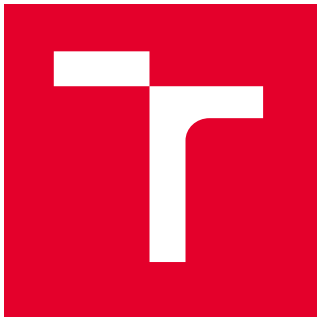
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2022 Bc. Branislav Hatala



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

TESTOVACÍ STOLICE PRO MIKROKONTROLÉROVÉ KITY

TEST BENCHES FOR MICROCONTROLLER KITS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — **Bc. Branislav Hatala**

AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Zdeněk Bradáč, Ph.D.**

SUPERVISOR

BRNO 2022

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Branislav Hatala ***ID:*** 195308

Ročník: 2 ***Akademický rok:*** 2021/22

NÁZEV TÉMATU:

Testovací stolice pro mikrokontrolérové kity

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte koncepci elektronického systému pro komplexní testování výukových kitů. Zaměřte se mikrokontrolérový systém v takové konfiguraci, aby umožnil bezpečné připojení spektra HW modulů a jejich komplexní obvodový test. Systém navrhněte, realizujte, osaďte a oživte. Vybavte programovým vybavením pro komplexní testy a demonstrujte správnou funkci.

1. Proveďte literární rešerši a internetový průzkum.
2. Navrhněte koncepci systému tak, aby byl vysoce spolehlivý.
3. Navrhněte obvodová schémata a zrealizujte HW a oživte ho.
4. Vytvořte programové vybavení pro komplexní obvodové testování HW. Vybavte možností tvořit testovací scénáře.
5. Vytvořte programové vybavení pro testování SW úloh běžících na výukových kitech. Vybavte možností tvořit testovací scénáře.
6. Oživte, otestujte a zhodnoťte dosažené výsledky.

DOPORUČENÁ LITERATURA:

Pavel Herout: Učebnice jazyka C, KOPP, 2004, IV. přepracované vydání, ISBN 80-7232-220-6

Dle pokynů vedoucího práce.

Termín zadání: 7.2.2022 ***Termín odevzdání:*** 18.5.2022

Vedoucí práce: doc. Ing. Zdeněk Bradáč, Ph.D.

doc. Ing. Petr Fiedler, Ph.D.
předseda rady studijního programu

ABSTRACT

Táto práca sa zaoberá návrhom a zostrojením testovacej stolice pre mikrokontrolérové kity. Hlavným účelom testovacej stolice je test funkčnosti modulov výukového kitu. Samotná testovacia stolica má modulárny hardvér a umožnuje použitie akéhokoľvek dostačujúceho embedded systému ako svojho systémového jadra. Hardware systému je rozšíritelný na úlohu testovania firmwaru výukových kitov.

KEYWORDS

Testovací stolice, vývojový kit, mkl-27z, periférne zariadenia, návrh PCB

ABSTRAKT

The aim of this work is development and creation of modular test bed for embedded education kit. Purpose of test bed is to test hardware functionality of education kit modules. The test bed's hardware is modular and can support any sufficient core(embedded system). The hardware supports higher functionality, including testing firmware of education kit.

KLÍČOVÁ SLOVA

Test bed, development kit, mkl-27z,peripherals,PCB design

HATALA, Branislav. *Testovací stolice pro mikrokontrolérové kity* . Brno: Brno University of Technology, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2022, 65 p. Master's Thesis. Advised by doc. Ing. Zdeněk Bradáč, Ph.D.

Author's Declaration

Author:	Bc. Branislav Hatala
Author's ID:	195308
Paper type:	Master's Thesis
Academic year:	2021/22
Topic:	Testovací stolice pro mikrokontrolérové kity

I declare that I have written this paper independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the paper and listed in the comprehensive bibliography at the end of the paper.

As the author, I furthermore declare that, with respect to the creation of this paper, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll. of the Czech Republic, Section 2, Head VI, Part 4.

Brno ....

author's signature*

*The author signs only in the printed version.

ACKNOWLEDGEMENT

Rád by som na tomto mieste poďakoval vedúcemu diplomovej práce doc. Ing. Zdeněkovi Bradáčovi, Ph.D. za odborné vedenie, konzultácie a trpezlivosť. Tiež by som chcel chcel vyjadriť svoju vďaku konzultantovi Ing. Jakubovi Armovi, Ph.D za podnetné rady a pomoc so smerovaním práce.

Contents

List of Figures

List of Tables

Listings

Úvod

Nutnosť efektívne testovať funkčnosť hardware je v prípade vypožičiavania kitov na dištančnú výuku relevantná potreba. Možnosť efektívného a objektívneho spôsobu ako automatizovane testovať firmware od žiakov spôsobom, ktorý by im poskytol efektívnu spätnú väzbu sa javí ako užitočný nástroj.[1]

0.1 Ciele práce

Táto práca sa venuje návrhu a zostrojeniu testovacej stolice pre mikrokontrolérové kity (ďalej už len prípravok), konkrétne kit používaný vo výuke predmetu MPC-POR a jeho periferných zariadení. Účelom prípravku je testovať hardware obsiahnutý vo vývojovom kite a zároveň byť základom systému pre automatizované testovanie vypracovaných zadaniach vo výuke, ktorá tento kit využíva.[1]

Čo sa týka hardwaru, prípravok je modulárný a umožňuje použiť akúkoľvek adekvátnu platformu na jeho riadenie. V tejto práci je použitá platforma využívajúca MCU mkl27z. Tá je k prípravku pripájaná cez kolíkovú lištu(na prípravku). Pripájanie testovaného kitu sa uskutočnuje cez kolíkové lišty. Podobným spôsobom sa pripájajú aj periférne zariadenia.[1]

1 Teoretická časť študentskej práce

1.0.1 Základná koncepcia konštrukcie[1]

Prípravok je určený najmä na testovanie funkčnosti periférií z výukového kitu. Zároveň má byť rozšíriteľné na úlohu automatizovaného testovania pre software. Táto skutočnosť musí byť braná v úvahu od začiatku procesu návrhu dosky plošného spoja.

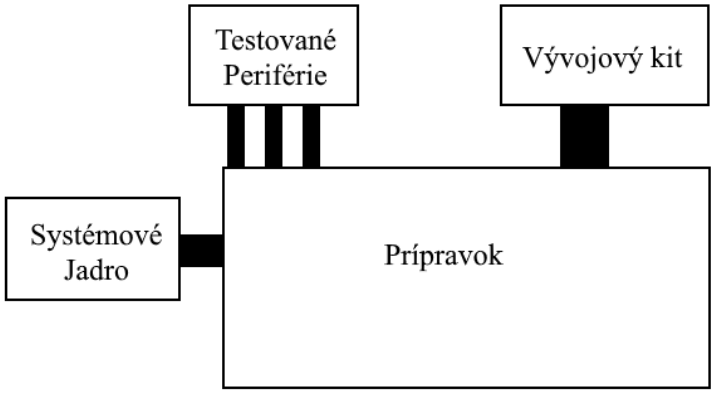


Fig. 1.1: Modularita riešenia

Z praktických dôvodov je dobré voliť modulárny design. Spôsob akým je modularita rozvrhnutá je znázornení na ukážke 1.1, prípravok samotný zabezpečuje prepojenia modulov, ochrany a ovládacie prvky. Oddelenie systémového jadra umožňuje zmenu riadiacej platformy bez nutnosti upravovať samotný PCB prípravku. Systémové jadro plní úlohu testovania a ovládania prípravku, môže to byť mikrokontrolérový obvod, alebo iné embedded riešenie. Prípravok obsahuje v sebe konektorové sloty na jednotlivé testované periférie. Slot pre modul LCD displeju je zároveň aj slotom pre vývojový kit, zapojenie vývojového kitu umožňuje využiť prípravok aj na testovanie funkčností programu vývojového kitu. To otvorí možnosti v automatizácií testovania výukových zadaní.

1.1 Štruktúra testovacej úlohy [1]

Úloha testovania a jej realizácia musí byť praktická pre koncového užívateľa. Testy je vhodné rozdeliť na jednotkové procedúry, z ktorých každá overuje nedeliteľnú funkcionalitu, alebo funkcionalitu prakticky nedeliteľnú. Napríklad u kláves a tlačidiel sa dá takto ich funkcionalita rozdeliť na tri testy:

- Otestovanie prítomnosti správnej logickej úrovne na vstupe pre stlačené tlačidlo.
- Otestovanie absencie zmeny logickej úrovne na vstupoch nestlačených tlačidiel.
- Otestovanie doby *bouncingu* pri stlačení a uvoľnení tlačidla.

Cieľom užívateľa je zistiť či testované zariadenie prešlo úspešne všetkými testami, to je primárna úloha prípravku. Ak zariadenie neprešlo všetkými testami, tak zistiť ktorými jednotlivými testami prešlo a ktoré z nich zlyhali je druhotná úloha zariadenia. Predpokladá sa, že vo väčšine prípadov budú zariadenia prechádzať testami a prípadné zisťovanie podrobností o zariadeniach ktoré testom neprešli, má menšiu prioritu čo sa týka rýchlosti a užívateľskej príhodnosti.

Z praktických dôvodov je vhodné, aby zariadenie vykonávalo svoju primárnu úlohu s minimálnym zásahom užívateľa. Musí byť možné testovať zariadenia jednotlivo bez vzájomných závislostí. Zároveň túto primárnu funkcionalitu by malo byť možné vykonať bez nutnosti pripojenia virtuálneho sériového portu, ktorý by pre túto funkcionalitu bol nadbytočný. Vhodný spôsob ako prípravok ovládať za účelom využívania jeho primárnej funkcionality je priradiť ovládacie a indikačné prvky naň. Matica tlačidiel sa javí ako efektívny spôsob na ovládanie spúšťaniu testov zariadení, čo sa hodí ale v prípade že sú zariadenie testované jedno po druhom. V prípade že úloha testovania je určená skôr na testovanie celej sady modulov naraz, je vhodnejšie spustiť testy naraz jedným tlačidlom. Tlačidlo je možné nahradiť dokonca aj iným modulom a tak prípravok môže ušetriť ďalší pin.

Indikáciu úspešnosti je vhodné vykonávať LED indikátorom, ktorý indikuje úspešnosť testu. Jeho výstup bude ukazovať výsledok posledného testu a indikátor zmení svoj výstup začiatkom vykonávania testu. Takéto ovládanie je praktické pre vykonávanie primárnej úlohy zariadenia.

Druhotná úloha zariadenia vyžaduje komplikovanejšie komunikačné rozhranie. Vhodným kandidátom je virtuálna sériová linka, tá by slúžila ako výstup pre detailné informácie z jednotlivých vstupov. Zariadenie môže vykonávať svoju primárnu úlohu aj bez nutnosti pripojiť toto rozhranie a užívateľ tak nie je zbytočne brzdený pri používaní zariadenia na jeho primárny účel.

1.2 Periférie a rozhrania kitu [1]

Testovaný kit obsahuje:

- ATmega328P Xplained mini,
- LCD keypad shield 1602A,
- Relé modul,
- Enkodér s tlačidlom,
- Modul RTC + EEPROM,
- Modul s termistorom,

- Reproduktor 8Ω 0.5W,
- Maticová klávesnica.

Reproduktor je zariadenie ktoré nie je používané vo výuke a tak bude práca s nim vynechaná aj v tejto práci.

1.2.1 ATmega328P Xplained mini

ATmega328P Xplained mini je vývojový kit využívajúci mikrokontrolér ATmega328P. Zahrňuje integrovaný debugger z rozhraním USB. Vývody PCB sú kompatibilné s arduinom a rad pinov portu *C* je dlhší o dva piny.

Predvolené operačné napätie kitu je 5 V, ktoré je získavané priamo z *USB* rozhrania. Napájacie napätie je možné zmeniť na 3.3 V, tým že sa zaspájkuje na dosku plošného spoja príslušný rezistor a *jumper* slúžiaci na prepínanie zdrojov napätia. Použitie napájacieho napätia 3.3 V limituje frekvenciu procesoru na 8 MHz. Pri konštrukcií prípravku sa preto berie ohľad na to, že vývojový kit bude pracovať s 5 V napájaním.

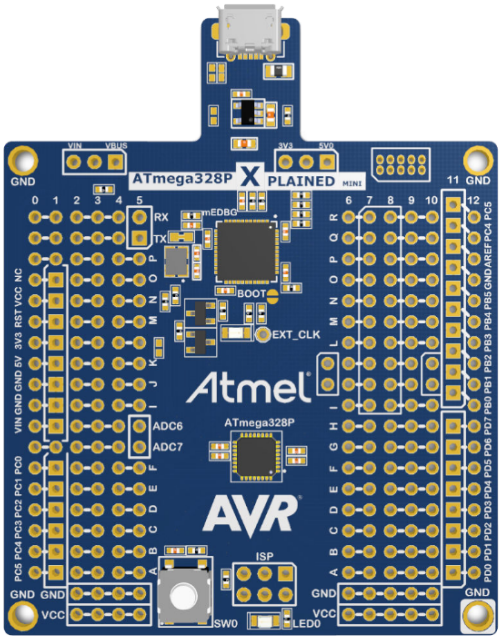


Fig. 1.2: ATmega328P Xplained Mini

1.2.2 LCD keypad shield 1602A

Ide o shield, ktorý má pevne dané vstupy. Kombinuje tlačidlá a LCD maticový display, tlačidlá môžu reagovať len na stlačenie jedného z nich. Stlačenie tlačidla je

indikované napäťovou úrovňou spoločného vývodu, ich používanie vyžaduje použitie Analógovo-digitálneho-prevodníku.

Samotný display je ovládaný cez množinu *General Purpose Input Output*. Prenos dát môže fungovať v štvor alebo osem pinovom režime. Jedná sa o paralelnú komunikáciu.

Testovacia procedúra pre overenie plnej funkčnosti by mala overiť opakovaný, dynamický výpis na display a zároveň testovanie stlačení tlačidiel. Zároveň by testovacie procedúry pre jednotlivé zariadenia nemali byť závislé na sebe navzájom, vypisovanie stlačení tlačidiel z lcd modulu na display je vhodný spôsob ako to docieliť. Podstatným problémom testovania užívateľských rozhraní je nutnosť zásahu užívateľa.

LCD modul je určený pre napájacie napätie 5 V, čo v prípade použitia 3.3 V platformy vyžaduje ošetrenie rozdielu napäťových úrovní. Dokumentácia dohľadateľná k tomuto modulu je neadekvátna a nezmieňuje napätia logických úrovní. Bežne zariadenia fungujúce na úrovni 5 V pracujú obdobnými úrovňami ako sú tie u *Transistor-to-Transistor Logic*, čo umožňuje čítať signály z 3.3 V zariadení, ale v tomto prípade to nie je potvrdené. Pre bežné užívanie a tým aj testovanie postačí použiť vstupy lcd modulu len na zápis do zariadenia, tak stačí ošetriť napäťový rozdiel v jednom smere pre prípad použitia 3.3 V platformy.

Analógový výstup slúžiaci na čítanie stlačení tlačidla je tiež v rozsahu 0 - 5 V, v prípade použitia 3.3 V platformy je možné napäťový rozdiel ošetriť deličom napätia.

1.2.3 Zariadenia pripájané cez GPIO

Niekoľko zariadení vo výukovom kite využíva na pripojenie len samotné *General Purpose Input Output*, prípadne *Interrupt on change* modul. Medzi tieto jednoduché zariadenia patria ovládacia prvky ako:

- Relé modul,
- Enkodér s tlačidlom,
- Maticová klávesnica.

Maticová klávesnica

Používaná maticová klávesnica pozostáva zo štyroch riadkov a štyroch stĺpcov. V laboratórnych úlohách sú využité len 3 stĺpce ktoré ovládajú číslicovú časť. Táto skutočnosť naznačuje, že napájanie stĺpcov a čítanie riadkov je výhodnejší postup. Pre praktické využitie zariadenia funkcionalita *interrupt on change* nie je nutná, hodnoty riadkov v prípade voľby vhodnej skenovacej frekvencie je možné čítať pred výmenou napájaného stĺpcu a postupné skenovanie riadku spolu s obmedzenou rýchlosťou človeka je adekvátny filter.

Pre plné otestovanie klávesnice je nutné použiť 4 digitálne vstupy a 4 digitálne výstupy.

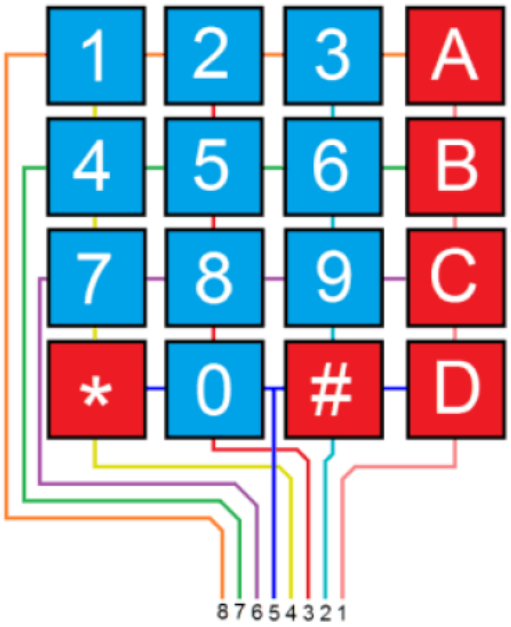


Fig. 1.3: Používaná maticová klávesnica

Testovanie vyžaduje postupné stlačenie všetkých tlačidiel užívateľom, čo je úkon zvládnuteľný do 10 sekúnd. Vhodné je testovať aj dĺžku okna v ktorom dochádza k bouncingu.

Enkodér s tlačidlom

Enkodér s integrovaným tlačidlom mimo napájania vyžaduje použitie 3 digitálnych vstupov. Charakter používania enkodéru v praxi bežne vyžaduje použitie *Interrupt on change* modulu, keďže vyhodnocovanie jeho akcií je na časovanie náročnejší úkon než u klávesníc a tlačidiel.

Relé modul

Obsahuje dva relé spínače. Jeho plné otestovanie je možné bez participácie užívateľa. Potrebné sú dva digitálne výstupy a štyri vstupy. V prípade testovania správnosti fungovania výstupov *normally open* a *normally closed*. V tomto prípade sa na spoločný kontakt privedie napätie zodpovedajúce logickej úrovni 1 a otestuje sa logická úroveň na výstupoch pre oba stavy vstupu.

