VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH A REALIZACE VÝVOJOVÉ DESKY PRO PLATFORMU BEAGLEBOARD

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

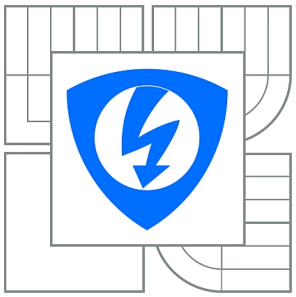
AUTOR PRÁCE
AUTHOR

Bc. ANDREJ ŤAPUCHA

BRNO 2014

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH A REALIZACE VÝVOJOVÉ DESKY PRO PLATFORMU BEAGLEBOARD

DESIGN AND REALIZATION OF A DEVELOPEMENT BOARD FOR BEAGLEBOARD

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

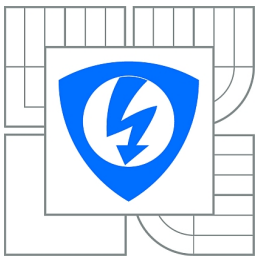
AUTOR PRÁCE — Bc. ANDREJ ŤAPUCHA

AUTHOR

VEDOUCÍ PRÁCE — Ing. ONDŘEJ KRAJSA, Ph.D.

SUPERVISOR

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Bc. Andrej Ťapucha	***ID:***	125669
Ročník:	2	***Akademický rok:***	2013/2014

NÁZEV TÉMATU:

Návrh a realizace vývojové desky pro platformu Beagleboard

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte rozšiřující desku pro vývojový kit BeagleBoard. Na této desce realizujte především AD/DA převodníky, klávesnici, display, ovládání relé a GPIO vstupy/výstupy. K jednotlivým prvkům vytvořte příklady ovládání v OS Linux v jazyku C/C++.

DOPORUČENÁ LITERATURA:

[1] RICHARDSON, By Matt. Beginning BeagleBone: creating Linux-powered electronics projects. Farnham: O'Reilly. ISBN 978-144-9345-372.

Termín zadání:	10.2.2014	***Termín odevzdání:***	30.5.2014

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

doc. Ing. Jiří Mišurec, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Náplňou diplomovej práce je navrhnúť a vytvoriť rozširujúcu dosku pre jedno doskový počítač beagleboard. Prvou a veľmi dôležitou častou bolo zistiť všetky možnosti o rozšírenie samotného počítača. Zvoliť správne rozširujúci port. Naštudovať jeho možnosti, rozloženie jednotlivých pinov. Nasledovalo vhodné zvolenie rozširujúcich obvodov ktoré by demonštrovali funkciu jednotlivých rozšírení. Bola navrhnutá testovací plošný spoj ktorý obsahuje obvody ktoré menia napäťovú úroveň z beagleboardu kvôli ochrane mini počítača aby pri testovaní jednotlivých rozširovacích periférii nedošlo k zničeniu samotného beagleboardu alebo jeho časti. K samotným perifériám sa viažu jednotlivé časti teoretického úvodu. Presnejšie ide o rozšírenie o LCD displej, zvolený displej je segmentový a obsahuje dva riadky po 16 znakov. Pri možnostiach rozšírenia o ďalšie súčasti bolo potrebné riešiť problém z nedostatkov výstupných pinov, tento problém bol vyriešený pomocou expandéru. Ten rozširuje samotný beagleboard o ďalších 16 pinov. Na týchto pinoch je pripojený LCD displej spolu z maticovou klávesnicou. Súčasťou vytvorenej dosky sú aj dva obvody teplotných čidiel, ale majú rozdielny typ komunikácie jedno pomocou I2C a druhé pomocou 1-Wire zbernice. Doska obsahuje aj výstupy rôznych ďalších zberníc napríklad UART zbernicu. Obsahuje aj výstupný pin ktorým je možné riadiť PWM moduláciu. Zvyšné piny sú univerzálne GPIO piny ktorým je možné programovo definovať funkciu pomocou operačného systému Linux. Výsledkom práce je finálna doska z balíkom ovládacích programov pomocou ktorých je možné overiť jednotlivé periféria.

KĽÚČOVÉ SLOVÁ

Beagleboard-xM, Mini počítač, Linux, Ubuntu, Rozširujúci port, PWM, SPI, I2C, LCD, UART

ABSTRACT

Contents of the thesis is to design and create an expansion board for the BeagleBoard computer one plate . The first and very important part was to find out all the possibilities for an extension of the computer itself . Choose the right expansion port . To study the possibility distribution of individual pins . Followed by the appropriate extension election districts that demonstrate the function of each enlargement . Test was designed PCB that contains circuits that change the voltage level of the BeagleBoard to protect a mini computer to the testing of each gusset periphery prevent damage to itself or part of the BeagleBoard . The peripherals themselves bind different parts of theoretical introduction. More specifically the extension of the LCD display is selected segment and includes two lines of 16 characters . When the possibility to add additional components were necessary to deal with the shortcomings of the output pins , this problem was solved by using an expander . He extends itself BeagleBoard by another 16 pins . On these pins are connected together on the LCD matrix keyboard . The creation of the board are two circuit temperature sensors , but have a different type of communication using I2C and one second using 1 -Wire bus . The board also contains various other bus outputs , for example, UART bus. It also contains an output pin which can be controlled PWM modulation . The remaining pins are versatile GPIO pins which can be programmed to define a function using the Linux operating system . The result is a final board of the package control programs which make it possible to verify the individual peripherals.

KEYWORDS

BeagleBoard-xM, Mini computer, Linux, Ubuntu, expansion port, PWM, SPI, I2C, LCD, UART

ŤAPUCHA, Andrej *Návrh a realizace vývojové desky pro platformu Beagleboard*: semestrálny projekt. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Fakulta elektrotechniky a komunikačních technologií , 2014. 65 s. Vedúci práce bol Ing. Ondřej Krajsa, Ph.D.

