ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA ELEKTROTECHNICKÁ

KATEDRA ŘÍDICÍ TECHNIKY

BAKALÁŘSKÁ PRÁCE

Moderní pracoviště pro výuku předmětu A3B35APE - Aplikovaná elektronika

Modern laboratory task for education subject A3B35APE - Applied electronics

autor: **Martin Makovička** **2010**

vedoucí práce: **Ing. Martin Hlinovský Ph.D.**

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Martin Makovička**

Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný
Obor: Kybernetika a měření

Název tématu: **Moderní pracoviště pro výuku předmětu A3B35APE - Aplikovaná elektronika**

Pokyny pro vypracování:

1. Seznamte se s mikrokontroléry založené na architektuře ARM (konkrétní typ bude specifikován vedoucím bakalářské práce).
2. Analyzujte současnou součástkovou základnu.
3. Po dohodě s vedoucím práce navrhněte a realizujte integrované laboratorní pracoviště pro výuku předmětu A3B35APE – Aplikovaná elektronika.
4. Vypracujte návod k dané úloze ve formě webové prezentace.

Seznam odborné literatury:

Dodá vedoucí práce

Vedoucí: Ing. Martin Hlinovský, Ph.D.

Platnost zadání: do konce zimního semestru 2010/2011

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

L.S.

doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 4. 11. 2009

Abstrakt

Předkládaná bakalářská práce se zabývá návrhem laboratorního pracoviště, na kterém bude probíhat výuka zaměřená na návrh a konstrukci elektronických obvodů používaných zejména v oblasti robotiky a řídicí techniky. Pro pracoviště byla zvolena forma mobilního robota, na kterém lze prakticky předvádět využití sestavených obvodů. Po analýze různých možných řešení byla využita koncepce NXT Intelligent Brick ze stavebnice Lego mindstorms a dále byly využity volně šiřitelné prostředky pro práci s tímto modulem. V rámci práce byl navržen prototyp procesorového modulu pracoviště, který je odvozen od NXT Intelligent Brick a přizpůsoben potřebám laboratorního pracoviště. Od prototypu je očekáváno ověření použitelnosti tohoto konceptu pro výuku, zdokonalení jeho mechanické konstrukce a jeho použití k návrhu úloh dle jeho možností. Vyzkoušení práce s prototypem však z časových důvodů není součástí této práce.

Klíčová slova

aplikovaná elektronika, architektura ARM, laboratorní pracoviště, Lego Mindstorms, NXT

Abstract

The bachelor thesis deals with the laboratory workplace which is going to be aimed for teaching design and construction of electronic circuits be used particularly in a field of robotics and control technology. A form of mobile robot was chosen for the workplace which can demonstrate practical use of assembled circuits. After analyzing various possible solutions there have been used the concept of NXT Intelligent Brick from Lego mindstorms and freeware tools for working with this module. During the thesis, a prototype of processor module was designed for the workplace which is derived from NXT Intelligent Brick and adapted to the needs of the laboratory workplace. The prototype is supposed to verify the usability of this concept for teaching, improving the mechanical design and its usage to design exercises according to that potential. Testing the prototype is not part of this thesis because of press of time.

Key words

Applied electronics, architecture ARM, laboratory workspace, Lego mindstorms, NXT

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne 28.5.2010 Martin Makovička

……..

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce panu Ing. Martinu Hlinovskému za vznášení připomínek a nápadů zvláště během návrhu řešení pracoviště.

1 ÚVOD 10

1.1 POŽADAVKY NA PRACOVIŠTĚ 10

1.2 POŽADAVKY NA ÚLOHY 12

1.3 OSOBNÍ VIZE 13

2 ANALÝZA MOŽNOSTÍ ŘEŠENÍ 15

2.1 NÁVRH KONCEPCE PŘÍPRAVKU 16

2.2 BLOKOVÉ SCHÉMA 18

2.3 VÝBĚR POHONU 19

2.3.1 Dimenzování motoru 19

2.3.2 Odhad hmotnosti přípravku 20

2.3.3 Výběr vhodného motoru 21

2.4 MOŽNOST POUŽITÍ HOTOVÉHO MODULU 23

3 NÁVRHY ÚLOH 25

3.1 OPTICKÉ SENZORY 25

3.1.1 Barevný senzor 25

3.1.2 Senzor pro sledování čáry 26

3.2 ÚLOHA I2C SBĚRNICI 28

4 POPIS MINDSTORMS NXT 29

4.1 NXT INTELLIGENT BRICK 29

4.2 PERIFERIE PRO NXT 31

4.3 SOFTWARE K NXT 32

5 POPIS A NÁVRH OBVODŮ PRACOVIŠTĚ 35

5.1 BLOKOVÉ SCHÉMA 35

5.2 PROCESORY A JEJICH PERIFERIE 36

5.2.1 Zapojení procesoru ARM 36

5.2.2 Zapojení procesoru AVR 37

5.2.3 Audio výstup 38

5.2.4 Zapojení Bluetooth konektoru 40

5.3 NAPÁJENÍ 41
5.3.1 Přívod napájení 41
5.3.2 Zdroj napájení 5V 42
5.3.3 Spínání napájení a zdroj 3,3V 43
5.3.4 Měření napětí baterie 43
5.3.5 Proudová pojistka 45
5.4 MOTORY 45
5.4.1 Ovládání motorů 45
5.4.2 Zpracování signálů z inkrementálních senzorů 46
5.5 VSTUPNĚ VÝSTUPNÍ OBVODY 47
5.5.1 Analogové vstupy 47
5.5.2 Digitální vstup 48
5.5.3 RS485 49
5.5.4 Sběrnice I2C 50
5.5.5 Přidané vstupně výstupní porty 51
5.6 KLÁVESNICE A LCD DISPLEJ 52
5.6.1 Klávesnice 52
5.6.2 LCD displej 54
5.7 KONEKTORY 55
6 NÁVRH PROTOTYPU 57
6.1 UCHYCENÍ MOTORŮ 57
6.2 VOLBA KOL 58
6.3 HLAVNÍ KONSTRUKCE PROTOTYPU 59
6.4 VOLBA SOUČÁSTEK PROTOTYPU 60
6.5 NÁVRH PLOŠNÉHO SPOJE 61
7 POPIS SOFTWARU 62
7.1 SOFTWARE PROCESORU ARM 62
7.1.1 Vývojové prostředí 62
7.1.2 Struktura projektu 63
7.1.3 Úprava ovládání displeje 64

7.1.4 Plánované úpravy *65*
7.2 SOFTWARE PROCESORU AVR 66
7.2.1 Vývojové prostředí *66*
7.2.2 Struktura zdrojových souborů *67*
7.2.3 Provedené úpravy *67*
7.2.4 Plánované úpravy *67*
7.3 BRICX COMMAND CENTER 68
7.3.1 Instalace *68*
7.3.2 Používání *68*
8 ZÁVĚR 70
POUŽITÁ LITERATURA: 71
ELEKTRONICKÉ PŘÍLOHY: 73
PŘÍLOHY: 74

1 Úvod

Tato práce se zabývá návrhem a stavbou pracoviště pro výuku předmětu aplikovaná elektronika (A3B35APE), jehož výuka začne v letním semestru školního roku 2011/2012. Tento předmět bude zaměřen na návrh elektronických obvodů používaných v oblasti řídicí techniky a robotiky, zvláště pak na jejich praktickou aplikaci. Podrobnější informace o předmětu lze získat ze studijního plánu oboru Kybernetika robotika (1)[1], jiné webové stránky s informacemi o tomto předmětu v současné době neexistují, ale je plánována jejich tvorba. Pracoviště by mělo umožňovat připojení a otestování řady úloh zaměřených na konstrukci běžně využívaných elektronických obvodů. Kromě připojení úlohy by pracoviště mělo podat studentovi informace o úloze, po připojení ověřit funkci obvodu, obvod prakticky předvést a nakonec instruovat studenta v průběhu měření parametrů obvodu.

Mojí motivací pro tuto práci je především rozšíření znalostí ohledně mikroprocesorů s architekturou ARM, se kterými jsem se zatím moc nesetkal, a to jak po stránce hardwaru, tak zejména po stránce softwaru. Další motivací je vytvoření přípravku, který bude používán při výuce a zpříjemní studium mnoha studentům a zároveň pomůže vyučujícím.

Výsledkem této práce by v ideálním případě měl být kompletně navržený modul včetně dokumentace a návodu k obsluze. Vzhledem k rozsahu práce a časovým možnostem lze očekávat, že modul nebude zcela dokončen. Hlavním cílem tedy je vypracování návrhu modulu, včetně popisu možností jeho využití a v rámci možností i jeho realizace. Práce by měla být doplněna o kompletní dokumentaci, tak aby mohlo být pracoviště dále udržováno a rozvíjeno.

1.1 Požadavky na pracoviště

Během prvních schůzek s vedoucím práce, a současně zadavatelem úkolu, panem Ing. Hlinovským jsme se dohodli na základních požadavcích na pracoviště a na jeho fyzickou podobu. Požadavky, které se pokusím sepsat v této kapitole, však byly stanoveny s tím, že nejsou konečné a budou v závislosti na nových nápadech a skutečnostech upravovány.

[1] V době psaní této práce je popis předmětu pouze předběžný a bude upravován v rámci přípravy výuky, pravděpodobně bude ovlivněn i výsledky této bakalářské práce.

Jádrem pracoviště by měl být procesorový modul s procesorem ARM, který bude obsluhovat připojitelné moduly. Na samotný procesor zatím nejsou žádné zvláštní požadavky, musí ale umožnit splnění ostatních požadavků.

Procesorový modul by měl mít alespoň čtyři vstupně výstupní porty, ty by měly být v maximální možné míře univerzální tak, aby umožňovaly připojení různých druhů modulů, v případě složitější obsluhy modulu lze počítat s využitím více portů pro jeden modul. Pro snadnou použitelnost by porty měly být zaměnitelné, některým portům lze přidat speciální funkce, ale ty by neměly výrazně omezit kompatibilitu. V případě potřeby je možné vytvořit specifický port, ale takových portů by mělo být minimum a nelze je počítat do minimálního počtu portů.

Dále byl zdůrazněn požadavek na využití některých z běžně používaných sériových rozhraní tak, aby bylo umožněno na nich měřit. Zejména byla zdůrazněna sběrnice I2C, která je hojně používaná na různých modulech, zejména na senzorech v robotice. Pro sériové rozhraní je prozatím plánován samostatný konektor, pro I2C by bylo vhodné umožnit připojení více zařízení najednou, aby mohla být využita jeho adresovatelnost, zároveň se tak usnadní případná rozšiřitelnost pracoviště pro případ potřeby vytvořit rozsáhlejší úlohy.

Pro zvýšení atraktivity přípravku a umožnění názornějšího předvedení funkce úloh bylo rozhodnuto vybavit přípravek motory a udělat tak z něj mobilního robota. Motory budou využity zejména k předvedení senzorů, na jejichž návrhu se studenti budou podílet. Aby pro předvedení přípravku v pohybu nebylo potřeba připojit více senzorů, bylo by vhodné k motorům připojit inkrementální senzory polohy, které by zároveň mohly být pro některou úlohu. Pro řízení motorů je vhodné použít hotovou řídicí jednotku, čímž se zjednoduší návrh zařízení.

Jako uživatelské rozhraní bude sloužit LCD displej schopný zobrazit dostatek informací a k ovládání by měly být použity alespoň 4 tlačítka, lépe však 4 směrová tlačítka a jedno potvrzovací. Software musí umožňovat snadné nahrání nových programů pro případ aktualizace, nejlépe přes USB. Pro snadnost použití má přípravek umožňovat uložit více programů pro úlohy nebo jejich části, z nichž uživatel při spuštění vybere tu, kterou právě potřebuje. Nepředpokládá se, že by studenti na přípravku programovali, jako maximální zásah studenta do běhu se očekává nastavení předem připravených konstant.

Jako zajímavá alternativní možnost ovládání se jeví ovládání přípravku z PC například přes USB rozhraní. Toto je ale nutno nejdříve promyslet a během návrhu s touto možností počítat, realizace však proběhne až po odladění základní verze pracoviště. Výhodou pro ovládání by mohlo být použití některé bezdrátové technologie, pravděpodobně Bluetooth, které se pro tyto

účely hodí a k tomuto účelu je také hojně využíváno. Na Bluetooth by mohlo být předvedeno jeho možné využití a zároveň by adaptér mohl sloužit jako ukázka ovládání periferie přes sériové rozhraní. U použití bezdrátové technologie je však třeba nejdříve zvážit, zda bude přínos pro výuku dostatečný, aby převážil vyšší cenu přípravku a komplikovanější návrh, zejména na straně softwaru.

Z hlediska požadavků na fyzickou podobu zařízení jsme zatím dospěli k názoru, že v této fázi přípravy zatím nelze určit podrobnosti a na podobě zařízení se dohodneme v průběhu návrhu. Jedinými požadavky na hardware je v současné době snadné ovládání a logické rozmístění prvků tak, aby se student na přípravku rychle zorientoval a jeho používání mu nečinilo potíže.

1.2 Požadavky na úlohy

Jednotlivé úlohy by se měly skládat z modulů, z nichž některé student sám navrhne a postaví a jiné budou již hotové a sloužit tak k doplnění úlohy. Pro moduly, které bude navrhovat student, jsou ke zvážení dvě možnosti. Jednou je holý plošný spoj, který si student sám osadí a případně ho i navrhne, a druhá možnost je na připravené moduly osazovat součástky do patic a vytvořit tak pro každý modul speciální nepájivé pole. V rámci úspory nákladů na výrobu modulů je možné sjednotit moduly hotové a osazované studentem tak, že pro osazování studentem budou připraveny otvory pro vývodové součástky, kam mohou být osazeny i patice pro opakované osazení modulu, a pro hotový modul pozice pro SMD součástky, na nichž může být snadno zamaskováno jejich označení, aby bylo zabráněno opisování jejich hodnot studenty.

Jako možná úloha byla nejvíce zmiňována úloha zaměřená na sériové rozhraní I2C, kde by měli studenti za úkol ze signálu na sběrnici zjistit adresu zařízení, pro které jsou data určena a na modulu ji správně nastavit.

Další úlohy by měly být především součástí senzorického systému, zaměřeny buď na úpravu signálu senzorů, nebo přímo na návrh senzorů.

1.3 Osobní vize

V této části bych rád shrnul své původní představy o podobě a použití pracoviště po úvaze nad zadáním a nad praktickým využitím pracoviště během výuky.

Student by měl před příchodem na cvičení prostudovat problematiku, kterou se na cvičení bude zabývat. Po příchodu na pracoviště si vyzvedne materiál, který bude k práci potřebovat, zapne přípravek a spustí program pro aktuální cvičení. Pro některá cvičení může být programů i několik. Na začátku programu může být student vyzván, aby odpověděl na kontrolní otázky zobrazené na displeji pracoviště. Pokud by student v odpovědích na otázky neuspěl, pracoviště by vyslalo jasný zvukový signál a student by musel počkat na cvičícího, který provede jeho přezkoušení. V případě prokázání připravenosti na cvičení by student pokračoval v práci na úloze.

Práce na úloze lze rozdělit na několik částí, jednou z nich je příprava jednotlivých modulů, které budou později připojeny k hlavní procesorové desce. V této části by měl student pracovat samostatně podle přiložené dokumentace. Od procesorového modulu se čeká maximálně vygenerování hodnot, které se mohou každé cvičení měnit a díky nimž bude každé řešení úlohy specifické. Procesorový modul by si měl vygenerované úlohy uložit, aby podle nich mohl vyhodnocovat další průběh řešení úlohy.

Druhou částí je připojení modulů k procesorové desce. V této fázi by měl procesorový modul studenta navádět, kam má které moduly připojit. Toto navádění je navrženo proto, aby měla procesorová deska usnadněno rozpoznání a kontrolu připojených modulů. Pokud by úloha vyžadovala studentovo zamyšlení nad správným zapojením některého modulu, navádění bude vynecháno a student bude vyzván k jeho zapojení.

Dalším krokem je kontrola správného zapojení modulů procesorovou deskou, kontrola bude prováděna otestováním správné funkce modulu. Pokud by se úloha skládala z většího množství nezávislých modulů, mohou být kontrolovány i po částech. V závislosti a charakteru úlohy může procesorový modul upozornit na špatný modul, nebo výsledek testu použít pouze pro vlastní potřebu a zjištění přítomnosti chyby nechat na studentovi.

Po otestování úlohy by mělo následovat předvedení funkce celé sestavené úlohy, což by mělo motivovat studenty k úspěšnému dokončení úlohy. S předváděním úlohy mohou být spojené další úkoly, jako například nastavení některých modulů.

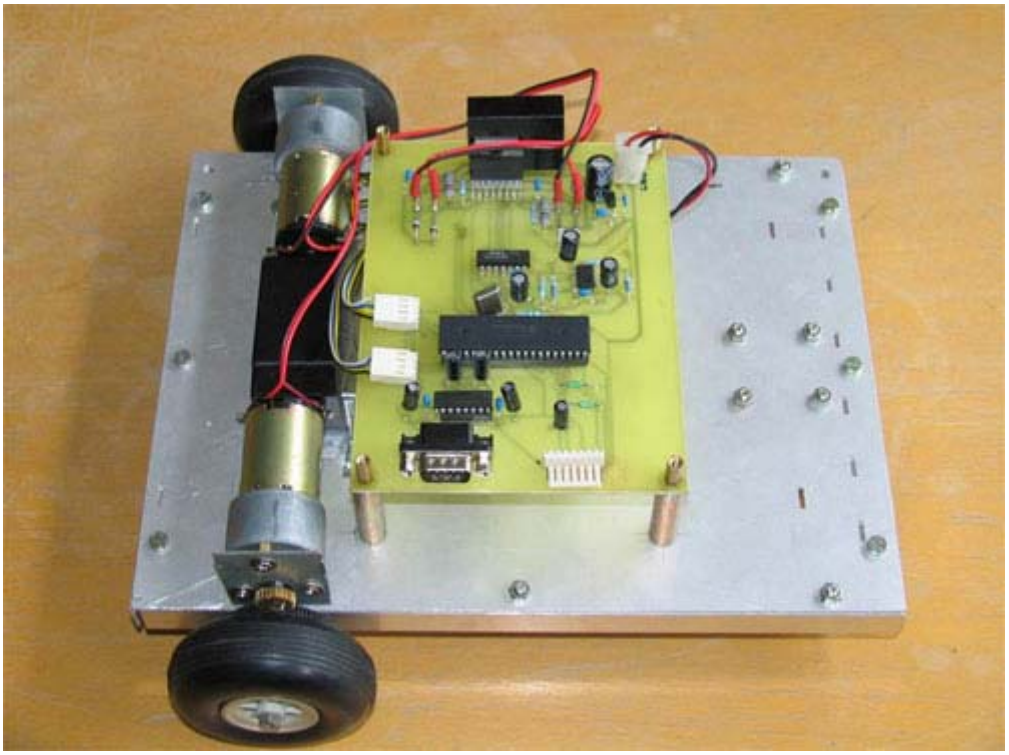
Na závěr některých úloh by mělo následovat měření na postaveném modulu nebo na celém kompletu.

Celá tato vize se snaží být co nejvíce nápomocna studentům a zároveň šetřit čas cvičícímu, který se místo kontroly studentů může věnovat vysvětlování nejasností, případně řešit vzniklé problémy. Další výhodou zamýšleného pracoviště je možnost vytvořit úlohy, jejichž parametry se mohou měnit. U úloh by se například mohlo měnit požadované zesílení zesilovače, významy pinů na konektoru a podobně. Tyto obměny mají za cíl přinutit studenta nad úlohou pokaždé přemýšlet a to i přesto, že řešení úlohy již zná od jiného studenta.

2 Analýza možností řešení

Tato kapitola shrnuje můj postup ve fázi analýzy možných řešení pracoviště na základě průzkumu dostupných součástek.

Prvním krokem při přípravě bakalářské práce jsem začal hledat nějaké mobilní roboty, ze kterých bych se mohl inspirovat a zejména získat přehled o používaných motorech, senzorech a dalších periferiích. Z nalezených zdrojů jsem asi nejvíce informací použil ze stránek (2), jde o webové stránky studenta ČVUT, který se zajímá o elektroniku a robotiku. Zde jsem získal námět na řešení optických senzorů a stránky mi také pomohly s představou o koncepci elektroniky. Dále jsem objevil dva internetové obchody v Anglii (3),(4), které mají široký sortiment, a jeden český (5), u kterého bude snazší a levnější materiál objednat. Sortiment těchto obchodů jsem v průběhu výběru vhodných komponent bral jako základ dostupného zboží, ke kterému jsem se snažil v případě potřeby najít vhodnější náhradu jinde, ale většinou jsem nic vhodnějšího nenašel.



Obr. 1 Mobilní robot pro výuku navigačních systémů a strojového vnímání vyrobený na katedře kybernetiky FAV ZČU[2]

[2] Fakulta aplikovaných věd západočeské univerzity

Během hledání zdrojů jsem zjistil, že na Fakultě aplikovaných věd Západočeské univerzity se věnovali stavbě podobného přípravku (6), přípravek je zobrazen na Obr. 1. Tento mne zaujal, neboť jeho mechanické řešení přibližně odpovídalo mé představě a díky němu jsem si svou představu zdokonalil. Oproti mobilnímu robotu ZČU by však měl mít náš přípravek rozsáhlejší elektroniku, neboť bude mít více konektorů pro připojení periferií a bude ho možno ovládat pomocí kláves z připevněného displeje. Původní přípravek slouží především k výuce prostorové orientace robotů a studenti se zaměřují i na programování samotných algoritmů řízení robotů. Náš přípravek by měl být zaměřen výhradně na návrh a konstrukci periferních bloků a s programováním by studenti v tomto předmětu neměli přijít do styku.

2.1 Návrh koncepce přípravku

Než jsem se začal poohlížet po konkrétních komponentech pro konstrukci, shrnul jsem si požadavky na přípravek, abych získal přehled, jaká rozhraní budu potřebovat od procesoru a zda bude reálné vejít se na jeden procesor. Pokud jsem předpokládal, že pro nějakou část přípravku použiji hotový modul, podíval jsem se na různé druhy modulů a zjistil, jaké rozhraní bude pravděpodobně pro tento modul využito. Cílem bylo vytvořit prvotní blokové schéma, které mi pomůže při výběru konkrétních komponent.

Přípravek by rozhodně měl obsahovat rozhraní USB, v zadání bylo toto rozhraní zmíněno pouze jako příklad, ale pro daný účel je nevhodnější. Rozhraní USB je umístěno na všech současných počítačích a řada mikroprocesorů ho využívá k nahrávání softwaru.

K ovládání přípravku by mělo sloužit 5 tlačítek, pro jejich obsluhu bude potřeba 5 vstupů, bylo by možné použít i maticové snímání kláves se 3 vstupy nebo jeden analogový vstup. Momentálně jde spíše o odhad potřebných prostředků, proto bude konečná podoba obsluhy klávesnice řešena až v závislosti na volných prostředcích vybraného procesoru. Klávesnice bude doplněna maticovým LCD displejem, který by měl zobrazit dostatek informací pro pohodlnou práci s přípravkem. Displej bude pravděpodobně 4 řádkový, což ovšem na jeho ovládací rozhraní nemá vliv, protože LCD displeje potřebné velikosti mají obvykle stejná rozhraní. Nejčastějším rozhraním pro displeje je paralelní rozhraní skládající se z 8 datových vstupně výstupních signálů a 3 řídicích signálů, většina takovýchto displejů podporuje i zjednodušenou variantu, kde jsou pouze 4 datové signály. Další možností ovládání displeje je

použití sériového rozhraní, nejčastěji pomocí SPI nebo I2C. Displeje se sériovým rozhraním se však vyskytují méně a je tedy menší výběr. Pro ovládání displeje paralelním rozhraním by tedy bylo zapotřebí 11 vstupně výstupních signálů, což je vzhledem k plánovaným periferiím a jednomu procesoru velký luxus, proto budu počítat se zjednodušeným paralelním rozhraním, na které potřebuji 7 vstupně výstupních signálů, nebo se sériovým rozhraním.

