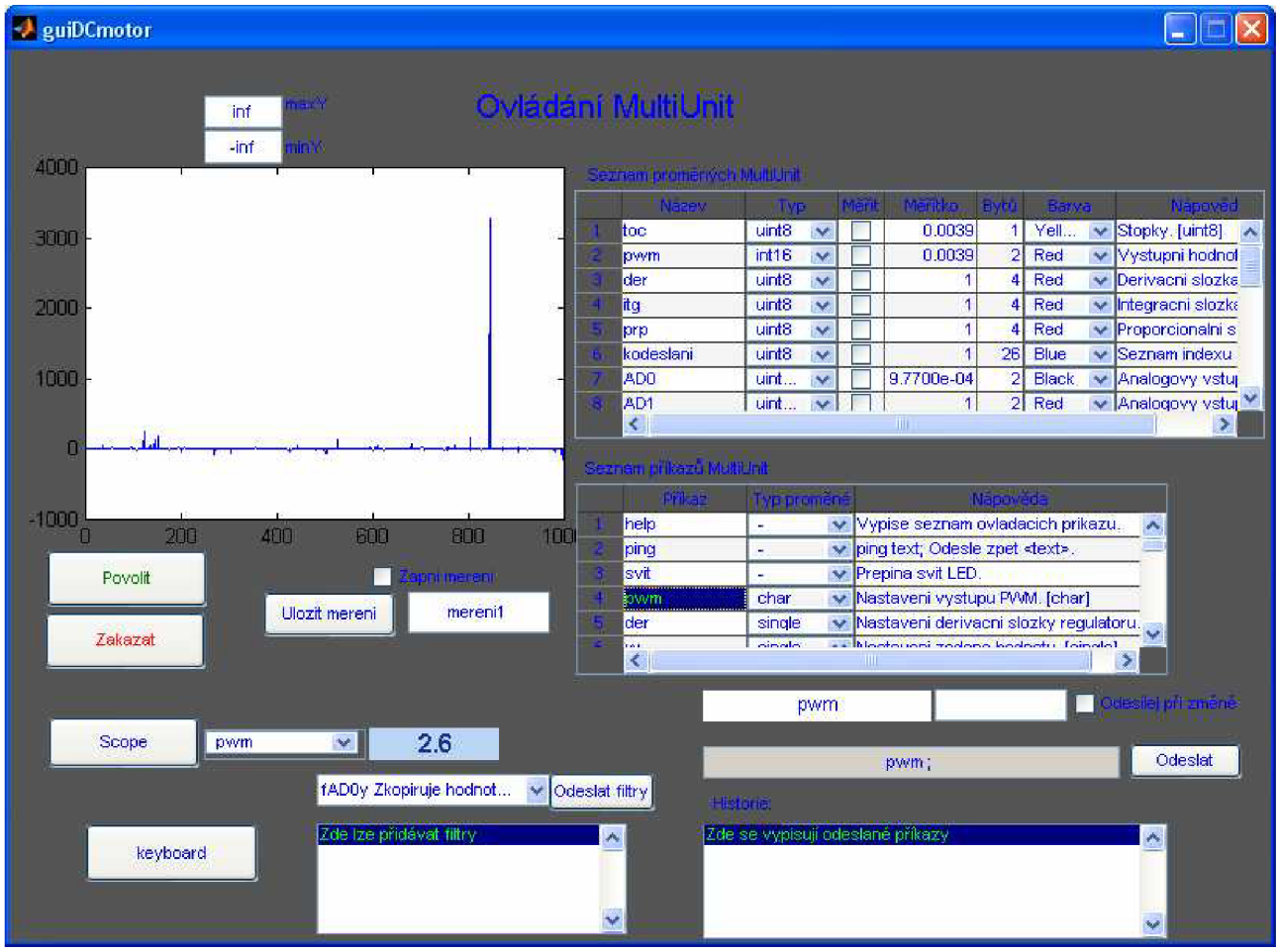
```
KF(namerena_data,casovy_krok)
```

Poté nastavíme v uživatelském rozhraní parametry. Po dokončení úprav a zavření GUI se výsledné koeficienty objeví v base workspace jako proměnná ‚K'.



Obr. 29. Uživatelské rozhraní pro nalezení koeficientů KF

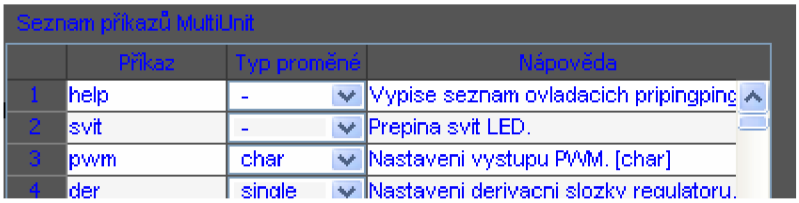
II.2 Uživatelské rozhraní pro ovládání MFU168



Obr. 30. Uživatelské rozhraní pro ovládání MFU168

Komunikace MFU168 byla navržena na bázi textových příkazů, které nejsou závislé na operačním systému nebo speciálním softwaru. Tím je zajištěna funkce v různorodém prostředí, ale pro nového uživatele se stává systém nepřátelským. Proto bylo navrženo uživatelské prostředí v jazyce MATLAB, které umožní snazší zadávání příkazů a ovládání MFU168.

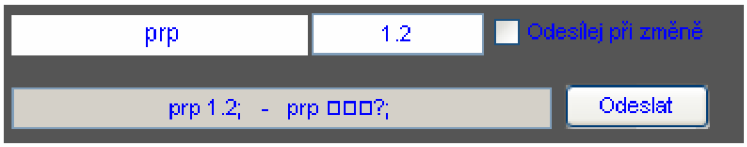
V tomto prostředí lze nalézt všechny příkazy pro ovládání MFU168, viz Obr. 31. Automaticky se obnovují po spuštění, takže pokud by byli v MFU168 nějaké další přídány/odebrány, i v tomto seznamu se přidají/odeberou při startu aplikace. Součástí seznamu příkazů je krátká nápověda, která je uložena přímo v MFU168 a obnovuje se s příkazy po spuštění GUI.



Obr. 31. Seznam příkazů MFU168

Další kolonkou v seznamu příkazů je Typ proměnné. Zde se nastavuje způsob, jakým GUI interpretuje parametry zadané před odesláním příkazu do MFU168. Pokud necháme „-“ pomlčku, příkaz je bez parametrů.

Výběrem myši nebo klávesnicí aktivujeme v seznamu příkazů položku. Aktivní příkaz se zobrazuje pod seznamem, vybereme např. příkaz pro nastavení proporcionálního zesílení regulátoru, viz. Obr. 32. Do políčka vpravo můžeme zadat parametr pro zesílení.