Relé modul je určený pôvodne pre arduino platformu, ktorá pracuje na úrovni 5 V a zároveň je známa vysokými prúdmi, ktoré môžu tiecť cez jej *GPIO*. To môže byť nevhodné pre niektoré platformy, ktoré nie sú schopné dosahovať také prúdy na výstupoch. Pri riešení je nutné brať do úvahy spínací prúd, ktorého uvádzaná hodnota je do 20 mA.

1.2.4 Zariadenia s I2C zbernicou

Modul RTC(DS1307) a EEPROM(AT24C32) pamäť sú I2C zariadenia na jednej doske plošného spoja. RTC modul má definovanú rýchlosť *I2C* zbernice 100 kHz a tým určuje frekvenciu zbernice. *I2C* zbernica zakladá svoju funkcionalitu na vysielaní, ktoré pripojené zariadenia vykonávajú stiahnutím napätia zbernice na zemniace napätie. Zbernica je cez *pull-up* rezistory pripojená na napájacie napätie. 5 V zariadenia sú schopné čítať logickú jednotku vysielanú zariadeniami s napájaním 3.3 V pre spôsob, akým je na zbernici vysielané býva možné pripájať 5 V zariadenie na 3.3 V *I2C* zbernicu. Pre možnosť výberu napätia logickej jednotky zbernice *I2C* je vhodné pridať jumper, čo umožní voľbu napájacieho napätia v prípade použitia len 5 V zariadení.

V prípade že testovaný modul obsahuje v sebe zabudované *pull-up* rezistory na 5 V, je nutné ošetriť zmenu kľudového napätia na zbernici pri použití 3.3 V zariadenia.

RTC modul nevyužíva len I2C zbernicu na komunikáciu. Má digitálny výstup, ktorým vysiela svoje hodinové pulzy. Prípravok by mal testovať prítomnosť týchto pulzov a zároveň aj ich približnú presnosť. Funkčnosť samotnej I2C zbernice je možné otestovať zápisom a čítaním z pamäte zariadenia.

1.3 Možné problémy a ich riešenia pri návrhu [1]

1.3.1 Prevody napäťových úrovní analógových

Pasívny napäťový delič

Najjednoduchší prvok, ktorým je možné jednosmerne znižovať napäťovú úroveň je pasívny napäťový delič. V prípravku sa nachádzajú dva prípady, kde je možné ho použiť. Výstup tlačidla lcd shieldu a výstup modulu termistoru. Veľkou nevýhodou pasívneho napäťového deliča je jeho nepoužiteľnosť v prípade neadekvátnych výstupných odporov.

Takéto zapojenie je použiteľné aj pre digitálne signály. V tom prípade je možné použiť polovodičové prvky ako zenerové diódy pre väčšiu spoľahlivosť.

Aktívny napäťový delič

Aktívny napäťový delič je spôsob ako vyriešiť problém pasívneho napäťového deliča s jeho závislosťou na vysokom výstupnom odpore. Jeho zapojenie je zobrazené na ukážke 1.4 , je realizovateľný aj s jedným operačným zosilňovačom. Zosilňovač U1A predradený pred napäťový delič slúži na impedančné oddelenie zdroja signálu a napäťového deliča, kde rieši problém závislosti na vysokom výstupnom odpore. Zosilňovač U1B tvorí aktívny zosilňovač. Nie je úplne nutný, ale ide o dobrý spôsob, ako zúžitkovať operačný zosilňovač, ktorý je v púzdre navyše.

Pri výbere súčiastok je dôležité, aby použitý operačný zosilňovač bol *unity-gain stable* a bolo ho tak možné zapojiť do obvodu s jednotkovým zosilnením.

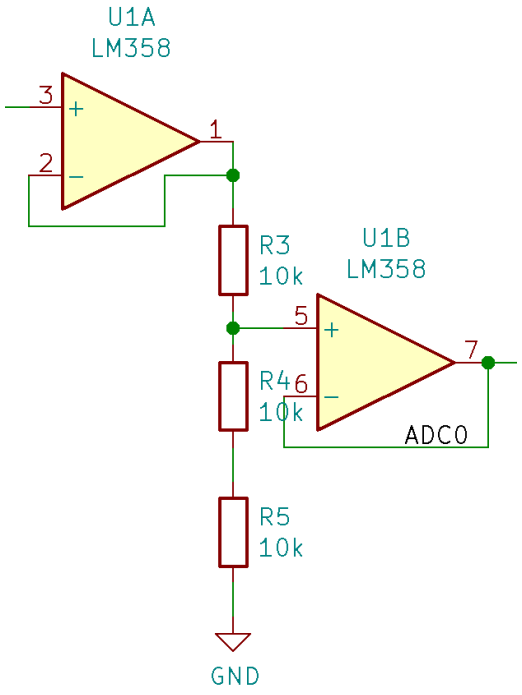


Fig. 1.4: Aktívny napäťový delič

Neinvertujúci zosilňovač

Najjednoduchší spôsob ako previesť analógové napätie na vyšší rozsah je použiť neinvertujúci zosilňovač zobrazený na ukážke 1.5 . Je to jednoduché zapojenie z negatívnou spätnou väzbou, ktorá je vedená cez napäťový delič. Problémom využitia neinvertujúceho zosilňovača je spôsob jeho ovládania, používa analógový vstup a tak vyžaduje, aby použitý mikrokontrolér obsahoval digitálne analógový výstup. Digitálno-analógové prevodníky nie sú úplne bežné v dostatočnom počte pre túto aplikáciu.

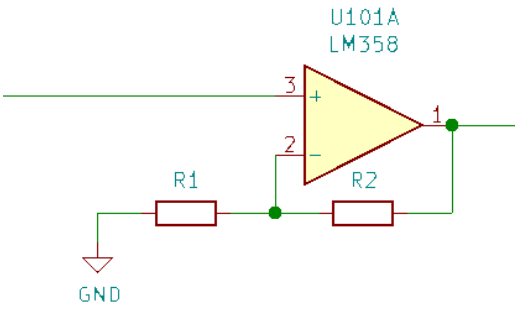


Fig. 1.5: Neinvertujúci zosilňovač

Externý Analógovo digitálny prevodník

Využitie externého analógovo-digitálneho prevodníku je spôsob, ako obísť problém rôznych napäťových úrovní. Analógovo-digitálne prevodníky sú dostupné zo širokým výberom rozhraní od rôznych paralelných a sériových vrátane *uart*,*usb*,*spi* a *i2c*. Pri použití rôznych napájacích napätí je i2c výhodnou možnosťou.

Ďalšia výhoda externého analógovo-digitálneho prevodníku sa prejaví v aplikáciach vyžadujúcich presnosť alebo rýchlosť. To najmä v aplikáciach kde využitie analógovo digitálneho prevodníku s postupnou sériovou aproximáciou, ktorý je v mikrokontroléroch najrozšírenejší nie je vhodné. Použitie externého prevodníku umožňuje vyberať medzi princípmi prevodu bez ohľadu na použitú platformu.

Ďalšou výhodou externého prevodníku je možnosť odľahčiť nároky pre návrh dosky plošného spoja s mikrokontrolérom, keďže analógové vlastnosti prestanú byť dôležité pre časť PCB s mikrokontrolérom a tieto nároky na kvalitný analógový design sa presunú na oddelenú analógovú časť, ktorá môže byť zároveň aj viac vzdialená od ostatných signálov a zberníc.

Externý Digitálne analógový prevodník

Zatiaľ, čo analógovo-digitálne prevodníky sú v niekoľkých kanáloch bežne integrované v bežných mikrokontrolérových platformách, digitálne-analógové prevodníky nie sú samozrejmosťou a ich počet kanálov býva obmedzený na jeden.

Oba typy prevodníkov zdieľajú výhody a komunikačné rozhrania.

1.3.2 Prevody napäťových úrovní digitálnych

zapojenie s otvoreným kolektorom a open drain circuit

Digitálne signály je možné jednosmerne prevádzať na vyššie úrovne pomocou relatívne jednoduchého zapojenia jedného tranzistoru a pullup rezistoru. Tranzistor môže byť bipolárny, alebo unipolárny. Keď je tranzistor zatvorený na výstupe je napätie logickej jednotky, jeho otvorením je úroveň výstupu stiahnutá na zem. Veľkosť pullup rezistoru znižuje rýchlosť nábehu signálu a kľudový prúd tečúci cez tranzistor.

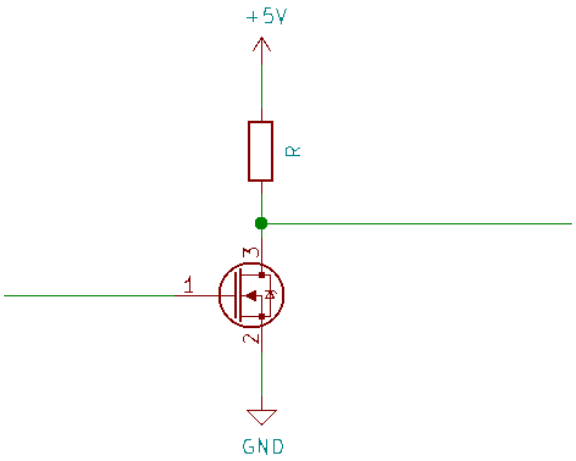


Fig. 1.6: Open drain zapojenie

Obojstranné zapojenie s unipolárnym tranzistorom

Unipolárny tranzistor je možné použiť aj na obojstranné prevádzanie napäťových úrovní. Spôsob zapojenie pre obe signálové linky i2c obvodu je zobrazená na ukážke 1.7. Fungovanie tohoto obvodu sa dá zhrnúť nasledovných bodoch:

- Ak žiadne zariadenie nevysiela, napätia oboch zberníc sú vytiahnuté ich *pull-up* rezistormi na ich kľudové napätia. Tranzistor je zatvorený, keďže napätie *gate-source* je nulové. Zabudovaná dióda tranzistoru neprepúšťa prúd, keďže je polarizovaná v závernom smere.
- Ak vysiela zariadenie na 5 V strane, stiahne napätie na nulu, zabudovaná dióda sa tým stáva polarizovaná v otvorenom smere pre prúd tečúci z 3.3 V strany.
- Ak vysiela zariadenie na 3.3 V strane, napätie *source* elektródy je stiahnuté na nulu, zatiaľ čo napätie hradla tranzistoru je 3.3 V.

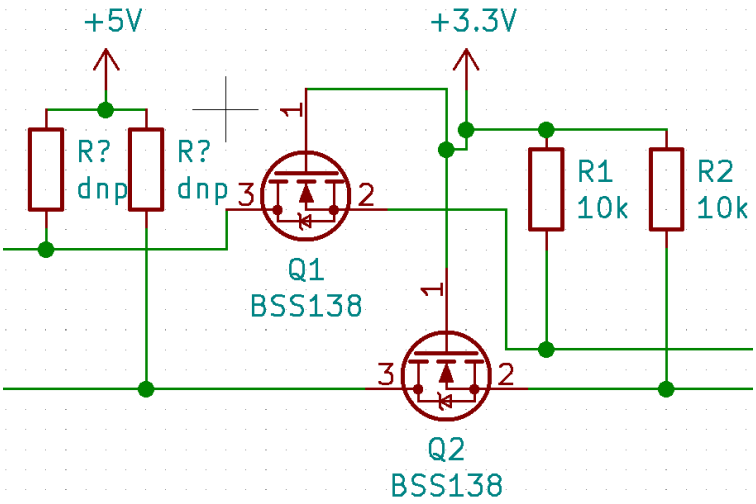


Fig. 1.7: Obvod pre prepojenie I2C zberníc s rôznym napätím

Jednosmerné prevodníky digitálnych napäťových úrovní

Existujú integrované obvody, ktoré slúžia na prevod napäťových úrovní v jednom smere. Existujú fixné varianty alebo varianty, kde sa smer prevodu prepína pomocou digitálneho vstupu. Využitie jednosmerných prevodníkov v aplikáciach kde kombinácia vstupov a výstupov nie je pevne daná vedie k redundancií a dodatočnému prepájaniu.

Obojsmerné prevodníky digitálnych napäťových úrovní

Sú dostupné rôzne prevodníky digitálnych napäťových úrovní, ktoré pre každý kanál detekujú smer. Jedným takým je aj TXB0108 od texas instrument. Jedná sa o 8 bitový obojsmerný prevodník napäťových úrovní s automatickým detekovaním smeru a ±15 kV ESD ochranou.

Spôsob akým tento obvod zaisťuje obojsmernosť je vysielaním slabých logických signálov (*weak logic low/high*. To znamená, že logická úroveň je zabezpečená pomocou slabého prúdu. Slabú logickú úroveň je vysielajúce zariadenie na pripojenej linke schopné stiahnuť na svoju požadovanú vysielanú úroveň. V tejto situácií sa vysielač prevodníka napäťovej úrovne správa ako pullup, alebo pulldown rezistor, na ktorého zbernici začalo vysielať zariadenie jemu protichodný signál. Zapojenie akým je tento spôsob vysielania realizovaný je zobrazený na ukážke 1.8, kde je zobrazená časť prevodníku pre jednu signálovú linku. Na ukážke vidieť, ako je prevod napäťovej úrovne na slabú zabezpečený 4k rezistorom na výstupe. Pre zlepšenie vlastností signálu a odstránenie jedného z nepriaznivých následkov použitia slabej

signálovej úrovne je tento obvod v *TXB0108* rozšírený o takzvaný *one-shot*, nazývaný tiež ako hranový akcelerátor. Tento obvod reaguje na zmenu vysielanej logickej úrovne vstupu a na krátku dobu pripojí výstup priamo na napájacie napätie alebo zem. Toto krátke napojenie zásobuje silnú logickú úroveň tým, že funkčne obchádza výstupný 4k rezistor. *One-Shot* obvod je aktívny po dobu približne 10ns , nevýhodou tohoto obvodu je jeho nevhodnosť pri veľkých kapacitných záťažiach, pri ktorých napäťová úroveň nestihne dosiahnúť požadovanú logickú úroveň, kým funguje *one-shot* obvod.

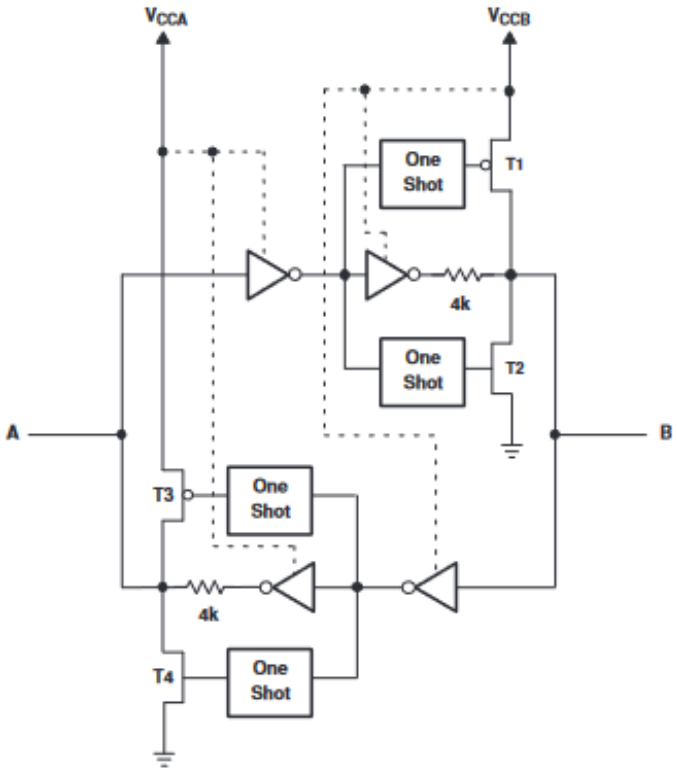


Fig. 1.8: Schéma jednej bunky prevodníku TXB0108 [2]

GPIO expandér

General purpose input output expander je externý integrovaný obvod, ktorý funguje ako digitálne vstupy a výstupy v mikrokontroléri.

Pri použití na premostenie rôznych napäťových úrovní sa použije rovnaký postup ako pri analógovo-digitálnom a digitálno-analógovom prevodníku, zariadenie sa zapojí na požadované napätie a zvolí sa *i2c* ako komunikačné rozhranie.

1.3.3 Zdieľanie vstupov

Prípravok na vykonávanie svojej úlohy vyžaduje značné množstvo vstupov aj výstupov. Nie všetky funkcie prípravku ktoré zaberajú vstupy a výstupy pracujú nevyhnutne súčasne čo umožňuje viacerým funkciám zdieľať rovnaké piny mikrokontroléru a tým znížiť ich počet.

Existujú rôzne metódy takéhoto zdieľania vstupov a výstupov, ktoré sa líšia vo svojej rozšíriteľnosti, limitáciách a iných vlastnostiach.

Multiplexer

Využitie multiplexeru je spôsob, ako jedným digitálnym vstupom čítať hodnotu z n vstupov na prepínanie medzi vstupmi je využitých m digitálnych výstupov, kde $n = 2^m$, v prípade že prepínanie vstupov nie je časovo kritické, je možno zredukovať počet potrebných výstupov na prepínanie pripojením počítadla pred riadiace vstupy multiplexeru. To zníži počet potrebných výstupov na dva inkrement a reset.

Existujú aj analógové multiplexeri na trhu, čo umožňujú použiť túto metódu na analógové signály. Tento prístup umožňuje vysielať analógový signál len na jeden vstup.

Použitie multiplexeru znemožňuje získavať presné časovanie a spoľahlivo spúšťať *interrupt on change* pre jednotlivé vstupy. To neplatí pre situácie, kde je nutné detektovať len zmenu jedného zo vstupných signálov.

dekodér

Využitie dekodéru umožňuje pomocou n vstupov ovládať 2^n výstupov takým spôsobom, že kombináciou vstupov sa určuje na ktorom výstupe je logická jednotka, zatiaľ čo na ostatných je logická nula. Využitie dekodéru je vhodné na ovládanie stĺpcov maticovej klávesnice.