Pro řízení motorů bude pravděpodobně použit již hotový modul řízený přes I2C. Pokud by nebyl použit hotový modul, bylo by pro řízení motorů potřeba minimálně 2 výstupní piny pro PWM[3] signál a 2 vstupní piny pro dekódování signálu z otáčkového senzoru na každý motor. Pro ovládání motorů tedy bude potřeba I2C sběrnice, nebo 8 vstupně výstupních pinů.

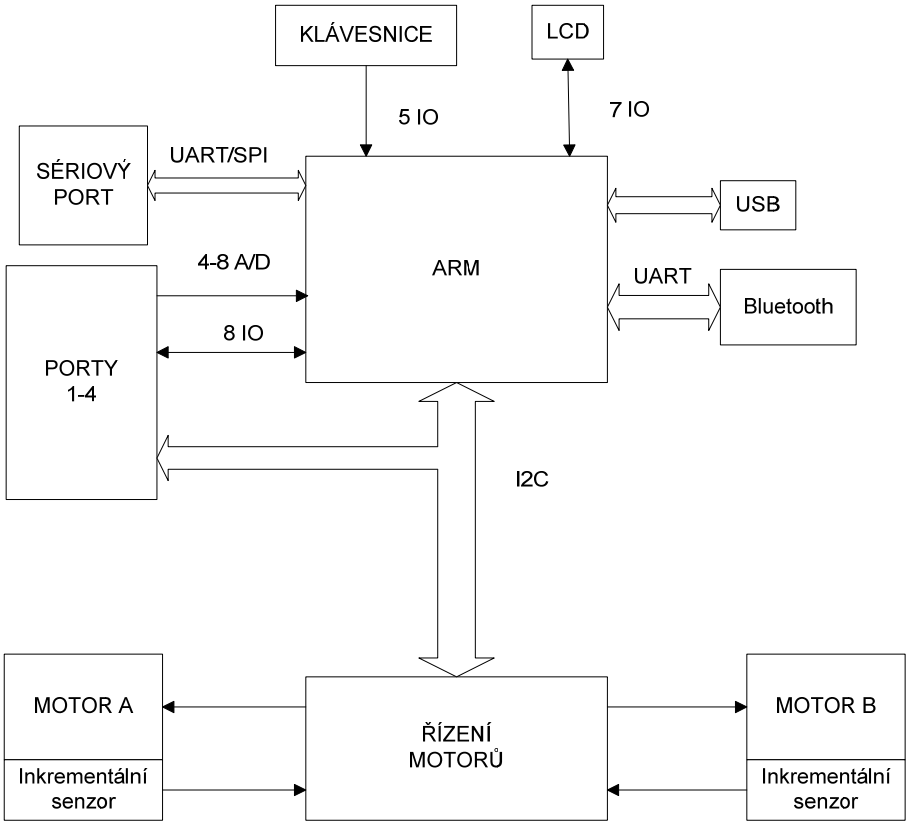
U vstupně výstupních portů jsem se nakonec, po úvahách nad možným řešením periferních modulů, rozhodl každý ze 4 univerzálních portů osadit 2 vstupně výstupními piny a jedním až dvěma analogovými vstupy. Na univerzálních portech bude také I2C sběrnice pro případné rozšíření možností modulů, zejména pro případ potřeby analogového výstupu, který na procesorech není a osazení DA převodníku na každý port by bylo nákladné, proto se zdá výhodnější, v případě potřeby, DA převodník osadit na modul. Na univerzálních portech budou rovněž umístěny piny pro napájení modulů.

Jediný snadno dostupný Bluetooth modul, který jsem našel, je stejný typ jako byl použit u jednoho robota z již zmiňovaných stránek (2). Jde o modul firmy connectBlue (7), její levnější výrobky se pohybují do 1000 Kč a všechny mají základní rozhraní UART, lepší produkty podporují i více sériových rozhraní. Pro Bluetooth modul tedy budu počítat s rozhraním UART.

[3] Pulsně šířková modulace

2.2 Blokové schéma

Ze seznamu požadavků v předchozí kapitole jsem sestavil předběžné blokové schéma, ze kterého budu vycházet pro výběr vhodných komponent. Na schématu vidím, která část vyžaduje které připojení a mohu tak snadno reagovat na změnu připojení některé periferie a případně změnit plánovaný typ procesoru. Blokové schéma je zobrazeno na Obr. 2. Zatím ještě zvažuji sériové rozhraní pro sériový port a nejraději bych I2C sběrnici rozdělil, aby bylo ovládání motorů odděleno od sběrnice přístupné studentům. Jednak by mohlo dojít k nechtěnému rušení komunikace s řídicím modulem motorů následkem špatného zapojení, což by však vzhledem k plánovanému využití motorů pouze k předvedení hotové úlohy tolik nevadilo. Vážnějším důvodem však je, že komunikace motorů s procesorem, by při úlohách, kde budou studenti sledovat data na sběrnici, mohla studentům značně komplikovat práci.



Obr. 2 Předpokládaný blokový diagram přípravku

2.3 Výběr pohonu

Jako pomocníka pro návrh pohonného systému jsem používal zejména knihu Mobilní roboty (8). Na základě informací z této knihy jsem se rozhodl, že nejlepší bude použít podvozek s diferenčně řízenými koly, tedy s dvěma motory umístěnými v jedné ose, z nichž každý je řízen zvlášť a pro zajištění stability bývají doplněny jedním nebo více podpůrnými kolečky.. Motory bude nejlepší volit stejnosměrné s převodovkou, použití krokových motorů by nebylo vhodné zejména z důvodu vyšší hmotnosti a složitějšího řízení.

Nejdříve jsem začal hledat, zda by nevyhovoval nějaký hotový podvozek, což by mi ušetřilo práci s návrhem uchycení motorů. Během hledání podvozků jsem však zjistil, že většina podvozků je určena pro roboty s malou základnou a předpokládá se jejich rozšiřování do výšky, což se z důvodu přístupnosti pro náš přípravek nehodí, navíc jen málo podvozků má inkrementální senzory a jejich přidání by bylo pracné, což znamená ztrátu výhody použití hotového podvozku. Z těchto důvodů jsem použití hotového podvozku rychle zavrhl a začal jsem hledat pouze vhodné motory.

2.3.1 Dimenzování motoru

Abych vybral vhodné motory, musel jsem nejdříve určit, jak silné motory potřebuji, abych nepoužil motory, které přípravek neutáhnou, nebo motory, které budou příliš výkonné a tím i těžké. Pro výpočet jsem použil vzorců (1) z literatury (8), konkrétně z kapitoly 2.3, zabývající se právě dimenzování motorů

$$\begin{aligned} P_m &= mgv\left(\mu\cos\alpha + \sin\alpha + \frac{a}{g}\vartheta\right) \\ P'_m &= k_b\, P_m \\ M &= \frac{P'_m\, r}{v} \end{aligned} \qquad \textbf{(1)}$$

Kde P_m je požadovaný výkon motoru, P'_m upravený výkon, M požadovaný moment, m hmotnost robota, v požadovaná rychlost robota, μ trakční součinitel tření, α sklon, který má robot vyjet, a požadované zrychleni robota, g gravitační konstanta, ϑ součinitel rotačních odporů k_b bezpečnostní faktor a r poloměr kola.

Tento vzorec není zcela přesný, ovšem vzhledem k tomu, že některé konstanty jsou špatně zjistitelné a většinu z nich budu pouze odhadovat, bude tato zjednodušená forma na odhad potřebné síly motoru dostačující. Navíc je do vzorce přidán takzvaný bezpečnostní faktor, který upravuje potřebný výkon tak, aby zakryl nepřesnosti, a snižuje tak riziko příliš malého motoru. Po zvolení konkrétních komponent pohonného subsystému je navíc doporučeno výpočet provést znovu dle přesnějšího vzorce s použitím konkrétních parametrů motoru.

Část konstant volím stejně, jako v příkladu u použitého vzorce v uvedené literatuře, je to především proto, že tyto hodnoty nemohu zjistit nebo v případě rychlosti a zrychlení se mi tyto hodnoty zdají rozumné. Hodnotu trakčního součinitele tření μ tedy volím 0,3, hodnotu součinitele rotačních odporů ϑ =1,3, rychlost v =0,15 m/s a zrychlení a=0,8. Dle průměru možných motorů a dostupných kol předpokládám použití kol o průměru 4 cm, poloměr kola r tedy bude 0,02 m. Vzhledem k tomu, že je očekáván pohyb přípravku pouze na pracovním stole nebo na připravené rovné dráze, volím sklon dráhy α =0°. Bezpečnostní faktor k_b zajišťuje zvýšení potřebného výkonu, aby byly kompenzovány zanedbané ztráty v převodovce a špatně stanovené konstanty. Tuto hodnotu jsem zvolil stejně, jako je uvedeno v literatuře (6), tedy k_b =1,5.

2.3.2 Odhad hmotnosti přípravku

K výpočtu dále potřebuji znát hmotnost přípravku, tu musím odhadnout, protože v současné době mám pouze hrubou představu o finální podobě přípravku a z komponent nemám zatím vybráno nic. V průběhu odhadu budu brát v úvahu vždy těžší variantu, abych si byl jist, že budou motory dostatečně silné, až bude návrh přípravku před dokončením, mohu zvážit, zda by nešel použít slabší motor.

Prozatím se budu držet konstrukce zobrazené na Obr. 1, základ tedy bude tvořit nosná deska z hliníku, případně ze sklotextitu. Budu vycházet z těžší varianty, tedy desky z hliníku o tloušťce 1,5 mm. Předpokládám, že rozměr základny bude 20 x 16 cm což je při hustotě hliníkových desek 2700 kg/m^3 hmotnost 130 g.

Hmotnosti motorů, které by mohly výkonově odpovídat potřebám, jsou do 200g, celková hmotnost dvou motorů tedy bude 400 g.

Pro odhad hmotnosti plošného spoje s procesorem a displejem použiju údaje u hotových procesorových modulů v některém internetovém obchodě. Jako referenční modul jsem

vybral procesorový modul Orangutan Robot Controller Mega 8 z internetového obchodu (3), který je mimo jiné osazen malým displejem a jeho rozměry jsou 50,8 mm x 47 mm při hmotnosti 28 g. Velikost mého procesorového modulu odhaduji na 100 mm x 150 mm, hmotnost tedy bude šestkrát větší než referenční modul, tedy přibližně 170 g, stejnou hmotnost budu počítat pro připojené moduly.

Přípravek bude vhodné vzhledem ke své mobilitě napájen bateriemi. Jako baterie připadají v úvahu buď klasické NiMH/NiCD článkové baterie velikosti AA nebo baterie Liion, které jsou menší a lehčí. Napájecí napětí přípravku, které potřebuji znát pro určení počtu článků, bude určeno až podle volby motorů, ale maximálně bude 12 V. Pro výpočet opět použiji těžší variantu, tedy NiMH články pro napětí 12 V. Jeden NiMH článek má napětí 1,2 V, budu jich tedy potřebovat 10. Dle informací v internetovém obchodu GM electronic(9) je hmotnost baterie AA [4] přibližně 29 g, celková hmotnost baterií tedy bude 300g.

Tab. 1 Přehled odhadu hmotnosti jednotlivých částí přípravku

Položka	Hmotnost
Hliníková deska 200 x 160 x 1,5 mm	130g
Motory	400g
Baterie 10x NiMH AA	300g
Procesorová deska	170g
Moduly	170g
Celkem	1170g

V je shrnut odhad hmotnosti přípravku popsán v této kapitole. Odhad celkové hmotnosti přípravku tedy je 1,2 Kg.

2.3.3 Výběr vhodného motoru

V předchozích kapitolách jsem určil všechny hodnoty z rovnic (1), nyní tedy mohu do rovnic dosadit a spočítat potřebný moment motoru. Rovnice (1) však počítá pouze s jedním motorem, vzhledem k tomu, že použiji motory dva, musím ještě výsledný moment vydělit dvěma.

[4] Byla vybrána pásková baterie s označením B-N2000AALF, u které je udána hmotnost

$$
\begin{aligned}
P_m &= 1,2\,Kg\;9,8\,ms^{-2}\;0,15ms^{-1}\left(0,3\cos 0° + \sin 0° + \frac{0,8ms^{-2}}{9,8ms^{-2}}1,3\right) = 0,71W \\
P'_m &= 1,5*0,71W = 1,07W \\
M &= \frac{1,07W\;0,02\,m}{2*0,15ms^{-1}} = 0,07164\,Nm
\end{aligned}
\qquad \textbf{(2)}
$$

Hledám tedy motor s převodovkou, inkrementálním senzorem, napájecí napětí zvolím mezi 6 a 12 V a s krouticím momentem 0,07 Nm.

Motory s odpovídajícími parametry jsem nalezl dva, oba z internetového obchodu Snail Instruments (5). První motor EMG30, který má napájecí napětí 12V, maximální otáčky 200 ot/min[5] a stojí 1270 Kč. Druhým motorem je motor GHM-04, který má napájecí napětí 7,2 V, maximální otáčky 175 ot/min a cena 812 Kč. Tento motor sice přímo nemá inkrementální senzor, ale lze k němu dokoupit senzor QME-01 za 864 Kč, který je uzpůsoben k připojení k tomuto typu motoru. Při porovnání těchto motorů jsem dospěl k závěru, že lepší by bylo použít motor EMG30 a to nejen kvůli nižší ceně, ale i kvůli hladkému tvaru oproti motoru GHM-04, který se skládá ze tří válců různých průměrů, což by mohlo působit potíže při zabudování motorů. K motoru EMG30 existuje i řídicí jednotka komunikující přes I2C, nevýhodou je ovšem celková cena. Cena samotných motorů a řídicí jednotky je 4400 Kč, což nabádá k hledání jiného, levnějšího řešení.

Jako levnější řešení motorů jsem našel motory ze stavebnice Lego Mindstorms NXT, které dle cen na stránkách této stavebnice (10) stojí 600 Kč. Prodejce motorů sice neuvádí žádné parametry, ale z praktických zkušeností vedoucího bakalářské práce, pana Ing. Hlinovského, z výuky předmětu Roboti (11) jsou motory pro pohon přípravku dostatečně silné. Tento motor má integrován i inkrementální senzor a dle parametrů řídicí jednotky, která motor ovládá, je jeho napájecí napětí 7,5 až 9 V, přesnější parametry se mi bohužel získat nepodařilo.

Použití motorů ze stavebnice Lego Mindstorms NXT sebou přináší i menší komplikace, jednak s uchycením samotných motorů, které jsou uzpůsobeny pouze k uchycení pomocí dílů stavebnice Lego, a s přívodním kabelem, protože konektor použitý na motoru je specifický

[5] Otáčky za minutu

pro stavebnici Lego Mindstorms NXT a není snadno k dostání. Problém s uchycením motorů se chystám vyřešit vytvořením základní konstrukce pro uchycení motorů ze stavebnice lego a tuto konstrukci pak napojím na hlavní konstrukci přípravku i za cenu poškození konstrukčních dílů z lega.

Připojení kabelem lze vyřešit několika způsoby, buď lze koupit a upravit originální kabely ke stavebnici Lego Mindstorms NXT, nebo lze objednat samostatné konektory v internetovém obchodě Mindsensors (12), případně podle návodu na stránce (13) upravit klasický konektor RJ-12 tak, aby šel použít se stavebnicí Lego Mindstorms NXT.

Po zvážení možností jsem se rozhodl použít motory ze stavebnice Lego Mindstorms NXT.

2.4 Možnost použití hotového modulu

Při hledání informací o motorech ze stavebnice Lego Mindstorms NXT jsem zjistil, že řídicí modul této stavebnice, nazývaný NXT Intelligent Brick, je otevřený projekt s firmwarem pod licencí OpenSource a se zveřejněným schématem. Začal jsem se tedy o tento modul více zajímat a zjistil jsem, že vnitřní struktura modulu je podobná té, která byla plánována pro laboratorní pracoviště, a rovněž vstupně výstupní porty jsou podobné mému plánu. K NXT dále existuje množství programovacích nástrojů, z nichž některé jsou placené a použití některých je vztaženo k NXT, ale řada z programovacích nástrojů je zdarma a většinou pod licencí OpenSource. Pokud bychom tedy vytvořily přípravek tak, aby byl kompatibilní s firmwarem NXT Brick, získali bychom možnost využívat řadu již odladěných nástrojů a funkcí, jejichž naprogramování by jinak zabralo spoustu času a úsilí.

V průběhu konzultace s vedoucím bakalářské práce, kde byl probrán aktuální stav práce a různé možnosti kam práci dále vést, jsme se dohodli, že pozměníme směr vývoje a dál se bude pokračovat přizpůsobením NXT Brick požadavkům pro výuku. Před tímto rozhodnutím byly probírány celkem tři možnosti. První možností bylo použít originální modul z NXT a kolem něj vytvořit rozhraní umožňující připojení vlastních modulů. To by znamenalo vytvoření konstrukce pro uchycení modulu NXT a její spojení se zbytkem přípravku, dále vytvořit kabelové redukce, ke kterým by šlo snadno připojit jednotlivé moduly, na kterých budou pracovat studenti. Jako nevýhodu této varianty jsme viděli především nestejnorodost přípravku, obtížnější zapracování NXT do přípravku a nemožnost ovlivnit vnitřek procesorového modulu.

Druhou možností bylo vytvoření procesorového modulu odvozeného od NXT a upraveného dle potřeb, ale pokud možno kompatibilního s původním NXT. Tato varianta kombinuje původní záměr vytvořit nový procesorový modul a možnost využití NXT Brick. Toto řešení přináší drobné nevýhody plynoucí z nutnosti zachovat velkou část z původního NXT, ovšem přináší výhodu použití ověřeného schématu a softwarových nástrojů. V porovnání s první variantou má tato výhodu snadnějšího přizpůsobení tvaru přípravku potřebám předmětu a vzhledem k pořizovací ceně NXT Brick 4500 Kč předpokládáme i možnost nižších pořizovacích nákladů.

Poslední diskutovanou variantou bylo pokračování v samostatném vývoji vlastního přípravku pouze s možnou inspirací z mindstorms NXT. Tato varianta byla nakonec stanovena jako záložní pro případ, že by se při realizaci druhé možnosti objevil nějaký závažný problém, který by nebylo možné vyřešit.

3 Návrhy úloh

Cílem této kapitoly je shrnout mnou sesbírané informace a nápady pro návrh jednotlivých úloh, aby mé poznatky mohly být využity během jejich přípravy. Vzhledem k tomu, že na podrobnější přípravu úloh nezbyl žádný čas, tak se většina mé práce na úlohách skládala pouze z nápadů, hledání a studování zdrojů. Vymýšlením jednotlivých úloh jsem se zabýval pouze v průběhu specifikace parametrů pracoviště, abych měl představu o potřebách pro připojení modulů, v dalším průběhu práce jsem si pouze upřesňoval svoji představu o úlohách.

3.1 Optické senzory

V průběhu prvních konzultací, kde jsme probírali především podrobnosti zadání a použití pracoviště, bylo zmíněno několik typů optických senzorů, které mohou dobře posloužit k doplnění výuky. Především šlo o senzor rozpoznávající barvy a o senzor, který by umožnil rozpoznání čáry na podlaze a jízdu podle ní.

3.1.1 Barevný senzor

Jednoduché barevné senzory zpravidla fungují na principu osvětlení zkoumaného předmětu světlem známé barvy a měření intenzity odraženého světla. Takto se provede měření pro několik, zpravidla tří, barev a z intenzit světla pak lze určit správnou barvu. Problémem této metody je její náchylnost vůči okolnímu světlu a dlouhá doba měření. V době měření by senzor měl být co nejblíže měřenému předmětu. Dále se musí zajistit zakrytí měřené oblasti tak, aby na senzor dopadalo co nejméně okolního světla. Čím větší je intenzita okolního světla vzhledem ke zdroji barevného světla senzoru, tím hůře se bude barva rozpoznávat. Podobně by zdroje světla uvnitř senzoru měly být umístěny tak, aby se světlo z nich dostalo na senzor pouze odrazem od měřeného předmětu.

Při konstrukci barevného senzoru je zapotřebí vybrat dostatečně citlivý senzor intenzity světla, který má pokud možno stejnou citlivost pro celé viditelné spektrum. Citlivost je zpravidla nepřímo úměrná rychlosti měření, při výběru senzoru je proto nutno volit kompromis mezi citlivostí senzoru a rychlostí měření. Při volbě rychlosti reakce světelného senzoru musí být brán v úvahu fakt, že pro každé získání hodnoty musí být provedena minimálně čtyři měření, tři pro zjištění poměru barev a jedno pro zjištění úrovně okolního světla.

Při výběru LED diod jako zdrojů barevného světla je zapotřebí vhodně zvolit jejich parametry a předem si vyzkoušet jejich seřízení se senzorem. Pro seřizování je výhodné měnit jas jednotlivých diod změnou předřadných odporů, i tak může být vhodný výběr diod obtížný. Problémy se seřizováním vyvážení barev (v tomto případě zelené) jsou zmíněny na webové stránce (14).

Dle webové stránky (15) ovlivňuje přesnost měření také rozmístění jednotlivých LED diod. Jejich paprsky by měly mířit na jedno místo, aby byla měřená oblast co nejmenší a zároveň by jejich paprsky měly být rovnoběžné, aby při naklonění určovaného předmětu nedocházelo u některé diody k většímu odrazu než u ostatních, měření by tím bylo zkresleno. Po těchto připomínkách se zdá být logické použití diody obsahující všechny tři barvy. Zdroj ale také uvádí praktickou zkoušku jedné RGB diody, jejíž jednotlivé barevné složky se na měřeném předmětu značně rozcházely, to bylo pravděpodobně způsobeno nepatrným odstupem jednotlivých LED diod na čipu a kulatého tvaru těla LED diody. Dále je uváděno, že byl vyzkoušen pouze jeden typ RGB diody a mohlo tedy jít o vadu pouze jediného typu.

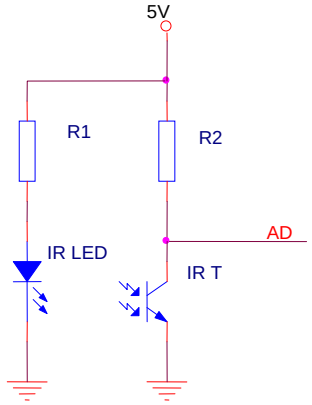
Po průzkumu různých informací o sestavení barevného senzoru jsem tedy zjistil, že konstrukci samotné elektroniky lze provést velice snadno, problematické však je vybrat vhodné součástky a nastavit senzor tak, aby rozpoznával širší spektrum barev. Při využití tohoto senzoru v rámci úlohy bych na studentech nechal navrhnout rozmístění diod a světelného senzoru

3.1.2 Senzor pro sledování čáry

Senzor pro sledování čáry je vlastně jednoduší variantou barevného senzoru, který rozpoznává pouze světlost snímaného povrchu, v mnoha případech tento senzor rozpoznává pouze dvě úrovně odraženého světla. Jako snímaný povrch se používá jednobarevný povrch, na kterém je vyznačená kontrastní čára, ve většině případů se používá světlý povrch a černá čára, ovšem ve složitějších aplikacích bývá použit i různobarevný podklad a čára, toto řešení je ovšem komplikovanější a většinou využívá barevného senzoru.

Pro snímání se nejčastěji používá infračerveného záření (IR), které má tu výhodu, že senzor není tolik rušen viditelným světlem a naopak záření senzoru neobtěžuje lidi v okolí. Podmínkou použití IR záření však je zachování podmínky kontrastní čáry oproti podkladu i pro IR záření. V knize Mobilní roboty(8) v kapitole 3.2.6 je uvedena asi nejjednodušší konstrukce senzoru (Obr. 3). Výstup fototranzistoru je přiveden přímo na vstup AD převodníku,

tato varianta umožňuje snadnou konstrukci s minimem součástek a veškeré vyhodnocování provádí mikroprocesor. Často se však používá zapojení doplněné zesilovačem a odpory R1 a R2 z Obr. 3 bývají doplněny proměnnými odpory, kterými se nastavuje proud fotodiodou, respektive citlivost fototranzistoru. Pokud je potřeba co nejvíce snížit nároky na zpracování signálu, bývá výstup senzoru místo do AD převodníku zapojen na komparátor, na kterém se poté nastavuje hranice mezi tmavou a světlou barvou.