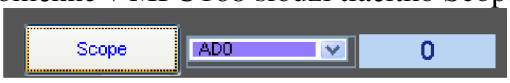


Obr. 32. Zadávání parametrů a odesílání příkazů

Výsledný textový řetězec vznikne složením příkazu a parametru a zobrazí se v šedém poli. MFU168 podporuje i binární zadávání parametrů, to je však pro člověka nečitelné, proto je v šedém poli první informativní tvar výsledného příkazu a za pomlčkou je skutečný tvar, který bude po stisku tlačítka Odeslat odeslán do MFU168.
Všechny odeslané příkazy se zobrazují v historii, která se nachází vpravo dole, viz. Obr. 33.

Obr. 33. Historie odeslaných příkazů

Pro zjištění hodnoty nějaké proměnné v MFU168 slouží tlačítko Scope (Obr. 34).

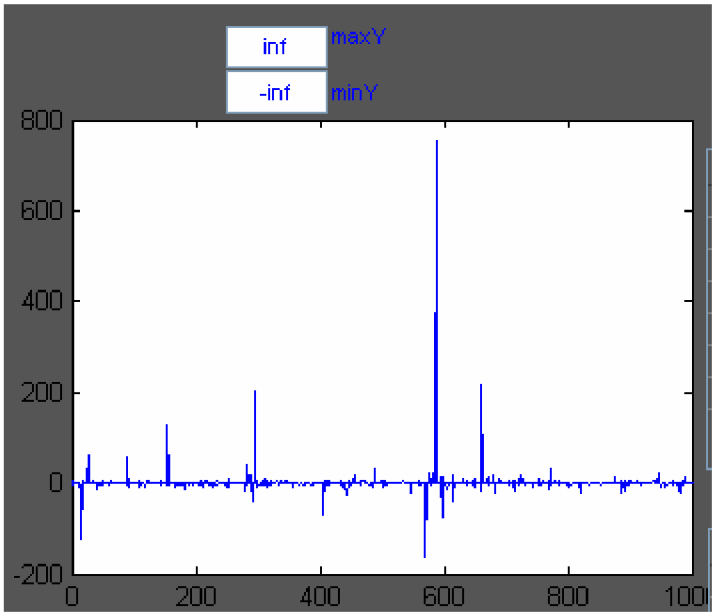


Obr. 34. Zjištění hodnoty proměnné

Kterou proměnnou chceme číst vybereme z rozbalovacího menu (Obr. 35), po stisku tlačítka Scope se hodnota proměnné objeví v modrém poli vpravo (Obr. 34).

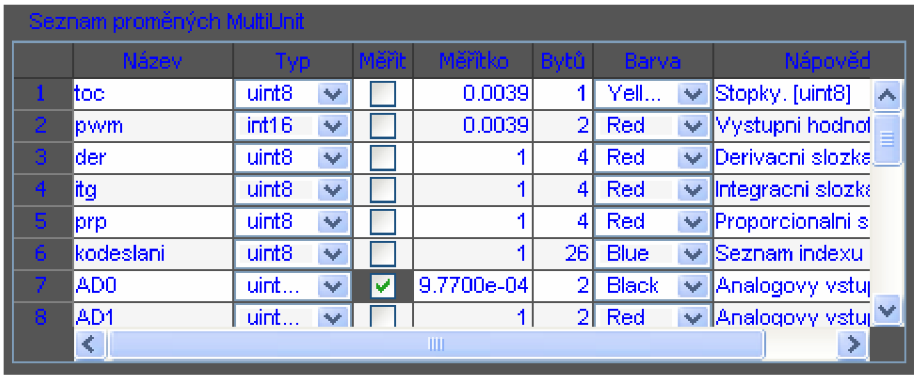
Obr. 35 Rozbalovací seznam proměnných

Pro dlouhodobé měření veličin nebo zobrazování proměnné je více vhodný graf (Obr. 36). Na ose x je zobrazen počet přijatých hodnot, v případě většího počtu se začne osa rolovat a zobrazovat pouze poslední relevantní data. Meze na ose y jsou nastaveny editačními poli, které se nachází nad grafem (Obr. 36 nahoře).



Obr. 36 Integrovaný graf

Výběr, které proměnné chceme měřit, se provádí v seznamu proměnných (Obr. 37). Ve sloupci Měřit zaškrtnutím zvolíme proměnné, lze volit i více najednou. Rozhranní automaticky odešle konfigurační údaje do MFU168 (pokud je připojená) a data se začnou zobrazovat v grafu (Obr. 36). Pro lepší orientaci ve více měřených datech lze u každé proměnné volit měřítko a barvu, kterou se budou zobrazovat. Ve sloupci Typ je nutno zvolit typ proměnné, aby data přijímaná z MFU168 byla správně interpretována, lze totiž přijímat jak znaménková a neznaménková čísla, tak čísla s plovoucí desetinnou čárkou. Všechna nastavení jsou před zavřením GUI uložena a není je nutné zadávat znovu při spuštění.

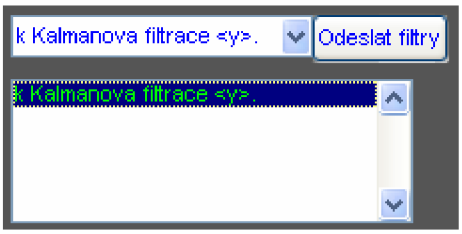


Obr. 37 Seznam proměnných v MFU168

Export naměřených dat lze provést tlačítkem Ulozit mereni nacházejícím se pod grafem (Obr. 38). Naměřená data se uloží do base workspace, název proměnné definujeme v poli vedle tlačítka Ulozit mereni. V base workspace můžeme po ukončení GUI Ovládací rozhraní MFU168 dále pracovat s naměřenými daty, vytvořit lépe graficky upravené grafy nebo data exportovat a ukládat na disk.

Obr. 38 Uložit měření

MFU168 obsahuje řadu filtrů, jejich seznam je automaticky aktualizován po spuštění GUI (Obr. 38). Ze seznamu můžeme vybírat postupně, které filtry budou v MFU168 aktivní. Po stisku tlačítka Odeslat filtry se nastavení odešle do MFU168.