Dekodér je možné doplniť obvodom s pamäťou, čo umožňuje mať na viacerých vstupoch naraz logickú jednotku, reset môže byť vykonávaný jedným z výstupov dekodéru alebo zvlášť, v prípade prítomnosti hodinového vstupu na registry. Je však nutné použiť prídavný výstup zo systému na jeho ovládanie. Takéto riešenie je vhodné pre ovládanie signálnych LED diód.

Integrovaný spínač

Existujú integrované obvody, ktoré sa správajú ako spínač s unipolárnym tranzistorom. Jedným konkrétnym príkladom je TMUX721x. Schéma popisujúca jeho správanie je na ukážke 1.9. Jeden takýto obvod je možné použiť na zdvojnásobenie akejkoľvek kombinácie vstupov a výstupov. Na to je najvhodnejší TMUX7213,

pri ktorom zapojenie zahrňuje zapojenie vstupov po dvoch a prepojenie všetkých ovládacích vstupov v jeden.

Dostupné sú aj varianty vhodné pre použitie na analógové účely. Ich zopnutý odpor sa pohybuje v rozmedzí stoviek až jednotiek Ω a spínací čas môže dosahovať až jednotky nanosekúnd. Najrýchlejšie varianty nie sú práve tie s najmenším odporom a dosahujú minimálne hodnoty 15 Ω.

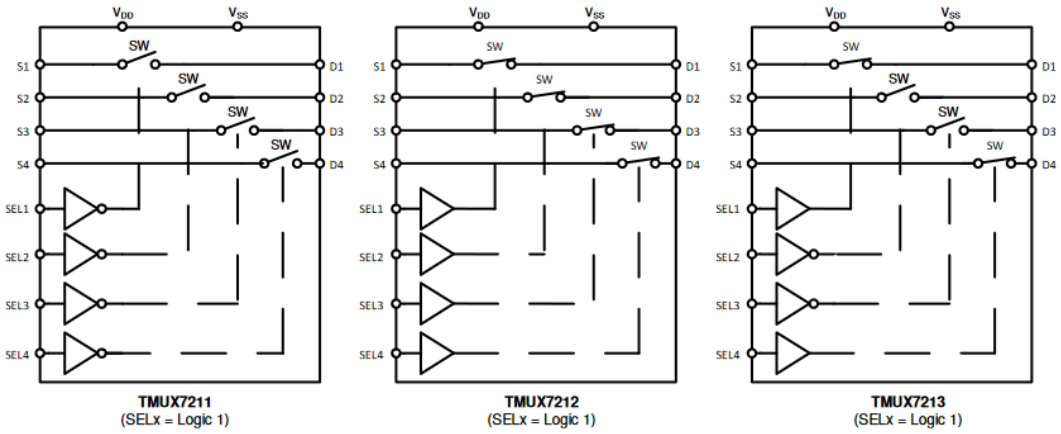


Fig. 1.9: Logická schéma spínacích obvodov TMUX721x [3]

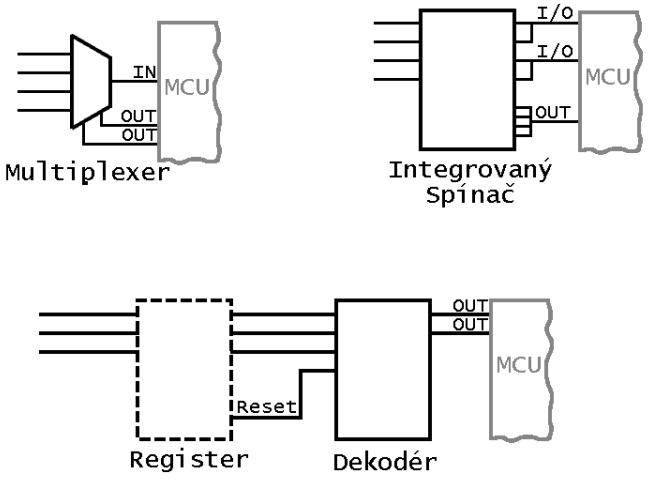


Fig. 1.10: Zapojenie rôznych riešení zdieľania pinov.

1.3.4 Získavanie a stabilizácia napájacích napätí

Poskytovanie napájacích napätí *5V* a 3.3 V je úlohou, ktorú je vrámci modularity a univerzality modulov lepšie riešiť vrámci modulu systémového jadra. Ako zdroj napájania je vhodný *USB* port. Jeho napätie sa pohybuje v rozmedzí 4.75 V až 5.25 V, čo znemožňuje jeho priame prepojenie na 5 V a vyžaduje reguláciu. s reguláciou je možné dosiahnúť adekvátnu kvalitu napájania pre analógové účely.

Z dôvodu energetickej efektivity sú najvhodnejšie dve voľby princípov regulátorov LDO regulátor, alebo Spínací regulátor.

LDO regulátory

Jedná sa o *Low Dropout* regulátory. Tieto regulátory mávajú nízke stratové prúdy v jednotkách od stoviek *nA* do jednotiek *uA* a závisí od výstupného zaťaženia regulátora.

Trvalý výstupný prúd sa pohybuje u 3.3 V do 300 mAa u 5V do 150 mA. Výhodou LDO regulátorov je aj nízky výstupný šum. Ich funkčný princíp ich predurčuje pre znižovanie napätia, čo ich robí vhodným na získavania 3.3 V z USB napätia.

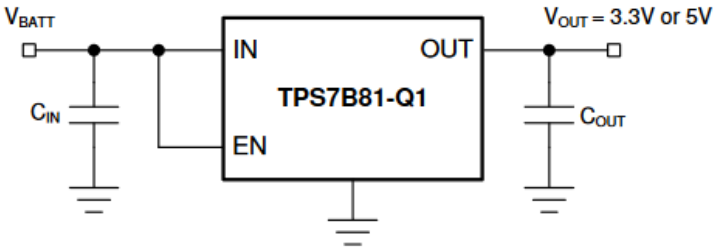


Fig. 1.11: Typické zapojenie LDO regulátoru

Spínacie regulátory

Spínacie regulátori využívajú ako akumulátor energie kombináciu kondenzátoru a cievky. Sú schopné zvyšovať aj znižovať napäťovú úroveň automatickým prepínaním medzi režimom *step-down* a *boost-mode*.

Výhodou spínacích regulátorov je robustnosť, efektivita a vysoký trvalý výstupný prúd. Nevýhodou môže byť spínací šum.

1.3.5 Vedenie dráh plošného spoja

Vodiče a dráhy na doske plošného spoja majú parazitické vlastnosti, ktoré je nutné zohľadniť pre funkčnosť a spoľahlivosť systému. Parazitické vlastnosti ktoré vo

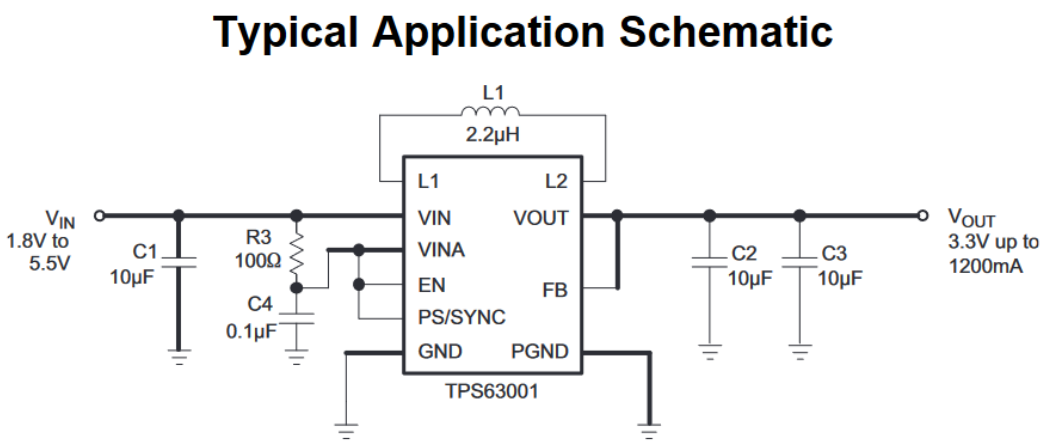


Fig. 1.12: Typická aplikácia spínaného regulátoru TPS6300x.

väčšine situácií prevládajú sú zobrazené na ukážke 1.13 Vzťah pre približný výpočet

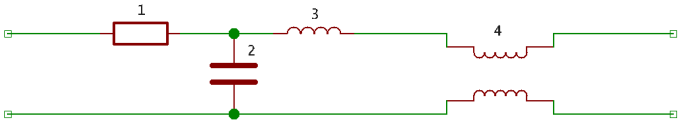


Fig. 1.13: Ukážka vybraných parazitických vlastností vodičov: 1. rezistivita, 2. vzájomná kapacita, 3. indukčnosť, 4. vzájomná indukčnosť.

indukčnosti dráhy plošného spoja je vzorec 1.1, ukážka 1.14 dovysvetluje význam premenných z nasledujúcich vzorcov. Vzorec naznačuje bežný spôsob potlačenia indukčnosti vedenia bežne odporúčaný v katalógových listoch, zväčšením šírky dráhy W.

Zo vzťahov 1.1, 1.2 a 1.3 vidieť, že dosahovanie optimálnych parametrov dráh plošného spoja je hľadanie kompromisu. Znižovanie indukčnosti a odporu zvyšuje kapacitu a naopak.

$$L \approx \frac{\mu_0 \cdot \mu_r \cdot (H + T/2) \cdot \ell}{W} \quad [\mathrm{H}] \tag{1.1}$$

$$R = \rho \cdot \frac{\ell}{W \cdot T} \quad [\Omega] \tag{1.2}$$

$$C = \epsilon \cdot \frac{W \cdot \ell}{H} \quad [\mathrm{F}] \tag{1.3}$$

Ďalšia dôležitá parazitická vlastnosť zapojení je prítomnosť slučiek, na ktorých sa môže indukovať napätie. Tieto slučky sa môžu vyskytnúť s osou kolmou k doske

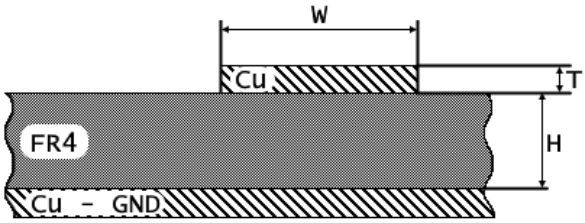


Fig. 1.14: Ukážka rozmerov pre výpočtové vzťahy 1.1, 1.2 a 1.3.

plošného spoja, alebo z osou v ploche. Takéto slučky sú tvorené prekovmi. Negatívny vplyv slučiek je možné eliminovať znížením ich plochy. To sa dosahuje vedením vodičov, ktoré tvoria slučku v tesnej blízkosti. Ak slučka pozostáva zo zemniaceho vodiča, tak jej negatívny vplyv je potlačený použitím rozliatej zeme.

Signálové vodiče sa ovplyvňujú kapacitnými a indukčnostnými väzbami. Tieto väzby sú najsilnejšie v prípade paralelne vedených vodičov a najnižšie v prípade vodičov vedených kolmo k sebe. V prípade kapacitnej väzby je tento jav ovplyvnení najmä efektívnou vzdialenosťou. V prípade indukčnostných väzieb je závažným faktorom uhol medzi vodičmi. Vedenie vodivých dráh na rôznych vrstvách dosky plošného spoja často vedie k použitiu 45° uhlov medzi vodičmi na rôznych vrstvách, v takom prípade sa zníži zároveň priebeh funkcie efektívnej vzdialenosti medzi vodičmi a zároveň aj ich uhol zoslabí indukčné väzby na $sin(\alpha)$ násobok pôvodnej veľkosti kde $\alpha = 45$ř tento princíp používajú aj rezolvery na meranie uhlu natočenia.

Diferenciálne vedenia

Prípravok a riadiaci modul používajú niekoľko komunikačných rozhraní, u ktorých je nutné dbať na zníženie rozdielu medzi vzdialenosťami ich vodičov a ich kapacít. Takúto povahu majú rozhrania ako *I2C*, *SPI* a *UART*. Viesť tieto rozhrania ako diferenciálny pár s väčším rozostupom a menšou šírkou vodičov je praktický spôsob ako to zaručiť.

USB je rozhranie, ktoré je nutné viezť ako diferenciálny pár. Elektrické vlastnosti stanovené v štandarde *USB* sú prísne a vedú k veľkej robustnosti. *USB* zbernicu je

možné realizovať aj subštandardne, čo uľahčuje jej implementáciu.

USB zbernica je v literatúre bežne označovaná ako diferenciálna zbernica, hoci to nie je úplne pravda. Väčšina komunikácie je diferenciálna a pozostáva zo stavov zbernice J a K, existuje ešte jeden stav využívaný na komunikáciu *SE0*(Single ended zero). Tento stav zbernice sa používa pri indikácií konca paketu, odpojenia zariadenia a reset zariadenia. Tabuľka 1.1 obsahuje prehlad povolených stavov usb zbernice pre full-speed komunikáciu.

Stav	D+	D-
J	H	L
K	L	H
SE0	L	L

Tab. 1.1: Tabuľka povolených stavov USB zbernice pre full-speed režim.

Rýchlosť signálov na USB zbernici je vysoká a štandard diktuje diferenciálnu impedanciu vedenia 90 Ω. Dosahovanie tejto hodnoty na dvojvrstvovej doske plošného spoja je obtiažne a pre jej dosiahnuteľnosť s praktickou šírkou vodičov a ich medzerou vyžaduje použitie koplanárneho vlnovodu. Dosiahnutie štandardom predpísaných parametrov nie je nevyhnutné pre funkciu rozhrania, mnohý výrobcovia skúšobných kitov na parametre nedbajú a dokonca vedú zbernice cez prekovy.

1.3.6 Ochrana pred ESD

Mnohé elektronické súčiastky integrujú do seba ochranu pred elektrostatickým výbojom, aj *TXB0108* implementuje v sebe ochranu pred *ESD*.

Najjednoduchšou a najkompaktnejšou formou ochrany pred ESD je použitie *TVS* diódy. TVS diódy sa zapájajú v závernom smere čo najbližšie k vstupom do plošného spoja, v prípade že napätie na dióde prekročí úroveň V_{BR}, dióda sa otvára a skratuje napätie elektrostatického výboja. Schematická značka a ekvivalentný obvod TVS diódy sú zobrazené na ukážke 1.16, *V-A* charakteristika TVS diódy je zobrazená na ukážke 1.17.

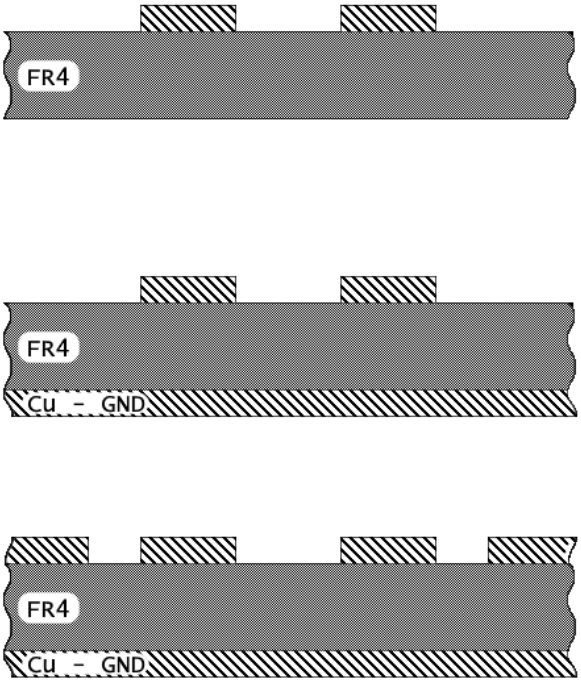


Fig. 1.15: Ukážka zjednodušených možností vedenia diferenciálneho páru, možnosti sú zoradené zhora nadol podľa diferenciálnej impedancie vzostupne Spodná konštrukcia je koplanárny vlnovod.

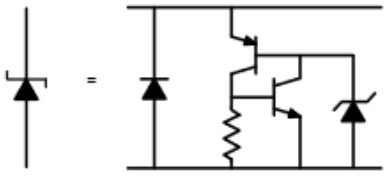


Fig. 1.16: Schematické značenie v datasheete ESD diódy ESD9101. [6]

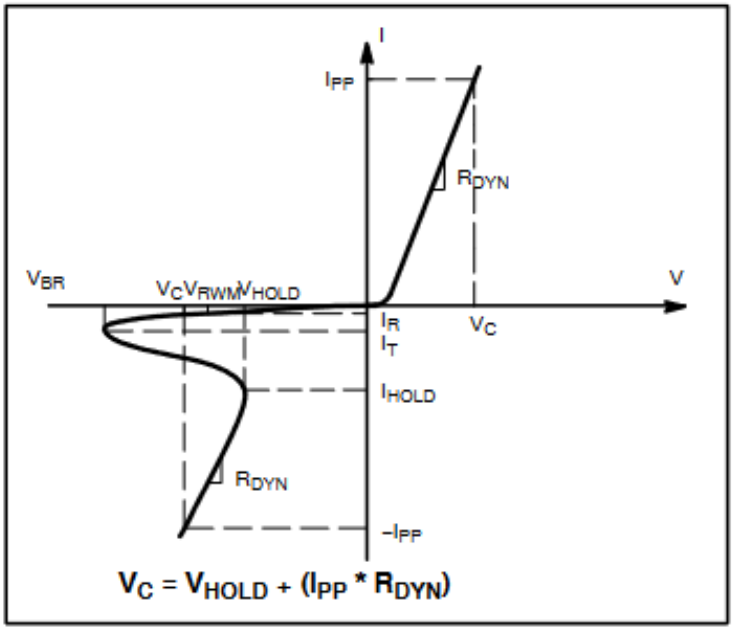


Fig. 1.17: *V-A* charakteristika ESD diódy ESD9101. [6]

1.4 Softvérová architektúra

Softvérová architektúra systému by mala mať nasledovné vlastnosti:

- Modularita,
- Migrovateľnosť,
- Rekonfigurovateľnosť,
- Rozsah použiteľnosti.

Vo svojom účely aj v spôsobe ich dosiahnutia sa spomenuté vlastnosti prelínajú. Tak snaha dosiahnutia jednej vedie k dosiahnutiu ostatných.