PREHLÁSENIE

Prehlasujem, že som svoj semestrálny projekt na tému „Návrh a realizace vývojové desky pro platformu Beagleboard“ vypracoval samostatne pod vedením vedúceho semestrálneho projektu, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedeného semestrálneho projektu ďalej prehlasujem, že v súvislosti s vytvorením tohoto semestrálneho projektu som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o právu autorském, o právoch súvisejúcich s právom autorským a o zmeně niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhé, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

Brno ....

(podpis autora)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Technicka 12, 61600 Brno
Czech Republic
http://www.six.feec.vutbr.cz

POĎAKOVANIE

Výzkum popsaný v tomto semestrálním projektu 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)

OBSAH

Úvod **12**

1 Riešenie študentskej práce **13**
1.1 BeagleBoard . . . 13
1.1.1 História . . . 13
1.1.2 Parametre . . . 13
1.1.3 Procesor DM3730 . . . 14
1.1.4 Indikačné LED . . . 15
1.1.5 Vstupné spínače . . . 15
1.1.6 Rozloženie prvkov . . . 15
1.1.7 Software . . . 17
1.2 Menič napäťovej úrovne . . . 17
1.3 Pulzná šírková modulácia . . . 19
1.4 LCD displej . . . 20
1.4.1 Komunikácia . . . 21
1.5 Maticová klávesnica . . . 22
1.6 I2C zbernica . . . 24
1.6.1 Komunikácia I2C . . . 25
1.6.2 LM 92 - teplotné čidlo . . . 26
1.6.3 Teplotné čidlo DS18S20 . . . 27
1.6.4 Komunikácia s čidlom . . . 29
1.7 SPI zbernica . . . 30
1.7.1 Popis pinov . . . 30
1.7.2 Zahájenie komunikácie . . . 31
1.7.3 Prenos dát . . . 31
1.8 AD/DA prevodník . . . 32
1.8.1 Komunikácia z obvodom . . . 32
1.9 Budič linky . . . 33
1.10 Výkonové spínače . . . 33
1.11 Expandér . . . 35

2 Výsledky študentskej práce **36**
2.1 Inštalácia systému . . . 36
2.2 Nastavenie statickej IP adresy . . . 37
2.3 Nastavenie rozširujúceho portu . . . 38
2.4 I2C rozhranie . . . 40
2.4.1 Komunikácia cez I2C v jazyku C . . . 43

2.5 Rozhranie 1-WIRE 43
2.6 Ovládanie GPIO pinov 44
2.6.1 LED kontrolka 44
2.6.2 Snímanie hodnoty zo spínača 44
2.7 Generovanie PWM modulácie 45
2.7.1 PIN MUX pre PWM 46
2.7.2 Aktivácia a registre 46
2.8 SPI rozhranie 48
2.9 Testovacie dosky 50
2.9.1 Doska z LED diódami 50
2.9.2 Doska z teplotným senzorom LM92 51
2.9.3 Doska z prevodníkom napäťovej úrovne 51
2.10 Návrh a výroba dosky 51

3 Záver **52**

Literatúra **53**

Zoznam symbolov, veličín a skratiek **55**

Zoznam príloh **56**

A Tabuľka parametrov **57**

B Testovacia dosky **58**

C Finálna dosky **61**

D Tabuľka rozširujúceho konektora **65**

ZOZNAM OBRÁZKOV

1.1 Systém Beagleboard xM Rev.C. . . . 14
1.2 Umiestnenie indikačných LED . . . 15
1.3 Umiestnenie ovládacích tlačidiel . . . 16
1.4 Umiestnenie integrovaných obvodov . . . 17
1.5 Menič napäťovej úrovne, možnosť 1. . . . 18
1.6 Jedno hradlo obvodu TXB0108. . . . 19
1.7 Obvod PCA9306. . . . 19
1.8 PWM modulácia. . . . 20
1.9 Zapojenie LCD displeja. . . . 22
1.10 Zapojenie klávesnice. . . . 23
1.11 Rozloženie kláves klávesnice. . . . 23
1.12 Diagram klávesnice. . . . 24
1.13 HW architekúra I2C. . . . 25
1.14 Komunikácia štart a stop . . . 25
1.15 Posielanie adresy . . . 26
1.16 Zapojenie obvodu LM92. . . . 27
1.17 Púzdra vyhotovenia DS18B20 . . . 28
1.18 SPI zapojenie . . . 30
1.19 Zbernica rozhrania RS - 485 . . . 34
1.20 Zapojenie s tranzistormi . . . 34
1.21 Zapojenie s tranzistorom a relé . . . 35
1.22 Vnútorná štruktúra MCP23017 . . . 35
2.1 I2C v rozširovacom konektore. . . . 41
2.2 Led dióda pripojená na GPIO 139. . . . 45
2.3 Spínač zapojený na GPIO 139. . . . 46
2.4 PWM a registre. . . . 47
B.1 Testovacia doska z diódami. . . . 58
B.2 Testovacia doska z diódami osadená. . . . 58
B.3 Testovacia doska z LM92. . . . 59
B.4 Testovacia doska z LM92 osadená. . . . 59
B.5 Testovacia doska univerzálna. . . . 60
B.6 Testovacia doska univerzálna osadená. . . . 60
C.1 Schéma finálne dosky. . . . 61
C.2 Plošný spoj zo spodnej strany. . . . 62
C.3 Plošný spoj z vrchnej strany. . . . 62
C.4 Rozvrhnutie súčiastok zo spodnej strany. . . . 63
C.5 Rozvrhnutie súčiastok z vrchnej strany. . . . 63

C.6 Zoznam použitých súčiastok. 64

ZOZNAM TABULIEK

1.1 Zapojenie pinov na LCD displeji 21
1.2 Rozloženie znakov na displeji 2x16 znakov 22
1.3 Popis jednotlivých pinov obvodu. 27
1.4 Jednotlivé funkcie. 28
1.5 Hlavný register. 28
1.6 LSB bajt 0 29
1.7 MSB bajt 1 29
1.8 Povolené rozsahy obvodu 32
2.1 Nastavenie módov rozširovacieho portu 40
2.2 Adresy v pamäti pre jednotlivé časovače. 47
2.3 Tabuľka registrov. 47
2.4 Tabuľka pinou rozhrania SPI. 49
A.1 Prehľadná tabuľka parametrov 57
D.1 Tabuľka módou rozširujúceho konektora 65