Obr. 39 Seznam filtrů MFU168

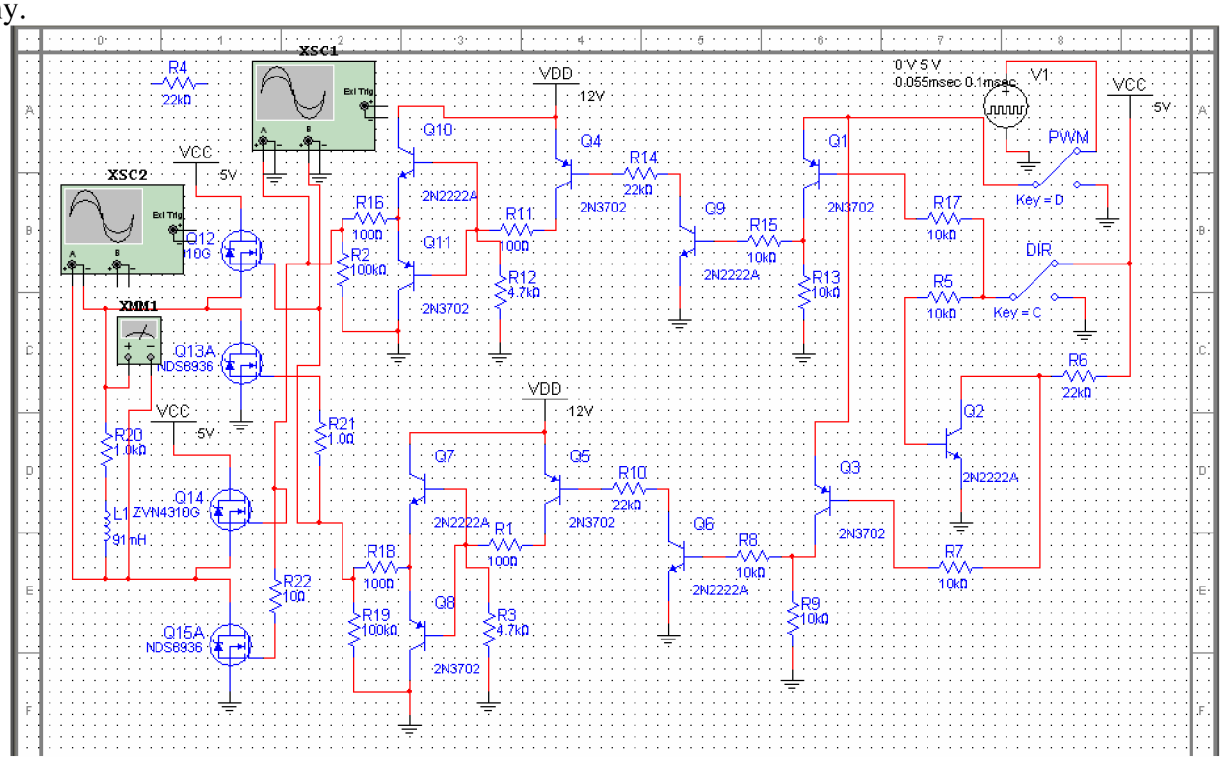
Posledním ovládacím prvkem v uživatelském rozhraní a možná nejdůležitějším jsou tlačítka Povolit a Zakazat (Obr. 40). Zakazat slouží k okamžitému pozastavení činnosti MFU168, velmi vhodné při nestandardních anebo kritických stavech. Povolit naopak spouští činnost MFU168.

Obr. 40 Povolit - Zakazat

III H-můstky

III.1 Diskrétní součástky (tranzistory MOSFET)

První z variant bylo sestavit a vyzkoušet výkonový člen vytvořený z diskrétních součástek. Na internetu se nachází celá řada zapojení a doporučení. Ideální řešení vhodné právě pro tuto aplikaci bylo nakonec složeno z několika návodů a návrhů. Protože se jedná o nové a nevyzkoušené zapojení, byla vhodné před samotným sestavením obvodu provést simulaci a doladit celé zapojení. Pro tento účel byl využit program pro simulaci elektrických obvodů MultiSim 9. Na Obr. 41 je ukázka simulačního schématu, v programu byl obvod testován na různé typy zatížení, řízení a poruchy.



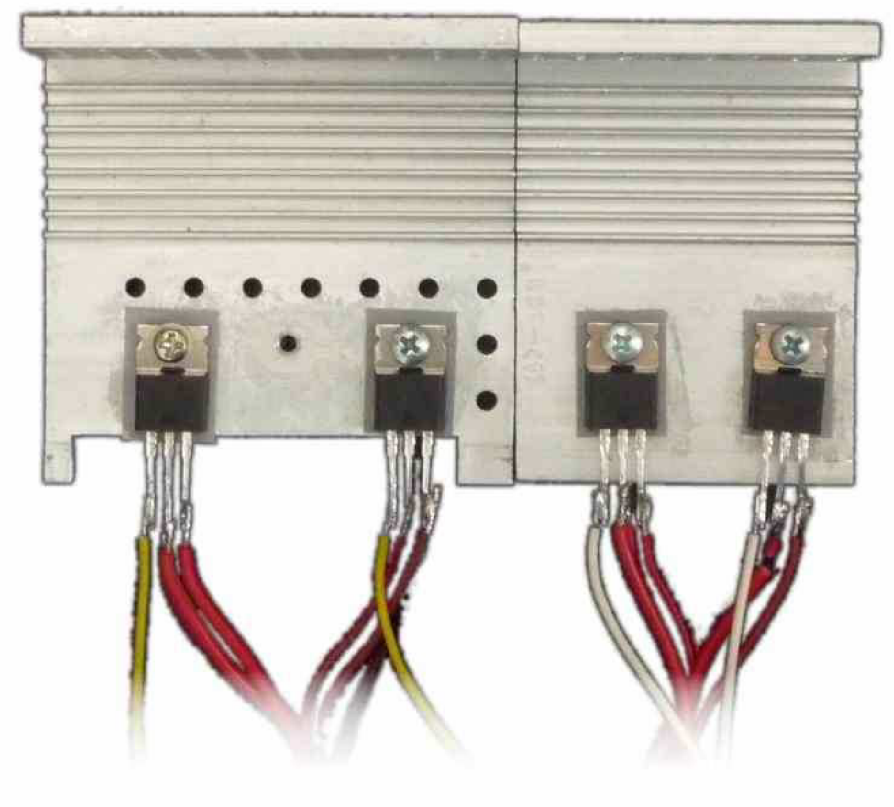
Obr. 41 Simulace v programu MultiSim 9

Výkonový obvod se skládá ze čtyř tranzistorů typu MOSFET, přesné označení je IRF3205. Při bližším pohledu na simulační schéma na Obr. 41 lze oponovat, že se jedná o jiné tranzistory. To je pravda, avšak tranzistor IRF3205 je nový typ, který aplikace MultiSim 9 zatím v knihovně neobsahuje. V simulaci byl tedy použit jiný tranzistor, avšak jeho parametry se s použitým tranzistorem téměř shodují.

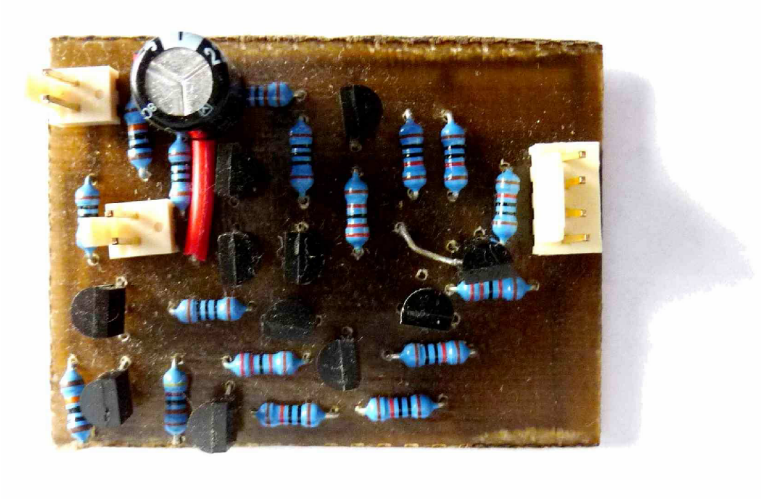
Tab. 4: Parametry tranzistoru IRF3205:

VDSS	55V
RDS(on)	8.0mΩ
I_D	110A

Po dokončení simulací byl navržen plošný spoj a začala fáze testování. Pro výkonové tranzistory byl použit dostatečně velký chladič a přívodní kabely (Obr. 42), můstek je dimenzován na proudy až 20A. Řídicí elektronika (Obr. 43) byla umístěna dále od tranzistorů, aby se snížilo rušení vlivem velkých proudů tekoucích ve výkonové části.