Obr. 3 Nejjednodušší zapojení senzoru čáry

Požadavky na hardwarové uspořádání jsou celkem malé, je nutno pouze zajistit, aby dioda i fototranzistor mířily na stejné místo a odražený paprsek šel přímo k fototranzistoru. Dále by vzdálenost mezi senzorem a podlahou měla být minimální. Průběh měření je rovněž jednoduchý, zvláště díky tomu, že senzor zpravidla bývá umístěn na podvozku robota, kde se nepředpokládá rušení okolním světlem nebo IR zářením.

Pro tuto úlohu bych doporučoval použít variantu s komparátorem na výstupu, studenti si tak při návrhu celé úlohy mohou vyzkoušet návrh zesilovače, komparátoru a princip optického senzoru. Úkolem studenta by bylo kromě sestavení senzoru také jeho nastavení tak, aby správně rozpoznával studentům předloženou čáru a dále by studenti mohli porovnávat spolehlivost svého nastavení i pro jinou kombinaci podkladu a čar. Úloha by mohla být poměrně atraktivní, neboť po připojení senzoru k pracovišti bude přípravek schopen samostatně úlohu předvést a úspěšnost plnění úlohy mohou studenti zhodnotit sami. Tato úloha se mi, vzhledem k minimálním nárokům na připojené moduly i samotný procesorový modul, jeví jako vhodná pro seznámení studentů s přípravkem.

3.2 Úloha I2C sběrnici

V požadavcích na pracoviště bylo zdůrazněno použití I2C sběrnice s možností její aplikace při úlohách. Hlavní využití I2C sběrnice v úloze by měl být praktický průzkum její funkce. Jako nejlepší způsob uvedení studentů do I2C sběrnice se zdá využití její adresovatelnosti a přimět studenty k vyčtení požadované adresy ze sběrnice a dle požadované adresy nastavit cílové zařízení.

Úloha by měla probíhat následovně. Po spuštění programu pro tuto úlohu na procesorovém modulu bude náhodně vybraná adresa některého z čipu, který mají studenti k dispozici, na tuto adresu se bude procesorový modul periodicky snažit posílat data. Student bude muset připojit k I2C sběrnici osciloskop a zjistit cílovou adresu. Podle adresy a katalogových listů jednotlivých I2C čipů student vybere ten, jehož adresa odpovídá adrese požadované, dle katalogových listů ho zapojí. Po připojení sestaveného modulu k procesorové desce bude vyhodnocena správnost zapojení, po čemž student může pokračovat v další části úlohy, kterou by mělo být ověřování vlastností I2C sběrnice měřením.

Tato úloha je koncipována tak, aby se studenti seznámili s I2C sběrnicí a zároveň prostřednictvím hledání správného čipu v katalogových listech získali přehled o dostupných periferiích pro I2C rozhraní.

4 Popis mindstorms NXT

V této kapitole podám základní informace o Lego mindstorms NXT zejména z pohledu použití pro laboratorní pracoviště. Dále uvedu některé informace o prvcích, které nejsou přímo součástí NXT mindstorms, ale rozšiřují jeho možnosti. Většinu informací o Lego mindstorms jsem čerpal z internetových stránek stavebnice (10), zejména ze souborů Hardware developer kit, které jsou umístěny v elektronických přílohách (1) a (2). Některé podrobnější informace jsem používal i z knihy(16).

Mechanickými díly vychází Lego mindstorms ze stavebnice Lego Technic, Lego mindstorms je však uzpůsobeno především na stavbu robotů a automatů, tomu také odpovídá jeho vybavení. Pod označením mindstorms NXT, nebo jenom NXT je myšleno především elektronické vybavení stavebnice Lego mindstorms se skládá především z procesorového modulu (Obr. 4), dalším vybavením jsou různé druhy senzorů a motorů.

Obr. 4 NXT Brick s připojeným základním příslušenstvím

4.1 NXT Intelligent Brick

Hlavní částí NXT je řídicí modul nazývaný NXT Intelligent Brick, nebo pouze NXT Brick, který je řízen mikroprocesorem AT91SAM7S256 vyráběný firmou Atmel (17). Modul lze připojit k PC pomocí USB, přes které lze modul ovládat a programovat. K ovládání a komunikaci může rovněž sloužit Bluetooth, které umožňuje NXT Brick se spojit s PC, nebo s jiným NXT Brick.

K ovládání samotné NXT Brick slouží 4 klávesy umístěné na čelní straně, 2 klávesy slouží k posuvu do stran, jedna pro potvrzení volby a poslední slouží pro návrat zpět. Pro zobrazování informací slouží grafický LCD displej o rozměrech 100 x 64 pixelů, na kterém se zobrazuje menu firmwaru a při běhu uživatelského programu na něj lze vypsat až 8 řádek

textu nebo zobrazit grafický obrazec. Uživatelský výstup je doplněn o vnitřní miniaturní reproduktor, který je schopen generovat zvuk od jednoduchých tónů po přehrávání hudby ze souboru uloženém v paměti.

Nejdůležitější na NXT Brick jsou však konektory pro připojení periferních modulů. Je jich celkem 7, tři slouží pro připojení motorů a zbývající čtyři pro připojení senzorů. Všechny tyto konektory jsou typu RJ-12, ovšem zámek není uprostřed, jak bývá obvyklé, ale je na pravé straně, tudíž nelze přímo použít standardní konektory. Ty jsou dostupné pouze v internetovém obchodě Mindsensors(12), který se specializuje na prodej elektronických doplňků k Lego Mindstorms. Ukázka konektorů NXT je na Obr. 5.

a) **b)**

Obr. 5 Konektory na mindstorms NXT a) konektor na NXT

b) vlevo klasická zástrčka RJ-12, vpravo zástrčka NXT

Každý z konektorů pro motory obsahuje dva piny s PWM výstupem pro napájení motorů s maximálním trvalým odběrem do 0,7A, dva piny jako vstup pro inkrementální senzor otáček motoru a napájecí napětí 4,3V.

Napájecí napětí 4,3V je přítomno na všech vstupně výstupních konektorech a je připojeno na společný proudový omezovač, který dovolí odebírat přibližně 180mA pro všechna zařízená připojená na napájecí napětí 4,3V. Přesnou hodnotu napájecího napětí označeného jako 4,3V však nelze přesně určit, dle knihy Extreme NXT(16) se maximální napětí blíží 5V, s rostoucím odebíraným proudem pak klesá. Napájení motorů je identické s napájecím napětím, mělo by se tedy pohybovat mezi 7,5V a 9V.

Každý z konektorů pro připojení senzorů obsahuje napájecí napětí 4,3V, které již bylo popsáno v předchozím odstavci. Dále obsahují jeden analogový vstup, který obsahuje pull-up[6] rezistor na 5V a je schopen pracovat v takzvaném napájecím režimu. V tomto režimu se

[6] Rezistor připojený proti napájecímu napětí, čímž je na výstupu v klidovém stavu zajištěna logická 1

na analogový vstup přepíná proudový zdroj dodávající přibližně 14mA do napětí o něco málo nižší než je napájecí napětí modulu. Pokud je zapnut napájecí režim analogového vstupu, je připojen proudový zdroj na dobu 3ms, poté je odpojen a 0,1ms je měřeno napětí na vstupu, poté se opět připojí proudový zdroj a celý cyklus se opakuje. Dále jsou konektory pro senzory osazeny dvěma vstupně výstupními piny s logickými úrovněmi 3,3V, jsou však 5V tolerantní[7] a měly by být schopny snést i větší napětí. Tyto piny se mohou chovat i jako I2C sběrnice, každý konektor může samostatně fungovat jako master I2C sběrnice, pro její používání je však nutno zajistit připojení externích pull-up rezistorů, které jsou pro fungování I2C sběrnice nutné.

Všechny konektory pro připojení senzorů jsou stejné, drobnou výjimkou však je konektor 4, který má navíc na vstupně výstupní piny připojeno rozhraní RS485. Zároveň však lze používat pouze jednu funkci vstupně výstupních pinů.

4.2 Periferie pro NXT

Základními periferiemi, které jsou dodávány s Lego mindstorms NXT jsou motory se zabudovanými inkrementálními senzory, kontaktní spínače, barevný senzor, mikrofon a ultrazvukový senzor pro měření vzdálenosti.

Kromě těchto základních senzorů přímo od Lega lze objednat další senzory a rozšiřující moduly od jiných výrobců, kteří se specializují na komponenty ke stavebnici Lego mindstorms, například (12) a (18). Tito výrobci nabízí senzory jako například elektronický kompas, senzory pro měření různých fyzikálních veličin a kameru, která poskytuje informace o pozici předem definovaných barevných předmětů. Kromě senzorů jsou nabízeny i adaptéry, které rozšiřují možnosti NXT Brick. Jde především o různé multiplexory, které umožní připojit více standardních senzorů a převodníky na jiná rozhraní, která nejsou podporována přímo NXT jako například PS2 a USB.

Pro laboratorní pracoviště však nemám v plánu použít původní periferie z NXT, pokud by přesto byly nějaké použity, bude to pravděpodobně pouze k doplnění úlohy, nebo pro porovnání parametrů s moduly vlastní výroby. Několik návodů na vlastní senzory je například

[7] Na vstupu jsou schopny zpravovat i logické signály 5V logiky

v knize Extreme NXT (16), jde ale spíše o jednoduché senzory ukazující možnost rozšíření NXT.

4.3 Software k NXT

V této části popíši software, který lze použít pro práci s NXT, zejména s ohledem na jeho využití při práci s laboratorním pracovištěm.

Při popisu jednotlivých programů pro práci s NXT začnu od toho základního, čímž je samotný firmware v NXT Brick. Standardní firmware v NXT Brick je samostatný operační systém, který spravuje prostředky NXT a poskytuje je spuštěným programům a připojeným zařízením. Kromě přístupu k periferiím poskytuje NXT také přístup k souborovému systému ve Flash paměti procesoru, kde jsou také ukládány soubory s uživatelskými programy a daty. Uživatelské prostředí firmwaru má příjemnou grafickou podobu a umožňuje kromě spouštění uživatelských programů také sledovat aktuální hodnoty připojených senzorů, správu souborového systému a dokonce i vytvoření jednoduchého programu přímo v NXT Brick. Firmware dále v základu obsahuje několik programů určených pro vyzkoušení správné funkce motorů a základních senzorů.

Uživatelské programy jsou v NXT Brick spouštěny pod virtuálním strojem, který používá vlastní instrukce. Popis uložení dat programových souborů pro NXT spolu s popisem instrukcí visuálního stroje je v souboru Software developer kit na stránkách Logo mindstorms(10), který je umístěn i v elektronické příloze 4.

K programování NXT existuje řada programovacích jazyků, z nichž některé vyžadují speciální firmware. Přehled některých programovacích jazyků, které se používají pro programování NXT Brick a připadají v úvahu k použití pro programování laboratorního pracoviště, je zobrazen v Tab. 2. Jde o stručné shrnutí z tabulky umístěné na webové stránce (19), přehled dalších programovacích jazyků pro NXT Brick včetně informací o nich lze získat například na anglické Wikipedii (20).

Tab. 2 Přehled některých programovacích jazyků a prostředí pro NXT

	NXT-G	RoboLab	NBC	NXC	RobotC	LabVIEW	leJOS NXJ
Jazyk	Grafický	Grafický	Asembler	Podobný C	C	Grafický	Java
Firmware	Standard	Standard	Standard	Standard	Standard	Standard	Vlastní
Podpora událostí	Ne	Ano	Ne	Ne	Ano	Ne	Ano
Licence	Spolu s NXT	Placený/ 30 dní	Opensource	Opensource	Placený/ 30 dní	Placený/ 30 dní	Freeware

Programovací jazyky založené na grafickém programování jsou vhodné zejména k rychlému vytvoření programů a jsou vhodné i pro ty, kteří neumí žádný programovací jazyk. Tyto programovací jazyky jsou tedy určeny spíše pro děti a školní výuku. Laboratorní pracoviště se však bude programovat pouze v rámci vytváření úlohy a grafické programování by v tomto případě mohlo být obtížnější. V neprospěch grafických prostředí hovoří i fakt, že nejsou zdarma, snad s výjimkou NXT-G, který je dodáván s NXT Brick, ale ani ten není zdarma. NXT-G se navíc vyznačuje poměrně malou výkonností a velkou velikostí programů.

Programovací prostředí RobotC, které používá programovací jazyk C, vyniká oproti ostatním prostředím především ve svojí efektivitou. Lze v něm snadno napsat velice rychlé programy s malou velikostí programu. Toto prostředí se tedy hodí zejména pro aplikace, kde je zapotřebí vysoká efektivita, lze ho však získat pouze jako 30 denní zkušební verzi, nebo je ho nutné koupit.

Z programových řešení, která jsou zdarma, se pravděpodobně nejvíce používají programovací jazyky NBC[8] a NXC[9], oba jsou integrovány ve vývojovém prostředí BricX Command Center(21), které je šířeno pod licencí Opensource. Jazyk NBC je programovací jazyk na bázi asembleru a NXC je programovací jazyk vytvořený pro NXT po vzoru jazyka C. Oba jazyky jsou schopny plně využít možnosti poskytované virtuálním strojem v NXT a poskytují řadu rozšířených funkcí, které programátorovi zjednoduší některé činnosti. Vývojové prostředí pak poskytuje množství funkcí pro správu NXT Brick včetně správy souborů v NXT a nahrávání firmwaru, čímž je umožněno používat k práci s NXT Brick pouze toto vývojové prostředí. Další výhodou tohoto programové řešení je jeho rozšířenost a tím pádem i poměrně široká komu-

[8] Next Byte Code
[9] Not eXactly C

nita, na kterou se lze obracet s problémy.

Jako příklad dalších programových řešení jsem uvedl leJOS NXJ, které používá jako programovací jazyk Javu. Toto řešení podobně jako řada dalších však vyžaduje použití jiného firmwaru, který není kompatibilní s původním, a také se po prozkoumání informací na internetu zdá, že není tolik rozšířené jako zmiňované NXC. Použití jiného firmwaru bylo zamítnuto zejména kvůli možnosti kompatibility programů pro laboratorní pracoviště a původní NXT Brick z důvodu přenositelnosti programů. Díky přenositelnosti by například studenti mohli sestavit senzory pro laboratorní pracoviště a vyzkoušet je s pomocí programu, který dříve vytvořili v předmětu Roboti(11), kde se pracuje se stavebnicí Lego mindstorms NXT. Studenti by tak získali možnost vytvořit téměř celého mobilního robota s výjimkou řídicí jednotky a po částech ověřit jeho funkci.

Pro programování laboratorního pracoviště byl tedy prozatím zvolen programovací jazyk NXC s vývojovým prostředím BricXcc , které by mělo plně postačovat veškerým požadavkům pro práci s pracovištěm.

5 Popis a návrh obvodů pracoviště

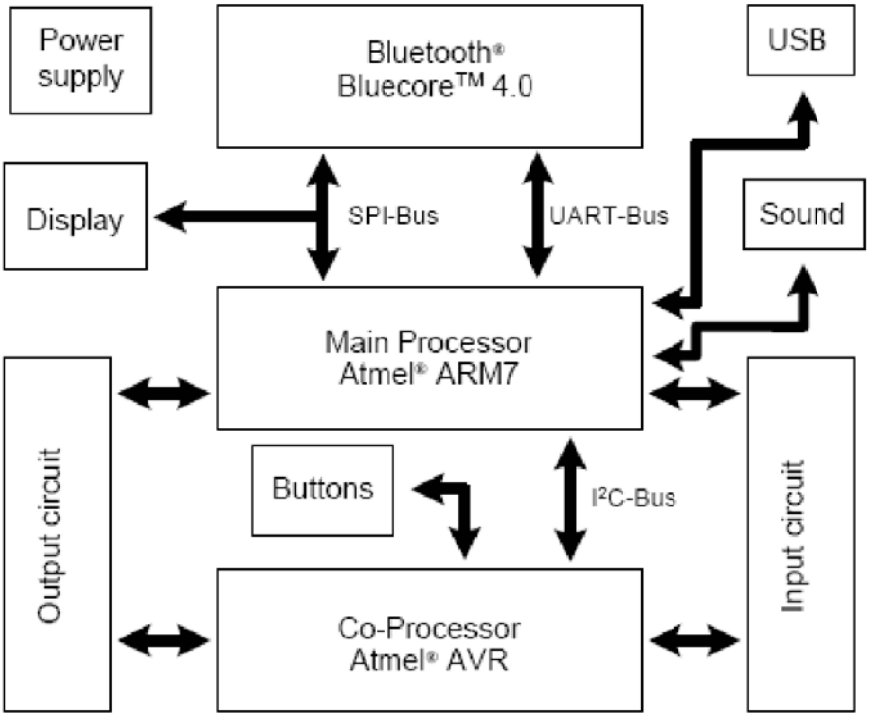
V této kapitole popíši funkci jednotlivých částí původní NXT Brick a zároveň popíši změny, které jsem v zapojení provedl včetně výměny součástek za dostupné typy. Pro zjišťování funkce NXT Brick jsem používal soubory z Hardware developer kit na stránkách Lego mindstorms(10), které jsou umístěny v elektronické příloze 1, 2 a 3. Pro zjištění funkce některých částí s návazností na mikroprocesory jsem občas nahlédl do zdrojových kódů firmwaru, který se rovněž nachází na stránkách Lego mindstorms(10) a v elektronické příloze 5 a 6. Při výběru dostupných součástek jsem vycházel z online katalogu firmy GM electronic(9) a z katalogu firmy Farnell (22), tento katalog mi pro objednávání hůře dostupných dílů doporučil vedoucí bakalářské práce a to především proto, že od distributora této firmy již dříve katedra řídicí techniky spokojeně nakupovala některé součástky. Při návrhu jsem používal datasheety nalezené u zmíněných prodejců, nebo jinde na internetu, jejich výběr je přiložen v elektronické příloze 13. Výsledná schémata pracoviště se nachází v přílohách 1 až 5.

5.1 Blokové schéma

Při pohledu na blokové schéma NXT Brick na Obr. 6 je vidět, že se příliš neliší od blokového schématu, které jsem původně navrhl (Obr. 2). Asi největší rozdíl je v použití druhého procesoru místo mnou plánovaného řídícího modulu pro motory. Dále je výstup inkrementálních senzorů zapojen do hlavního procesoru ARM, to je vyváženo přesunutím snímání klávesnice a analogových vstupů na koprocesor AVR. Klávesnice je k AVR připojena, aby procesor AVR mohl na základě stisku kláves řídit napájení procesoru ARM, který je v úsporném režimu vypnut. Napájecí napětí procesoru ARM (3,3V) bylo pravděpodobně důvodem, proč byly analogové vstupy přivedeny na procesor AVR, který má napájecí napětí 5V. Díky tomu je měřicí rozsah analogových vstupů 5V, což je běžné napětí pro práci s integrovanými obvody.

Již menšími rozdíly v blokovém schématu jsou přidání audio výstupu a sběrnice SPI, o kterou se dělí LCD displej a Bluetooth adaptér. Bluetooth adaptér však dle dokumentace SPI rozhraní běžně nepoužívá, veškerá komunikace probíhá přes rozhraní UART, SPI rozhraní slouží pouze pro nahrání nového firmwaru do Bluetooth adaptéru a tato funkce není ve standardním softwaru NXT Brick podporována.

Při porovnání blokových schémat jsem tedy dospěl k závěru, že bych za daných podmínek pravděpodobně dospěl k podobnému výsledku a použití základu NXT Brick pro laboratorní pracoviště tedy nebylo chybou.



Obr. 6 Blokové schéma NXT Brick

5.2 Procesory a jejich periferie

V této části budou popsány obvody, které přímo souvisí s procesory, celé schéma této části je v příloze Příloha 1.

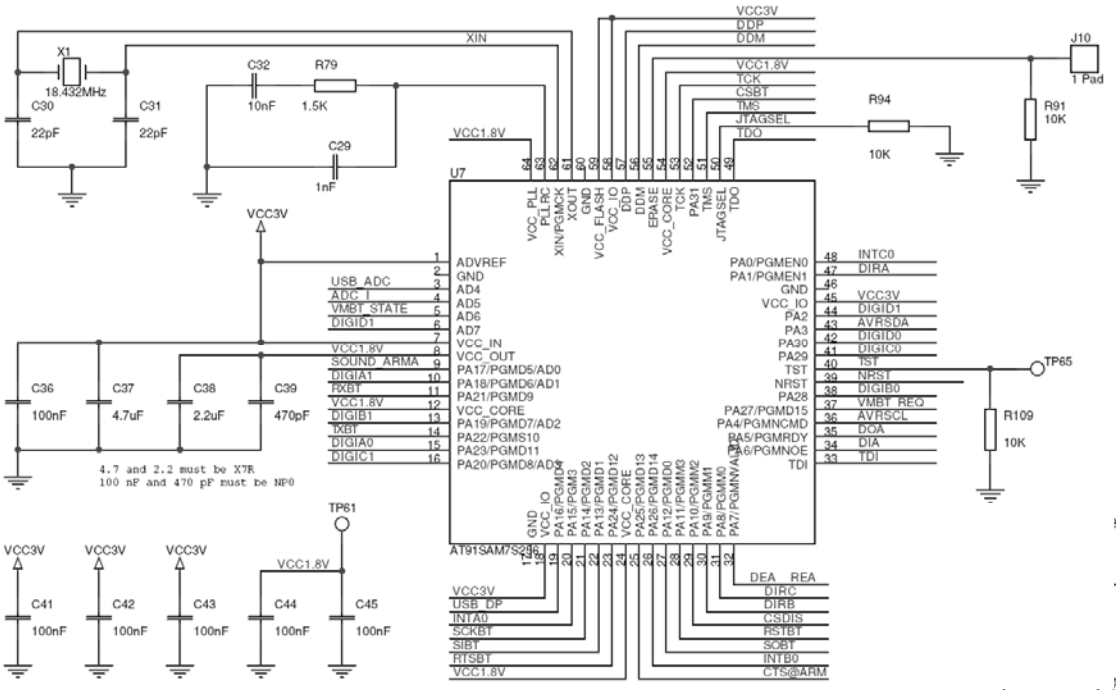
5.2.1 Zapojení procesoru ARM

V NXT Brick je jako hlavní procesor použit procesor AT91SAM7S256 od firmy Atmel (17), jde o procesor s jádrem ARM7TDMI a kapacitou flash paměti 256 kB. Tento procesor má napájení 3,3V, jádro však používá napětí 1,8V, které si procesor sám generuje, ale je nutno toto napětí rozvést na příslušné piny.

Zapojení procesoru v NXT Brick zobrazené na Obr. 7 odpovídá požadavkům uvedeným v datasheetu a to včetně připojení USB, které na obrázku není, ale je možné si ho prohlédnout v příloze Příloha 1. Dle informací, jak ve schématu tak v datasheetu procesoru, je

nutné použít správné typy kondenzátorů *C36,C37,C38* a *C39*, pokud by byly použity jiné typy, mohla by být ohrožena funkčnost procesoru.

V zapojení samotného procesoru jsem neměl důvod provádět žádné změny, neboť celá tato část schématu je zapojena přesně podle datasheetu, jedinou výjimkou jsou pull-down[10] rezistory *R91,R94* a *R100*, protože dle datasheetu jsou v procesoru interní pull-down rezistory a externí rezistory jsou tedy zbytečné. Z tohoto důvodu jsem tedy tyto rezistory ze schématu odstranil a na piny 55 a 50 jsem připojil testovací body, neboť by piny zůstaly bez spojení a v případě experimentů se zapojením přípravku, by nebylo možno na tyto piny nic připojit.