ÚVOD

Práca popisuje návrh a následnú realizáciu vývojovej dosky pre minipočítač Beagleboard. V úvode práce je rozobratá presná špecifikácia minipočítača. Bolo potrebné naštudovať aké periféria samotný počítač obsahuje a následne správne zvoliť rozširovacie moduly. Výber vhodných pinov v rozširovacom konektore bol taktiež dôležitý pretože samotné piny majú rôzne režimy kde v každom režime môžu spĺňať rozličnú funkciu. V teoretickom úvode sú popísané možnosti ako rozšíriť vývojovú dosku, už samotný minipočítač obsahuje periféria ako je USB, ethernet, audio výstup a vstup, video výstup, RS 232, preto ide skorej o špeciálne periférie. Presnejšie o modul z LCD displejom, modul z PWM moduláciou, modul z maticovou klávesnicou, moduly pre rozhrania I2C, SPI a UART. Po teoretickom naštudovaní vyššie uvedených rozšírení boli vyrobené prototypové dosky na ktorých bola odskúšaná funkčnosť. Boli vyrobené 4 rôzne dosky. Následne bola vytvorená finálna doska. Tá bola navrhnutá za účelom zlúčenia všetkých periférií na jednu dosku. Doska obsahuje obvody ktorými je možné vyskúšať funkčnosť jednotlivých periférií, obsahuje taktiež konektory aby bolo možné všetky periféria vyviesť na nepájivé pole, kde je možné pracovať zo zapojením bez potreby prerábať vývojovú dosku.

1 RIEŠENIE ŠTUDENTSKEJ PRÁCE

1.1 BeagleBoard

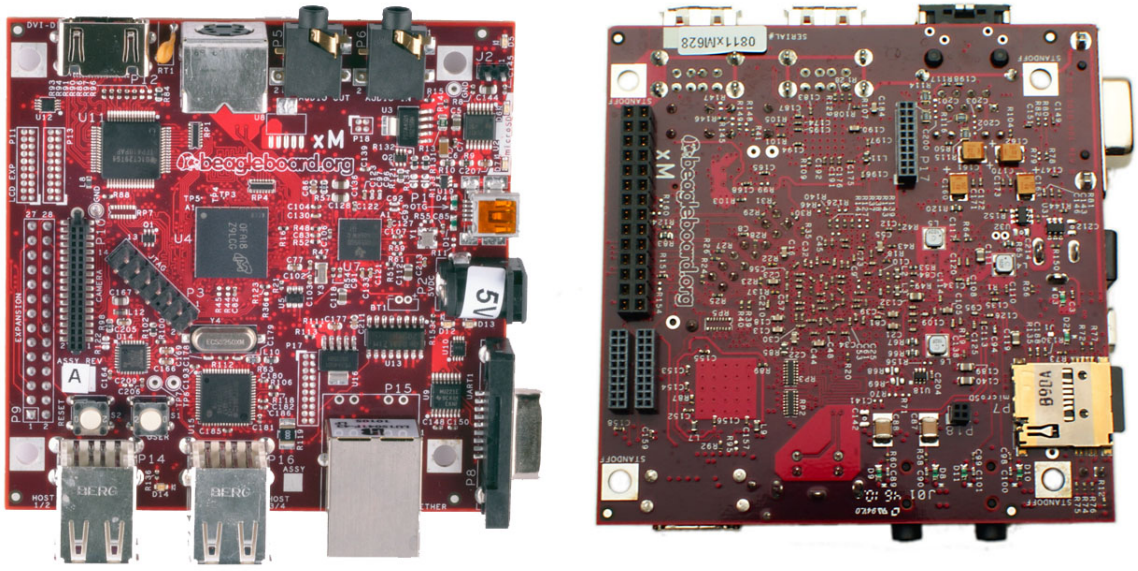
V práci bol využitý jednodoskový mikropočítač z rady Beagleboar z označením xM. Tento model je z roku 2010. Tohoročný model má označenie BeagleBone Black. Zostrojený bol firmou Texas Instrument a firmou Digi-Key. Navrhnutý je na platforme open-source využívajúci procesor OMAP 3535. Doska je vytvorená konštruktérmi, ktorí dbali na jednoduchosť a použiteľnosť, preto obsahuje mnoho portou a periférií. Vhodný software ovládajúci celé zariadenie je taktiež open-source a voľne dostupný na internete. Celé zariadenie je aj cenovo dostupné a bolo vytvorené prioritne na vzdelávacie a študijné účely a to, rozvíjať schopnosti študentov.

1.1.1 História

Prvým vyrobeným zariadením bol BeagleBoard. Bol uvedený na trh 28 .júla 2008. O rok neskôr nasledovala verzia BeagleBoard rev.C uvedená na trh 13 .mája 2009, túto verziu vidíme na obr. 1.1. Nasledoval model z označením BeagleBoard-xM, ktorý bol uvedený na trh 14 .septembra 2010. Toto zariadenie budem využívať v diplomovej práci. Nasledoval model BeagleBone uvedený na trh 31 .októbra 2011. Po úspešnom uvedení týchto zariadení na trh vývoj nezaháľal a uviedol najnovší model z označením BeagleBone Black. Všetky tieto zariadenia sú cenovo dostupné, preto mali veľký úspech medzi užívateľmi. Parametre jednotlivých modelov sú zrovnané v tab. A.1. [7]

1.1.2 Parametre

- procesor DM3730 taktovaný na 1 GHz,
- operačná pamäť 512 Mbit na frekvencií 166 MHz,
- pamäť operačného systému na micro-SD karte,
- sieťový konektor RJ 45 na pripojenie na sieť,
- 4 x USB port na pripojenie periférií,
- USB-OTG,
- rozširujúci port GPIO,
- port JTAG,
- port sériovej linky na konektore D-Sub typu DE-9 F,
- porty grafických zobrazovačov : DVI-D, S-Video, LCD port,
- audio porty Jack 3,5 mm výstup a vstup,
- napájací konektor +5 V.