Modularita zvyšuje migrovateľnosť, pri dobrej architektúre umožňuje vymeniť jadro systému za akúkoľvek adekvátnu platformu. To všetko pri minimálnych zmenách kódu ktoré sú obmedzené na kód ktorý obsluhuje periférie jadra ktoré interagujú s prípravkom.

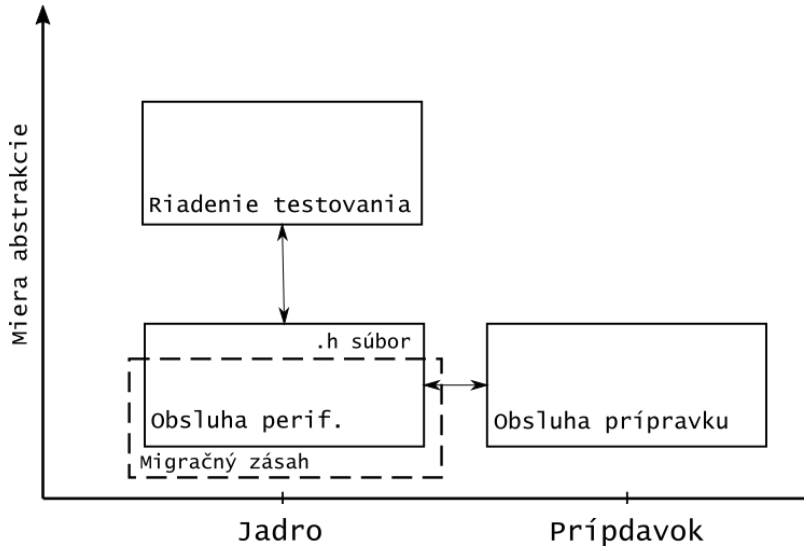


Fig. 1.18: Zobrazenie zásahu nutného pri migrácií na iné jadro.

Migrácia systémového jadra na inú platformu pri architektúre podobnej ako na ukážke 1.18 umožňuje minimálne zmeny v kóde. Tak v prípade migrácie je nutné len zmeniť kód, ktorý obsluhuje periférie zariadenia, kde spôsob fungovania potrebných funkcií je definovaný návrhom prípravku. Pri migrácií je vďaka tomu možné využiť starý .h súbor, ktorý definuje funkcie obsluhy HW jadra, ktoré kód pre obsluhu HW prípravku potrebuje.

Rekonfigurovateľnosť umožňuje množstvo prepojek, vďaka ktorým je možné meniť spôsob prepojenia spojov vedúcich do slotu na výukový kit/lcd display a zároveň aj ostatných testovaných súčastí. Rekonfigurovatľnosť HW dodáva zároveň prípravku rezilienciu voči návrhovým chybám a môže umožniť ich jednoduché riešenie.

Súčasťou rekonfigurovateľnosti je aj spôsob riešenia otázok testovania SW bežiaceho na kite. To vyžaduje možnosť meniť a spúšťať rôzne testovacie scenáre neznáme v čase a tvorbe tejto práci.

Úloha testovania HW kitu je zjednodušená tým, že sa nepredpokladajú zmeny v ich fungovaní po dokončení zariadenia, tak je možné vyriešiť ju hard-code riešením, v ktorom sa celá implementuje v programe samotnom.

Úloha testovania SW nie je už tak jednoduchá, keďže zadania pre SW nie sú nemenné, čo vyžaduje rekonfigurovateľnosť. Užitočnosť zariadenia je do značnej miery určená tak, aby bolo užívateľovi umožnené meniť procedúru testu SW.

V prípade niektorých platforiem použitých ako jadro je to jednoduché a hard-code prístup je dostačujúci. To platí pre prípady z integrovaným riešením programovania mikrokontrolérovej platformy. V prípade platforiem, ktoré sa spoliehajú na externé programátory ako P&E micro, je nutné implementovať dodatočné riešenie v prípade, že užívateľ externým programátorom nedisponuje.

Spôsobov ako meniť testovací scenár existuje široké spektrum, mnohé sú si navzájom veľmi podobné a jedná sa skôr o varianty rovnakého prístupu. Najzaujímavejšie možnosti sú:

- Sprostredkovanie udalostí
- Realizácia testov pomocou Lua
- Vytvorenie kódovacieho jazyka na mieru
- Hard-Code riešenie(Správanie programované priamo C, zmena preprogramovaním SW jadra.)

1.4.1 Sprostredkovanie udalostí

Tento spôsob riešenia spočíva v sledovaní udalostí, ktoré nastávajú na prepojení medzi jadrom systému a testovaným kitom.

Sledované by boli udalosti, ale aj stavy digitálnych výstupov kitu. Rovnako by bolo pracované s I2C zbernicou a každým výstupným rozhraním kitu. Zaznamenané údaje spolu z ich potrebnými údajmi by boli štruktúrovane posielané cez pripojené rozhranie do PC. V tomto riešení vyhodnocovanie správania testovaného kitu je vykonávané programom bežiacom na PC, ktorý komunikuje zo zariadením.

Vhodný spôsob komunikácie je sériová linka virtuálna, skutočná, alebo sprostredkovaná, ako to na dnešných vývojových platformách vrátane Xplained mini býva. Obslužný program by vyhodnocoval prichádzajúce udalosti, prípadne ich absenciu a vyvolával udalosti, na ktoré má testovaný kit reagovať. Program by za účelom vyvolávania týchto udalostí posielal do zariadenia príkazy v podobe štrukturovaných dát.

1.4.2 Realizácia testov pomocou Lua

Jazyk Lua je výkonný, efektívny, ľahký a zabudovateľný skriptovací jazyk. Podporuje procedurálne programovanie, objektovo orientované programovanie, funkcionálne programovanie, programovanie založené na údajoch a popis údajov.[7]

Jazyk Lua kombinuje jednoduchú procedurálnu syntax s výkonnými konštrukciami na opis údajov založenými na asociatívnych poliach a rozšíriteľnej sémantike. Jazyk Lua je dynamicky typovaný, beží na základe interpretácie bajt-kódu pomocou virtuálneho stroja založeného na registroch a má automatickú správu pamäte s inkrementálnym garbage collection systémom, vďaka čomu je ideálny na konfiguráciu, skriptovanie a rýchle prototypovanie.[7]

Lua je rýchly a malý jazyk, ktorý možno ľahko vložiť do rôznych aplikácií. Lua má jednoduché a dobre zdokumentované API, ktoré umožňuje silnú integráciu s kódom napísaným v iných jazykoch. Jazyk Lua je možné ľahko rozšíriť o knižnice napísané v iných jazykoch. Pomocou jazyka Lua možno tiež ľahko rozširovať programy napísané v iných jazykoch. Lua sa používa na rozšírenie programov napísaných nielen v jazykoch C a C++, ale aj v jazykoch Java, C#, Smalltalk, Fortran, Ada, Erlang a dokonca aj v iných skriptovacích jazykoch, ako sú Perl a Ruby.[7]

Lua je v praxi používaná najmä spôsobom, ktorý umožňuje v Lua kóde užívatelovi prepisovať callback funkcie, ktoré sú volané z vývojárskeho kódu. Užívateľ píše kód v ktorom môže volať funkcie implementované v C, ktoré vývojáry sprostredkovali. Takto je užívatelovy poskytnuté rozhranie, ktoré mu umožňuje meniť správanie aplikácie. Lua sa v tejto forme preslávila najviac v hernom priemysle, kde bola použitá v mnohých tituloch. V embedded je Lua menej známa, využíva ju napríklad Lego Mindstorms, ktoré používa používa pbLua. Ďalším zaujímavým užívateľom je Acrios Systems, ktorý vo svojich Internet of Things prevodníkoch implementuje eLua.

Kód v jazyku C, ktorý implementuje virtuálny stroj pre Lua, pracuje s premennou typu lua_state*, ktorá reprezentuje kontext Lua. Detaily sa môžu líšiť medzi verziami Lua. Implementácia Lua v jazyku C pozostáva z nasledujúcich častí:

- Inicializáciu Lua kontextu.
- Načítanie nutných Lua knižníc.
- Vytvorenie Lua knižnice na volanie aplikačných C funkcií.
- Volanie Lua funkcií z C kódu.

```
lua_State *lua = luaL_newstate()';
```

Inicializuje lua kontext, ten sa používa ako argument každej funkcie čo pracuje s Lua virtuálnym strojom.

```c
lua_State* luainit(void)
{

    lua_State *lua = luaL_newstate();
    luaL_openlibs(lua);

    static const struct luaL_Reg hwapi_lib[] = {
            {"ComWrite", hwapi_VcomWrite},
            {"delayms", hwapi_delayms},
            {"adcRead", hwapi_adcRead},
            {"i2cRead", hwapi_i2cRead},
            {"i2cWrite", hwapi_i2cWrite},
            {"brdDacWrite", hwapi_brdDacWrite},
            {"uartWrite", hwapi_uartWrite},
            {"uartRead", hwapi_uartRead},
            {NULL, NULL}};

        luaL_setfuncs(lua, hwapi_lib, 0);

        return lua;
}
```

Fig. 1.19: Ukážka kódu pre inicializáciu Lua virtuálneho stroja pre verziu Lua5.4.4
.

`luaL_openlibs(lua);`

Nahrádza otvára štandardné knižnice pre daný stav. V starších verziách Lua ako 5.1 a tým pádom aj eLua ktorá z verzie 5.1 vychádza sa postupne otvárajú knižnice nasledovným spôsobom:

```
luaopen_base(L);              /* opens the basic library */
      luaopen_table(L);             /* opens the table library */
      luaopen_io(L);                /* opens the I/O library */
      luaopen_string(L);            /* opens the string lib. */
      luaopen_math(L);              /* opens the math lib. */
```

Definícia pola štruktúr *luaL_reg* slúži na previazanie funkcií napísaných v C s ich menami, ktorými majú byť volané v Lua kóde. Štruktúra *luaL_reg* má v prvom člene reťazec, ktorým bude funkcia volaná z Lua kódu a v druhom ukazateľ na volanú funkciu. Pole končí nulovou štruktúrov, ktorej obe polia sú NULL. Následne volaná funkcia berie toto pole ako argument a slúži na registráciu funkcií.

`luaL_setfuncs(lua, hwapi_lib, 0);`

Lua 5.1 a eLua používajú namiesto predošlej funkcie nasledovnú:

```
luaL_register(lua, "api" , hwapi_lib);
```

Kde druhý argument slúži ako meno knižnice a funkcie sú v Lua kóde volané pomocou api.meno_funkcie().

Funkcie jazyka C takto sprístupnené do Lua majú správanie ktoré sa dá zhrnúť nasledovne:

- Sú statické funkcie.
- Nemali by používať statické a globálne premenné v C kóde.
- Majú návratovú hodnotu typu int.
- Majú jediný argument typu lua_state* reprezentujúci Lua kontext.
- Cez lua_state pristupujú k zásobníku z ktorého berú argumenty funkcie dodané z Lua prostredia.
- Návratové hodnoty do Lua prostredia pridávajú prostredníctvom spomenutého zásobníku.
- Ako C funkcia majú návratovú hodnotu ktorá reprezentuje počet hodnôt ktoré vrátili za behu do Lua prostredia cez zásobník.

C funkcia môže získavať hodnoty argumentov s ktorými bola volaná z Lua kódu zo zásobníka pomocou funkcií ako:

```
int lua_toboolean(lua_State *L, int index);
double lua_tonumber(lua_State *L, int index);
const *lua_tostring(lua_State *L, int index);
```

Overiť typ hodnoty v zásobníku je možné funkciami ako:

```
int lua_isinteger(lua_State *L, int index);
int lua_isnumber(lua_State *L, int index);
int lua_isstring(lua_State *L, int index);
```

Na získanie dĺžky objektu v zásobníku na danom indexe, tak aj na získanie dĺžky reťazcov, slúži nasledovná funkcia:

```
lua_Unsigned lua_rawlen(lua_State *L, int index);
```

Návratové hodnoty do Lua kódu sú riešené pomocou funkcií ako:

```
void lua_pushnil(lua_State *L);
void lua_pushboolean(lua_State *L, int bool);
void lua_pushnumber(lua_State *L, double n);
void lua_pushlstring(lua_State *L, const char *s, size_t length);
void lua_pushstring(lua_State *L, const char *s);
```

Pri používaní týchto funkcie je nutné dbať na to, aby zásobník nebol veľkostne prekročený. Zásobník Lua funkcia má minimálnu definovanú veľkosť nastavenú na 20, táto veľkosť sa dá meniť definíciou LUA_MINSTACK, ktorá sa nachádza v súbore lua.h. V prípade že je nutné zabezpečiť väčšiu veľkosť zásobníku, je možné volať nasledovnú funkciu:

```
int lua_checkstack(lua_State *L, int n);
```

Táto funkcia zabezpečí v zásobníku ďalších n miest. Návratová hodnota je nenulová, ak sa operácia podarila, v prípade že zásobník nebolo možné zväčšiť, návratová hodnota je nula.

Beh Lua programu môže byť zabezpečený dvoma spôsobmi.

- Just-in-time kompiláciou kódu za behu.
- Predkompilovaným binárnym súborom.

Pre embedded platformy je vhodnejší druhý spôsob a to najmä pre jednoduchosť a šetrenie výpočtovým výkonom. Binárny súbor pre Lua virtuálny stroj má pôvodnú príponu .out a je tvorený pomocou nástroja luac. Ide o Lua kompilátor, ktorý je súčasťou súborov projektu Lua. Lua kód je možné skompilovať na desktope a jeho binárny súbor preniesť do mikrokontrolérovej platformy a je následne spustený. Luac nemá informácie o tom aké Lua funkcie virtuálny stroj v mikrokontrolérovej platforme implementuje, problematika kompilovania užívateľom definovaných funkcií je riešená zaujímavým spôsobom, ktorý je zjavný z ukážky 1.20, na ktorej je zobrazený Lua kód a obsah výsledku jeho kompilácie.

Využitie jazyka Lua je vhodným spôsobom, ako pridať upraviteľnosť do správania systémov. Lua prináša mnoho možností na relatívne jednoduchú aplikáciu, ktorá je predmetom tejto práce ale aj mnoho týchto možností nadbytočných a tie dokážu skôr prinášať problémy s implementáciou a stabilitou. Ďalšou nevýhodou Lua je to, že je nevyhovuje MISRA smerniciam, čo môže byť nevýhodou v embedded platformách. Lua do veľkej miery pracuje s dynamicky alokovanými hodnotami a reťazcami znakov, čo je suboptimálne pre jednoduché úlohu fungujúce na jednoduchých platformách.

1.4.3 Jazyk implementovaný na mieru

Niektoré myšlienky na ktorých je predošlý prístup založený je možné využiť jednoduchším spôsobom než implementovať Lua. Na plnenie úlohy zariadenia stačí jednoduchší systém, ktorý nepotrebuje pracovať s dynamicky alokovanou pamäťou a funguje na podstatne jednoduchšom virtuálnom stroji. Pre plnenie úlohy zariadenie je nutné len niekoľko úkonov, ktoré sú jednoduché:

- Volať funkcie na prácu s HW.

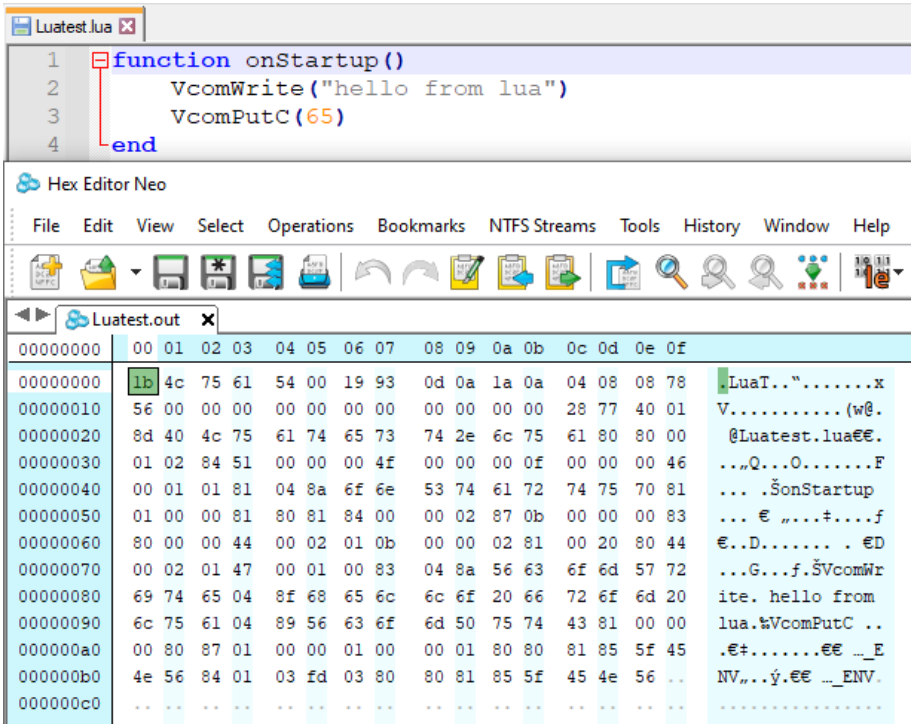


Fig. 1.20: Porovnanie Lua kódu s obsahom jeho skompilovaného binárneho súboru.

- Vyhodnocovať návratové hodnoty.
- Vykonávať základné operácie s premennými.
- Implementovať funkcie ktoré budú slúžiť na spracovanie udalostí.

Tento virtuálny stroj by pracoval len zo statickou pamäťou. Na fungovanie by mu vystačil jeden, prípadne viac rozdelených zásobníkov na prácu s pamäťou a jeden zásobník, ktorý by slúžil na uloženie inštrukcií.

Inštrukcie by boli interpretované s úrovňou abstrakcie rovnakou ako C funkcie a rezervované slová. Logické, binárne a matematické operácie by mali svoje inštrukcie, tak ako vetvenia a cykly. Takto by každý operátor, funkcia či rezervované slovo mali svoju inštrukciu. Inštrukcie ako číselné hodnoty je možné interpretovať pomocou switch štruktúry, alebo indexovaním pola ukazovateľov na funkcie. Argumenty a návratové hodnoty je možné riešiť pomocou zásobníku.