Obr. 7 Zapojení procesoru ARM

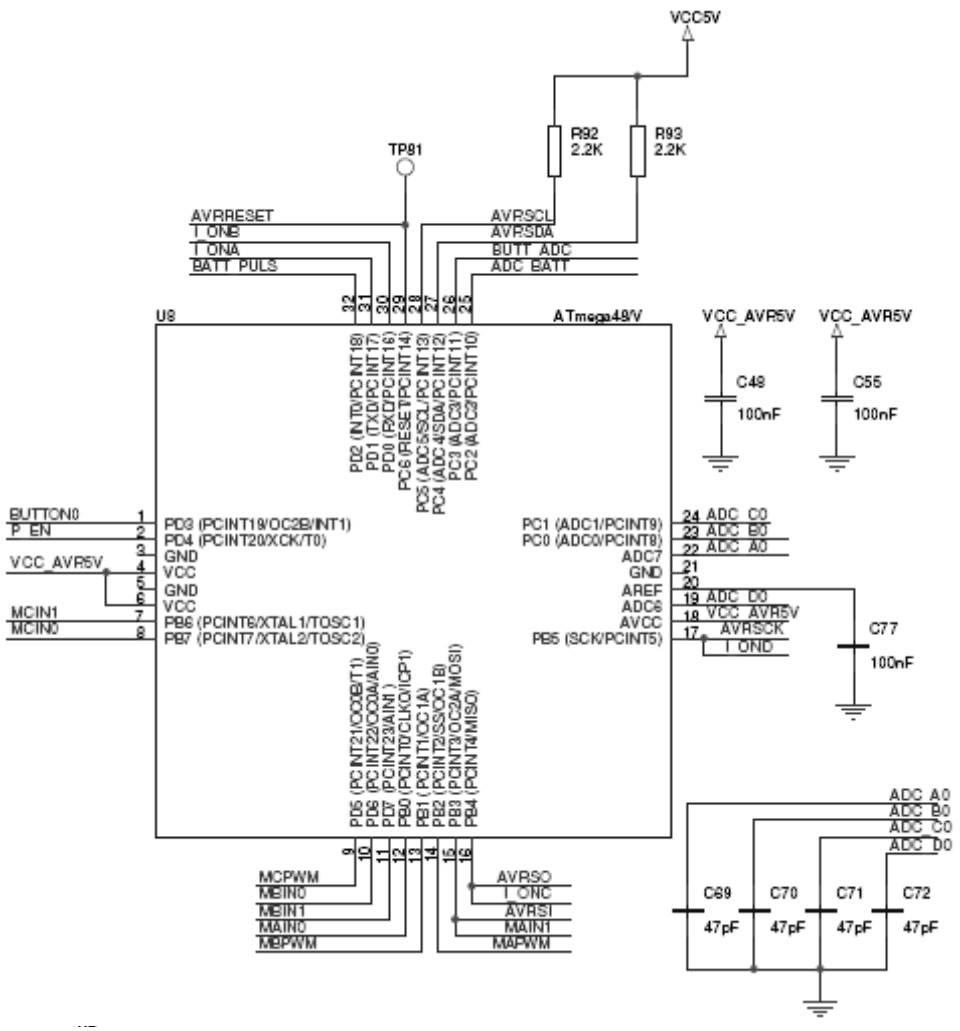
5.2.2 Zapojení procesoru AVR

Jako koprocesor je použit osmibitový mikroprocesor ATmega48 s jádrem AVR rovněž od firmy Atmel. Jeho funkce v NXT Brick je hlavně měření napětí na analogových vstupech, čtení hodnot z klávesnice a ovládání motorů pomocí PWM. Vedlejší funkce procesoru AVR je sledování napájecího napětí a ovládání napájení pro úsporu energie v době, kdy zařízení není používáno. Komunikace mezi procesory ARM a AVR probíhá po interní I2C sběrnici, po které

[10] Rezistor připojený proti zemi, čímž je na výstupu v klidovém stavu zajištěna logická 0

se posílají potřebné údaje, více o komunikaci mezi procesory bude popsáno v kapitole o softwaru obou procesorů. I2C sběrnice má v tomto případě vysokou logickou úroveň 5V, což je možné jen díky tomu, že procesor ARM má vstupy 5V tolerantní, jinak by mohlo dojít k poškození vstupů procesoru ARM.

Díky tomu, že je u procesoru AVR používán interní oscilátor, nejsou k jeho chodu třeba téměř žádné externí komponenty. Ve schématu s jeho zapojením (Obr. 8) jsou pouze kondenzátory pro filtraci napájecího napětí, kondenzátory pro filtraci analogových vstupů a pull-up rezistory *R92* a *R93*, které jsou nutné pro funkci sběrnice I2C.



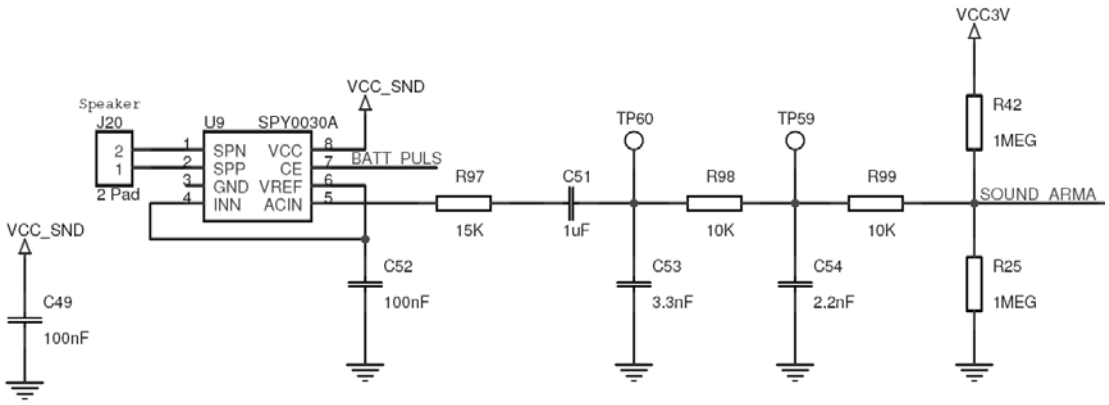
Obr. 8 Zapojení procesoru AVR

5.2.3 Audio výstup

Audiosignál je generován pomocí PWM signálu z procesoru ARM, tento signál je pak upraven dolními propustmi, které odstraní vzniklý šum a poté je signál zesílen v audio zesilovači. Původně byl použit zesilovač SPY0030A do firmy SUNPLUS, tento zesilovač však není

dostupný a proto jsem za něj musel hledat náhradu.

SPY0030A je 1W operační audio zesilovač s nastavitelným zesílením, jehož maximální zesílení je 20. Tento zesilovač má symetrický výstup a je vybaven vstupním signálem pro jeho vypnutí, čímž se sníží jeho spotřeba. Vypínání zesilovače je v obvodu využíváno procesorem AVR v případě, že procesor ARM nekomunikuje. V tomto případě začne procesor AVR zapínat a vypínat zesilovač, což se projevuje jemným cvakáním v reproduktoru.



Obr. 9 Původní zapojení audio výstupu

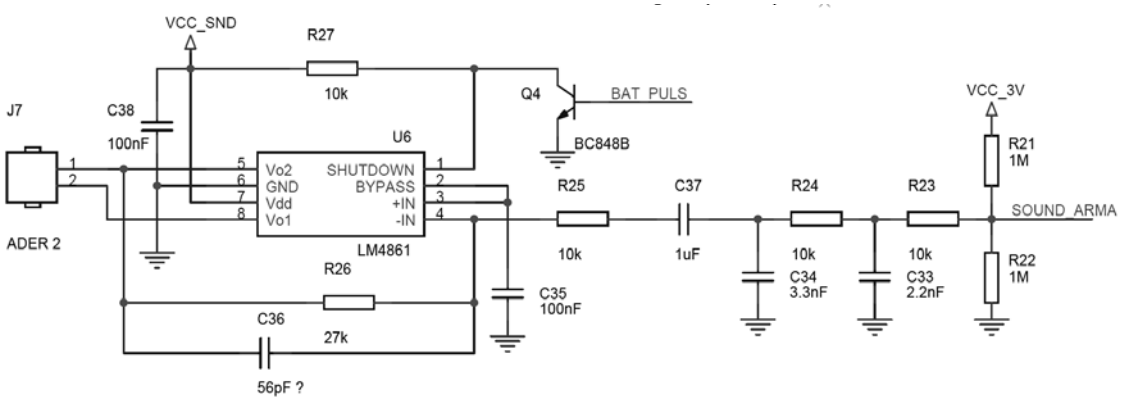
Pro výběr vhodné náhrady zesilovače potřebuji nejdříve zjistit jeho zesílení, to je dle datasheetu výrobce závislé na velikosti odporu R97, výpočet zesílení je uveden ve vzorci (3). Potřebuji tedy zesilovač, který má zesílení alespoň 5.

$$A = 2 * \frac{50000}{5000 + R} = 2 * \frac{50000}{5000 + 15000} = 5 \qquad \textbf{(3)}$$

Z dostupných zesilovačů jsem vybral symetrický operační audio zesilovač LM4861, který je u dodavatele součástek běžně skladem. Tento zesilovač má maximální zesílení 10 a výkon 1,1W. Drobným rozdílem oproti SPY0030A je invertovaný vstup pro zapnutí zesilovače, tento problém jsem však vyřešil přidáním tranzistoru Q4 (Obr. 9 Původní zapojení audio výstupu, díky kterému signál pro zapnutí zesilovače invertuji.

Zesílení zesilovače LM4861 se nastavuje pomocí rezistorů *R25* a *R26*, dle datasheetu se vypočítá ze vzorce (4). Z výpočtu po dosazení vyplývá, že pro dosažení požadovaného zesílení musí být poměr odporů *R26* a *R25* měl být 2,5. Tomu přibližně odpovídají hodnoty R25=10k a R26=27k.

$$A = 2 * \frac{R26}{R25}$$
$$\frac{A}{2} = \frac{R26}{R25} \rightarrow \frac{R26}{R25} = \frac{5}{2} = 2,5 \qquad \textbf{(4)}$$

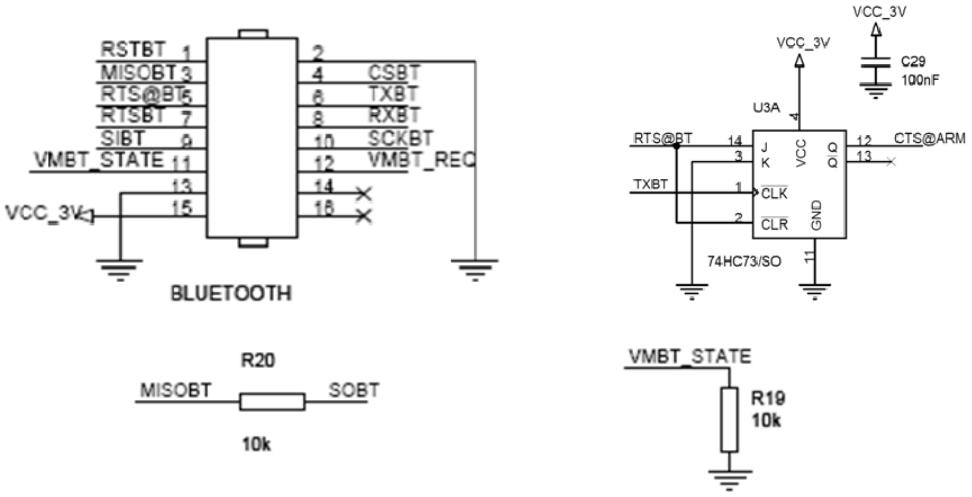


Obr. 10 Nové zapojení audio výstupu

5.2.4 Zapojení Bluetooth konektoru

Původní Bluetooth adaptér Bluecore od firmy CSR**(23)** je připojen pomocí sériového rozhraní UART, SPI a 3 vstupně výstupních signálů. Pro běžnou komunikaci je používáno rozhraní UART, rozhraní SPI je použito pouze pro možnost aktualizovat firmware v Bluetooth adaptéru.

Na zapojení konektoru pro Bluetooth adaptér (Obr. 11) jsem nic neměnil, aby bylo možné připojit stejný typ adaptéru, jako byl použit v NXT Brick.



Obr. 11 Schéma zapojení Bluetooth konektoru

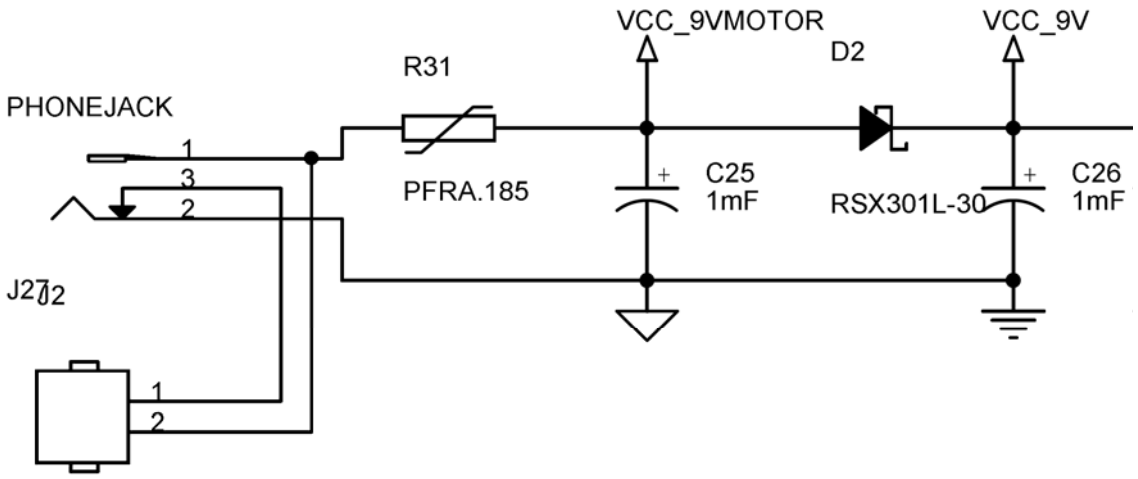
5.3 Napájení

Tato část se zabývá popisem napájecích obvodů a obvodů souvisejících s napájením, celé schéma této části laboratorního pracoviště je v příloze Příloha 2.

5.3.1 Přívod napájení

Uvnitř přístroje se vyskytují 4 typy napájení, 9V pro napájení motorů, 5V pro napájení procesoru AVR, 5V řízených procesorem AVR pro snížení spotřeby v době, kdy zařízení není používáno a 3,3V pro napájení procesoru ARM a jeho periferií rovněž řízeno procesorem AVR.

Schéma upraveného přívodu napájení 9V je zobrazeno na Obr. 12. Na přívodu napájení je umístěna vratná proudová pojistka s vybavovacím proudem 1,85A, hned za ní Shotkyho dioda s malým úbytkem, která působí jako ochrana proti přepólování. Napájení motorů je připojeno ještě před ochranou diodu, aby mohly být napájeny přímo napětím z přívodního napájení, úbytek napětí na diodě by snižoval jejich výkon.



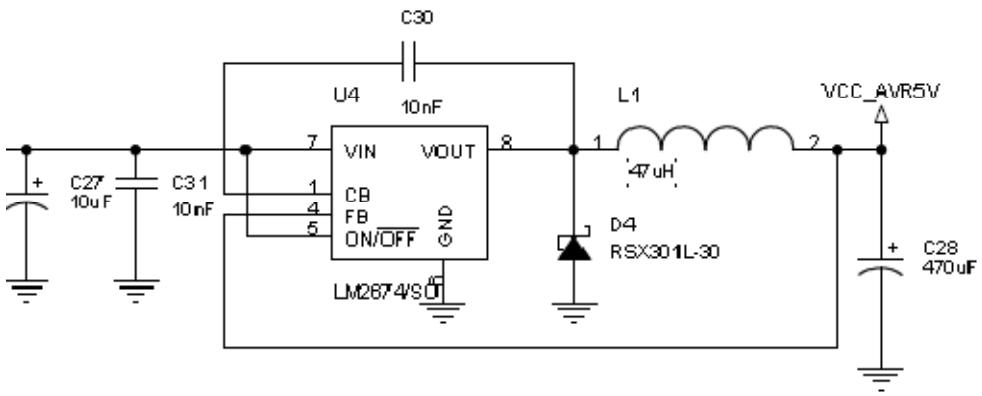
Obr. 12 Upravené schéma přívodu napájení

Oproti původnímu zapojení této části jsem přidal napájecí konektor s vnitřním kontaktem pro indikaci připojeného kabelu, který je využíván k odpojování napájení z baterií. Původně bylo zvažováno přidání nabíjecího obvodu pro baterie, ale návrh této části byl odložen na později s tím, že bude lepší nejdříve ověřit použitelnost navrhovaného pracoviště.

5.3.2 Zdroj napájení 5V

Na napájení 9V je dále připojen 5V napěťový regulátor, který dodává napájení pro celý zbytek přípravku. Původní regulátor XC9302 není dostupný a proto jsem za něj musel hledat náhradu. Dle informací v Hardware developer kit (elektronická příloha 2) protéká 5V napájením maximální proud 271mA, k tomu je třeba připočíst odběr pro periferní zařízení, což je v původním zapojení maximálně 180mA. Je tedy potřeba 5V napěťový regulátor, který je schopen dodávat minimálně 450mA.

Z dostupných napěťových regulátorů jsem vybral regulátor LM2674, který vyhovuje všem požadavkům a navíc potřebuje minimální množství externích komponent, což usnadní návrh plošného spoje. Zapojení regulátoru (Obr. 13) jsem provedl podle datasheetu, podle kterého jsem také vybral cívku 47µH.

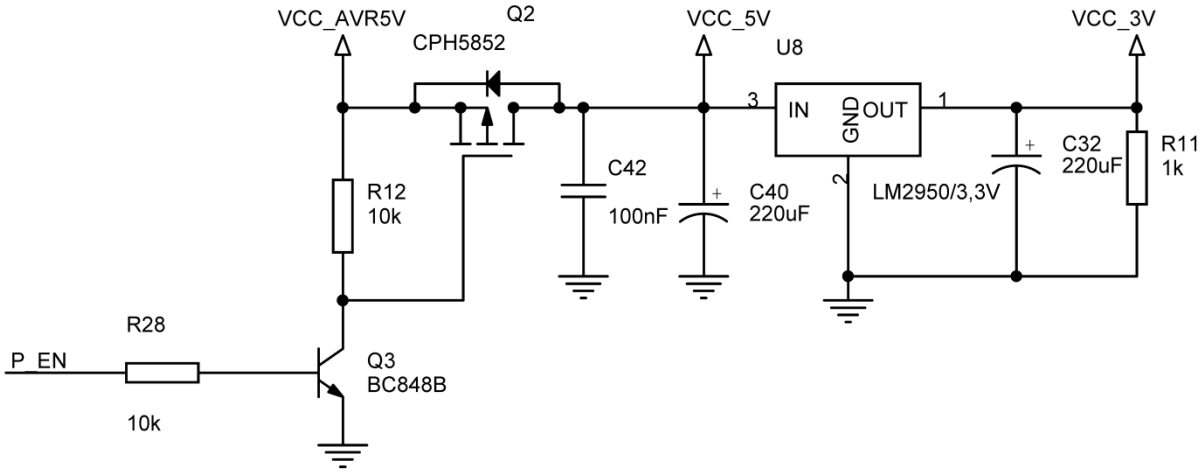


Obr. 13 Zapojení 5V napěťového regulátoru

5.3.3 Spínání napájení a zdroj 3,3V

Poslední úrovní napěťové regulace (Obr. 14) je napěťový stabilizátor 3,3V a spínání napájení řízené procesorem AVR.

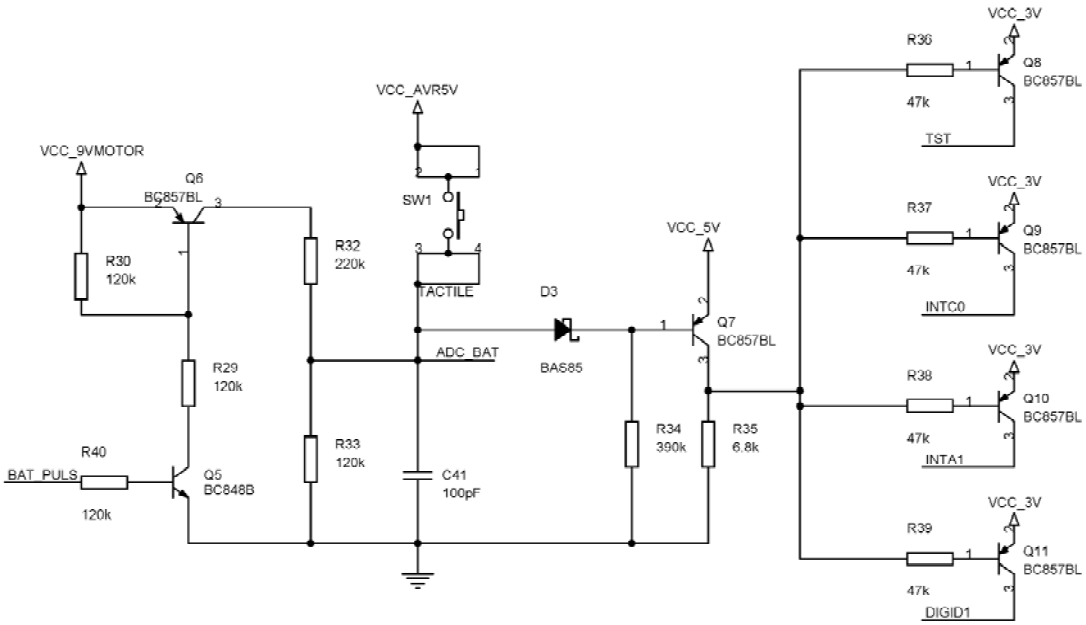
Původní nízko úbytkový napěťový regulátor XC6202P33 byl nahrazen dostupným regulátorem LM2950 s podobnými parametry. Rovněž tranzistory byly nahrazeny dostupnými typy s odpovídajícími parametry.



Obr. 14 Upravené zapojení napěťového stabilizátoru 3,3V a spínání napájecího napětí

5.3.4Měření napětí baterie

NXT Brick je také vybaveno obvodem pro měření napájecího napětí. Procesor AVR pak v případě příliš vysokého nebo nízkého napájecího napětí vypne procesor ARM spolu s většinou obvodů.



Obr. 15 Upravené schéma obvodu měření napájení a resetovacího obvodu

Samotný měřicí obvod (Obr. 15) se skládá z odporového děliče složeného z odporů *R32 a R33*, který je připojen na jeden z analogových vstupů procesoru AVR. Měřicí obvod je připojován pomocí spínacího obvodu složeného z tranzistorů *Q5* a *Q6*, spínání řídí rovněž procesor AVR. Původní zapojení v NXT Brick obsahovalo ještě spínač, který byl sepnut vždy při připojení originální nabíjecí baterie, zapojený paralelně k tranzistoru *Q5*. Spínač tedy nastálo připojil napětí baterie k měřicímu obvodu a díky tomu procesor AVR rozpoznával nabíjecí baterii. U laboratorního pracoviště všade nebude potřeba rozpoznávat přítomnost nabíjecí baterie, bude dostačující ukazatel napájecího napětí, a proto byl tento spínač v návrhu vynechán.

K obvodu pro měření napájecího napětí je přidružen obvod sloužící k uvedení procesoru ARM do režimu pro nahrávání firmwaru. K uvedení procesoru ARM do tohoto stavu je zapotřebí po dobu alespoň 10s připojit 4 piny k napájecímu napětí. Tyto piny jsou k tomuto účelu určeny výrobcem procesoru a postup je funkční vždy, když je k procesoru připojeno napájení. Ovládání tohoto režimu je umožněno pouze pomocí tlačítka SW1, které sepne celý obvod. Tlačítko SW1 je zároveň připojeno k obvodu sledování napájecího napětí, kde se jeho sepnutí projeví napětím 5V na analogovém vstupu procesoru AVR, zatímco napájecí napětí se na stejném vstupu projeví napětím do 3,2V. Procesor AVR tak může reagovat na požadavek pro vstup do režimu pro nahrávání firmwaru zapnutím speciálního napájecího režimu.