Obr. 1.1: Systém Beagleboard xM Rev.C.

1.1.3 Procesor DM3730

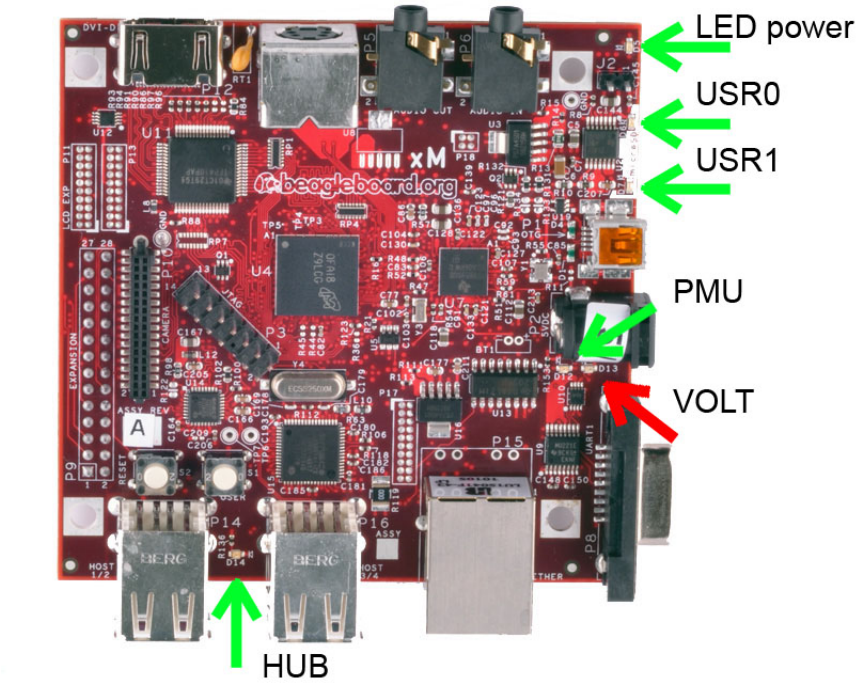
Tento procesor pochádza z rodiny ARM procesorov. Kompatibilný z treťou generáciou OMAP. Niektoré z jeho špecifikácií si rozoberieme. [3] [5]

- ARM architektúra, jadro Cortex-8, frekvencia max 1 MHz, ale podporuje aj 300, 600 a 800 MHz,
- vysokorýchlostný akcelerátor pre audio, video a obrázky,
- C64x L1/L2 pamäť,
- externé rozhranie pre pamäť,
- nízke napájacie napätie od 0,9 V do 3,0 V,
- Sériová zbernica 5 kanálová,
- podpora USB-OTG,
- UART 4x,
- I2C zbernice 3x,
- signálový procesor pre spracovanie obrazu,
- 32 bit watchdog časovač,
- 32 bit 32 kHz synchrónny časovač,
- 188 GPIO
- výroba technológiou 45 nm,
- 32 Kbit ROM,
- 64 Kbit zdielanej SRAM,
- testovacie rozhranie JTAG.

1.1.4 Indikačné LED

Na začiatok si popíšeme indikačné LED, ktoré náš BeagleBoard-xM obsahuje. Tieto LED nám indikujú rôzne stavové ukazovatele a rôzne kritické situácie, ich umiestnenie vidíme na obr. C.3.

- **LED power** indikuje pripojenie napájania,
- **USR0 a USR1** detekuje prítomnosť operačného programu,
- **VOLT** bliká pri prekročení napájacieho napätia,
- **PMU** detekuje údržbu zdroja,
- **HUB** indikuje pripojenie USB zariadenia.



Obr. 1.2: Umiestnenie indikačných LED

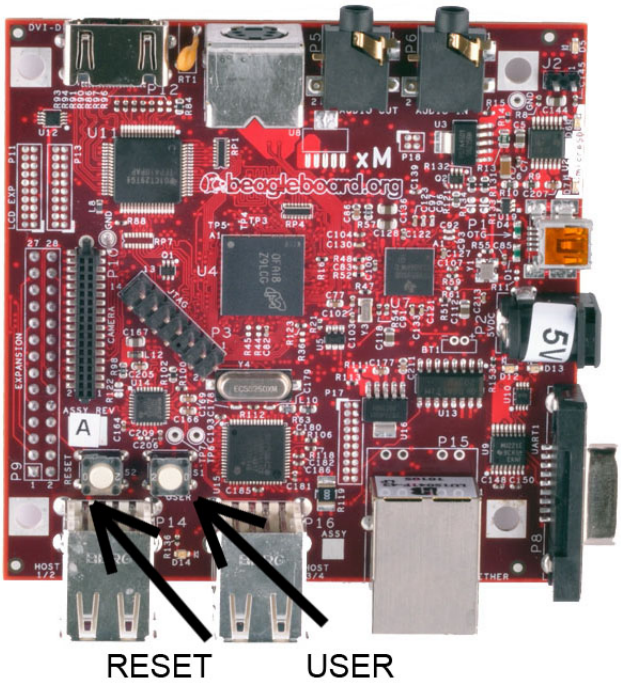
1.1.5 Vstupné spínače

Zariadenie obsahuje dva vstupné spínače, umiestnené na doske vidíme na obr. 1.3.

- **Tlačítko A** inak sa dá nazvať aj RESET, slúži na reštartovanie systému,
- **Tlačítko B** pri stlačení tohoto tlačítka pri štarte systému, systém nenabehne.