Obr. 42 Výkonová část s tranzistory IRF3205

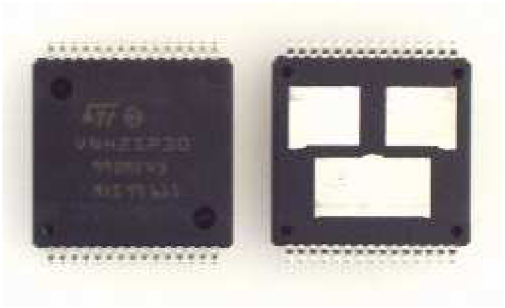


Obr. 43 Budič

Výroba obvodu probíhala v domácích podmínkách, všechny potřebné součástky jsou běžně dostupné v ČR. Cena součástek je do 150Kč. Obvod byl úspěšně testován a jeví se jako funkční. Nevýhodou může být absence proudové ochrany. Ta je z části kompenzována zvolenými tranzistory, které mají maximální proud 110A a při uvažovaném napětí do 12V nebude takového proudu nikdy dosaženo. Nevýhoda obvodu je jeho jednoduché zapojení, jež nenabízí žádné další funkce a ochrany, které se vyskytují u výkonových integrovaných obvodů. Tato nevýhoda je však i výhodou, neboť při nesprávném zacházení není nutné měnit celý výkonový prvek, ale stačí jen ten zničený (např. proražený tranzistor na vstupu po připojení obrácené polarity napájecího napětí).

III.2 ST VNH2SP30 30A Motor Driver

Při průzkumu trhu byl zvolen jako vhodný integrovaný výkonový obvod VNH2SP30 od firmy ST Microelectronics. Tento obvod nabízí dostatečný výkon pro řízení modelů a motorů uvažovaných v této práci. Obsahuje proudovou ochranu, podpěťovou a přepěťovou ochranu, tepelnou ochranu a další. Řízení probíhá přes logiku TTL, je tedy možné integrovaný obvod připojit bez převodníku přímo k jednočipu. Cena je řádově příznivých 100-150Kč. Nevýhodou je však pro domácí výrobu pouzdro, které je typu SMD a to zhoršuje úspěšné zprovoznění. Při menších proudech řádově do 3A je možné připojit pouze piny na okraji pouzdra, avšak při větších proudech je nutné připájet i spodní plošky Obr. 44 vpravo. Tím dosáhneme většího průřezu vodiče pro výkonovou část a také lepší odvod tepla z integrovaného obvodu. Zapájet spodní plošky značně ztěžuje výrobu v domácích podmínkách, ale není nerealizovatelné.

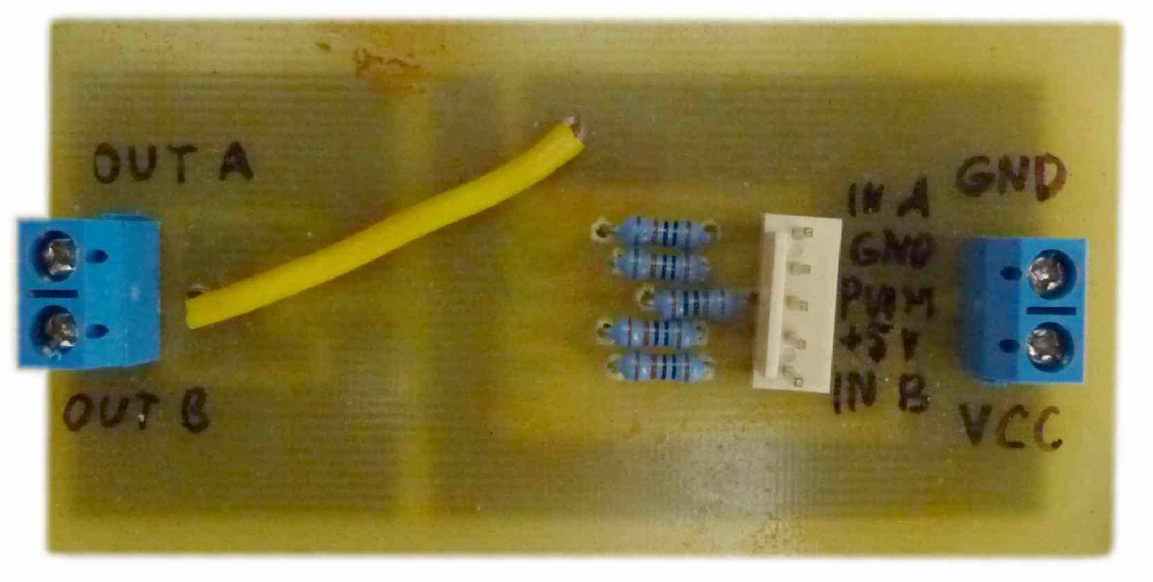


Obr. 44 Pouzdro H-můstku VNH2SP30

Bylo navrženo a vyrobeno DPS výkonové části. Horní strana DPS je vyfocena na Obr. 45, v pravé části se nachází šroubovací svorky pro zdroj napětí (označeno VCC a GND), dále vlevo je zásuvka se zámkem pro řídicí signály (označeno InA, GND, PWM, +5V a InB) a na levé straně jsou šroubovací výstupní svorky (označeno outA a outB).

