VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2016 Bc. Jan Macháček

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

ZAŘÍZENÍ PRO AUTOMATIZOVANÉ TESTOVÁNÍ HARDWARU
AUTOMATED HARDWARE TESTING EQUIPMENT

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR
Bc. Jan Macháček

VEDOUCÍ PRÁCE
SUPERVISOR
Ing. Ondřej Krajsa, Ph.D.

BRNO 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Jan Macháček ***ID:*** 146895
Ročník: 2 ***Akademický rok:*** 2015/16

NÁZEV TÉMATU:

Zařízení pro automatizované testování hardwaru

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte zařízení pro automatizované testování hardwaru. Zařízení bude umožňovat připojení externích měřicích přístrojů přes USB sběrnici. Naměřená data se budou odesílat na externí server a zobrazovat na webovém rozhraní.

DOPORUČENÁ LITERATURA:

[1] GIOMETTI, Rodolfo. BeagleBone Essentials. Packt Publishing, 2015. ISBN 9781784393526

[2] SADIQ, H M Irfan. Using Yocto Project with BeagleBone Black. Packt Publishing, 2015. ISBN 9781785289736.

[3] HUGHES, John M. Real world instrumentation with Python. 1st ed. Beijing: O´Reilly, 2010, xxi, 595 s. : il. ISBN 978-0-596-80956-0.

Termín zadání: 1.2.2016 ***Termín odevzdání:*** 25.5.2016

Vedoucí práce: Ing. Ondřej Krajsa, Ph.D.
Konzultant diplomové práce:

doc. Ing. Jiří Mišurec, CSc., *předseda oborové rady*

ABSTRAKT

Tato práce se zabývá návrhem systému pro automatické testování hardwaru na platformě Beagle Bone Black. Cílem této práce je vyvinout komplexní zařízení, díky kterému bude možné otestovat funkcionalitu obecně elektronických zařízení v mezi stupni dokončení. K tomuto účelu využít externích měřících přístrojů a navrhnout HW, pro jejich ovládání.

KLÍČOVÁ SLOVA

Beaglebone Black, expanzní deska, ovládání měřících přístrojů, testování hardwaru

ABSTRACT

This thesis is focused on the design of system for automated testing of hardware on Beagle Bone Black platform. The purpose of this thesis is to develop a complex device which will be able to test the functionality of general electronic devices before completion. Use the external measuring devices and design hardware to control them.

KEYWORDS

Beaglebone Black, expansion board, control of measuring instruments, test hardware

MACHÁČEK, Jan *Zařízení pro automatizované testování hardwaru*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 67 s. Vedoucí práce byl Ing. Ondřej Krajsa, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Zařízení pro automatizované testování hardwaru“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu Diplomové práce panu Ing. Ondřeji Krajsovi, Ph.D. za odborné vedení, konzultace a podnětné návrhy k práci.

Brno ....

podpis autora(-ky)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Technicka 12, CZ-61200 Brno
Czech Republic
http://www.six.feec.vutbr.cz

PODĚKOVÁNÍ

Výzkum popsaný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno . . . . . .

. . . . . .
podpis autora(-ky)

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Tato práce se zabývá návrhem systému pro automatické testování hardwaru. Cílem této práce je vyvinout komplexní zařízení, díky kterému bude možné otestovat funkcionalitu obecně elektronických zařízení v mezi stupni dokončení. S tímto problémem se dnes potýká valná většina menších a středních firem, které se zabývají vývojem elektroniky. Tyto firmy většinou nemají vlastní sériovou výrobu a tak oslovují subdodavatele, kteří pro ně zajišťují výrobu DPS a osazování součástkami. Při výrobě však velmi často dochází k chybám, studeným spojům či k osazení součástky s chybnou jmenovitou hodnotou nebo k neosazení vůbec. Proto je nutné, pokud chce firma minimalizovat počet reklamací, po výrobě osazené DPS testovat. Typicky pro menší série na míru mívají tyto desky řadu specifik a tak běžně tito subdodavatelé nejsou schopni testování zajistit. Pokud však ano, bývá to často velmi drahé a takto zhotovené zařízení není v drtivé většině dále využitelné. Proto vznikla myšlenka navrhnout systém, který bude schopen pokrýt výše popsané testování, bude dostatečně modulární, aby pokryl co nejvíce možných testů. Hlavní podmínkou bylo využít některou z platforem linuxových minipočítačů (Raspberry pi nebo Beagle Bone Black) ve spojení s opensource softwarem.

Využití

Jak už bylo uvedeno v úvodu, zařízení je koncipováno především pro použití v menších firmách nebo pro specifické testování, kde potenciální série nepřesahují tisíce kusů. Důležitým aspektem je, že zařízení ke svému testování využívá běžné přístrojové měřící přístroje, připojené k Beaglebone Black pomocí USB. Díky tomu je zařízení výrazně levnější než konkurence a s přesností vyplývající z připojených přístrojů. Výsledné zařízení by pak mělo být nasazeno právě například u subdodavatele, kde jej budou obsluhovat pracovníci výroby a kde bude testovat právě vyrobená zařízení. Samotný test však musí vyhotovit programátor podle předpisu designera. K tomu by měla stačit základní znalost jazyku Python s využitím předpřipravených tříd a jejich metod. Díky tomu by mělo být možné vytvořit téměř jakýkoliv testovací scénář.

Požadavky

- Celý systém musí být založený na platformě Beaglebone Black.
- Navrhnout vhodné periferie pro testování vnějších signálů.
- Schopnost zařízení komunikovat s testovanými zařízeními pomocí nejběžněji používaných sběrnic.
- Měření a testování realizovat pomocí externích měřících přístrojů.
- Navrhnout vhodné ovládání přes webové rozhraní.

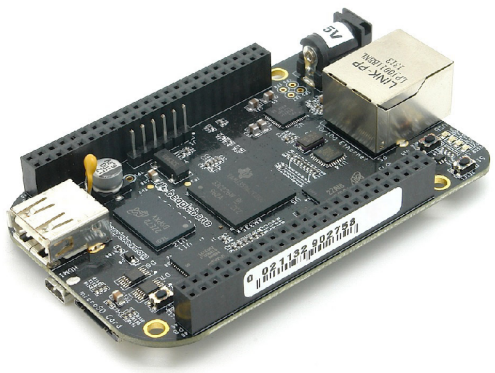
1 TESTOVACÍ PLATFORMA

1.1 Beglebone Black

Jak je uvedeno v úvodu, Beaglebone Black je základním prvkem celého zařízení. Jedná se o malý jednočipový plnohodnotný počítač založený na architektuře ARM. Beaglebone Black byl vybrán z důvodu nízké ceny a velké univerzálnosti použití. Důvodem, proč byla dána přednost Beagle Bone Black před Raspberry Pi, je větší počet GPIO pinů a větší množství dostupných sběrnic. Další výhodou Beaglebone Black je, že obsahuje 2 PRU 32 bitové mikro kontroléry, které mohou být využité například pro komunikaci se zařízením, které používá nějaký nestandardní protokol. Beaglebone Black má následující hardwarové parametry:

Procesor	AM335x 1GHz ARM® Cortex-A8
Operační paměť:	512MB DDR3 RAM
Úložiště:	2GB eMMC nebo SD karta
Periferie:	1*USB, 1*mini-USB
	1*Ethernetový port 100Mbps

Tab. 1.1: Parametry Beaglebone Black



Obr. 1.1: Beaglebone Black

Hlavní předností Beaglebone Black je však jeho GPIO port, obsahuje dva označené jako 8 a 9 oba po 46 pinech. Na těchto pinech se nachází ve velké množství I/O pinu a použitelných sběrnic. Zde je výčet:

- 7 Analogových pinů

- 65 Digitálních pinu na 3.3 V
- 2x I2C sběrnice
- 2x SPI sběrnice
- 2x CAN sběrnice
- 4x UART
- 8x PWM
- A/D Převodník

[1]

1.1.1 Operační systém

Výrobce a komunita nabízí nepřeberné množství operačních systémů, které je možné na Beaglebone Black používat. Na výběr jsou jak linuxové distribuce, tak operační systémy zakládající se na systému android. Existuje i několik experimentálních distribucí, které mají však velmi specifickou funkcionalitu. Byla zvolena distribuce Debian 8 „jessie“. Důvody pro výběr této distribuce byly předchozí zkušenosti právě s distribucí Debian a potom především stabilita systému a kompromis mezi svižností systému a možnostmi, které právě Debian 8 nabízí. Důležitým aspektem je, že distribuce jako taková se těší velké oblibě a má tak velkou uživatelskou základnu. Díky tomu existuje velké množství tutoriálů a návodů na technických fórech především v anglickém jazyce.

Při prvotním spuštění Beaglebone Black je nutné jej nejdříve nastavit a nainstalovat požadovanou distribuci. Beaglebone Black má pro nahrání operačního systému dvě možná úložiště, ze kterých může systém bootovat. Je to buď vnitřní eMMC paměť osazená na desce s kapacitou 2GB nebo lze do Beaglebone připojit paměťovou kartu typu MicroSD. Výhodou eMMc paměti oproti paměťové kartě je přenosová rychlost, která je řádově několikrát vyšší, než je tomu u nejrychlejších mircoSD pamětí. Beaglebone Black ještě obsahuje jakési vlastní vývojové prostředí, na které se lze připojit přes USB. Stačí jednoduše připojit Beaglebone Black pomocí USB kabelu do PC. Potom je nutné si z oficiální stránky stáhnout USB driver `http://beagleboard.org/getting-started`. USB propojení s Beaglebone Black se potom tváří jako emulované ethernetové zařízení. Poté se lze na zařízení připojit pomocí IP adresy `http://192.168.7.2`. Na této adrese se nachází webové stránky výrobce, kde je množství tzn. Bonescriptů. Jedná se o ukázku základních funkcionalit. Nachází se zde i Cloud9 IDE vývojové prostředí, ve kterém lze již programovat a manipulovat s GPIO porty. [2]

Z důvodu malé kapacity eMMC paměti bylo tedy třeba použít microSD paměť. V případě použití distribuce Debianu 8.2 je nutná kapacita alespoň 4GB. Na pa-

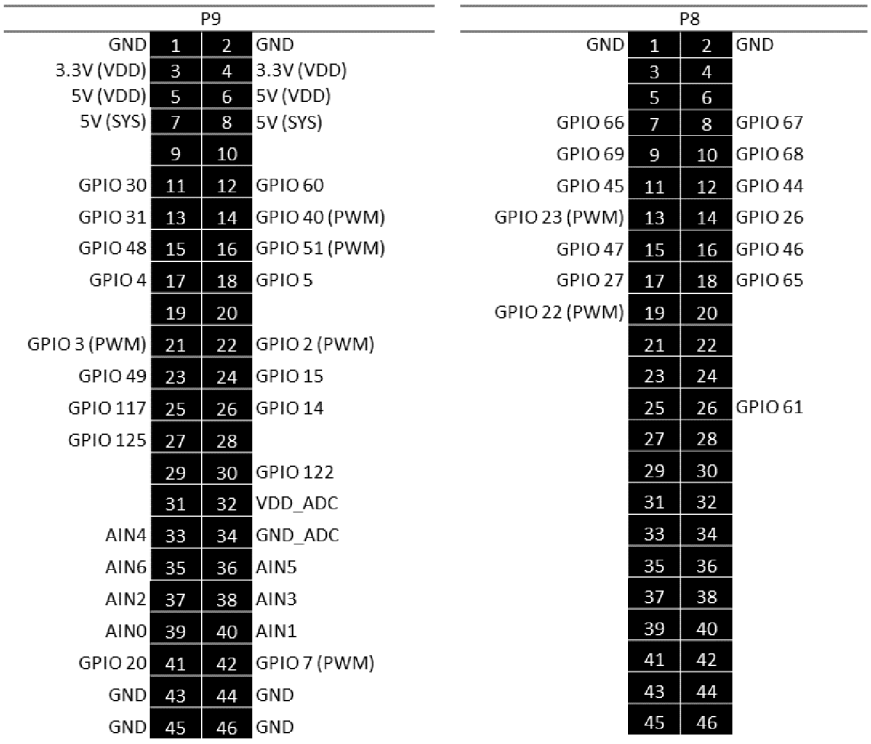
měťovou kartu je třeba nejdříve nakopírovat image distribuce, tak aby z karty bylo možné nabootovat. K tomu slouží například program Win32 Disk Imager. Po té už stačí do vypnutého Beaglebone Black vložit paměťovou kartu a při připojování napájení držet tlačítko USER/BOOT. Potom již následuje standardní procedura dokončení instalace a prvotního nastavení distribuce Debian, nastavení uživatelského účtu, síťového rozhraní atd. Všechny distribuce Debian mají již defaultně povolen a nastaven SSH server pro připojení, stačí tedy využít SSH klienta - například program „PuTTY“. Všechna další nastavení budou prováděna již v terminálu přes program „PuTTY“.

1.1.2 Programování

Programovat Beaglebone Black lze v několika různých programovacích jazycích. Nejčastěji jsou využívány Python, C nebo C++. Z důvodu dřívější zkušenosti a lepší znalosti byl vybrán skriptovací jazyk Python. Konkrétně Python ve verzi 2. 7. Důvodem, proč byla vybrána starší verze jazyka oproti aktuálně nejnovější Python 3.4, je ten, že většina podpůrných ovladačů pro různé periferie, např. smbus, je napsaná právě ve verzi 2. 7. Existují sice ovladače v novější verzi, avšak nejsou plně funkční. Další motivací pro využití Pythonu oproti C++ je velká komunita a množství dostupných návodů.

1.1.3 GPIO konektor

GPIO porty obr.:1.2 jsou jednou z největších předností Beaglebone Black. Skládájí se ze dvou konektorů každý po 46 pinech. Z těch 92 pinů je 65 ovladatelných digitálních vstupů nebo výstupů. Dále jsou zde vyvedeny všechny napěťové úrovně, které se nacházejí na desce. Konkrétně 3,3 V a 5 V, tyto piny lze využít k napájení některých malých obvodů, avšak jejich maximální odběr může být do 300 mA. Digitální piny mají 3,3 V logiku a jsou schopny dodat nebo sepnout proud do země v rozmezí 4-5 mA, na což je třeba brát ohled, protože je velmi snadné je zničit. Teoretické použití 65 digitálních pinů je však reálně možné pouze při přepnutí tzn. multiplexních módů (device tree). Těch má Beaglebone Black celkem 8, respektive 0 – 7. Tyto módy zpřístupňují například komunikační sběrnice. Protože Beaglebone Black má tak velké množství komunikačních sběrnic a funkcionalit, že by nebylo fyzicky možné všechny tyto sběrnice vyvést na vlastní piny, funkce většiny pinů se překrývají a v každém módu je funkce pinů jinak definována. Pro lepší orientaci je zde program Pin Mux Utility od firmy Texas Instruments, kde lze například i zadat požadované rozhraní a program zobrazí, ve kterém módu mají piny pro onu funkci být, a zda nedochází ke kolizím. [3] Program zároveň slouží pro generování



P9			
GND	1	2	GND
3.3V (VDD)	3	4	3.3V (VDD)
5V (VDD)	5	6	5V (VDD)
5V (SYS)	7	8	5V (SYS)
	9	10	
GPIO 30	11	12	GPIO 60
GPIO 31	13	14	GPIO 40 (PWM)
GPIO 48	15	16	GPIO 51 (PWM)
GPIO 4	17	18	GPIO 5
	19	20	
GPIO 3 (PWM)	21	22	GPIO 2 (PWM)
GPIO 49	23	24	GPIO 15
GPIO 117	25	26	GPIO 14
GPIO 125	27	28	
	29	30	GPIO 122
	31	32	VDD_ADC
AIN4	33	34	GND_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO 20	41	42	GPIO 7 (PWM)
GND	43	44	GND
GND	45	46	GND

P8			
GND	1	2	GND
	3	4	
	5	6	
GPIO 66	7	8	GPIO 67
GPIO 69	9	10	GPIO 68
GPIO 45	11	12	GPIO 44
GPIO 23 (PWM)	13	14	GPIO 26
GPIO 47	15	16	GPIO 46
GPIO 27	17	18	GPIO 65
GPIO 22 (PWM)	19	20	
	21	22	
	23	24	
	25	26	GPIO 61
	27	28	
	29	30	
	31	32	
	33	34	
	35	36	
	37	38	
	39	40	
	41	42	
	43	44	
	45	46	

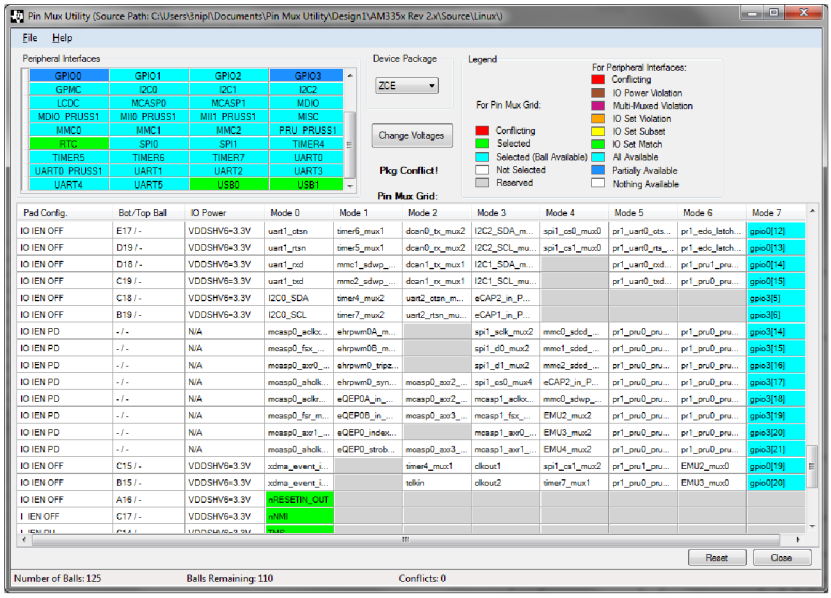
Obr. 1.2: GPIO konektor

souborů pinmux.h, který slouží právě pro nastavení pinů do požadovaného módu. Pro následující aplikaci je nutná dostupnost: I2C1, I2C2, Uart2, Uart4, Can1. Tyto sběrnice byly zvoleny tak, aby nedocházelo ke zmíněnému konfliktu. Ve výchozím stavu však tyto piny mají jinou funkci, protože je nutné nahrát soubory Device Tree overaly. To lze provést stažením souboru z gitu, nebo je na kopírovat z přílohy (složka: „overlay“, do adresáře /lib/firmware). [5] Pro úspěšnou instalaci je třeba zadat následující sekvenci příkazů s právy roota:

