

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

BAKALÁŘSKÁ PRÁCE

2014

Josef Navrátil

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

**Laboratorní přípravy pro výuku Virtuální
instrumentace**

**Laboratory Demonstration Devices for Virtual
Instrumentation Education**

2014

Josef Navrátil

Zadání bakalářské práce

Student: **Josef Navrátil**

Studijní program: B2649 Elektrotechnika

Studijní obor: 2601R004 Měřicí a řídicí technika

Téma: **Laboratorní přípravy pro výuku Virtuální instrumentace
Laboratory Demonstration Devices for Virtual Instrumentation
Education**

Zásady pro vypracování:

Náplní práce je návrh a konstrukce přípravků se senzory neelektrických veličin a modulem Arduino pro účely výuky automatizace měření a testování. Přípravky budou pro interakci s počítačem využívat rozhraní multifunkční karty NI PCI 6221 a sériového či USB portu. Součástí práce je tvorba vzorových příkladů v LabVIEW využívajících přípravky.

Body zadání:

1. Seznámení se s problematikou Arduino.
2. Návrh a výběr vhodných senzorů.
3. Návrh a realizace HW.
4. Návrh a realizace SW pro Arduino a SW příkladů v LabVIEW.
5. Zhodnocení dosažených výsledků práce.

Seznam doporučené odborné literatury:

- [1] MARGOLIS, Michael. *Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects*. 2nd Edition. Sebastopol, Ca: O'Reilly Media, 2012. 724 s. ISBN 0-596-80247-1.
- [2] MONK, Simon. *30 Arduino projects for the evil genius*. New York: McGraw-Hill, c2010, xiii, 191 p. ISBN 00-717-4133-X.
- [3] BITTER, Rick, Taqi MOHIUDDIN and Matt NAWROCKI. *LabView advanced programming techniques*. 2nd ed. Boca Raton: CRC Press, c2007, 499 s. ISBN 08-493-3325-3.

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

Vedoucí bakalářské práce: **doc. Ing. Petr Bilík, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení Studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Datum: 7.5.2014

Podpis: J. Novák

Abstrakt

Cílem práce bylo navrhnout a zkonstruovat sadu přípravků pro výuku virtuální instrumentace. Jedná se o přípravek s mikrokontrolérem a LCD displejem využívající platformu Arduino, jehož účelem je simulovat testované zařízení. Dále přípravek se senzory jako je například digitální teploměr, ultrazvukový dálkoměr nebo hodiny reálného času komunikující po sběrnici I2C. Třetí přípravek je osazen krokovým motorem, modelářským servomotorem a senzory pro zavedení zpětné vazby. Ke všem prvkům na přípravcích vznikly rovněž vzorové aplikace v LabVIEW, které demonstrují možnosti použití ve výuce. Program pro kontrolér vznikl ve vývojovém prostředí pro Arduino.

Klíčová slova

Virtuální instrumentace, LabVIEW, Arduino, mikrokontrolér, senzory neelektrických veličin, krokový motor, servomotor, I2C sběrnice

Abstract

The target of this work was to design and build a set of laboratory devices for education of virtual instrumentation. First is device with microcontroller and LCD display based on Arduino platform, which serves as the device under test. The second laboratory device contains several types of sensor such as digital thermometer, ultrasonic rangefinder or real time clock. The clock module uses I2C bus for communication. The third device is equipped with stepper motor and servomotor including sensors for feedback. For all components on laboratory devices were made example applications in LabVIEW to illustrate how to use them in education. Program for microcontroller was written in development environment for Arduino platform.

Key Words

Virtual instrumentation, LabVIEW, Arduino, microcontroller, sensors, stepper motor, servomotor, I2C bus

Obsah

| | | |
|-------|--|----|
| 1 | Úvod..... | 1 |
| 2 | Arduino..... | 2 |
| 2.1 | Charakteristika platformy..... | 2 |
| 2.2 | Hardware..... | 2 |
| 2.2.1 | Arduino UNO R3..... | 2 |
| 2.2.2 | Možnost návrhu vlastního hardwaru..... | 3 |
| 2.2.3 | Sanguino..... | 4 |
| 2.3 | Software..... | 4 |
| 2.4 | Programovací jazyk..... | 5 |
| 3 | Virtuální instrumentace..... | 6 |
| 4 | LabVIEW..... | 7 |
| 4.1 | Základní informace..... | 7 |
| 4.2 | Programování..... | 7 |
| 4.3 | Propojení LabVIEW a Arduino..... | 9 |
| 4.3.1 | Rozhraní..... | 9 |
| 4.3.2 | Toolkit pro Arduino..... | 9 |
| 5 | Přípravek s motory..... | 11 |
| 5.1 | Deska plošných spojů..... | 11 |
| 5.2 | Krokový motor s inkrementálním snímačem..... | 12 |
| 5.2.1 | Krokový motor..... | 12 |
| 5.2.2 | Inkrementální snímač otáček S3806..... | 13 |
| 5.2.3 | Budič krokového motoru..... | 13 |
| 5.2.4 | Připojení a použití..... | 14 |
| 5.2.5 | Vzorové VI..... | 15 |
| 5.3 | Servomotor a akcelerometr..... | 15 |
| 5.3.1 | Servomotor Vigor VS-2..... | 16 |
| 5.3.2 | Akcelerometr..... | 16 |
| 5.3.3 | Připojení a použití..... | 16 |
| 5.3.4 | Vzorové VI..... | 17 |
| 6 | Přípravek s mikrokontrolérem..... | 18 |
| 6.1 | Účel a základní informace..... | 18 |
| 6.2 | Popis jednotlivých prvků..... | 19 |

| | | |
|-------|---------------------------------------|----|
| 6.3 | Programování | 22 |
| 6.4 | Použití..... | 22 |
| 6.5 | Laboratorní úlohy | 22 |
| 6.5.1 | Obecné informace..... | 22 |
| 6.5.2 | Generátor digitálních signálů | 23 |
| 6.5.3 | Napětím řízený oscilátor..... | 24 |
| 6.5.4 | Spínač s hysterezí | 25 |
| 6.5.5 | Číslicový oscilátor..... | 26 |
| 6.5.6 | Číslicový oscilátor USB | 28 |
| 6.5.7 | I2C slave..... | 30 |
| 6.5.8 | Digitální teploměr..... | 31 |
| 6.5.9 | IMU | 32 |
| 7 | Přípravek se senzory..... | 33 |
| 7.1 | Účel a základní informace | 33 |
| 7.2 | LED, tlačítka, potenciometr | 33 |
| 7.3 | Ultrazvukový dálkoměr SRF05 | 34 |
| 7.4 | Infračervený dálkoměr | 35 |
| 7.5 | Mechanický inkrementální snímač..... | 36 |
| 7.6 | Analogový teploměr TMP36GT9Z | 36 |
| 7.7 | Společné demonstrační VI..... | 36 |
| 7.8 | Digitální teploměr DS18B20 | 37 |
| 7.9 | Hodiny reálného času (RTC)..... | 38 |
| 7.10 | Inertial Measurement Unit (IMU) | 40 |
| 8 | I2C pro LabVIEW | 42 |
| 8.1 | Základní informace..... | 42 |
| 8.2 | Nastavení..... | 43 |
| 8.3 | Zápis dat | 43 |
| 8.4 | Čtení dat | 43 |
| 8.5 | Příklad použití | 44 |
| 9 | Závěr..... | 45 |
| 10 | Citovaná literatura | 46 |
| 11 | Seznam příloh..... | 49 |
| 11.1 | Tištěné přílohy..... | 49 |

| | | |
|------|--------------------------------------|----|
| 11.2 | Elektronické přílohy (obsah CD)..... | 49 |
|------|--------------------------------------|----|

1 Úvod

Cílem této práce je navrhnout a následně vytvořit laboratorní přípravky pro výuku virtuální instrumentace. Aktuálně vybavení učebny sestává z výpočetní techniky s měřicími kartami NI PCI 6221 a systému NI ELVIS - NI Educational Laboratory Virtual Instrumentation Suite. Laboratorní modul obsahuje kromě velkého nepájivého kontaktního pole integrovanou multifunkční převodníkovou kartu připojitelnou k PC přes USB. Tento víceúčelový hardware lze spolu s dodávaným softwarem použít jako celou řadu virtuálních přístrojů – digitální multimetr, signální generátor, osciloskop a další. Primárně slouží NI ELVIS ve výuce pro analýzu a testování elektronických obvodů sestavených na kontaktním poli s uplatněním principů virtuální instrumentace.

Nevýhodou základního provedení NI ELVIS je absence jakýchkoliv prvků pro fyzickou interakci s okolím, tedy senzorů neelektrických veličin a akčních členů, například modelu elektrického pohonu. Nedostatek lze pochopitelně odstranit nákupem rozšiřujících modulů, které se instalují místo kontaktního pole. National Instruments nabízí například modul se stejnosměrným motorem, inverzním kyvadlem nebo senzory pro studium biomedicínského inženýrství. [1] Vybavení učebny těmito přípravky v dostatečném počtu ale brání velmi vysoká cena v řádu desítek tisíc korun za kus.

Výsledkem bakalářské práce je sada laboratorních přípravků skládající se ze tří základních modulů. Prvním modulem je programovatelná laboratorní úloha s mikrokontrolérem založená na vývojové platformě Arduino. V paměti mikrokontroléru je uloženo několik programů - samostatných úloh, které student vybere a spustí dle pokynů vyučujícího. Spuštěný program zajišťuje obsluhu periférií mikrokontroléru tak, aby fungoval jako testované zařízení, například napěťově řízený oscilátor. Úkolem studenta bude zařízení otestovat s využitím převodníkové karty a softwaru LabVIEW.

Druhý navržený modul obsahuje základní elektronické prvky (LED, potenciometr, spínač...) a především senzory pro měření neelektrických veličin, které bude možné připojit k převodníkové kartě. Díky takovému přípravku si student osvojí základní principy virtuální instrumentace během měření například ultrazvukovým čidlem vzdálenosti.

Třetí modul je tvořen akčními členy doplněnými o zpětnou vazbu. Přínos úlohy pro studenta by měl spočívat především v možnosti prakticky si vyzkoušet řízení modelového elektrického pohonu a následně kontrolu jeho chodu pomocí příslušného senzoru. Konkrétně byl pro tento účel zvolen krokový motor mechanicky spojený s inkrementálním snímačem otáček. Součástí bakalářské práce je rovněž vytvoření vzorových příkladů v LabVIEW, které budou demonstrovat možnosti využití přípravků během výuky.

2 Arduino

2.1 Charakteristika platformy

Arduino je open-source vývojová platforma zaměřená na práci s mikrokontroléry Atmel. Projekt Arduino si klade za cíl zjednodušit vývoj vlastního hardwaru řízeného mikrokontrolérem tak, aby byla tato problematika přístupná daleko širší skupině osob, než dříve. Uplatnění najde především mezi amatérskými nadšenci, kterým nabízí možnost tvorby vlastních programovatelných zařízení velmi snadno i v domácích podmínkách bez speciálního vybavení. Ze stejného důvodu lze Arduino považovat i za dobrou pomůcku pro výuku technických oborů se zaměřením na měření, řízení, programování a elektroniku. Studentům umožní rychleji a snáze proniknout do problematiky mikrokontrolérů, přičemž i náklady na pořízení hardwaru jsou poměrně nízké.

Pojem Arduino pochopitelně zahrnuje nejen samotný hardware, ale především vlastní programovací jazyk a vývojové prostředí, které je zdarma a lze ho stáhnout z webových stránek projektu. [2] K tomu je třeba přičíst volnou dostupnost veškeré dokumentace. K Arduino navíc patří velká komunita uživatelů a fanoušků po celém světě, kteří publikují své vlastní projekty, vytvářejí knihovny, návody, doplňkový hardware pro rozšíření a diskutují na internetových fórech. Díky tomu může začínající i pokročilý uživatel Arduina počítat s tím, že na případný problém kromě něj narazil nejspíš i někdo jiný, nebo že k použitému čidlu nebo jinému zařízení najde příklad s kódem nebo knihovnu.

Mezi naprosto klíčové vlastnosti Arduina patří, že pro nahrání vlastního programu vytvořeného na PC do mikrokontroléru není potřeba programátor ani žádných složitých nastavení. Arduino hardware totiž ve většině případů obsahuje standardní USB rozhraní, které funguje jako emulace sériové linky. Lze ho tedy připojit k notebooku nebo PC a bez jakéhokoliv dalšího vybavení začít programovat.

2.2 Hardware

V rámci projektu Arduino vznikla celá řada verzí a provedení hardwaru, nazývaného *boards* (desky), navíc jsou neustále vyvíjeny nové. Nicméně vždy existuje základní, nejpoužívanější deska vhodná pro začátek. V době vzniku bakalářské práce je aktuální verze toho základního hardwaru Arduino UNO R3. Na trhu je dostupná rovněž větší a výkonnější deska Arduino Mega 2560, dále také desky se speciálními vlastnostmi - například integrovaný Bluetooth modul, Ethernet nebo WiFi. V případě požadavků na co nejmenší rozměry lze zvolit výrobek Arduino Nano nebo Mini. Funkčnost Arduina můžeme dále rozšiřovat pomocí doplňkových desek, takzvaných *shields* (štíty). Kromě originálních shieldů – například Ethernet Shield, Motor Shield, GSM Shield [3], existuje také celá řada neoficiálních, ale kompatibilních výrobků. Je to dáno jednak volně dostupnou dokumentací (open-source) a rovněž popularitou Arduina.

2.2.1 Arduino UNO R3

Rozměry desky provedené v charakteristické modré barvě jsou přibližně 55 x 75 mm. Základ tvoří mikrokontrolér Atmel Atmega328. Deska nabízí celkem 14 digitálních pinů, které lze použít jako vstupy i výstupy, na 6 z nich lze generovat signál s PWM. Dále je zde k dispozici 6 analogových vstupů, keramický rezonátor taktující procesor v kontroléru na 16 MHz, USB konektor pro připojení k PC nebo notebooku a napájecí konektor. Vzhledem k tomu, že Atmega328 nemá integrovanou podporu USB rozhraní, je mezi ním a USB konektorem zařazen mikrokontrolér Atmega16U2, který plní funkci převodníku. Na desce nalezneme kromě několika podpůrných součástek, jako jsou

kondenzátory nebo stabilizátor napětí také ICSP (In Circuit Serial Programming) konektor, resetovací tlačítko a čtyři barevné LED. Konkrétně to jsou indikátory zapnutého napájení (ON), komunikace (RX, TX) a poslední LED lze ovládat softwarově, navíc po startu indikuje funkci bootloderu.



Obr. 1 Arduino UNO

Bootloderem se rozumí malý program uložený v paměti mikrokontroléru již výrobcem vývojové desky, který umožňuje programování přes sériovou linku. Díky tomu, jak už bylo dříve zmíněno, odpadá použití speciálního programátoru.

Tab. 1 Shrnutí parametrů Arduino UNO R3 [4]

| | |
|---|--|
| Mikrokontrolér: | ATmega328 |
| Pracovní napětí: | 5V |
| Vstupní napětí (doporučené): | 7-12V |
| Vstupní napětí (limity): | 6-20V |
| Digitální I/O (vstupně-výstupní) piny: | 14 (6 x PWM) |
| Analogové vstupy: | 6 |
| Maximální proud I/O pinu: | 40 mA |
| Maximální proud 3.3 V pinu: | 50 mA |
| Flash paměť: | 32 kB (ATmega328), přibližně 0.5 kB zabírá bootloder |
| SRAM: | 2 kB (ATmega328) |
| EEPROM: | 1 kB (ATmega328) |
| Takt procesoru: | 16 MHz |

2.2.2 Možnost návrhu vlastního hardwaru

Vzhledem k tomu, že Arduino je open-source projekt, nic nebrání návrhu vlastního hardwaru – desky plošných spojů s mikrokontrolérem, který by byl s touto platformou kompatibilní. Pro zkušenější uživatele je to ideální možnost, jak vytvořit hardware, který by co nejlépe odpovídal požadavkům a zamýšlené aplikaci. Pro kompatibilitu s platformou Arduino je třeba dodržet několik pravidel. Jednak musí být zvolen podporovaný mikrokontrolér, například tedy Atmega328 z Arduino UNO. Dále se

musí shodovat způsob komunikace s počítačem, což zajišťuje USB převodník. Těch pochopitelně existuje více variant. A v neposlední řadě je nutné dodržet hodnotu taktovací frekvence.

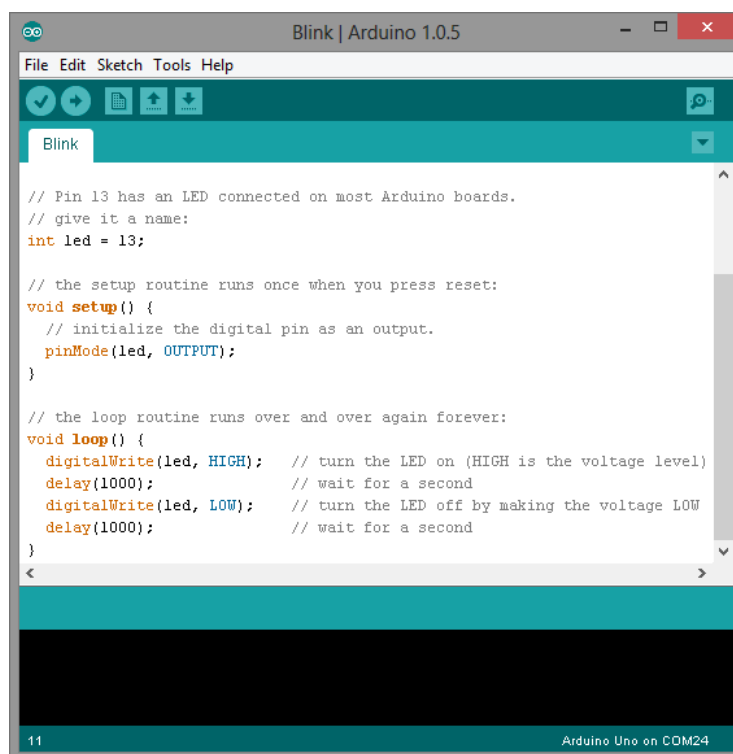
Hardware navržený podle těchto zásad není problém pro začátek sestavit i na nepájivém kontaktním poli. Po připojení k počítači s ním dále lze nakládat jako s Arduinem, využívat vývojové prostředí, programovací jazyk s knihovny atd. Výše uvedené poznatky jsou naprosto klíčové pro další části bakalářské práce, protože využití originální Arduino desky (například UNO R3) by pro tvorbu laboratorního přípravku nebylo příliš výhodné. Nelze totiž zapomínat na hlavní účel Arduino desek a tím je rychlá tvorba prototypu zařízení. Doplnkové komponenty jako třeba displej nebo tlačítka se připojují pomocí vodičů, případně kontaktního pole. Takto sestavený přípravek by ale pro laboratoř s velkým průchodem studentů nebyl vhodný. Pro trvalé nasazení zařízení založeného na platformě Arduino je daleko výhodnější navrhnout vlastní desku plošných spojů se všemi potřebnými komponenty.

2.2.3 Sanguino

Sanguino je projekt příbuzný Arduinu a sdílí stejný programovací jazyk i vývojové prostředí. Pouze rozšiřuje kompatibilitu na kontrolér Atmega644P. Sanguino by se tedy dalo nazvat jednou z mnoha neoficiálních variant Arduina. Výhodou kontroléru Atmega644P je oproti verzi Atmega328 větší počet vstupně výstupních pinů, větší paměť a také obsahuje rovnou dvě rozhraní pro sériovou komunikaci (UART). Hardwarové parametry ovšem nelze srovnávat s vyšší řadou Arduina – verzemi Mega, takže se Sanguino hodí nejlépe pro středně složité aplikace, na které nedostačuje Arduino UNO a naopak Arduino Mega by zůstalo z velké části nevyužité. Jako hlavní nevýhodu Sanguina lze uvést, že neobsahuje USB port, ale je nutné použít externí převodník. V případě návrhu vlastní desky plošných spojů kompatibilní s platformou Arduino ale tato nevýhoda ztrácí význam. Právě mikrokontrolér Atmega644P bude využit při návrhu laboratorního přípravku.

2.3 Software

Vývojové prostředí pro Arduino je k dispozici ke stažení z webových stránek ve verzích pro Windows, Linux a Mac OS X. Na první pohled vypadá skutečně minimalisticky, obsahuje pouze editační okno programu, dole lištu pro zobrazení oznámení a chyb, nahoře pak několik málo tlačítek (uložení programu, nahrání do Arduina atd.) a menu. Ke správnému fungování musí uživatel nastavit jen dvě zásadní věci – typ připojeného Arduina a správný port. Prostředí obsahuje navíc velmi užitečný nástroj, kterým je Serial Monitor – jinak řečeno terminál, kterým lze přijímat a odesílat data do Arduina pomocí virtuální sériové linky. To je v podstatě všechno potřebné pro práci s vývojovou platformou Arduino. Pro úplnost je vhodné uvést, že program pro Arduino se v angličtině běžně nazývá *sketch*.



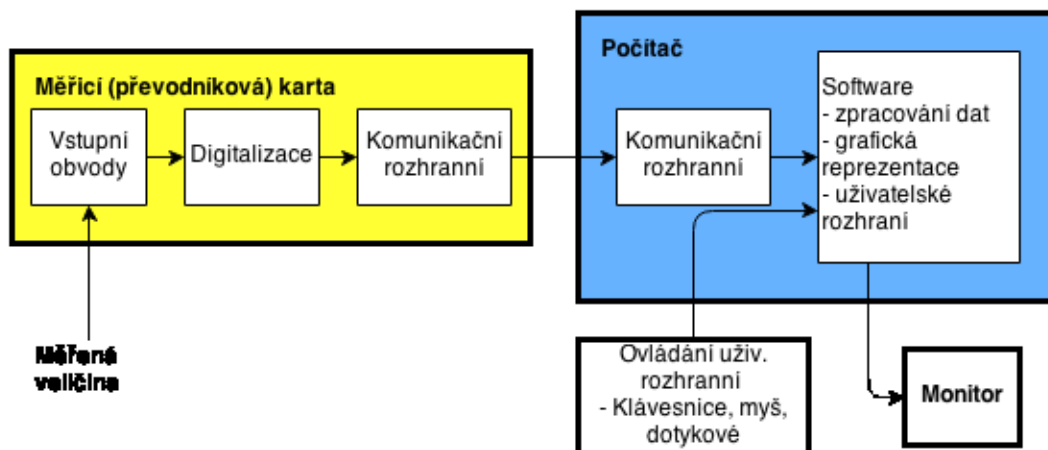
Obr. 2 Vývojové prostředí pro Arduino s otevřeným vzorovým programem

2.4 Programovací jazyk

Arduino používá vlastní programovací jazyk, který ovšem syntaxí a vlastnostmi vychází z rozšířeného jazyka C/C++. Obsluha hardwaru mikrokontroléru je založena na AVR Libc, tedy standardní knihovně pro Atmel AVR 8-bitové RISC mikrokontroléry. [5] Jazyk pro Arduino pak tvoří určitou nástavbu nad C/C++ a AVR Libc, která se skládá z celé řady maker a knihoven podstatně usnadňujících práci s periferiemi kontroléru. Uživatel Arduina je tedy ušetřen programování na nižší úrovni blíže k hardwaru, například konfigurace registrů pro čtení analogových vstupů, nastavení časovačů ke generování PWM a podobně. Obsluha nejběžnějších zařízení, jako jsou například LCD displeje, modelářské servomotory a podobně již kompletně řeší oficiální knihovny. Stejně tak komunikaci pomocí standardních sběrnic – SPI, I2C. Kromě oficiálních knihoven od autorů projektu je na internetu dostupná celá řada těch neoficiálních od uživatelů.