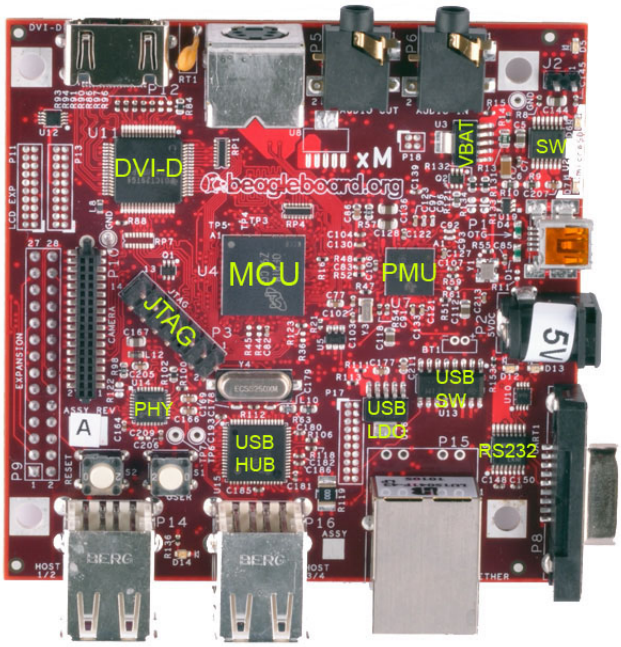
1.1.6 Rozloženie prvkov

Na obr. 1.4. nižšie je vyobrazená doska celého počítača z vrchnej strany. Na nej sú zobrazené hlavné integrované obvody. Ich význam a popis je uvedený nižšie.



Obr. 1.3: Umiestnenie ovládacích tlačidiel

- **MCU** procesor architektúry ARM, postavený na jadre Cortex A8 na frekvencií 1 GHz,
- **RAM** operačná pamäť o veľkosti 512 MB, typ pamäti je MDDR čo je synchrónna dynamická operačná pamäť,
- **USB HUB, USB SW, USB LDO** integrovaný USB rozbočovač, obvod je jeho riadiacim prvkom,
- **RS 232** obvod obsluhujúci sériové rozhranie RS 232,
- **PMU** je ovládač dvoch indikačných LED, obsahuje PWM moduláciu, dokáže riadiť jas týchto LED, z procesorom komunikuje cez I2C zbernicu,
- **DVI-D** obvod slúžiaci na prevod obrazového signálu na HDMI a DVI-D rozhranie, možnosť pripojiť obrazovku,
- **PHY** obvod združuje vstupy z USB a ethernetu a prevádza ich na signáli pre procesor,
- **VBAT** regulátor a stabilizátor vstupného napätia obsahuje aj tepelnú a prúdovú ochranu.



Obr. 1.4: Umiestnenie integrovaných obvodov

1.1.7 Software

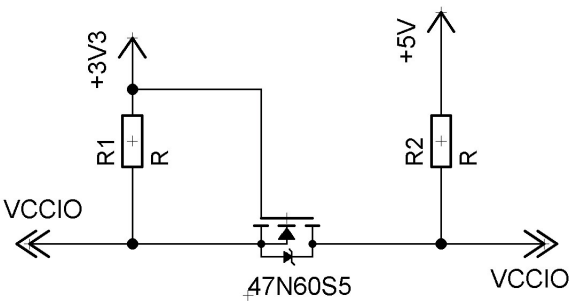
Platforma Unix - Angstrom Linux, Fedora, Ubuntu, Gentoo, Arch Linux ARM, Maemo Linux a FreeBSD,
Platforma Android - Android 4.1.2,
Platforma Windows - Windows CE,

1.2 Menič napäťovej úrovne

Takýto menič napäťovej úrovne je potrebné použiť na každý vstup alebo výstup, pretože logika Beaglebordu-xM je len 1,8 V. Väčšina použitých integrovaných obvodov pracuje na vyššej napäťovej úrovni a to 5 V. Zámena týchto úrovní by mohla zapríčiniť zničenie Beaglebordu a jeho GPIO pinou. Tomuto chceme zabrániť, preto uvediem tri možné riešenia tohoto problému.

1. Možnosť: využijeme základné súčiastky a to dva odpory a jeden tranzistor, typu Mos-fet z kanálom N. Zapojenie je vidieť na obr. 1.5.

Takýto jednoduchý obvod pracuje spoľahlivo a rýchlo až do 400 kbit/s. Princíp: obe zariadenia sú v kľude, žiadne z nich nerobí log 0: pull = up rezistory zabezpe-



Obr. 1.5: Menič napäťovej úrovne, možnosť 1.

čujú stav log 1 na oboch stranách G a S tranzistora je na rovnakej úrovni 3,3 V, čím tranzistor nepracuje, teda sú obe strany zbernice oddelené. Ak 3,3 V zariadenie vygeneruje log 0, tranzistor sa otvorí, čím k zariadeniu pripojí 5V pull-up, ktorý je týmto pripojený cez otvorený kolektor 3 V zariadenia ku GND.Tým pádom sa na vstupe 5 V zariadenie objaví log 0. V opačnom smere ak 5 V zariadenie vygeneruje log 0, je využitá vnútorná dióda v mosfet tranzistore, pomocou ktorej sa "stiahne"pull=up rezistor 3 V strany ku GND, čím sa na vstupe 3,3 V zariadenia objaví log 0. Toto zapojenie chráni 3,3 V stranu pred napäťovými špičkami z 5 V strany. [1] [2] [18]

2. Možnosť:

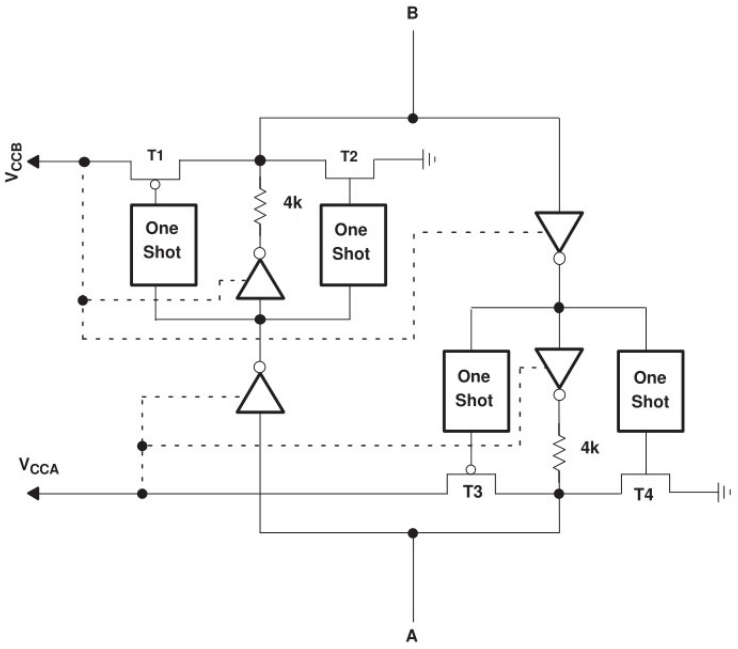
využiť komerčne vyrábané obvody napr. GTL2003 alebo TXB0108. Tieto obvody majú vo vnútri podobnú architektúru ako vidíme na obr. 1.6. Tým, že hustota súčiastok je veľká a veľkosť súčiastok je minimálna, je možné 8 portov zmestiť do malého púzdra, dokonca aj SMD. Preto je vhodné využiť takéhoto obvodu.