Listing 1.1: Změna deviceTree.

```
cd /lib/firmware
wget https://raw.githubusercontent.com/
 lgxlogic/CBB-Serial/master/overlay/cape-CBB-Serial-r01.dtbo

wget https://raw.githubusercontent.com/
 lgxlogic/CBB-Serial/master/overlay/BB-UART2-RTSCTS-00A0.dtbo

wget https://raw.githubusercontent.com/
 lgxlogic/CBB-Serial/master/overlay/BB-UART4-RTSCTS-00A0.dtbo
```

```
echo cape-CBB-Serial:r01 > /sys/devices/bone_capemgr.*/slots
```



Obr. 1.3: Ukázka programu Pin Mux Utility

Program zároveň slouží pro generování souboru mux.h, který slouží právě pro nastavení pinů do požadovaného módu. Pro následující aplikaci je nutná dostupnost: I2C0, I2C1, Uart2, Uart4, SPI1 a CAN1. Pro samostatné programování GPIO pinů a dalších periferií jsou k dispozici volně šiřitelné knihovny od adafruit.com „Adafruit-BeagleBone-IO-Python“, jejich instalace je nutná pro pozdější programování. Zde je sled příkazů, které je nutné zadat do terminálu:

Listing 1.2: Instalace knihoven od adafruit.com.

```
sudo ntpdate pool.ntp.org
sudo apt-get update
sudo apt-get install build-essential python-dev python-pip python-smbus -y
git clone git://github.com/adafruit/adafruit-beaglebone-io-python.git
cd adafruit-beaglebone-io-python
sudo python setup.py install
cd ..
sudo rm -rf adafruit-beaglebone-io-python
```

Po dokončení instalace je nutné otestovat, zda vše proběhlo korektně, zavoláním následujícího příkazu:

Listing 1.3: Test instalace

```
sudo python -c "import Adafruit_BBIO.GPIO as GPIO; print GPIO"
                    #odpoved by mela byt nasledujici:
<module 'Adafruit_BBIO.GPIO' from
'/usr/local/lib/python2.7/dist-packages/Adafruit_BBIO/GPIO.so'>
```

Pokud se objeví chybová hláška, je nutné instalační sekvenci opakovat. [4]

1.1.4 Sběrnice I2C (Inter-Integrated Circuit)

Součástí GPIO portu je sběrnice I2C. Tato sběrnice propojuje jedno zařízení Master a více zařízení typu Slave. Propojení je realizované pomocí dvou vodičů SDA - datový vodič a SCL - hodinový signál. Jedná se tedy o poloduplexní komunikaci, kdy je v daném okamžiku vysíláno pouze jedním směrem. Komunikace probíhá tak, že Master může vysílat vždy, Slave pouze na vyžádání Mastera. Rozlišení více zařízení typu Slave je zajištěno pomocí adresy, tedy každé takové zařízení má svoji vlastní adresu. Počet zařízení typu Slave je limitován počtem adres a kapacitancí sběrnice. Po připojení nějakého integrovaného obvodu pomocí I2C si lze na BeagleBone Black ověřit jeho funkčnost a správné zapojení následujícím příkazem:

Listing 1.4: Test I2C Sběrnice

```
i2cdetect -y -r 1
```

Po zadání se objeví seznam všech dostupných adres.

1.1.5 Sběrnice SPI (Serial Peripheral Interface)

Je to externí sériová sběrnice sloužící k propojení dvou nebo více komunikujících uzlů. Podobně, jako je tomu u I2C, se zde dělí zařízení na Master a Slave. Hodinový signál (vodič SCK) je distribuován do všech uzlů sběrnice. Pro posílání dat jsou zde dva vodiče MISO a MOSI. Dalším vodičem této sběrnice je CS, který slouží k výběru aktivního uzlu sběrnice. Sběrnice používá full-duplexní přenos dat. Nevýhodou SPI je, že data je pomocí této sběrnice možné přenášet pouze na kratší vzdálenost, což je způsobeno nutností synchronizace hodinového signálu. Další nevýhodou je, že přenos neobsahuje žádné potvrzení doručení. Tedy v případě, kdy některé zařízení nedokáže dostatečně rychle zpracovávat přijímaná data, dojde k nedoručení a tím i k chybě. SPI sběrnice je však velmi široce využívána.

1.1.6 Sběrnice RS-485

Sběrnice RS485 je další z komunikačních sběrnic, kterou lze využít ke komunikaci s testovaným zařízením. Sběrnice je využívána především v průmyslovém prostředí, má stejný základ jako RS232, liší se však jinými logickými napěťovými úrovněmi. Jedná se o poloduplexní dvouvodičové sériové spojení. Délka sběrnice může být až 1200m a maximální počet uzlů je 32. Logické úrovně jsou reprezentovány diferenčním napětím mezi dvěma vodiči, typicky označované jako A a B. Logické úrovně jsou detekovány jak A – B < -200 mV pro „log 1“ a A – B > +200m V pro „log 0“. Při komunikaci na větší vzdálenost je však doporučováno propojit I vodič GND. Přenos dat probíhá pomocí 7 nebo 8 bitových rámců. Jelikož se jedná o half-duplex, je začátek a konec rámce označen 1 start-bitem a ukončen 1 nebo 2 stopbity. Hlavní úskalí této sběrnice je, že je použita její dvouvodičová varianta, pro obousměrnou komunikaci jsou využívány pouze dva vodiče. Tento problém lze vyřešit implementací komunikačního protokolu, který ale není součástí standardu RS-485. Jedná se o typický problém jednoho vícebodového spoje, tedy přístup ke sdílenému médiu. Je tedy nutné upravit komunikační individuálně pro testovaný hardware. [14]

1.1.7 Sběrnice RS-232

Jedná se o sériovou sběrnici. Dříve byla velmi často používaná, dnes je spíš na ústupu. Nejčastěji využívaným konektorem je D-SUB. Standard definuje asynchronní sériovou komunikaci pro přenos dat. Počet datových bitů je volitelný, obvykle se používá 8 bitů, lze se také setkat se 7 nebo 9 bity. Logický stav „log 0“/„log 1“ přenášených dat je reprezentován pomocí dvou možných úrovní napětí, které jsou bipolární a dle zařízení mohou nabývat hodnot ±5 V, ±10 V, ±12 V nebo ±15 V. Sběrnice je definována devíti vodiči, avšak využití jednotlivých vodičů závisí na aplikaci a ve většině případů nejsou ke komunikaci využity. Jednotlivé vodiče popisuje tab.:1.2. Jelikož v dnešní době již většina zařízení nevyužívá některé z popsaných vodičů, nebo jen velmi vzácně (typicky modemy a staré telefony), není třeba s nimi počítat v implementaci v teto aplikaci. [13]

Zkratka pinu	Název	Popis
TX, TD	Transmit Data	Data posílaná z DTE do DCE
RX, RD	Recieve Data	Data přijímaná v DTE z DCE
RTS	Request to Send	Požadavek na vysílání
CTS	Clear to Send	Povolení k vysílání
DSR	Data Set Ready	Signalizační bit
GND	Ground	Signálová zem
DCD	(Data) Carrier Detect	Signalizační bit
DTR	Data Terminal Ready	
RI	Ring Indicator	

Tab. 1.2: Význam vodíčů RS-232

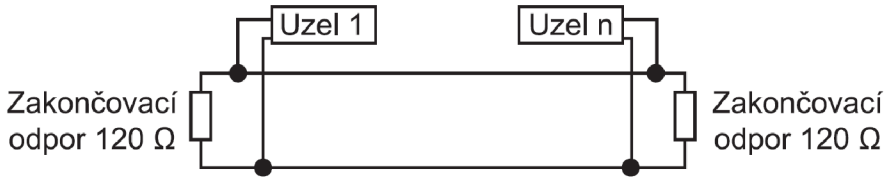
1.1.8 Sběrnice CAN (Controller Area Network)

Jedná se o sériovou datovou sběrnici, vyvinutou firmou Robert Bosch GmbH. Nejčastěji je využívána pro vnitřní komunikační síť senzorů a u funkčních jednotek automobilů. Používá se i pro automobilovou diagnostiku. Can sběrnice se je definována normou ISO 11898. Norma popisuje chování sběrnice CAN na fyzické, síťové a transportní vrstvě. Umožňuje distribuované řízení systémů v reálním čase s vysokou mírou zabezpečení proti chybám. Protokol je typu multi-master, každý uzel může být master a řídit tak chování jiných uzlů. Jelikož sběrnice funguje jako jeden vícebodový spoj, obsahuje protokol metodu řízení přístupu k médiu s náhodným přístupem. Ten řeší kolize na základě prioritního rozhodování. Zprávy vysílané po sběrnici neobsahují žádnou informaci o cílovém uzlu, kterému jsou určeny, přijímají je tedy všechny připojené uzly. Každá zpráva obsahuje identifikátor, jenže definuje cílový uzel a prioritu. Priorita je využita ve chvíli, kdy dojde ke kolizi. Zpráva s vyšší prioritou je doručena na úkor zprávy s nižší prioritou.

Fyzická vrstva

CAN definuje vlastní rozhraní rozhraní k fyzickému přenosovému médiu. Standard protokolu CAN definuje dvě vzájemně komplementární hodnoty bitů na sběrnici - dominant a recessive. To znamená, že logické úrovně signálu nejsou pevně dány a záleží na konkrétní realizaci fyzické vrstvy. Konkrétně byla použita diferenciální sběrnice dle normy, která definuje velikost rozdílového napětí mezi úrovní dominant Vdiff = 2 V a recessive Vdiff = 0 V ve schématu označené jako CAN H a CAN L. Ke sběrnici se na každém konci uzlu typicky připojuje zakončovací odpor o hodnotě 120 Ω, který složí jako přizpůsobení vedení. Nejčastěji se používají konektory D-SUB, v aplikaci jsou pak napájecí vodiče společně s datovými vyvedeny na svor-

kovnice viz. obr.:1.4.[10]



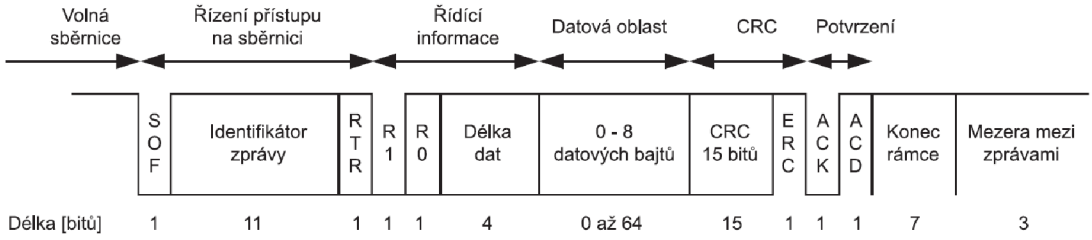
Obr. 1.4: Přizpůsobení vedení

Linková vrstva

Specifikace protokolu CAN definuje čtyři typy zpráv:

- datová zpráva
- žádost o data
- zpráva o chybě
- zpráva o přetížení

Datová zpráva a žádost o data slouží k samotnému přenosu dat. Pole obsahující data můžou být dlouhá až 8 B. Pokud je třeba vykonat nějaký příkaz, jako např.: ON nebo OFF, může se datové pole úplně vynechat. Jednoduché příkazy lze posílat jen pomocí identifikátoru zpráv. Protokol využívá dva typy možných rámců Standard Frame a Extended Frame. Rozdíl mezi nimi je v délce pole identifikátoru zprávy. U Extended je to 29 bitů a u Standard 11. Na obrázku obr.:1.5 je vyobrazen datový rámec CAN sběrnice.[10]



Obr. 1.5: CAN rámec

Název Pole	Délka	Význam
SOF	1b	Sekvence bitů značící začátek zprávy
Identifikátor zprávy	11b (29b)	Řízení přístupu ke sběrnici
RTR	1b	Označení datové zprávy nebo žádost o přístup
R1, R2	1b	Rezervováno pro krátké příkazy
Délka dat	4b	Délka datové zprávy
Datová oblast	0-8B	Pole pro přenášená data
CRC	15b	Kontrolní součet
ERC	1b	CRC oddělovač
ACK	1b	Potvrzení
ADC	1b	Oddělovač
Konec rámce	7b	Sekvence bitů značící konec zprávy
Mezera mezi zprávami	3b	Mezera mezi rámci

Tab. 1.3: Význam polí CAN rámce

Jednotlivé významy a obsahy polí se mění v závislosti na typu rámece, jenž může mít hodnotu Recessive nebo dominant. Informace o zapojení a ukázka komunikace se nachází v kapitole: 3.6

1.2 Připojení přístrojů

1.2.1 GPIB (General Purpose Interface Bus)

Jedná se o rozhraní pro měřící a zkušební přístroje vyvinuté společností HP v roce 1972. Je jedním z nejrozšířenějším komunikačním rozhraním používané v měřící technice. Jedná se o paralelní sběrnici, která má 24 vodičů. Přenos probíhá asynchronně. Maximální možná vzdálenost mezi dvěma zařízeními jsou 2 m. I když lze rozhraní stále najít u nejnovějších přístrojů, je jejich použití nepraktické, vzhledem k počtu použitých vodičů a velikosti konektoru. Maximální vzdálenost 2 m mezi dvěma prvky je také velmi limitující. Z toho důvodu bylo upuštěno z dřívějšího návrhu připojovat se k měřícím přístrojům pomocí GPIB. Jako alternativa bylo vybráno rozhraní USB. USB rozhraním disponuje drtivá většina všech vzdáleně ovladatelných měřících přístrojů a jeho použití je pro uživatele pohodlnější. Mimo to ze standardu USB 2.0 vyplývá, že maximální vzdálenost může být až 5 m.

1.2.2 VISA (Virtual Instrument Software Architecture)

Jedná se o standard popisující specifikaci pro komunikaci s měřícími přístroji. V podstatě je to předpis, jak komunikovat s měřícími přístroji, a to nezávisle na tom, jakým rozhraním jsou k počítači připojeny. VISA podporuje většinu dostupných interface, jako je GPIB, RS232, USB či TCP/IP. VISA je kromě standardu množství knihoven, pro různé druhy operačních systémů. Tyto knihovny obsahují specifikace pro komunikaci s přístroji. Knihovny VISA byly postupně vyvíjeny a přetvářeny pro potřeby specifických programovacích jazyků a doplňovány výrobci.

1.2.3 PYVISA a NIVISA

PYVISA a NIVISA představuje ve spojení velmi silný nástroj jak ovládat měřící přístroje. Jak bylo již zmíněno k programování systému„ pro psaní samotných scénářů testu byl zvolen jazyk Python. Jazyk Python nemůže přímo komunikovat s VISOU. K tomuto účelu slouží PYVISA, což je wrapper pro sdílené knihovny NIVISA. PYVISA pravidelně vychází v nových verzích a je doporučeno vždy používat tu nejaktuálnější.

Jelikož výrobci měřících přístrojů většinou cílí na uživatele OS Windows, je logické, že většina podpůrných nástrojů je vyvíjna právě pro tento operační systém. NIVISA proto není výjimkou. Avšak společnost po nátlaku vývojářů nakonec vydala i instalaci NIVISA pro Linux. Bohužel, oficiálně podporované distrubuce jsou jen RedHat, Scientific Linux a openSUSE. Nicméně lze provést instalaci i na ostatních distribucích, Nacianl instruments však nezaručuje jejich plnou funkcionalitu. Instalaci lze stáhnout z `http://ftp.ni.com/support/softlib/visa/NI-VISA/15.0/Linux/NI-VISA-15.0.0.iso` . Konkrétně ve verzi 15. Jelikož se jedná o ISO, je třeba jej nejdřív připojit a obsah extrahovat do připravené složky. Pak nainstalovat „alien“ pro instalaci rpm balíčků. To se provede následovně, příkazy je nutné zadávat s právy roota:

Listing 1.5: Instalace NI-VISA

```
sudo mkdir /mnt/iso/"

mount -o loop <cesta k souboru.iso> /mnt/iso"
cp -R /mnt/iso/* ~/NI/
sudo apt-get install alien
cd ~/NI/ and execute
sudo ./INSTALL --nodeps
```

Po sekvenci příkazů by se měl v instalační složce objevit soubor „visaconf“. Pro instalaci PYVISA a NIVISA bylo nutné nejdříve nainstalovat instalační balíček pip, neboť jej debian defaultně neobsahuje. To se provede následujícím příkazem:

Listing 1.6: Instalace balíčkového systému pip

```
sudo apt-get install python-pip
```

Po provedení instalace lze přejít k samotné instalaci PYVISA, což se provede příkazem:

Listing 1.7: Instalace PYVISA

```
sudo pip install -U pyvisa
```

Po úspěšné instalaci bylo třeba vyzkoušet, zda je instalace funkční. K tomu stačí vytvořit malý skript v jazyku Python:

Listing 1.8: Test funkční instalace PYVISA

```
import visa
rm = visa.ResourceManager()
rm.list_resources()
print(rm)
my_instrument = rm.open_resource('GPIB0::14::INSTR')
print(my_instrument.query('*IDN?'))
```

Tento skript vypíše dvě různé informace. Tou první je důkaz, zda byla instalace úspěšná a zda má PYVISA správně nastavenou cestu ke knihovnám NIVISA. Druhý výpis má za následek to, že pokud jsou k zařízení připojeny již nějaké měřící přístroje a vše pracuje správně, tak je na ně odeslán SCPI příkaz *IDN?. Ten má za následek, že měřící přístroje ohlásí v předepsaném formátu (příklad pro USB): `USB[board]::manufacturer ID::model code::serial number[::USB interface number][::INSTR]`

Po instalaci bylo možné přejít k samotnému programování. S měřícími přístroji je komunikace v zásadě snadná, probíhá pomocí SCPI příkazů, které jsou popsány v kapitole o SCPI příkazech. Jediným stěžejním je vyčítání a zapisování hodnot (buď změřených, nebo pro nastavení). Ty lze vyčítat jako binární číslo nebo jako ASCII hodnotu. Zde je příklad pro vyčtení naměřeného proudu z multimetru, nejdříve jako ASCII a poté jako binární číslo:

Listing 1.9: Čtení z multimetru

```
values.append(inst.query_ascii_values('CURR?'))

values.append(inst.query_binary_values
('CURR?', datatype='d',is_big_endian=True))
```

V obou případech jsou hodnoty přidány do seznamu, kde s nimi lze dále pracovat. [7]