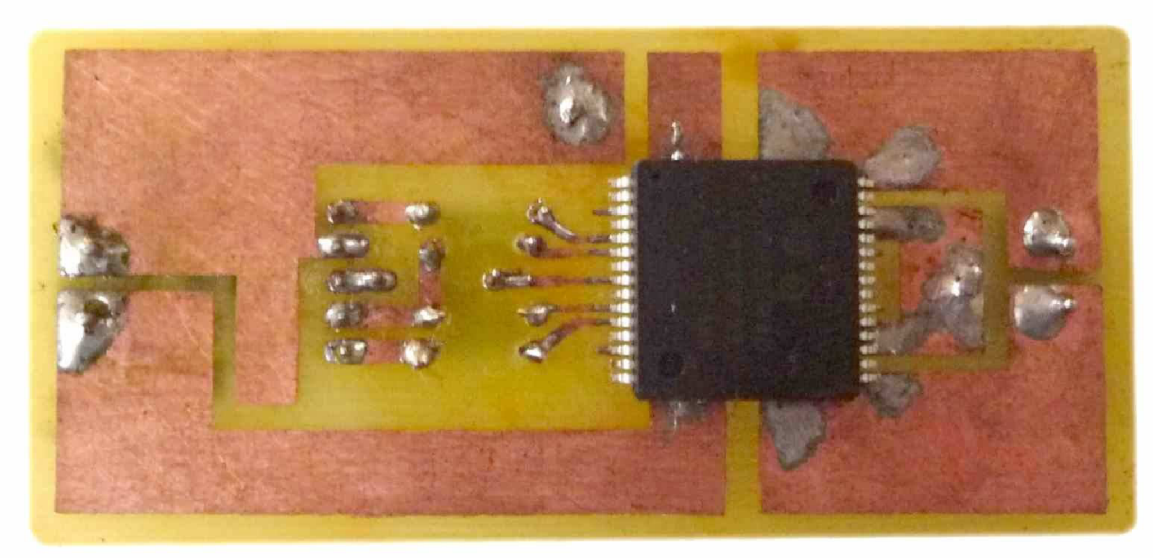
Tab. 5: Logika řízení H-můstku VNH2SP30

IN_A	IN_B	OUT_A	OUT_B	Operating mode
1	1	H	H	Brake to V_{CC}
1	0	H	L	Clockwise (CW)
0	1	L	H	Counterclockwise (CCW)
0	0	L	L	Brake to GND



Obr. 45 Horní strana DPS H-můstku VNH2SP30

Spodní strana DPS je vyfocena na Obr. 46, s ohledem na velké proudy byly navrženy široké cesty na DPS. Na fotografii lze místy pozorovat rozteklý cín kolem pouzdra H-můstku, je to z důvodu pájení spodních plošek pouzdra integrovaného obvodu.



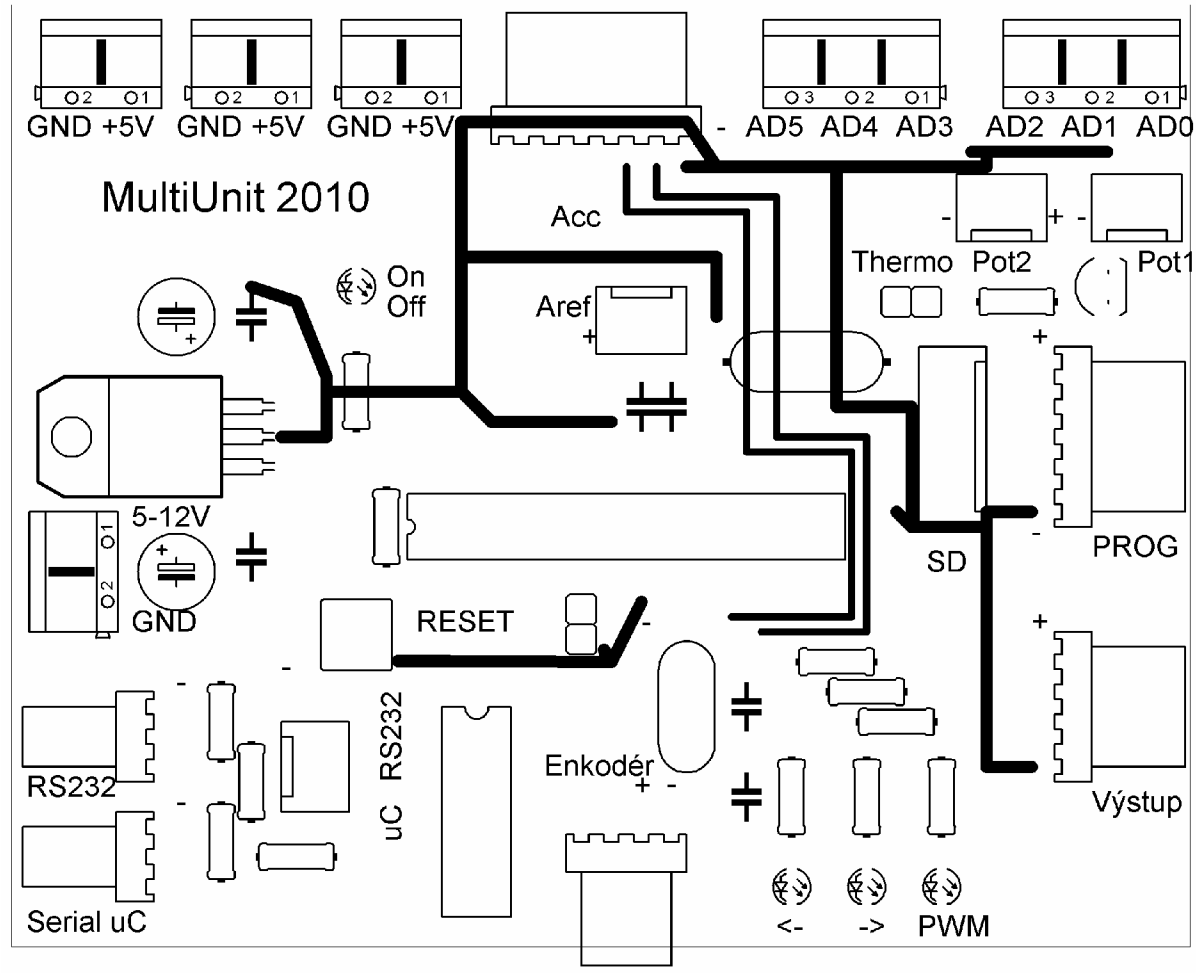
Obr. 46 Spodní strana DPS H-můstku VNH2SP30

Tab. 6: Tabulka parametrů H-můstku [5]

	VNH3SP30	VNH2SP30
Operating supply voltage (Vcc)	5.5 – 36 V	5.5 – 16 V
Maximum current rating	30 A	30 A
MOSFET on-resistance (per leg)	34 mΩ	19 mΩ
Maximum PWM frequency	10 kHz	20 kHz
Current sense	none	approximately 0.13 V/A
Over-voltage shutoff	36 V	16 V minimum (19 V typical)
Time to overheat at 20 A	8 seconds	35 seconds
Time to overheat at 15 A	30 seconds	150 seconds
Current for infinite run time	9:00 dop.	14 A