3 Virtuální instrumentace

Virtuální instrumentace je disciplína, která se zabývá využitím počítače jako náhrady klasického měřicího přístroje. Z hlediska požadavků na měření totiž počítač, ať už ve stolní nebo přenosné variantě, obsahuje zobrazovací zařízení pro prezentaci změřených hodnot a rovněž dostatečný výpočetní výkon pro případné zpracování, další analýzu a podobně. Počítači chybí pouze nejnižší fyzická vrstva, hardware, který by zvládl například měřit napětí. Toto lze vyřešit připojením měřicí (převodníkové) karty.



Obr. 3 Ideové schéma virtuálního měřicího přístroje

K hlavním přednostem takového řešení patří především daleko větší flexibilita. Běžný měřicí přístroj, například multimetr, nabízí konečný počet funkcí a nelze jej dále rozvíjet nebo měnit. Obsahuje totiž jednoduše integrované obvody, případně mikrokontrolér s obvykle neměnným firmwaru. Počítač vybavený kartou schopnou měřit napětí můžeme využít rovněž místo multimetru, nicméně změnou spuštěného softwaru se tento měřicí systém promění v osciloskop. Samozřejmě za předpokladu, že je karta dostatečně rychlá. Spuštěním dalšího softwaru je možné například provést rychlou Fourierovu transformaci a podobně. Software se tedy stává naprosto klíčovou součástí a promění obyčejný počítač ve virtuálního měřicího přístroj. Vzhledem k tomu, že software lze snadno měnit a aktualizovat, může být k dispozici vždy nejnovější verze přizpůsobená požadavkům uživatele. V případě, že tento měřicí systém už nesplňuje nové hardwarové požadavky – výkon, paměť, počet vstupů, stačí obvykle vyměnit pouze jednu z komponent počítače nebo měřicí kartu. Klasický měřicí přístroj je zpravidla třeba vyměnit kompletně za novější nebo vybavenější model.

Pomocí měřicího systému s počítačem je také daleko snadnější proces měření automatizovat, zejména pokud se jedná o měření a následné zpracování velkého množství veličin – například monitorování výrobní linky a podobně.

4 LabVIEW

4.1 Základní informace

Pod pojmem LabVIEW se rozumí programovací prostředí vyvinuté společností National Instruments, které využívá ke tvorbě aplikací grafický jazyk. Zkráceně také G-jazyk nebo v angličtině Graphical Language. [6] Na rozdíl od klasických textově orientovaných programovacích jazyků, jako je například C#, neprobíhá tvorba programu psaním kódu, ale sestavováním diagramu z grafických bloků plnících určitou funkci. Tento princip zpřístupňuje programování i skupině uživatelů, která nemá zkušenosti s běžnými jazyky odvozenými od jazyka C nebo Basicu. Pokud technik dokáže navrhnout funkčnost požadovaného softwaru například ve formě vývojového diagramu, musí jej následně programátor převést do příslušného programovacího jazyka. Nicméně když máme k dispozici jazyk, který se svou strukturou velmi blíží vývojovému diagramu s návrhem softwaru, podstatně se zkrátí a zjednoduší celý proces vývoje. Vytvořit vlastní software tak může i člověk bez znalosti jiného textově orientovaného jazyka, pouze na základě toho, že ví, jak by výsledný software měl pracovat a je schopen svou představu sestavit ve formě diagramu.

Na základě předchozích faktů by se mohlo zdát, že LabVIEW patří především do výuky nebo snad jen začínajícím programátorům. To ovšem není pravda, jedná se rovnocennou alternativou ke klasickému textovému programování. Naopak díky výrazné orientaci technickým směrem nabízí LabVIEW celou řadu funkcí, které u jiných jazyků nenajdeme.

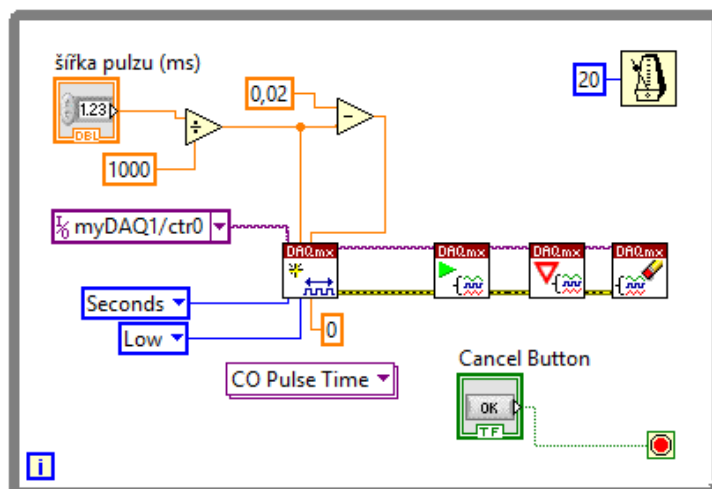
4.2 Programování

Výsledkem programování v LabVIEW je takzvané VI, zkráceně Virtual Instrument neboli virtuální přístroj. Toto názvosloví vychází z faktu, že program na obrazovce zastává stejnou funkci jako klasický měřicí přístroj. Každé VI obsahuje dvě základní a neoddělitelné části – čelní panel a diagram. Čelní panel obsahuje ovládací prvky – tlačítka, přepínače, otočné knoflíky a samozřejmě také indikátory – LED, ručkové ukazatele, grafy. Tento čelní panel slouží stejnému účelu jako u fyzického přístroje, tedy zajišťuje interakci s uživatelem. Z pohledu programování se jedná o GUI (Graphical User Interface – grafické uživatelské rozhraní). Jak bude celé VI fungovat, definuje programátor v blokovém diagramu, zde se vytváří pomocí grafického jazyka vlastní program. Pomocí ikon jsou v blokovém diagramu reprezentovány prvky čelního panelu. Lze z nich vyčítat, případně do nich ukládat hodnoty podle toho, zda se jedná o ovládací prvek nebo indikátor. Jednotlivé bloky diagramu se spojují pomocí vodičů (wires). [6]

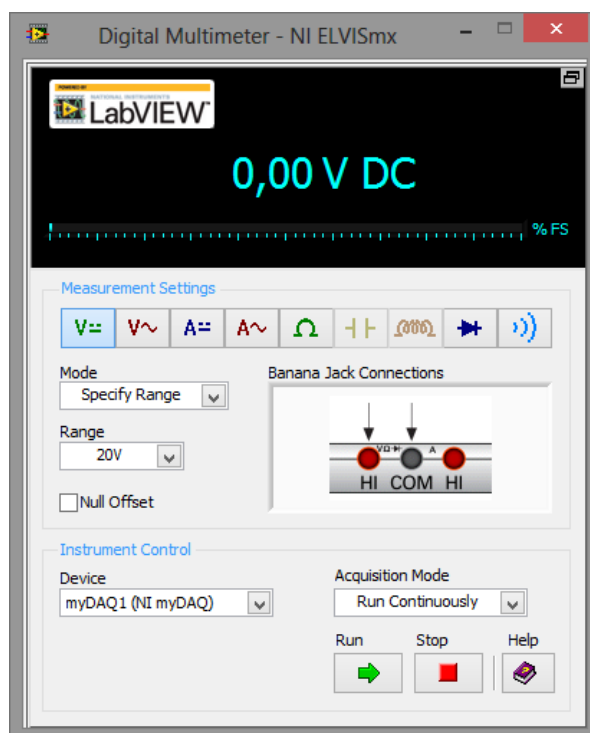
Programátor má k dispozici všechny myslitelné funkce známe z jiných jazyků, například knihovnu matematických operací, bloky pro řízení toku programu – smyčky, strukturu case a podobně. LabVIEW rovněž nabízí obvyklé datové typy, jako jsou boolean, integer, floating-point nebo string. Včetně možnosti vytvářet pole těchto datových typů. Průběh programu je řízen tokem dat, nikoliv striktně posloupností příkazů jako například u textového jazyka C. Jednotlivé funkce se vykonávají až tehdy, jsou-li známé jejich vstupní hodnoty, které přivádíme pomocí vodičů do konektoru. Každé VI respektive blok diagramu, které očekává nějaké vstupní hodnoty nebo chce předat výsledek svého výpočtu dále, má tedy svůj konektor pro připojení k jiným VI.

Společnost National Instruments se kromě programovacího prostředí LabVIEW zabývá také vývojem hardwaru pro měření požadovaných veličin a případně i následné ovlivnění monitorovaného procesu

pomocí výstupů. Tento hardware se obvykle nazývá převodníkovou kartou a dohromady s LabVIEW tvoří velmi silný nástroj pro vývoj měřicích a řídicích systémů.



Obr. 4 Příklad blokového diagramu v LabVIEW



Obr. 5 Čelní panel virtuálního měřicího přístroje

4.3 Propojení LabVIEW a Arduino

4.3.1 Rozhraní

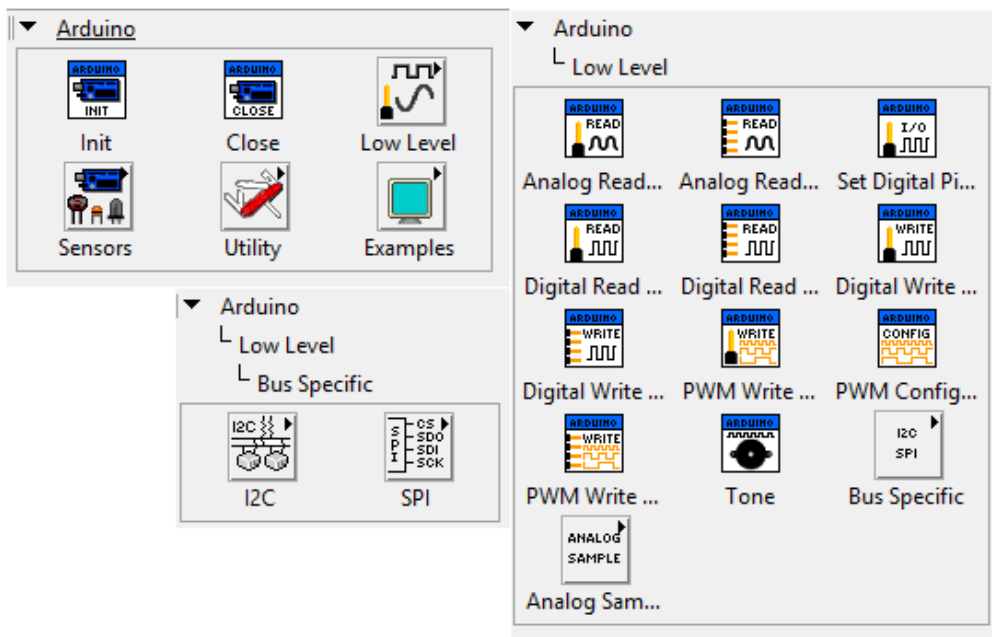
Mikrokontroléry používané v platformě Arduino jsou vybaveny rozhraním UART, což je zkratka anglických slov Universal Asynchronous Receiver/Transmitter, univerzální asynchronní přijímač/vysílač. Jedná se tedy o zařízení pro sériový přenos dat. [7] V případě kontroléru probíhá sériová komunikace v napěťových úrovních TTL. Pro připojení k počítači je nutné tyto napěťové úrovně přizpůsobit tak, aby byly kompatibilní se standardem RS232. Například pomocí integrovaného obvodu MAX232. Toto rozhraní však v současné době ustupuje a zejména přenosné počítače už jím zpravidla nedisponují. Je tedy výhodnější použít některý z převodníků UART-USB, které po připojení k počítači a korektní instalaci ovladačů vytvoří virtuální sériovou linku, respektive port označovaný jako COM. Starší verze hardwaru Arduino měly ještě převodníky na RS232, dnes už obsahují právě konvertory na USB.

National Instruments poskytuje pro LabVIEW širokou škálu ovladačů (NI-VISA) a to včetně podpory sériové komunikace. [8] Z hlediska uživatele a tvorby programu v LabVIEW není rozdíl mezi tím, zda se jedná o fyzickou sériovou linku standardu RS232 a nebo virtuální variantu realizovanou skrze USB. Pomocí VISA tedy můžeme realizovat oboustrannou sériovou komunikaci mezi Arduinem a LabVIEW.

4.3.2 Toolkit pro Arduino

Jednou z velmi zajímavých možností, jak využít Arduino spolu s LabVIEW nabízí toolkit, který umožní přistupovat k hardwarovým prostředkům mikrokontroléru přímo z blokového diagramu VI. Lze tak jednak získávat data – číst analogové i digitální vstupy, ale také generovat signály, například s PWM a komunikovat po sběrnicích SPI a I2C. Všechny tyto funkce se ovládají graficky pomocí blokových diagramů. Aby Arduino s LabVIEW toolkitem spolupracovalo, musí v něm být nahrán speciální program označený jako LIFA base (LabVIEW Interface for Arduino).[9] Pochopitelně musí být po celou dobu připojeno k počítači, pomocí USB, sériové linky nebo Bluetooth. Princip funkce je takový, že VI spuštěné v počítači komunikuje s Arduinem a odesílá mu instrukce na základě bloků v diagramu. Program LIFA base spuštěný v Arduinu pak dle přijatých instrukcí manipuluje s hardwarovými prostředky (digitální a analogové vstupy/výstupy). Nastavuje požadované logické úrovně, generuje PWM nebo odesílá zpět naměřené hodnoty. V případě přerušení komunikace – ať už úmyslného nebo nechtěného odpojení od počítače se jeví Arduino jako nefunkční, protože program čeká na instrukce z LabVIEW. Toolkit tedy není možné využít tak, že by se pomocí něj graficky vytvořil program pro Arduino, který by se následně uložil do jeho paměti a běžel nezávisle na PC. Arduino se pouze stane periferií, kterou lze z LabVIEW běžícího na PC ovládat

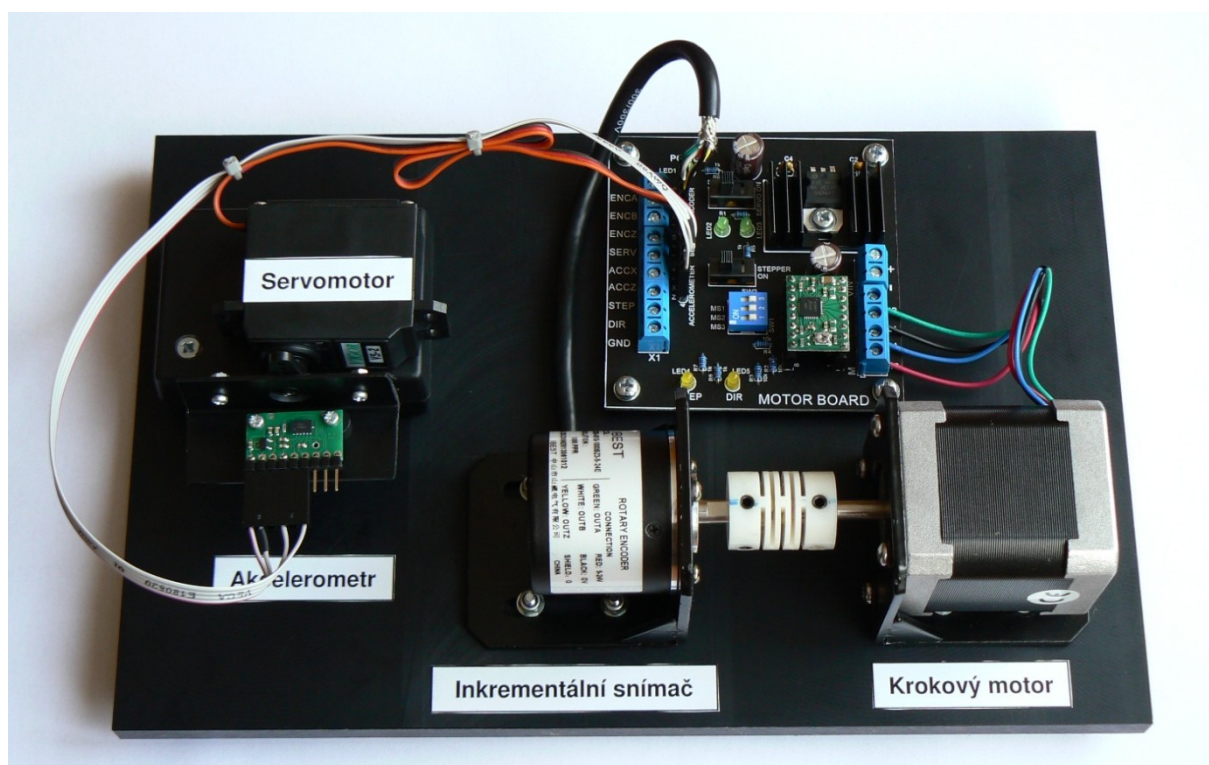
Arduino v kombinaci s toolkitem pro LabVIEW může posloužit především jako velmi levná a jednoduchá alternativa k měřicím kartám pro účely výuky. Některé základní úlohy lze totiž pomocí Arduina realizovat za cenu v řádu několik set korun, kdežto měřicí karta NI-myDAQ určená studentům stojí okolo 5 000 Kč.



Obr. 6 Příklad dostupných funkcí v Arduino Toolkitu

5 Přípravek s motory

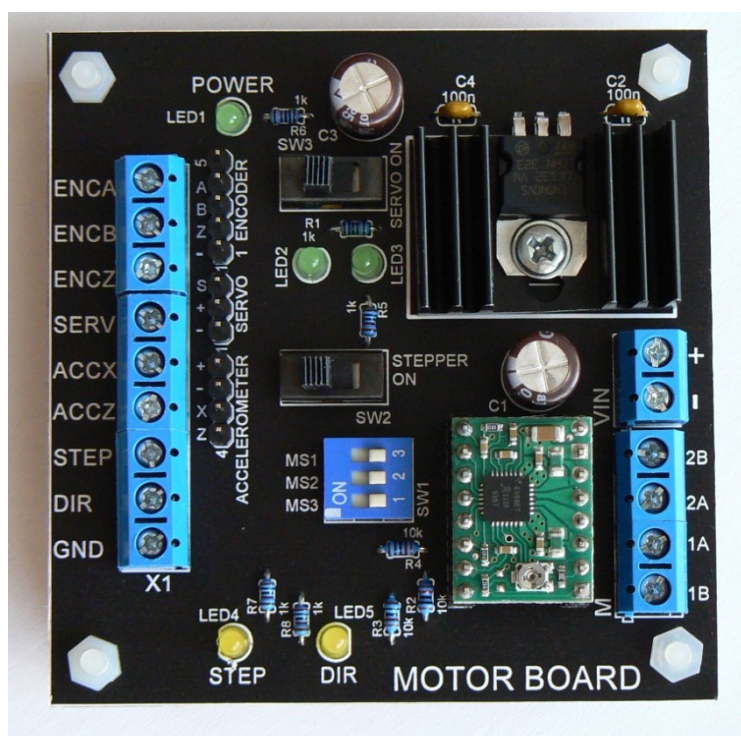
Účelem laboratorního přípravku s motory je nabídnout studentům možnost vyzkoušet si prakticky řízení modelového elektrického pohonu ze softwaru v počítači. Přípravek je osazen krokovým motorem, inkrementálním snímačem, modelářským servomotorem a akcelerometrem. Data ze senzorů lze přes převodníkovou kartu přivést zpět do počítače, zobrazit a případně dále analyzovat. Koncepce přípravku odráží nejen často používaný princip – vybavení elektrického pohonu senzorem pro zavedení zpětné vazby, ale také skutečnost, že i procedury prováděné během měření a testování je třeba automatizovat. A právě při automatizaci měření a testování vzniká potřeba mít možnost nejen měřit požadované veličiny, ale také s testovaným objektem manipulovat.



Obr. 7 Přípravek s motory

5.1 Deska plošných spojů

Deska plošných spojů obsahuje elektronické prvky nutné pro funkci motorů a snímačů, slouží rovněž pro připojení přípravku k měřicí kartě pomocí svorkovnice X1. Pro napájení je nutné zvolit zdroj s hodnotou napětí 9 V až 12 V stejnosměrných (svorkovnice označená VIN). Přítomnost napájecího napětí indikuje LED1 (opatřená popisem power). Pro logické obvody a servomotor se napájecí napětí stabilizuje na hodnotu 5 V. K připojení fází krokového motoru slouží svorkovnice M s kontakty 1A, 1B, 2A, 2B. Akcelerometr, servomotor a inkrementální snímač se připojují pomocí konektorů se standardní roztečí 2,54 mm. Detailnější informace ke všem indikačním LED a spínačům jsou uvedeny v následujících kapitolách s ohledem na to, zda funkčně náleží krokovému motoru nebo servomotoru.



Obr. 8 Deska plošných spojů na přípravku s motory

5.2 Krokový motor s inkrementálním snímačem

Hřídel krokového motoru a inkrementálního snímače jsou mechanicky spojeny pomocí pružné plastové spojky. Jedna otáčka motoru tedy přesně odpovídá jedné otáčce snímače. K upevnění motoru i snímače slouží stejný kovový úhelník (určený pro motory s přírubou NEMA17). Před montáží snímače bylo tedy nutné na úhelníku převrtat montážní otvory. Buzení krokového motoru zajišťuje modul s integrovaným obvodem A4988.

5.2.1 Krokový motor

Jedná se o standardní krokový motor s přírubou NEMA17 a 200 kroky na otáčku. Lze ho budit jak unipolárně – připojení šesti vodičů, tak bipolárně – čtyřmi vodiči. [10] Na přípravku je použito bipolární buzení. Motor nabízí firma Pololu jako položku #1200.

Tab. 2 Parametry krokového motoru [10]

| | |
|--|-----------------------------------|
| Rozměry | 42,3 x 42,3 x 48 mm (bez hřídele) |
| Hmotnost | 350 g |
| Průměr hřídele | 5 mm (tvar písmene “D”) |
| Kroky na otáčku | 200 |
| Jmenovité napětí | 4 V |
| Jmenovitý proud | 1,2 A na fázi |
| Odpor | 3,3 Ω na fázi |
| Krouticí moment při zablokování | 0,31 Nm |



Obr. 9 Inkrementální snímač (vlevo) a krokový motor (vpravo)

5.2.2 Inkrementální snímač otáček S3806

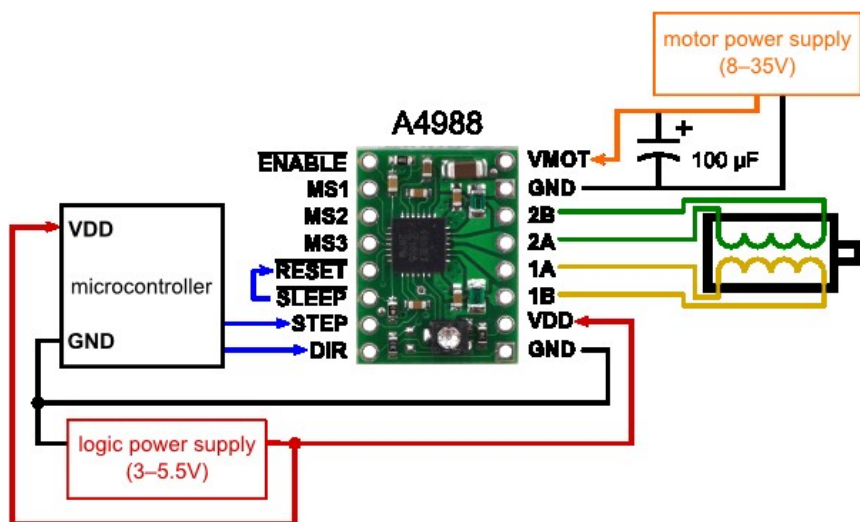
Poskytuje informaci o otáčení hřídele krokového motoru s rozlišením 1000 impulzů na otáčku. V přípravku je použit snímač kvadraturní, s dvěma výstupy (A, B) posunutými o 90° elektrických, což slouží k rozlišení směru otáčení. Navíc třetí výstup (Z) poskytuje vždy jeden impulz za celou otáčku.[11] Měřicí karta National Instruments zvládne zpracovat signál ze všech tří výstupů.