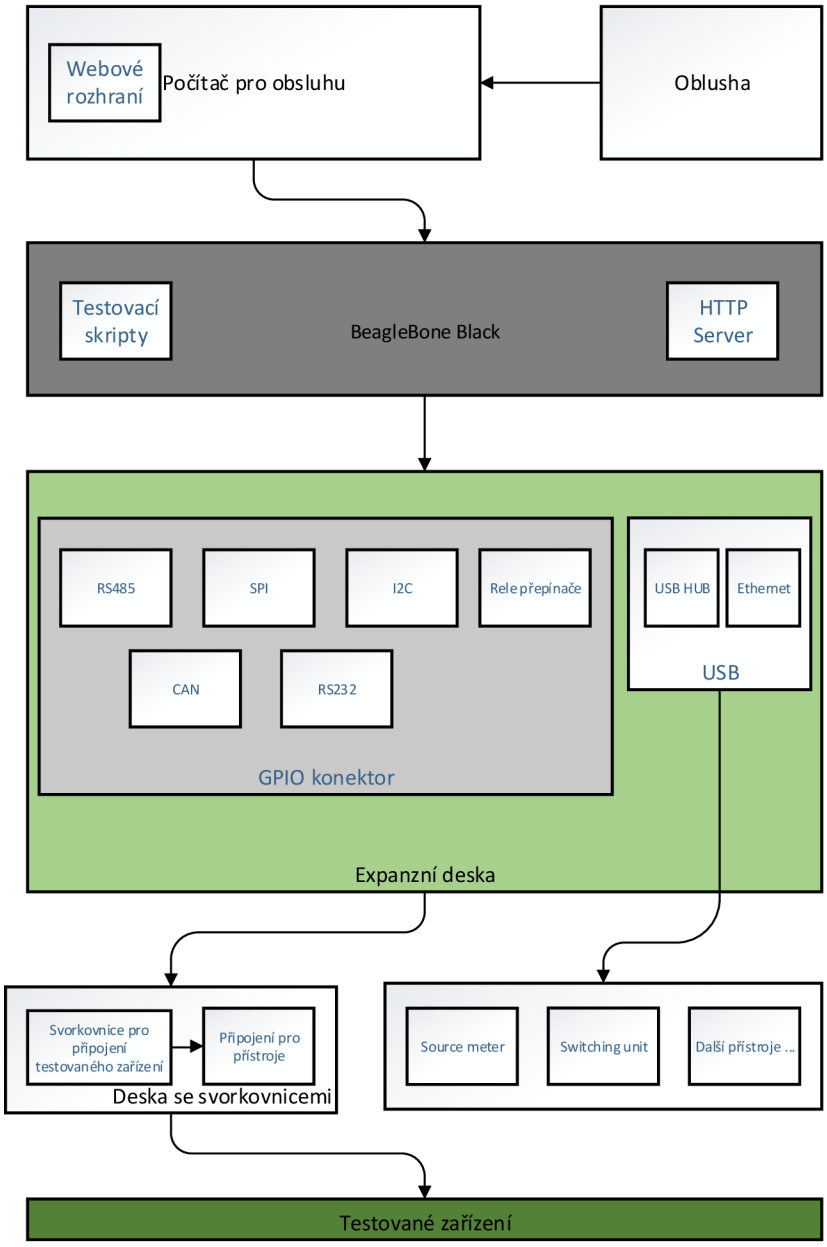
1.2.4 SCPI

(Standard Commands for Programmable Instruments) Standard SCPI je souhrn příkazů a pravidel pro komunikaci mezi řídící jednotkou, zde Beagle Bone Black a přístrojem v automatizovaném měřícím systému. Je nezávislý na technickém řešení a protokolu přenosu dat. Standard je pevně definován specifikací IEEE 488.2 a to včetně syntaxe. SCPI příkazy jsou pouze řetězce znaků oddělované určitými znaky. Příkazy jsou hierarchicky uspořádány do logických skupin. Každá skupina má potom vlastní kořenovou strukturu. Předpis všech dostupných SCPI příkazů je vždy dodáván výrobcem a typicky bývá součástí manuálu. Bohužel, i když jsou SCPI příkazy standardizovány, nebývá syntaxe napříč všemi výrobci u zařízení stejného typu kompatibilní. V syntaxi se vždy nachází drobné rozdíly. Některé firmy, jako například Keithley, začínají prosazovat vlastní standard ovládacích příkazů. U nejnovějších přístrojů od firmy Keithley se již nepočítá se standardní podporou SCPI příkazů a nahradili je vlastním symbolickým jazykem TSP. Výhodou TSP jazyka oproti SCPI je to, že je více přizpůsoben potřebám moderních programovacích jazyků. Počítá se například s dynamickým polem, díky čemuž odpadá repetitivní zadávání totožných příkazů. Nevýhodou je však prozatím jeho omezená podpora od ostatních výrobců. Bylo tedy nutné vystačit si s SCPI.

Zadávání příkazu z knihovny PYVISA v kombinaci s velkým počtem SCPI příkazu bylo neslučitelné s myšlenkou snadné tvorby testovacího plánu. Bylo proto vytvořeno množství tříd s funkcemi pro usnadnění obsluhy externích přístrojů. V třídách se však nachází nejvíce používané příkazy. Většina měřících přístrojů obsahuje velké množství funkcionalit, které pro běžné testování nejsou potřebné. Pokud to předpis vyžaduje, je možné snadno požadovanou funkci doplnit.

2 BLOKOVÉ SCHÉMA

Na obr: 2.1 lze vidět blokové schéma celého navrženého systému. Zařízení lze rozdělit na tři funkční bloky. První blok je samotný Beaglebone Black, jakožto řídící prvek celého systému. Na Beaglebone Black se nachází testy jako jednotlivé pythonové skripty a webové stránky, které zajišťují ovládací interface. Samotné webové stránky běží na Apache2 z XAMP, což je balíček jenž obsahuje vše potřebné pro tuto aplikaci (Apache2,MySQL, PHP, atd.). Druhým funkčním blokem je expanzní deska, do které se Beaglebone Black zasune pomocí GPIO konektorů P8, P9 a vertikálního USB konektoru. Na expanzní desce se nachází GPIO expandery pro ovládání přepínací logiky. Dále jsou zde z Beaglebone Black vyvedeny komunikační sběrnice I2C A UART a CAN. Z UART je pak dále pomocí driveru vytvořená sběrnice RS485 a RS232. Všechny signály a výstupy relé přepínačů, které jsou určeny pro testování nebo komunikaci s testovaným zařízením, jsou vyvedeny na připojitelné DPS se svorkovnicemi. Součástí expanzní desky je USB HUB a další ethernetový port. Ty jsou připojeny pomocí rozhraní USB. USB HUB slouží k připojení externích měřících přístrojů. Další ethernetový port pak slouží jako možnost otestování např. síťové karty testovaného zařízení, nebo rovněž pro komunikaci s měřícími přístroji, které disponují ethernetovým portem a výrobce nenabízí ovladač pro USB. Pro ovládání celého zařízení je pak nutný další počítač, kterým lze buď spouště a vyhodnocovat testy přes webové rozhraní a nebo se připojit přes SSH a spravovat samotné testy. Samotný testovaný prvek je propojen dle předem navržené logiky do svorkovnic, respektive všechny jeho vstupy výstupy a testované části. Dále v závislosti na charakteru testovaného prvku je případně možná komunikace s ním pomocí dostupných sběrnic.



Obr. 2.1: Blokové schéma

3 EXPANZNÍ DESKA

3.1 USB a Ethernet

Jelikož bylo v návrhu upuštěno od rozhraní GPIB, bylo nutné zajistit další USB porty pro připojení externích měřících zařízení, jelikož Beagle Bone Black disponuje pouze dvěma. Z toho jeden je typu mikroUSB. Dalším požadavkem na expanzní desku byl druhý ethernetový port pro testovací účely. Proto bylo nutné nějakým způsobem tyto požadavky implementovat do expanzní desky. K tomuto účelu byl vybrán integrovaný obvod LAN9512, který v sobě zahrnuje jeden ethernetový port a USB HUB s možností vyvedení dvou konektorů. USB HUB podporuje zařízení typu USB 2.0. Ethernetový port může pracovat s rychlostmi 10/100Mbit. Integrovaný obvod má již integrovanou vlastní MAC adresu. Celý obvod je napájen 3,3 V. Obvod je dodávám s ovladači pro všechny běžné operační systémy včetně Linuxu. Ovladač lze stáhnout ze stránek výrobce `http://www.microchip.com/SWLibraryWeb/product.aspx?product=SRC-LAN95xx-LINUX`, rovněž lze ovladač najít v příloze. Použitá distrubuce Debianu přímo neobsahuje tento ovladač, jak tomu je například u distribucí pro Raspberry PI. Je třeba ovladač nainstalovat. Balík Lan9500_linux_1.01.33.tar.gz obsahuje jak ovladač pro Ethernet tak pro dvojci USB. Instalace byla provedena zadáním sekvence příkazů dle manuálu s právy roota: [6]

Listing 3.1: Instalace ovladače pro LAN9512

```
tar -xzvf Lan9500_linux_1.01.33.tar.gz

cd smsc9500/

./configure
make
make install

cd smsc9500/
insmod smscusbnet.ko
insmod smsc9500.ko
```

Tím je samotná instalace hotová. Nakonec bylo třeba softwarově přidat eth rozhraní. V manuálu od výrobce je tento postup uveden pomocí wizardu v grafickém prostředí. Jelikož Beaglebone Black nedisponuje žádným konektorem umožňujícím připojení monitoru, je třeba využít například program VNC. Následně v grafic-

kém prostředí spustí wizard „Add new Device Type“ -> „Ethernet connection“ -> „Smsc9500“ a provést požadované nastavení.

Schéma zapojení bylo z valné části přebráno z datasheetového zapojení výrobce, bylo modifikováno pouze napájení dvou USB typu Host, kde byl změněn typ proudové ochrany. U ethernetu byl zvolen jiný konektor.

Schéma zapojení je rozděleno do několika logických bloků (viz schéma). První částí je filtrace napájení. Filtrace je zde důležitá především kvůli citlivosti integrovaného obvodu na zvlnění napájecího napětí. Dalším blokem je USB upstream, což vyvedení USB konektoru pro připojení celého obvodu k Beagle Bone Black. Následuje proudová ochrana pro dvojici USB konektorů typu Host. Ta je tvořena dvojicí obvodů LM3525. Dalším prvkem je volitelná EEPROM paměť. Paměť slouží např. pro změnu defaultní MAC adresy a pro standardní nastavení při spuštění.

USB 2.0 a ethernet jsou nejsložitější částí expanzní desky, především v pozdějším návrhu layoutu, kde je nutné brát zřetel na řadu specifik, která s sebou nesou. Jsou to například maximální možné vzdálenosti signálových cest nebo jejich vzájemná vzdálenost. Soupis všech specifik lze nalézt na `http://www.ti.com/lit/an/spraar7e/spraar7e.pdf` a `http://www.ti.com/lit/an/spma036b/spma036b.pdf`. Sám výrobce dává k dispozici layout Guidelines `http://www.microchip.com/wwwAppNotes/AppNotes.aspx?appnote=en562751`. Zřetel je třeba brát zejména při vedení diferenčních párů RX/TX u ethernetu a stejně tak u USB.

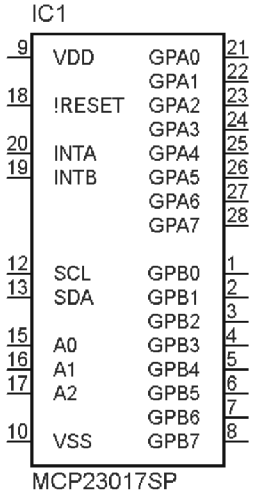
3.2 I2C expadér a přepínací logika

3.2.1 MCP 23017

Je to 16 bitový expandér, který se připojuje pomocí komunikační sběrnice I2C. Celý obvod lze napájet od 1,8 V do 5,5 V. Při napájecím napětí 5 V dokáže každý pin dodat maximální proud o hodnotě 24 mA. Další výhodou je, že expandér již obsahuje interní pull-up rezistory na všech GP pinech, které lze pomocí příslušného příkazu odpojit nebo připojit. Ty jsou vhodné při vyčítání stavu z jednotlivých pinů, kdy, stejně jako u GPIO pinů, může nastat hazardní stav.

Rozšiřující obvod je 16 bitový, tyto bity jsou rozděleny do dvou registrů GPA a GPB. Komunikace přes 2C probíhá přes piny 12 a 13. Každý takto připojený expandér rozšiřuje celkový počet GPIO pinů o 16. Každý takový obvod je 3 bitově adresovatelný pomocí pinů A0, A1 a A2. Celkově tedy lze připojit až osm takových expandérů, čímž lze získat až 128 I/O vstupů, přičemž lze každý zvlášť ovládat.

Důležitým prvkem při komunikaci s expandérem je adresování. U každého obvodu je nutné nastavit jeho adresu, pomocí které pak bude komunikovat přes sběrnici. Adresa se stanovuje zapojením. K tomuto účelu slouží piny A0, A1, A2. Tyto



Obr. 3.1: Obvod MCP23017

piny se připojí buď na napájení nebo na zem, podle toho, jakou adresu chceme zvolit. Pokud všechny piny připojíme na zem, bude adresa 0x20. Číslo za (x) vždy začíná číslem 20, k tomuto číslu se pak přičítá námi nastavené číslo v binární hodnotě. Tedy pokud je například připojen pin A2 na VCC, bude tím reprezentovat číslo 4. Když toto číslo přičteme k číslu 20, získáme výslednou adresu 0x24. Pomocí skriptu lze opět nastavit každý port jako vstup nebo výstup. Jelikož má expandér rozdělené jednotlivé piny do dvou registrů, lze ovládat souhrnně i jednotlivé registry. Nastavování jednotlivých registrů pro vstup nebo výstup se provádí pomocí následujících příkazů a adres v hexadecimálním tvaru. [15]

Příklad nastavení všech pinů v registru GPB jako výstupních při zavolání z terminálu:

Listing 3.2: Ovládání GPIO

```
sudo i2cset -y 1 0x20 0x01 0x00 # nastaveni GPIO registru jako Vystup
```

Adresa v Hexadecimalnim tvaru	Funkce
0x20	adresa obvodu
0x01	adresa registru GPB
0x00	nastaví všechny piny jako výstupní

Tab. 3.1: Adresování MCP23017 výstup

Při programování je však tento přístup značně nepraktický, proto byla modifikována třída z Adafruit pro přístup na I2C sběrnici právě pro obvody typu MCP230XX.

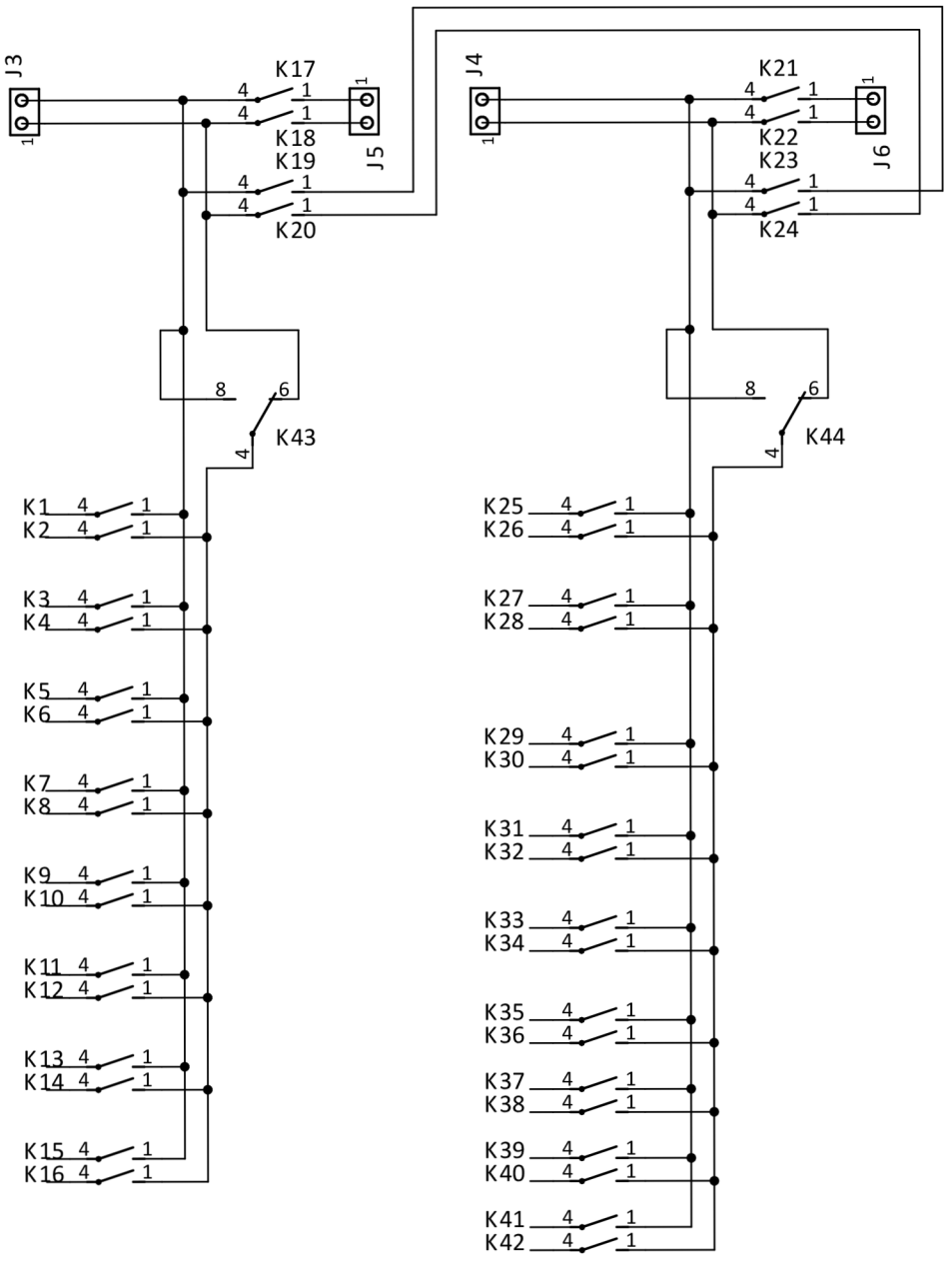
Následující příklad demonstruje ovládání pomocí předepsaných funkcí.