3. Možnosť: PCA9306

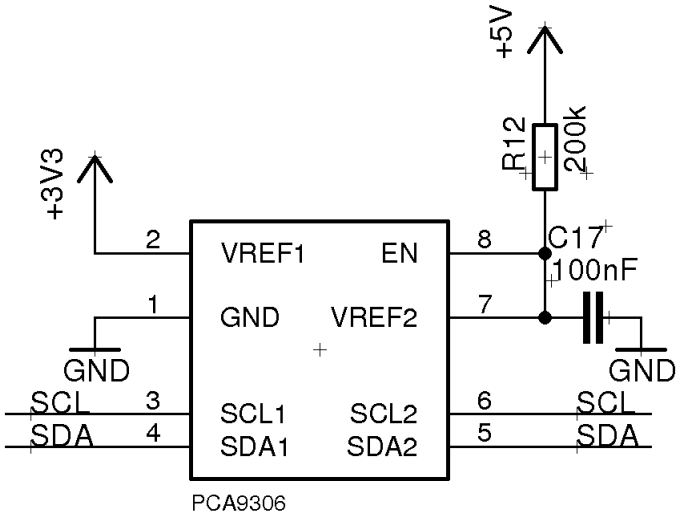
Obvod umožňuje prevod I2C zbernice z jednej napäťovej úrovne na druhú. Keďže v práci využívame I2C zbernicu z beagleboardu, ktorého napäťová úroveň je len 1,8 V a zariadenia, s ktorými chceme komunikovať sú na napäťovej úrovni 5 V, je potrebné využiť integrovaný obvod PCA9306. Jeho parametre sú uvedené nižšie v tabuľke.

- vstupné napätie v rozsahu od 1,2 V do 3,3 V,
- výstupné napätie v rozsahu od 1,8 V do 5,5 V,
- oneskorenie nepresahujúce 1,5 ns,
- otvorený kolektor na portoch SCL1, SDA1, SCL2, SDA2,
- vysoká vstupná impedancia,
- prevádzková frekvencia až 400 Khz,
- možnosť zakázať vysielanie pomocou pinu EN.

Schému integrovaného obvodu vidíme na obr. 1.7. Použil som štandardné zapojenie uvedené v datasheete tohoto obvodu.



Obr. 1.6: Jedno hradlo obvodu TXB0108.



Obr. 1.7: Obvod PCA9306.

1.3 Pulzná šírková modulácia

Skratka tejto modulácie je aj PWM alebo PSM. Základom tejto modulácie je zmena striedy (šírky impulzu). Touto moduláciou je možné efektívne riadiť dodávaný elektrický výkon. Účinnosť tejto regulácie je v tom, že ovládací tranzistor môže nado-

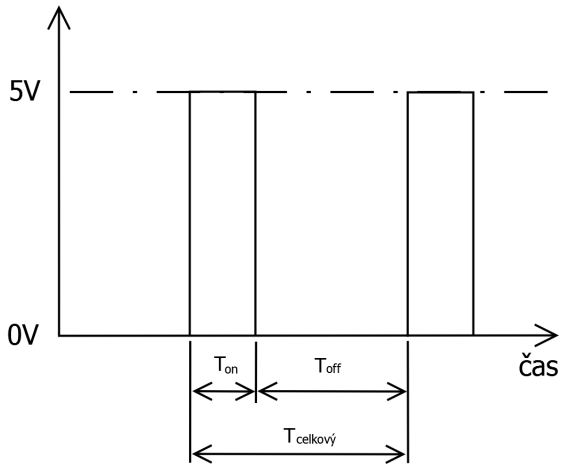
búdať iba dva stavy a to 0 a 1, preto na ňom nevznikajú tepelné straty. Tento typ ovládania nemá len pozitíva ale aj negatíva, jedným z nich je vznikajúce rušenie spôsobené vysokou rýchlosťou zapínania a vypínania výkonového obvodu. Výstupný výkon sa dá jednoducho vypočítať pomocou vzorca 1.3 .Je to vlastne pomer dĺžky Log 1 a dĺžky Log O. [1] [2]

$$T_{celkový} = T_{log\,1} + T_{log\,0}, \tag{1.1}$$

$$D = \frac{T_{log\,1}}{T_{celkový}} = \frac{T_{log\,1}}{T_{log\,1} + T_{log\,0}}, \tag{1.2}$$

$$V_{výstup} = D * V_{vstupné} = \frac{T_{log\,1}}{T_{celkový}} * V_{vstupné}. \tag{1.3}$$

- $T_{celkový}$ - celkový čas trvania periódy ,
- $T_{log\,1}$ - čas impulzu ON ,
- $T_{log\,0}$ - čas impulzu OFF ,
- D - strieda alebo duty cycle ,vyjadrené v percentách ,
- $V_{vstupné}$ - vstupné napätie ,
- $V_{výstup}$ - výstupné napätie.



Obr. 1.8: PWM modulácia.

1.4 LCD displej

LCD skratka znamená Liquid crystal display, takzvaný displej z tekutými kryštálmi. Tieto displeje sa začali objavovať v 80tych rokoch. V dnešnej dobe sú už všade okolo nás. Takýto displej obsahuje zobrazovač z tekutými kryštálmi a plošný spoj, na ktorom je umiestnený budič k zobrazovaču. Typický budič má označenie HD 44780.