Tab. 3 Parametry inkrementálního snímače [11]

| | |
|-------------------------------------|-----------|
| Průměr pouzdra | 38 mm |
| Průměr hřídele | 6 mm |
| Napájecí napětí | 5-24 V ss |
| Maximální výstupní frekvence | 300 kHz |
| Typ výstupu | napěťový |
| Výstupy | A, B, Z |

5.2.3 Budič krokového motoru

Deska plošných spojů je navržena pro umístění driveru s integrovaným obvodem A4988. Modul prodáváný firmou Pololu usnadňuje použití integrovaného obvodu, jehož miniaturní pouzdro není příliš vhodné pro ruční osazování. Navíc obsahuje další součástky nutné pro správnou funkci obvodu. Problematika buzení krokového motoru je driverem s A4988 již kompletně vyřešena. Obsahuje vstupy STEP a DIR pro řídicí signály definující směr otáčení a počet kroků, výstup pro čtyřvodičové připojení motoru (bipolární buzení) a navíc také vstupy pro výběr režimu mikrokrokování.[12] Použití modulu dostatečně objasňuje schéma z webové dokumentace na Obr. 10.



Obr. 10 Použití driveru krokového motoru s IO A4988[12]

Volba mikrokrokování se provádí pomocí pole mikropínačů SW1, přepínačem SW2 lze budič uzamknout respektive povolit (vstup ENABLE). Povolený budič indikuje zelená LED2. Žluté LED4 a LED5 svítí v případě přivedení logické jedničky na vstupy STEP a DIR. Driver není v DPS pevně zapájen, ale pouze zasunut v dutinkových lištách, aby byla možná rychlá výměna. Shrnutí funkce jednotlivých prvků se nachází v tabulce. V režimu zamknutí driveru je motor odpojen od napájení, lze s ním volně rukou otáčet a nebude tedy reagovat na žádný signál na vstupech STEP a DIR.

Tab. 4 Přehled funkcí jednotlivých prvků DPS

| Prvek | Funkce |
|------------------------|-------------------------------|
| Pole 3 mikropínačů SW1 | Volba mikrokrokování |
| Přepínač SW2 | Povolení/zamknutí budiče |
| Zelená LED2 | Indikace povoleného budiče |
| Žlutá LED4 | Logická úroveň na vstupu STEP |
| Žlutá LED5 | Logická úroveň na vstupu DIR |

Tab. 5 Konfigurace mikrokrokování pomocí mikropínačů

| S1 | S2 | S3 | Režim mikrokrokování |
|-----|-----|-----|----------------------|
| OFF | OFF | OFF | Plný krok |
| OFF | OFF | ON | 1/2 kroku |
| OFF | ON | OFF | 1/4 kroku |
| OFF | ON | ON | 1/8 kroku |
| ON | ON | ON | 1/16 kroku |

5.2.4 Připojení a použití

Vstupní signály úlohy jsou dva STEP a DIR, které slouží k řízení krokového motoru. ENKA a ENKB slouží jako výstupy z inkrementálního snímače, ENKZ je volitelný. Přípravku je potřeba dodat silové napájení o hodnotě 9 V až 12 V a propojit jeho zem se zemí měřicí karty. Přepínač SW2 musí být v pravé poloze (označené STEPPER ON).

Tab. 6 Připojení přípravku

| Svorkovnice | Funkce | Komentář |
|-------------|----------------------|---------------------------------|
| ENKA | Kanál A ink. snímače | Obdélníkový signál 0 – 5 V |
| ENKB | Kanál B ink. snímače | Obdélníkový signál 0 – 5 V |
| ENKZ | Kanál Z ink. snímače | Obdélníkový signál 0 – 5 V |
| STEP | Motor - krok | Vstup TTL |
| DIR | Motor - směr | Vstup TTL |
| GND | Zem | Společná pro signály i napájení |

5.2.5 Vzorové VI

Příklad v LabVIEW vytvořený k této úloze umožňuje vygenerovat zadaný počet impulzů pro driver krokového motoru a rovněž řídit směr otáčení. Časování impulzů je řízeno pouze softwarově, takže skutečná rychlost otáčení motoru bude záviset na vytížení procesoru. Vzhledem k principu úlohy to ale není na závadu. Pomocí čítače měřicí karty se vyhodnocují signály z inkrementálního snímače a výsledkem je údaj o natočení ve stupních. Jako signály STEP a DIR lze použít kterékoli digitální výstupy karty, signály z inkrementálního snímače musí být vždy přivedeny na terminály čítače v souladu s technickou dokumentací karty NI PCI-6221.



Obr. 11 Demonstrační VI ke krokovému motoru

5.3 Servomotor a akcelerometr

Na unášeci modelářského servomotoru Vigor VS-2 je připevněn kříž s plastovým úhelníkem, čímž vzniká malá plošina s možností rotace (náklonu) v rozsahu přibližně 180°. Respektive se může vůči vodorovné poloze natočit až o -90° nebo +90°. Modul Pololu1251 s tříosým akcelerometrem MMA7361L je uchycen na plošině tak, aby kladný směr osy Z směřoval vzhůru. Přípravek slouží k měření závislosti výstupu akcelerometru na orientaci vůči tíhovému poli země a díky použití servomotoru může student tento úkol automatizovat. Následně se nabízí možnost použít akcelerometr jako zpětnou vazbu – úhel náklonu vypočítaný z dat akcelerometru musí alespoň přibližně odpovídat zadané hodnotě natočení servomotoru. Vzhledem k velmi jednoduché technologii jsou výsledky měření pouze demonstrační, neočekává se vysoká přesnost. Nicméně lze tak ověřit nejzákladnější princip pro zjištění náklonu (obecně orientace v prostoru) [13] používaný například v mobilních telefonech nebo bezpilotních letounech.

5.3.1 Servomotor Vigor VS-2

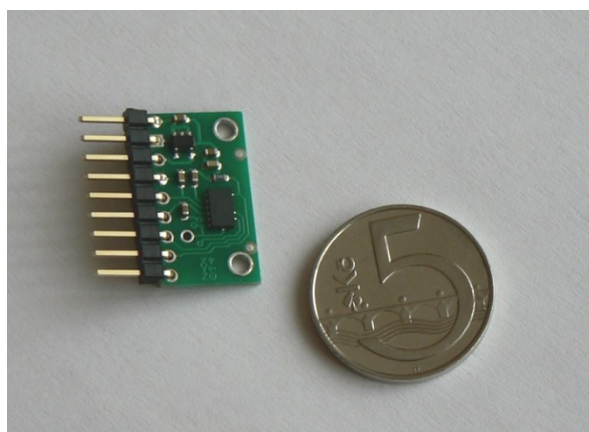
Servomotor nevyžaduje žádnou další elektroniku pro buzení, stačí připojit napájení v rozsahu 4,8 V až 6 V. Jako řídicí signál (žlutý vodič) se používá kladný puls šířky 1 ms až 2 ms opakovaný každých 20 ms, přičemž 1,5 ms odpovídá neutrální poloze. Nicméně neexistuje žádný standard definující přesný vztah mezi šířkou pulzu a polohou osy servomotoru. Obvykle se uvádí, že rozsah 1 až 2 ms by měl odpovídat pohybu přibližně o 90°. [14] Rovněž není stanovena minimální a maximální šířka pulzu. Použité servo Vigor by dle katalogového listu mělo reagovat na pulz v rozsahu 800 až 2200 μ s. [15] Ověření mechanických krajních poloh a požadovaných výchylek se musí provést experimentálně pro každý typ modelářského servomotoru.

Tab. 7 Parametry servomotoru Vigor VS-2 [15]

| | | |
|----------------------------------|-----------------------|------------|
| Pracovní napětí | 4,8 Važ 6 V | |
| Rozměry | 40,6 x 20,0 x 38,9 mm | |
| Hmotnost | 37 g | |
| Řídicí pulz | 800 až 2200 μ s | |
| | 4,8 V | 6 V |
| Maximální kroučící moment | 0,31 Nm | 0,34 Nm |
| Odběr proudu (pohyb) | 0,2 A | 0,25 A |
| Proud při zablokování | 0,8 A | 1,0 A |

5.3.2 Akcelerometr

Základem modulu je tříosý akcelerometr Freescale MMA7361L s analogovým výstupem a nastavitelnou citlivostí $\pm 1,5$ g nebo ± 6 g. Modul již obsahuje 3,3 V regulátor napětí, takže napájení lze zvolit z rozsahu 2,2 až 16 V. [16] Na přípravku je použito stejné napájecí napětí ze stabilizátoru jako pro servomotor. Výstupem jsou pájecí body označené X, Y, Z, na nich naměříme napětí odpovídající zrychlení v příslušné ose. Pokud kontakt GSEL zůstane nepřipojen, nastaví se základní citlivost $\pm 1,5$ g [16], která odpovídá záměru užití akcelerometru v úloze.



Obr. 12 Modul akcelerometru

5.3.3 Připojení a použití

Jako jediný řídicí signál slouží pulz pro servomotor, který je třeba generovat dle zásad uvedených v kapitole 5.3.1 a přivést na svorkovnici SERV. Hned vedle jsou k dispozici napětí z akcelerometru AKCX a AKCZ. Připojení osy Z je volitelné, pro základní funkci úlohy stačí pouze osa X. Přípravku

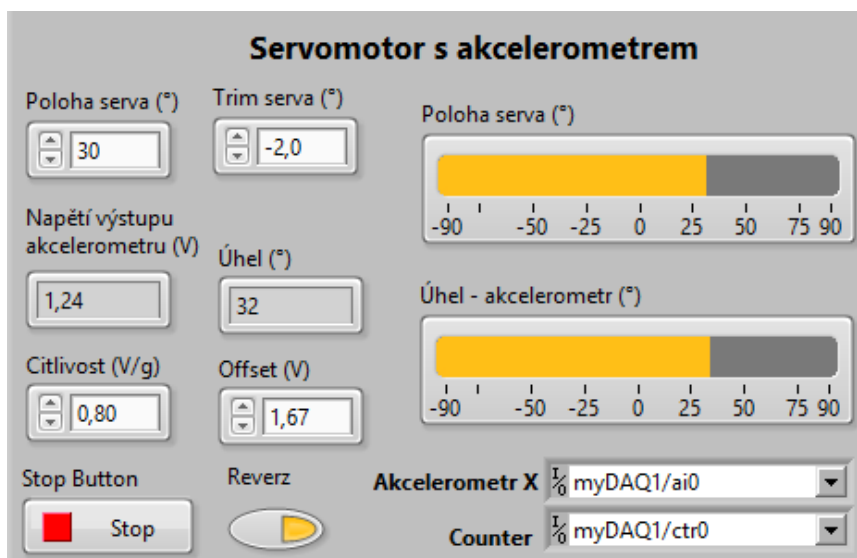
je potřeba dodat silové napájení o hodnotě 9 V až 12 V a propojit jeho zem se zemí měřicí karty. Shrnutí obsahuje Tab. 8. K zapnutí respektive vypnutí napájení servomotoru slouží přepínač SW3, přičemž aktuální stav indikuje LED3.

Tab. 8 Připojení přípravku

| Svorkovnice | Funkce | Komentář |
|-------------|---------------------------|---------------------------------|
| SERV | Řídicí signál servomotoru | Kladný pulz 800 až 2200 μ s |
| AKCX | Osa X akcelerometru | Analogový výstup, 0 -3,3 V |
| AKCZ | Osa Z akcelerometru | Analogový výstup, 0 -3,3 V |
| GND | Zem | Společná pro signály i napájení |

5.3.4 Vzorové VI

Umožňuje zadávat polohu servomotoru ve stupních, tento údaj převede na řídicí impulz a generuje pomocí časovače měřicí karty. Ke korekci nulové polohy servomotoru slouží ovládací prvek *Trim*. Poté měří napětí z akcelerometru analogovým vstupem a provede výpočet náklonu. Pro správné fungování musí být nastavena *citlivost* a *offset* dle katalogového listu akcelerometru MMA7361L. Výpočet náklonu i servopulzu zajišťují subVI, ve kterých je přímo v diagramu zdokumentován princip jejich funkce.

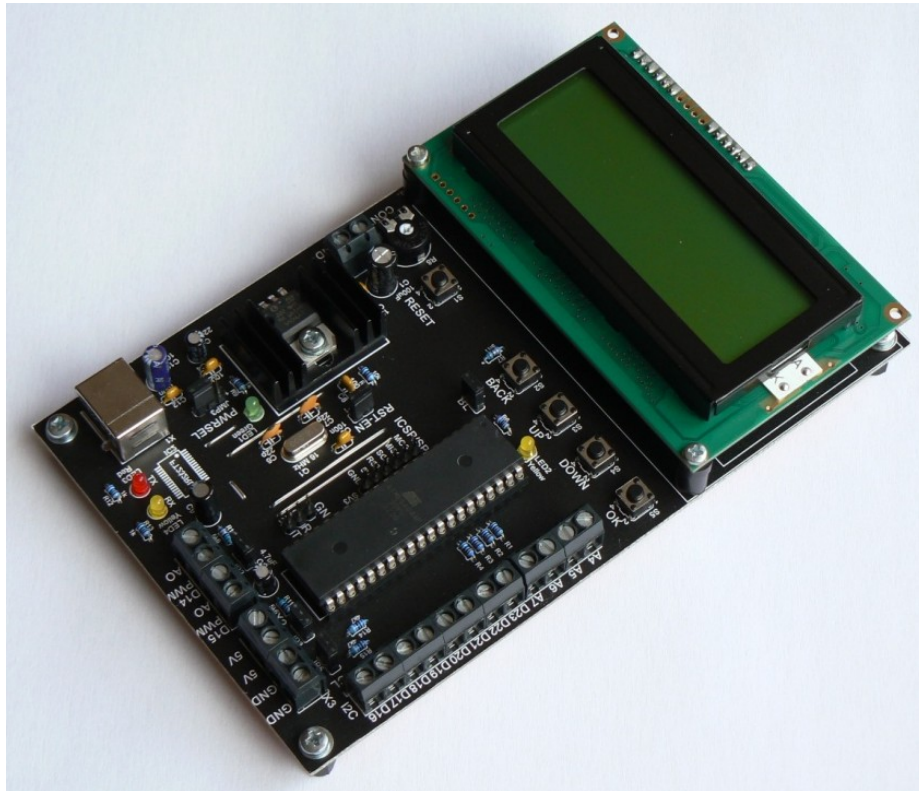


Obr. 13 Použití servomotoru a akcelerometru v LabVIEW

6 Přípravek s mikrokontrolérem

6.1 Účel a základní informace

Programovatelný přípravek bude ve výuce plnit roli testovaného zařízení, které bude mít student za úkol přezkoušet pomocí převodníkové karty a softwaru LabVIEW. K tomuto účelu vzniklo pro přípravek několik různých úloh, které je možné přepínat dle pokynů vyučujícího.



Obr. 14 Přípravek s mikrokontrolérem kompatibilní s platformou Arduino

Základem zařízení je 8 bitový mikrokontrolér Atmega644P, který kromě hlavního programu obsahuje rovněž bootloader zajišťující kompatibilitu s platformou Arduino. Jako hlavní komunikační rozhraní, které slouží i k programování, je použito USB a to prostřednictvím převodníku FT232RL. Pro snadnou obsluhu ze strany studenta nebo vyučujícího obsahuje přípravek čtyřřádkový alfanumerický LCD displej. Pomocí něj a čtyř ovládacích tlačítek se volí a spouští jednotlivé úlohy. Displej rovněž slouží k zobrazení potřebných informací o úlohách, přijatých dat a podobně. K měřicí kartě se přípravek připojuje pomocí svorkovnic. Napájení zajišťuje volitelně buď USB port nebo stabilizátor spolu s externím zdrojem napětí – například adaptérem do zásuvky 230 V.

| | |
|-------------------------------------|--|
| Mikrokontrolér: | Atmega644P |
| Takt procesoru: | 16 MHz |
| Hlavní komunikační rozhraní: | USB (převodník FT232RL) |
| Displej: | 4 x 20 znaků |
| Napájení: | Volitelně USB nebo přes stabilizátor (doporučeno max. 12 V DC) |
| Pracovní napětí: | 5 V |
| Počet digitálních I/O: | 10 (z toho dva s PWM) |
| Počet analogových vstupů: | 4 |
| Rozlišení AD převodníků: | 10 bitů |
| Ovládání: | 4 tlačítka + reset |
| Další možnosti komunikace: | SPI, I2C |

Tab. 9 Shrnutí základních parametrů přípravku

| | |
|---------------------------------------|-----------------------------|
| Architektura: | RISC, Atmel AVR 8 bit |
| Počet I/O pinů: | 32 (6 x PWM) |
| Počet analogových vstupů: | 8 |
| Rozlišení AD převodníků: | 10 bitů |
| Flash paměť: | 64 kB |
| RAM: | 4 kB |
| EEPROM: | 2 kB |
| Časovače: | 2 x 8 bitový, 1 x 16 bitový |
| Pracovní napětí: | 2,7 V až 5,5 V |
| Maximální taktovací frekvence: | 20 MHz |
| Komunikace: | 2 x UART, SPI, I2C |

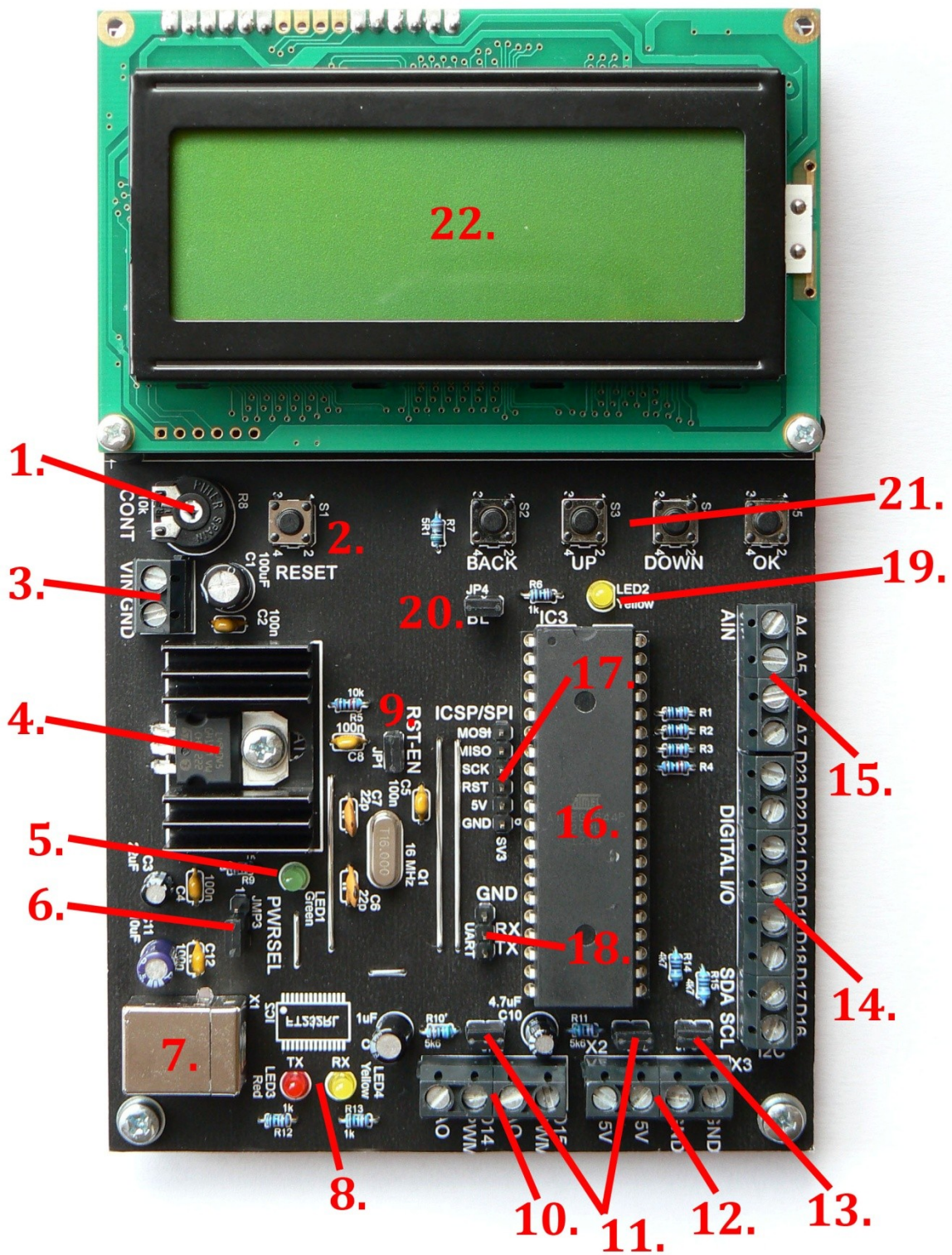
Tab. 10 Základní parametry mikrokontroléru Atmega644P [17]

6.2 Popis jednotlivých prvků

V následujícím seznamu jsou shrnuty všechny důležité prvky, indikátory a konektory (nebo svorkovnice) spolu s popisem jejich funkce. Číslování je shodné s Obr. 15. Tento seznam má především doplnit schéma zapojení a objasnit funkce konkrétních prvků.

1. **Trimr** - volba kontrastu LCD displeje.
2. **Resetovací tlačítko** – způsobí kompletní reset kontroléru a následně se znovu spustí obslužný program přípravku.
3. **Vstupní svorkovnice napájení (VIN)** – sem lze připojit zdroj stejnosměrného napětí, které následně přes stabilizátor napájí celý přípravek (volba jumperem PWRSEL)
4. **Stabilizátor napětí** – výstup 5 V
5. **LED1** (zelená) – indikuje připojené napětí ke svorkovnici VIN
6. **Jumper PWRSEL** – volba napájení, přesunutím směrem ke stabilizátoru se aktivuje napájení ze svorkovnice VIN. V poloze blíže k USB konektoru je zvoleno napájení z PC.
7. **USB konektor** pro komunikaci a napájení
8. **LED3 a LED4** – indikátory komunikace přes USB.

9. **Jumper RST-EN** – povolení resetu přípravku z USB, musí být zapojen před programováním, jinak nelze nový program vůbec do přípravku nahrát. Při normálním použití je tento reset nežádoucí a jumper by měl zůstat rozpojen.
10. **Svorkovnice digitálních pinů D14 a D15** – lze je využít jako vstupy i výstupy. Nicméně vzhledem k možnosti generovat PWM a koncepci laboratorních úloh se předpokládá častější využití v roli výstupů. Volitelně může student k D14 a D15 připojit dolnoproductné RC články pomocí jumperů JP2 a JP3. Signály z RC článků jsou dostupné na svorkovnicích AO.
11. **JP2 a JP3** – jumpery připojující RC články k výstupům D14 a D16
12. **Svorkovnice napájení** – zdroj napětí 5 V pro napájení dalších zařízení a především země, která slouží jako reference pro všechny digitální i analogové vstupy/výstupy.
13. **JP5** – připojení pullup rezistorů I2C sběrnice k pinům D17 (SDA) a D16 (SCL). Pokud mají být použity jako obyčejné vstupy/výstupy, musí být jumper rozpojen.
14. **DIGITAL I/O** – svorkovnice digitálních vstupů/výstupů D16 až D23. Slouží k připojení výhradně TTL signálů v rozsahu 0 V až 5 V. Maximální rozsah vstupního napětí definuje katalogový list mikrokontroléru.[17]
15. **AIN** – svorkovnice analogových vstupů, která slouží k přivedení signálů z analogových výstupů měřicí karty. Vzhledem k tomu, že měřicí karty v učebně disponují výstupy v rozsahu ± 10 V, kdežto vstupy kontroléru akceptují pouze 0 V až 5 V, jsou ošetřeny předřadnými rezistory. Jejich úkolem je omezit proud tekoucí do vstupů, pokud nedopatřením dojde k nastavení příliš vysokého napětí na výstupu karty.
16. **Kontrolér Atmega 644P**
17. **Konektor ICSP/SPI** – může být využit buď k naprogramování přípravku pomocí programátoru nebo ke komunikaci přes rozhraní SPI. Předpokládá se využití především pro servisní účely nebo případně k rozšíření přípravku o další laboratorní úlohy zaměřené právě na SPI.
18. **UART** – volná sériová linka v TTL napěťových úrovních. Aktuálně nevyužitá, lze do budoucna začlenit do případných dalších laboratorních úloh.
19. **LED2** (žlutá) – indikuje funkci bootladeru, ale je možné ji rozsvítit i pomocí programu.
20. **BL** – jumper aktivující podsvícení displeje.
21. **Ovládací tlačítka přípravku** – slouží k výběru, spouštění a vypínání jednotlivých úloh. Rovněž také k přepínání dodatečných funkcí.
22. **LCD displej** – zobrazení čtyř řádků po 20 alfanumerických znacích. Primárním účelem je zobrazení menu k výběru úlohy a následné potřebných informací.