Listing 3.3: Ovládání GPIO pomocí upravené knihovny

```
#!/usr/bin/python
# −∗− encoding: utf−8 −∗−
from Adafruit_I2C import Adafruit_I2C
import time
import smbus

class MCP230XX(object):

    mcp = Adafruit_MCP23017(address = 0x20, num_gpios = 16)



    # postupne nastaveni vstupnich pinu
    mcp.config(0, OUTPUT)
    mcp.config(1, OUTPUT)
    mcp.config(2, OUTPUT)

    # Povoleni pull−up rezistoru na pinu 3
    mcp.pullup(3, 1)
    # Vypis aktualniho stavu na pinu 3
    print "%d: %x" % (3, mcp.input(3) >> 3)

   # Zapnuti a vypnuti pinu 0
      mcp.output(0, 1) # Pin 0 High
      mcp.output(0, 0) # Pin 1 Low
```

3.2.2 Propojování signálů

Nedílnou součástí testovacího zařízení je jeho přepínací pole. Testovací pole, které je vyobrazeno na obr: 3.2, je prostředkem k připojení měřícího přístroje k testovanému prvku. Základem přepínací logiky je, jak už bylo zmíněno, expandér MCP23017. Ten je propojen s obvodem ULN2003, což je pole darlingtonových tranzistorů, které zajišťují samotné spínání a rozpínání součástek. Samotný obvod ULN2003 není v tomto zapojení bezpodmínečně nutný, neboť obvod MCP23017 dokáže dodat dostatečný proud pro sepnutí relé SIL05-1A72-71LHR, avšak obvod obsahuje i ochrannou diodu proti proudové špičce při přepínaní relé a má sám velmi malé rozměry. Relé pro přepínání signálu pak bylo vybráno zmíněné SIL05-1A72-71LHR. Relé je typu



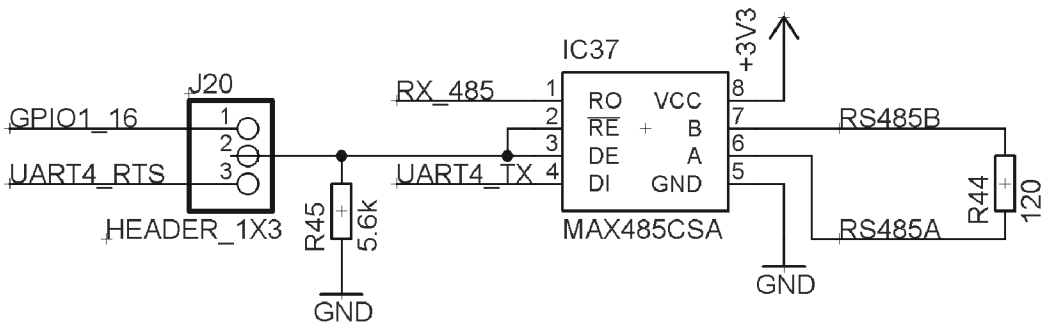
Obr. 3.2: Propojovací pole

SPST-NO, což znamená, že relé je spínací a v defaultním stavu je otevřené. Relé má DC cívku buzenou 5 V. Odpor cívky je 500 Ω. Z ohmová zákona tedy plyne, že k sepnutí relé stačí proud 10 mA. Relé je obzvláště vhodné pro měřící účely díky nízkému odporu při sepnutých kontaktech, který činí 150 Ω. Další výhodou je nízká doba rozepnutí 0,1 ms a doba sepnutí 0,7 ms. Jediným limitujícím faktorem je maximální protékající proud při sepnutí. Spínat a rozpínat pod zátěží lze pouze

500 mA a maximální možný protékající proud pak může být až 1 A při trvale sepnutých kontaktech. Na tuto limitaci je třeba brát zřetel především při designování nových testů.

Přepínací logiku lze rozdělit na dvě části, první část je samotné zapojení relé. Zapojení relé je provedeno do dvou samostatných větví, každá větev představuje jednu dvoupólovou sběrnici. Relé jsou primárně brána jako pár a to tak, že liché číslo kontaktu je vždy plus (HI) a sudé číslo mínus (LOW). Pár relé je pak propojen se všemi ostatními relé pomocí dvou společných cest podle polarity. Obě větve lze propojit pomocí kontaktů relé (K19,20 a K23, K24). Svorky J3, J4, J5 a J6 slouží pro připojení měřících přístrojů, popřípadě napájecího zdroje nebo například nějakého signálu. Zvláštní funkci mají relé K43 a K44. V některých případech může být dvoupólový režim (diferenční) nevhodný a docházelo by tak redukci použitelných relé na polovinu. Relé K43 a K44 umožňují přepnout logiku do jednopólového režimu a připojit tak všechny relé kontakty, na společnou kladnou cestu.

3.3 Sběrnice RS-485



Obr. 3.3: Izolace RS485

Obr.:3.3 vyobrazuje zapojení zapojení RS-485, respektive části, která se stará o převod UART4 na RS-485. Pro tento účel byl vybrán obvod MAX485CSA, který převádí signál z Uartu na stadardní signál RS-485. Ze schématu je patrné, že na pin „RO“ je připojen UART4RX a na pin „DI“ UART4TX. RE a DE jsou propojeny a slouží k řízení směru komunikace. Právě řízení komunikace ve dvouvodičovém provedení je největší úskalí a z důvodu budoucího ladění programu pro řízení komunikace je v obvodu přidán jamper J20. Díky tomu si lze vybrat, kterým pinem se bude přepínat komunikace přijímáním a odesíláním. Výhodou GPIO16 je jeho velmi snadná softwarova ovladatelnost, avšak vzhledem k tomu, že komunikace může mít vysoké nároky na

časové zpoždění, může být lepší využít pin RTS, který je pro tuto aplikaci vhodnější, avšak náročnější na ovládání. Při komunikaci na větší vzdálenost musí být v zapojení zahrnuta tzn. kompenzace vedení. Tato kompenzace se nazývá terminátor. Terminátor je rezistor, který propojuje oba datové vodiče (obr.:3.4 odpor R44). Ideální hodnota rezistoru je 110 Ω. Úlohou terminátora je zabránit odrazům signálu od konců vedení. [14]

Pro uvedení do provozu, je nutné kromě J20 vhodně propojit i J19. Nelze totiž komunikovat zároveň po sběrnici RS-232 a RS-485, protože obě sběrnice využívají UART4.

Nejdříve je nutné uvolnit pin RST pro určenou funkci. PIN RST a CTS jsou totiž sdíleny s HDMI a proto je nutné HDMI deaktivovat - to se provede následujícím postupem, opět musí mít uživatel práva roota:

Listing 3.4: Deaktivace HDMI

```
mkdir /mnt/boot
mount /dev/mmcblk0p1 /mnt/boot/
nano /mnt/boot/uEnv.txt
```

Sekvence příkazů vytvoří složku „boot“ v adresáři mnt a v ní textový soubor, do kterého je třeba přidat následující řádek:

Listing 3.5: Editace souboru uEnv.txt

```
optargs=capemgr.disable_partno=BB-BONELT-HDMI,BB-BONELT-HDMIN
```

Textový soubor je nutné po té uložit a provést reboot. Po té je ještě třeba zadat příkaz, který nám potvrdí že bylo rozhraní HDMI bylo deaktivováno.[12]

Listing 3.6: Vylistovaní aktivních rozhraní

```
cat /sys/devices/bone_capemgr.*/slots
```

Odpověd by měla vypadat takto:

Listing 3.7: Odpoveď

```
4: ff:P-O-L Bone-LT-eMMC-2G,00A0,Texas Instrument,BB-BONE-EMMC-2G
5: ff:P-O-- Bone-Black-HDMI,00A0,Texas Instrument,BB-BONELT-HDMI
6: ff:P-O-- Bone-Black-HDMIN,00A0,Texas Instrument,BB-BONELT-HDMIN
```

Pro komunikaci v Pythonu byla použita knihovna PySerial, kterou lze nainstalovat takto:

Listing 3.8: Instalace PySerial

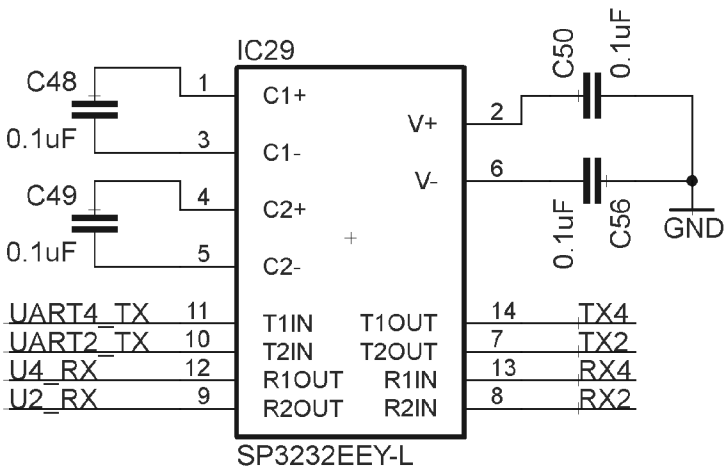
```
apt-get update
apt-get install python-serial
```

Komunikaci lze provést jako v následující ukázce. Nejdříve se vytvoří instace třídy serial a vyplní se parametry specifické pro RS-485. Pote se nastaví komunikace do módu RS-485. Pomocí struktury se nastaví volitelné parametry, jako jsou například volitelná zpoždění při střídání vysílacích stran. Potom již lze využít metody jenže skýta třída serial k odeslání požadovaných hodnot. [11]

Listing 3.9: Ukázka komunikace po RS-485

```
import serial, fcntl, struct, time

ser = serial.Serial(
    port='/dev/ttyO4', # nastaveni portu Uart4
    baudrate=9600,  # nastaveni hodnoty baudrate
    timeout=1,  # hodnota timeout
    parity=serial.PARITY_NONE, # Parita (none, odd, even)
    stopbits=serial.STOPBITS_ONE, # pocet stopbitu
    bytesize=serial.EIGHTBITS
)

TIOCSRS485 = 0x542F  # nastaveni modu RS-485
serial_rs485 = struct.pack('IIIIIIII', # nastaveni parametru
                           RS485_FLAGS, # flag
                           0, # zpozdeni pred odeslanim us
                           0, # zpozdeni po odeslani v us
                           RTS, # the pin number used for DE/RE
                           0, 0, 0, 0 # volitelne hodnoty
                           )

fd=ser.fileno()
fcntl.ioctl(fd, TIOCSRS485, serial_rs485)
time.sleep(0.2)
ser.write([0xFF, 0xA0, 0xFF]) # dodesle hexa hodnoty
time.sleep(0.2)

ser.close() # ukonceni
```

3.4 Sběrnice RS-232



Obr. 3.4: RS-232 reciever/traciever

O převod signálu z Uart2 a Uart4 se stará dvojice obvodů SPS3232EEY-L. Obvod převádí signál ze sériového rozhraní na standard RS-232. Obvod je napájen 3,3V. Jedná se o převodník TTL na RS232. Obsahuje dvakrát dva oddělovače konvertujících napěťové úrovně. Napětí pro RS 232 se získává pomocí nábojové pumpy a výstupní napětí závisí na kvalitě použitých kondenzátorů C48 a C49. Vyvedení dvojice rozhraní RS-232 je z toho důvodu, že propojení bývá typicky dvoubodové. Další rozhraní je tedy jako rezerva. [13]

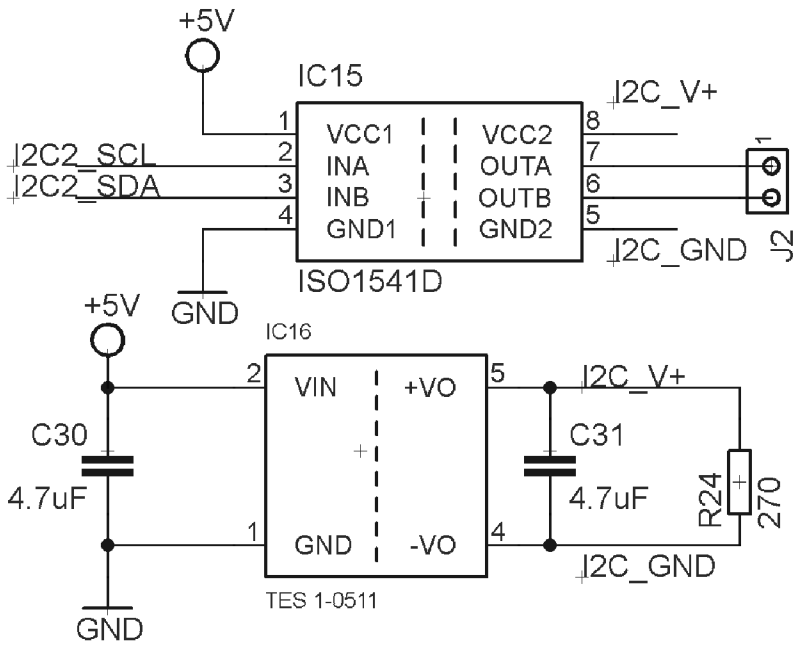
Pro komunikaci je nutné mít nainstalovanou knihovnu PySerial, stejně jako je tomu u RS-485. A pokud je pro komunikace nutné použít piny RTS a CTS, je nutné, aby HDMI bylo ve stavu disable. Pro komunikaci po RS-485 musí být J19 v poloze, jenže odpovídá RS-232 enable (viz. schéma v příloze).

Zde je ukázka komunikace po RS-232.

Listing 3.10: Ukázka komunikace po RS-485

```
import time, serial
#import tridy serial z knihovny PySerial
uart4_file = '/dev/ttyO4'
# nastavni uartu4 (pro uart2 '/dev/ttyO2')
baud = 115200  # nastaveni baudrate
ser = serial.Serial(uart4_file, baud)
# ser object tridy Serial
ser.write([0xff, 0xff, 0xff])
# Odeslani hexa-hodnoty
time.sleep(1)  # zpozdeni
ser.close() # ukonceni
```

3.5 Sběrnice I2C



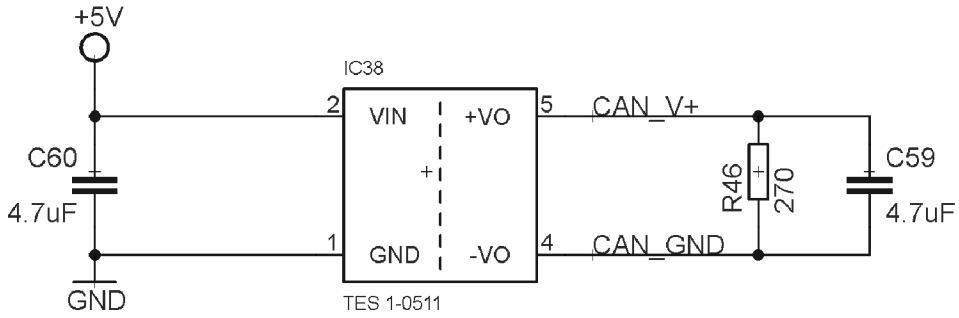
Obr. 3.5: Převodník RS485

Na obrázku obr.:3.5 je zobrazeno, jakým způsobem byla realizována izolace I2C sběrnice pro možnost propojení s testovaným zařízením. Izolace je provedena podobně, jako je tomu u sběrnice CAN. Izolování vnější sběrnice od zbytku obvodu

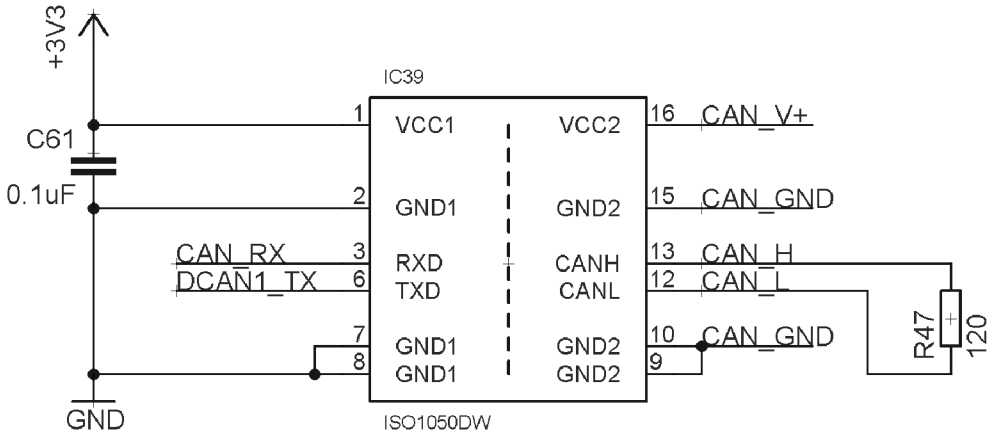
zajišťuje obvod ISO1541D. Tento obvod slouží pro galvanické oddělení sběrnice, vyžaduje však zvlášť napájení levé a pravé strany. Pro vytvoření oddělených 5V a GND byl využit modulový spínaný zdroj TES-10511. Pravá a levá strana musí být odděleny od zbytku obvodu. Pro vnější komunikaci bylo vybráno rozhraní I2C2, nedochází zde ke konfliktu pinů a ani není třeba dělat úpravy v device tree.

Komunikace v Pythonu se provádí skrz nainstalované knihovny od Adafruit, a to konkrétně třída Adafruit I2C, jenže obsahuje metody pro čtení a zápis. Z této metody vychází i již popsaný ovladač pro MC23017.