5.3.5 Proudová pojistka

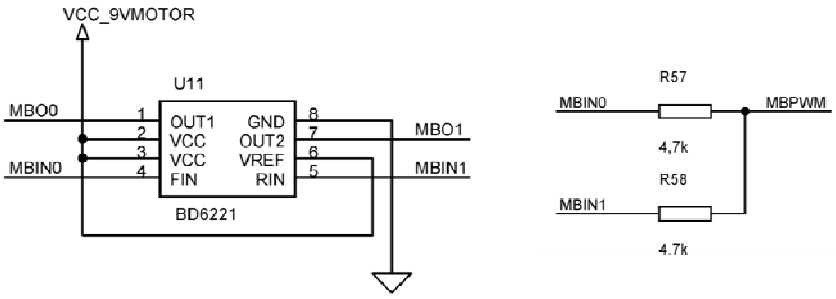
Napájení vyvedené na konektory NXT Brick je připojeno přes proudovou pojistku, která omezuje celkový odebíraný proud pro všechny zařízení připojené na napájecí napětí 4,3V na přibližně 180 mA. Vedlejším efektem je, že výstupní napětí klesá s rostoucím odběrem proudu. Výstupní napětí se dle informačních zdrojů pohybuje v rozmezí 4,8 V a 4,3V, toto napětí je však nevyhovující pro většinu běžných integrovaných obvodů s napájecím napětím 5V. Vzhledem k tomu, že se pro výuku nejvíce využívají právě integrované obvody s napájecím napětím 5V, rozhodl jsem se, že tuto proudovou pojistku v navrhovaném schématu vynechám a na konektory vyvedu přímo napájecí napětí 5V. Zvýšením napájecího napětí na konektorech nebude narušena kompatibilita s NXT Brick, neboť senzory jsou stavěny i na napájecí napětí 5V, protože toto napětí se při nezatíženém napájení může objevit i na NXT Brick.

5.4 Motory

Tato část popisuje části související s obsluhou motorů, kompletní schéma této části obvodu laboratorního pracoviště je v příloze Příloha 3.

5.4.1Ovládání motorů

Motory jsou ovládány PWM signálem z procesoru AVR, pro každý motor z něj jdou celkem 3 signály, které jsou pomocí dvou odporů (*R57* a *R58* na Obr. 16) sloučeny na dva. Z procesoru AVR je vysílán jeden PWM signál (na Obr. 16 označen MBPWM), který určuje střídu signálu, pomocí zbylých dvou signálů se určuje směr otáčení a případně i brzdný režim.



Slučování tří signálů z procesoru AVR na dva pro řízení PWM můstku probíhá násle-

dovně. Jeden ze dvou signálů pro určení směru bývá v režimu vysoké impedance a druhý je v nějaké logické úrovni. PWM signál, který je zapojen doprostřed odporového děliče, se vždy projeví na té straně, kde je pin procesoru v režimu vysoké impedance a na druhém signálu se neprojeví, neboť odpor, který je zapojen mezi signály, je mnohem větší než odpor přírodního vodiče signálu.

Výsledné chování slučování signálu je vidět v Tab. 3, kde sloupce MBIN označují logické hodnoty, jak je vysílá procesor a FIN a RIN jsou logické hodnoty tak, jak je přijme PWM můstek, v posledním sloupci je pak uvedeno chování motoru na tento stav. L znamená nízkou logickou úroveň, H vysokou úroveň, Z znamená vysokou impedanci a PWM značí průchozí PWM signál z procesoru.

Tab. 3 Tabulka slučování signálů pro ovládání motorů

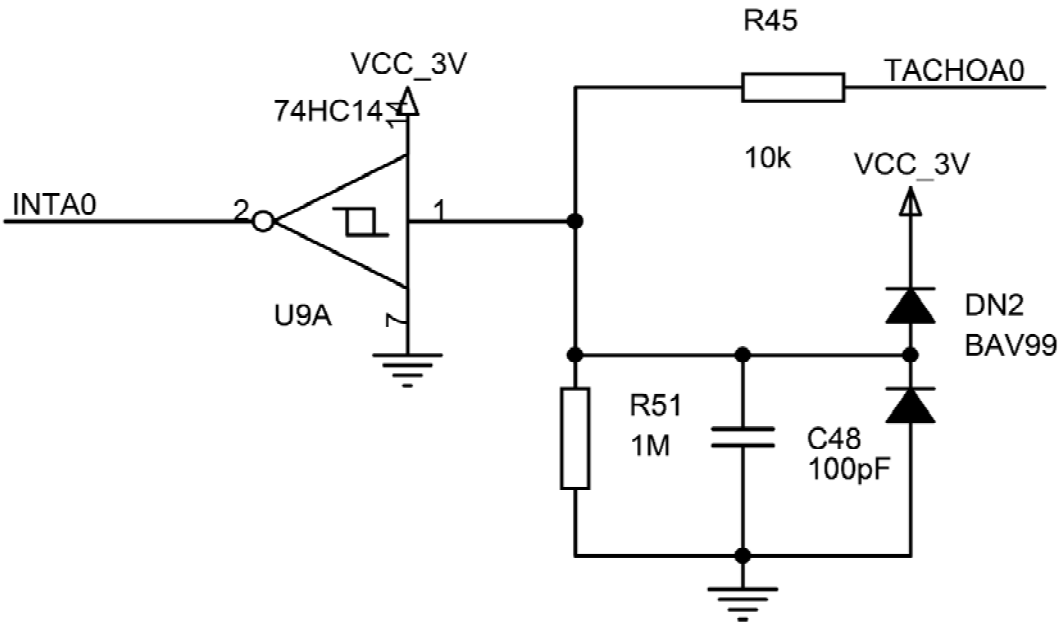
MBIN0	MBIN1	FIN	RIN	Motor
L	L	L	L	Nic
Z	L	PWM	L	PWM vpřed
L	Z	L	PWM	PWM vzad
H	H	H	H	brzda

5.4.2 Zpracování signálů z inkrementálních senzorů

Signál inkrementálních senzorů se vyhodnocuje v procesoru ARM, nejdříve je ale potřeba signál upravit a vstup ošetřit proti poškození při nesprávném zapojení. Pro úpravu signálu je použit invertující Schmitův klopný obvod, který signál z inkrementálního senzoru upraví na čistý obdélníkový, který pak procesor může spolehlivě zpracovat. Jako ošetření proti nesprávnému zapojení slouží dvojitá dioda BAV99 spolu s rezistorem o hodnotě 10kΩ, který je zapojen v série k přívodnímu pinu. Pokud by se na vstupu objevilo napětí větší než napájecí napětí klopného obvodu, tedy 3,3V, začne protékat proud diodou, která je zapojena proti napájecímu napětí, a na vstupu klopného obvodu tedy nebude vyšší napětí než napájecí napětí plus napětí diody v otevřeném směru, které v tomto případě dle datasheetu nepřesáhne 0,7V. Obdobně funguje i ochrana před záporným napětím, ochranou funkci zde však převezme dioda zapojená proti zemi.

Odpor *R45* je pro ochranou funkci diod nezbytný, právě na něm je totiž úbytek, který sníží napětí na vstupu klopného obvodu při otevření některé z diod. Schéma zapojení vstupu

pro signál jednoho inkrementálního senzoru je na Obr. 17, všechny vstupy pro inkrementální senzory jsou zapojeny stejně, pouze vstupy se signály INTA1 a INTC0 mají na výstupu klopného obvodu zapojen v sérii odpor o hodnotě 10k. Piny procesoru, na které vedou tyto dva signály, totiž zároveň slouží k uvedení procesoru do režimu pro nahrávání softwaru. Na tyto piny je při stisku tlačítka přivedena logická 1, pokud by v té době na výstupu klopného obvodu byla logická 0, znamenalo by to zkrat napájení a pravděpodobně by došlo k poškození klopného obvodu.



Obr. 17 Schéma vstupu pro inkrementální senzory

5.5 Vstupně výstupní obvody

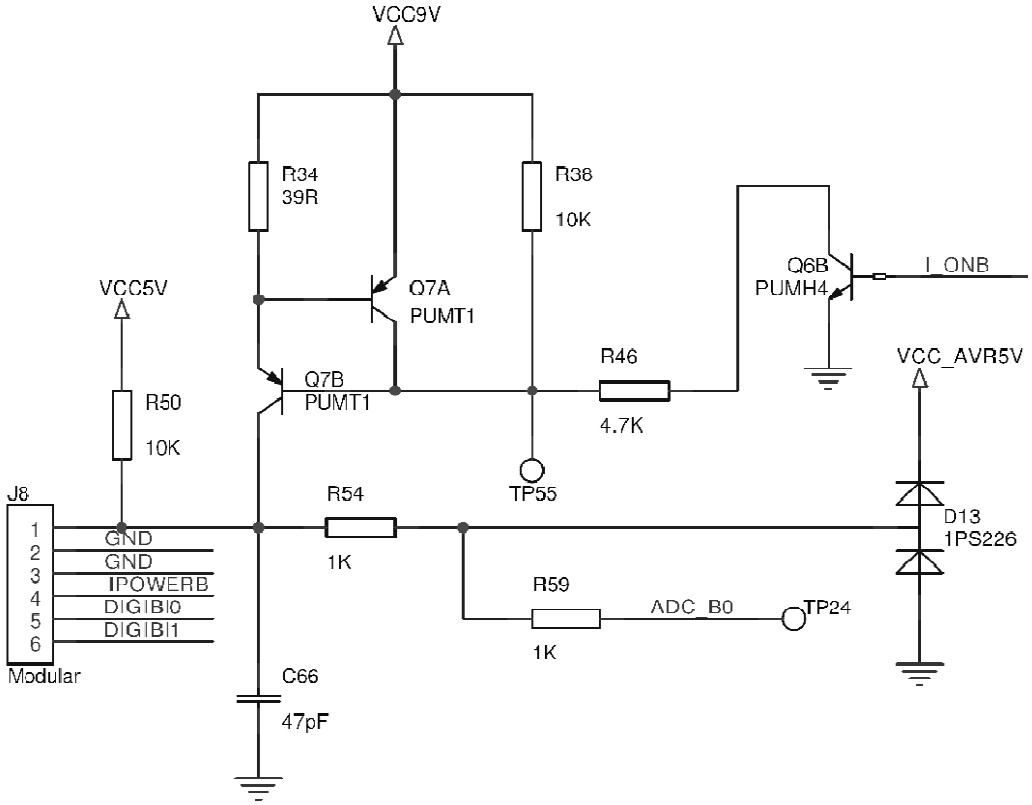
Tato část se věnuje zapojení vstupně výstupních konektorů a úpravám, které zde byly provedeny. Schéma této části laboratorního pracoviště se nachází v příloze Příloha 4.

5.5.1 Analogové vstupy

Analogové vstupy NXT Brick se vyznačují integrovaným pull-up rezistorem na napájení 5V, dále je lze přepnout do režimu proudového zdroje, ve kterém je na pin analogového vstupu v době, kdy nedochází k měření hodnoty, připojen proudový zdroj, kterým může být napájeno připojené zařízení. Proudový zdroj je zde přítomen hlavně kvůli možnosti připojení periferií Lego mindstorms RCX, což je předchůdce Lego mindstorms NXT. Vzhledem k tomu,

že kompatibilita s RCX pro přípravek není nutná a využití této vlastnosti na laboratorním pracovišti mi nepřipadalo reálné, rozhodl jsem se proudové zdroje ze všech analogových vstupů odstranit. Odstraněním proudových zdrojů se mimo jiné zjednodušilo obvodové řešení vstupních obvodů a získal jsem k dispozici 4 volné piny, které sloužily k zapínání proudových zdrojů.

Obvod (Obr. 18) analogového vstupu je dále vybaven ochranou před vysokým a záporným napětím, která funguje na stejném principu jako u vstupu pro inkrementální senzory, jen hodnota ochranného odporu je menší s ohledem na potřebu co nejmenšího úbytku napětí při měření.

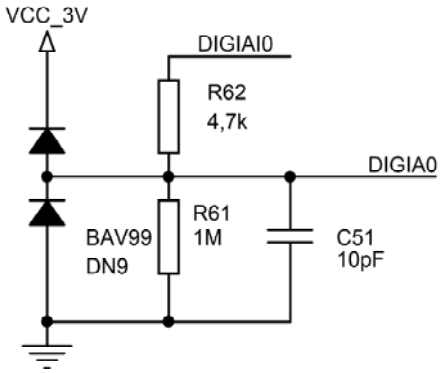


Obr. 18 Původní schéma analogového vstupu

5.5.2 Digitální vstup

Každý ze 4 vstupních konektorů obsahuje 2 digitální vstupně výstupní piny, které mohou být použity i jako I2C sběrnice. Ošetření těchto pinů je opět velice podobné ošetření ostatních pinů, tedy dvojitá dioda BAV99, která je připojena mezi napájecí napětí v závěrném směru tak, že společný vodič obou diod je připojen na vodič vedoucí k noze procesoru. Ochranný odpor je připojen mezi vstupní pin a společný pin diody, v případě přivedení nevhodného napětí tak nehrozí poškození procesoru. Při použití této ochrany je však nutno

počítat s tím, že v případě vyššího napětí je do napájecí větve (v tomto případě 3,3V) teče proud, který je nutno spotřebovat, jinak by mohlo docházet ke zhoršení kvality napájecího napětí. Vzhledem k očekávanému malému odběru napájecí větve 3,3V je na ní připojen zatěžovací odpor 1KΩ, který by měl spotřebovat přebytečnou energii. Tento částečně působí i jako zátěž napájecí větve 5V, neboť je na ni připojen stabilizátor pro napětí 3,3V.

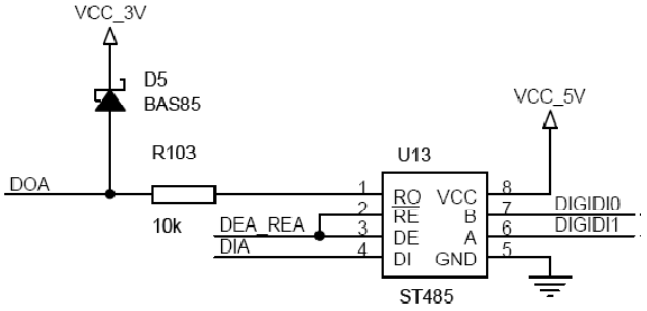


Obr. 19 Ošetření digitálních vstupů

5.5.3 RS485

NXT Brick je také vybavena sériovým rozhraním RS485, které je připojeno přímo na piny digitálních vstupů na čtvrtém vstupně výstupním konektoru. Tato sériová linka je připojena na rozhraní UART procesoru ARM přes převodník ST485(Obr. 20), který po hardwarové stránce zajišťuje splnění specifikace rozhraní RS485.

Vzhledem k tomu, že převodník ST485 je připojen přímo k digitálním pinům na konektoru, není chráněn před poškozením nevhodným napětím, jako je tomu u digitálních signálů. Toto ošetření není možné, protože by ochranné odpory zapojené v sérii s přívodními vodiči měly negativní vliv na vlastnosti sériové linky. Ochrana vstupů převodníku ST485 ale není nutná, protože jeho piny pro vnější komunikaci jsou dle datasheetu konstruovány na vstupní napětí ±14V, což je vzhledem k napětím, která budou na přípravku používána dostatečné.



Obr. 20 Zapojení převodníku ST485

5.5.4 Sběrnice I2C

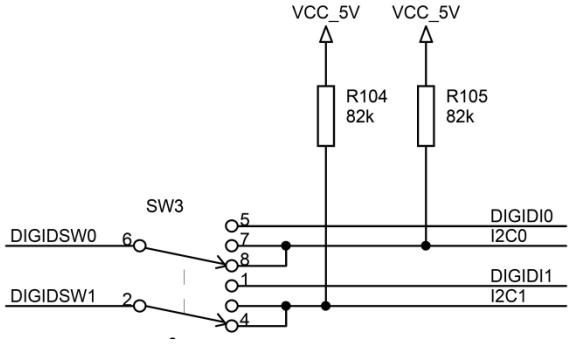
Při návrhu přípravku jsem plánoval umožnit připojení zařízení ke sběrnici I2C sběrnici na každém vstupně výstupním konektoru. Tento požadavek sice NXT Brick splňuje, ale při požití I2C sběrnice již není možné na konektorech využít digitální piny, což omezí použitelnost konektoru s I2C sběrnicí. Navíc je na každém konektoru vytvořena samostatná I2C sběrnice, což může být v některých případech výhodou, ale z hlediska výuky se tím ztratí příklad nejčastějšího použití I2C sběrnice jako prostředku pro připojení většího množství periferií. Proto jsem se začal zabývat tím, jak co nejjednodušeji vyřešit vytvoření jednotné I2C sběrnice.

Jako nejjednodušší varianta se jeví využití interní I2C sběrnice, která se používá pouze ke komunikaci mezi procesory AVR a ARM. Komunikační protokol pro komunikaci mezi těmito procesory však nepočítá s tím, že by na sběrnici komunikovala ještě další zařízení a při jeho úpravách by mohla vzniknout spousta problémů, nehledě na to, že by se musely vytvořit nebo upravit instrukce virtuálního stroje v procesoru ARM, aby takto vytvořená I2C sběrnice vůbec šla použít.

Další reálnou možností jak vytvořit I2C sběrnici, která by šla rozvést na všechny konektory, je použití digitálních pinů z jednoho vstupně výstupního konektoru pro vytvoření I2C sběrnice. Výhodou tohoto řešení oproti předchozímu je, že implementace I2C sběrnice je již ve firmwaru hotová a zbývá ji tedy pouze rozvést na všechny konektory. Nevýhodou je ztráta vstupně výstupních pinů na jednom konektoru, tato ztráta je však celkem přijatelná a mohu to vyřešit tak, abych v případě nevyužití I2C sběrnice mohl digitální vstupy využít standardním způsobem.

Pro vytvoření I2C sběrnice jsem se rozhodl použít konektor s číslem 4 hlavně proto, že na jeho digitální piny je připojeno i rozhraní RS485 a pokud odpojím digitální signály, tak na piny konektoru budou míst stále svojí funkci. Aby byla zachována plná kompatibilita s NXT Brick rozhodl jsem se ponechat možnost připojení digitálních signálů na konektor 4 tím, že jsem použil dvoupólový přepínač, který přepíná digitální signály mezi čtvrtým konektorem a sběrnicí I2C.

Sběrnici I2C jsem dále osadil pull-up rezistory, které v NXT Brick osazeny nebyly a v případě použití I2C sběrnice je bylo nutné připojit externě, nejčastěji byly osazovány v senzoru.



Obr. 21 Přepínač I2C sběrnice

5.5.5 Přidané vstupně výstupní porty

Uvolnění čtyř vstupně výstupních pinů na procesoru AVR mne přivedlo k myšlence pokusit se přidat další konektor pro připojení periferních zařízení. Při nahlédnutí do způsobu volání funkcí pro přístup ke vstupně výstupním konektorům jsem zjistil, že je možné pomocí stávajícího rozhraní ovládat i více těchto konektorů, pro jejich ovládání je však nutné do softwaru obou procesorů přidat podporu pro přístup k těmto konektorům.

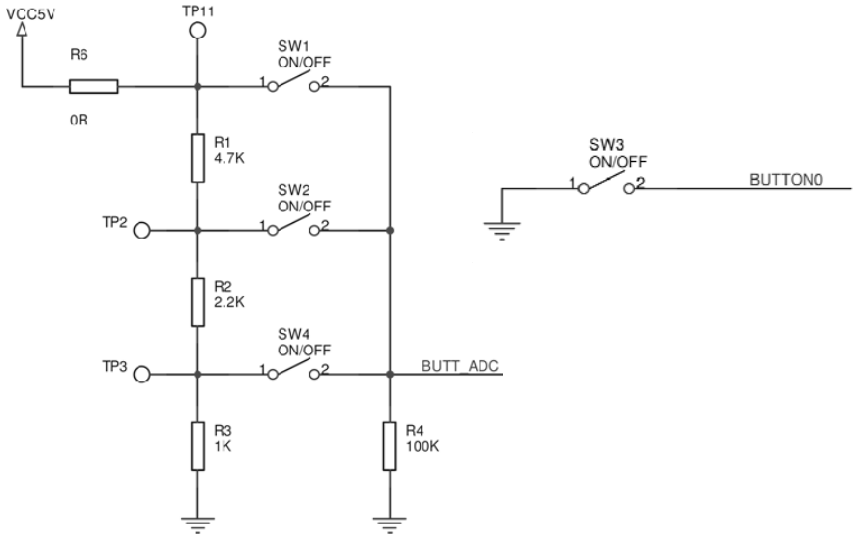
Pro dva přidané konektory jsem využil obdobné zapojení jako pro zbývající konektory, logické vstupy jsem přivedl od procesoru AVR a analogové vstupy jsem přivedl od procesoru ARM, kde byly tyto piny nevyužity. V zapojení analogových vstupů jsem na rozdíl od ostatních konektorů vynechal pull-up rezistor, aby tyto vstupy mohly být využity k přímému měření napětí a jejich rozsah mohl být zvýšen použitím externího odporového děliče.

5.6 Klávesnice a LCD displej

Vzhledem k nutnosti umístit klávesnici a displej tak, aby byly uživateli snadno přístupné, byly tyto části umístěny na samostatný plošný spoj, a tudíž budou i popsány samostatně. Kompletní schéma je v příloze Příloha 5.

5.6.1 Klávesnice

Původní klávesnice NXT Brick má celkem 4 tlačítka, šipky do stran, prostřední tlačítko ENTER a tlačítko EXIT, které slouží k návratu o úroveň zpět a ukončení běžícího programu. Tlačítko EXIT je z důvodu spolehlivého rozpoznání i při stisku více tlačítek připojeno samostatně k logickému vstupu procesoru AVR. Zbývající tři tlačítka jsou paralelně připojena k jednomu analogovému vstupu a k odporovému děliči, který přiřadí každému tlačítku unikátní napětí, podle kterého je pak tlačítko identifikováno.



Obr. 22 Zapojení klávesnice NXT Brick

Problém však nastává v případě, kdy je stisknuto více tlačítek současně, přes tlačítka totiž dojde ke zkratování odporů, které jsou zapojeny mezi stisknutými tlačítky, tím se změní poměry na odporovém děliči a tedy i výstupní napětí nebude odpovídat stisknutým tlačít-

kům. Problém lze částečně odstranit tak, abych byl schopen rozpoznat alespoň jedno ze stisknutých tlačítek. Proto musím zajistit, aby hodnota každého rezistoru v odporovém děliči byla větší než součet rezistorů, které jsou od něj směrem k zemi[11]. Přitom je třeba počítat s tolerancí rezistorů a je nutné trefit se do dostupných hodnot rezistorů. Pokud splním tyto podmínky, pak jsem vždy schopen určit sepnuté tlačítko s nejnižším napětím. Správné tlačítko pak poznám tak, že hodnota na výstupu bude větší nebo rovna hodnotě při stisku samotného tlačítka a zároveň menší než tlačítko s nejbližší vyšší hodnotou. Pro lepší představu uvedu příklad:

Pro jednoduchost použiji odpory zapojené v pořadí od země 1kΩ,2kΩ a 4kΩ. Budu uvažovat tlačítka připojená na napájecí napětí, mezi odpory 2kΩ a 4kΩ a mezi 1kΩ a 2kΩ. Při jednotlivém stisku tlačítka změřím na výstupu popořadě pro jednotlivá tlačítka napětí Ucc, 3/7 Ucc a 1/7 Ucc, kde Ucc je hodnota napájecího napětí. Pokud však stisknu obě tlačítka připojená k rezistoru 2kΩ, bude napětí na výstupu rovno 1/(1+4)=1/5=7/35 Ucc. Pokud toto napětí porovnám s napětím jednotlivých tlačítek 35/35Ucc, 15/35 a 5/35. Správně mi tedy vyšlo, že nejnižší stisknuté tlačítko je tlačítko mezi odpory 1kΩ a 2kΩ.

Pro laboratorní pracoviště jsem však plánoval použít k ovládání 4 kurzorové klávesy a jedno pro skok zpět, proto jsem se rozhodl přidat k tlačítkům z NXT Brick dvě tlačítka pro posun nahoru a dolů. Před tímto rozhodnutím jsem si nejdříve zjistil, jakým způsobem lze z programu přistupovat k tlačítkům a zda by bylo možné podobně přistupovat i k přidaným tlačítkům. Jak v instrukcích virtuálního stroje v NXT Brick, tak v programovacím jazyku NXC je přístup k tlačítkům řešen funkcemi, kterým se jako jeden z parametrů zadává index tlačítka, takže přístup k nově vytvořeným tlačítkům by měl být možný, mělo by stačit pouze ve firmwaru upravit několik funkcí pro čtení hodnoty z klávesnice a případně upravit funkce, které počítají s pevným počtem tlačítek.