Obr. 15 Fotografie přípravku s očíslovanými prvky

6.3 Programování

Celý program pro přípravek byl vytvořen ve vývojovém prostředí Arduino a nahrán přes rozhraní USB. Přípravek je tedy s platformou Arduino plně kompatibilní, což umožňuje využít jednak programovací jazyk s velmi širokou nabídkou knihoven a rovněž podporu komunity uživatelů Arduina. Pouze před prvním použitím přípravku v čerstvě nainstalovaném vývojovém prostředí je třeba stáhnout dle pokynů na webu soubory umožňující provoz platformy Arduino na kontroléru Atmega644P. [18]

K upravování stávajících laboratorních úloh v přípravku nebo tvorbě nových tedy stačí pouze volně dostupné Arduino vývojové prostředí a USB kabel. Vzhledem filozofii platformy Arduino není potřeba k programování hlubší znalost technických detailů o kontroléru, jeho registrech a podobně. Díky tomu má vyučující možnost poměrně snadno přípravek parametrizovat a upravit program dle vlastních představ.

V případě výměny mikrokontroléru nebo výroby nového přípravku, je však třeba do prázdného mikrokontroléru nejprve nahrát bootloader, aby bylo možné jej následně programovat jako Arduino. K tomuto úvodnímu kroku se musí použít programátor nebo případně jiné Arduino. [19] [20]

6.4 Použití

Ovládání přípravku by mělo být pro studenty velmi jednoduché, po připojení napájení a úvodní obrazovce se na displeji zobrazí menu s výběrem úlohy. Pomocí tlačítek UP a DOWN se vybere požadovaná laboratorní úloha, tlačítkem OK se spustí. Každá úloha má svou obrazovku, jež obsahuje jednak informace o použitých vstupech a výstupech, ale také dalších funkcích jako jsou například úmyslné chyby. Spuštěnou úlohu ukončí stisknutí tlačítka BACK. V případě neočekávaného stavu, poruchy a podobně použijte tlačítko RESET, které restartuje mikrokontrolér a spustí znovu program.

Při používání přípravku, zejména během připojování signálů na vstupy a výstupy, je **nutné dodržovat parametry uvedené v katalogových listech použitých součástek**. Především se jedná o napěťové limity stabilizátoru napětí a mikrokontroléru.

6.5 Laboratorní úlohy

6.5.1 Obecné informace

Následující kapitoly charakterizují aktuální stav laboratorních úloh připravených v podobě programu pro mikrokontrolér využívající platformu Arduino. Do budoucna lze pochopitelně nabídku úloh rozšiřovat nebo upravovat parametry a vlastnosti úloh stávajících. Veškeré parametry respektive konstanty nebo výpočetní vztahy jsou ve zdrojovém kódu programu uvedeny na začátku jednotlivých funkcí a řádně okomentovány. Takto bude mít vyučující možnost poměrně rychle přenastavit úlohu dle svých potřeb. Některé z úloh mají zavedenou úmyslnou chybu, konkrétně se jedná o oscilátory, aby studenti v rámci laboratorních prací mohli tyto chyby vyhodnocovat. Chybu lze za běhu úlohy volitelně zapnout nebo vypnout pomocí tlačítek.

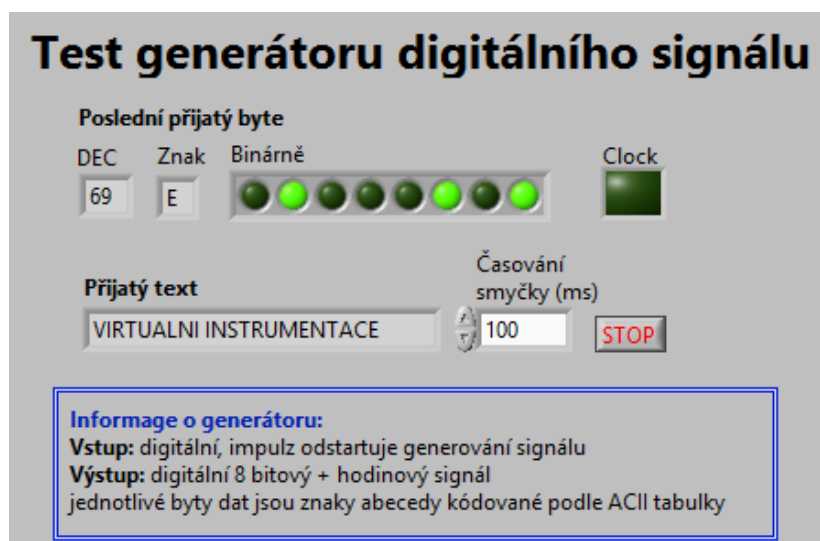


Tab. 11 Přehled obrazovek jednotlivých úloh včetně ukázky z menu pro výběr

6.5.2 Generátor digitálních signálů

Na osmibitovém výstupu (D16 až D23) přípravek generuje digitální signály – posloupnosti logických jedniček a nul. V každém okamžiku tedy naměříme na výstupu celkem osm logických úrovní, které dohromady reprezentují jeden bajt dat. K těmto datovým signálům se rovněž generuje i signál hodinový na pinu D15. Z hlediska principu se tedy jedná o paralelní přenos dat, přičemž cílové zařízení musí data číst vždy v okamžiku vzestupné hrany hodinového signálu. Jednotlivé bajty odeslaných dat jsou kódovány pomocí ASCII tabulky a při správném vyhodnocení odpovídají písmenům abecedy.

Přípravek data negeneruje trvale (cyklicky), ale pouze na základě detekce vzestupné hrany na vstupu D14. Teprve pak začne generovat hodinový signál a postupně odesílat jednotlivé bajty dat. Jakmile jsou odeslána všechna data, ukončí se i generování hodinového signálu. Rychlost přenosu dat (frekvenci hodin) může vyučující změnit v programu přípravku. Výchozí nastavení je 5 Hz, jinak řečeno 5 bajtů za sekundu, pro větší názornost principu přenosu dat a ladění může být vhodné rychlost ještě snížit.



Obr. 16 Vzorové VI ke generátoru digitálních signálů. Bylo přijato celkem 23 znaků (bajtů), které ve výsledku tvoří text „VIRTUALNI INSTRUMENTACE“.

6.5.3 Napětím řízený oscilátor

Obdélníkový signál oscilátoru s pevnou střídou a proměnnou frekvencí je generován na výstupu D15 v závislosti na změřeném napětí na analogovém vstupu A7. V režimu s vypnutou úmyslnou chybou se závislost frekvence na napětí nemění pro celý rozsah vstupu a definuje ji vztah (1)

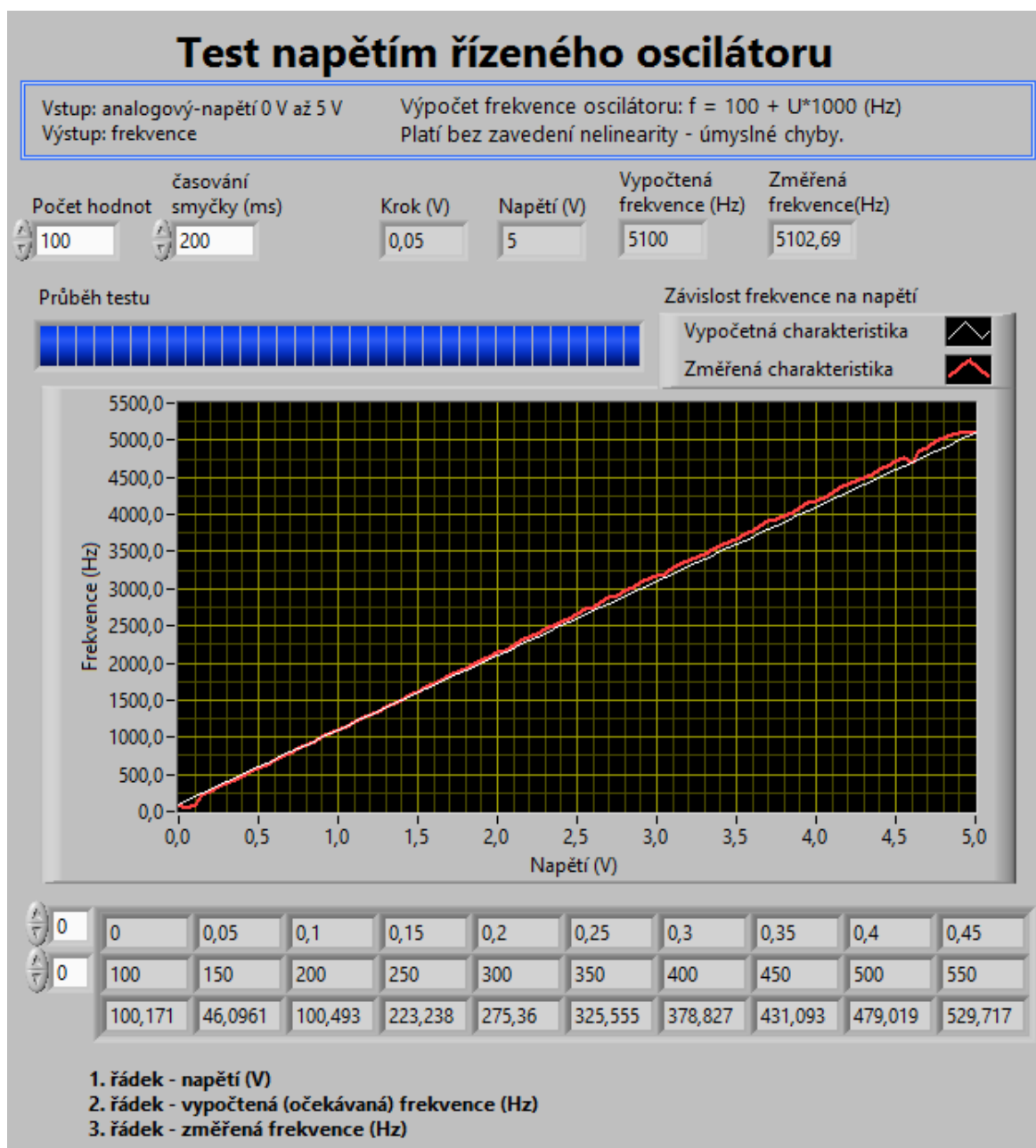
$$f = \text{offset} + k_0 \cdot U$$

offsetminimální frekvence oscilátoru (Hz)
 k_0konstanta úměrnosti
 Unapětí na vstupu (V)

(1)

Konstanta k_0 má výchozí hodnotu 1000, offset 100. Při nulovém vstupním napětí bude frekvence oscilátoru rovna 100 Hz. Po aktivování chyby jsou na rozsahu analogového vstupu definovány dvě lomové hodnoty napětí, které charakteristiku oscilátoru rozdělí na tři úseky. Každý úsek má svou závislost frekvence na napětí, jejíž předpis se ale shoduje s výše uvedeným vztahem. Liší se pouze konstanty. Změřená charakteristika oscilátoru – závislost frekvence na napětí, bude tedy vykazovat zřetelné odchylky od přímky, kterou bychom určili výpočtem ze vztahu shodného pro celý rozsah vstupu.

Při použití analogového výstupu měřicí karty musí studenti zohlednit, že vstupní rozsah AD převodníku mikrokontroléru je pouze 0 V až 5 V!



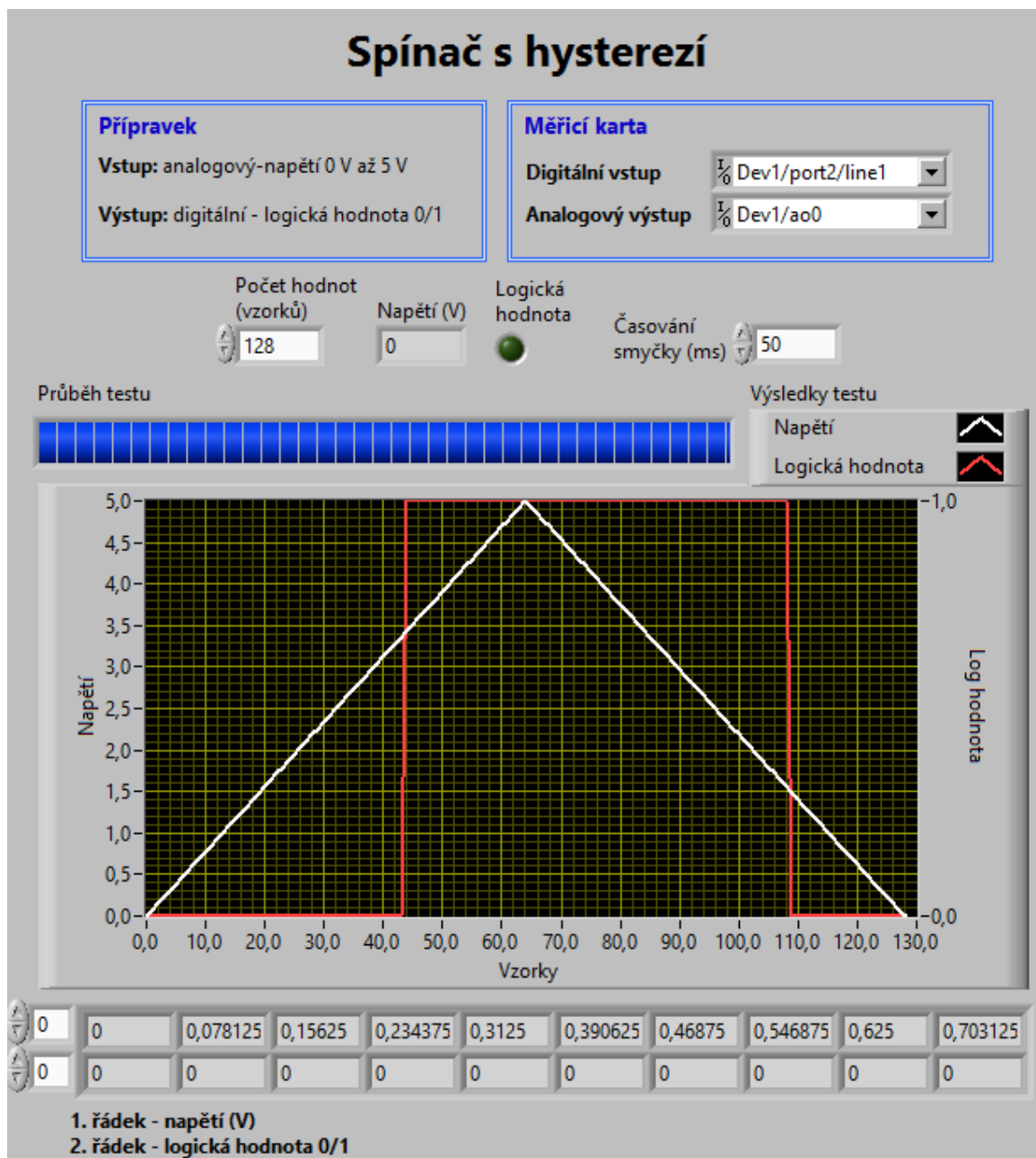
Obr. 17 Vzorové VI k napětím řízenému oscilátoru. Vypnutá chyba.

6.5.4 Spínač s hysterezí

Mikrokontrolér v přípravku vyhodnocuje napětí na analogovém vstupu A7, pokud překročí nastavenou horní mez, přejde výstup D15 do stavu logické jedničky. Při poklesu napětí pod dolní mez, bude na výstupu opět logická nula. Výchozí nastavení mezí v programu je 1,5 V pro dolní mez a 3,5 V pro horní mez. Funkci spínače s hysterezí je možné pomocí tlačítek invertovat. V tomto případě při překročení horní meze přejde výstup do LOG 0, následně při poklesu pod dolní mez do LOG 1.

Při použití analogového výstupu měřicí karty musí studenti zohlednit, že vstupní rozsah AD převodníku mikrokontroléru je pouze 0 V až 5 V!

Vzorové VI ke spínači s hysterezí (Obr. 18) generuje jednu periodu trojúhelníkového signálu v rozmezí 0 V až 5 V a vyhodnocuje logickou úroveň na výstupu. Výsledky prezentuje ve formě grafu a ukládá do pole.



Obr. 18 Vzorové VI ke spínači s hysterezí. V grafu je patrná změřená charakteristika bez invertování.

6.5.5 Číslicový oscilátor

V této úloze přípravek vyhodnocuje logické úrovně na 8 bitovém vstupu (portu mikrokontroléru) – piny D16 až D23, které reprezentují jeden bajt - rozsah hodnot 0 až 255. Od této hodnoty je odvozena frekvence obdélníkového signálu s pevnou střídou 50% na výstupu D15. Oscilátor nabízí dva režimy –

s vypnutou a zapnutou chybou. V případě vypnuté chyby je závislost frekvence lineární v celém rozsahu vstupu. Frekvenci lze určit na základě vztahu (2).

$$f = \text{offset} + k0 \cdot \text{portValue}$$

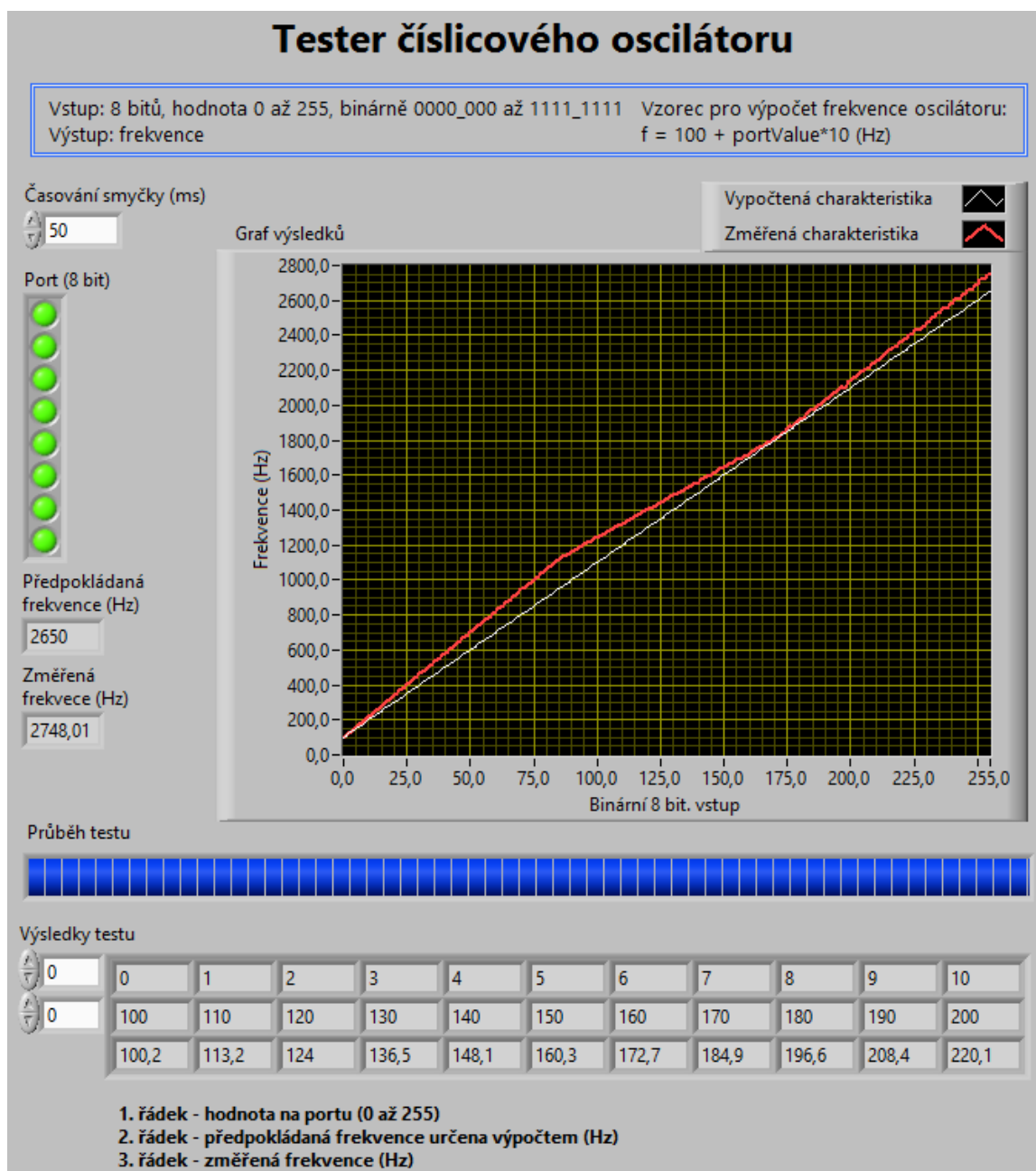
offset.....minimální frekvence oscilátoru
k0.....konstanta úměrnosti
portValue.....hodnota z 8 bitového vstupu

(2)

Parametry *offset* a *k0* lze upravit v programu přípravku, výchozí hodnoty jsou *offset* = 100 Hz, *k0* = 10. V případě, že budou všechny vstupy ve stavu logické nuly, na výstupu naměříme signál s frekvencí 100 Hz.

Chybu lze zapnout nebo vypnout pomocí tlačítek OK a DOWN. Pokud je chyba aktivní, jsou na rozsahu vstupu definovány dvě lomové hodnoty, které oddělují intervaly s různými parametry pro výpočet frekvence. Tato změna se při testování projeví jako výrazná chyba linearity, kterou mohou studenti vyhodnocovat.

Použití číslicového oscilátoru demonstruje příklad v LabVIEW (Obr. 19), který funguje jako jednoduchý tester. Postupně projde všech možných 256 kombinací, které lze na vstup přípravku přivést a měří výstupní frekvenci. Výsledek testu prezentuje graficky a všechny hodnoty ukládá do pole. Pro porovnání rovněž vypočítá pro každou hodnotu vstupu frekvenci, kterou bychom očekávali na základě znalosti funkce oscilátoru. V režimu s vypnutou chybou by se měla vypočtená a změřená charakteristika přibližně překrývat. Se zapnutou chybou bude patrná výrazná odchylka.