Takýto binárny kód by bol vo výsledku jednoduchší a kratší než ten, ktorý je výsledkom kompilácie Lua kódu. Prináša otázky ohľadne riešenia niektorých implementačných problémov, jedným z nich je spôsob ako rozlišovať či je argument inštrukcie konštanta, alebo premenná. Ide o pravdepodobne najzaujímavejšiu otázku architektúry virtuálneho stroja a kódu pre neho.

1.4.4 Výsledná architektúra spomenutých riešení

SW architektúra spomenutých riešení má spoločné prvky, ale aj rozdiely, ktoré vyplývajú z rozdielnosti prístupu. Ukážka 1.21 zobrazuje SW architektúru, farebné bloky rôznych farieb znázorňujú rôzne varianty, ktorých názov je uvedený v najvyššie umiestnených blokoch, zobrazených prerušovanou čiarou. Prepoje diagramu symbolizujú volanie funkcií, či sprostredkovanie funkcionality. Riešenie využívajúce jazyk Lua, alebo vlastný jazyk na mieru, obe pracujú s kompilovaným binárnym súborom. V oboch prípadoch je tento súbor načítaný a spustený vo virtuálnom stroji. Virtuálny stroj interpretuje binárny kód a pritom pracuje s HW tým, že volá funkcie najnižšej abstraktnej vrstvy. Túto najnižšiu abstraktnú vrstvu tvorí wrapper, ktorý slúži na volanie funkcií a tie sú špecifické pre HW použitého jadra.

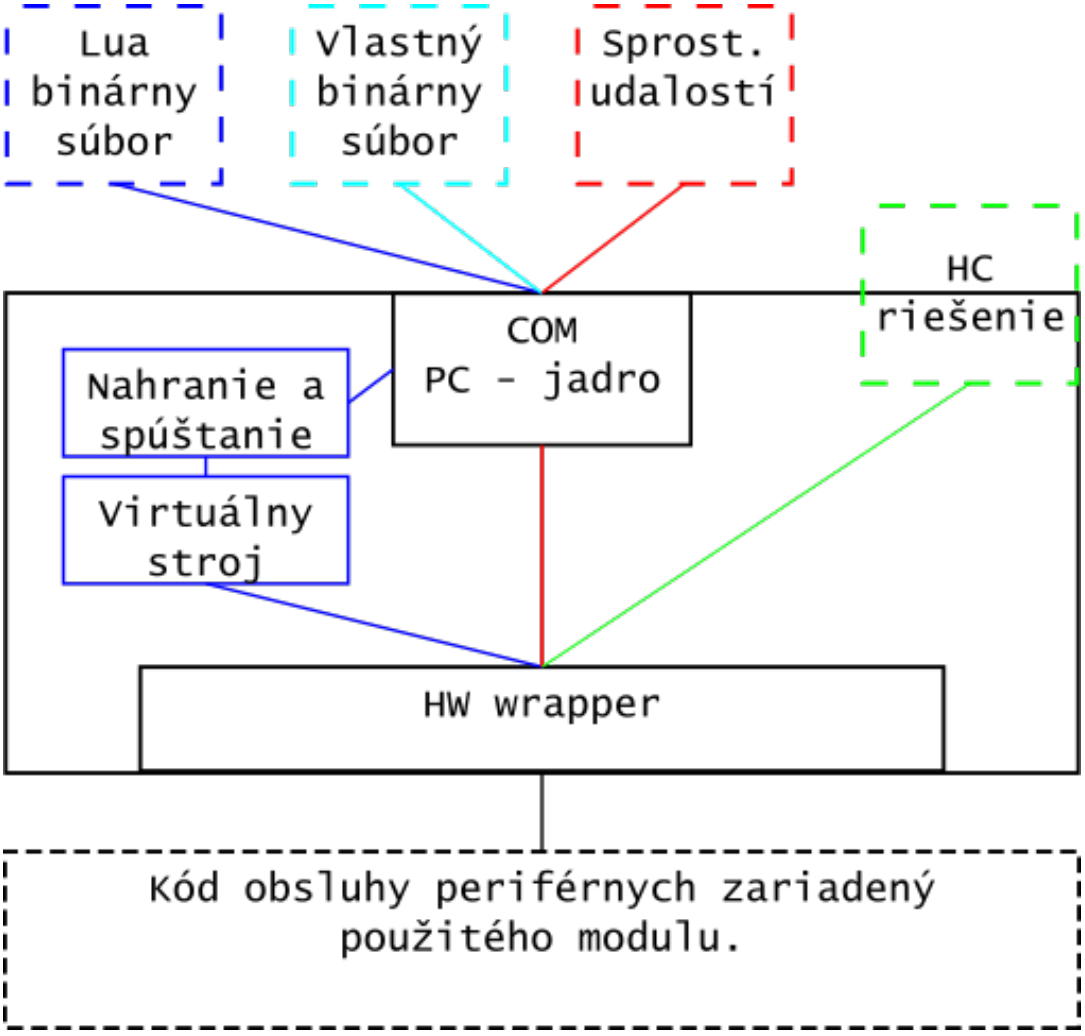


Fig. 1.21: Porovnanie variant a ich výslednej architektúry systému.

Voľba varianty je tiež podmienená použitým jadrom. Spôsob, akým je jadro

programované môže robiť Hard Code riešenie najideálnejším. Možnosť behu Lua virtuálneho stroja je tiež rozhodujúcim faktorom, jeho implementácia môže v závislosti od platformy prinášať rôzne problémy, ktoré môžu mať za následok výhodnosť iného riešenia.

Statická povaha testovania HW modulov navádza ku kombinácií hard code riešenia, prípadne jeho kombinácie s ktorýmkoľvek iným riešením. V prípade použitia Lua jazyka na mieru, alebo sprostredkovania udalostí, je možné vyriešiť úlohu testovania HW pomocou funkcie implementovanej v C.

COM rozhranie PC - jadro

Na splnenie úlohy zariadenia je nutné umožniť praktickým spôsobom ho ovládať, získavať z neho údaje a prípadne posielať do neho súbory. To všetko je najjednoduchšie dosiahnuteľné prepojením zariadenia s PC a to spôsobom, ktorý umožní komunikovať cez konzolové aplikácie. Toto riešenie je zároveň vhodným pre jeho multiplatformovosť, jednak z pohľadu na jadro systému a zároveň na operačný systém PC. Kvalitné konzolové aplikácie sú dostupné na všetky bežne používané operačné systémy. Rôzne systémy ktoré je možné použiť ako jadro systému, majú možnosť jednoduchej implementácie sériovej komunikácie, či sa jedná u priamy VCOM medzi použitým MCU a PC, alebo o systém kde je na PCB vedená sériová linka z MCU do komunikačného systému, ktorý túto linku sprostredkováva ako VCOM. Toto riešenie majú platformy ako Arduino a rôzne podobné vývojové kity.

Sériová linka umožňuje jednoducho aj posielať súbory. Cez konzolové aplikácie ako TeraTerm je možné posielať súbory rôznymi spôsobmi. Spôsob ako posielať súbory cez Tera Term je naznačený ukážkou 1.22. Tlačidlo transfer slúži na posielanie pomocou protokolov ako Xmodem, Ymodem, Zmodem... Tieto protokoly vyžadujú ich implementáciu na strane jadra systému, čo komplikuje ich použitie. Správanie týchto protokolov nie je ale pre túto aplikáciu vhodné, napr. Xmodem používa checksum v prípade jeho modernejšej varianty CRC, ktoré sú obe nadbytočné pre väčšinu jadier systému, keďže komunikujú cez VCOM kde USB protokol rieši bezchybnosť dát. Prípadne sériová linka je krátka a je vedená na PCB. Väčším problémom Xmodem protokolu je preddefinovaná veľkosť paketu 128 B, to vedie k problémom s kompatibilitou, keďže USB periférie majú obmedzenú veľkosť vyrovnávacej pamäte pre interrupt endpoint, ktorý VCOM rozhranie využíva na príjem dát na mkl27z tak, ako aj ostatných kinetis platformách a je táto veľkosť pre full-speed USB periférii maximálne 64 B.

Jednoduchším spôsobom ako poslať obsah súboru cez konzolovú aplikáciu je využiť možnosť *Send file....* Tá umožňuje posielať súbor spôsobom ,ktorý sa z pohľadu príjmajúceho zariadenia javí ako postupné posielanie znakov, ktoré sú ob-

sahom súboru. Je to jednoduché riešenie, ale prináša problémy s tým, že takto je možné posielať len určité hodnoty, ktoré zodpovedajú znakom, ktoré je možné posielať cez konzolu. Napríklad nie je možné poslať byte, ktorého hodnota je 0 alebo reprezentuje nezobraziteľný znak (non-printable). Riešenie tohoto problému je relatívne priamočiare. Stačí konverzia súboru, ktorý má byť poslaný na postupnosť hexadecimálnych hodnôt jednotlivých znakov, tak je možné poslať aj nezobraziteľné znaky. Potom všetky hodnoty, ktoré môže jeden byte nadobudnúť sú reprezentované dvojicou alfanumerických znakov, ktorými sa hexadecimálne hodnoty píšu.

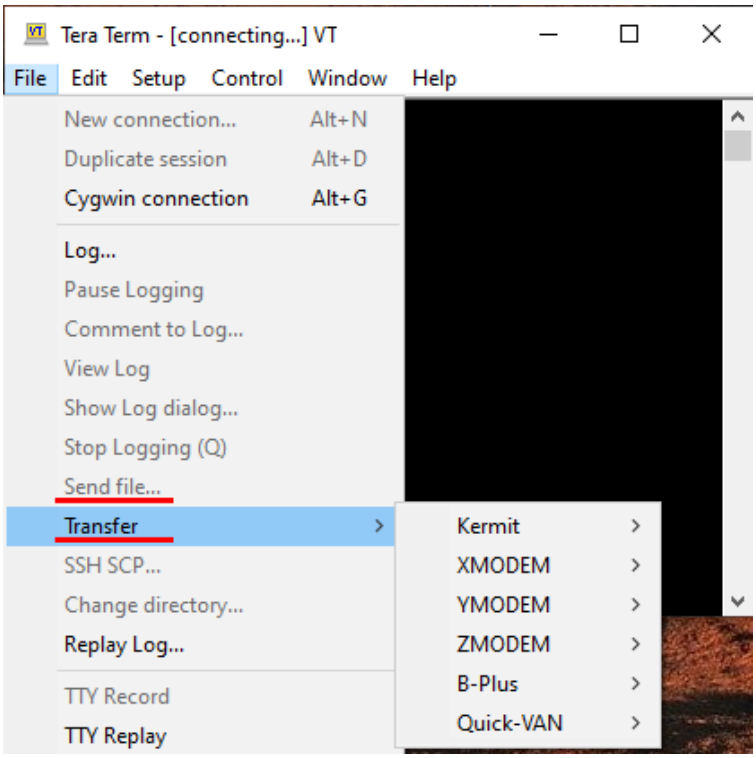


Fig. 1.22: Možnosti posielania súborov cez TeraTerm

2 Výsledky študentskej práce

Návrh dosky plošného spoja je prvým riešeným problémom, Navrhujú sa dve dosky plošného spoja, samotný prípravok a riadiaci modul.[1]

2.1 Návrh dosky plošného spoja modulu prípravku [1]

Prípravok obsahuje konektory pre jednotlivé testované moduly. Plánovaný spôsob používania je rozdelenie fungovania na jeden usecase pre test modulov dodávaných s kitom a druhý usecase pre pripojenie samotného vývojového kitu za účelom testovania správania sa jeho programu.

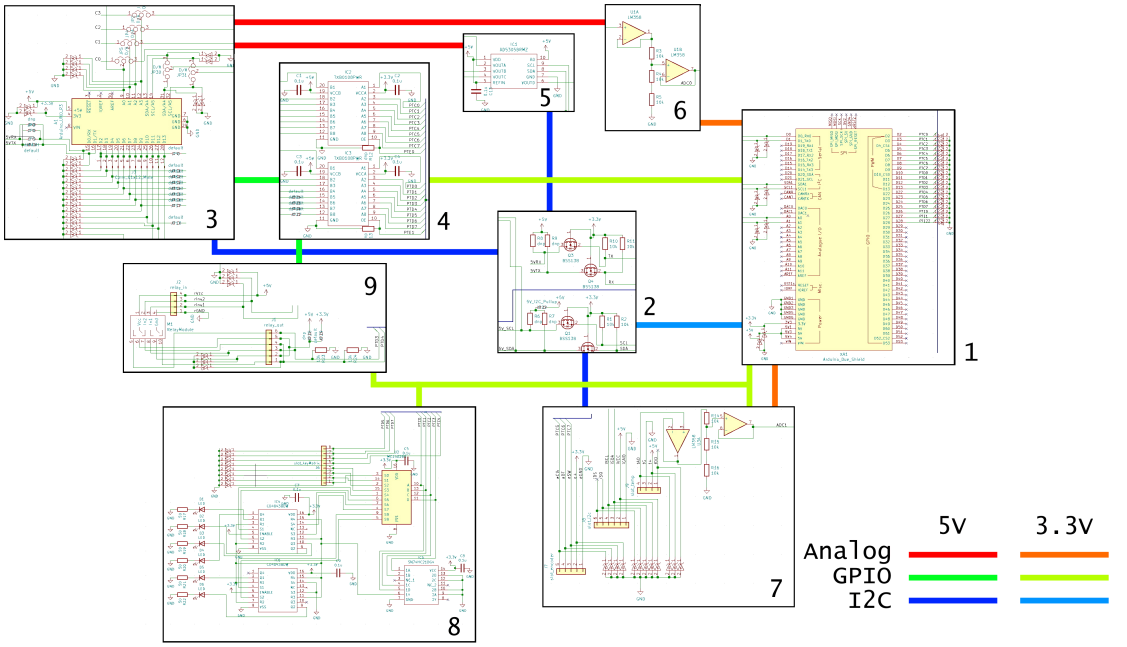


Fig. 2.1: Bloková schéma prípravku. Tabuľka v pravom dolnom rohu určuje význam prepojení.

Bloková schéma na ukážke 2.1 opisuje architektúru prípravku. Tá je rozdelená na nasledujúce bloky:

1. Zásuvka pre riadiaci modul,
2. Prevodník napätí i2c a uart rozhraní,
3. Zásuvka pre lcd modul a testovaný kit,
4. Prevodník digitálnych napäťových úrovní,
5. Digitálno-analógový prevodník,
6. Prevodník rozsahu signálu tlačidiel LCD modulu,

7. Časť pripájania i2c modulu, enkodéru a modulu s teplomerom.
8. Dekodér obsluhujúci maticovú klávesnicu a LED indikátory.
9. Časť pripájania relé modulu.

Zásuvka pre riadiaci modul

Riadiaci modul je tvarovo kompatibilný s Arduino DUE. Nie všetky piny sú ale využité, všetky využité piny sú chránené *TVS* diódami.

Prevodník napätí i2c a uart rozhraní

Pri tak nízkych rýchlostiach rozhraní aké sa vyskytujú v prípravku je implementácia rozhraní značne zjednodušená. Rýchlosť I2C je určená ako 100 kHz, pri takýchto rýchlostiach sú podstatné vlastnosti *rise-time*, *fall-time* signálu a kapacita zbernice.

Prevod napätí je zariadení v prípade *i2c* aj *uart* zbernice použitím obojstranného zapojenia s unipolárnym tranzistorom. Takéto zapojenie je vhodné pre zbernice ktorých kľudové napätie je rovné napájaciemu.

Ako tranzistor bol zvolený *BSS138* jedná sa o *jelly bean* súčiastku medzi MOSFET tranzistormi. Jedná sa o obohacovací unipolárny tranzistor s N-kanálom, prahovým napätím 1.5 V, kontinuálnym prúdom I_d 220 mA.

Zásuvka pre LCD modul a testovaný kit

LCD modul a vývojový kit majú spoločnú zásuvku keďže ich pripojenie zapadá do separátnych use-casov. V prípade testovania modulov bude v tejto zásuvke zapojený LCD modul. V prípade použitia pre testovanie správania programu výukového kitu, bude v celom prípravku pripojený len riadiaci modul a samotný kit *ATmega328P Xplained Mini* a prípravok bude emulovať správanie modulov s ktorými má program kitu interagovať.

Digitálne vstupy privedené z prevodníku úrovní sú vedené cez kolíkovú lištu a každý jeden cez svoj jumper, to umožňuje ľubovoľne meniť spôsob prepojenia digitálnych vstupov medzi riadiacou jednotkou a zásuvkou pre LCD modul a testovaný kit.

Prevodník digitálnych napäťových úrovní

Na prevod napätí pre relé modul a zásuvku pre lcd modul a testovaný kit je Použitý prevodník *TXB0108PWR* jedná sa o už spomenutý obojsmerný prevodník napäťových úrovní s automatickou detekciou smeru. Jeho reakcia na zmenu napäťovej úrovne má typickú dĺžku v rádoch ns a maximálny kontinuálny prúd výstupu je ± 50 mA.

Digitálno-analógový prevodník

Použitý prevodník je *AD5305BRMZ*, jeho napájacie napätie v prípravku je 5 V, prevodník komunikuje pomocou *I2C* rozhrania. *AD5305BRMZ* obsahuje 4 kanály ktoré sú cez prepájací jumper zavedené do zásuvkových pinov na ktorých má *ATmega328P Xplained Mini* analógové vstupy.

Prevodník rozsahu signálu tlačidiel LCD modulu

Jedná sa o aktívny napäťový delič využívajúci operačný zosilňovač *LM358* vstupný odpor zapojenia má veľkosť 4 GΩ. *LM358* je *jelly bean* súčiastka. Jedná sa o *unity gain stable* operačný zosilňovač kompatibilný s použitým napájacím napätím. Prevod analógových úrovní v tejto aplikácií je na presnosť a rýchlosť nenáročná úloha.

Časť pripájania i2c modulu , enkodéru a modulu s teplomerom

Enkodér s tlačidlom je možné napájať napätím 3.3 V a vďaka tomu je možné pripojiť ho priamo na vstupy riadiaceho modulu.

Modul s teplomerom je napájaný napätím 5 V a jeho analógový výstup je prevádzaný na rozsah 3.3 V rovnakým zapojením využívajúcim *LM358* ako prevodník rozsahu signálu tlačidiel LCD modulu. Digitálny výstup modulu s teplomerom je ignorovaný ale na má vlastnú LED diódu na module pre indikáciu.