Tyto tlačítka plánuji využít spíše pro ovládání programů úloh, jejich použití pro ovládání uživatelského rozhraní firmwaru považuji za neefektivní, neboť by to znamenalo pouze malé zlepšení ovladatelnosti za cenu přepisování velké části uživatelského rozhraní ve firmwaru. Obvodově jsem přidání tlačítek vyřešil pokračováním v paralelním zapojení tlačítek a

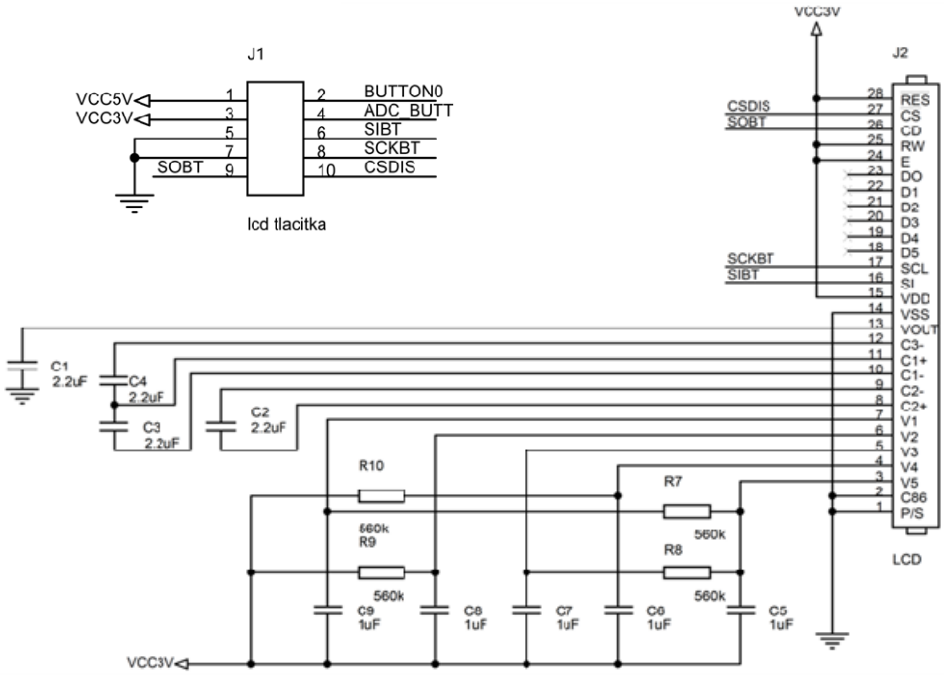
[11] Mohlo by být řešeno i obráceně, tedy od napájecího napětí

prodloužením odporového děliče o dva rezistory. Při volbě hodnot rezistorů jsem vycházel z původního zapojení a pouze jsem je doplnil hodnotami 10kΩ a 22kΩ. Tyto hodnoty splňují s dostatečnou rezervou podmínky pro čitelnost tlačítek i při vícenásobném stisku. Pro případ, že by se přidaná tlačítka neosvědčila a aby bylo umožněno ladění přípravku i v době, kdy nebudou odladěny softwarové úpravy pro přidání tlačítek, navrhl jsem zapojení tlačítek tak, aby bylo možno změnou osazení rezistorů používat i původní zapojení tlačítek. Osazení rezistorů pro verzi se čtyřmi tlačítky je patrné v příloze Příloha 5, pro verzi s přidanými tlačítky pak Příloha 6. Pro každou z možností zapojení však bude nutné používat jiný firmware, neboť nebude možné rozlišit, které zapojení je použito.

5.6.2 LCD displej

V NXT Brick je použit grafický LCD displej s rozlišením 100 * 64 pixelů, který je ovládán řadičem UC1601 přes rozhraní SPI. O tomto displeji se mi bohužel podařilo najít pouze výrobce tohoto řadiče firmu UltraChip(24), který k použitému řadiči uvádí pouze stručné informace o jeho základních vlastnostech. Při hledání displeje stejné specifikace jsem neuspěl a tak jsem se musel poohlédnout po displeji, který bych mohl použít s minimálním množstvím úprav. Nakonec jsem vybral displej Batron BTHQ 128064AVD-SRE-06-COG z katalogu firmy Farnell(22), od jejíhož distributora mám v plánu objednat hůře dostupné součástky.

Vybraný displej má rozlišení 128 x 64 a je řízen řadičem S1D10605D04B, který je schopen komunikovat po všech běžných rozhraních pro LCD displeje včetně požadovaného SPI. Rozdíl v rozlišení původního a tohoto displeje by neměl být problém. Zdá se, že některá z vývojových verzí NXT Brick již s tímto rozlišením pracovala a šířku displeje lze nastavit pomocí konstanty, takže pravděpodobně budu muset ve firmwaru pouze upravit některé příkazy pro LCD displej, které se u různých typů řadičů mohou lišit.



Obr. 23 Zapojení LCD displeje

Nový displej vyžaduje pro své napájení některé externí komponenty, ty jsem zapojil dle vzorového schématu v datascheetu displeje, celkové zapojení je zobrazeno na Obr. 23. Použitý displej, stejně jako ten původní, nevyužívá celé rozhraní SPI, je pouze schopen přes něj pouze přijímat povely a není schopen nijak odpovídat. Oproti tomu je vyžadováno použití dalšího signálu, kterým se rozlišuje, zda jsou do displeje posílána data pro zobrazení, nebo příkazy pro ovládání displeje. K tomuto účelu se používá nevyužitý signál SPI, který by měl sloužit k odesílání dat displeje. Aby to však bylo možné je třeba při používáním displeje v procesoru přepnout tento signál z SPI na vstupně výstupní, tato funkce je již ve firmwaru používána, pokud ale bude potřeba používat rozhraní SPI pro více zařízení (pomocí signálů CS[12]) je nutné ve firmwaru zajistit, aby k přepnutí došlo vždy při komunikaci s displejem a nikdy jindy.

5.7 Konektory

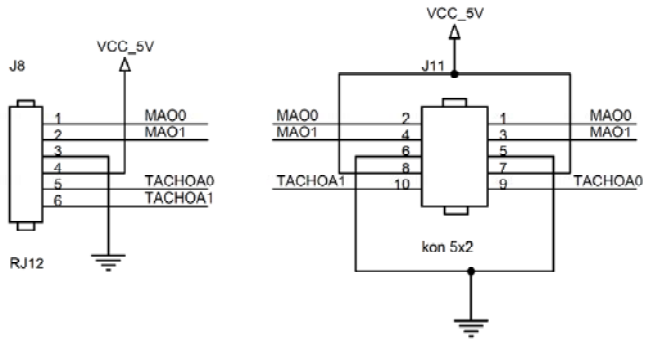
Pro připojení motorů jsem chtěl zachovat kompatibilitu s motory z mindstorms NXT, které mám v plánu pro stavbu pracoviště použít, a zároveň umožnit snadné připojení jiných

[12] Chip Select – signál po výběr aktivního čipu, bývá používán k připojení více zařízení přes SPI

zařízení, či měřicích přístrojů. Z tohoto důvodu jsem se rozhodl použít dvojici paralelně zapojených konektorů, konektor RJ-12 stejný jako na NXT Brick a desetipinový vidlicový konektor s roztečí 2,54 mm. Tento konektor umožní připojení modulů jak pomocí kabelu, tak i přímo plošného spoje modulu, který může konektor částečně využít ke svému uchycení.

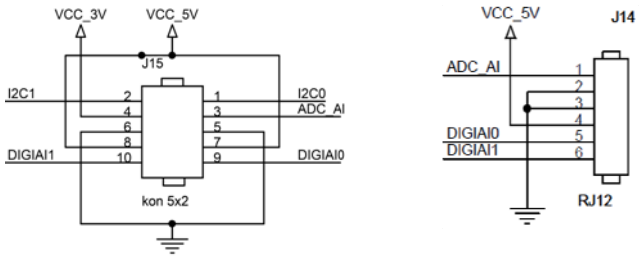
Zapojení konektorů RJ-12 jsem nechal stejné jako v NXT Brick, abych zajistil kompatibilitu obou zařízení. Při zapojení desetipinového konektoru jsem rovněž vycházel ze zapojení konektoru RJ-12, zejména s ohledem na jejich snadné propojení při návrhu plošného spoje.

Na deseti pinový konektor pro připojení motorů Obr. 24 jsem nepřidával žádné funkce, pouze jsem zdvojil piny pro řízení motorů a piny s napájecím napětím.



Obr. 24 Zapojení konektorů pro připojení motorů

Desetipinové vstupně výstupní konektory jsem oproti konektorům RJ-12 dovybavil dvěma piny s I2C sběrnicí a vzhledem k digitálním pinům s logikou 3,3V jsem přidal i napájecí napětí 3,3V, zejména proto, aby si studenti mohli vyzkoušet převod mezi logikou 5V a 3,3V.



Obr. 25 Zapojení vstupně výstupních konektorů

6 Návrh prototypu

S ohledem na ubíhající čas na dokončení práce jsem se, po dohodě s vedoucím, práce rozhodl začít navrhovat prototyp přípravku, na kterém by se zejména ověřila vhodnost návrhu pro výuku a mechanické řešení některých částí. Na prototypu bude také možné navrhovat a zkoušet úlohy, testovat úpravy firmwaru, vymýšlet vhodné rozmístění úchytů pro moduly a konektorů pro jejich připojení. V rámci této práce bude na prototypu provedeno jeho pouze oživení a ladění částí firmwaru nezbytných pro zprovoznění přípravku.

Veškeré výkresy zmíněné v této kapitole se nachází v elektronické příloze 14, kde se také nachází zdrojová data programu ORCAD, ze kterých byly výkresy generovány.

6.1 Uchycení motorů

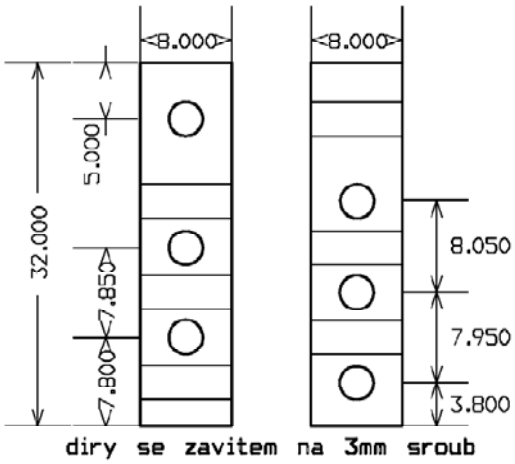
Původní myšlenku uchytit motory pomocí samostatně zakoupených dílů Lega jsem vzdal poté, co jsem zjistil, že díly, které jsou nejvhodnější k uchycení motorů, nejsou samostatně dostupné v internetovém obchodu Lega (25). Navíc první prototypové konstrukce, které se mi podařilo sestavit, nebyly příliš pevné a jejich uchycení ke zbytku přípravku by bylo obtížné. Proto jsem se rozhodl pokusit se o vytvoření jiného uchycení motorů z jiných dostupných zdrojů.

Obr. 26 Úchyt, který bude využit pro připevnění motorů

Při hledání možností uchycení jsem se zaměřil na možnost uchycení motoru za úchyt na spodní straně motoru (Obr. 26), který se zdá nejpevnější a nejsnáze využitelný. Do mezery v tomto úchytu stačí vložit konstrukční hranolek (Obr. 27), který se přišroubuje pomocí montážních otvorů Lega k motoru. Hranolek lze pak snadno připevnit ke spodní desce přípravku.

Požadavky na materiál pro výrobu hranolku jsou minimální, pevnost nemusí být příliš velká, jelikož samotné úchyty na motoru nejsou příliš pevné. Jako materiál pro výrobu tedy postačí například plast, hliník či jiný snadno obrobitelný materiál.

Pro přišroubování hranolku k motoru je potřeba tří šroubů, které přesně zapadnou do montážních otvorů lega, které mají průměr 4,7 mm a jsou hluboké 8 mm. Použití šroubu tohoto průměru nepřipadá v úvahu, jednak proto, že šrouby takového průměru se běžně nevyrábí a také proto, že po vyříznutí závitů na šrouby by v hranolku již zbylo málo místa pro šrouby připevňující jej k základně přípravku.



Obr. 27 Nákres hranolu pro připevnění motoru

Pro uchycení hranolku skrz montážní otvory Lega jsem se nakonec rozhodl použít distanční sloupky F1102SP z internetového obchodu GM electronic(9). Tyto sloupky mají udávaný průměr 4,8 mm, avšak do montážních otvorů na motorech pasují přesně. Do sloupků délky 8 mm jsem vyvrtal díry o průměru 3,2 mm, abych jimi mohl prostrčit šroub o průměru 3 mm a délce 16 mm.

Tyto distanční sloupky by také mohly být využity jako držák pro periferie NXT Brick v případě, že by některé tyto periferie byly zahrnuty do výuky.

6.2 Volba kol

Na motory ze stavebnice Lego mindstorms NXT je nejlepší využít některé z kol ze stavebnice Lego. Pro prototyp jsem zvolil pneumatiky o průměru 43,2 mm a šíři 22 mm s typovým označením 4184286 osazené na ráfkách s typovým označením 4297210. Takto sestavené kolo spojuje s motorem osička Lego délky 4 s typovým označením 370526. Veškeré tyto komponenty lze jednotlivě zakoupit v internetovém obchodě společnosti Lego(25), celková

cena těchto komponent pro oba motory je dle uvedených cen přibližně 45 Kč.

Při použití těchto kol bude mezi zemí a motory zhruba 1 cm volného místa, světlá výška robota pod motory však bude menší o tloušťku desky, na které budou motory uchyceny. Vzhledem k předpokládanému pohybu přípravku po rovném hladkém povrchu by měla bohatě stačit světlá výška 0,5 cm.

Místo předního kola jsem zvažoval použití všesměrové kuličky, například kuličku Pololu0950 z internetového obchodu Snail Instruments(5), která je pro tento účel navržena. Tyto kuličky jsou však poměrně drahé, zmíněná kulička patří mezi levnější a stojí 85 Kč, proto jsem se rozhodl pro prototyp místo kuličky vyzkoušet pouze kluznou podpěru. Jako nedostupnější a nevhodnější se mi na vyzkoušení zdál distanční sloupek *KDA6M3X10* z internetového obchodu GM electronic(9). Tento sloupek lze během testování prototypu dle získaných zkušeností upravovat nebo rovnou nahradit vhodnější komponentou, která umožní snadnější pohyb po jakémkoli předpokládaném povrchu.

6.3 Hlavní konstrukce prototypu

Pro finální verzi přípravku plánuji vytvořit konstrukci skládající se z podkladové desky, která bude pod plošným spojem a budou k ní přichyceny motory a ostatní části přípravku. Z vrchní strany bude plošný spoj chráněn plexisklem, které bude zakrývat důležitou elektroniku, ale v oblasti konektorů bude plošný spoj okryt, aby k nim byl umožněn snadný přístup.

Při návrhu prototypu jsem se však uchýlil ke zjednodušené variantě, kde je vynechána spodní nosná deska, jejíž funkci nahrazuje samotný plošný spoj. Výhodou této konstrukce je snadné sestavení, oživení a manipulace s prototypem. Po té co dojde k ujasnění požadavků na prostor pro moduly a senzory lze od prototypu odmontovat motory a prototyp přidělat na nosnou desku tak, jak je plánováno pro finální verzi.

Krycí plexisklo zároveň slouží jako úchyt pro LCD displej a klávesnici, které jsou k němu přišroubovány zespodu. Tlačítka klávesnice jsou prostrčeny skrz vyvrtané otvory v plexiskle, aby je mohl uživatel pohodlně používat a zároveň byl plošný spoj klávesnice chráněn před vnějšími vlivy. Uchycení samotného LCD displeje pod plexisklo nebylo možné, protože jde pouze o skleněnou destičku s páskovým vodičem, proto jsem pro uchycení zvolil originální držák, který je bohužel dostupný pouze v provedení pro verzi displeje s podsvícením, který je mimo jiné o 2,5 mm hlubší. Pro přesné napasování displeje do držáku a jeho uchycení pod plexisklo je tedy nutné provést následující úpravy: Zkrátit úchyty podél displeje, tak aby ne-

přesahovaly přes displej a vedle nich umístit distanční vložky, aby se displej nemohl samovolně pohybovat do stran.

Vzhledem k tomu, že držák displeje má montážní otvory o průměru 2,8 mm a na zbytku přípravku jsou použity šrouby o průměru 3 mm, rozhodl jsem se rozšířit montážní otvory držáku na 3,2 mm, abych mohl používat stejné průměry šroubů a hlavně závitníků. Montážní úchyty na horní části držáku byly odstraněny úplně, protože nebyly využity a přesahovaly by přes okraj plexiskla. Propojení hlavní desky a klávesnice jsem se rozhodl realizovat pomocí krátkého kabelu, jednak tím došlo ke zjednodušení návrhu prototypu a hlavně je možno s prototypem pracovat, i pokud je sejmuto krycí plexisklo s displejem a klávesnicí, což by v případě přímého propojení konektory nebylo možné.

Nákres plexiskla pro zakrytí plošného spoje včetně zakreslení montážních otvorů a děr pro tlačítka se nachází spolu s dalšími výkresy v elektronické příloze 14. Na horní část plexiskla prototypu počítám s osazením úchytů pro jeden menší pokusný modul a také pro několik senzorů ze stavebnice Lego mindstorms NXT.

6.4 Volba součástek prototypu

Většinu použitých součástek jsem již zvolil v průběhu obvodového návrhu, kde jsem vybíral pouze dostupné součástky. Během návrhu plošného spoje jsem vybíral pouze součástky konstrukčního charakteru, tedy především konektory.

Pro konektory kompatibilní s NXT Brick jsem se rozhodl použít standardní konektory RJ-12, které jsem nožem upravil tak, aby do nich šel zastrčit kabel s konektorem pro NXT Brick. K tomuto rozhodnutí mne vedla především finanční úspora za poštovné spojené s nákupem originálních konektorů, úprava několika konektorů pro prototyp není tolik náročná, ovšem pro finální verzi by již měly být použity originální konektory.

K připojení modulů jsem zvolil konektory typu MLW z katalogu GM electronic(9) a to zejména proto, že mají mechanickou ochranu před špatným zapojením a zároveň umožňují i propojení plošných spojů bez použití kabelu.

Jako napájecí konektory jsem použil dva piny s roztečí 2,54 mm pro připojení baterie a pro připojení napájecího adaptéru kulatý DC konektor SCD-016A, což je konektor určený k osazení na plošný spoj s průměrem středového kolíku 2,5 mm. Tento konektor vyžaduje pro osazení mimo jiné dva zářezy do plošného spoje, tyto zářezy by však znamenaly zvýšení

nákladů na výrobu plošného spoje a proto počítám s tím, že dva výstupky, které by do drážek měly zapadnout, budou při osazení odstraněny.

6.5 Návrh plošného spoje

Pro návrh plošného spoje i schématu jsem používal programové nástroje ORCAD(26), a to především proto, je katedra řídicí techniky vlastní licenci tohoto programu a také mám již s tímto nástrojem nějaké zkušenosti.

Vzhledem k tomu, že v době návrhu plošného spoje jsem neměl představu o podobě modulů, které se budou k přípravku připojovat, rozhodl jsem se proto nerozmisťovat konektory na místa, kde by měli být moduly, ale rozmístil jsem se po obvodu plošného spoje tak, aby k nim byl dobrý přístup a mohli být s modulem propojeny pomocí plochého kabelu. To mi umožnilo použít pro návrh prototypu menší plošný spoj. V průběhu návrhu jsem zjistil, že rozmístění konektorů kolem menšího plošného spoje znamená jejich nahuštění vedle sebe a při použití více z nich může být zapojení nepřehledné. Pro finální verzi přípravku tedy bude vhodnější navrhnout hlavní modul na větší plošný spoj a desetipinové konektory rozptýlit po okrajích desky tak, aby byly blízko pozicím pro moduly a měly kolem sebe dostatek místa.

Veškeré soubory s návrhem plošného spoje, osazovacími výkresy a seznamem použitých součástek se nachází v elektronické příloze 14.

7 Popis softwaru

Tato kapitola popisuje firmware NXT Brick a softwarové nástroje nezbytné pro práci s ním. Firmware je distribuován pod licencí Lego Open Source, a zdrojový kód je tedy volně ke stažení na stránkách Lego mindstorms(10). Informace o softwaru pro NXT Brick a jeho úpravách jsem čerpal ze stránek výrobců a z internetových diskusí[13], které se věnují stavebnici Lego Mindstorms NXT. Veškeré upravené zdrojové kódy uvedené v této kapitole se nacházejí v elektronické příloze 7.

7.1 Software procesoru ARM

Software pro procesor ARM obsahuje veškeré funkce NXT Brick s výjimkou řízení napájení a generování PWM signálů pro motory. Pro nahrání firmwaru do NXT Brick stačí USB kabel a software, který umí software do NXT Brick nahrát, například BricX command center, ten plánuji využít pro práci s pracovištěm, nebo NXT-G dodávané k NXT Brick. Firmware pro NXT Brick bývá nejčastěji ve formátu s příponou *rfw*, v něm je formát nejčastěji distribuován, případně s příponou *a79*, Což je formát, který nejčastěji generují kompilátory. BricXcc umí nahrát firmware v obou těchto formátech.

7.1.1 Vývojové prostředí

Pro kompilaci a editaci firmwaru pro procesor ARM lze používat dva nástroje. Prvním je souprava nástrojů NXTGCC (27). Tato sestava obsahuje vývojové prostředí Eclipse pro programovací jazyky C a assembler, kompilátor GCC a nástroje usnadňující práci s firmwarem NXT Brick. Pro přeložení originálního firmwaru se však musí nejdříve nutno spustit program, který upraví zdrojový kód tak, aby jej byl kompilátor GCC mohl přeložit. Pro samotný překlad je v NXTGCC připraven další program, který překlad řídí a zajistí, že výstupem překladu je rovnou formát, který lze nahrát do NXT Brick například pomocí vývojového prostředí BricX command center (21). Veškeré funkce nutné pro úpravu a překlad firmwaru jsou spustitelné přímo z prostředí Eclipse.

[13] Konkrétně (27) a (33)

Nevýhodou tohoto řešení se jeví jeho menší efektivita, podle stránek projektu (27) i podle mých pokusů je velikost přeloženého firmwaru přibližně o 50% větší než firmware, který je přeložen nástroji IAR. NXTGCC bylo používáno zejména v době, kdy nebyla dostupná žádná volná verze nástrojů umožňující překlad firmwaru NXT Brick. Dle informací, které jsem našel na zmíněných internetových fórech věnující se programování NXT Brick, již vývoj tohoto nástroje dále nepokračuje.

Druhou možností, jak pracovat s firmwarem NXT Brick, je použít vývojové prostředí IAR Embedded Workbench. Toto vývojové prostředí je sice placené, ale jako reakce na oblibu NXT Brick byla vytvořena verze IAR Embedded Workbench for LEGO MINDSTORMS NXT(28), upravená speciálně jen pro firmware NXT Brick a poskytované zdarma. Na plné verzi tohoto vývojového prostředí byl také firmware NXT Brick vyvíjen. Z těchto důvodů jsem se rozhodl pro práci s firmwarem procesoru ARM využívat právě toto vývojové prostředí.

Instalační soubor tohoto vývojového prostředí lze po vyplnění kontaktních údajů stáhnout ze stránky (28) a je také umístěno v elektronické příloze (8). Samotná instalace probíhá jednoduše, jen se musí vyplnit licenční čísla, která budou doručena na e-mail vyplněný před stažením instalace.

Po instalaci lze otevřít firmware, který již obsahuje soubory projektu pro toto vývojové prostředí. Projekt se otevře spuštěním souboru *LMS_ARM.eww*, který se nachází ve složce *\SAM7S256\Tools*. Firmware doporučuji stáhnout ze stránek IAR(28), protože ve verzi na stránkách Lego mindstorms(10) chybí několik souborů nutných pro překlad. Tato verze je také umístěna v elektronické příloze 5.

Pro přeložení projektu tak, aby šel do NXT Brick nahrát pomocí prostředí BricXcc nebo jiného nástroje určeného pro NXT Brick, je nutné v okně *workspace* přepnout nastavení projektu na *Bin Output.* Poté lze již projekt přeložit příkazem *Project->Make*. Přeložený firmware se pak uloží do souboru *\SAM7S256\Object\LMS_ARM.a79*.

7.1.2 Struktura projektu

Zdrojové soubory projektu se nachází v adresáři *\source*, ty jsou rozděleny na jednotlivé moduly, jejichž funkce většinou vyplívá již z názvu. Případně je funkce jednotlivých modulů popsána v dokumentu (29).