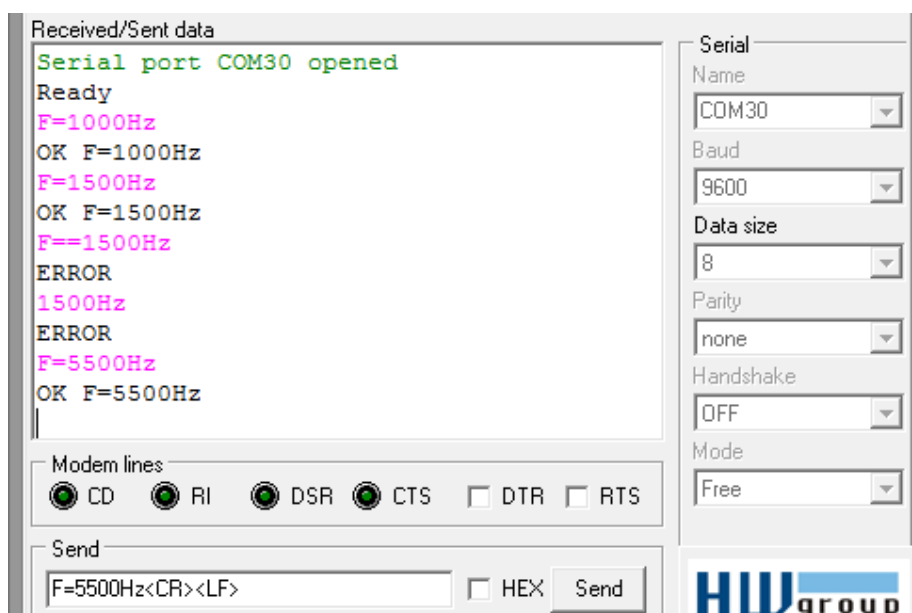


Obr. 19 Vzorový příklad v LabVIEW využívající přípravek v režimu číslicového oscilátoru (změřená charakteristika se zapnutou chybou)

6.5.6 Číslicový oscilátor USB

Frekvence oscilátoru se nastavuje pomocí příkazů odeslaných přes USB - virtuální sériovou linku s modulační rychlostí 9600 Bd. Každý příkaz musí splňovat formát „F=xxxHz“, kde xxx reprezentuje požadovanou frekvenci. Například odesláním „F=1000Hz“ tedy nastavíme frekvenci 1 kHz. Za každým příkazem musí následovat znak LF (ASCII 10) [21], fungovat bude rovněž odeslání dvojice

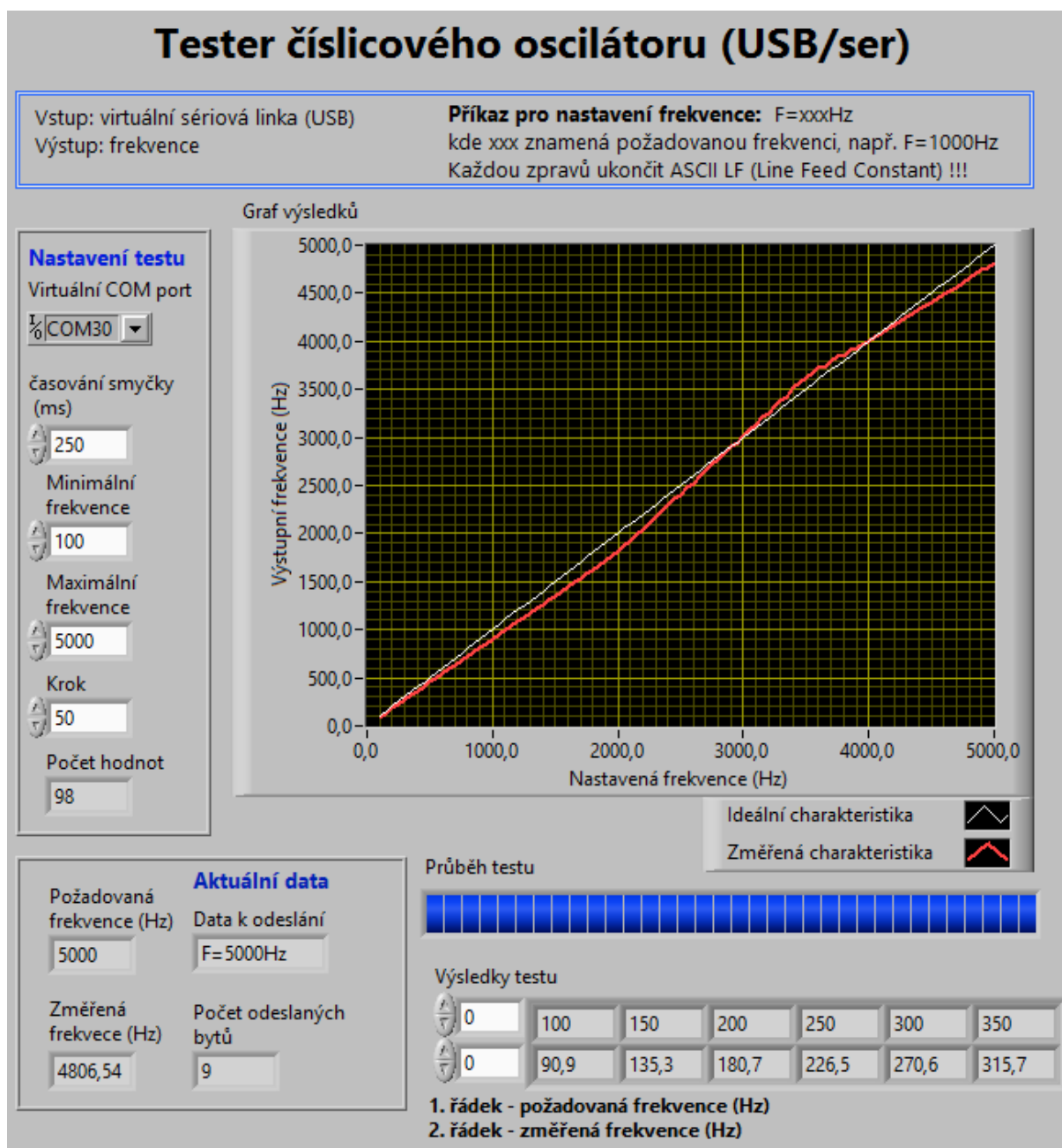
CR (ASCII 13) a LF. Platný příkaz přípravek potvrdí odesláním „OK F=xxxHz“, na neplatný příkaz odpoví „ERROR“. Za odpověďmi přípravku vždy následují znaky CR a LF.



Obr. 20 Ukázka komunikace s číslicovým oscilátorem

V režimu zapnuté chyby dojde k úpravě požadované frekvence dle zadaného vztahu, který lze upravovat změnou konstant v programu přípravku. Vztahy jsou definovány v programu celkem tři a charakteristika oscilátoru tedy obsahuje dvě lomové frekvence. Ty jsou ve výchozím nastavení 2000 Hz a 3500 Hz.

S aktivovanou chybou na výstupu naměříme frekvenci s podstatnou odchylkou od očekávané hodnoty. Pochopitelně i režim s vypnutou chybou je zatížen určitou nepřesností danou funkcí časovačů kontroléru, nicméně ta je vůči úmyslné chybě zanedbatelná.



Obr. 21 Vzorový příklad v LabVIEW využívající přípravek v režimu číslicového USB oscilátoru (změřená charakteristika se zapnutou chybou)

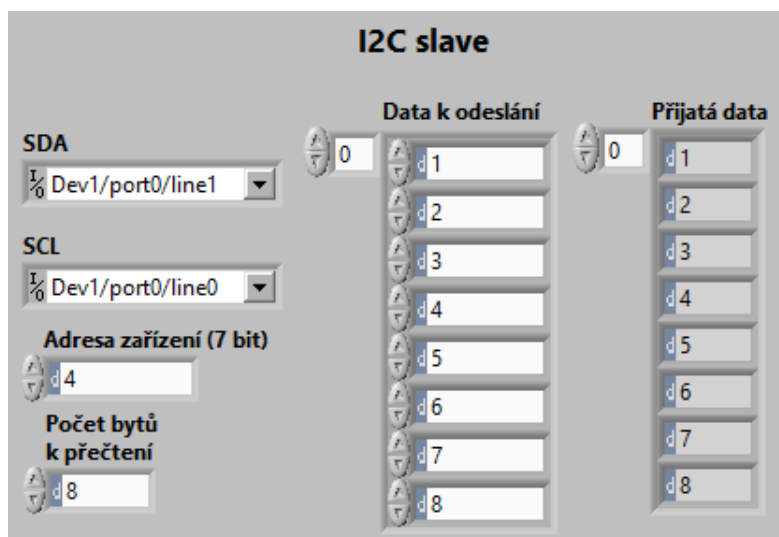
6.5.7 I2C slave

Pro laboratorní úlohy zaměřené na komunikaci přes sběrnici I2C nabízí přípravek režim, ve kterém se připojí ke sběrnici jako slave s předem definovanou adresou. Tu může změnit vyučující v programu, výchozí hodnota je 4 (dekadicky), neboli 100 binárně. Po aktivaci režimu tlačítkem OK bude zařízení vyčkávat, dokud mu master po sběrnici neodešle nějaká data. Jakmile k tomu dojde, uloží je do proměnné typu pole (bufferu) a vypíše na displej v desítkové soustavě. Každý bajt dat bude tedy zobrazen jako číslo od 0 do 255. Z důvodu velikosti displeje a přehlednosti pojme buffer pouze 8 bajtů, pokud počet přijatých dat tuto hodnotu překročí, začne se do bufferu zapisovat znovu od

začátku. Pokud odešle student do zařízení například celkem 9 bajtů, objeví se mu na displeji pouze ten poslední, s tímto faktem je třeba počítat.

V případě, že si master na sběrnici vyžádá čtení dat z přípravku, odešle mu mikrokontrolér naposledy přijaté bajty. Pokud použijeme předchozí příklad, když student odeslal 9 bajtů, dorazí mu zpět pouze jeden – poslední. Pokud v posledním zápisu obdržel přípravek maximálně 8 bajtů, odešle je zpět všechny. Při použití knihovny I2C komunikace platforma Arduino sama povolí interní pull-up rezistory, které jsou pro funkci sběrnice nezbytné. Jejich vysoká hodnota v řádu desítek kiloohmů [17] ale nemusí správné fungování vždy zaručit. Řešením je aktivace dodatečných pull-up rezistorů (4k7) na přípravku pomocí jumperu JP5.

Pro komunikaci po I2C vzniklo v rámci bakalářské několik vzorových VI, které demonstrují implementaci sběrnice v LabVIEW na nejnižší úrovni a to pomocí běžné měřicí karty a pouze softwarového časování. Tomuto tématu je věnována samostatná kapitola - 8 I2C pro LabVIEW.



Obr. 22 Vzorové VI využívající přípravek v režimu I2C slave (odeslaná a přijatá data se shodují)

Po ukončení práce s úlohou I2C slave **musí být před spuštěním jiné úlohy proveden reset přípravku**, jinak zůstanou piny D16 (SCL) a D17 (SDA) vyhrazeny pro komunikace po sběrnici a nebudou korektně fungovat.

6.5.8 Digitální teploměr

Přípravek s mikrokontrolérem v této úloze působí jako prostředník mezi digitálním teplotním čidlem DS18B20 (další informace v kapitole 7.8) a počítačem. Změřená teplota je odesílána skrze sériovou linku s rychlostí 9600 Bd v podobě textového řetězce. Formát řetězce je následující: „Teplota = xx.xx C“, přičemž písmena „x“ reprezentují údaj o teplotě s pevný počtem dvou desetinných míst. Každý řetězec s teplotou následuje dvojice znaků CR a LF. Perioda měření a tedy i časový rozestup mezi odesláním dvou údajů o teplotě je jedna sekunda. Pro správnou funkčnost úlohy musí být propojena svorkovnice TMPD (přípravek se senzory) a svorkovnice D14 (přípravek s mikrokontrolérem). Napájení přípravku se senzory může být realizováno ze svorkovnic 5V a GND na přípravku s mikrokontrolérem.

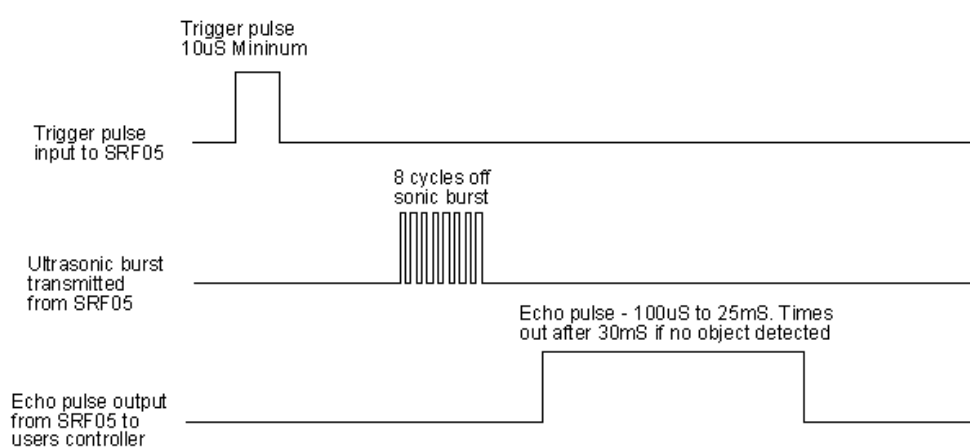
6.5.9 IMU

Použití tohoto režimu souvisí s kapitolou 7.10 Inertial Measurement Unit (IMU). Přípravek zajišťuje komunikaci s jednotlivými senzory v IMU pomocí sběrnice I2C a rovněž provádí vyhodnocení změřených veličin. Do počítače už odesílá výsledná data pomocí sériové linky. V této úloze je použita vyšší modulační rychlost – 115200 Bd. Při každém spuštění úlohy je potřeba provést inicializaci, při které musí být IMU v klidu a vodorovné poloze. Aby se zamezilo nechtěným chybám během inicializace, zobrazí přípravek po spuštění úlohy krátce informace o zapojení a následně upozorní studenta, že senzor musí být v klidu. Samotná inicializace se spustí až po stisknutí tlačítka. Další informace obsahuje již výše zmíněná kapitola 7.10.

7.3 Ultrazvukový dálkoměr SRF05

Zapnutí napájení pro ultrazvukový modul SRF05 se provádí pomocí jumperu JP2 a je indikováno zelenou LED5. Modul zvládne měřit pomocí odrazu ultrazvukových vln vzdálenost teoreticky až do 4 metrů. Vzhledem k účelu přípravku a poměrně širokému úhlu vyzařování senzoru se ale v laboratorních úlohách uplatní měření spíše jen v řádu několika desítek centimetrů.

K odstartování měření musí senzor detekovat na vstupu trigger (svorkovnice TRIG) kladný impuls dlouhý alespoň 10 μ s. Následně vyšle do prostoru 8 ultrazvukových impulsů a naslouchá případným odrazům. Výstupní signál senzoru (svorkovnice ECHO) má podobu kladného impulsu trvajícího 100 μ s až 25 ms, který je přímo úměrný vzdálenosti detekovaného předmětu. V případě, že se v dosahu senzoru žádná překážka nenalézá, trvá impuls 30 ms.[22]



Obr. 24 Průběh signálů na vstupu a výstupu senzoru SRF05 v čase [22]

Přesněji řečeno, délka impulsu odpovídá času, za který ultrazvukové vlny ze senzoru dorazily k překážce a zase zpět, dostaneme tedy informaci o dvojnásobku měřené vzdálenosti. Tuto skutečnost musíme zohlednit ve vztahu pro výpočet vzdálenost z délky impulsu (3), stejně tak jako rozdílnou rychlost šíření zvuku ve vzduchu v závislosti na teplotě (4).

$$2l = \frac{t_p}{10^6} \cdot c_{22} \cdot 100$$

lvzdálenost (cm) (3)

t_pdélka pulzu (μ s)

c_{22} ...rychlost zvuku ve vzduchu při 22°C

Rychlost zvuku c_{22} určíme ze vztahu [23 str. 224]:

$$c_{22} = 331,8 + 0,606t = 331,8 + 0,606 \cdot 22 = 345,132 \text{ m} \cdot \text{s}^{-1} \quad (4)$$

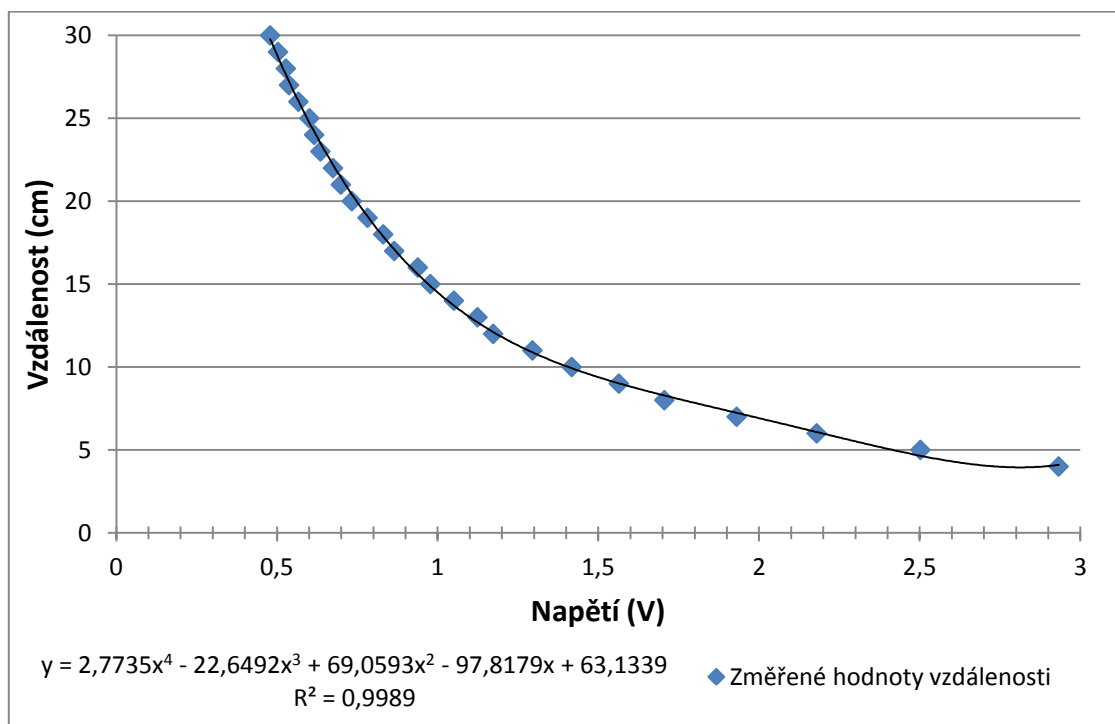
Po dosazení a úpravě:

$$2l = \frac{t_p}{10^6} \cdot 345,132 \cdot 100$$
$$l = \frac{t_p}{2 \cdot 10^4} \cdot 345,132 \quad (5)$$
$$l = t_p \cdot 0,0172566$$

Poslední dvě vyjádření vztahu (5) můžeme pohodlně použít k vypočtení vzdálenosti.

7.4 Infračervený dálkoměr

Senzor Sharp GP2Y0A41 využívající odrazu infračerveného paprsku pracuje v rozsahu 4 cm až 30 cm. Výstupem je napětí na svorkovnici IR, jehož hodnota se mění spolu s vzdáleností od překážky. Závislost napětí je bohužel silně nelineární, navíc s rostoucí vzdáleností podstatně klesá citlivost senzoru. [24] Tyto fakta výrazně snižují přesnost, která umožňuje využití infračerveného dálkoměru pouze pro orientační měření. Z experimentálně změřených dvojic hodnot napětí-vzdálenost lze ale sestavit grafickou závislost a následně ji pomocí softwaru (například Microsoft Excel) proložit křivkou polynomické regrese. Předpis funkce regresivní křivky (polynom) poslouží v LabVIEW pro orientační přepočet změřeného napětí na vzdálenost.

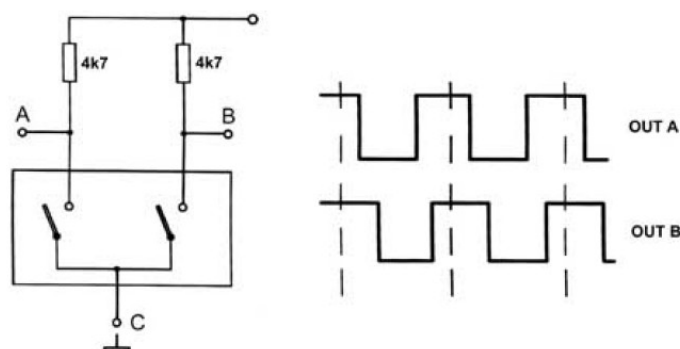


Obr. 25 Graf závislosti vzdálenosti na měřeném napětí s regresivní křivkou a vzorcem pro její výpočet

Vzorec v levé dolní části grafu je možné do LabVIEW vložit například ve formě Expression Node, kde jako hodnotu x přivedeme napětí ze senzoru ve voltech a výstupem y bude vzdálenost v centimetrech. Stejný princip přepočtu je součástí demonstračního VI k přípravku.

7.5 Mechanický inkrementální snímač

Mechanický inkrementální snímač poskytuje rozlišení 24 impulzů na otáčku, výstupy jsou dva – A, B vzájemně fázově posunuté o 90° elektrických.[25] Vzhledem k principu funkce (mechanické spínače) bylo nutné výstupy opatřit RC článkem a invertory se Schmittovým klopným obvodem pro eliminaci zákmitů. Bez něj pochopitelně LabVIEW po vyhodnocení signálů poskytovalo naprosto nesprávné údaje o natočení. Výstupy ze snímače nalezneme na přípravku na svorkovnicích ENCA a ENCB.



Obr. 26 Princip funkce mechanického inkrementálního snímače [25]

7.6 Analogový teploměr TMP36GT9Z

Integrovaný obvod v pouzdře TO-92 je schopen měřit teplotu v intervalu od -40°C do +125°C s typickou přesností $\pm 1^\circ\text{C}$. Výstupní napětí závisí na teplotě s konstantou úměrnosti 10 mV/°C a offsetem 0,5 V. Při teplotě 25°C tedy naměříme 0,75 V. [26] Z těchto údajů snadno vytvoříme vztahy pro přepočítání napětí na teplotu (6) a (7). K připojení teploměru k měřicí kartě slouží svorkovnice TMP.

$$t_{\circ\text{C}} = \frac{V_{\text{out}} - \text{offset}}{k_t} \quad (6)$$

Po dosažení $\text{offset} = 0,5 \text{ V}$ a $k_t = 0,01 \text{ V}$ získáme:

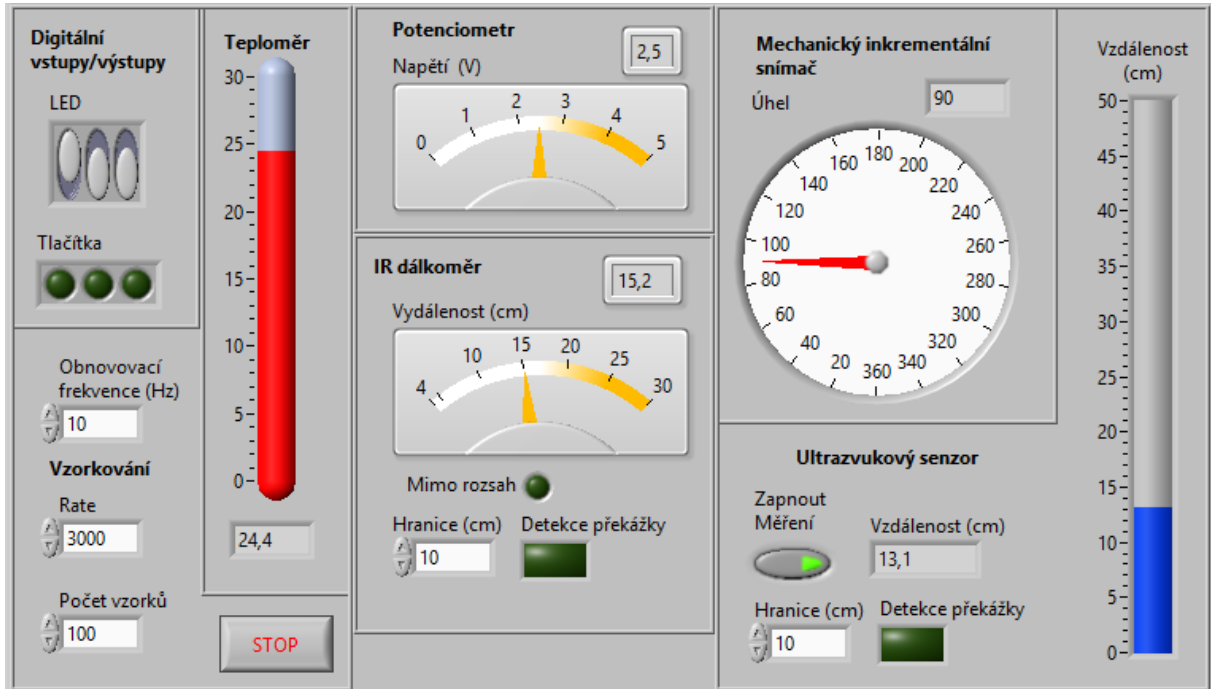
$$t_{\circ\text{C}} = \frac{V_{\text{out}} - 0,5}{0,01} \quad (7)$$

V_{out}napětí z teploměru (V)
 $t_{\circ\text{C}}$teplota (°C)

7.7 Společné demonstrační VI

Použití většiny senzorů na přípravku demonstruje společné vzorové VI. Konkrétně se jedná o tlačítka a LED, oba dálkoměry, analogový teploměr, potenciometr a mechanický inkrementální snímač. Principy měření a vztahy pro vyhodnocení signálu ze senzorů, které jsou v tomto VI použity, popisují samostatné kapitoly 7.2 až 7.6. Pomocí ovládacího prvku *obnovovací frekvence* se mění četnost aktualizace údajů na obrazovce (rovněž jak často karta fyzicky skutečně měří). V případě analogových senzorů dojde během měření k získání většího množství vzorků (nastavení *Počet vzorků*), ze kterých se následně vypočte průměr. Rychlost vzorkování karty ve vzorcích za sekundu lze měnit prvkem

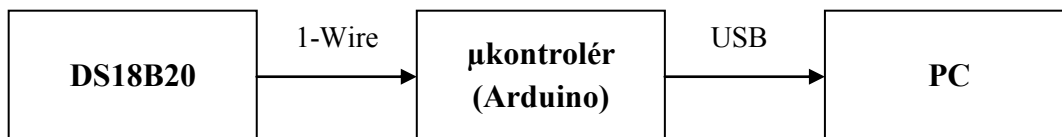
Rate. U obou dálkoměrů funguje detekce překážky na základě nastavené hraniční vzdálenosti, pokud je změřená vzdálenost menší, dojde k rozsvícení indikátorů. Funkci ultrazvukového dálkoměru podmiňuje generování startovacích impulzů, které se ovládá prvkem *Zapnout měření*.