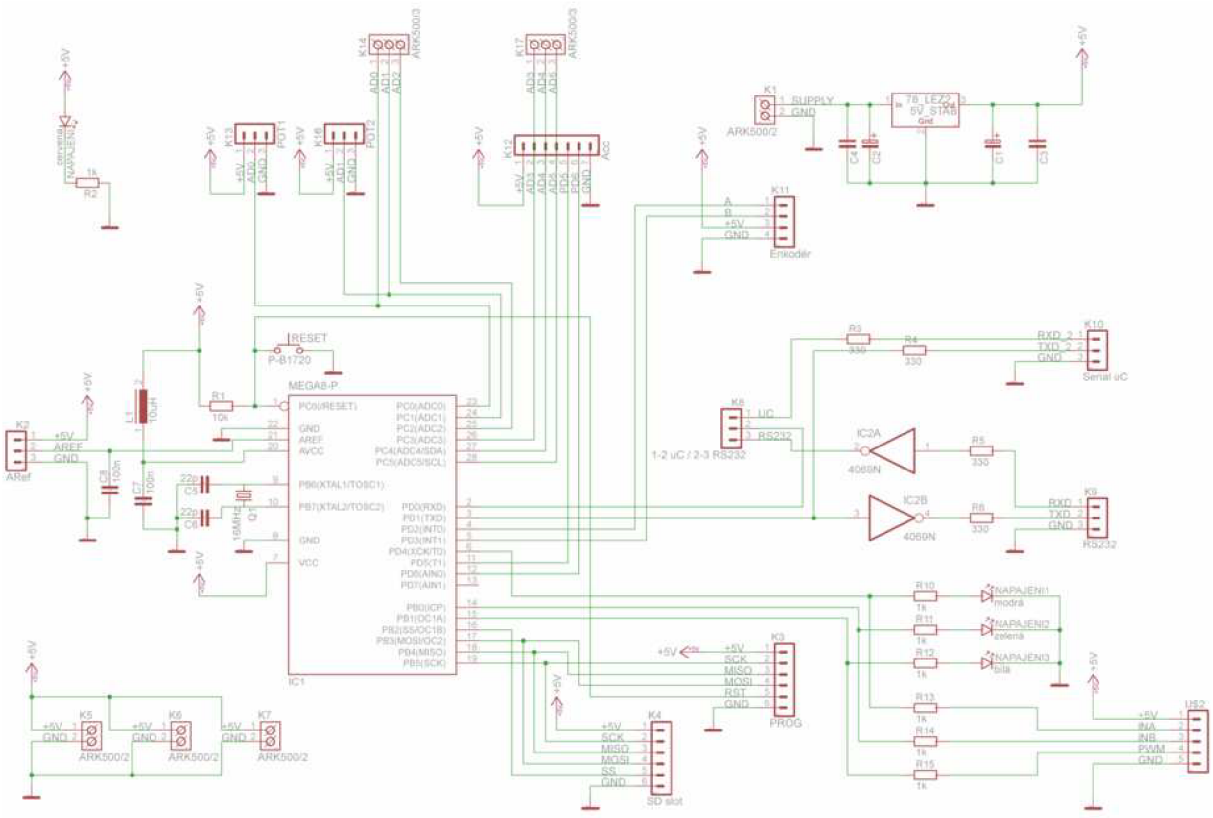
IV MFU168 jednotky

IV.1 Měřicí a řídicí jednotka

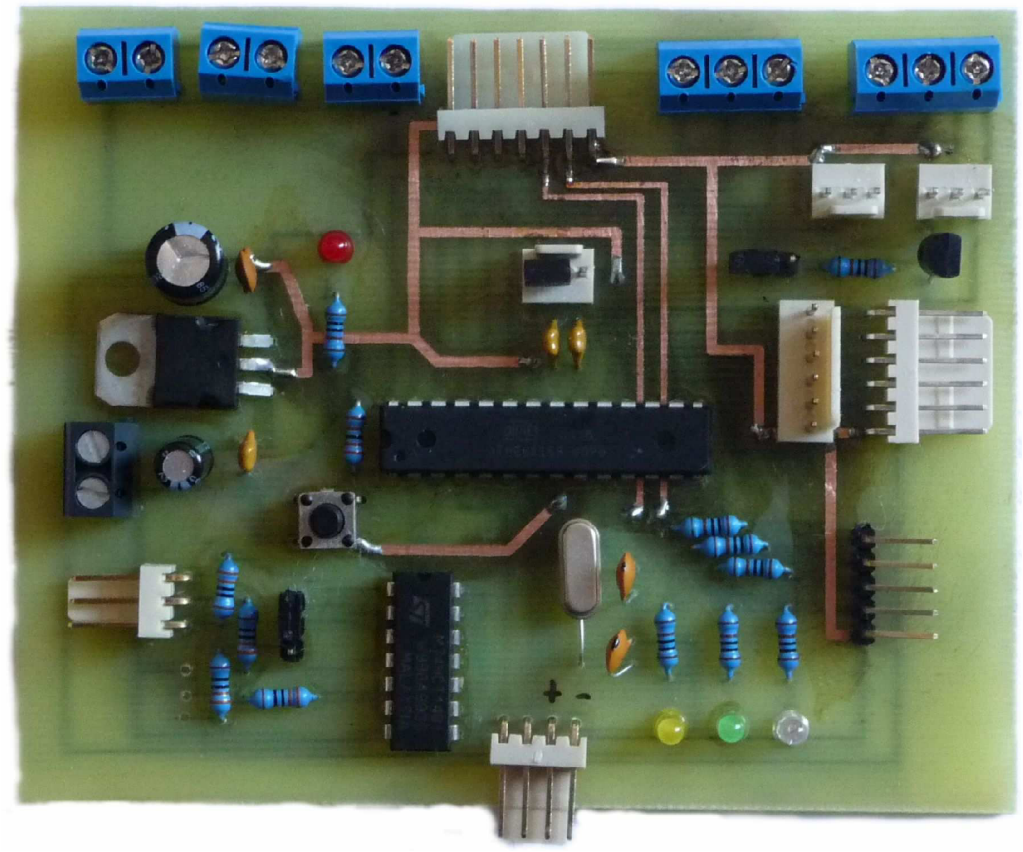


Obr. 47 Horní strana DPS Měřicí a řídicí jednotky

Cílem této práce bylo vytvořit hardware, který bude splňovat všechny požadavky zadání a bude se řídit pokyny vedoucího BP. Na Obr. 47 je horní strana DPS, pro rychlejší výrobu a tvorbu popisků bylo označení vstupů/výstupů *Měřicí a řídicí jednotky* integrováno do horní předlohy plošného spoje. Tato jednotka obsahuje všechny nutné hardwarové části a vstupy/výstupy potřebné k testování všech aplikací zmíněných v předchozích kapitolách této práce. Schéma zapojení *Měřicí a řídicí jednotky* je na Obr. 48. Její fyzické provedení je na Obr. 49, jedná se o první verzi, která ještě neobsahuje popisky na horní straně DPS. Funkčnost je bezvadná, úspěšně byly otestovány všechny úlohy zmíněné v této práci.