3.6 Sběrnice CAN



Obr. 3.6: Izolace CAN 1



Obr. 3.7: Izolace CAN 2

Sběrnice CAN je vyvedena na svorkovnice. Sběrnice však musí být izolována tak, aby nebylo možné zničit celé testovací zařízení například vlivem zkratované sběrnice u testovaného zařízení. Samotná izolace je provedena pomocí obvodu ISO1050DW

v kombinaci s modulovým zdrojem TES 1-0511, jenže slouží jako galvanické oddělení napájení pro pravou stranu izolátoru. V tomto zapojení je nutné dbát zvláštní pozornosti u layoutu desky tak, aby ta část obvodu, která se nachází za pomyslnou čerchovanou čárou, byla oddělena od zbytku obvodu. Při teoretické závadě na straně připojeného testovacího obvodu tak může být zničen jen obvod ISO105DW. Zapojení je vyobrazeno na obr.:3.6 a 3.7

Jak je uvedeno v úvodu, Beaglebone Black již obsahuje sběrnici CAN na GPIO pinech. Lze tedy přistoupit přímo ke komunikaci. Ke komunikaci přes sběrnici CAN slouží nástroj can-utils. Tento nástroj představuje možnost jak nastavit rozhraní, komunikovat nebo filtrovat komunikaci. Výhoda hotového nástroje spočívá především v tom, že z pohledu programátora testovacích předpisů není nutná znalost standardu pro CAN sběrnici, ani pro její principiální fungování. Programátor musí znát pouze ID sledovaných uzlů a případná data pro odesílání. [10] Pro instalaci Can-utils, je nutné zadat do console následující příkazy, opět s právy roota:

Listing 3.11: Instalace Can-utils

```
apt−get update && apt−get install git
cd /tmp
git clone git://gitorious.org/linux−can/can−utils.git
cd can−utils/
make
make install
cd ~
#nebo
sudo apt−get install can−utils
```

Poté není nutné ujistit se, zda je interface povolený, a to se provede pomocí:

Listing 3.12: Příkaz pro ověření can enable

```
sudo modprobe can
```

Před aktivací interface CAN1 je třeba nastavit bitrate. Jeho hodnota závisí na připojeném zařízení, respektive je nutné nastavit takovou hodnotu, která je totožná s testovacím zařízením. Hodnoty bitrate mohou být následující (1M, 800K, 500K, 250K, 125K, 50K, 20K, 10K).

Konkrétní rozhraní, v tomto případě CAN1, je aktivuje a nastaví bitrate pomocí následujících příkazů:

Listing 3.13: Nastavení CAN1

```
ip link set can1 type can bitrate 10000 listen-only on
sudo ifconfig can1 up
```

Odesílat či přijímat komunikaci lze následujícími příkazy:

Listing 3.14: Ukázka komunikace CAN1

```
candump can1 #slouzi pro zachytavani veskere komunikace na sbernici
candump can1,0x111:0xFFF # slouzi pro zobrazeni komunikace pouze z ID 111
cansend can1 111#1122334455667788
# Takto lze poslat data pro uzel s ID:111
# (0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88)
```

Volání z Python skriptu se provádí takto:

Listing 3.15: Volání z Pythonu

```
import subprocess # import tridy subprocess

output = subprocess.Popen(["candump","can1"],
                                    stdout=subprocess.PIPE).communicate()[0]
# Zavolani shell prikazu z python skriptu
# a prirazeni do promenne output

print output
#vypis promenne output, coz je textove pole
```

4 PŘÍSTROJE PRO TESTOVÁNÍ

Jak bylo již dříve uvedeno, externí měřící přístroje jsou pro celý systém klíčové. V prvotním návrhu bez použití dalšího USB HUB je počítání s připojením maximálně 2 měřících přístrojů. Použití typu měřicího přístroje záleží na charakteru zařízení, které je nutné testovat. Při návrhu byl kladen důraz na to, aby si celé zařízení vystačilo s běžnými měřícími přístroji. S měřícími přístroji se komunikuje pomocí SCPI příkazů. Lze tedy k testovacímu zařízení připojit jakýkoliv měřící přístroj, který lze připojit USB nebo ethernetem a řídit SCPI příkazy. Pro psaní testovaného scénáře je však velmi nepraktické zadávat do kódu přímo SCPI příkazy. Proto byla vybrána zařízení, která by měla být schopna pokrýt nejčastější požadavky na testování. A speciálně pro ně byly vytvořeny třídy a metody pro jejich ovládání tak, aby programátor testovacího scénáře nemusel znát výčet zmíněných SCPI příkazů.

Podporované typy zařízení:

- Sourcemetr (Agilent U2722A)
- Říditelný napájecí zdroj (TTI cpl303, Hameg HMP4030 - HMP2030)
- Multimetr (Agilent 34411A, Metex usb)
- Switching unit(Keithley 3706, Agilent 34972A)
- Generátor funkcí (Agilent 33220A)

Pomocí vytvořených tříd (ovladačů) lze typicky ovládat i jiné přístroje stejného typu a výrobce. Výrobci napříč svými zařízeními udržují syntaxi SCPI příkazů stejnou. Takže například pomocí ovladače, který je určen pro model Agilent U2722, lze po vhodné úpravě (viz. kapitola Software) ovládat téměř všechny multimetry od firmy Agilent. Každý přístroj má však své specifické funkce, které je třeba případně doplnit do ovladače.

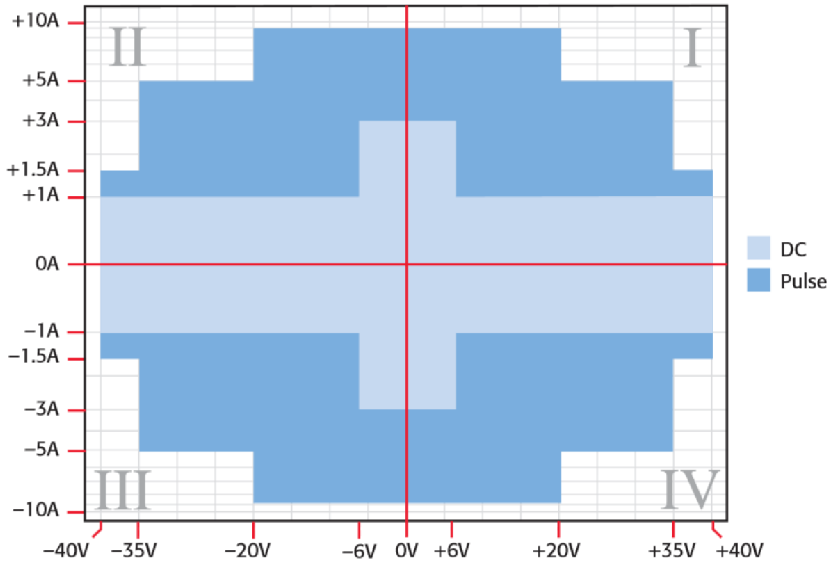
4.1 Switching unit

Switching unit je přístroj navržen právě pro testování či dlouhodobé měření. Primární funkcí switching unit je přepínání signálů a to digitálních i analogových. Přístroje typicky obsahují i multimetr, díky kterému jsou schopny signály nejenom přepínat, ale i měřit jejich veličiny. Přístroje jsou prodávány s množstvím rozšiřujících karet. Funkcionalita přístroje je pak ovlivněna tím, zda se připojitelné karty v zařízení nacházejí. Výrobce nabízí k oběma podporovaným modelům nespočet těchto karet, které lze rozdělit do dvou skupin podle účelu použití. První skupinou jsou karty sloužící k propojování různých signálů, k dispozici jsou maticové propo-

jovací pole, sběrnicové propojovací pole, mutiplexovací pole. Druhou skupinou jsou karty, které slouží k analýze signálu. Mnohdy bývají tyto funkce sloučeny i do jedné karty.

4.2 Sourcemetr

Sourcemetr je základním prvkem celého zařízení. Je to přístroj, který má celkem 4 pracovní módy a říká se též, že pracuje v čtyř kvadrantově, viz obr: 4.1.



Obr. 4.1: Soucemetr a jeho pracovní oblast[9]

V praxi to znamená, že se přístroj dokáže chovat jako zdroj napětí nebo zdroj proudu. Další funkcí je „Sink“, kdy se přístroj chová jako zátěž - opět buď napěťového nebo proudového charakteru. Nastavení probíhá vždy nejdříve výběrem módu „Sink“ nebo „Source“. Poté je nastavena jmenovitá hodnota napětí nebo proudu. Poté je nutné nastavit „complience“ což je maximální limit napětí nebo proudu, který zdroj může dodat.

Zde je příklad nastavení sourcemetru, jako zdroj napětí 5 V s limitací 300 mA a následné vyčtení protékajícího proudu.

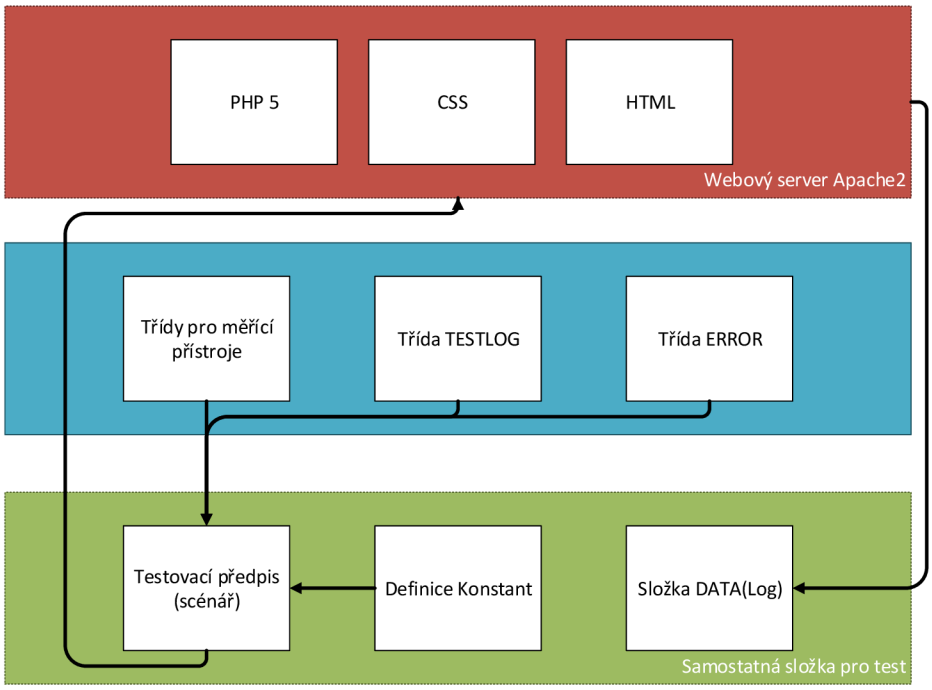
Listing 4.1: Nastavení sourcemetru

```
*RST (resetovani pristroje do defaultniho nastaveni)
:SOUR:FUNC VOLT (Nastaveni sourcemetru jako zdroj napeti)
:SOUR:VOLT:RANG 0.2 (Nastaveni mericiho rozsahu na 200mV)
:SOUR:VOLT:LEV 5 (Nastaveni hodnoty napeti na 5V)
:SENS:FUNC CURR (Vyber merene velicinu, proud)
:SENS:CURR:PROT 100e-3 (Nastaveni proudove limitace)
:SENS:CURR:RANG 10e-3 (Nastaveni mericiho rozsahu)
:OUTP ON (Zapnuti vystupu)
:READ? (Vycteni hodnoty proudu)
```

[9] SCPI příkazy jsou konkrétně pro Agilent U2722A od firmy Keysight Technologies. Při práci s tímto konkrétním přístrojem byla objevena řada specifik a chování. Prvním z problémů, které nastaly, byl jev, díky kterému docházelo ke zničení relé kontaktů, konkrétně k jejich „slepení“. Po diagnostice pomocí osciloskopu byla objevena neschopnost sourcemetru zalimitovat protékající proud několik nanosekund hned po sepnutí relé, pokud je sourcemetr používán jako zemnící prvek. Při sepnutí relé dojde na několik desítek nanosekund ke zkratu, což je důvodem zničení relé. Tento problém byl vyřešen zvolením jiného testovacího zapojení. Další nevýhodou je neschopnost nastavení sourcemetru konkrétní kvadrant, ve kterém má měřit. To vede k chybám především u chybových stavů, kdy, pokud Sourcemetr nenajde měřenou hladinu v kladném intervalu, změní svoji polaritu a hledá i v záporném intervalu, což může vést k poškození testovaného zařízení a je třeba s tím počítat při tvorbě testovacího scénáře.

5 SOFTWARE

Strukturu softwaru lze vidět na obrázku obr.:5.1. Celý systém lze rozvrhnout dvou celků, a to na část psanou v jazyku Python 2.7(3.4) a část ovládacího rozhraní, která je programovaná v PHP. O samotnou komunikaci s měřícími přístroji, komunikace přes sběrnice a ovládání propojovacího pole se stará právě Python. Pro systém testování bylo nutné naprogramovat systém, který propojí ovládání měřících přístrojů, zaznamenávání naměřených hodnot a logování chyb. To celé spojit do použitelného nástroje, díky kterému je možné napsat testovací scénář jen s pomocí připravených funkcí a metod.



Obr. 5.1: Struktura skriptů

Struktura je následující. Každý test se skládá z testovacího scénáře <název>-Test.py a z definice konstant <název>-Limits.py. V souboru definice konstant se nachází především limity pro měření. Důvod, proč jsou odděleny od testovacího scénáře, je ten, že testované zařízení se při vývoji testuje ve více verzích již při jeho vývoji. Různé revize však mohou dosahovat jiných hodnot a často je dolaďovat.

Upravovat tyto meze pak může i uživatel bez jakékoliv znalosti programovacího jazyku Python. Testovací scénář může využívat množství vytvořených tříd. Třídy jsou trojího typu. Třídy pro měřící přístroje, třída pro logování a třída pro zaznamenávání chyb. Z vytvořených tříd byla vytvořena generická dokumentace pomocí programu Doxygen. Dokumentaci a popis jednotlivých tříd lze nalézt v příloze (Software/Dokumentace Doxygen/html).

5.1 Python

5.1.1 Ovladače externích přístrojů

Ovladače přístrojů jsou třídy, které využivají VISU, pro komunikaci s přístroji. Každý přístroj má svoji vlastní třídu a metody v závislosti na jejich účelu. Metody vychází z použití SCPI příkazu a skládají tak dohromady řetězec či sekvence příkazů, které provádí požadovanou funkci. Například pokud požadujeme po voltmetru, aby změřil stejnosměrné napětí. Toto lze provést například pomocí metody config('volt:dc', 10, 1, True) a hodnotu následně vyčíst metodou read(). To stejné v SCPI by se provedlo: CONF:VOLT:DC, READ?. Rozdíl spočívá v tom, že metoda config v sobě skrývá veškerou funcionalitu CONF. Uživatel tedy nemusí studovat SCPI příkazy z manuálu. Třídy pro měřící přístroje jsou psány tak, aby názvy jednotlivých metod byly nejlépe totožné i totožnými parametry. Důvod je ten, aby budoucí uživatel měl co největší komfort a ovládání přístrojů od rozdílného výrobce se jevilo jako totožné. Každý ovladač (třída) obsahuje i self test. Ten slouží k ověření funkčnosti, napříkalad při připojení přístroje. Typicky bývá problém u nastavení Visa-name, které se na každé stanici mění. Kromě tříd pro ovladače, byla vytvořena třída visasim, která je v hodná pro vývoj další ovladačů i přesto, že fyzicky měřící přístroj není k dispozici.

5.1.2 Logování výsledku a chyb

Třída testlog, vytváří protokol o naměřených datech. Výstupem je textový soubor .log. Soubor v sobě obsahuje všechny důležité výsledky a to vždy s možným rozsahem. Třída obsahuje několik metod pro logování naměřených hodnot a pro logování možných chyb. Zároveň se stará o vytváření těchto log a složky DATA, kde jsou pak tyto soubory shromažďovány. Vygenerované soubory vytváří ve formátu «název-testu>-pořadové číslo.log". Každý test je opatřen pořadovým číslem, pokud není zadán vlastní název. Zároveň je ošetřeno vytváření logů s již použitým jménem. Typicky lze místo pořadového čísla uvést sériové číslo testovaného zařízení a

tím tak odlišit výsledky testů. Třída opět obsahuje self-test jako možnost ladění při budoucím rozšíření.

S třídou testlog se váže třída error. Jedná se jednoduchou třídu, jejíž jediným úkolem je počítaní chyb. Na konci každého testu se konečný počet chyb vypíše pomocí zmíněné třídy testlog. Uživatel tak hned vidí, zda test prošel či nikoliv.

5.1.3 Testovací scénář, metodika vytváření

Při psaní testovacího scénáře je nutné dodržovat určitou strukturu a pravidla. Každý test by měl obsahovat určité náležitosti, jako jsou obecné informace v hlavičce logu o jaký test se jedná a v jaké je aktuální verzi. Ukázkový test je obsažen příloze. Součástí testu je vždy měření času, jak dlouho test trvá. Zabezpečuje správné připojení všech testovacích přístrojů a na konci je jich správně odpojení, případně vyprázdnění mezi-paměti, pokud nastala chyba a příkaz nebyl dokončen. Přístroje je vždy nutné uvést do výchozího stavu, bez ohledu na to, jak test odpadl.

Každý test musí obsahovat některé prvky viz. ukázka kódu 5.1 . Nejdříve je nutné naimportovat třídu teslog, error, time, sys. Dále pak třídy pro ovládání přístrojů, kterými se bude testovat. Dále je pak nutné přidat soubor.py, kde jsou uloženy všechny konstanty. Pokud je ten určen pro testování skrze webové rozhraní, je třeba do testu přidat třídu Unbuffered. Třída zajišťuje vyobrazení aktuálních výpisů z konzole, v průběhu testu. Kdyby test tuto třídu neobsahoval, výpis z konzole by se zobrazil až po skončení testu. Dále lze v ukázkovém kódu vidět, funkci closeall, tato funkce zajištuje odpojení přístroje a vyprázdnění jeho mezi-paměti. Ve funkci close all, by se měli nacházet všechny přístroje, včetně logu. Následuje vytvoření instancí jednotlivých tříd s jejich parametry. Nejdříve je třeba vytvořit logovací soubor, do kterého, je pak důležité zapisovat všechny následující události. To je důležité při ladění chyb a zároveň lze tak rychle odstranit, například chybu zapříčiněnou nepřipojeného měřícího přístroje.

Následuje tělo testu, zde by se měli nacházet všechny měření. U všech kroků musí být zajištěno odchytávání chyb a vyjímek. S tím je spojeno i ukončování testu (finally), tento kód v bloku finally se vždy provede a zajistí zmíněné odpojení přístrojů a logu, ještě před tím však sečte chyby a jejich počet zobrazí. Pokud je test velmi dlouhý je vhodné jej rozdělit do menších dílčích bloku, které je možno libovolně do testu přidávat. Výrazně se tím zjednoduší budoucí ladění testu.