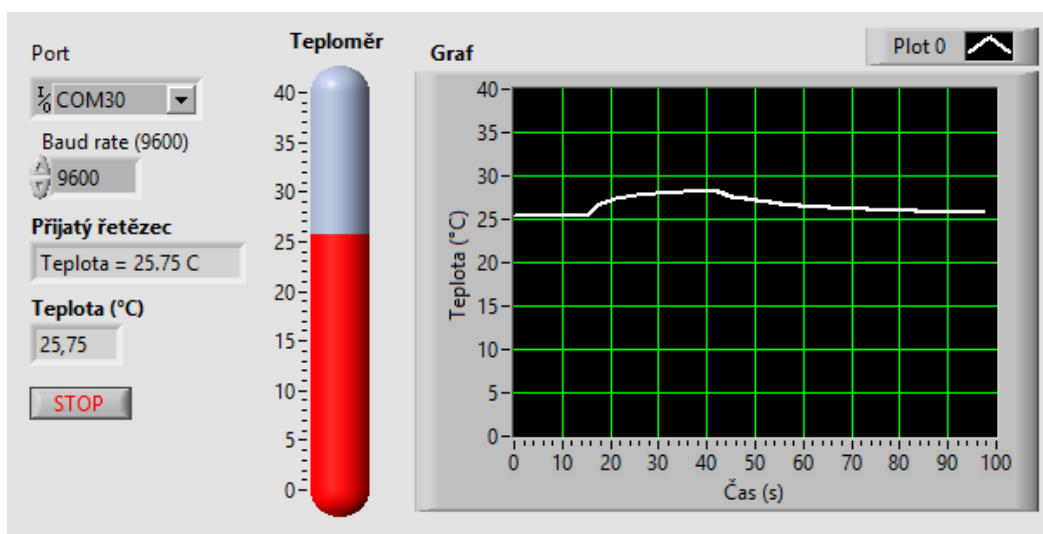
Obr. 27 Vzorové VI k přípravku se senzory

7.8 Digitální teploměr DS18B20

Digitální teploměr s rozsahem $-55\text{ }^{\circ}\text{C}$ až $+125\text{ }^{\circ}\text{C}$ a přesností měření $\pm 0,5\text{ }^{\circ}\text{C}$ na intervalu $-10\text{ }^{\circ}\text{C}$ až $+85\text{ }^{\circ}\text{C}$. Rozlišení může volit uživatel od 9 do 12 bitů. Ke komunikaci s čidlem slouží sběrnice 1-Wire, na které je každé zařízení identifikováno jedinečným 64 bitovým sériovým číslem.[27] Jak již vyplývá z názvu sběrnice, využívá ke komunikaci pouze jeden vodič, který je připojen ke svorkovnici TMPD. Využití čidla je prozatím podmíněno jeho připojením k přípravku s mikrokontrolérem, který obstará komunikaci po sběrnici 1-Wire a změřenou teplotu odešle přes USB do PC jako textový řetězec. Následně může student v LabVIEW tento údaj dále zpracovat, ukládat atd. Implementovat komunikaci po 1-Wire sběrnici přímo do LabVIEW nebylo předmětem této bakalářské práce. Pro platformu Arduino existují knihovny s high-level funkcemi jak pro sběrnici 1-Wire, tak přímo pro čidlo DS18B20. Proto nebudou detaily komunikace a funkce čidla dále rozebírány. Z pohledu studenta vznikne po propojení přípravků teploměr s výstupem přes virtuální sériovou linku (USB), se kterým bude pracovat.



Obr. 28 Použití snímače DS18B20



Obr. 29 Vizualizace k digitálnímu teploměru

7.9 Hodiny reálného času (RTC)

Modul firmy Sparkfun tvoří integrovaný obvod DS1307 spolu s krystalem a zálohovací 3V baterií, která zaručuje chod hodin i při výpadku napájení. Pomocí sběrnice I2C získáme z RAM paměti obvodu informace o čase – sekundy, minuty, hodiny, den v měsíci, měsíc, den v týdnu. Signální vodiče sběrnice jsou na přípravku přístupné skrze svorkovnice SDA (data) a SCL (hodiny). Vzhledem k tomu, že modul je možné od sběrnice fyzicky odpojit pomocí dvojitého mikrospínače SW1 (označený rovněž Connect), musí být před použitím přepínače v poloze ON. RAM obvodu DS1307 tvoří 56 bajtů (členění 56 x 8 bitů). Pouze prvních 8 bajtů slouží jako registry pro hodiny reálného času, zbytek zůstává volný. 7 bajtů obsahuje informace o čase, 8. bajt nastavuje chování výstupu s obdélníkovým signálem[28], který ale na přípravku není vyveden.

Členění paměti zachycují Obr. 30 a Obr. 31. Jednotlivé bajty obsahují údaje v BCD formátu (Binary Coded Decimal), čtveřice nejméně významných bitů (0 až 3) jsou jednotky, zbylé bity tvoří desítky dekadického čísla. Bity 4 až 7 mohou rovněž napevno obsahovat nuly (den), případně další nastavení nebo informace – záleží, zda je nastaven 24 nebo 12 hodinový režim. Veškeré podrobnosti jsou dostatečně rozebrány v katalogovém listu.[28]

V 24 hodinovém režimu nastaveném na přípravku stačí přečíst všech 7 bajtů a uplatnit na ně stejný algoritmus převodu z BCD. Pouze k číselnému údaji o roce se musí přičíst 2000, protože registr obsahuje pouze poslední dvojčíslí z letopočtu (pro letošní rok 14). Výše popsany postup využívá i vzorový příklad v LabVIEW (Obr. 32). Nejprve do hodin zapíše bajt s hodnotou nula, čímž nastaví register pointer na začátek paměti (adresu 0). Poté přečte 7 bajtů, které uloží do pole a nakonec je převede z BCD formátu na dekadický. Detailnější informace o I2C obsahuje samostatná kapitola - 8 I2C pro LabVIEW.

| | |
|-----|---------------|
| 00H | SECONDS |
| | MINUTES |
| | HOURS |
| | DAY |
| | DATE |
| | MONTH |
| | YEAR |
| 07H | CONTROL |
| 08H | RAM 56 x 8 |
| 3FH | |

Obr. 30 Přehled obsahu paměti DS1307 [28]

| | | | | | | | | | | |
|-----|------|------------|--------------|-------------|---------|-----|-----|-----|----------------------------|------|
| | BIT7 | | | | | | | | | BIT0 |
| 00H | CH | 10 SECONDS | | | SECONDS | | | | 00-59 | |
| | 0 | 10 MINUTES | | | MINUTES | | | | 00-59 | |
| | 0 | 12 / 24 | 10 HR A/P | 10 HR | HOURS | | | | 01-12 00-23 | |
| | 0 | 0 | 0 | 0 | 0 | DAY | | | | 1-7 |
| | 0 | 0 | 10 DATE | | DATE | | | | 01-28/29 01-30 01-31 | |
| | 0 | 0 | 0 | 10 MONTH | MONTH | | | | 01-12 | |
| | | 10 YEAR | | | YEAR | | | | 00-99 | |
| 07H | OUT | 0 | 0 | SQWE | 0 | 0 | RS1 | RS0 | | |

Obr. 31 Detailní členění registrů RTC v paměti DS1307 [28]

Hodiny reálného času (RTC)

SDA
Dev1/port2/line3

SCL
Dev1/port2/line2

Adresa (7 bit)
104

Byte k odeslání
0

Počet bytů k přijetí
7

Pro RTC DS1307:
Adresa: 104
Odeslat 0 (nastaví register pointer na začátek paměti)
Přijmout 7 bytů

Údaje z RTC

| | | |
|--------|--------|---------|
| Hodiny | Minuty | Sekundy |
| 16 | 23 | 20 |
| Den | Měsíc | Rok |
| 18 | 3 | 2014 |

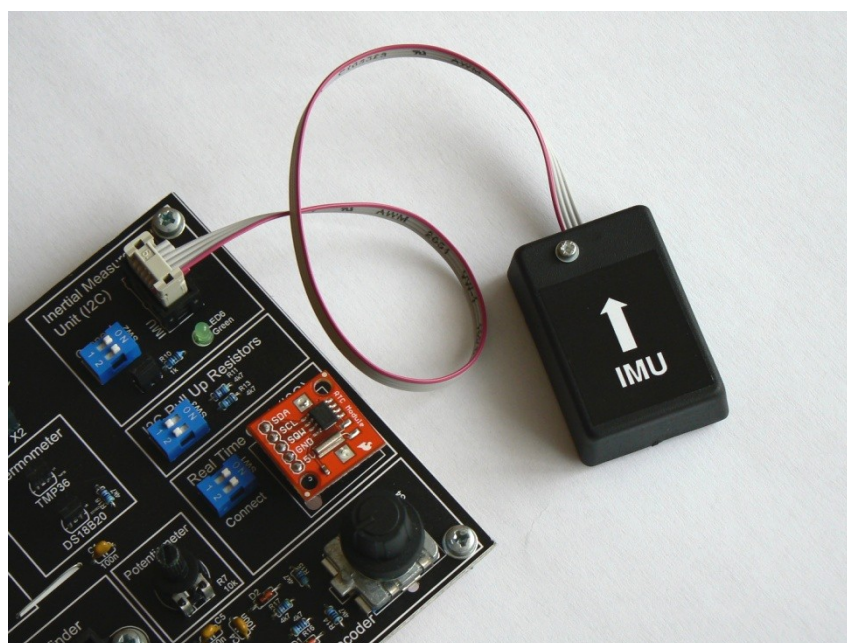
Den v týdnu: úterý

Obr. 32 Příklad použití hodin reálného času v LabVIEW

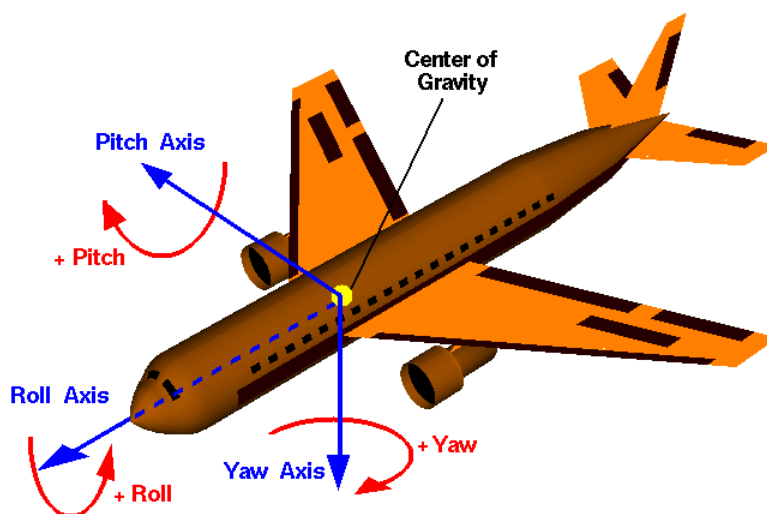
7.10 Inertial Measurement Unit (IMU)

IMU dodávaná firmou Pololu jako položka #1268 (MinIMU-9 v2) obsahuje tříosý akcelerometr LSM303DLHC, tříosý gyroskop L3GD20 a rovněž tříosý magnetometr.[29] Plastové pouzdro se senzorem se k přípravku připojuje pomocí plochého kabelu s konektorem. Vzhledem k vyšší ceně a rovněž podstatně složitější podstatě funkce v porovnání s ostatními senzory se předpokládá použití IMU pouze jako volitelného doplňku pro rozšíření výuky. Nepřítomnost IMU nijak neovlivní celkovou funkčnost přípravku, a proto jí nemusí být vybaveno každé pracoviště v učebně.

Pro využití IMU je nutné připojit přípravek se senzory k přípravku s mikrokontrolérem, jmenovitě se jedná o svorkovnice SDA a SCL (Na přípravku s mikrokontrolérem D17 a D16) a pochopitelně napájení 5 V a zem. Z názvu výše uvedených svorkovnic vyplývá, že senzor komunikuje po sběrnici I2C. Výrobce dodává k IMU zdrojový kód pro platformu Arduino, který zajistí veškeré výpočty se surovými daty ze všech tří senzorů a poskytne uživateli hotový výstup ve formě tří údajů definujících ve stupních polohu (náklon) senzoru v prostoru – pochopitelně ve třech osách. [30] Tyto údaje se v anglické literatuře označují jako roll, pitch, a yaw [31], jejichž význam objasňuje Obr. 34. Letadlo není zvoleno pro demonstraci těchto pojmů náhodou, ale proto, že prakticky se IMU uplatňují především v leteckém průmyslu.

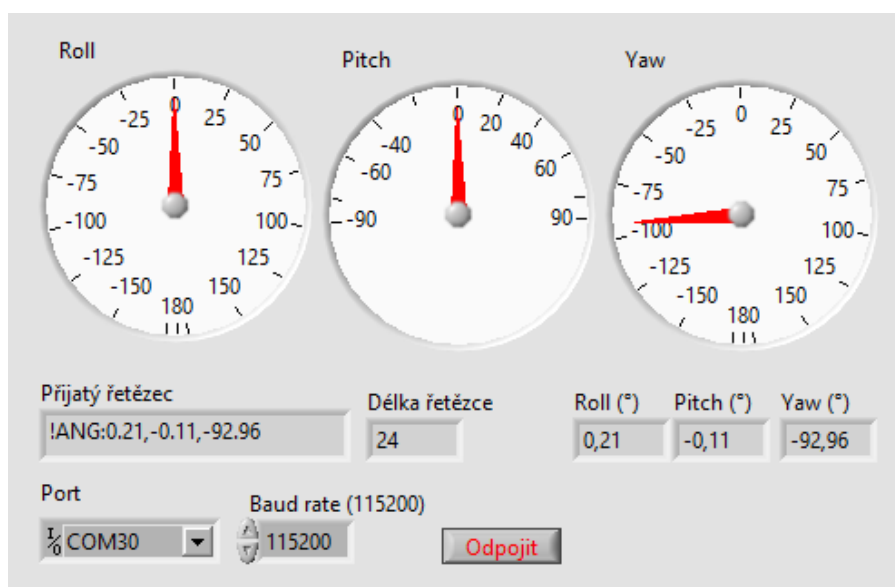


Obr. 33 Rozšíření přípravku o IMU



Obr. 34 Grafické znázornění významu údajů z IMU[31]

Údaje roll, pitch a yaw přípravek s mikrokontrolérem odesílá pomocí virtuální sériové linky do PC jako textový řetězec a úkolem studentů je další zpracování a grafická reprezentace pomocí LabVIEW. Formát textového řetězce je následující: „!ANG:0.21,-0.11,-92.96“ Přičemž jednotlivá desetinná čísla jsou náklony senzoru ve stupních a ve stejném pořadí, jak bylo výše uvedeno - roll, pitch a yaw. Modulační rychlost virtuálního sériového portu musí být nastavena na 115200 baudů.



Obr. 35 Příklad vizualizace veličin z IMU v LabVIEW

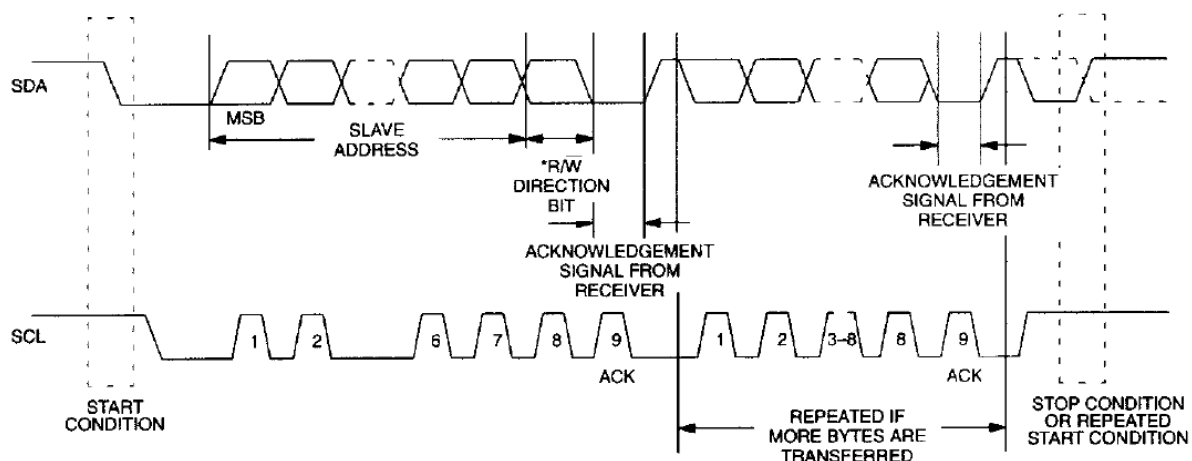
8 I2C pro LabVIEW

8.1 Základní informace

V rámci této bakalářské práce vznikly VI, které umožňují použít ke komunikaci přes sběrnici I2C kterékoliv digitální vstupy/výstupy na měřicí kartě. A to i na levnějších typech, které nepodporují hardwarové časování - například pro studenty určená karta NI myDAQ. Společnost National Instruments pochopitelně poskytuje kompletní řešení komunikace přes I2C v podobě zařízení NI USB-8452 včetně softwarové podpory. [32] Nicméně účelem následující části textu je ukázat, že komunikaci mohou studenti realizovat i na nejnižší úrovni pomocí měřicí karty, čímž si nejen rozšíří své dovednosti v LabVIEW, ale rovněž získají přehled o podstatě funkce sběrnice. Tento úkol může být řešen například v rámci semestrálního projektu.

Použití softwarového časování pochopitelně přináší určitá omezení, především v rychlosti. Místo standardního hodinového signálu s frekvencí 100 kHz, se generované hodiny pohybují okolo 1 kHz, přičemž tato frekvence ani není stabilní. Záleží na aktuálním vytížení procesoru PC, složitosti vykonávaného programu v LabVIEW atd. Proto funguje LabVIEW spolu s kartou na sběrnici vždy jako master a ostatní zařízení se podřizují jeho hodinovému signálu.

Komunikace na sběrnici I2C probíhá pomocí dvou datových vodičů SDA (data) a SCL (clock), přičemž klidovou hodnotu logické jedničky zajišťují pullup rezistory. Aby bit adresy nebo dat zařízení považovalo za platný, musí zůstat logická úroveň na vodiči SDA stabilní po celou dobu trvání kladného hodinového impulsu. Přesně zadaným přechodem stavů na vodičích SDA a SCL jsou dány speciální signály pro začátek a ukončení komunikace (start/stop condition). Zařízení přijímající data potvrzuje každý bajt pomocí takzvaného acknowledge bitu.[28]



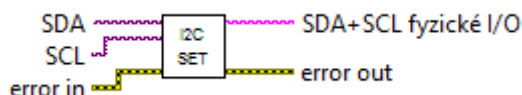
Obr. 36 Průběh signálů SDA a SCL během komunikace po sběrnici I2C[28]

V tomto ohledu jsou VI pro komunikaci zjednodušena. Pokud karta do cílového zařízení zapisuje data, vygeneruje sice jakožto master hodinový impuls pro potvrzovací bit, ale nekontroluje, zda cílové zařízení data opravdu potvrdilo nebo ne. Tato operace by vyžadovala přepnutí příslušného pinu karty z výstupu na vstup, což by zbytečně celý algoritmus zkomplikovalo a rovněž zpomalilo. V případě,

kdy karta z cílového zařízení čte, tak potvrzovací bit generuje, protože jinak by nedošlo k odeslání dalšího bajtu. [28]

8.2 Nastavení

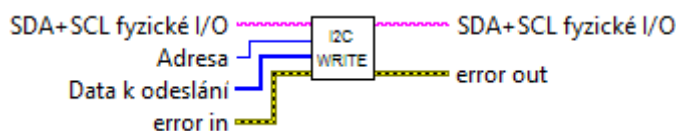
VI zajistí počáteční nastavení pro připojení k I2C sběrnici, na které měřicí karta funguje jako master. Volí se dva digitální I/O piny, které následně fungují jako SDA a SCL. Na fyzické výstupy se zatím nic nezapisuje.



Obr. 37 I2C set

8.3 Zápis dat

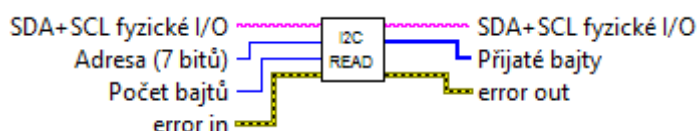
Zapíše na I2C sběrnici data (pole bajtů) určená pro zařízení se specifikovanou adresou. Pro odeslání pouze jednoho bajtu se musí jako parametr použít pole obsahující jen jeden prvek. Adresa je sice datového typu U8, ale dále se zpracovává pouze 7 bitů. Rozsah adres je tedy 0 až 127. Použití tohoto VI musí předcházet nastavení fyzických I/O pomocí I2C SET.



Obr. 38 I2C Write

8.4 Čtení dat

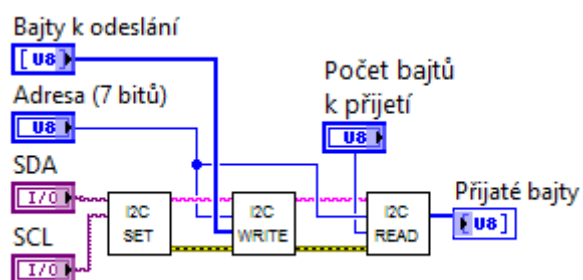
Přečte pomocí I2C sběrnice zadaný počet bajtů ze zařízení se specifikovanou adresou. Pro adresu platí stejný rozsah jako u zápisu. Přijátá data jsou ve formě pole bajtů – datový typ U8. Použití tohoto VI musí rovněž předcházet nastavení fyzických I/O pomocí I2C SET.



Obr. 39 I2C Read

8.5 Příklad použití

Nastavení vstupů a výstupů měřicí karty, zápis a čtení dat ze zařízení přes sběrnici I2C.