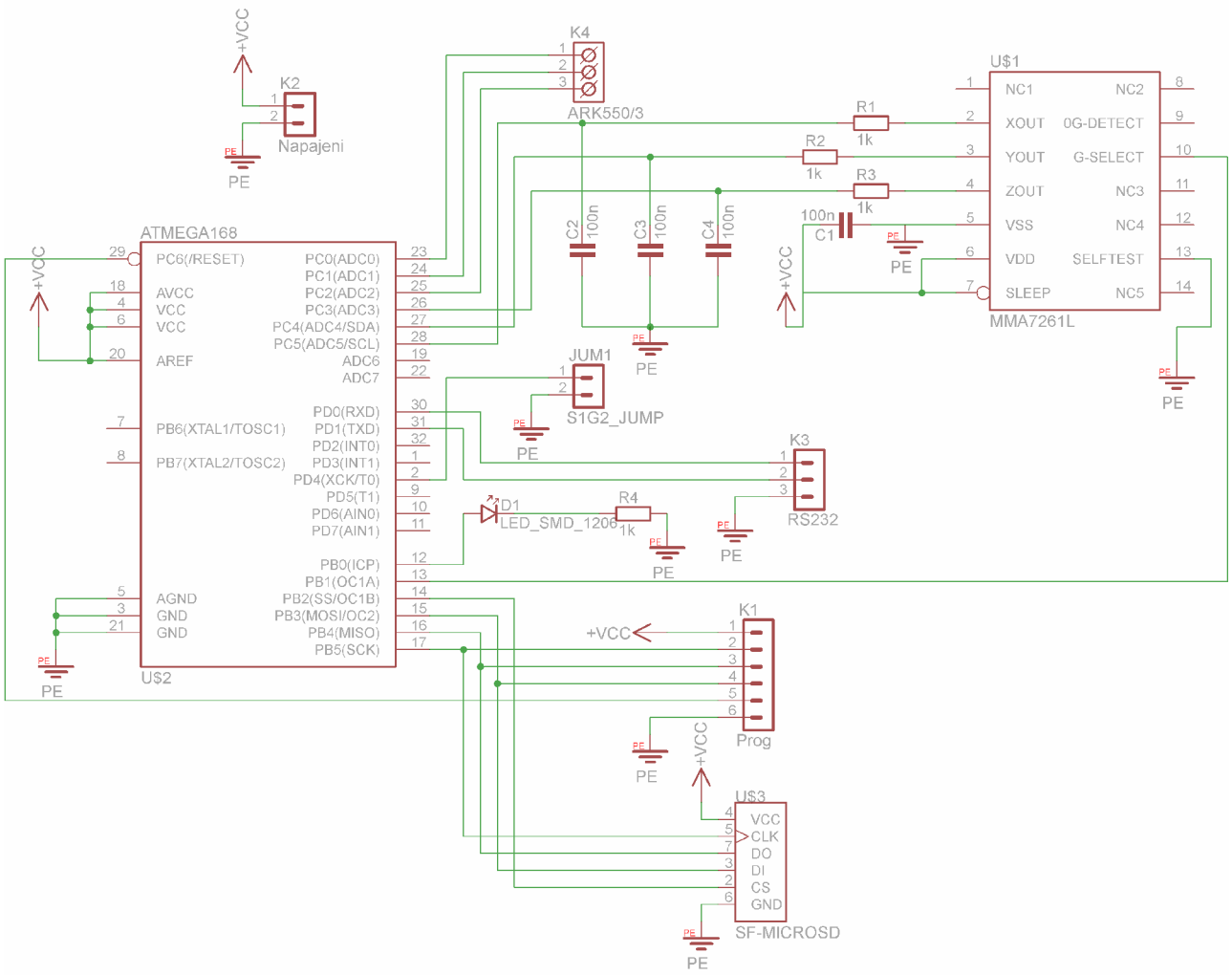
Obr. 48 Schéma zapojení Měřicí a řídicí jednotky



Obr. 49 Hardware měřicí a řídicí jednotky

IV.2 Miniaturní měřicí jednotka

Speciální verzí MFU168 je miniaturní provedení DPS, které obsahuje integrovaný akcelerometr a slot na MicroSD kartu. Hardware byl navržen s ohledem na požadavky vedoucího BP. Schéma zapojení je na Obr. 50.



Obr. 50 Schéma zapojení Miniaturní měřicí jednotky

V Ovládací funkce MFU168

V.1 Seznam příkazů v MFU168

Tab. 7: Příkazy v MFU168

Příkaz	Typ parametru	Krátká nápověda
help	-	Vypise seznam ovladacich prikazu.
ping	char	ping text; Odesle zpet <text>.
svit	-	Prepina svit LED.
pwm	char	Nastaveni vystupu PWM. [char]
der	single	Nastaveni derivacni slozky regulatoru. [single]
w	single	Nastaveni zadane hodnoty. [single]
itg	single	Nastaveni integracni slozky regulatoru. [single]
pVI	single	Koeficient omezeni integratoru, 1=max vykon. [single]
prp	single	Nastaveni proporcialniho zesileni regulatoru. [sinlge]
stop	-	Zastavi cinnost.
vykon	char	Omezeni vykonu: min0-max255 [char]
PWM_InA_B	-	Zapina vystup pro H-mustek.
vzorkovani	char	Vzorkovaci perioda, 1=1ms, 0-65535. [char]
povolit	-	Povoleni cinnosti.(1)Mereni (2)Filtr (3)Regulace (4)Odesilani [char]
zakazat	-	Zakaz cinnosti.(1)Mereni (2)Filtr (3)Regulace (4)Odesilani [char]
helpm	-	Vypise seznam promenych.
helpf	-	Vypise seznam filtru.
scope	char	scope promena; Vrati hodnotu promenne.
odesilatmsk	char	odesilatmsk 3,5,1; Nastaveni promenych k odesilani.
filtrmsk	char	filtrovatmsk 3,5,1; Nastaveni filtru.
meritAD0	-	Zapina AD prevodnik na pinu AD0.
meritAD01	-	Zapina AD prevodnik na pinu AD0 a AD1.
meritAD345	-	Zapina AD prevodnik na pinu AD3,AD4 a AD5.
rpokus	-	PID regulator s vylepsenimi.
pOdes	char	Faktor zpomaleni odesilani hodnot proti <vzorkovani>. [char]
pReg	char	Faktor zpomaleni regulace proti <vzorkovani>. [char]
reset	-	Vyresetovani MU.
gp	char	Obdelnikovy signal, nastaveni periody nasobkem <vzorkovani>. [char]
gh	single	Obdelnikovy signal, horni mez. [single]
gd	single	Obdelnikovy signal, dolni mez. [single]
K1	single	Nastaveni 1.koeficientu Kalmanova filtru. [single]
K2	single	Nastaveni 2.koeficientu Kalmanova filtru. [single]
enk	-	Zapnuti snimani signalu z enkoderu.
enk2	-	Zapnuti snimani signalu z enkoderu(pull-up).
zapisSD	-	Zapnuti odesilani promenych do SDkarty.
logSD	-	Logovani nastavovacich prikazu do SDkarty.
readSD	-	Precte blok pameti z SDkarty.
zapnicteniSD	-	Synchronizace nastaveni MU pro cteni z SDkarty.
cistSD	-	Cteni namerenych dat z SDkarty.

help
vypíše seznam všech podporovaných příkazů i s krátkou nápovědou

vzorkovani
jednotka pracuje v nekonečné smyčce s frekvencí 1000Hz (s periodou 1ms). Tímto příkazem lze nastavit vzorkovací frekvenci, např. při zadání příkazu „vzorkovani 500;“ jednotka vykonává příkazy každých 500ms (0,5s).

povolit
v nekonečné smyčce jsou spouštěny jednotlivé moduly (měření,regulace,odesílání,...). Ty lze samostatně zapínat/vypínat. Příkazem „povolit;“ se zapnou všechny moduly v nekonečné smyčce a budou se provádět. Lze zapnout pouze jeden z modulů, jako parametr příkazu se zadá index modulu, např. příkazem „povolit 3;“ se zapne třetí modul.