I2C modul je napájaný 5V napätím a na tomto napätí pracuje aj jeho zbernica. Digitálny výstup generátoru štvorcovej vlny je prevádzaný na úroveň riadiacej jednotky pomocou *AD5305BRMZ* prevodníku.

Dekodér obsluhujúci maticovú klávesnicu a LED indikátory.

Maticová klávesnica a LED indikátory ktoré indikujú úspešnosť testov pre každý modul vyžadujú 10 digitálnych výstupov, použitím dekodéru sú obslúžené pomocou štyroch. Použitý je dekodér *MC14028B*.

Výstupy dekodéru s LED diódami sú pre zjednodušenie riadenia doplnené SR klopným obvodom *CD4043BDW* ktorého reset je riešený 4 vstupovým AND hradlom *SN74HC21DG4*, reset je vykonaný privedením všetkých vstupov dekodéru do stavu logickej 1.

Časť pripájania relé modulu

Všetky testované moduly vrátane *lcd* modulu sú pripájané na kolíkové lišty. Relé modul má výstup riešený svorkovnicou a nie je možné ho napojiť na kolíkovú lištu.

Pripojenie relé modulu je primárne realizované pogo konektormi a ako záložné riešenie je vyvedená kolíková lišta na pripojenie pripájacích káblov.

Spínacie kontakty relé modulu obsahujú *pulldown* rezistory. Pripojenie relé modulu zmení na jednom zo spínacích modulov logickú úroveň, čo umožňuje spínanie testovania modulov.

2.2 Návrh dosky plošného spoja modulu systémového jadra [1]

Systémové jadro je break-out board pre *mkl27z* ktorý je zároveň kompatibilný aj s ďalšími mikrokontrolérmi z rodiny *Kinetis*. Modul systémového jadra má pinout kompatibilný s *Arduino DUE* nezapája ale všetky piny. Použité rozhrania rešpektujú primárne rozloženie pinov.

Systémové jadro je univerzálny modul ktorý je použiteľný aj mimo túto aplikáciu. Modul je programovaný cez *SWD 10* rozhranie a využíva interný oscilátor. Interný oscilátor mikrokontroléru *mkl27z* umožňuje beh USB zbernice bez externého kryštálového oscilátoru. Kryštálový oscilátor 32 kHz slúži len ako zdroj hodín pre *Real-Time-Clock*.

2.2.1 Napájacie napätie

Modul získava napájacie napätia z *USB* zbernice. Obsahuje 3 separátne zdroje napätia.

Jeden *Low dropout* regulátor na získavanie 3.3 V, ten obsahuje rezistor a jumper slúžiaci na meranie výstupného prúdu regulátoru.

Ďalšie dva regulátory sú pripojené cez jumper a tak je možné ich úplne odpojiť, slúžia na stabilizáciu napätia 5 V. Jeden *Low dropout TPS7B8750QKVURQ1* a druhý spínaný regulátor *TPS63000* zobrazený na ukážkach 2.2 a 2.3.

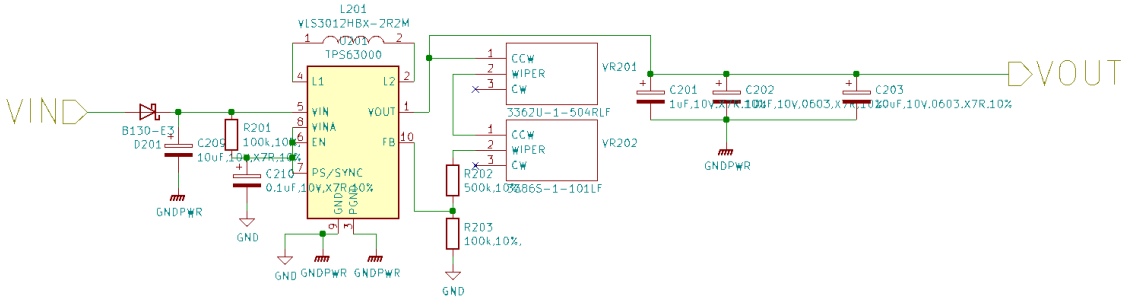


Fig. 2.2: Schéma regulátoru TPS63000.

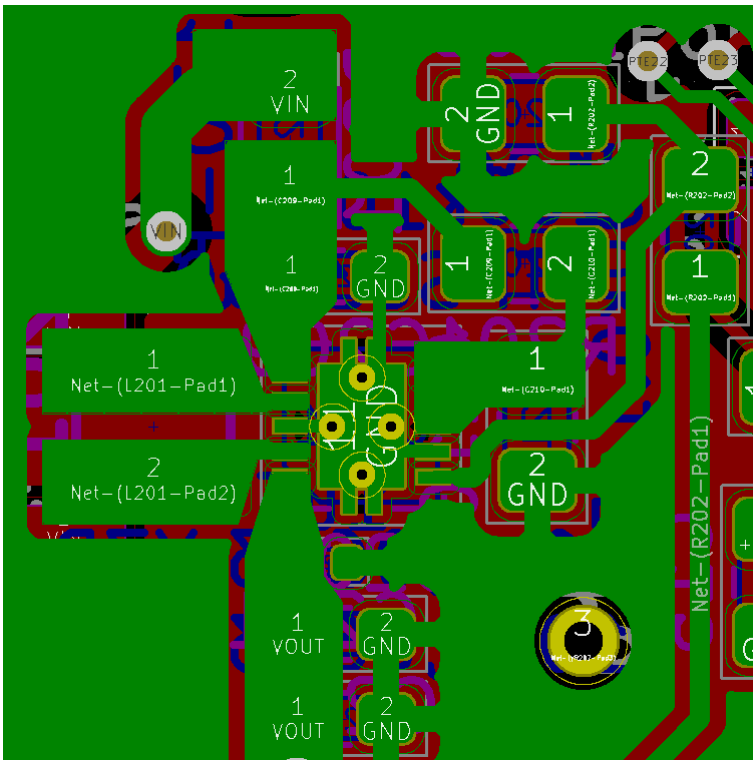


Fig. 2.3: Realizácia regulátoru TPS63000.

2.2.2 Vedenie USB rozhrania

Vedenie USB dátových vodičov je realizované rovnakým spôsobom ako firma *NXP* používa vo svojich *freedom* doskách ako napríklad aj v prípade*FRDM-K64F*. Zariadenie využíva dva 33 Ω rezistory a jeho diferenciálny pár je vedený ako koplanárny vlnovod. Spôsob realizácie USB rozhrania vidieť na ukážkach 2.4 a 2.5.

Ukážka 2.4 zároveň zobrazuje tvs diódy, feritovú tlmivku ktorá chráni pred nárazovým prúdom a rezistor jumper kombináciu na meranie napájacieho prúdu zariadenia.

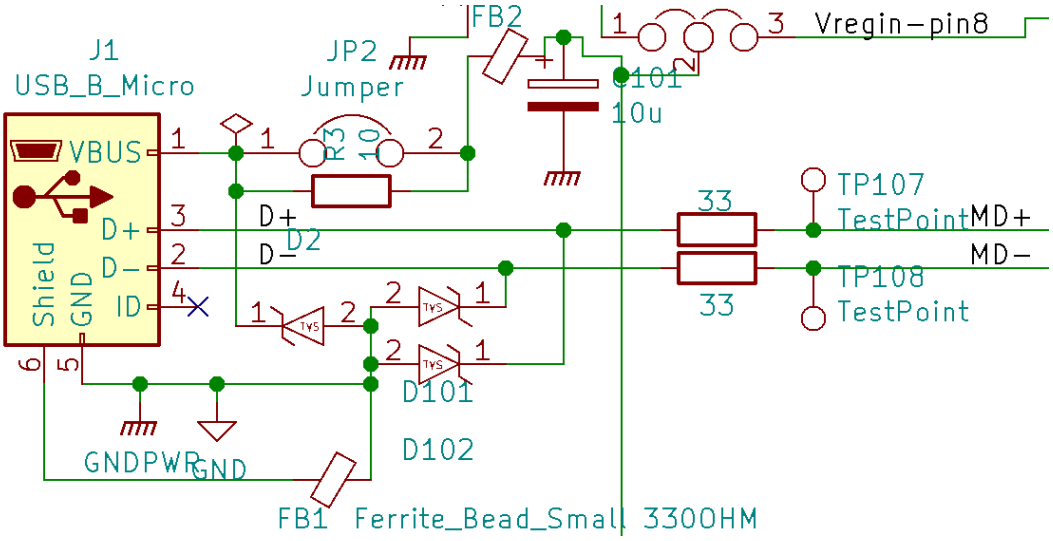


Fig. 2.4: Schéma USB rozhrania.

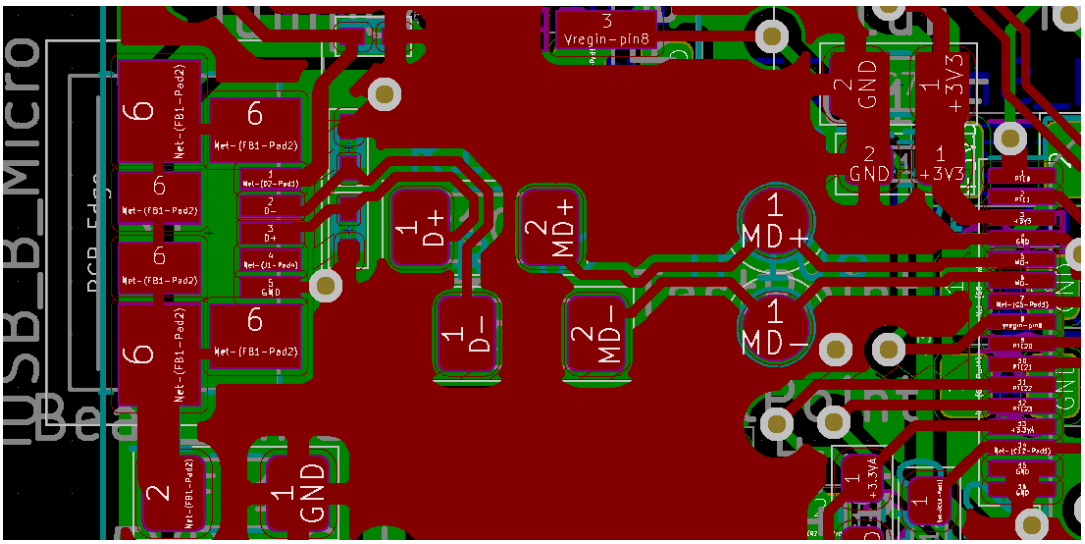


Fig. 2.5: Zapojenie USB rozhrania.

2.3 Procedúry testov HW modulov

Detailný popis testovacích procedúr pre jednotlivé moduly je opísaný v následujúcej časti.

Relé modul

Relé modul je ovládaný dvoma digitálnymi vstupmi Inv1 a Inv2 na ktorých je pred začatím testu a po jeho konci logická úroveň 0. Spínacia časť relé modulu je zapojená

nasledovne:

- Spoločné kontakty sú prepojené a sú zapojené na napájacie napätie 3.3 alebo 5 voltov,
- NC výstupy sú prepojené a sú zapojené na vedenie PTD4,
- NO výstupy sú prepojené a sú zapojené na vedenie PTD5.

Test modulu prebieha nasledovne:

1. Kontrola PLU(prítomnosti logickej úrovňe) 1 na NC a 0 na NO.
2. Privedenie log. 1 na vstup Inv1.
3. Kontrola PLU 1 na NO a NC.
4. Privedenie log. 1 na vstup Inv2.
5. Kontrola PLU 1 na NO a PLU 0 na NC.
6. Privedenie log. 0 na vstup Inv1.
7. Kontrola PLU 1 na NO a NC.

LCD modul

LCD modul pozostáva z LCD ktorého funkcionalitu spadá na užívateľa, jeho tlačidlá tiež vyžadujú akciu od užívateľa.

1. Výpis znakov do každého pola lcd.
2. Postupný test jednotlivých tlačidiel:
   (a) Test neutrálnej napäťovej úrovne.
   (b) AU(akcia užívateľa) - stlačenie tlačidla v poradí
   (c) Čakanie na ustálenie napäťovej úrovne
   (d) Kontrola správnosti napäťovej úrovne na základe predpísaného pásma pre dané tlačidlo.
   (e) AU - uvoľnenie tlačidla

Maticová klávesnica

1. Kontrola PLU 0 na všetkých riadkoch pri napájaní všetkých stĺpcov
2. AU - postupné stlačenie všetkých tlačidiel užívateľom, tlačidlá sú stláčané po riadkoch začínajúc horným. Tlačidlá sú v rámci riadku stláčané z ľava do prava.

Enkodér

Test pozostáva z akcií užívateľa ktoré sú vyhodnotené pomocou jednoduchej logiky na základe zmeny signálu CLK a stavu signálu TD počas tej zmeny.

1. AU - rotácia enkodéru v smere hodinových ručičiek
2. AU - rotácia enkodéru v protismere hodinových ručičiek
3. AU - stlačenie tlačidla

Teplomer

Testovanie modulu s teplomerom je komplikované nejednotnosťou modulov, čo umožňuje len rudimentárny test ktorý nemusí byť adekvátny alebo je nutný komplikovanejší a časovo náročný zásah užívateľa. Tento zásah zároveň vyžaduje nástroj, spočíva v nastavení kalibračného trimra na dosiahnutie správnej indikovanej teploty a následne dotyk čidla na zvýšenie indikovanej teploty a overenie fungovania aj mimo pracovný bod na ktorom bol modul nastavený. Kompromisom ktorý zjednodušuje proces je nechať samotné zhodnotenie fungovania čidla na užívateľa. LCD modul slúži na zobrazovanie meraných hodnôt.

1. AU - Nastavenie trimra na dosiahnutie smerodajnej hodnoty vypísanej na lcd module.
2. AU - Zahriatie čidla.
3. AU - Zhodnotenie funkčnosti na základe zmene vypisovanej hodnoty.

I2C modul

1. Nastavenie dátumu a času RTC.
2. Nastavenie generátoru obdĺžnikových pulzov.
3. Test prítomnosti pulzov.
4. Test správnosti hodnoty.
5. Prepis hodnôt EEPROM na logickú 1 na všetkých bitoch.
6. Kontrola správnosti hodnôt.
7. Prepis hodnôt EEPROM na logickú 0 na všetkých bitoch.
8. Kontrola správnosti hodnôt.

2.4 Problémy navrhnutej DPS

DPS navrhnutá vrámci tejto práce trpí na niekoľko problémov a nedostatkov. Niektoré sú následkom krízy v dostupnosti elektronických komponentov a kompromisov ktoré situácia vyžadovala. Pri výbere komponentov dochádzalo bežne k nedostupnosti vybraných komponentov a dokonca ich alternatív, mnohé súčiastky boli zvolené čisto na bázy toho čo bolo po ruke.

Jedny z takto zvolených komponentov boli operačné zosilňovače LM358, ktoré by bolo v prípade dostupnosti alternatív lepšie nahradiť vhodnejšími. Na tieto zosilňovače sú kladené nároky ktoré vyžadujú fungovanie v režime napäťového sledovača.

Ďalšími komponentmi ktoré boli volené neoptimálne je niekoľko kondenzátorov ktoré neboli v potrebnom čase dostupné vo variante s ideálnym dielektrikom pre danú aplikáciu.

Niekoľko komponentov nebolo možné zohnať, jedným z nich ne predlžovacia DPS zásuvka ktorá funguje z jednej strany ako zásuvková rada a z druhej ako kolíková lišta, tá mala byť osadená do miesta pre kit kde by z jednej strany slúžila na pripojenie lcd modulu a z druhej na pripojenie kitu.

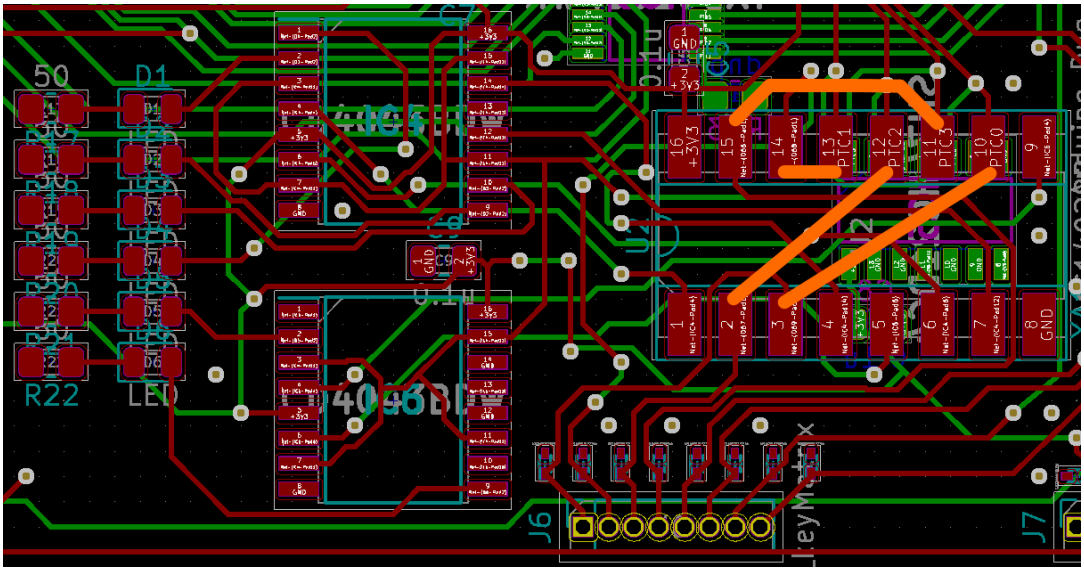


Fig. 2.6: Obvod LED indikátorov, komponenty z ľava do prava:1. rezistory 2. LED diódy 3. RS klopné obvody 4. dekodér. Na obvode dekodéru sú znázornené prepoje ktorými je možné ho nahradiť pre zfunkčnenie obvodu maticovej klávesnice bez jeho použitia.