Obr. 40 Příklad použití VI pro komunikaci přes I2C

9 Závěr

V rámci bakalářské práce se podařilo naplnit zadání v podobě tří plně funkčních laboratorních přípravků, které poslouží ke zkvalitnění výuky virtuální instrumentace. Kromě hardwaru vznikla celá řada ukázkových aplikací v LabVIEW s cílem demonstrovat možnosti využití přípravků ve výuce. Tato komplexní sada hardwaru a softwaru může být využita jednak k tvorbě nových laboratorních úloh, které studenti plní průběžně, ale rovněž k zadání dlouhodobých samostatných prací respektive semestrálních projektů. Navíc znalosti a dovednosti, které si studenti osvojí při práci na těchto úlohách, přesahují hranice virtuální instrumentace a mohou být uplatněny i v jiných oblastech. Předně se jedná o praktické zkušenosti se zpracováním signálů z různých druhů senzorů nebo řízení jednoduchého elektrického pohonu. Při návrhu přípravků byly použity pouze cenově dostupné komponenty, které je možné běžně zakoupit například v obchodech se zaměřením na hobby elektroniku a robotiku. Díky tomuto faktu mají studenti možnost nabyté zkušenosti dále rozvíjet například i v rámci svých volnočasových aktivit a experimentů.

U všech tří přípravků lze najít prostor pro další vylepšení a rozvoj. Nejflexibilnější je v tomto ohledu přípravek s mikrokontrolérem, jehož kompletní program vznikl ve vývojovém prostředí pro Arduino a lze jej tedy velmi snadno modifikovat nebo doplnit o další úlohy. U přípravků se senzory se nabízí především implementace komunikačních protokolů k IMU a digitálnímu teplotnímu čidlu DS18B20 přímo do LabVIEW s využitím hardwarového časování na měřicí kartě. Rovněž komunikace pomocí sběrnice I2C byla v rámci bakalářské práce vyzkoušena pouze v omezeném rozsahu, ačkoliv je naprosto univerzální a bude možné ji využít k připojení dalších typů zařízení. Přípravek s motory je z hlediska možných vylepšení ze všech tří nejzavřenější, nicméně například při měření s akcelerometrem je možné využít nejen osu X, ale rovněž informace z osy Z, čímž se otevírá cesta ke komplikovanějšímu algoritmu pro výpočet orientace.

Mezi výhody zvolené koncepce přípravků patří rovněž univerzálnost. Připojení pomocí svorkovnic se sice může jevit jako neelegantní, ale hardware od společnosti National Instruments tento způsob standardně využívá. Takže i přes původní určení přípravků pro použití s kartou NI PCI 6221 bude možné je připojit například k NI myDAQ nebo myRIO, které jsou studentům přístupnější.

Negativa celého řešení se pravděpodobně projeví až při provozu v laboratoři, může se jednat zejména o slabiny v mechanickém provedení. Vzhledem k tomu, že dva přípravky jsou tvořeny pouze deskou plošných spojů s nekrytými komponenty, budou citlivé na nešetrné zacházení ze strany studentů. Je to ovšem daň za názornost přípravků, které nejsou pro studenta pouze „černou krabicí“ o jejímž obsahu nemá ponětí, ale vidí do detailu, s čím pracuje.

10 Citovaná literatura

1. **National Instruments.** NI ELVIS Compare and Buy. *National Instruments*. [Online] [Citace: 28. březen 2014.] <http://www.ni.com/ni-elvis/buy/>.
2. **Arduino.** Download the Arduino Software. *Arduino*. [Online] [Citace: 28. březen 2014.] <http://arduino.cc/en/Main/Software>.
3. —. Products. *Arduino*. [Online] [Citace: 28. březen 2014.] <http://arduino.cc/en/Main/Products>.
4. —. Arduino Uno. *Arduino*. [Online] [Citace: 29. březen 2014.] <http://arduino.cc/en/Main/ArduinoBoardUno>.
5. **AVR Libc community.** AVR Libc - Home Page. *AVR Libc*. [Online] [Citace: 31. březen 2014.] <http://www.nongnu.org/avr-libc/user-manual/index.html>.
6. **Wikipedia.** LabVIEW. *Wikipedia, the free encyclopedia*. [Online] 24. březen 2014. [Citace: 31. březen 2014.] <http://en.wikipedia.org/wiki/LabVIEW>.
7. —. Universal asynchronous receiver/transmitter. *Wikipedia*. [Online] [Citace: 25. březen 2014.] <http://en.wikipedia.org/wiki/UART>.
8. **National Instruments.** National Instruments VISA. *National Instruments*. [Online] [Citace: 28. prosinec 2014.] <https://www.ni.com/visa/>.
9. —. LabVIEW Interface for Arduino. *National Instruments*. [Online] [Citace: 25. březen 2014.] <https://decibel.ni.com/content/groups/labview-interface-for-arduino>.
10. **Pololu Robotics & Electronics.** Stepper Motor: Unipolar/Bipolar, 200 Steps/Rev, 42×48mm, 4V, 1.2 A/Phase. *Pololu Robotics & Electronics*. [Online] [Citace: 20. březen 2014.] <http://www.pololu.com/product/1200>.
11. **CNCshop.** S3806 rotační inkrementální enkodér. *CNCshop*. [Online] [Citace: 20. březen 2014.] <http://www.cncshop.cz/s3806-rotacni-inkrementalni-enkoder>.
12. **Pololu Robotics & Electronics.** A4988 Stepper Motor Driver Carrier. *Pololu Robotics & Electronics*. [Online] [Citace: 23. březen 2014.] <http://www.pololu.com/product/1182>.
13. **Clifford, Michelle a Gomez, Leticia.** Measuring Tilt with Low-g Accelerometers. *Freescale Semiconductor*. [Online] květen 2005. [Citace: 15. prosinec 2014.] http://cache.freescale.com/files/sensors/doc/app_note/AN3107.pdf.
14. **Pololu Robotics & Electronics.** Servo control interface in detail. *Pololu Robotics & Electronics*. [Online] [Citace: 23. březen 2014.] <http://www.pololu.com/blog/17/servo-control-interface-in-detail>.
15. **Vigor Precision LTD.** VS-2 SERVO. *Vigor Precision*. [Online] 30. červenec 2009. [Citace: 7. leden 2014.] <http://www.vigorprecision.com.hk/uploadfile/20120530/20120530145729271.pdf>.

16. **Pololu Robotics & Electronics.** MMA7361LC 3-Axis Accelerometer $\pm 1.5/6g$ with Voltage Regulator. *Pololu Robotics & Electronics*. [Online] [Citace: 28. leden 2014.] <http://www.pololu.com/product/1251>.
17. **Atmel Corporation.** ATmega644P. *Atmel Corporation*. [Online] [Citace: 25. listopad 2013.] <http://www.atmel.com/devices/atmega644p.aspx>.
18. **Sanguino.** Sanguino For Windows. *Sanguino*. [Online] [Citace: 10. listopad 2013.] <http://sanguino.cc/softwareforwindows>.
19. **Arduino.** Bootloader Development. *Arduino*. [Online] [Citace: 12. prosinec 2013.] <http://arduino.cc/en/Hacking/Bootloader>.
20. —. Using an Arduino as an AVR ISP. *Arduino*. [Online] [Citace: 6. listopad 2013.] <http://arduino.cc/en/Tutorial/ArduinoISP>.
21. **Margolis, Michael.** *Arduino Cookbook*. Sebastopol : O'Reilly, 2011. ISBN: 978-0-596-80247-9.
22. **Robot Electronics.** SRF05 - Ultra-Sonic Ranger. *Robot Electronics*. [Online] [Citace: 18. únor 2014.] <http://www.robot-electronics.co.uk/html/srf05tech.htm>.
23. **Mikulčák, Jiří.** *Matematické, fyzikální a chemické tabulky & vzorce*. Praha : Prometheus, 2008. ISBN: 978-80-7196-264-9.
24. **Sharp.** GP2Y0A41 - Analog Output Type Distance Measuring Sensor. *Shenzhen Dongbao Electronic*. [Online] 25. Duben 2005. [Citace: 25. leden 2014.] <http://www.szdongbao.com/pdf/GP2Y0A41SK0F.pdf>.
25. **GM Electronic.** Rotary Encoder RE24. *GM Electronic*. [Online] [Citace: 16. leden 2014.] <http://www.gme.cz/img/cache/doc/532/089/p-re24-datasheet-1.pdf>.
26. **Analog Devices.** Low Voltage Temperature Sensors. *Analog Devices*. [Online] [Citace: 23. duben 2014.] http://www.analog.com/static/imported-files/data_sheets/TMP35_36_37.pdf.
27. **Maxim Integrated.** DS18B20 Programmable Resolution 1-Wire Digital Thermometer. *Maxim Integrated*. [Online] 2006. [Citace: 10. únor 2014.] <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>.
28. —. DS1307 64 x 8, Serial, I2C Real-Time Clock. *Maxim Integrated*. [Online] 2008. [Citace: 2. duben 2014.] <http://datasheets.maximintegrated.com/en/ds/DS1307.pdf>.
29. **Pololu Robotics & Electronics.** MinIMU-9 v2 Gyro, Accelerometer, and Compass. *Pololu Robotics & Electronics*. [Online] [Citace: 18. únor 2014.] <http://www.pololu.com/product/1268>.
30. —. Pololu MinIMU-9 + Arduino AHRS (Attitude and Heading Reference System). *GitHub*. [Online] 27. listopad 2013. [Citace: 3. prosinec 2014.] <https://github.com/pololu/minimu-9-ahrs-arduino>.

31. **Wikipedia.** Flight dynamics. *Wikipedia, the free encyclopedia*. [Online] 9. září 2013. [Citace: 20. březen 2014.] http://en.wikipedia.org/wiki/Flight_dynamics.

32. **National Instruments.** NI USB-8452 I2C and SPI Interface. *National Instruments*. [Online] [Citace: 30. březen 2014.] <http://sine.ni.com/nips/cds/view/p/lang/cs/nid/210256>.

11 Seznam příloh

11.1 Tištěné přílohy

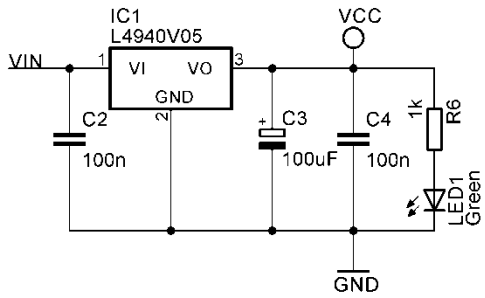
- I. Schéma zapojení přípravku s motory
- II. Návrh desky plošných spojů přípravku s motory
- III. Seznam součástek přípravku s motory
- IV. Schéma zapojení přípravku s mikrokontrolérem
- V. Návrh desky plošných spojů přípravku s mikrokontrolérem
- VI. Seznam součástek přípravku s mikrokontrolérem
- VII. Schéma přípravku se senzory
- VIII. Návrh desky plošných spojů přípravku se senzory
- IX. Seznam součástek přípravku se senzory

11.2 Elektronické přílohy (obsah CD)

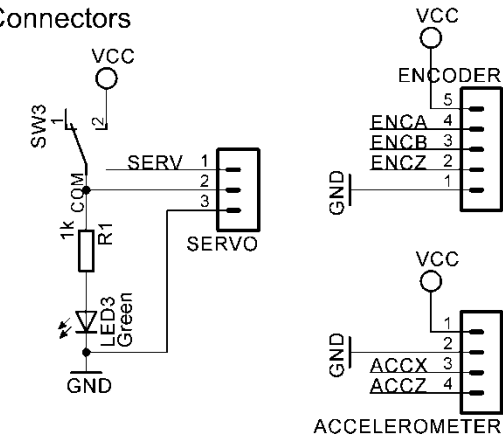
| | Název | Komentář |
|----------------|---|--|
| Adresář | Arduino/Přípravek_s_mikrokontrolerem | Obsahuje kompletní program pro přípravek |
| Soubory | Atmega_piny.h | Přehled vstupů/výstupů ATmega644P |
| | Cislicovy_osc.ino | Číslicový oscilátor |
| | Cislicovy_osc_USB.ino | Číslicový oscilátor řízený přes USB rozhraní |
| | Digitalni_teplomer | Digitální teploměr (DS18B20) |
| | Gnerátor_dig_signalu.ino | Generátor digitálních signálů |
| | I2C.ino | Úloha I2C slave |
| | IMU.ino | Komunikace a vyhodnocení dat z IMU |
| | Menu.ino | Menu přípravku (výběr úlohy) |
| | Napetim_riz_osc.ino | Napětím řízený oscilátor |
| | Pripravek_s_mikrokontrolerem.ino | Hlavní část programu |
| | Spinac_hystereze.ino | Spínač s hysterezí |
| Adresář | DPS | Schémata a podklady pro výrobu DPS |
| Soubory | Pripravek_s_mikrokontrolerem.brd | DPS |
| | Pripravek_s_mikrokontrolerem.sch | Schéma |
| | Pripravek_s_motory.brd | DPS |
| | Pripravek_s_motory.sch | Schéma |
| | Pripravek_se_senzory.brd | DPS |
| | Pripravek_se_senzory.sch | Schéma |
| | Seznamy_soucastek.xlsx | Seznamy součástek pro všechny přípravky |
| Adresář | LabVIEW/I2C | Vzorové VI pro práci se sběrnici I2C |
| Soubory | I2C_read.vi | Čtení dat ze sběrnice |
| | I2C_settings.vi | Nastavení sběrnice |
| | I2C_write.vi | Zápis dat na sběrnici |
| Adresář | LabVIEW/Přípravek_s_mikrokontrolerem | Vzorové VI pro přípravek s mikrokontrolérem |
| Soubory | Cislicovy_osc.vi | Číslicový oscilátor |

| | | |
|----------------|-------------------------------------|--|
| | Cislicovy_osc_USB.vi | Číslicový oscilátor řízený přes USB rozhraní |
| | Generator_dig_signalu.vi | Generátor digitálních signálů |
| | I2C_slave.vi | Úloha I2C slave |
| | Napetim_rizeny_oscilator.vi | Napětím řízený oscilátor |
| | Spinac_s_hysterezi.vi | Spínač s hysterezí |
| Adresář | LabVIEW/Pripravek_s_motory | Vzorové VI pro přípravek s motory |
| Soubory | Acc_vypocet_uhlu.vi | Výpočet úhlu náklonu z napětí akcelerometru |
| | KrokovyM_inkremental.vi | Úloha s krokovým motorem a ink. snímačem |
| | Servo_akcelerometr.vi | Úloha se servomotorem a akcelerometrem |
| | Vypocet_servopulzu.vi | Výpočet délky impulzu pro servomotor |
| Adresář | LabVIEW/Pripravek_se_senzory | Vzorové VI pro přípravek se senzory |
| Soubory | BCD_byte_to_DEC.vi | Převod bajtu v BCD kódu na dekadické číslo |
| | Digitalni_teplomer.vi | Digitální teploměr (DS18B20) |
| | IMU.vi | Zobrazení dat z IMU |
| | RTC.vi | Hodiny reálného času - čtení pomocí I2C |
| | Senzory_spolecne.vi | Společné demonstrační VI pro senzory |

Power Supply



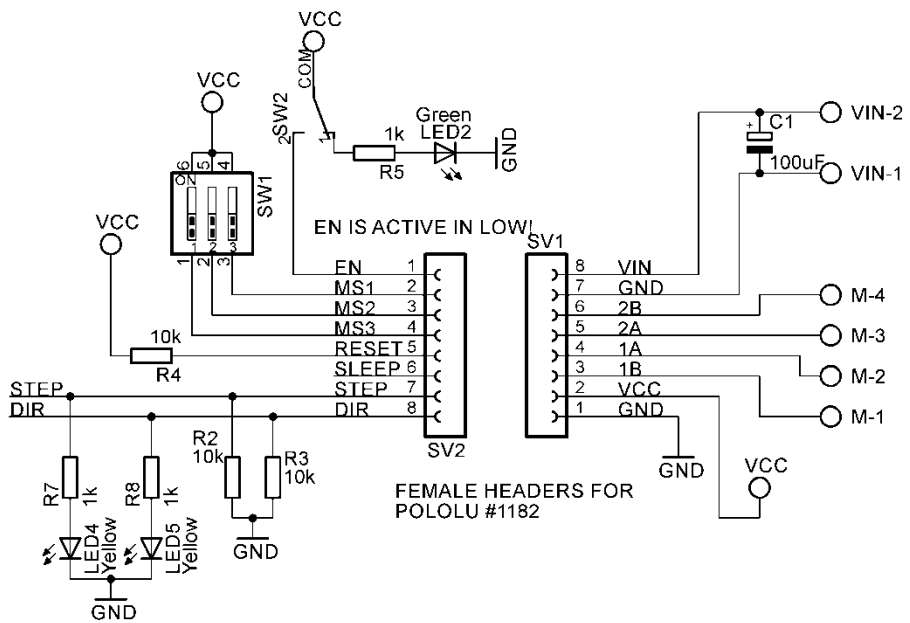
Connectors



Terminal

- X1-1 ○ ENCA
- X1-2 ○ ENCB
- X1-3 ○ ENCZ
- X1-4 ○ SERV
- X1-5 ○ ACCX
- X1-6 ○ ACCZ
- X1-7 ○ STEP
- X1-8 ○ DIR
- X1-9 ○ GND

Stepper Driver



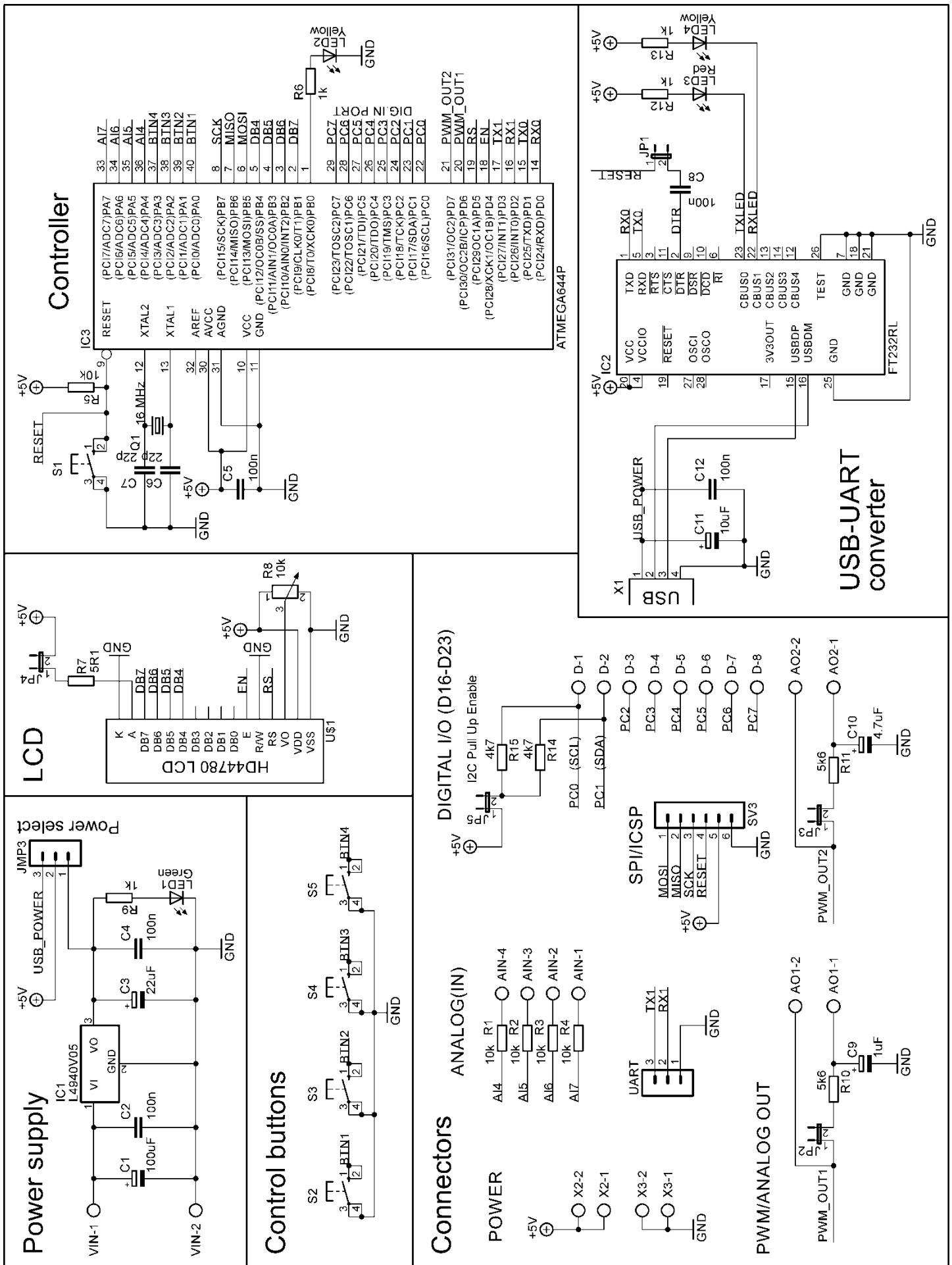
Seznam součástek - přípravek s motory

| Označení | Součástka | Hodnota | Kód | Prodejce | Počet |
|----------------|-------------------------------------|---------------|------------|----------|-------|
| - | Chladič V4330N pro stabilizátor | - | 620-008 | GME | 1 |
| - | Počítačový konektor, rozteč 2,54 mm | 3 piny | 840-003 | GME | 1 |
| - | Počítačový konektor, rozteč 2,54 mm | 4 piny | 840-005 | GME | 1 |
| - | Počítačový konektor, rozteč 2,54 mm | 5 pinů | 840-011 | GME | 1 |
| - | Počítačový konektor, rozteč 2,54 mm | 6 pinů | 840-006 | GME | 1 |
| - | Pin do konektoru | - | 840-004 | GME | 20 |
| - | Fotocuprexit 75x100 mm | - | 661-073 | GME | 1 |
| SW1 | DIP spínač do DPS | 3 póly | 632-039 | GME | 1 |
| SV1,SV2 | Dutinková lišta pro budič KM | 10 pinů | 832-041 | GME | 2 |
| X1 | Svorkovnice PTR AK500/3 | - | 821-008 | GME | 3 |
| M,VIN | Svorkovnice PTR AK500/2 | - | 821-017 | GME | 3 |
| R2,R3,R4 | Rezistor 0204 | 10k | 119-396 | GME | 3 |
| R1,R5,R6,R7,R8 | Rezistor 0204 | 1k | 119-397 | GME | 5 |
| LED4,LED5 | LED 3 mm | žlutá | 511-202 | GME | 2 |
| LED1,LED2,LED3 | LED 3 mm | zelená | 511-051 | GME | 3 |
| IC1 | Stabilizátor L4940V05 | - | 330-067 | GME | 1 |
| C2,C4 | Keramický kondenzátor | 100nF/63V | 120-060 | GME | 2 |
| C1,C2 | Elektrolyt. kondenzátor (low ESR) | 100μF/25V | 123-442 | GME | 2 |
| - | Oboustranný kolík přímý (lámací) | 20 pinů | 832-017 | GME | 1 |
| SW2,SW3 | Posuvný spínač do DPS | 1 pól | 631-382 | GME | 2 |
| - | Budič krokového motoru s IO A4988 | - | pololu1182 | SNAIL | 1 |
| - | Servomotor Vigor-VS2 | - | - | SNAIL | 1 |
| - | Akcelerometr | - | Pololu1251 | SNAIL | 1 |
| - | Krokový motor (NEMA17) | - | Pololu1200 | SNAIL | 1 |
| - | Pružná spojka | - | - | CNCS | 1 |
| - | Montážní úhelník pro NEMA17 | - | ada1297 | SNAIL | 2 |
| - | Inkrementální snímač S3806 | 1000 pulzů/ot | - | CNCS | 1 |