Listing 5.1: Příprava nového scénáře

```
import time, sys
import cdaq34972
import testlog
from error import*
from nameLimits import *
class Unbuffered(object): # trida pro stream vypisu
    def __init__(self, stream):
        self.stream = stream
    def write(self, data):
        self.stream.write(data)
        self.stream.flush()
    def __getattr__(self, attr):
        return getattr(self.stream, attr)

def closeall(): # funkce pro odpojeni
    try:
        if DMM != None:
            DMM.close()
            Log.logText('DMM closed.')
                                if Log != None:
            Log.closeLog()
sys.stdout = Unbuffered(sys.stdout) # SSH terminal
Log = testlog.TestLog('HTAS', sys.argv[0])
Log.openLog() # start logging (define file name)
Log.logText('Script version: ' + copyright)
Log.logText('Log file: ' + Log.fileName)
DMM = cdaq34972.Daq34972(rm, '34972A')
DMM.open() #
Log.logText('DMM connected.')
try:
....
finally:
    if Err.getErrorCount() > 0:
        Log.logText("### %d error(s) found !" % Err.getErrorCount())
    else:
        Log.logText("All test passed OK, no errors found :-).")
    Log.logText("### TEST finished at %s" %
```

```
                time.asctime(time.localtime(time.time())))
closeall()
sys.exit(0)
```

5.1.4 Možné rozšíření

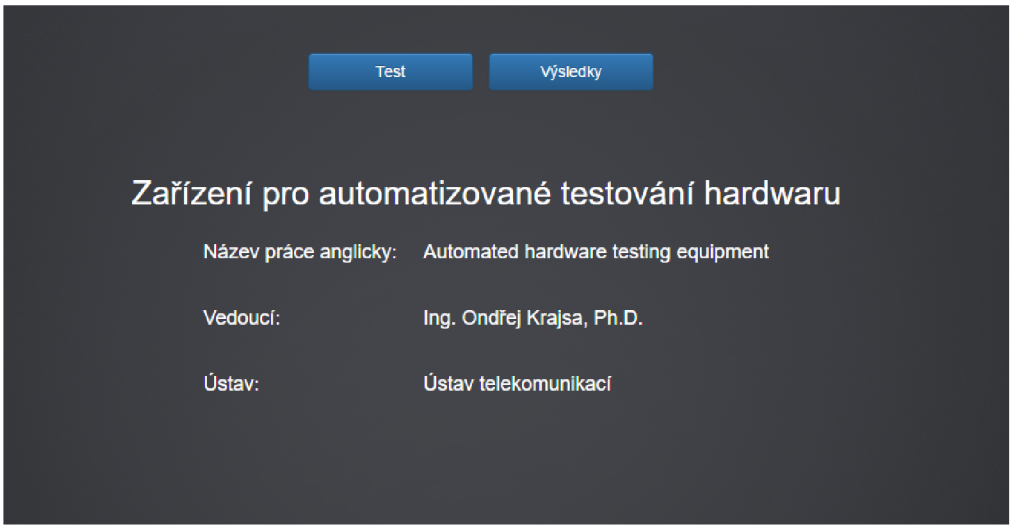
Pro vytváření testovacích scénářů musí být jejich autor pokročilý uživatel pythonu, což může být jedním z omezujících faktorů použití. Důvodem je, že pro každý nový přístroj je nutné napsat nový ovladač (třídu) a nebo použít některou starou a vhodně ji rozšířit. Zde je jakési pomezí univerzálnosti nástroje a komfortu jeho užívání. Další vývoj by bylo možné vést směrem vizualizačního programování. Google blokly, je služba jenž nabízí generování funkčního kódu na základě spojování grafických objektů. Již nabízí i možnost vytváření vlastních objektů. Lze tedy teoreticky vytvořit sadu vlastních objektů, které budou reprezentovat třídy a funkce, které jsou již hotové. A pomocí jich vytvářet testovací scénář bez znalosti programovacího jazyku python.

5.2 Webové rozhraní

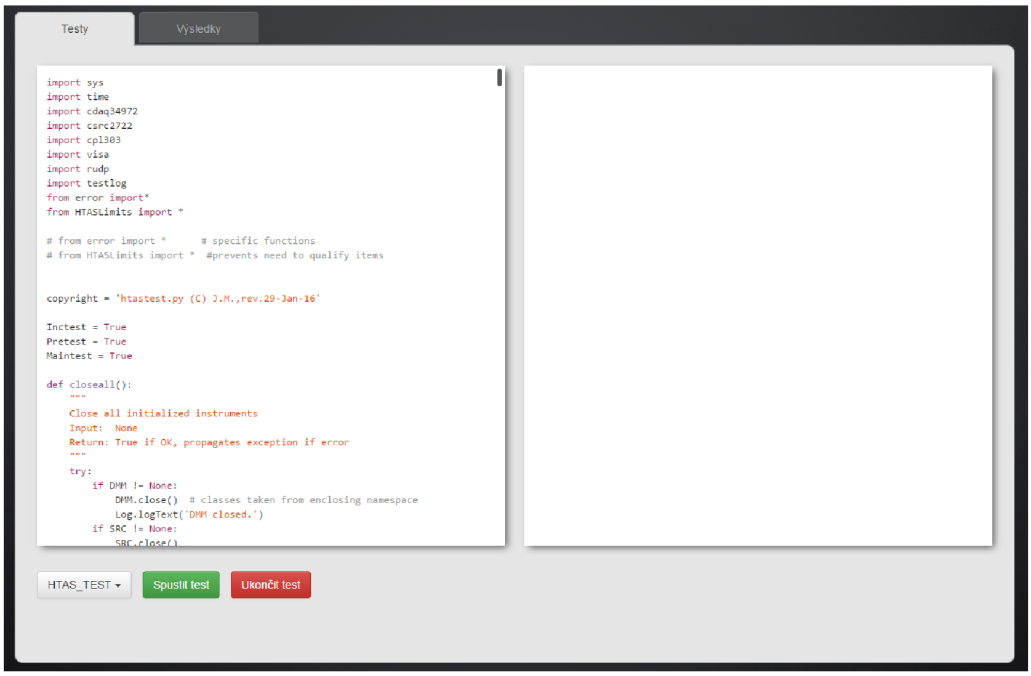
Celé zařízení lze ovládat dvěma možnými způsoby. První způsob ovládání je přímo pomocí terminálu či pomocí grafického UI Debianu, tedy pomocí vzdáleného připojení přes SSH, například pomocí Putty nebo VNC. Tento způsob je však vhodný spíše pro administraci a uživatel je nucen znát celou funkcionalitu systému. Lze tedy tímto způsobem spouštět připravené testy, ale především je upravovat nebo tvořit nové. Webové rozhraní je primárně navrženo pro obsluhu. Webové rozhraní běží přímo na Beaglebone Black. Pro vykonávání testu je třeba připojit se na zařízení pomocí webového prohlížeče na odpovídající IP adresu. Webové zařízení disponuje pouze nejnutnější funkcionalitou tak, aby obsluha nebyla schopná nějakým způsobem test ovlivnit. Webové rozhraní předem počítá již s vypracovanými testovacími scénáři, které bude obsluha opakovaně spouštět. Z každého měření se uchovává textový dokument s naměřenými hodnotami. Tento dokument se uchovává na serveru. Složku s logy lze například pravidelně synchronizovat se vzdáleným úložištěm. V rámci web serveru a celého zařízení není řešená problematika zabezpečení. Zabezpečení bývá ve firmách tradičně řešeno tak, že testovací stanoviště nemá přístup k internetu a pouze omezený přístup do vnitřní sítě.

5.2.1 Uživatelské rozhraní

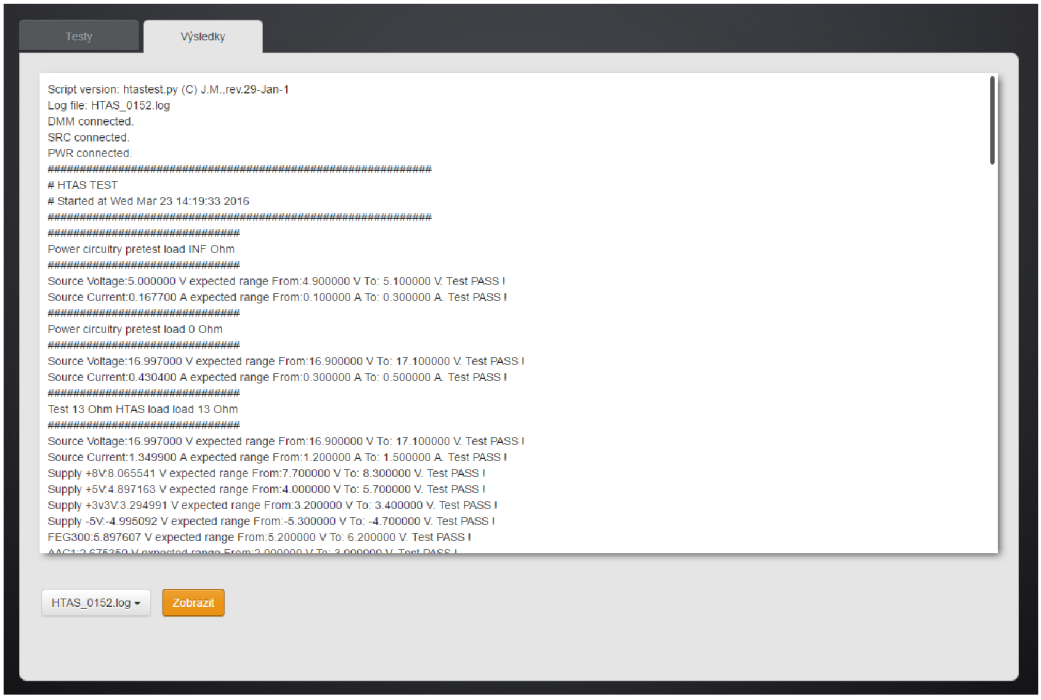
Pro snažší ovládání slouží vytvořenéuživatelské rozhraní, které je rozděleno do dvou stránek. Úvodní stránka obsahuje základní informace o diplomové práci se dvěma tlačítky pro vstup do hlavní sekce, viz obr.:5.2. Hlavní sekce obsahuje dvě záložky: Testy a Výsledky, jak lze vidět na obr.:5.3 a obr.:5.4.



Obr. 5.2: Struktura skriptů



Obr. 5.3: Struktura skriptů



Obr. 5.4: Struktura skriptů

5.2.2 Hlavní sekce

Levá část stránky slouží pro zobrazení zdrojového kódu prováděného testu. Zdrojový kód je automaticky načten po vybrání položky z rozbalovací nabídky. Stisknutím tlačítka „Spustit test“ je spuštěn samotný skript a jeho výstup je přesměrován do pravé části stránky, která slouží jako konzole. Ta je tvořena elementem iframe, do kterého je vypisován výstup z php skriptu.

5.2.3 Spuštění testu

Jak je zmíněno v předchozím textu, test je spuštěn pomocí php skriptu. Tento skript je spuštěn nastavením zdrojové adresy elementu iframe po stisknutí tlačítka „Spustit test“. Spuštění samotného testu je realizováno zavoláním metody proc-open, která je předávána cesta k příkazu z uživatelského rozhraní. Jelikož je nutné mít výstup z konzole v reálném čase, je vypnuto ukládání výstupů do vyrovnávací paměti a výstup je rovnou vypisován do elementu iframe. Ze stejného důvodu je nutné vypnout ukládání do vyrovnávací paměti i v testovacím skriptu, v opačném případě dojde k předání dat na výstup až po skončení testu.

5.2.4 Ukončení testu

Pokud je nutné ukončit právě probíhající skript, lze stisknout tlačítko „Ukončit test“. Toto tlačítko pomocí AJAX dotazu spustí php skript, který provede ukončení právě probíhajícího testu pomocí příkazu kill. Jako vstupní parametr metody slouží PID procesu, které je získáno již při spuštění skriptu metodou proc-open. Tato metoda funguje pouze v systémech Linux, systém Windows využívá pro ukončení skriptu metodu taskkill, kde jako vstupní parametr slouží také PID procesu.

5.2.5 Sekce výsledky

V této sekci lze zobrazit výsledky získané z provedených testů. Do rozbalovací nabídky jsou po načtení stránky přidány všechny názvy souborů vyskytující se ve složce s výsledky. Po stisknutí tlačítka je pomocí AJAX dotazu získán celý obsah souboru, který je poté v uživatelském rozhraní zobrazen. Jelikož by bylo nutné pro zobrazení nových výsledků vždy aktualizovat webovou stránku, což je velmi nepraktické, obsahuje uživatelské rozhraní „EventListener“. Po úspěšném dokončení testu je z php skriptu starající se o spuštění procesu vyvolána událost informující o úspěšném dokončení testu. Tato událost je v uživatelském rozhraní zachycena a je provedeno znovunačtení všech názvů položek ze složky s výsledky. Tím je docíleno zobrazení aktuálních hodnot výsledků v rozbalovací nabídce.

5.2.6 Instalace uživatelského rozhraní

Pro správné fungování uživatelského rozhraní je nutné mít funkční apache server a knihovnu php. V systému Windows lze využít balíčků jako např. XAMPP, které již obsahují všechny potřebné balíčky. Ve výchozím stavu má Debian port 80 již obsazený. Nachází se na něm start-up prezentace Beaglebone Black. Kromě webserveru běží ještě řada dalších služeb, které se vážou k funkcionalitě webu. Všechny tyto služby je nutné deaktiovovat. To lze provést pomocí následujících příkazů. [8]

Listing 5.2: Zakázání nepotřebných služeb

```
sudo systemctl disable cloud9.service
sudo systemctl disable gateone.service
sudo systemctl disable bonescript.service
sudo systemctl disable bonescript-autorun.service
sudo systemctl disable avahi-daemon.service
sudo systemctl disable gdm.service
sudo systemctl disable mpd.service
sudo reboot
```

Protože se již ve výchozím stavu v systému nachází apache2, je třeba jej odinstalovat a zase poté opět nainstalovat. To se provede pomocí následujících příkazů.

Listing 5.3: Instalace webserveru Apache2 a php5

```
apt-get remove apache2
sudo apt-get autoremove

sudo apt-get install apache2 -y
sudo apt-get install php5 libapache2-mod-php5 -y
```

Těmito příkazy se nainstaluje apache2 server a knihovna php5, víc není pro spuštění uživatelského rozhraní potřeba. Avšak nyní je po zadání webové adresy dostupný pouze výchozí index.html soubor. Kořenová složka serveru je umístěna v /var/www/html/, zde je nutné nahradit výchozí indexovou stránku soubory vytvořeného uživatelského rozhraní. Po zadaní adresy by již mělo být uživatelské rozhraní přístupné.

6 ZÁVĚR

Cílem této diplomové práce bylo navrhnout zařízení pro testování HW na platformě Beaglebone Black. Prvním bodem práce bylo seznámení se s principy a možnostmi řízení platformy Beaglebone Black a nalézt způsob, jak ji propojit s externími měřicími přístroji a ovládat je. Na základě těchto znalostí navrhnout komplexní systém, který bude schopen otestovat HW. Zařízení by mělo být schopné komunikovat s testovaným hardwarem pomocí běžně používaných sběrnic.

Byl vytvořen hardware i software. Z hardwarové části byla vytvořena expanzní deska pro BeagleBone Black. Expanzní deska obsahuje rozšíření o ethernet a dva USB porty. Dále zajišťuje propojování signálů od testovaného zařízení k měřícím přístrojům. Na expanzní desku byly vyvedeny sběrnice I2C, RS-232, RS485 a CAN, pomocí kterých lze komunikovat s testovaným zařízením. Softwarovou část tvoří třídy pro ovládání a komunikaci s měřícími přístroji, dále byly popsány základní principy jak komunikovat po sběrnicích. Hlavní část testovacího nástroje tvoří třídy pro generování protokolu z testu a reportování chyb. Kromě programů naprogramovaných v jazyku Python byl vytvořen ovládací interface. Ovládací interface má formu webových stránek, slouží jako ovládací rozhraní celé testovací platformy a byl navržen tak, aby jeho ovládání bylo co nejjednodušší.

Celý systém jako celek by měl být schopen pokrýt většinu základních požadavků na testovací platformu. Při vývoji softwaru byl kladen důraz především na snadnou rozšířitelnost a modifikovatelnost. Byl vytvořen postup vytváření testovacích scénářů a definována jejich struktura a náležitosti. Zařízení je téměř připraveno pro nasazení do praxe, ideálně je však potřeba testovací platformu upravit na míru pro konkrétního zadavatele. Požadavky na testovací nástroj se vždy zakládají na charakteru testovaného hardwaru.