Ďalším komponentom ktorý bol dodaný ale v nesprávnom púzdre bol dekodér ktorý slúži na ovládanie obvodu LED indikátorov a stĺpcov maticovej klávesnice. Nahradenie chýbajúceho dekodéru prepojmi ktoré prepoja jeho pôvodné vstupy priamo na obvod stĺpcov maticovej klávesnice, toto prepojenie je znázornené oranžovými čiarami na ukážke 2.6. Funkčnosť LED indikátorov je plne nahraditeľná sériovou komunikáciou a výpismi v konzolovom okne. Spôsob zfunkčnenia samotných LED indikátorov bez použitia dekodéru je možné rôznymi spôsobmi. Najjednoduchší z nich by bol pridanie káblových spojov na voľné výstupy jadra na prípravku, tieto spoje by boli prispájkované v najjednoduchšom prípade na anódu LED (Elektróda vpravo) a voľný pin na prípojke do ktorej sa pripája jadro(pin závisí od použitého jadra).

V prípade jadra bolo pre nedostupnosť mikrokontroléru so správnou veľkosťou pamäti nutné improvizovať. Jediné dostupné dva kusy boli už naspájkované na iných DPS. Oživiť mikrokontrolér jadra sa ale nepodarilo napriek tomu že jadro návrhovo vychádza z overeného návrhu obvodu. Pri snahe programovať MCU jadra

dochádzalo k chybe. S jedným MCU nebolo možné pripojiť sa na programovacie rozhranie. Napájacie napätie MCU bolo na všetkých vstupoch stabilné a malo správnu úroveň. Napätie RESET prívodu nevykazovalo anomálne správanie pri snahách MCU programovať. Druhé MCU po naspájkovaní do obvodu nebolo programátorom vôbec detekované.

Príčina nefunkčnosti jadra nebola zistená, nebolo ani zistené či dôvod nefunkčnosti MCU pochádza z DPS alebo z MCU samotných. Tie boli počas svojho života vystavované vplyvom ktoré ich mohli zničiť ak napr. tepelné namáhanie pri prespájkovaní alebo ESD.

DPS prípravku pozostáva zo štandardných zapojení použitých súčiastok ktoré sú popísané v ich katalógových listoch. Test s vybranými napäťovými úrovňami ukázal že prevodníky digitálnych úrovní fungujú. Prevodník analógového napätia pre výstup tlačidla lcd modulu je funkčný, ten pre modul teplomeru momentálne nie. Keďže sa jedná o rovnaký obvod ale s inou súčiastkou od iného výrobcu, pravdepodobne je tento problém riešiteľný výmenou operačného zosilňovača.

2.5 Migrácie na iné jadro

SW architektúra je vytvorená tak aby umožnila jednoduchú portovateľnosť na akúkoľvek platformu ktorá je na úlohu adekvátna. Zásahy na DPS prípravku samotného sú nutné len v prípade migrácie na jadro ktoré funguje na inom pracovnom napätí.

2.5.1 Konverzie pre podporu 5 V jadra

Prípravok bol navrhnutý primárne pre 3.3 V jadro, v prípade migrácie na jadro ktoré pracuje na napäťovej úrovni piatich voltov je nutné vykonať niekoľko opatrení. Tieto opatrenia je nutné vykonať na nasledovných obvodoch:

- Prevodník i2c rozhrania.
- Prevodník sériového rozhrania.
- Prevodníky napäťových úrovní GPIO rozhrania.
- Prevodník úrovňe napätia tlačidla lcd modulu.
- Prevodník úrovňe napätia výstupu modulu teplomeru.
- LED indikátory.

Zmeny ktoré vyžadujú odpájanie súčiastky a prepojenie kontaktov jej obvodu na DPS sú vyznačené oranžovou farbou. Každá oranžová čiara znázorňuje prepojenie ktoré je nutné vytvoriť medzi plôškami DPS ktorých sa jej konce dotýkajú.

Modrou farbou sú prekrížené súčiastky ktoré majú byť vrámci konverzie len odstránené.

Ukážky 2.7 a 2.8 zobrazujú miesta na ktorých sa na DPS prípravku nachádzajú obvody ktoré je v prípade zmeny napätia použitého jadra nutné vykonať.

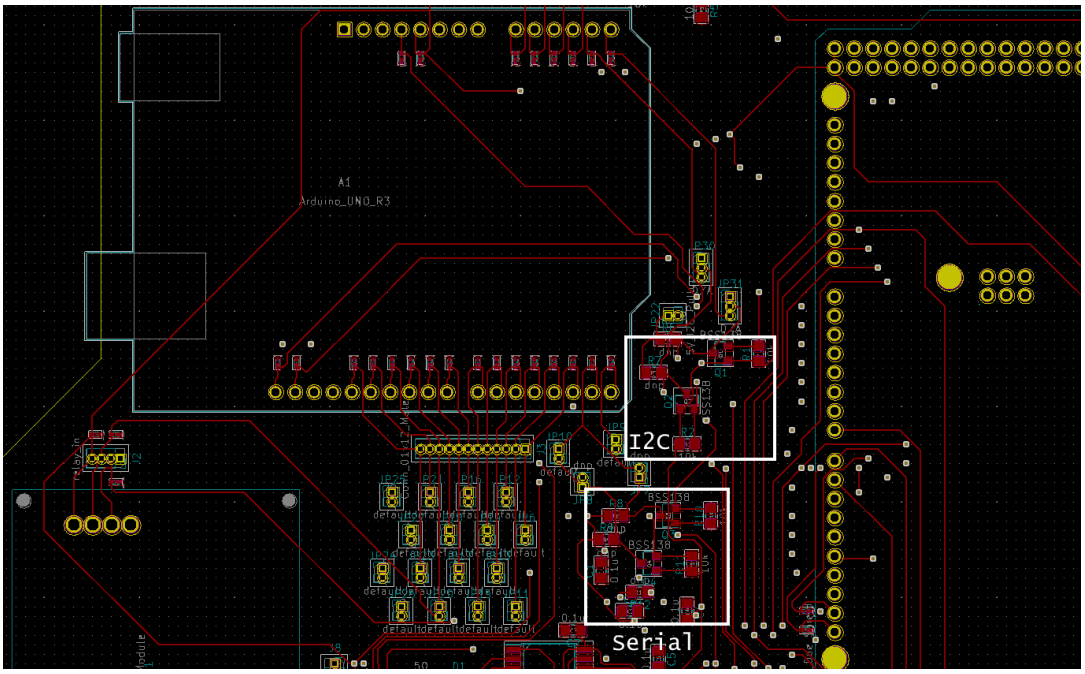


Fig. 2.7: Umiestnenie i2c a sériového prevodníku napätí.

Ukážka 2.9 zobrazuje schému prevodníkov napäťových úrovní sériového a i2c rozhrania. Z fungovania tohoto obvodu vyplýva že pre konverziu je nutné odstrániť pull-up rezistory 3.3 V časti obvodu. Ďalej je nutné odpájať unipolárne tranzistory a nahradiť ich prepojom. Umiestnenie súčiastok na PCB a spôsobu prepojenia ktorým je nutné tranzistory nahradiť sú zobrazené na ukážkach 2.10 a 2.11.

Konverzný úkon prevodníku digitálnych úrovní je zobrazený na ukážke 2.12.

Konverzné úkony na prevodníku úrovňe napätia tlačidla lcd modulu a prevodníku úrovňe napätia výstupu modulu teplomeru spočívajú len v odpájaní duálneho operačného zosilňovača a prepojenia vstupu a výstupu jeho obvodu. Toto prepojenie môže byť pre jednoduchosť vedené aj medzi plôškami 6 a 3. Všetky pomocné súčiastky oboch obvodov sú odstránením púzdra operačných zosilňovačov vypojené.

2.6 SW architektúra

Pre problémy s HW jadra a situáciou ktorá znemožnila nájsť náhradu včas, bolo úsilie smerované na architektúru a cross-platformovú časť kódu. Táto časť kódu pozostáva z nasledovných súborov:

- scriptingInterface

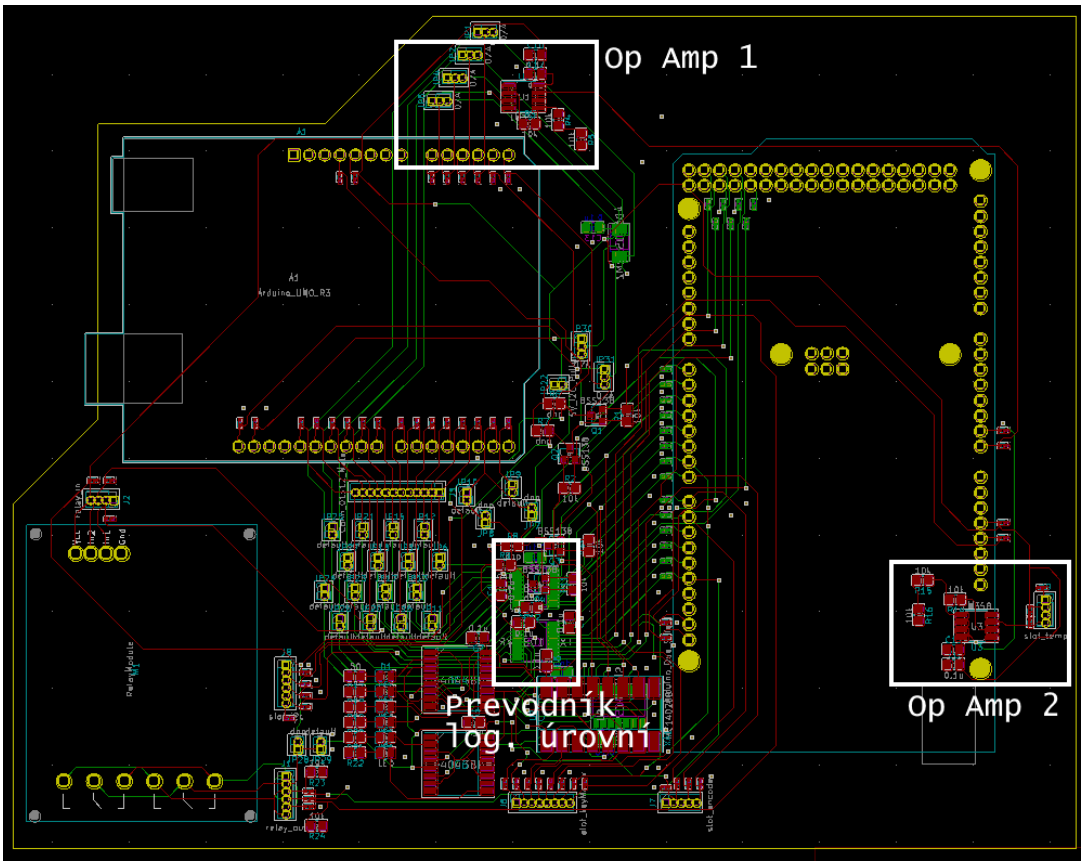


Fig. 2.8: Umiestnenie prevodníku digitálnych a analógových napätí.

- HWTest
- highlevelSettings
- fixtureBrd
- LowLevelWrapper

súbory scriptingInterface.c a ScriptingInterface.c slúžia na obsluhu Lua virtuálneho stroja. ScriptingInterface obsahuje definície funkcií ktoré slúžia ako wrapper na volanie C funkcií Lua funkciami, obsahuje C funkciu *lua_State* luainit(void)* ktorá vytvorí Lua kontext, importuje základné Lua knižnice a namapuje Lua funkcie s C funkciami. Táto funkcia je volaná funkciou *int luamain(char* payload, int payloadLen, int error, int intArg, int event)* ak Lua kontext neexistuje v čase volania.

Funkcia *int luamain(char* payload, int payloadLen, int error, int intArg, int event)* slúži na spustenie Lua virtuálneho stroja, kde argument payload slúži na dodanie Lua binárneho kódu na spustenie, payLoadLen má obsahovať jeho dĺžku a event vyberá udalosť ktorá sa má v Lua virtuálnom stroji počas behu volať. Momentálne je vytvorená len jedna, ich pridávanie je ale jednoduché. Súbor usb_device_interface_0_cic_ obsahuje príklad načítania a spustenia Lua kódu, ten sa nachádza na rozsahu riadkov

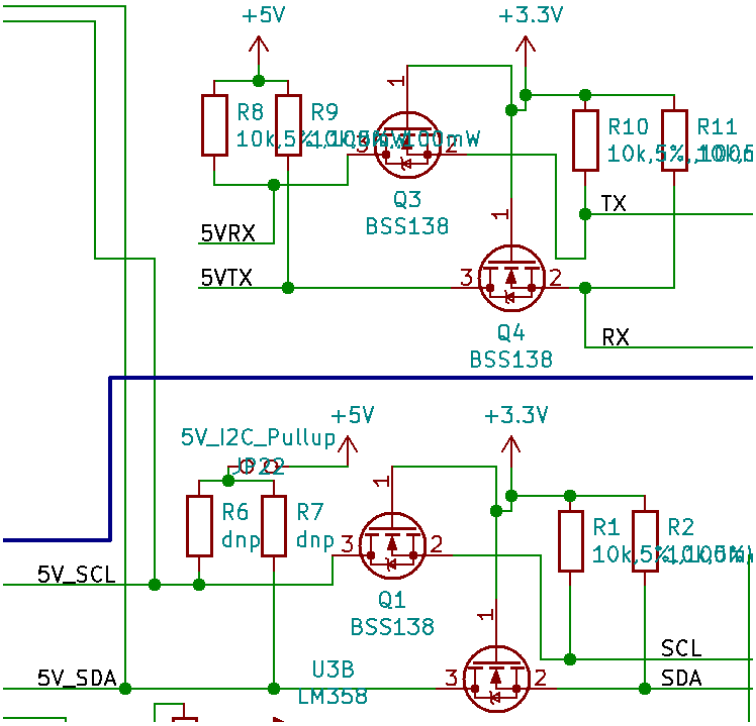


Fig. 2.9: Schéma I2C a sériového prevodníku napäťových úrovní.

191 až 299.

HWTest obsahuje kód pre testovanie modulov obsiahnutých v kite. Obsahuje jednu funkciu v ktorej je hard-code na skúšanie modulov. Je možné volať túto funkciu na test modulov alebo ju nahradiť vlastnou, prípadne použiť prístup písania testov v Lua.

súbor highlevelSetting obsahuje len definície na prepínanie medzi zvolenými high level riešeniami (Prepínanie medzi Lua, Custom jazyka a Hard-Code prístupu) z ktorých je pripravená implementácia pre Lua a Hard-Code.

Súbor fixtureBrd obsahuje funkcie na prácu z HW ktorý sa nachádza na prípravku a je nezávislý od voľby jadra. To umožňuje jeho implementáciu v podobe cross-platformového kódu.

Súbor LowLeverWrappers obsahuje wrapery ktoré je pri portovaní a implementácií jadra definovať tak aby volali platformovo závislé funkcie na obsluhu periférií jadra. Definície funkcií tohoto súboru vychádzajú z tých aké sa bežne používajú v API rôznych platforiem. Názvy obsiahnutých funkcií sú väčšinou samo-opisné. Funkcie čo začínajú slovom COM_ slúžia na komunikáciu s PC cez sériové rozhranie, či sa jedná o VCOM alebo iné. Mapovanie pinov a periférnych zariadený je v HW také ako zodpovedá platforme arduino Due, (zobrazené na ukážke 2.15) a pri implementácií je nutné dbať na to aby čísla pinov boli v súlade s použitou predlohou.

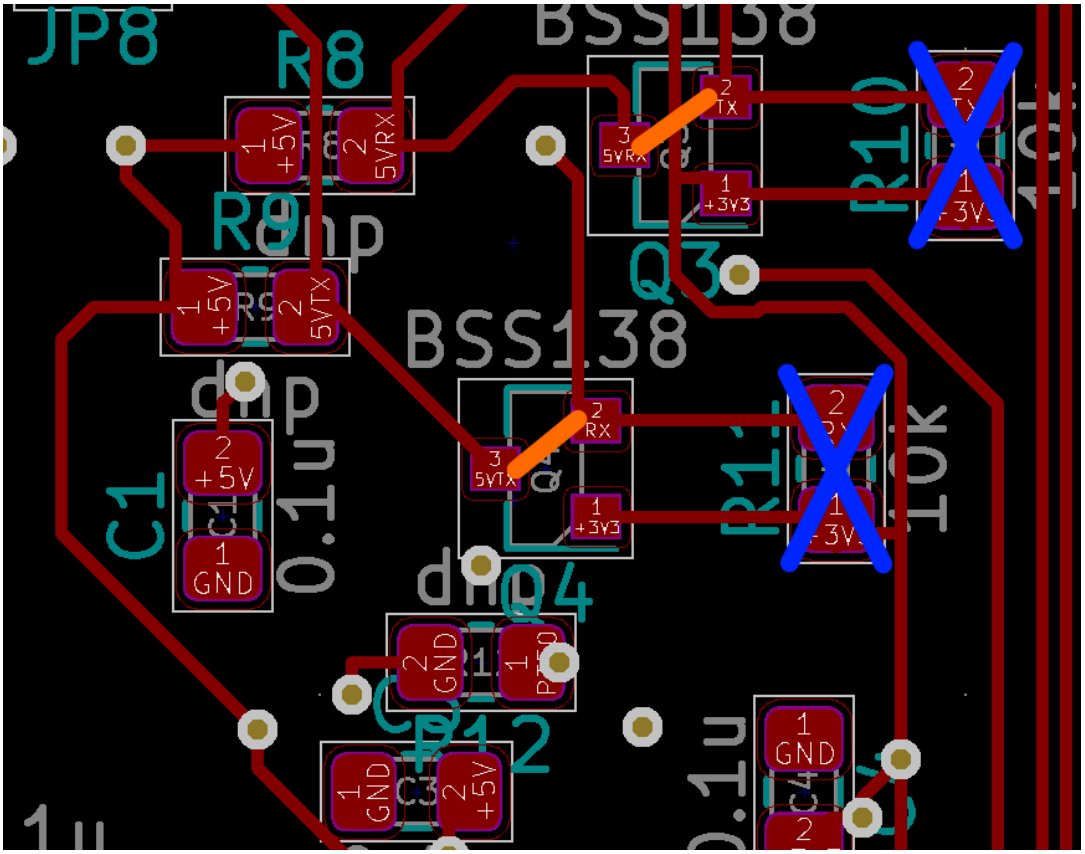


Fig. 2.10: Konverzné úkony na prevodníku napätí sériového rozhrania medzi jadrom a kitom.