GME - GM electronic - www.gme.cz

SNAIL - Snail Instruments - www.snailshop.cz

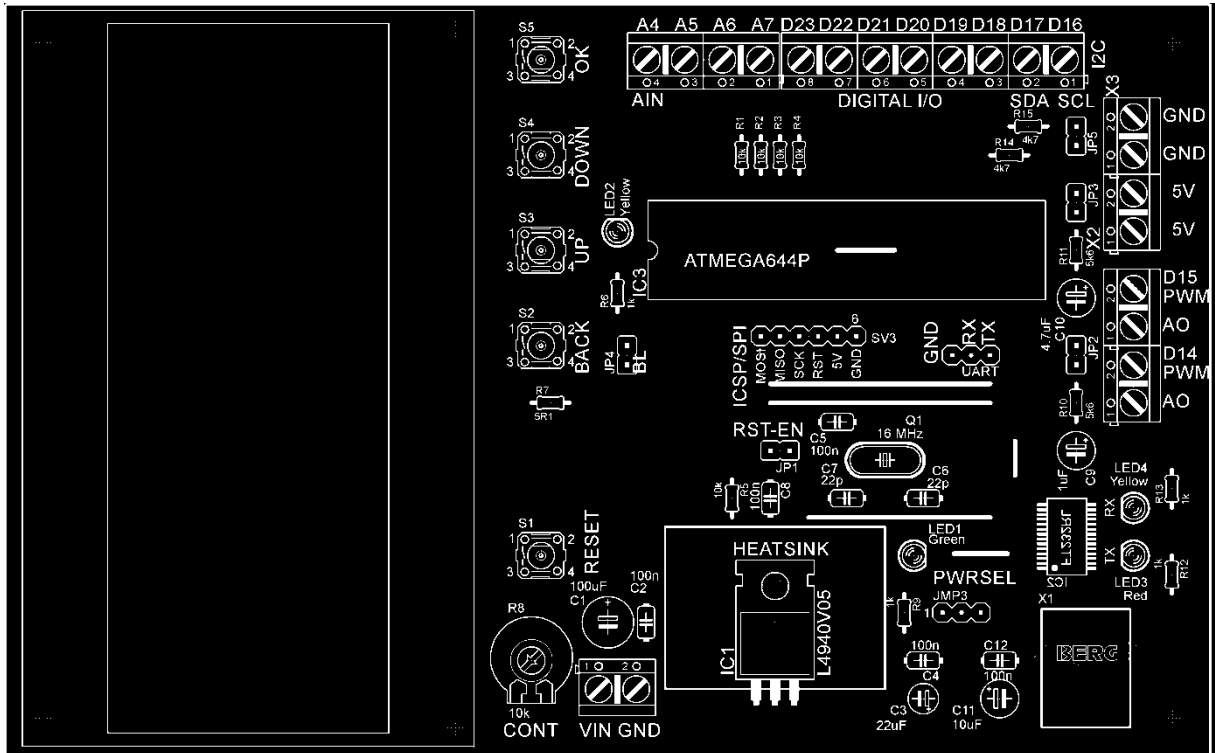
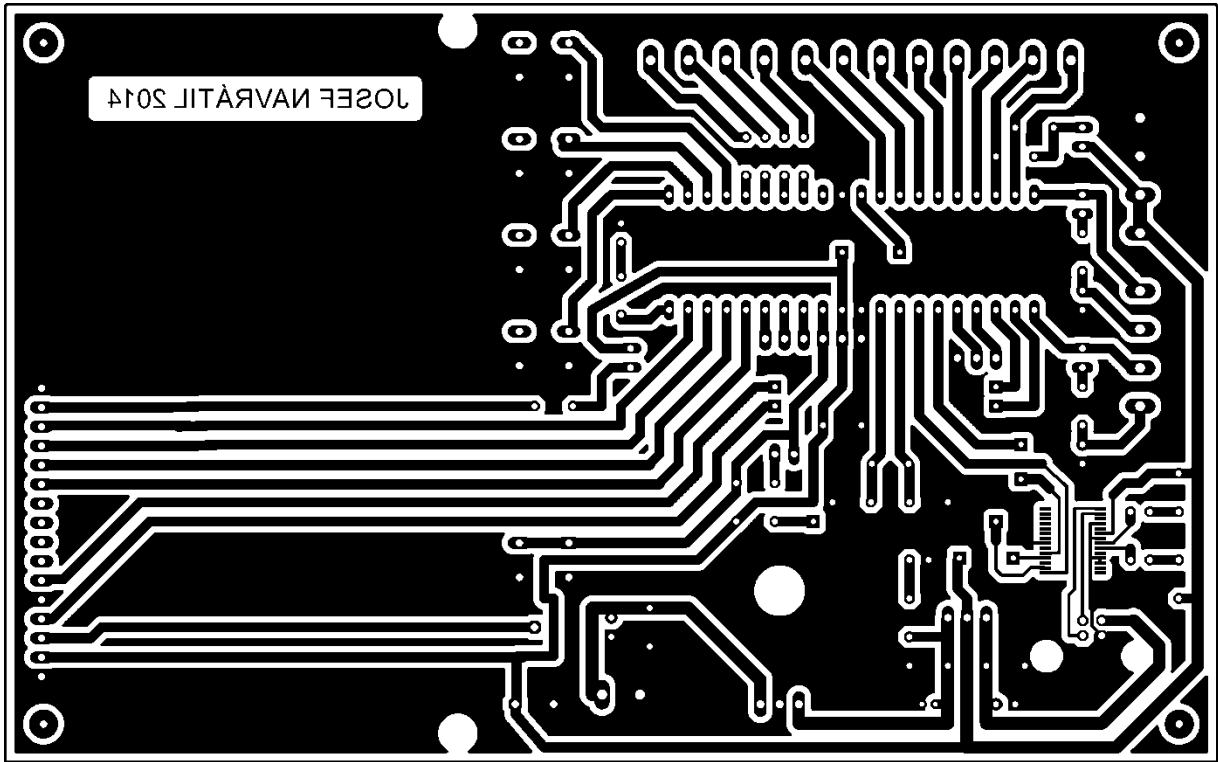
CNCS -CNC shop - www.cncshop.cz



Controller

| | | | |
|----|----------|----------------------|------------------|
| 33 | AI7 | (PC17/ADC7)PA7 | RESET |
| 34 | AI6 | (PC16/ADC6)PA6 | XTAL2 |
| 35 | AI5 | (PC15/ADC5)PA5 | XTAL1 |
| 36 | AI4 | (PC14/ADC4)PA4 | AREF |
| 37 | AIN4 | (PC13/ADC3)PA3 | AVCC |
| 38 | AIN3 | (PC12/ADC2)PA2 | AGND |
| 39 | AIN2 | (PC11/ADC1)PA1 | VCC |
| 40 | AIN1 | (PC10/ADC0)PA0 | GND |
| 8 | SCK | (PC115/SCK)PB7 | GND |
| 7 | MISO | (PC114/MISO)PB6 | (PC110/AIN0)PB2 |
| 6 | MOSI | (PC113/MOSI)PB5 | (PC109/CLK0)PB1 |
| 5 | DB4 | (PC112/OC0B)PB4 | (PC108/TXC0)PB0 |
| 4 | DB5 | (PC111/AIN1/OC0A)PB3 | (PC123/TOSC2)PC7 |
| 3 | DB6 | (PC110/AIN0)PB2 | (PC122/TOSC1)PC6 |
| 2 | DB7 | (PC109/CLK0)PB1 | (PC121/TDI)PC5 |
| 1 | | (PC108/TXC0)PB0 | (PC120/TDO)PC4 |
| 29 | PC7 | | (PC119/TMS)PC3 |
| 28 | PC6 | | (PC118/TCK)PC2 |
| 27 | PC5 | | (PC117/SDA)PC1 |
| 26 | PC4 | | (PC116/SCL)PC0 |
| 25 | PC3 | | |
| 24 | PC2 | | |
| 23 | PC1 | | |
| 22 | PC0 | | |
| 21 | PWM_OUT2 | (PC131/OC2)PD7 | |
| 20 | PWM_OUT1 | (PC130/OC2B/ICP)PD6 | |
| 19 | RS | (PC129/OC1A)PD5 | |
| 18 | EN | (PC128/XCK1/OC1B)PD4 | |
| 17 | TX1 | (PC127/INT1)PD3 | |
| 16 | RX1 | (PC126/INT0)PD2 | |
| 15 | TX0 | (PC125/TXD)PD1 | |
| 14 | RX0 | (PC124/RXD)PD0 | |

ATMEGA644P

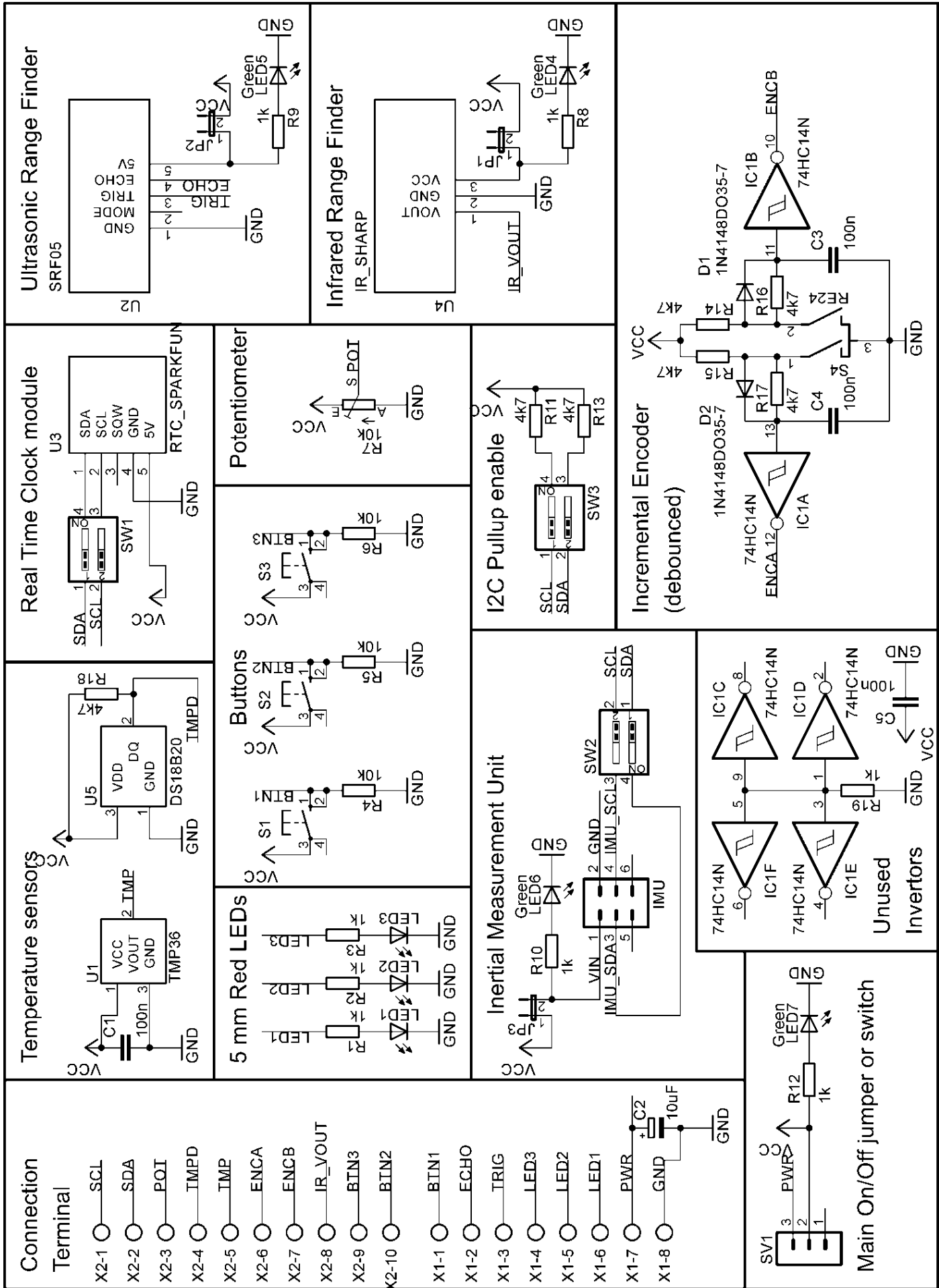


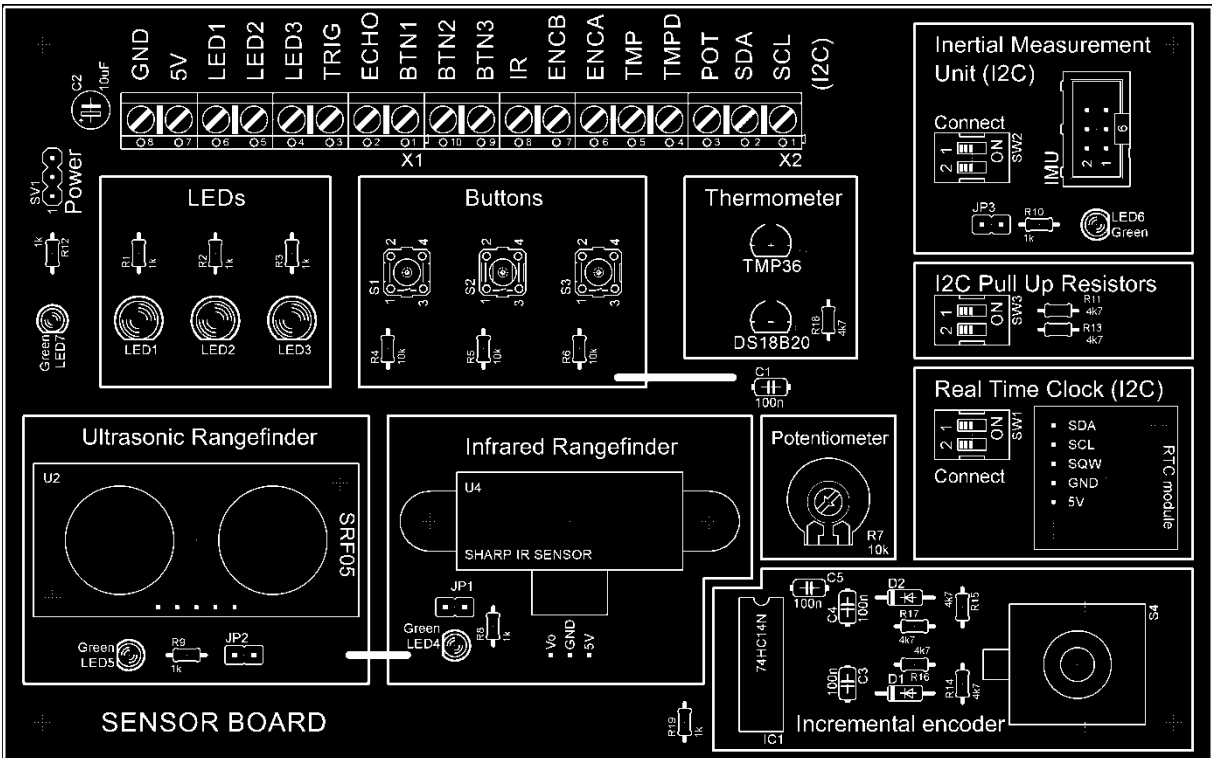
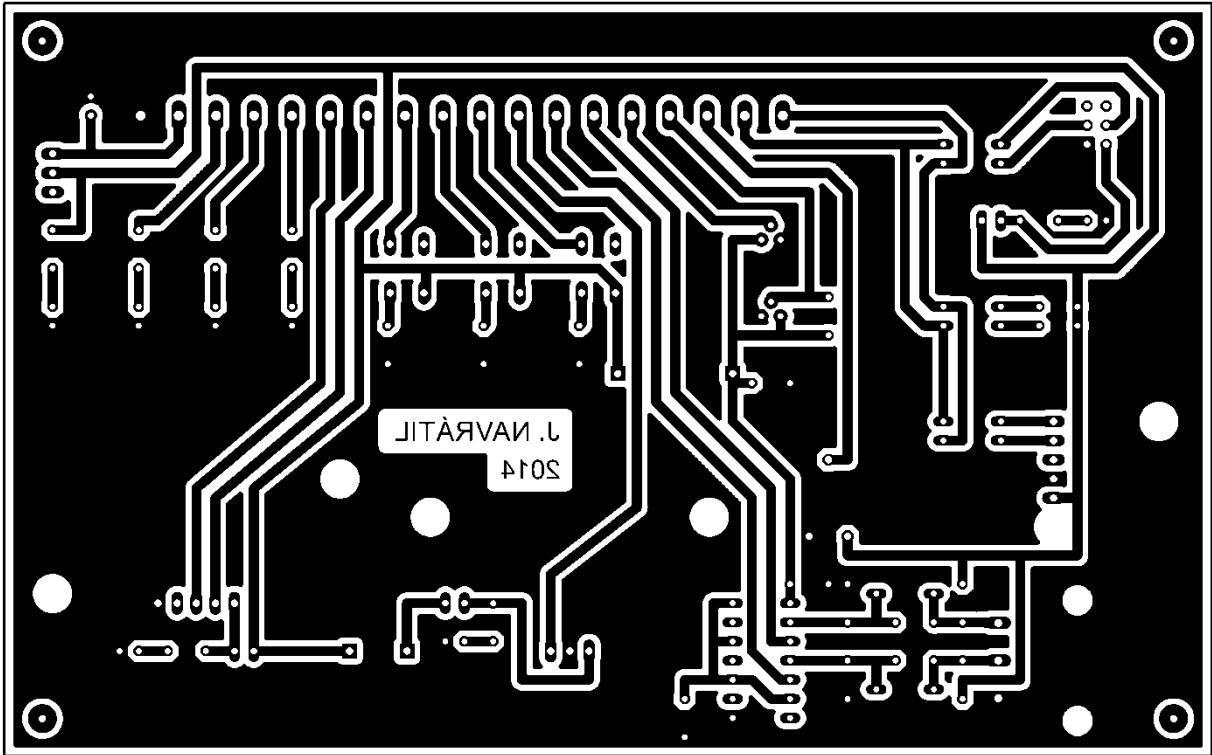
Seznam součástek - přípravek s mikrokontrolérem

| Označení | Součástka | Hodnota | Kód | Prodejce | Počet |
|-----------------|----------------------------------|-----------|----------|----------|-------|
| R8 | Odporový trimr PT10VK010 | 10k | 112-049 | GME | 1 |
| R7 | Rezistor 0204 | 5R1 | 119-257 | GME | 1 |
| R1,R2,R3,R4,R5 | Rezistor 0204 | 10k | 119-396 | GME | 5 |
| R6,R9,R12,R13 | Rezistor 0204 | 1k | 119-397 | GME | 4 |
| R14,R15 | Rezistor 0204 | 4k7 | 119-395 | GME | 2 |
| R10,R11 | Rezistor 0204 | 5k6 | 119-055 | GME | 2 |
| C2,C4,C5,C8,C12 | Keramický kondenzátor | 100nF/63V | 120-060 | GME | 5 |
| C6,C7 | Keramický kondenzátor | 22pF/50V | 120-125 | GME | 2 |
| C3 | Elektrolytický kondenzátor | 22μF/10V | 123-217 | GME | 1 |
| C9 | Elektrolytický kondenzátor | 1μF/63V | 123-453 | GME | 1 |
| C10 | Elektrolytický kondenzátor | 4,7μF/50V | 123-041 | GME | 1 |
| C11 | Elektrolytický kondenzátor | 10μF/63V | 123-483 | GME | 1 |
| C1 | Elektrolytický kondenzátor | 100μF/25V | 123-649 | GME | 1 |
| Q1 | Krystal HC49US | 16 MHz | 131-075 | GME | 1 |
| IC1 | Stabilizátor L4940V05 | - | 330-067 | GME | 1 |
| LED1 | LED 3 mm | Zelená | 511-051 | GME | 1 |
| LED2,LED4 | LED 3 mm | Žlutá | 511-202 | GME | 2 |
| LED3 | LED 3 mm | Červená | 511-201 | GME | 1 |
| - | Chladič V4330N pro stabilizátor | - | 620-008 | GME | 1 |
| S1,S2,S3,S4,S5 | Tlačítko TC-0103-T | - | 630-305 | GME | 5 |
| - | Fotocuprexit 160x100 mm | - | 661-117 | GME | 1 |
| - | Svorkovnice ARK500/2 | - | 821-017 | GME | 11 |
| - | Jumper | - | 832-013 | GME | 6 |
| - | Oboustranný kolík přímý (lámací) | 40 pinů | 832-021 | GME | 1 |
| X1 | USB B konektor do DPS | - | 832-120 | GME | 1 |
| - | Patice DIL40PZ | 40 pinů | 824-011 | GME | 1 |
| - | LCD displej MC2004B-SYL/H | - | 513-125 | GME | 1 |
| IC2 | Převodník USB-UART FT232RL | - | 959-303 | GME | 1 |
| IC3 | Mikrokontrolér ATmega644P-20PU | - | 1140-140 | PSE | 1 |

GME - GM electronic - www.gme.cz

PSE - PS electronic - www.pselectronic.cz





Seznam součástek - přípravek se senzory

| Označení | Součástka | Hodnota | Kód | Prodejce | Počet |
|--------------------------------|-------------------------------------|-----------|------------|----------|-------|
| R11,R13,R14,R15,R16,R17 R18 | Rezistor 0204 | 4k7 | 119-395 | GME | 7 |
| R4,R5,R6 | Rezistor 0204 | 10k | 119-396 | GME | 3 |
| R1,R2,R3,R8,R9, R10,R12,R19 | Rezistor 0204 | 1k | 119-397 | GME | 8 |
| C1,C3,C4,C5 | Keramický kondenzátor | 100nF/63V | 120-060 | GME | 4 |
| C2 | Elektrolytický kondenzátor | 10µF/63V | 123-483 | GME | 1 |
| D1,D2 | Dioda 1N4148 | - | 220-003 | GME | 2 |
| IC1 | 6x Schmitt. invertor, DIP14 | - | 425-009 | GME | 1 |
| LED4,LED5,LED6, LED7 | LED 3 mm | Zelená | 511-051 | GME | 4 |
| LED1,LED2,LED3 | LED 5 mm | Červená | 511-325 | GME | 3 |
| U1 | TMP36GT9Z | - | 530-093 | GME | 1 |
| S4 | Mechanický inkrem. snímač P-RE24 | - | 532-089 | GME | 1 |
| S1,S2,S3 | Tlačítko TC-0103-T | - | 630-305 | GME | 3 |
| SV1 | Přepínač P-B143-1 | - | 631-354 | GME | 1 |
| SW1,SW2,SW3 | DIP spínač do DPS | 2 póly | 632-002 | GME | 3 |
| - | Fotocuprexit 160x100 mm | - | 661-117 | GME | 1 |
| - | konektor pro plochý kabel | 6 pinů | 800-081 | GME | 1 |
| IMU | konektor do DPS | 6 pinů | 800-257 | GME | 1 |
| - | Krabička U-KP32 (pro IMU) | - | 622-537 | GME | 1 |
| - | Plochý kabel (28AWG) | 10 žil | 650-043 | GME | 1 |
| - | Svorkovnice PTR AK500/3 | - | 821-008 | GME | 6 |
| IC1 | Patice DIL14PZ | 14 pinů | 824-003 | GME | 1 |
| - | Jumper | - | 832-013 | GME | 4 |
| - | Oboustranný kolík přímý (lámací) | 20 pinů | 832-017 | GME | 1 |
| U2,U3 | Dutinková lišta | 5 pinů | 832-040 | GME | 2 |
| R7 | Rezistorový trimr ležatý | 10k | 112-360 | GME | 1 |
| - | Osa pro trimr | - | 624-163 | GME | 1 |
| U5 | Digitální teploměr DS18B20 | - | 530-067 | GME | 1 |
| U2 | Ultrazvukový dálkoměr SRF05 | - | 115-002 | SNAIL | 1 |
| U4 | IR dálkoměr GP2Y0A41 | - | - | SNAIL | 1 |
| U3 | Hodiny reálného času | - | SP70 | HWK | 1 |
| - | MinIMU-9 v2 | - | pololu1268 | SNAIL | 1 |

GME - GM electronic - www.gme.cz

SNAIL - Snail Instruments - www.snailshop.cz

HWK - HW Kitchen - www.hwkitchen.com