Tieto displeje sa vyrábajú v rôznych veľkostiach a to : **8x2, 16x1, 16x2, 16x4, 20x2, 24x2, 40x2, 40x4**, prvé číslo znamená počet znakov v riadku a druhé číslo znamená počet riadkov. Vyrábajú sa aj displeje grafické, kde sa neuvádza počet znakov ale rozlíšenie v počte pixelov alebo obrazových bodov. Takýto displej sa označuje ako grafický LCD. Displeje môžu obsahovať aj podsvietenie. Zapojenie jednotlivých pinov je v tab. 1.1.

Tab. 1.1: Zapojenie pinov na LCD displeji

PIN	Názov	Popis
1	V_{ss}	napájanie zem
2	V_{dd}	napájanie +5 V
3	V_c	kontrast displeja rozsah 0 V až +5 V
4	RS	výber registra 0 = inštrukcie , 1 = dáta
5	R/W	výber čítanie = 1 , zápis = 0
6	E	povolenie
7	DB0	dáta zbernica 0
8	DB1	dáta zbernica 1
9	DB2	dáta zbernica 2
10	DB3	dáta zbernica 3
11	DB4	dáta zbernica 4
12	DB5	dáta zbernica 5
13	DB6	dáta zbernica 6
14	DB7	dáta zbernica 7
15	A	podsvietenie - anóda LED
16	K	podsvietenie - katóda LED

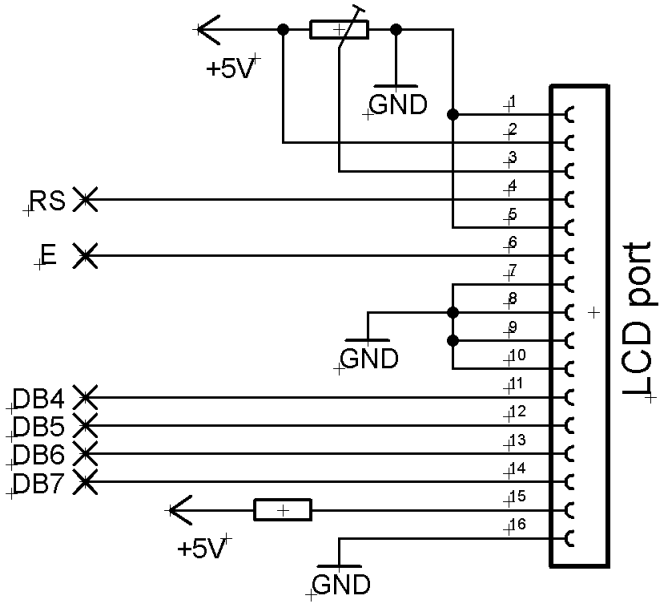
1.4.1 Komunikácia

Komunikovať z radičom LCD je možné dvoma spôsobmi a to po 8 bitovej zbernici, kde využijeme DB0 až DB7 alebo 4 bitovej zbernici a to využijeme DB4 až DB7. Rozdiel je v tom, že v prípade 4 bitovej komunikácií je o polovicu pomalšie, pretože dáta musia byť posielané dva krát po 4 bity a piny DB0 až DB3 musia byť pripojené na zem. Pri displeji, ktorý zobrazuje znaky sa znak vykresľuje do matice 5x8 bodov. Ďalej budeme uvažovať že, máme zvolený displej o dvoch riadkoch a 16 znakoch. V tab. 1.2 je zobrazená adresa jednotlivých znakov na displeji. [15]

Samotné zapojenie displeja vidíme na obr. 1.9.

Tab. 1.2: Rozloženie znakov na displeji 2x16 znakov

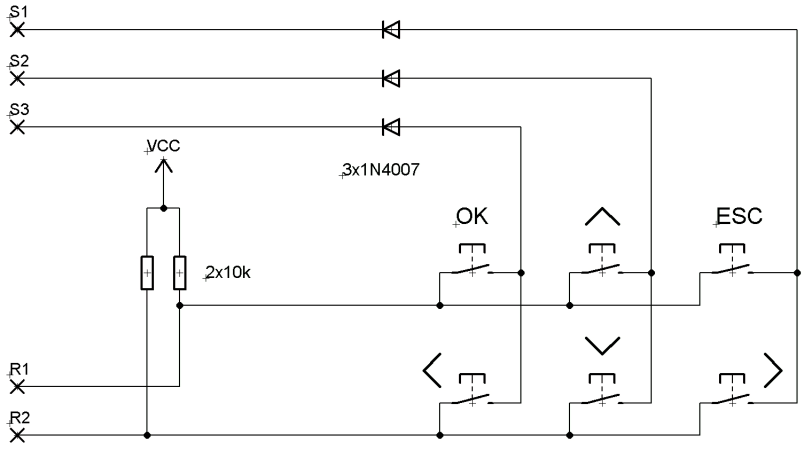
LCD pozícia	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
riadok 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
riadok 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F



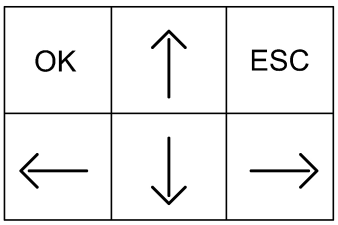
Obr. 1.9: Zapojenie LCD displeja.