LITERATURA

[1] What is BeagleBone Black?. *Beagleboard.org* [online]. 2015 [cit. 2015-12-15]. Dostupné z: http://beagleboard.org/black

[2] Getting Started with BeagleBone & BeagleBone Black. *Beagleboard.org* [online]. 2015 [cit. 2015-12-15]. Dostupné z: http://beagleboard.org/getting-started#hardware

[3] BBB - Working with the PRU-ICSS/PRUSSv2. *Element14.com* [online]. 2015 [cit. 2015-12-15]. Dostupné z: http://www.element14.com/community/community/designcenter/single-board-computers/next-gen_beaglebone/blog/2013/05/22/bbb--working-with-the-pru-icssprussv2

[4] Setting up IO Python Library on BeagleBone Black. *Adafruit.com* [online]. 2015 [cit. 2015-12-15]. Dostupné z:https://learn.adafruit.com/setting-up-io-python-library-on-beaglebone-black/installation-on-ubuntu

[5] CBB-Serial. *Github* [online]. [cit. 2016-04-25]. Dostupné z:https://github.com/lgxlogic/CBB-Serial/tree/master/overlay

[6] *LAN9512 Software User Manual Revision 1.2 (05-01-09)* [online]. 2009 [cit. 2016-05-25]. Dostupné z:http://ww1.microchip.com/downloads/en//softwarelibrary/man-lan95xx/lan9500_lan9500a_lan951x%20software%20user%20manual%20rev.%201.2%20%2805-01-09%29.pdf

[7] PyVISA: Control your instruments with Python. *Pyvisa.readthedocs.org* [online]. 2015 [cit. 2015-12-15]. Dostupné z:https://pyvisa.readthedocs.org/en/stable/python-library-on-beaglebone-black/installation-on-ubuntu

[8] BeagleBone Black webserver install. *Element14.com* [online]. 2015 [cit. 2016-05-25]. Dostupné z:https://www.element14.com/community/community/designcenter/single-board-computers/next-gen_beaglebone/blog/2013/11/20/beaglebone-web-server--setup

[9] *U2722A/U2723A USB Modular Source Measure Units User's Guide* [online]. 2014 [cit. 2016-05-25]. Dostupné z:https://www.keysight.com/main/gated.jspx?lb=1&gatedId=1432725&cc=CZ&lc=eng&parentContId=1369935&parentContType=pt&parentNid=-35489.774878&fileType=VIEWABLE

[10] CAN - popis struktury. *Vyvoj.hw.cz* [online]. 2004 [cit. 2016-05-25]. Dostupné z:http://vyvoj.hw.cz/navrh-obvodu/rozhrani/can-popis-struktury.html

[11] Pythonhosted.org. *PySerial* [online]. 2015 [cit. 2016-04-25]. Dostupné z: https://pythonhosted.org/pyserial/

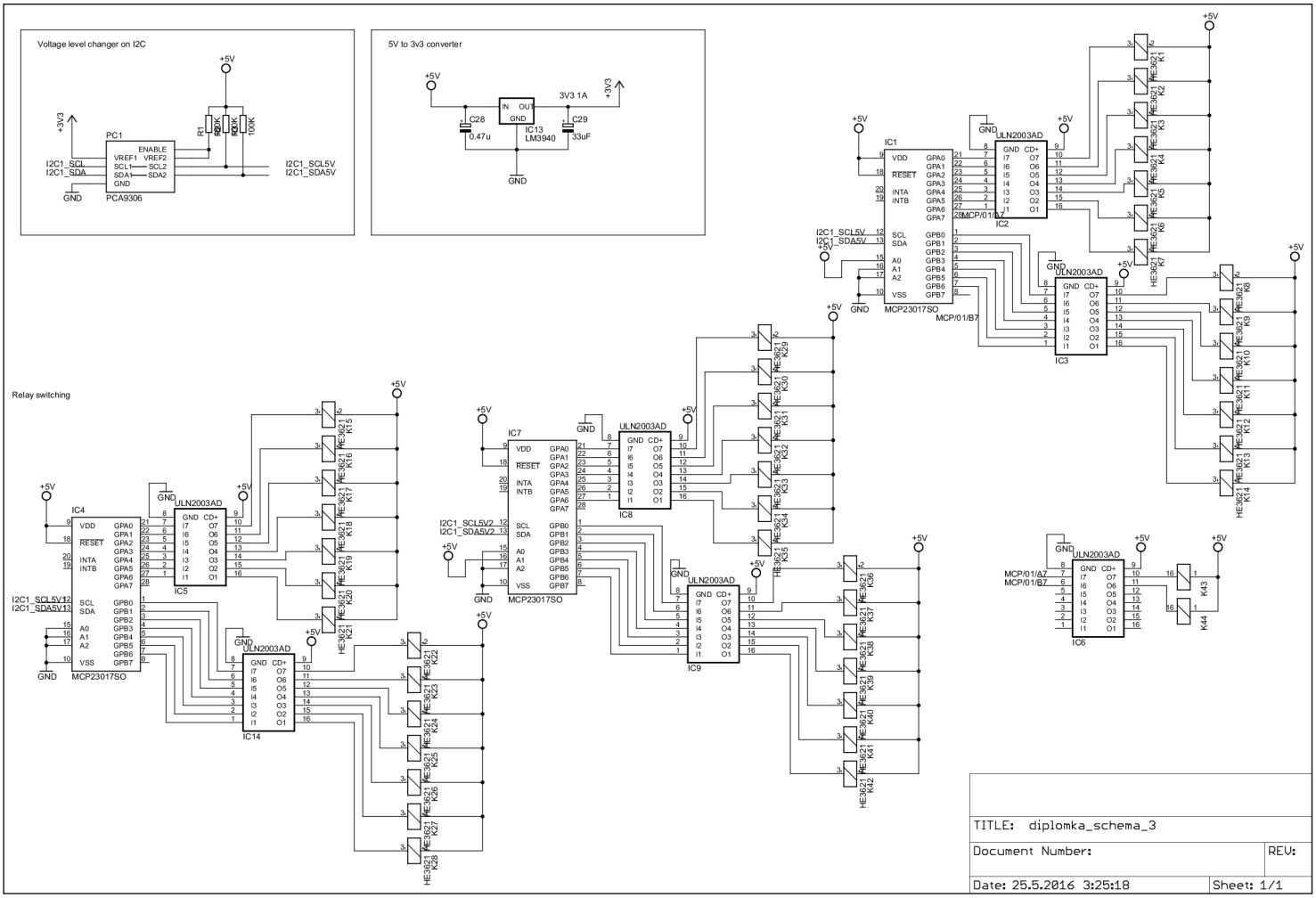
[12] Setting up WiFi with BeagleBone Black. *Adafruit.com* [online]. 2015 [cit. 2016-05-25]. Dostupné z: https://learn.adafruit.com/setting-up-wifi-with-beaglebone-black/hardware

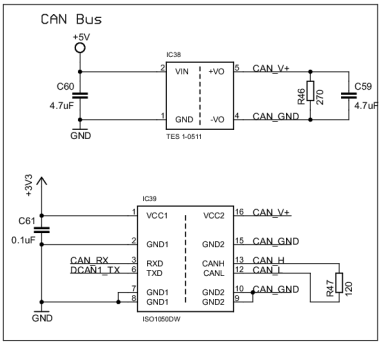
[13] *RS-232* [online]. , 1 [cit. 2016-05-25]. Dostupné z: https://cs.wikipedia.org/wiki/RS-232

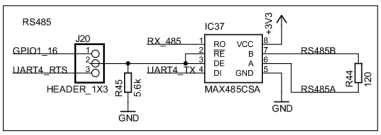
[14] *RS 485 422* [online]. Dostupné z: http://vyvoj.hw.cz/teorie-a-praxe/dokumentace/rs-485-422.html

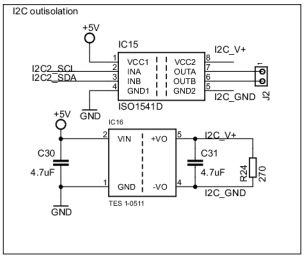
[15] MACHÁČEK, Jan. *Návrh řešení inteligentního domu s bezdrátovými senzory.* Brno, 2014. Bakalářská práce. VUT v Brně. Vedoucí práce Ing. Ondřej Krajsa, Ph.D.

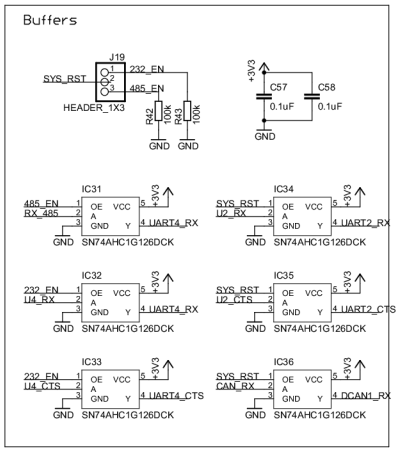
.1 Schéma zapojení

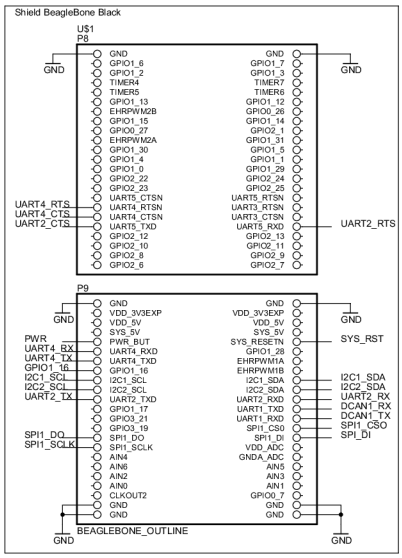

Voltage level changer on I2C
PC1
PCA9306
ENABLE
VREF1
VREF2
SCL1
SCL2
SDA1
SDA2
GND
I2C1_SCL
I2C1_SDA
I2C1_SCL5V
I2C1_SDA5V
+3V3
+5V
R1
R2
R3
100K
5V to 3v3 converter
IC13
LM3940
3V3 1A
IN
OUT
C28
0.47u
C29
33uF
Relay switching
IC1
IC2
IC3
IC4
IC5
IC6
IC7
IC8
IC9
IC14
MCP23017SO
MCP/01/B7
ULN2003AD
HE3621
K1
K2
K3
K4
K5
K6
K7
K8
K9
K10
K11
K12
K13
K14
K15
K16
K17
K18
K19
K20
K21
K22
K23
K24
K25
K26
K27
K28
K29
K30
K31
K32
K33
K34
K35
K36
K37
K38
K39
K40
K41
K42
K43
K44
I2C1_SCL5V12
I2C1_SDA5V13
I2C1_SCL5V2
I2C1_SDA5V2
MCP/01/A7
MCP/01/A7
MCP/01/B7
TITLE: diplomka_schema_3
Document Number:
REV:
Date: 25.5.2016 3:25:18
Sheet: 1/1

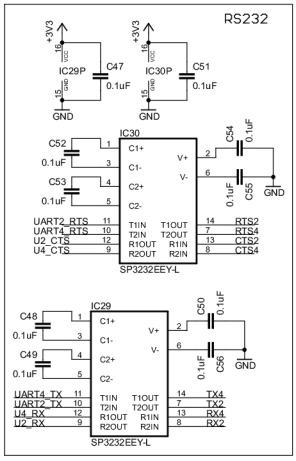


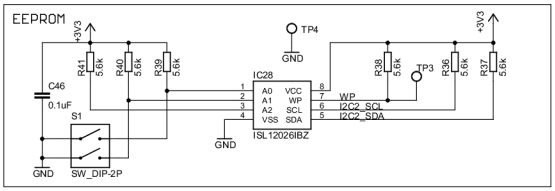












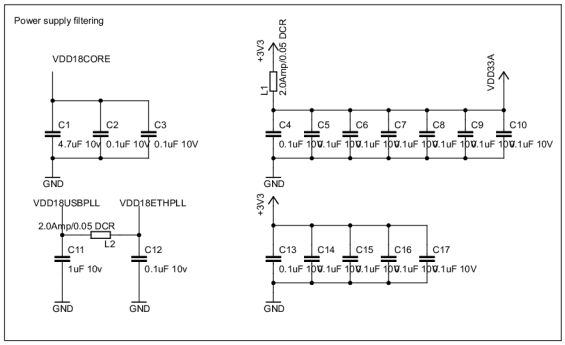
TITLE: diplomka_schema_3

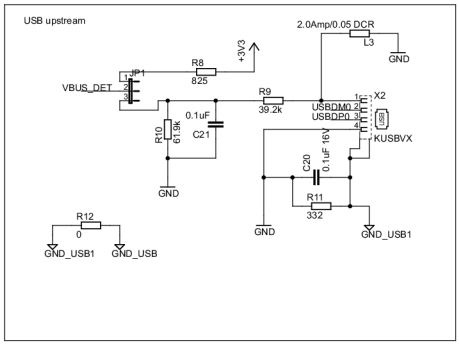
Document Number:

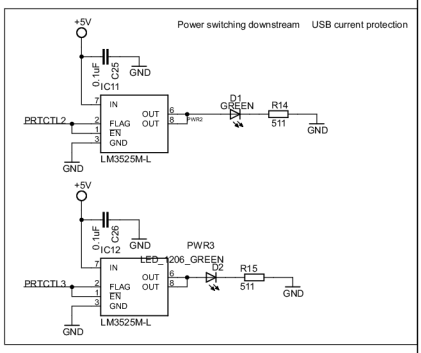
REV:

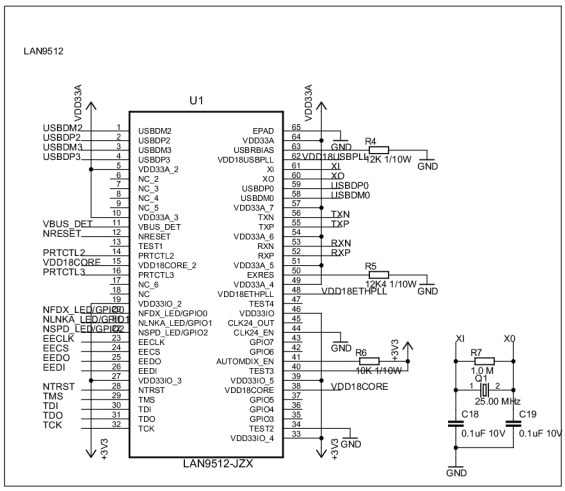
Date: 25.5.2016 3:25:18

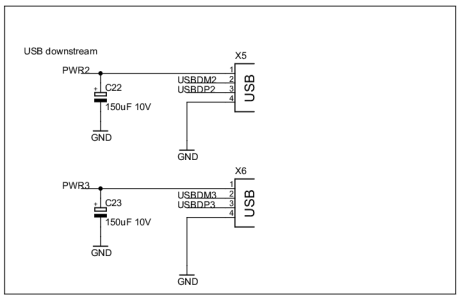
Sheet: 1/1

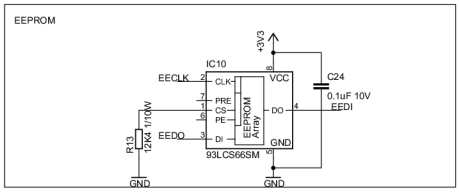


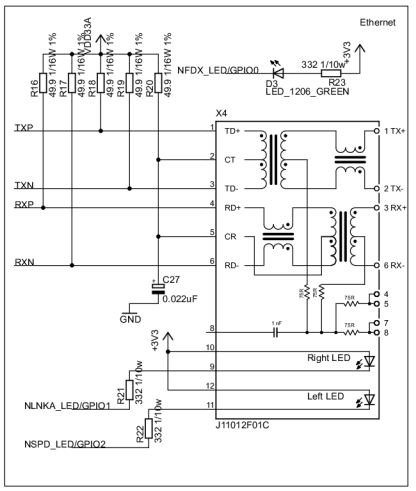




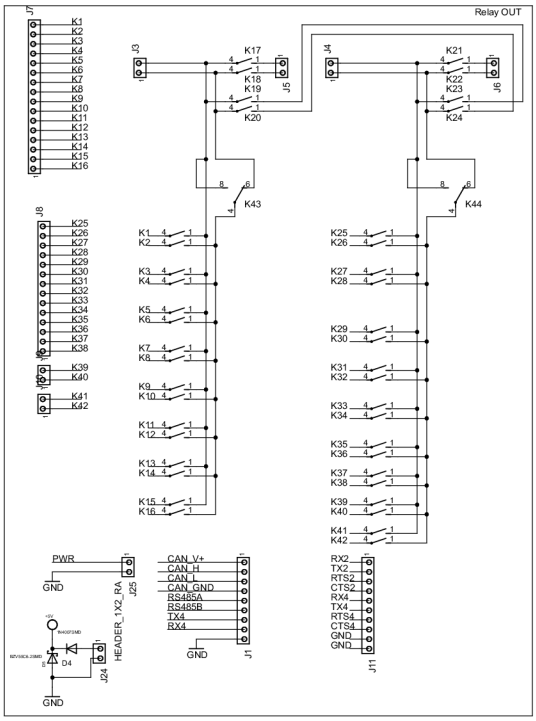






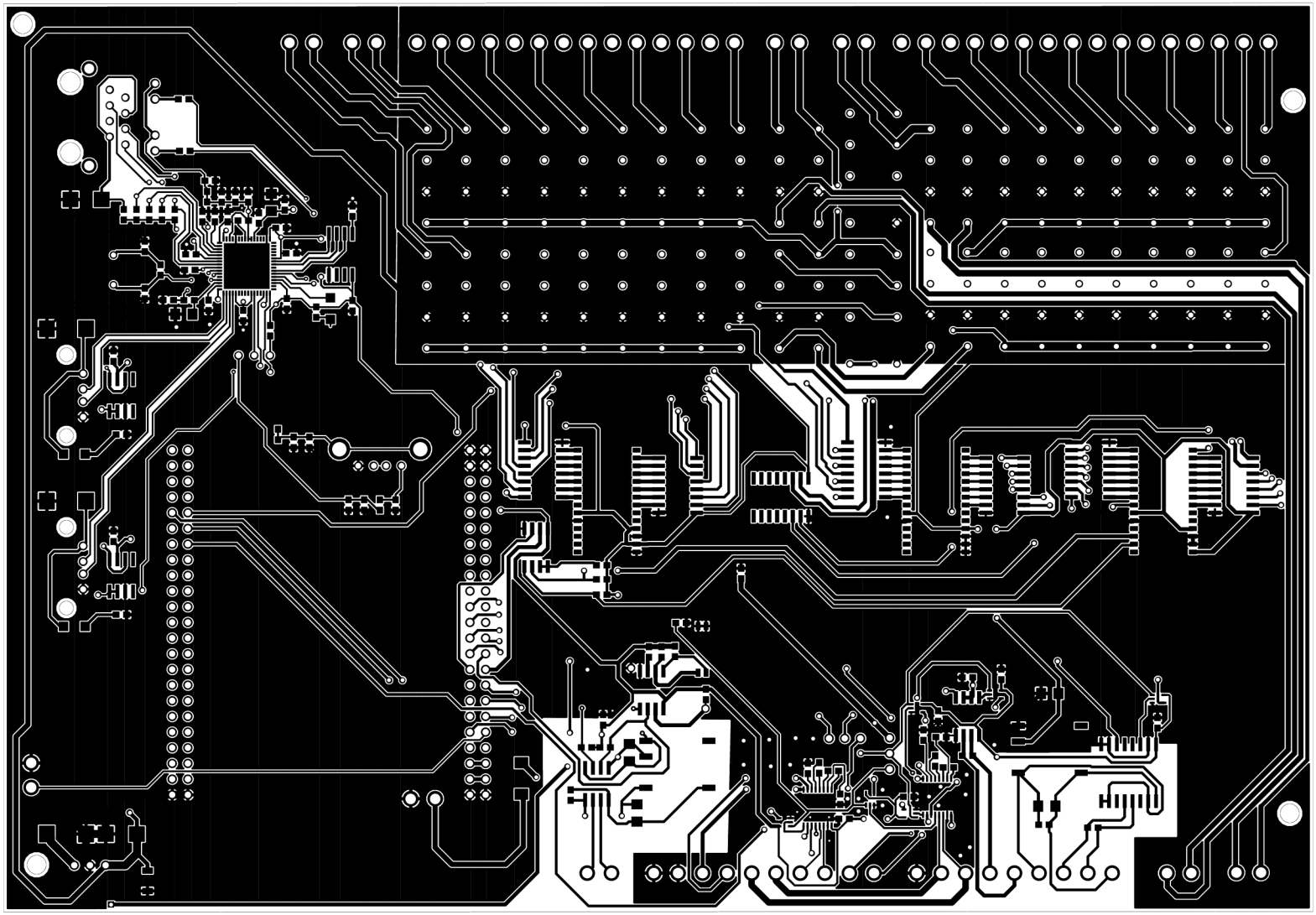


TITLE: diplomka_schema_3	
Document Number:	REV:
Date: 25.5.2016 3:25:18	Sheet: 1/1