Každý modul se skládá z několika souborů, ty lze rozdělit na několik skupin. Základní rozdělení je na fyzickou vrstvu (d_ před názvem) a abstraktní vrstvu (předpona c_). Z tohoto

označení se vymyká modul *m_sched*. Ten se spouští ve smyčce metody *main* a stará se o spouštění periodické obsluhy ostatních modulů.

Zdrojové soubory se dále dělí podle přípon. Soubory s příponami *c* a *h* jsou dle zvyklostí zdrojové soubory jazyka C. Soubory s příponou iom patří zpravidla k modulům abstraktní vrstvy a obsahují definice typů a konstanty modulů, které potřebují i jiné moduly. Přípona *r* přísluší naopak k modulům fyzické vrstvy a v těchto souborech jsou definované funkce pro nastavení a ovládání periferií procesoru. Přípony *rms* a txt mají soubory obsahující data pro uživatelské rozhraní.

7.1.3 Úprava ovládání displeje

Protože jsem použil jiný typ displeje, než který je použit v NXT Brick, musím upravit jeho ovládání i v softwaru. Veškeré odesílání příkazů pro displej se provádí přes funkce v souboru *d_display.r*, z pohledu příkazů pro displej jsou v něm důležité zejména konstanta *ACTUAL_WIDTH*, pole *DisplayInitString* a *DisplayLineString*. Ostatní funkce nezávisí na typu displeje, nebo využívají SPI rozhraní, které je pro oba displeje stejné.

V poli *DisplayLineString* se ukládají povely pro změnu aktuálně adresovaného řádku. Dle datasheetu displeje jsem zjistil, že příkazy v původním datasheetu se shodují a tudíž je nemusím měnit. Další konstantou je pole *DisplayInitString*, hodnoty z tohoto pole jsou zasílány při inicializaci displeje. Dle datasheetu k displeji a jeho řadiči[14] a komentářů u jednotlivých příkazů ve zdrojovém kódu jsem sestavoval příkazy pro nový displej. V průběhu hledání správných hodnot pro příkazy jsem zjistil, že příkazy pro displej, který byl pravděpodobně použit v některém z vývojových prototypů NXT Brick, jsou shodné, tudíž jsem příkazy pouze překontroloval a použil je. Poslední *ACTUAL_WIDTH* určuje šířku displeje v pixelech, tuto konstantu jsem tedy nastavil na 128, což je šířka nového displeje, výška displeje se s původním shoduje.

[14] Oba datasheety jsou umístěny v elektronické příloze 13 (displej a s1d10605)

7.1.4 Plánované úpravy

V této podkapitole popíši mé poznatky pro softwarové změny, na které jsem prototyp hardwarově připravil, ale k jejich realizaci již nebyl čas. Tato část by měla sloužit především pro pokračování vývoje laboratorního pracoviště.

Přidání tlačítek

Pro přidání tlačítek pro pohyb nahoru a dolů bude nejdříve potřeba tyto tlačítka dodefinovat. Přidělení indexu tlačítkům a zároveň nastavení jejich počtu se provede přidáním hodnot do enumeračního typu v souboru *c_buttom.iom.* Tím by mělo dojít k přidání kompletní podpory pro správu těchto tlačítek, aby bylo zajištěno také jejich správné rozpoznání, bude zapotřebí upravit jejich čtení z analogové hodnoty přijaté z procesoru ARM. Kontrola stavu tlačítek se provádí ve funkci *BUTTONRead* v souboru *d_button.r*. Zde je zapotřebí upravit úrovně pro rozpoznání stávajících tlačítek a přidat rozpoznání nových tlačítek. Informace o přidaných tlačítkách obsahuje kapitola 5.6.1.

Podpory pro přidané konektory

Implementace podpory dvou přidaných konektorů bude vyžadovat rozsáhlejší úpravy softwaru v obou procesorech. Aby bylo možné plně ovládat digitální vstupy připojené k procesoru ARM, bude nutné upravit struktury *IOTOAVR* a *IOFROMAVR* v souboru *m_sched.h*, které slouží pro komunikaci s procesorem AVR. Stejně je třeba upravit tyto struktury v procesoru AVR. V souboru *m_sched.h* bude také potřeba změnit konstantu *NOS_OF_AVR_INPUTS*, podle níž se kontroluje platnost indexu při operacích s porty a vytvářejí se datové struktury pro obsluhu portů. Funkce pro obsluhu operací s porty v souboru *d_input.r* bude nutno upravit tak, aby pro přidané porty braly hodnoty ze správného umístění.

7.2 Software procesoru AVR

Procesor AVR se stará o správu režimu napájení pro procesor ARM a celý zbytek přípravku. Pro nahrání firmwaru je zapotřebí ISP programátor podporující procesor ATmega48 od společnosti Atmel(17) a příslušný software, který umí pomocí tohoto programátoru software nahrát. Programátor se k přípravku připojuje na konektor J6, jehož zapojení je zobrazeno ve schématu procesorové části v příloze Příloha 1.

7.2.1 Vývojové prostředí

V dokumentu přiloženém k firmwaru NXT Brick z internetových stránek Lego mindstorms(10) je napsáno, že k přeložení firmwaru pro procesor AVR bylo použito vývojové prostředí AVR Studio 4(30) od firmy Atmel. Při pokusu o přeložení v tomto vývojovém prostředí jsem však zjistil, že firmware vyžaduje použití některého kompilátoru icca90 nebo iccavr od firmy IAR, které AVR Studio neobsahuje. Proto jsem se rozhodl použít opět vývojové studio IAR Embedded Workbench, tentokrát ve verzi kickstart pro procesory AVR. Tato verze je zdarma s omezením velikosti zdrojového kódu na 4kB, což vzhledem k velikosti flash paměti procesoru stačí.

Vývojové prostředí je podobně jako u verze pro ARM možné stáhnout ze stránek IAR(31)[15], viz elektronická příloha 9. Před stažením se opět musí vyplnit registrační údaje. Instalace probíhá stejným způsobem jako verze pro procesor ARM, je tedy nutné vyplnění licenčních údajů přijatých e-mailem.

Firmware pro procesor AVR ze stránek Lego mindstorms(10) obsahuje pouze zdrojové soubory, nikoliv soubory s projektem pro vývojové prostředí, ty je třeba teprve vytvořit. Ve vývojovém prostředí IAR workbench pro AVR zvolíme *Project->Create New Project*, v dialogovém okně vybereme *Empty project*, potvrdíme a vybereme místo, kde se má projekt vytvořit, nejlépe do složky, kde máme zkopírovaný firmware pro procesor AVR. Dále vybereme Project->Add Files a přidáme všechny soubory firmwaru pro AVR, které se nachází ve složce \Source\. Tímto jsme vytvořili projekt obsahující všechny potřebné soubory, abychom ho mohli přeložit, musíme ho ještě nastavit. Nejdříve v okně workspace změníme nastavení z *Debug* na *Release*. Následně přes *Project->Options* otevřeme nastavení projektu a

[15] Stáhnout verzi pro osmibitový procesor AVR

v *General options-Target* nastavíme v *Processor configuration* správný typ procesoru tedy *„cpu=m48, ATmega48“.* Dále v *Linker->Output* nastavíme takový formát výstupního souboru, který podporuje použitý programátor procesoru AVR. Tím je nastavení projektu hotové, můžeme ho pro vyzkoušení přeložit a začít provádět úpravy.

7.2.2 Struktura zdrojových souborů

Struktura zdrojových souborů firmwaru procesoru AVR se podobá té v procesoru ARM, jen je jednodušší. Většina modulů pracuje na fyzické vrstvě a mají předponu *d_*, nejvíce kódu těchto modulů se navíc soustřeďuje v souborech pro ovládání periferií procesoru s příponou *r.* V abstraktní vrstvě pracuje pouze modul *c_armcomm*, tento modul je volán periodicky a obstarává sběr informací pro procesor ARM a jejich odesílání. Modul *m_sched* je opět používán přímo metodou *main* a jeho funkce jsou pouze přesměrovány na funkce z modulu *c_armcomm*.

7.2.3 Provedené úpravy

Protože jsem z přípravku odstranil proudové zdroje, které byly řízeny z procesoru AVR, odstranil jsem zároveň i funkce pro jejich ovládání. V souboru *d_input.r* byly smazány veškeré funkce, které sloužily pouze k nastavení proudových zdrojů, a rovněž jsem odstranil všechna volání těchto funkcí. V této souvislosti jsem také zrušil funkci *dInputDeselect* ze souboru *d_input.c*, která sloužila pouze k ovládání proudových zdrojů, a také všechny její výskyty v souboru *c_armcomm.c*.

7.2.4 Plánované úpravy

Úpravy související se změnami hardwaru, které mohou být v budoucnu použity, nebudou mít na procesoru AVR takový rozsah jako na procesoru ARM. Vzhledem k tomu, že hodnoty ze vstupu klávesnice se posílají v nezměněné podobě do procesoru ARM, není potřeba pro přidání tlačítek jakkoli měnit software procesoru AVR.

Pro přidání nových portů bude v první řadě potřeba upravit struktury *IOFROMAVR* a *IOTOAVR* v souboru *m_sched.h* stejně jako v procesoru ARM. Dále se musí do modulu d_input přidat funkce, které budou nastavovat stav přidaných digitálních pinů dle struktury *IOTOAVR* a naopak stav pinů doplňovat do struktury *IOFROMAVR.* Tyto funkce by měly být spouštěny ze stavového automatu v modulu c_armcomm.

7.3 BricX Command Center

Při návrhu změn v zapojení přípravku jsem počítal především s programováním v jazyce NXC, jenž je součástí vývojového prostředí BriXcc. Programovací jazyk NXC nabízí několik výhod např. jeho oblíbenost v komunitě kolem mindstorms NXT a jeho široký rozsah funkcí, které pohodlně zapouzdřují práci s některými prostředky, ale také umožňují práci s NXT Brick na nejnižší úrovni. Výhodný je také fakt, že jak překladač NXC tak i vývojové prostředí BricXcc jsou zdarma.

Vývojové prostředí BricXcc podporuje kromě jazyka NXC také další programovací jazyky, a to i pro jiné programovatelné moduly než NXT Brick. Vedle funkcí sloužící pro samotné programování umí má BricXcc také schopnost komunikovat s NXT Brick[16] a provádět s ním různé operace jako nahrávání firmwaru, správa souborů uložených v NXT Brick a sledování různých parametrů.

7.3.1 Instalace

Instalaci BricX Command Center lze stáhnout ze stránek projektu(32), instalace zároveň umístěna v elektronické příloze 10. Samotná instalace pak vyžaduje zadání složky, kam se má BricXcc nainstalovat. Aby mohlo BricXcc komunikovat s NXT Brick, a v našem případě i s laboratorním pracovištěm, je nutné ještě doinstalovat USB ovladač pro NXT Brick, který lze získat na stránkách Lego mindstorms(10) nebo v elektronické příloze 11.

7.3.2 Používání

Po spuštění BricX Command Center se objeví dialog k jakému zařízení se připojit, vybereme rozhraní USB, typ NXT, standardní firmware a potvrdíme. Pokud zařízení není připojeno, zobrazí se informace o tom, že některé funkce BricXcc nebudou bez připojeného zařízení dostupné a spustí se hlavní okno vývojového prostředí.

Před zahájením programování v jazyce NXC je dobré nejprve zkontrolovat nastavení správného programovacího jazyka. Volbou *Edit->Preferences* otevřeme okno s nastavením a na záložce *Compile*r zvolíme volbu NXC.

Nyní již můžeme programovat. Programovací jazyk NXC vychází z jazyka C a snaží se

[16] Kromě toho i s jinými zařízeními

dodržovat jeho syntaxi, jde ovšem o zjednodušenou verzi a některé prvky jazyku C neodpovídají. Podrobné informace lze získat na stránce NXC(33), kde se nachází dokumentace, návody pro začátečníky a vzorové kódy.

Protože překladač NXC kontroluje u některých funkcí rozsah zadaných indexů, je nutné přistupovat k přidaným portům a tlačítkům jiným způsobem. Asi nejspolehlivějším způsobem jak přistupovat k funkcím NXT Brick, aniž by překladač jakkoli kontroloval zadané parametry, je použití takzvaných nízko úrovňových systémových funkcí[17], pro čtení tlačítek a ovládání portů jde konkrétně o funkce SysReadButton respektive SysIOMapRead/Write. Popis těchto funkcí se nachází v dokumentaci NXC (elektronická příloha 12).

[17] **Low-level System Functions**

8 Závěr

Po analýze různých možností realizace laboratorního pracoviště pro výuku předmětu Aplikovaná elektronika jsem uznal za nejvhodnější využít k návrhu základ procesorového modulu NXT Brick ze stavebnice Lego mindstorms NXT, která se již na katedře řídicí techniky k výuce využívá. Toto řešení mi ušetřilo spoustu práce zejména při vytváření softwaru, jelikož jsem využil OpenSourcový firmware pro NXT Brick a další nástroje určené k ovládání a programování tohoto modulu. Firmware jsem částečně upravil tak, aby bylo možno přípravek oživit a vyzkoušet základní funkce. Úpravy firmwaru pro zprovoznění přidaných obvodových funkcí, by vyžadovaly ladění přímo na přípravku, k čemuž zatím nebyla příležitost, a proto jsem postup úprav popsal v kapitole 7. Práce také doplněna o informace a nápady, které jsem v průběhu návrhu pracoviště nasbíral, ale již jsem je neměl možnost využít.

Výsledkem této práce není hotové laboratorní pracoviště, což ani nebylo očekáváno, nýbrž návrh jeho prototyp, na kterém je možné ověřit vhodnost návrhu pro výuku a zároveň na něm lze zkoušet další úpravy pro jeho zdokonalení jako příprava pro výrobu finální verze laboratorního pracoviště.

Bohužel zatím nebyl tento prototyp nijak vyzkoušen, neboť došlo k třítýdennímu zpoždění při výrobě desky plošných spojů zaviněným ze strany výrobce. Z tohoto důvodu ještě nemám v okamžiku odevzdání této práce k dispozici hotový plošný spoj, který bych mohl osadit a vyzkoušet. Očekávám však, že s oživením prototypu by neměl být problém a pokud nedojde k dalšímu zpoždění výroby plošného spoje, měl bych být schopen prototyp oživit a vyzkoušet do obhajoby bakalářské práce. O oživení a vyzkoušení prototypu bude vypracována zpráva obsahující shrnující postup sestavení prototypu a shrnutí jeho vlastností. Zpráva bude poté přidána k bakalářské práci jako příloha.

Použitá literatura:

1. Popis předmětu Aplikovaná elektronika. *Studijní plán oboru Kybernetika robotika.* [Online] http://www.fel.cvut.cz/education/bk/predmety/12/53/p12538104.html.

2. **Hrouda, Michal.** *Robotika a Elektronika Mica.* [Online] mica.ic.cz.

3. *Technobots shop.* [Online] www.technobots.co.uk.

4. *Active robots.* [Online] www.active-robots.com.

5. *Snail instruments.* [Online] shop.snailinstruments.com.

6. Mobilní robot pro výuku navigačních systémů a strojového vnímání. *Katedra kybernetiky FAV, ZČU.* [Online] http://www.kky.zcu.cz/cs/research-fields/mobile-robot.

7. *Industrial Wireless products.* [Online] ConnectBlue. www.connectblue.com.

8. **Novák, Petr.** *Mobilní roboty.* Praha : BEN, 2005.

9. *GM electronic.* [Online] www.gme.cz.

10. *LEGO.com MINDSTORMS.* [Online] mindstorms.lego.com.

11. *Stránka předmětu Roboti.* [Online] http://support.dce.felk.cvut.cz/roboti/.

12. *mindsensors.com.* [Online] www.mindsensors.com.

13. *Lego NXT hacks a and robots.* [Online] http://www.philohome.com/nxt.htm.

14. *Color senzor.* [Online] http://www.philohome.com/sensors/colorsensor.htm.

15. Cadmium-Sulfide Color Sensor. [Online] http://www.robotroom.com/ColorSensor.html.

16. **Gasperi, Michael, Hurbain, Philippe E. a Hurbain, Isabelle L.** *Extreme NXT: Extending the LEGO Mindstorms NXT to the Next Level.* Berkeley : APRESS, 2007.

17. *Atmel corporation.* [Online] www.atmel.com.

18. *HiTechnic Products.* [Online] www.hitechnic.com.

19. NXT Programing software. [Online] http://www.teamhassenplug.org/NXT/NXTSoftware.html.

20. Lego Mindstorms NXT. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Lego_Mindstorms_NXT#Programming.

21. *Bricx Command Center.* [Online] http://bricxcc.sourceforge.net/.

22. *Farnell.* [Online] http://cz.farnell.com/.

23. *CSR.* [Online] http://www.csr.com/products/overview.htm.

24. *UltraChip LCD ID driver design.* [Online] http://www.ultrachip.com/.

25. *Lego shop.* [Online] http://shop.lego.com/.

26. *Cadence OrCAD solutions.* [Online] http://www.cadence.com/products/orcad/pages/.

27. *NXTGCC.* [Online] http://nxtgcc.sourceforge.net/.

28. *IAR Embedded Workbench for LEGO MINDSTORMS NXT.* [Online] http://www.iar.com/mindstorms.

29. *Mindstorms Tutorial.* [Online] http://nxtgcc.sourceforge.net/EMSOFT-2009-Mindstorms-Tutorial.pdf.

30. *AVR studio 4.* [Online] http://www.atmel.com/avrstudio.

31. *IAR Download evaluation software.* [Online] http://www.iar.com/website1/1.0.1.0/675/1/.

32. *BricX Command Center.* [Online] http://bricxcc.sourceforge.net/.

33. *Not eXactly C.* [Online] http://bricxcc.sourceforge.net/nbc/index.html.

34. *NXT Firmware Hacking.* [Online] http://news.lugnet.com/robotics/nxt/nxthacking/.

35. *nxtasy.org forums.* [Online] http://forums.nxtasy.org/.

Elektronické přílohy:

Elektronické přílohy jsou umístěny na přiloženém CD.

1. **Hardware schematic** /LEGO MINDSTORMS NXT/LEGO MINDSTORMS NXT Hardware Developer Kit/Appendix 1-LEGO MINDSTORMS NXT hardware schematic.pdf
2. **Hardware developer kit** /LEGO MINDSTORMS NXT/LEGO MINDSTORMS NXT Hardware Developer Kit/LEGO MINDSTORMS NXT Hardware Developer Kit.pdf
3. **Keypad schematic** /LEGO MINDSTORMS NXT/LEGO MINDSTORMS NXT Hardware Developer Kit/LEGO MINDSTORMS NXT Keypad schematic.pdf
4. **NXT executable file spec** /LEGO MINDSTORMS NXT /LEGO NXT executable file specifications.pdf
5. **Firmware ARM** /LEGO MINDSTORMS NXT/LEGO MINDSTORMS NXT Firmware Open Source - 129/AT91SAM7S256/
6. **Firmware AVR** /LEGO MINDSTORMS NXT/LEGO MINDSTORMS NXT Firmware Open Source - 129/ATmega48/
7. **Upravený firmware** /firmware/
8. **IAR workbench ARM** /programy/IAR workbench ARM/
9. **IAR workbench AVR** /programy/IAR workbench AVR/
10. **BricXcc** /programy/bricxcc/
11. **NXT Brick driver** /programy/MINDSTORMS NXT Driver v1.1.3/
12. **NXC Guide** /NXC_Guide.pdf
13. **Výběr použitých datasheetů** /datasheets/
14. **Výkresy a jejich zdrojová data** /vykresy/

Přílohy:

Příloha 1: **Schéma zapojení hlavní desky – procesory**

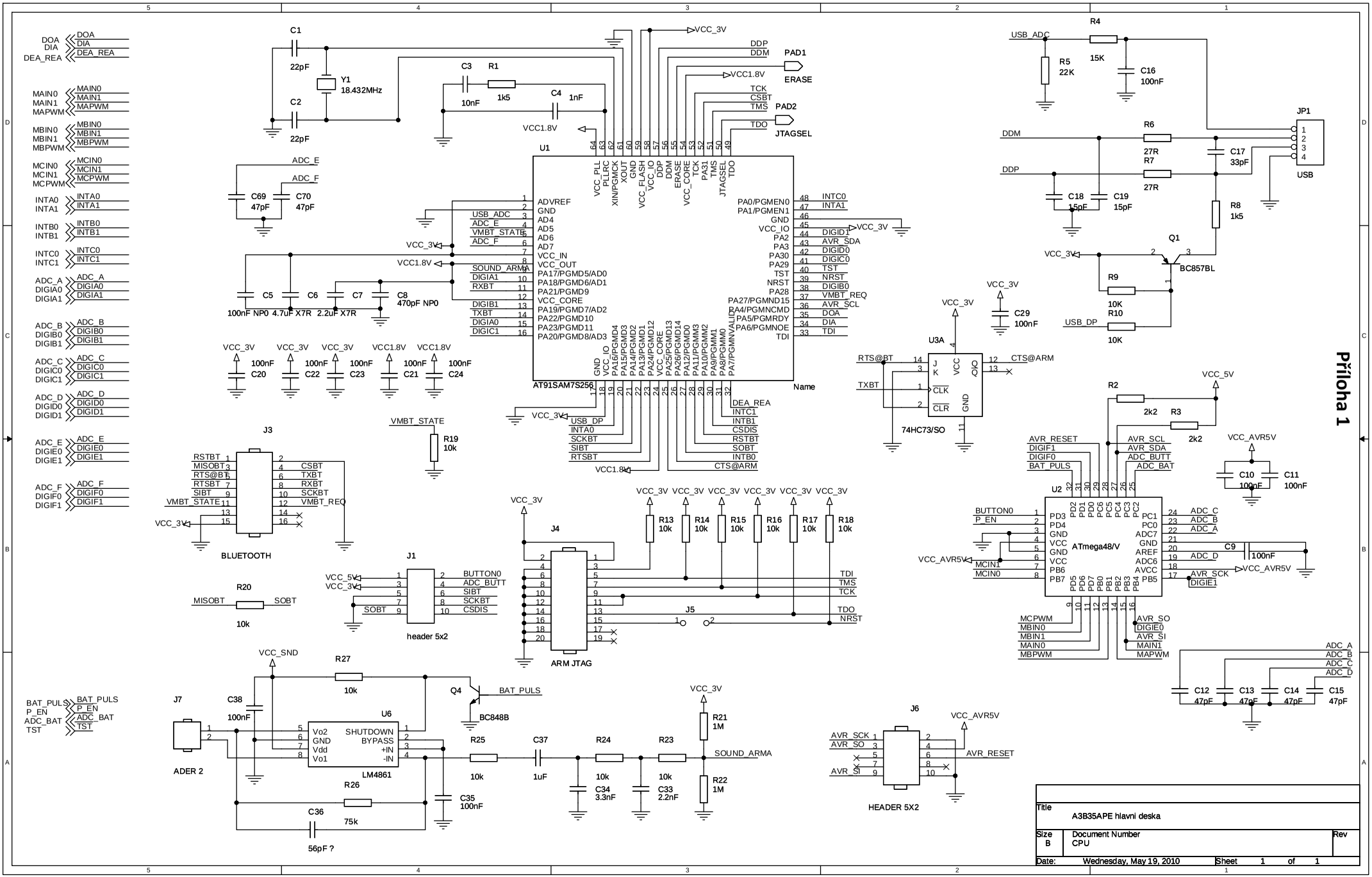
Příloha 2: **Schéma zapojení hlavní desky – napájení**