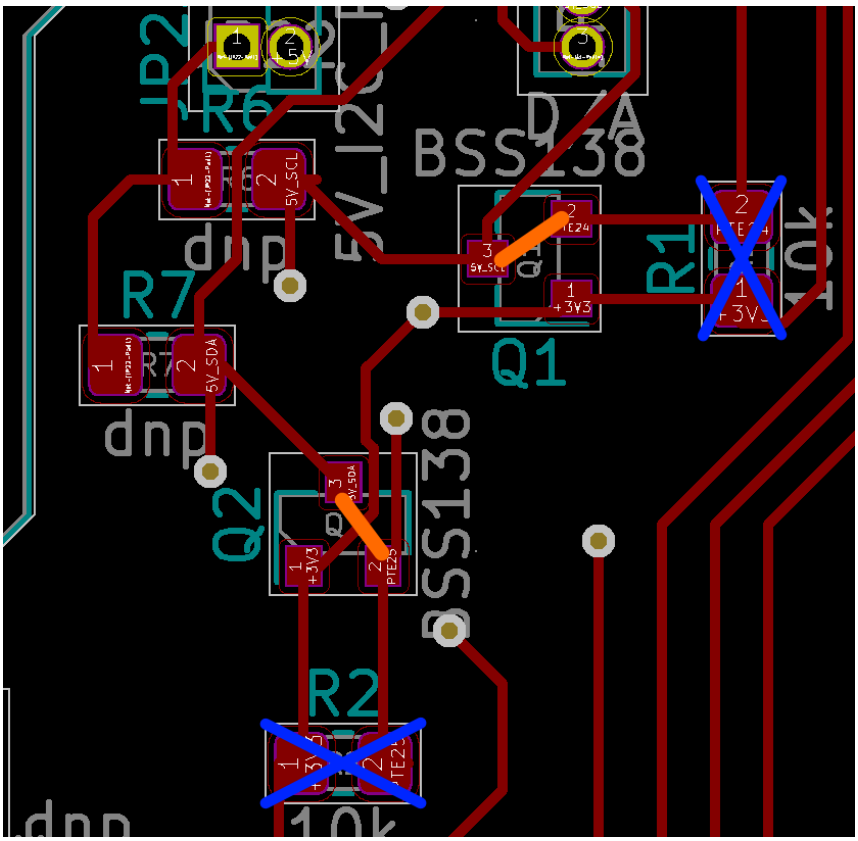


Fig. 2.11: Konverzné úkony na prevodníku napätí i2c rozhrania medzi jadrom a kitom.

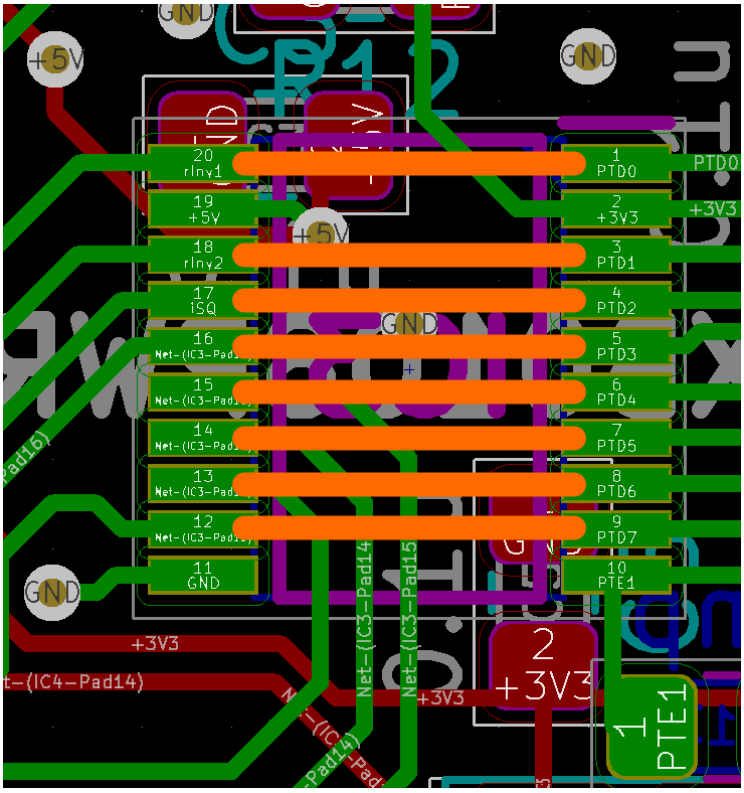


Fig. 2.12: Konverzné úkony na prevodníku napäťových úrovní pre GPIO.

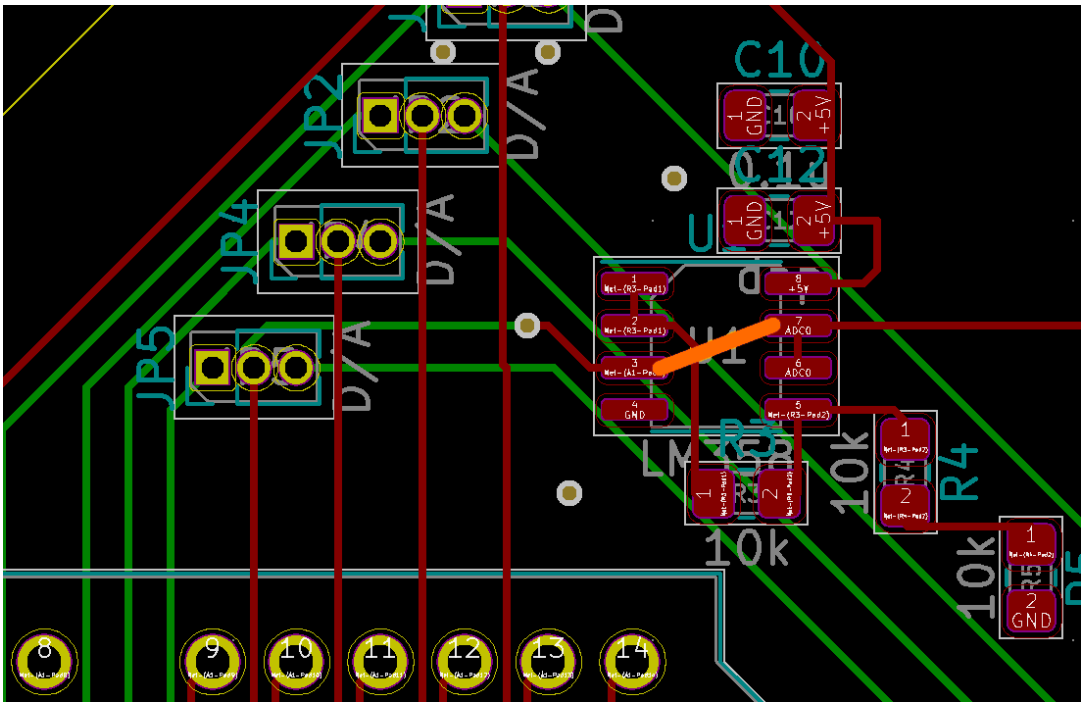


Fig. 2.13: Konverzný úkon na prevodníku analógového výstupu tlačidiel lcd modulu.

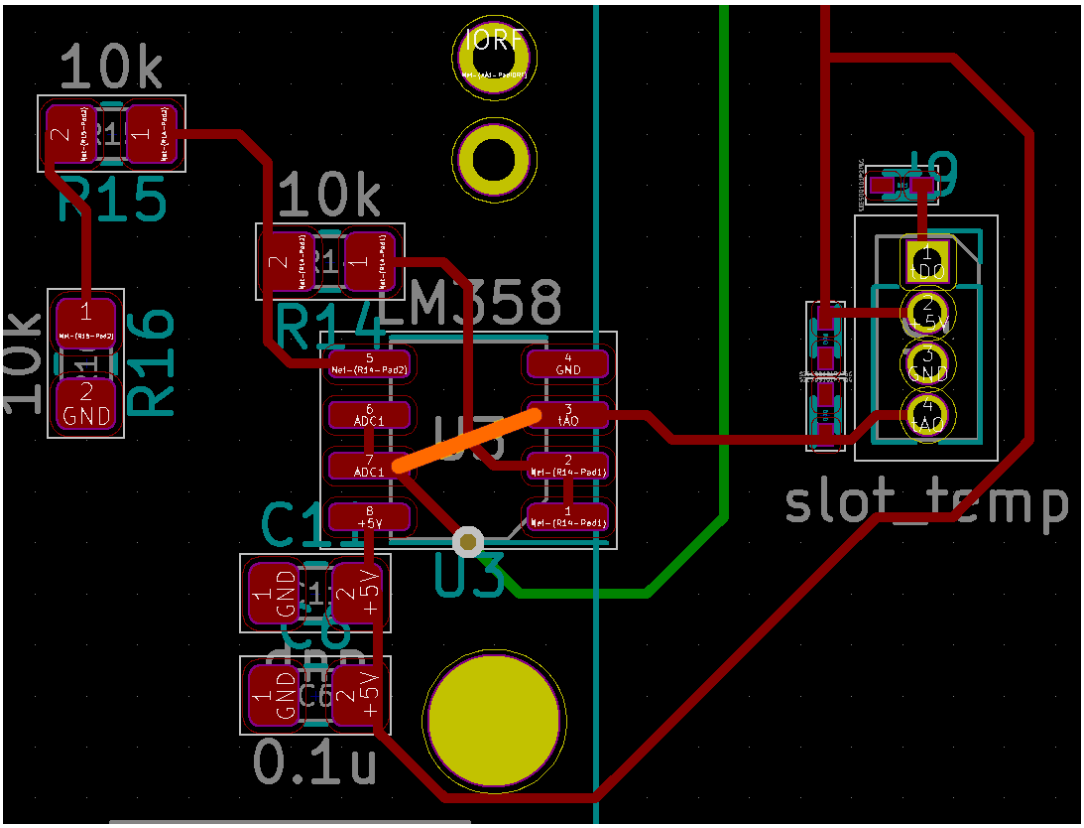


Fig. 2.14: Konverzný úkon na prevodníku výstupu modulu teplomeru.

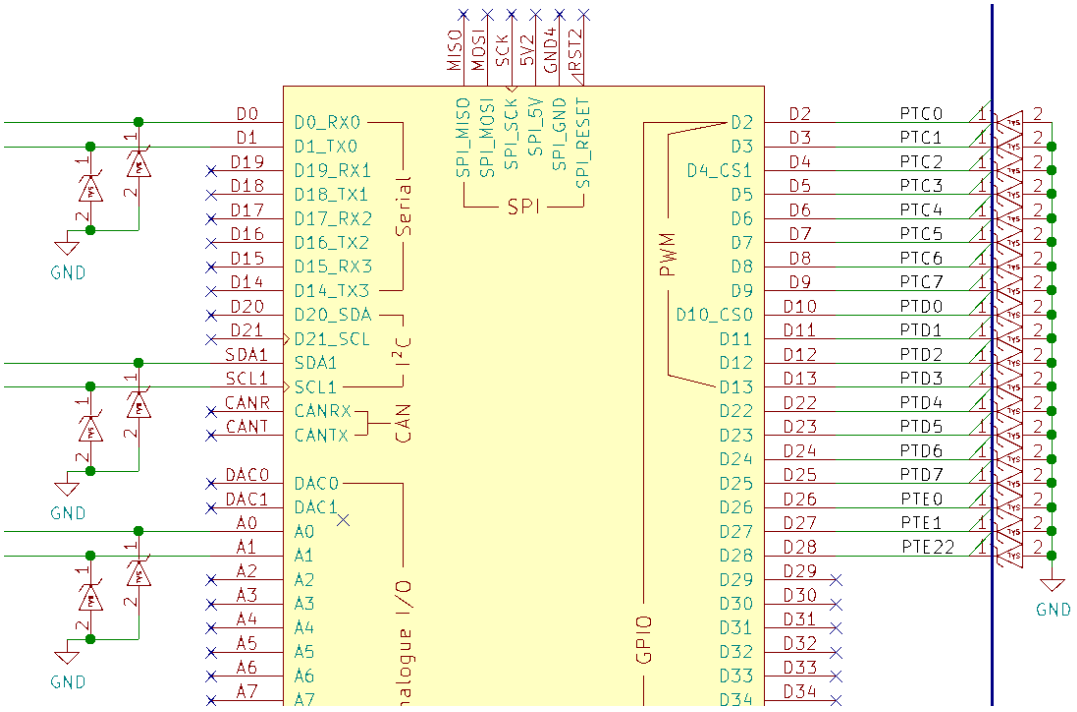


Fig. 2.15: Detail vývodov zásuvky pre modul jadra.

Záver

Táto práca sa zaoberá návrhom Dosky plošného spoja testovacej stolice pre periférne zariadenia mikrokontrolérového kitu ATmega328P Xplained Mini. Práca zároveň rozoberá rôzne možnosti riešenia problémov vyskytujúcich sa v tejto aplikácií.[1]

Samotný návrh HW je urobený modulárne a umožňuje implementáciu úlohy overovania fungovalia SW implementovaného na ATmega328P Xplained Mini emuláciou jeho periferných zariadení.[1]

Navrhnuté boli oba moduly, prípravok aj jadro systému. Samotné jadro systému.[1]

HW prípravku je až na niektoré neoptimálne zvolené alebo nedostupné súčiastky funkčné. Vrátane najdôležitejších obvodov ktoré slúžia na prevod napäťových úrovní. Pre plnú funkčnosť je nutné prispájkovať niektoré ďalšie komponenty ktoré nebolo možné napriek značnému úsiliu obstarať. Obvod LED indikátorov nebol zfunkčnený pre chýbajúci dekodér, jedná sa ale len o menej dôležitý modul keďže celá jeho funkcionalita je suplovateľná komunikáciou s PC cez konzolovú sériovú aplikáciu.

HW jadra systému sa ukázal ako nefunkčný, nebolo ale možné rozlíšiť či sa jedná o závadu na DPS alebo mikrokontroléry ktorý bol v oboch prípadoch získaný odpájaním z už používanej DPS z iného projektu.

SW architektúra bola vypracovaná pre migrovateľmnosť a možnosť implementovať zmeny jednoducho. Pre nemožnosť zohnať náhražku za nefunkčné jadro, bola spracovaná len cross-platformová časť C kódu ktorá vyžaduje pri migrácií len mierne úpravy.

Táto architektúra je smerovaná pre štyri varianty takzvaného high-level riešenia (riešenia otázky menenia správania zariadenia na riešenie odlišných úloh). Implementácia Lua prístupu a hard-code prístupu bola do detailu vypracovaná, pričom je možné do architektúri implementovať aj riešenie z vlastným jazykom na mieru alebo sprostredkovaním udalostí. Pri migrácií na iné jadro systému je nutné len implementovať platformovo závislú nízku vrstvu ktorá spravuje HW a úlohu načítania binárneho kódu LUA, ukážka je ale súčasťou súboru ktorý sa nachádza v priloženom projekte.

Bibliography

[1] *HATALA, Branislav. Testovací stolice pro mikrokontrolérové kity* [online]. Dostupné z URL:
<https://www.vutbr.cz/studenti/zav-prace/detail/138256>.

[2] *datasheet TXB0108 8-Bit Bidirectional Voltage-Level Translator with Auto-Direction Sensing and ±15-kV ESD Protection* [online]. Dostupné z URL:
<https://www.ti.com/lit/ds/symlink/txb0108.pdf?ts=1640067626682&ref_url=https%253A%252F%252Fwww.google.com%252F>.

[3] *TMUX721x 44 V, Low-RON, 1:1 (SPST), 4-Channel Precision Switches with Latch-Up Immunity and 1.8-V Logic* [online]. Dostupné z URL:
<https://www.ti.com/lit/ds/symlink/tmux7211.pdf?ts=1640180739216&ref_url=https%253A%252F%252Feu.mouser.com%252F>.

[4] *TPS7B81-Q1 150-mA, Off-Battery, Ultra-Low-IQ (3-µA), Low-Dropout Regulator* [online]. Dostupné z URL:
<https://www.ti.com/lit/ds/symlink/tps7b81-q1.pdf?ts=1640321934218&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTPS7B81-Q1>.

[5] *TPS6300x High-Efficient Single Inductor Buck-Boost Converter With 1.8-A Switches* [online]. Dostupné z URL:
<https://www.ti.com/lit/ds/symlink/tps63000.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1640344078925&ref_url=https%253A%252F%252Feu.mouser.com%252F>.

[6] *Low Capacitance ESD Protection Diodes for High Speed Data Lines* [online]. Dostupné z URL:
<https://eu.mouser.com/datasheet/2/308/1/ESD9101_D-2311464.pdf>.

[7] *Lua* [online]. Dostupné z URL:
<https://www.lua.org/about.html>.

Symbols and abbreviations

V_{BR}	Breakdown voltage
USB	Universal serial bus

List of appendices

A Obsah elektronické přílohy

- `/` ..... kořenový adresář přiloženého archivu
  - `PCB` ..... Súbory dosky plošného spoja (KiCad)
    - `jadro` ..... Systémové jadro
      - `sym-lib-table` .....
      - `RegTPS63000.sch` .....
      - `RegTPS7B87.sch` .....
      - `PWR-mkl27z-rescue.lib` .....
      - `PWR-mkl27z-rescue.dcm` .....
      - `PWR-mkl27z-cache.lib` .....
      - `PWR-mkl27z.xml` .....
      - `PWR-mkl27z.sch` ..... hlavná schéma
      - `PWR-mkl27z.pro` ..... súbor projektu
      - `PWR-mkl27z.net` .....
      - `PWR-mkl27z.kicad_pcb` ..... súbor dosky plošného spoja
      - `arduino.lib` .....
      - `fp-info-cache` .....
      - `sym-lib-table` .....
      - `TestBed.csv` .....
      - `TestBed.kicad_pcb` ..... súbor dosky plošného spoja
      - `TestBed.net` .....
      - `TestBed.pro` ..... súbor projektu
      - `TestBed.sch` ..... hlavná schéma
      - `TestBed.xml` .....
      - `TestBed-cache.lib` .....
      - `TestBed-rescue.dcm` .....
      - `TestBed-rescue.lib` .....
      - `XplainedAndLcdSlot.sch` .....
  - `Firmware` ..... adresár projektu SW modulu jadra
  - `PC-SW` ..... adresár obsahujúci utilitu na konverziu Lua binárneho súboru na prenostieľný formát