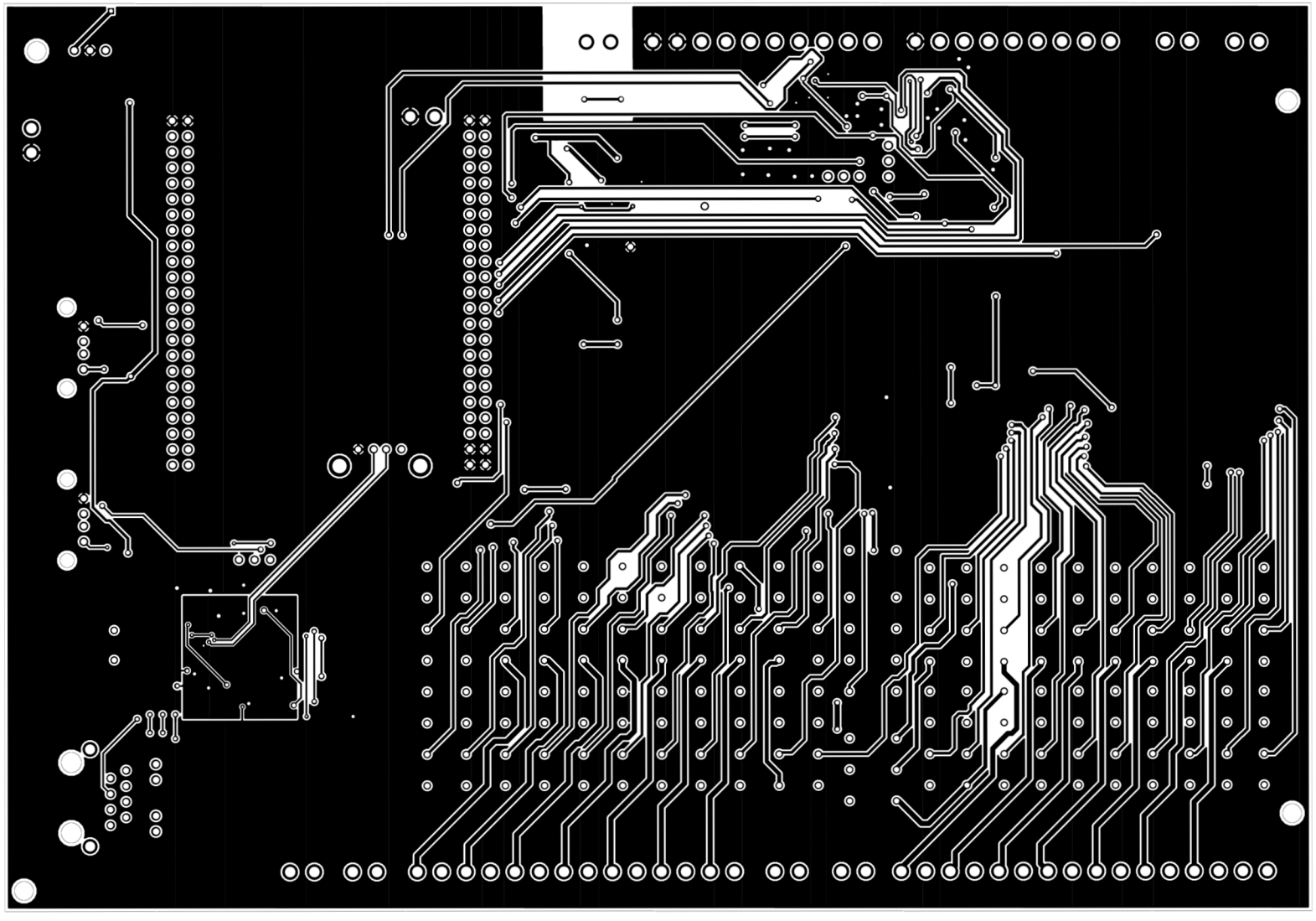


TITLE: diplomka_schema_3	
Document Number:	REV:
Date: 25.5.2016 3:25:18	Sheet: 1/1

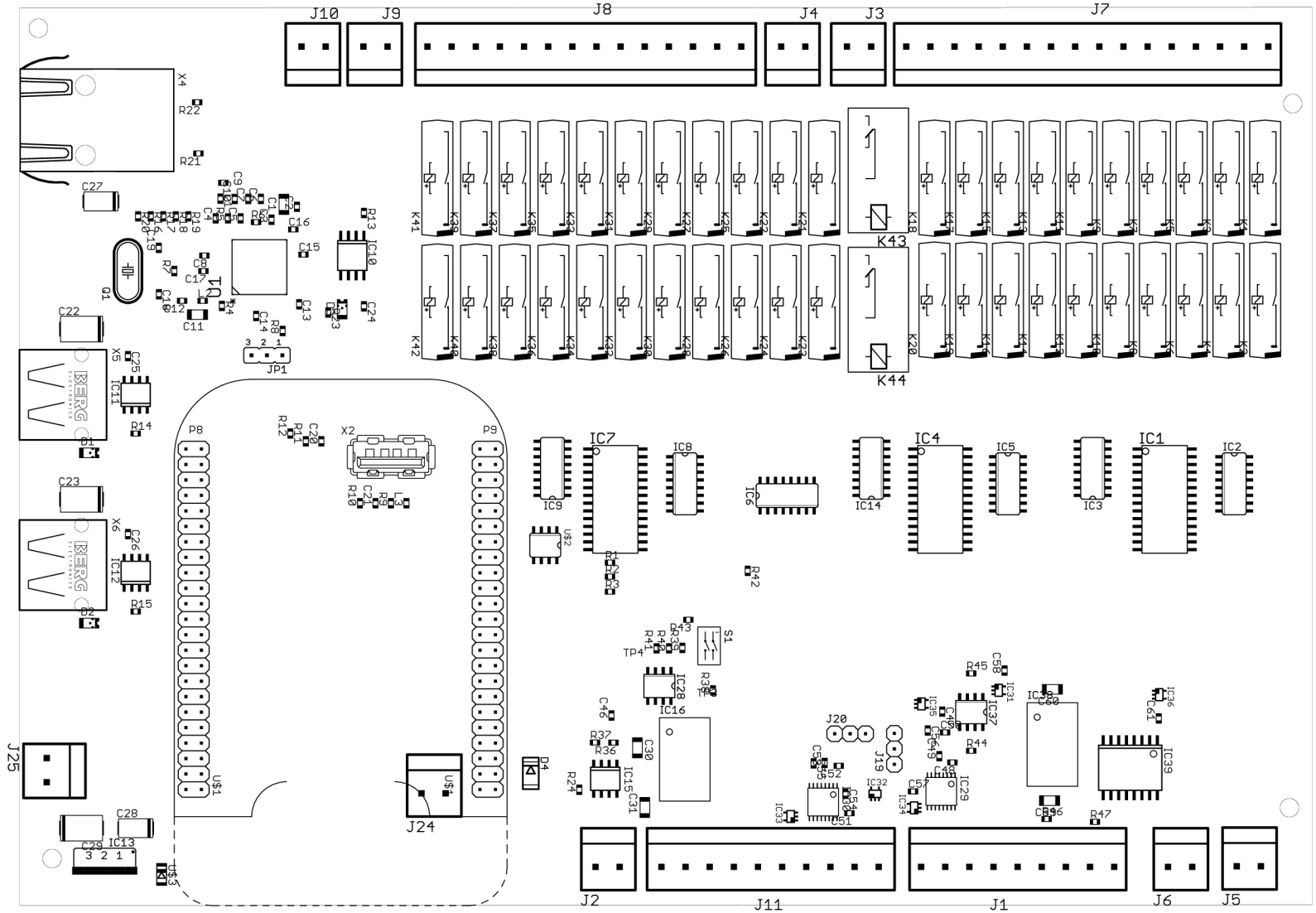
.2 Deska plošných spojů, vrchní strana



.3 Deska plošných spojů, spodní strana



.4 Osazovací plná spodní strany


J10
J9
J8
J4
J3
J7
K43
K44
IC7
IC8
IC9
IC6
IC14
IC4
IC5
IC3
IC1
IC2
P8
P9
X2
X4
X5
X6
JP1
S1
TP4
J24
J25
J2
J11
J1
J6
J5
J20
J19
D1
D2
D4
Q1
U1
U$1
U$2
U$3
U$4
U$5
U$6
U$7
U$8
U$9
U$10
U$11
U$12
U$13
U$14
U$15
U$16
U$17
U$18
U$19
U$20
U$21
U$22
U$23
U$24
U$25
U$26
U$27
U$28
U$29
U$30
U$31
U$32
U$33
U$34
U$35
U$36
U$37
U$38
U$39
U$40
U$41
U$42
U$43
U$44
U$45
U$46
U$47
U$48
U$49
U$50
U$51
U$52
U$53
U$54
U$55
U$56
U$57
U$58
U$59
U$60
U$61
U$62
U$63
U$64
U$65
U$66
U$67
U$68
U$69
U$70
U$71
U$72
U$73
U$74
U$75
U$76
U$77
U$78
U$79
U$80
U$81
U$82
U$83
U$84
U$85
U$86
U$87
U$88
U$89
U$90
U$91
U$92
U$93
U$94
U$95
U$96
U$97
U$98
U$99
U$100
U$101
U$102
U$103
U$104
U$105
U$106
U$107
U$108
U$109
U$110
U$111
U$112
U$113
U$114
U$115
U$116
U$117
U$118
U$119
U$120
U$121
U$122
U$123
U$124
U$125
U$126
U$127
U$128
U$129
U$130
U$131
U$132
U$133
U$134
U$135
U$136
U$137
U$138
U$139
U$140
U$141
U$142
U$143
U$144
U$145
U$146
U$147
U$148
U$149
U$150
U$151
U$152
U$153
U$154
U$155
U$156
U$157
U$158
U$159
U$160
U$161
U$162
U$163
U$164
U$165
U$166
U$167
U$168
U$169
U$170
U$171
U$172
U$173
U$174
U$175
U$176
U$177
U$178
U$179
U$180
U$181
U$182
U$183
U$184
U$185
U$186
U$187
U$188
U$189
U$190
U$191
U$192
U$193
U$194
U$195
U$196
U$197
U$198
U$199
U$200
U$201
U$202
U$203
U$204
U$205
U$206
U$207
U$208
U$209
U$210
U$211
U$212
U$213
U$214
U$215
U$216
U$217
U$218
U$219
U$220
U$221
U$222
U$223
U$224
U$225
U$226
U$227
U$228
U$229
U$230
U$231
U$232
U$233
U$234
U$235
U$236
U$237
U$238
U$239
U$240
U$241
U$242
U$243
U$244
U$245
U$246
U$247
U$248
U$249
U$250
U$251
U$252
U$253
U$254
U$255
U$256
U$257
U$258
U$259
U$260
U$261
U$262
U$263
U$264
U$265
U$266
U$267
U$268
U$269
U$270
U$271
U$272
U$273
U$274
U$275
U$276
U$277
U$278
U$279
U$280
U$281
U$282
U$283
U$284
U$285
U$286
U$287
U$288
U$289
U$290
U$291
U$292
U$293
U$294
U$295
U$296
U$297
U$298
U$299
U$300
U$301
U$302
U$303
U$304
U$305
U$306
U$307
U$308
U$309
U$310
U$311
U$312
U$313
U$314
U$315
U$316
U$317
U$318
U$319
U$320
U$321
U$322
U$323
U$324
U$325
U$326
U$327
U$328
U$329
U$330
U$331
U$332
U$333
U$334
U$335
U$336
U$337
U$338
U$339
U$340
U$341
U$342
U$343
U$344
U$345
U$346
U$347
U$348
U$349
U$350
U$351
U$352
U$353
U$354
U$355
U$356
U$357
U$358
U$359
U$360
U$361
U$362
U$363
U$364
U$365
U$366
U$367
U$368
U$369
U$370
U$371
U$372
U$373
U$374
U$375
U$376
U$377
U$378
U$379
U$380
U$381
U$382
U$383
U$384
U$385
U$386
U$387
U$388
U$389
U$390
U$391
U$392
U$393
U$394
U$395
U$396
U$397
U$398
U$399
U$400
U$401
U$402
U$403
U$404
U$405
U$406
U$407
U$408
U$409
U$410
U$411
U$412
U$413
U$414
U$415
U$416
U$417
U$418
U$419
U$420
U$421
U$422
U$423
U$424
U$425
U$426
U$427
U$428
U$429
U$430
U$431
U$432
U$433
U$434
U$435
U$436
U$437
U$438
U$439
U$440
U$441
U$442
U$443
U$444
U$445
U$446
U$447
U$448
U$449
U$450
U$451
U$452
U$453
U$454
U$455
U$456
U$457
U$458
U$459
U$460
U$461
U$462
U$463
U$464
U$465
U$466
U$467
U$468
U$469
U$470
U$471
U$472
U$473
U$474
U$475
U$476
U$477
U$478
U$479
U$480
U$481
U$482
U$483
U$484
U$485
U$486
U$487
U$488
U$489
U$490
U$491
U$492
U$493
U$494
U$495
U$496
U$497
U$498
U$499
U$500
U$501
U$502
U$503
U$504
U$505
U$506
U$507
U$508
U$509
U$510
U$511
U$512
U$513
U$514
U$515
U$516
U$517
U$518
U$519
U$520
U$521
U$522
U$523
U$524
U$525
U$526
U$527
U$528
U$529
U$530
U$531
U$532
U$533
U$534
U$535
U$536
U$537
U$538
U$539
U$540
U$541
U$542
U$543
U$544
U$545
U$546
U$547
U$548
U$549
U$550
U$551
U$552
U$553
U$554
U$555
U$556
U$557
U$558
U$559
U$560
U$561
U$562
U$563
U$564
U$565
U$566
U$567
U$568
U$569
U$570
U$571
U$572
U$573
U$574
U$575
U$576
U$577
U$578
U$579
U$580
U$581
U$582
U$583
U$584
U$585
U$586
U$587
U$588
U$589
U$590
U$591
U$592
U$593
U$594
U$595
U$596
U$597
U$598
U$599
U$600
U$601
U$602
U$603
U$604
U$605
U$606
U$607
U$608
U$609
U$610
U$611
U$612
U$613
U$614
U$615
U$616
U$617
U$618
U$619
U$620
U$621
U$622
U$623
U$624
U$625
U$626
U$627
U$628
U$629
U$630
U$631
U$632
U$633
U$634
U$635
U$636
U$637
U$638
U$639
U$640
U$641
U$642
U$643
U$644
U$645
U$646
U$647
U$648
U$649
U$650
U$651
U$652
U$653
U$654
U$655
U$656
U$657
U$658
U$659
U$660
U$661
U$662
U$663
U$664
U$665
U$666
U$667
U$668
U$669
U$670
U$671
U$672
U$673
U$674
U$675
U$676
U$677
U$678
U$679
U$680
U$681
U$682
U$683
U$684
U$685
U$686
U$687
U$688
U$689
U$690
U$691
U$692
U$693
U$694
U$695
U$696
U$697
U$698
U$699
U$700
U$701
U$702
U$703
U$704
U$705
U$706
U$707
U$708
U$709
U$710
U$711
U$712
U$713
U$714
U$715
U$716
U$717
U$718
U$719
U$720
U$721
U$722
U$723
U$724
U$725
U$726
U$727
U$728
U$729
U$730
U$731
U$732
U$733
U$734
U$735
U$736
U$737
U$738
U$739
U$740
U$741
U$742
U$743
U$744
U$745
U$746
U$747
U$748
U$749
U$750
U$751
U$752
U$753
U$754
U$755
U$756
U$757
U$758
U$759
U$760
U$761
U$762
U$763
U$764
U$765
U$766
U$767
U$768
U$769
U$770
U$771
U$772
U$773
U$774
U$775
U$776
U$777
U$778
U$779
U$780
U$781
U$782
U$783
U$784
U$785
U$786
U$787
U$788
U$789
U$790
U$791
U$792
U$793
U$794
U$795
U$796
U$797
U$798
U$799
U$800
U$801
U$802
U$803
U$804
U$805
U$806
U$807
U$808
U$809
U$810
U$811
U$812
U$813
U$814
U$815
U$816
U$817
U$818
U$819
U$820
U$821
U$822
U$823
U$824
U$825
U$826
U$827
U$828
U$829
U$830
U$831
U$832
U$833
U$834
U$835
U$836
U$837
U$838
U$839
U$840
U$841
U$842
U$843
U$844
U$845
U$846
U$847
U$848
U$849
U$850
U$851
U$852
U$853
U$854
U$855
U$856
U$857
U$858
U$859
U$860
U$861
U$862
U$863
U$864
U$865
U$866
U$867
U$868
U$869
U$870
U$871
U$872
U$873
U$874
U$875
U$876
U$877
U$878
U$879
U$880
U$881
U$882
U$883
U$884
U$885
U$886
U$887
U$888
U$889
U$890
U$891
U$892
U$893
U$894
U$895
U$896
U$897
U$898
U$899
U$900
U$901
U$902
U$903
U$904
U$905
U$906
U$907
U$908
U$909
U$910
U$911
U$912
U$913
U$914
U$915
U$916
U$917
U$918
U$919
U$920
U$921
U$922
U$923
U$924
U$925
U$926
U$927
U$928
U$929
U$930
U$931
U$932
U$933
U$934
U$935
U$936
U$937
U$938
U$939
U$940
U$941
U$942
U$943
U$944
U$945
U$946
U$947
U$948
U$949
U$950
U$951
U$952
U$953
U$954
U$955
U$956
U$957
U$958
U$959
U$960
U$961
U$962
U$963
U$964
U$965
U$966
U$967
U$968
U$969
U$970
U$971
U$972
U$973
U$974
U$975
U$976
U$977
U$978
U$979
U$980
U$981
U$982
U$983
U$984
U$985
U$986
U$987
U$988
U$989
U$990
U$991
U$992
U$993
U$994
U$995
U$996
U$997
U$998
U$999
U$1000

.5 Obsah přílohy na CD

- Eagle files
  - diplomova-prace.brd
  - diplomova-prace.sch
  - PDF
  - seznam součástek.csv
- Software
  - Dokumentace Doxygen
  - Overlay
  - Převzané knihovny
  - Ukázkový test
  - Vytvořené Knihovny
  - Webové rozhraní
- Utility
  - PinMux