zakazat
v nekonečné smyčce jsou spouštěny jednotlivé moduly (měření,regulace,odesílání,...). Ty lze samostatně zapínat/vypínat. Příkazem „zakazat;“ se vypnou všechny moduly v nekonečné smyčce a nebudou se provádět. Lze vypnout pouze jeden z modulů, jako parametr příkazu se zadá index modulu, např. příkazem „zakazat 2;“ se vypne druhý modul.

helpm
vypíše seznam všech proměnných, které lze z jednotky číst

scope
příkaz pro jednorázové zjištění hodnoty proměnné v jednotce. Příkazem „helpm;“ lze zjistit názvy všech proměnných, které můžeme číst. Pro zjištění jejich obsahu stačí zadat příkaz scope s názvem proměnné jako parametr, např „scope itg;“ odešle hodnotu proměnné itg.

meritAD0
zapne v jednotce analogově-digitální převodník, čte hodnotu z pinu ADC0 a ukládá do proměnné AD0. Toto měření se provádí v nekonečné smyčce v bloku měření.

V.2 Popis filtrů v MFU168

Tab. 8: Filtry v MFU168

Filtr	Krátká nápověda
fAD0y	Zkopiruje hodnotu z <AD0> do <y>.
gpuls	Generuje obdelnikovy signal do <gv>.
fgvw	Zkopiruje hodnotu z <gv> do <w>.
fAD1w	Zkopiruje hodnotu z <AD1> do <w>.
k	Kalmanova filtrace <y>.
zpomal	Pauza 13ms.
fenky	Zkopiruje hodnotu z <enk> do <y>.
fderenky	Zkopiruje hodnotu derivace z <enk> do <y>.

V.3 Popis proměnných v MFU168

Tab. 9: Proměnné v MFU168

Proměnná	Typ	Počet bytů	Krátká nápověda
toc	uint8	1	Stopky. [uint8]
pwm	int16	2	Vystupni hodnota regulace. [int16]
der	single	4	Derivacni slozka regulatoru. [single]
itg	single	4	Integracni slozka regulatoru. [single]
prp	single	4	Proporcionalni slozka regulatoru. [single]
kodeslani	uint8	26	Seznam indexu odesilanych promenych. [uint8]
AD0	uint16	2	Analogovy vstup 0. [uint16]
AD1	uint16	2	Analogovy vstup 1. [uint16]
AD3	uint16	2	Analogovy vstup 3. [uint16]
AD4	uint16	2	Analogovy vstup 4. [uint16]
AD5	uint16	2	Analogovy vstup 5. [uint16]
y	uint8	4	Vstupni hodnota do regulatoru. [single]
w	single	4	Zadana hodnota. [single]
e	single	4	Regulacni odchylka. [single]
v	single	4	Rychlost-zmena vstupni hodnoty regulatoru. [single]
dt	uint8	4	Perioda regulatoru. [single]
K1	single	4	Zesileni Kalman. filtru 1 [single]
K2	single	4	Zesileni Kalman. filtru 2 [single]
VI	single	4	Hodnota naintegrovane slozky. [single]
maxVI	single	4	Max naintegrovane slozky. [single]
maxvykon	uint8	1	Maximalni vykon. [uint8]
akce	single	4	Akcni velicina. [single]
cpu	uint16	2	Vytizenost MFU168,255=100% [uint16]
kfiltrovani	uint8	9	Indexy spustenych filtru. [uint8]
enk	int16	4	Vystup z enkoderu. [int16]

VI Proměnné v MFU168

VI.1 Přidání nové proměnné do MFU168

V souboru *init.h* jsou uloženy globální proměnné MFU168. Firmware MFU168 je psán v jazyce C, je využívána knihovna avr-libc uzpůsobená možnostem mikročipů AVR. V manuálu k WINAVR můžeme nalézt typy proměnných, které lze nativně definovat. Jako příklad můžeme ukázat, jakým způsobem lze do MFU168 přidat proměnnou pro nastavení proporcionálního zesílení PID regulátoru. Neboť proporcionální zesílení může nabývat reálných hodnot, je vhodné použít proměnnou typu double (číslo s plovoucí desetinnou čárkou). Název proměnné zvolíme např. *Vprp*. V souboru *init.h* tedy přidáme definici nové proměnné *Vprp*:

```
double Vprp=0;
```

Soubor *init.h* je vložen do programu MFU168 jako první, naše vytvořená proměnná bude tedy viditelná v celém programu. Jazyk C dovoluje vytvořit proměnou i dále v kódu programu, ale potom není zajištěno, že při úpravách jiných souborů programu bude tato proměnná viditelná, protože se může stát, že námi upravovaný kód je vložen dříve, než definujeme proměnnou. Nemůžeme tedy o ní vědět a náš pokus skončí chybovou hláškou.

VI.2 Nastavení proměnné z PC

Pro nastavení proměnné *Vprp* z PC je nutné definovat další parametry a vytvořit v MFU168 funkci, která umožní příjem hodnoty z PC a její uložení do proměnné. Nastavení proměnné z PC probíhá formou textové zprávy. Pro rychlejší vytváření zprávy bylo vytvořeno makro s názvem `VYTVOR_ZPRAVU`. Toto makro přijímá tři parametry:

`VYTVOR_ZPRAVU("název",příkaz,"nápověda")`
“název“ - textový řetězec, kterým z PC zadáváme, jakou proměnnou chceme změnit
příkaz – název funkce v MFU168, která přečte hodnotu proměnné a uloží ji
“nápověda“ – je zobrazena v PC pro lepší orientaci uživatele

Definujeme tedy novou zprávu, následující kód je zapsán do souboru *prikazy.h*

```
VYTVOR_ZPRAVU("prp",prp,
"Nastaveni proporcialniho zesileni regulatoru. [single]");
```

Jako druhý parametr jsme zvolili název funkce *prp*, která se vykoná po doručení textové zprávy s názvem „`prp`“ (první parametr). Není však zatím nikde definovaná. Je vhodné uspořádat všechny funkce podobného charakteru do stejného souboru, kterým je zde *prikazy.c*. Pro definování funkce je nutné použít další makro s názvem `PRIKAZ`, které přijímá jako parametr název funkce – *prp*.

```
PRIKAZ(prp){
     RAWPREVOD(Vprp);
     return 1;}
```

Proměnná *prp* je typu double, pro její přečtení a uložení bylo použito další makro s názvem `RAWPREVOD`, které přijímá jako parametr název proměnné typu double, zde *Vprp*. Toto makro se postará o správné interpretování doručených dat a jejich uložení do proměnné *Vprp*.

Posledním krokem je zaregistrování příkazu v seznamu zpráv makrem `Z`. Seznam zpráv se nachází v souboru *prikazy.h*.

```
ZPRAVA povel[] PROGMEM={
     Z(chyba),
     Z(help),
...
     Z(prp),
...
     Z(PcistSD),
};
```

Po zaregistrování lze příkaz zavolat z PC.