Příloha 3: **Schéma zapojení hlavní desky – výstupy**

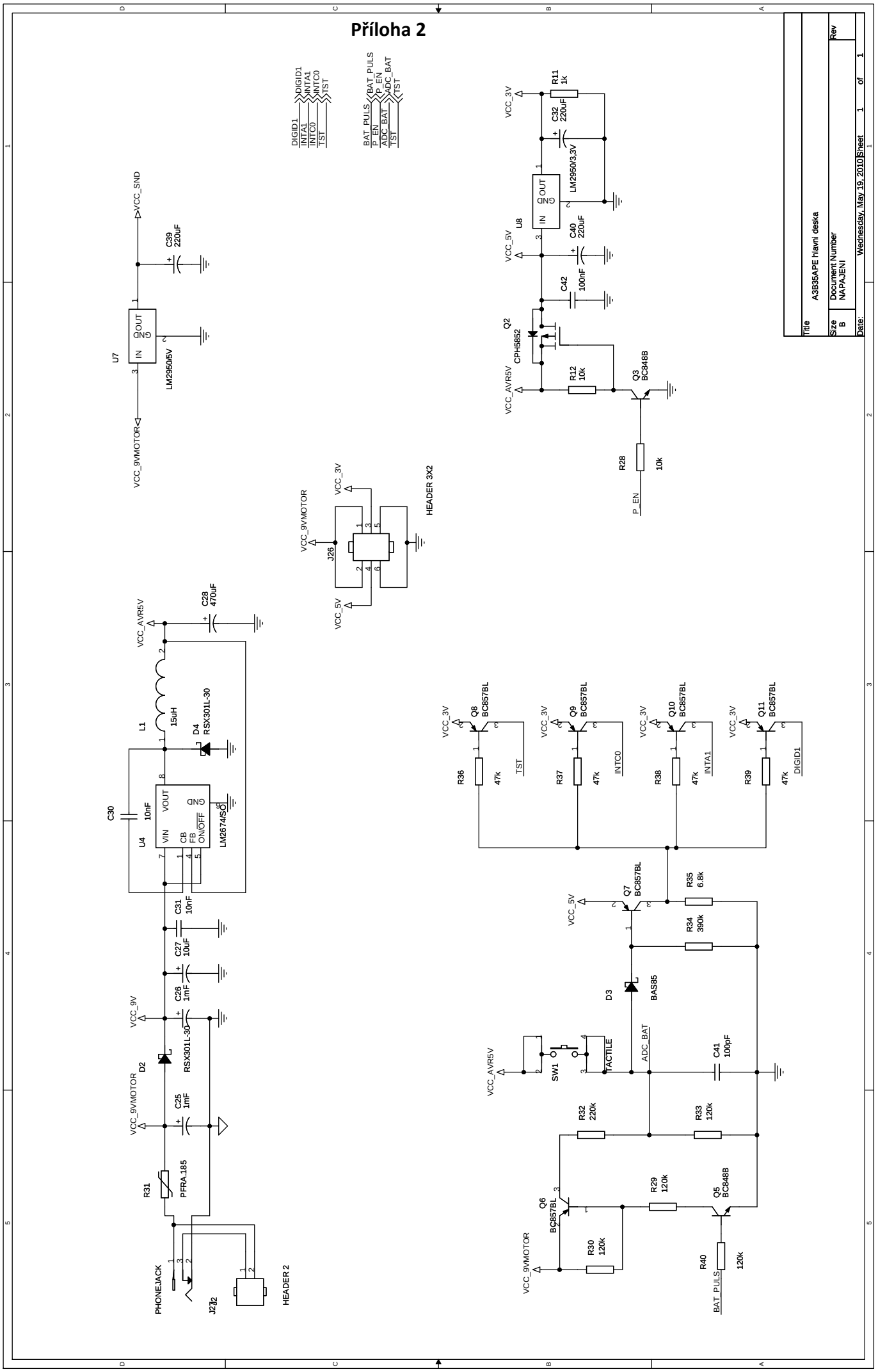
Příloha 4: **Schéma zapojení hlavní desky – vstupy**

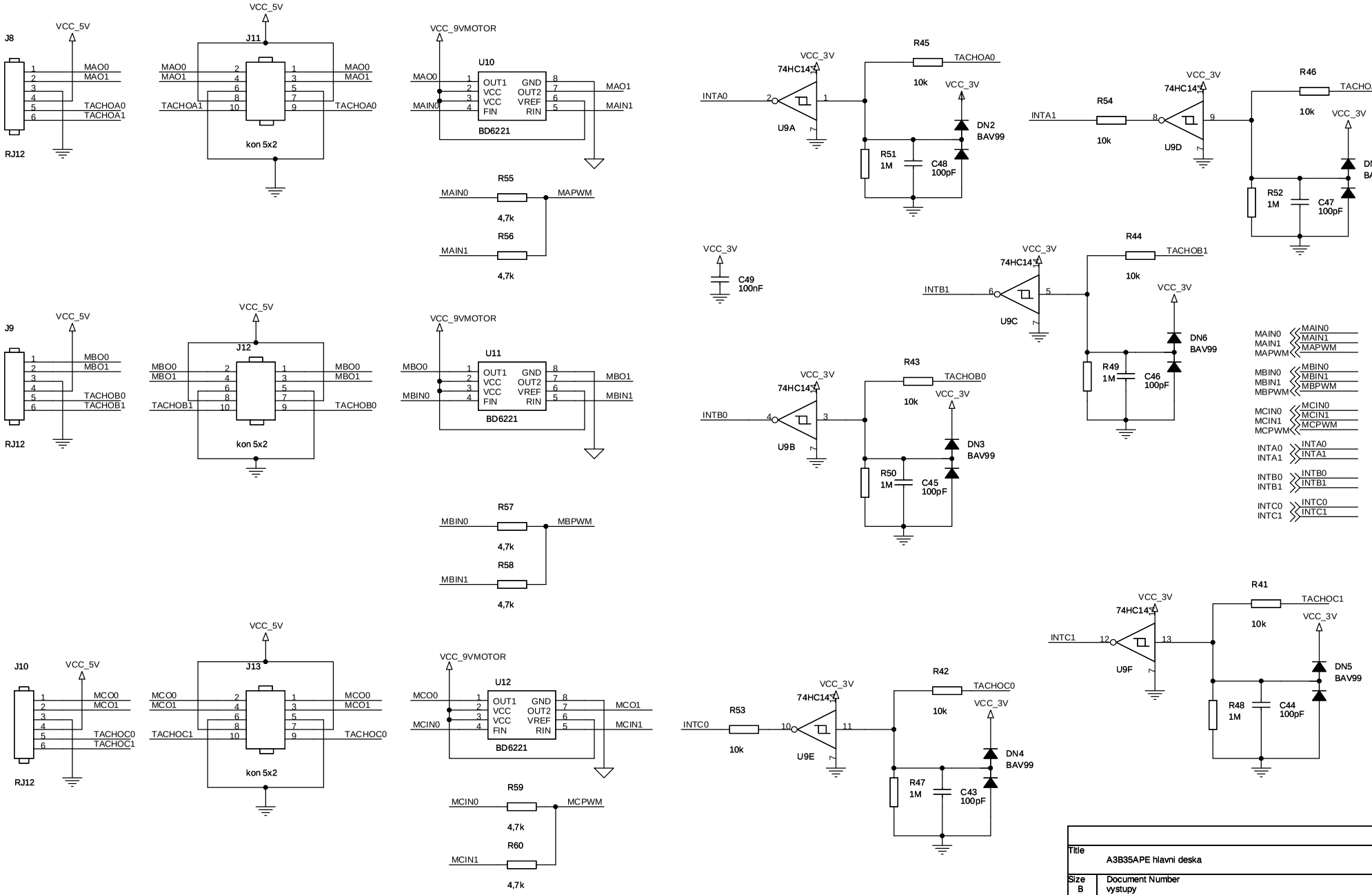
Příloha 5: **Schéma zapojení klávesnice a displeje – čtyř tlačítková verze**

Příloha 6: **Schéma zapojení klávesnice a displeje – šesti tlačítková verze**

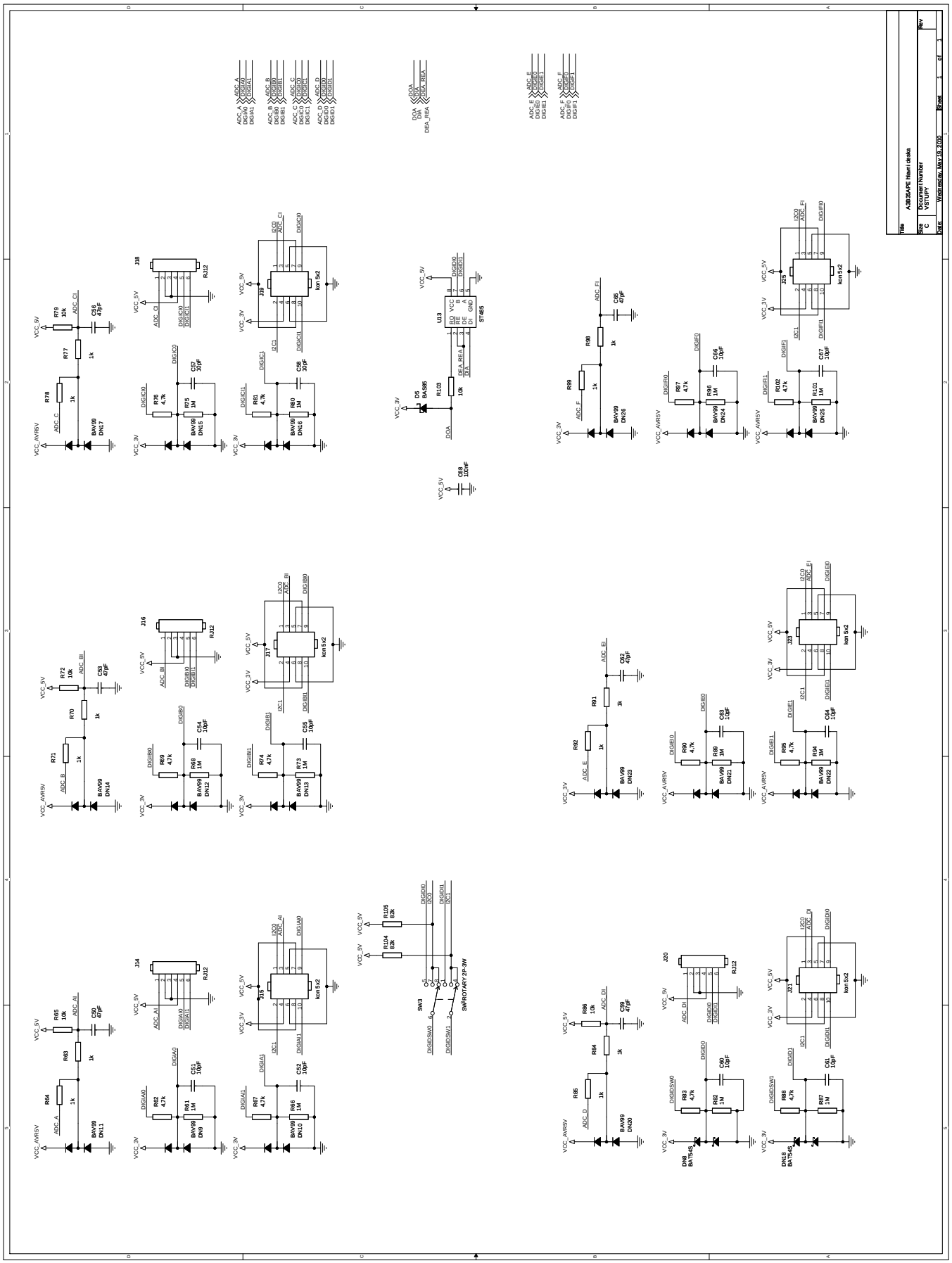

Příloha 1
U1
AT91SAM7S256
Name
U2
ATmega48/V
U3A
74HC73/SO
Q1
BC857BL
JP1
USB
J3
BLUETOOTH
J1
header 5x2
J4
ARM JTAG
J5
J6
HEADER 5X2
J7
ADER 2
U6
LM4861
Q4
BC848B
Y1
18.432MHz
PAD1
ERASE
PAD2
JTAGSEL
Title
A3B35APE hlavni deska
Size
B
Document Number
CPU
Rev
Date:
Wednesday, May 19, 2010
Sheet
1
of
1

Příloha 2

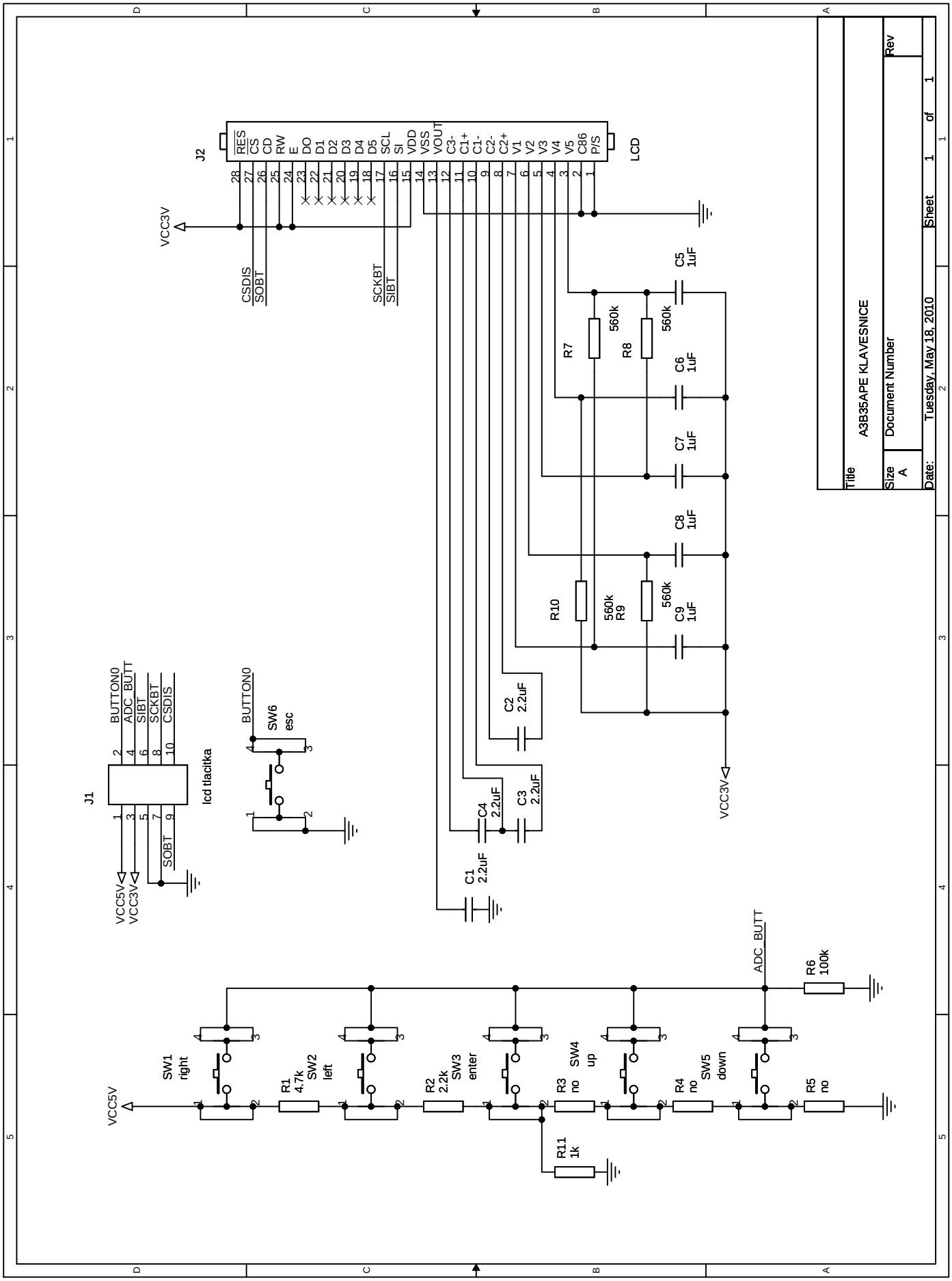

DIGID1
INTA1
INTC0
TST
BAT_PULS
P_EN
ADC_BAT
VCC_SND
C39 220uF
U7
LM2950/5V
VCC_9VMOTOR
VCC_3V
R11 1k
C32 220uF
LM2950/3.3V
U8
C40 220uF
C42 100nF
VCC_5V
Q2 CPH5852
VCC_AVR5V
R12 10k
Q3 BC848B
R28 10k
P_EN
J26
HEADER 3X2
C28 470uF
L1 15uH
D4 RSX301L-30
C30 10nF
U4 LM2674/SO
VIN VOUT CB FB ON/OFF GND
C31 10nF
C27 10uF
C26 1mF
VCC_9V
D2 RSX301L-30
C25 1mF
R31 PFRA.185
PHONEJACK
J2
HEADER 2
Q8 BC857BL
Q9 BC857BL
Q10 BC857BL
Q11 BC857BL
R36 47k
R37 47k
R38 47k
R39 47k
Q7 BC857BL
R35 6.8k
R34 390k
D3 BAS85
SW1
TACTILE
ADC_BAT
C41 100pF
R32 220k
R33 120k
R29 120k
Q6 BC857BL
Q5 BC848B
R30 120k
R40 120k
Title A3B35APE hlavni deska
Size B
Document Number NAPAJENI
Rev
Date: Wednesday, May 19, 2010
Sheet 1 of 1


J8
RJ12
VCC_5V
MAO0
MAO1
TACHOA0
TACHOA1
J11
kon 5x2
VCC_9VMOTOR
U10
OUT1
VCC
VCC
FIN
GND
OUT2
VREF
RIN
BD6221
MAIN0
MAIN1
MAPWM
R55
4,7k
R56
4,7k
J9
RJ12
MBO0
MBO1
TACHOB0
TACHOB1
J12
kon 5x2
U11
MBIN0
MBIN1
MBPWM
R57
4,7k
R58
4,7k
J10
RJ12
MCO0
MCO1
TACHOC0
TACHOC1
J13
kon 5x2
U12
MCIN0
MCIN1
MCPWM
R59
4,7k
R60
4,7k
INTA0
INTA1
INTB0
INTB1
INTC0
INTC1
74HC14
U9A
U9D
U9C
U9B
U9F
U9E
R45
R46
R44
R43
R41
R42
R53
R54
R49
R50
R47
R51
R52
R48
10k
1M
C48
C47
C46
C45
C44
C43
100pF
C49
100nF
DN2
DN3
DN4
DN5
DN6
DN7
BAV99
Příloha 3
Title
A3B35APE hlavni deska
Size
B
Document Number
vystupy
Rev
Date:
Wednesday, May 19, 2010
Sheet
1
of
1

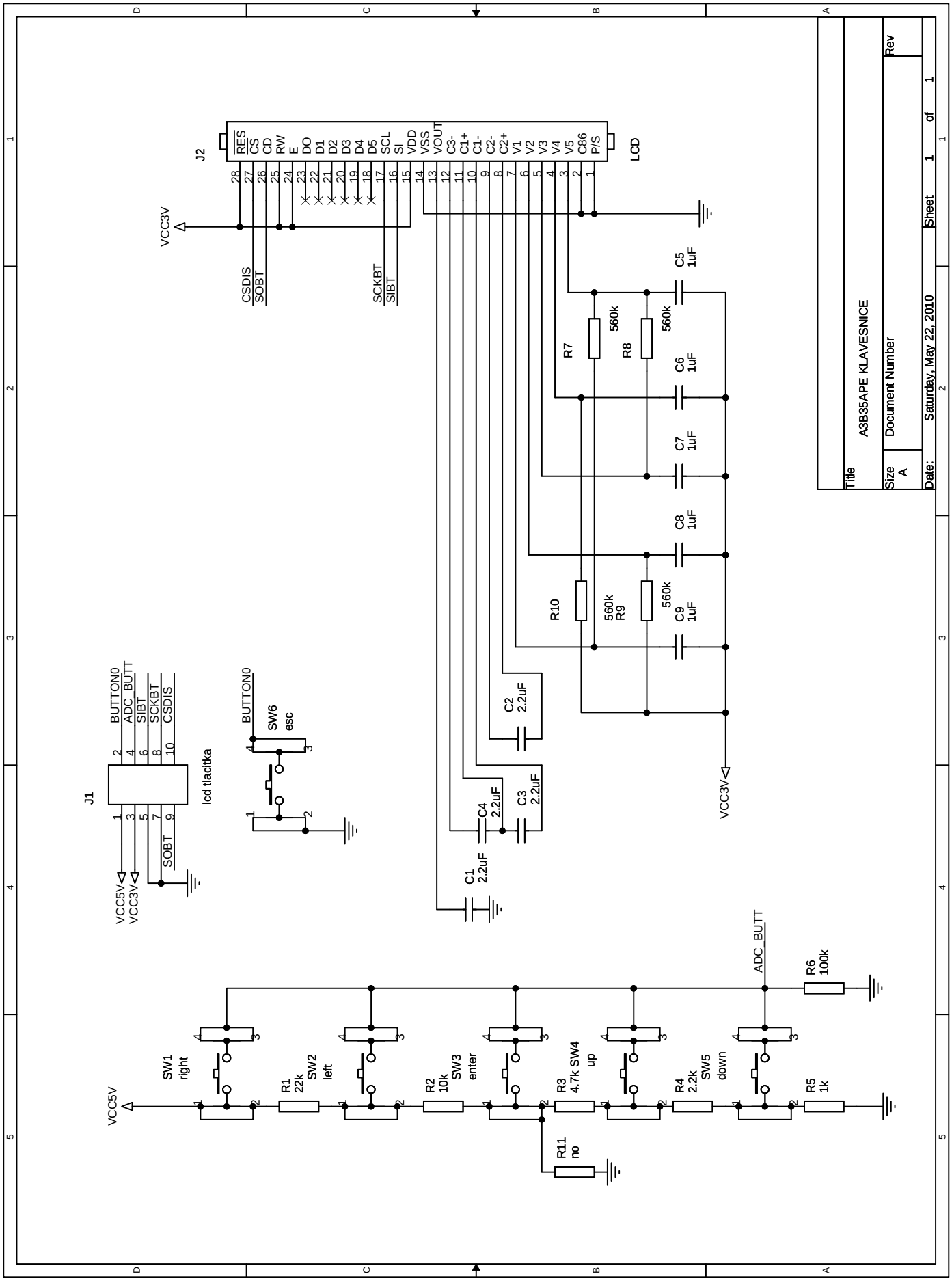
Příloha 4


Title
A38:35:PE Horní deska
Size
C
Document Number
VSTUPY
Rev
Date: Wednesday, May 19, 2010
Sheet 1 of 1

Příloha 5


J2
LCD
RES
CS
CD
RW
E
DO
D1
D2
D3
D4
D5
SCL
SI
VDD
VSS
VOUT
C3-
C1+
C1-
C2-
C2+
V1
V2
V3
V4
V5
C86
P/S
VCC3V
CSDIS
SOBT
SCKBT
SIBT
R7
560k
R8
560k
R10
560k
R9
560k
C5
1uF
C6
1uF
C7
1uF
C8
1uF
C9
1uF
C1
2.2uF
C2
2.2uF
C3
2.2uF
C4
2.2uF
VCC3V
J1
BUTTON0
ADC_BUTT
SIBT
SCKBT
CSDIS
SOBT
lcd tlacitka
VCC5V
VCC3V
BUTTON0
SW6
esc
VCC5V
SW1
right
R1
4.7k
SW2
left
R2
2.2k
SW3
enter
R3
no
SW4
up
R4
no
SW5
down
R5
no
R11
1k
ADC_BUTT
R6
100k
Title
A3B35APE KLAVESNICE
Size
A
Document Number
Rev
Date:
Tuesday, May 18, 2010
Sheet
1
of
1

Příloha 6


J2
LCD
RES
CS
CD
RW
E
DO
D1
D2
D3
D4
D5
SCL
SI
VDD
VSS
VOUT
C3-
C1+
C1-
C2-
C2+
V1
V2
V3
V4
V5
C86
P/S
VCC3V
CSDIS
SOBT
SCKBT
SIBT
R7
560k
R8
560k
R10
560k
R9
560k
C5
1uF
C6
1uF
C7
1uF
C8
1uF
C9
1uF
C2
2.2uF
C3
2.2uF
C4
2.2uF
C1
2.2uF
J1
BUTTON0
ADC_BUTT
SIBT
SCKBT
CSDIS
SOBT
lcd tlacitka
VCC5V
VCC3V
SW6
esc
SW1
right
SW2
left
SW3
enter
SW4
up
SW5
down
R1
22k
R2
10k
R3
4.7k
R4
2.2k
R5
1k
R6
100k
R11
no
ADC_BUTT
Title
A3B35APE KLAVESNICE
Size
A
Document Number
Rev
Date:
Saturday, May 22, 2010
Sheet
1
of
1