1.5 Maticová klávesnica

Maticový typ klávesnice sa obvykle používa v tom prípade, že potrebujeme snímať viacero spínačov. Obvyklá komerčne predávaná klávesnica má rozloženie 4x4 kláves. V prípade, že by sme zapojili túto klávesnicu tak, že každý zo spínačov by bol pripojený k jednému vstupu, využili by sme 16 vstupov, čo je príliš veľa. Preto bol vymyslený tento princíp maticového zapojenia, kde na 16 kláves je potrebné iba 8 vstupov. Čo je polovičná úspora. Pre naše riešenie ale klávesnicu 4x4 nepotrebujeme, preto bola zvolená menšia alternatíva a to 3x2, aj pri tejto menšej klávesnici by sme využili bez maticového zapojenia 6 vstupov, v prípade maticového zapojenia je to len 5 vstupov. Keďže beagleboard má málo vstupno/výstupných pinov, ušetrenie čo len jedného je výhodné. Zapojenie klávesnice vidíme na obr. 1.10. Nami zvolené rozloženie kláves je na obr. 1.11. [16]

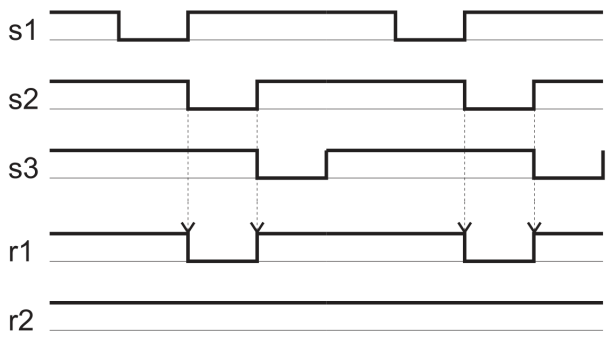


Obr. 1.10: Zapojenie klávesnice.



Obr. 1.11: Rozloženie kláves klávesnice.

Ako takáto klávesnica funguje si popíšeme v nasledujúcej časti. Všetky riadky, v našom prípade dva, sú cez rezistor pripojené k napájaciemu napätiu, čo na nich definuje úroveň $\log 1$. Na všetkých stĺpcoch prebieha postupne prepnutie do $\log 0$ a to tak, že ostatné stĺpce sú v $\log 1$, vidíme na obr. 1.12. Toto prepínanie prebieha rýchlo, nie je definovaná rýchlosť, ale je vhodné toto prepínanie nadstaviť aspoň na 10 Hz, aby stlačenie klávesnice bolo zosnímané. Pri stlačení klávesy, kde na stĺpci je aktuálne úroveň $\log 0$ sa táto úroveň prepojí na daný riadok, tým aj tento riadok prejde do stavu $\log 0$. Vyhodnotenie je následne jednoduché, pretože vieme na ktorom stĺpci bola $\log 0$ a vieme aj na ktorý riadok nám $\log 0$ prišla na vstup. Ukážku logiky pri snímaní spínačov vidíme na obr. 1.12. V prípade zobrazenom na obrázku vidíme, že bolo stlačené tlačítko v stĺpci dva a riadku jedna.



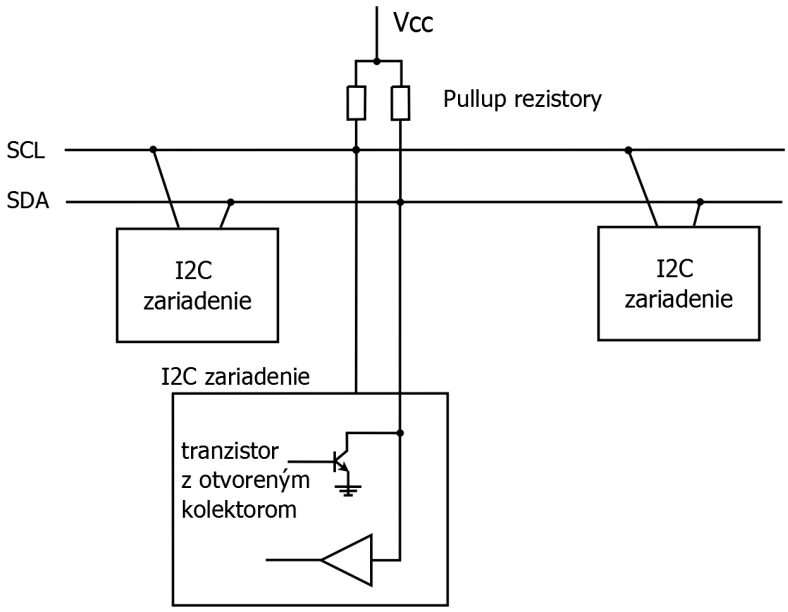
Obr. 1.12: Diagram klávesnice.

1.6 I2C zbernica

Ide o komunikačnú zbernicu a protokol vyvinutý firmou Phillips. Využitie bolo v TV prijímačoch, kde integrované obvody spolu komunikovali pri použití malého množstva pinov. Tento protokol nemá definované špeciálne konektory a káble. Samotný protokol je založený na jednoduchých hardvérových štandardoch. Protokol I2C sa rozšíril a v dnešnej dobre ho využívajú mnohé obvody, napr. pamäte EEPROM, senzory teploty, expandéry, obvody reálneho času, RF tunery, video a audio enkódery, audio procesory. [1] [2] Táto zbernica môže pracovať v rýchlostiach :

- **Slow** - pomalá -pod 100 Kbps,
- **Fast** - rýchla - 400 Kbps,
- **High speed** -super rýchla - 3,4 M,bps.

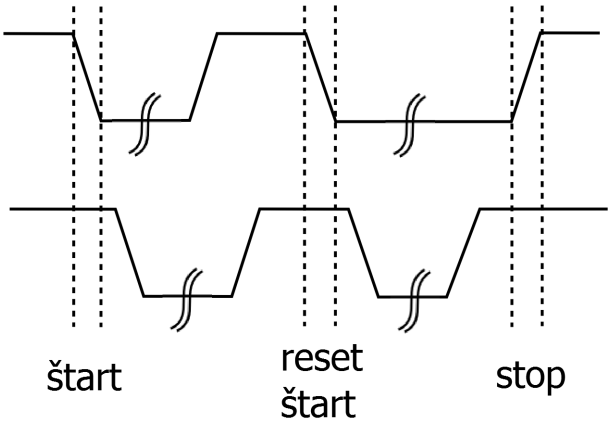
Protokol je limitovaný dĺžkou zbernice na 4 metre, kvôli udržaniu komunikačnej rýchlosti. Kapacita vedenia je približne 400 pF. Samotná komunikácia používa dva vodiče **SDA** - serial data a **SCL** - serial clock. Zbernica je väčšinou polovičný duplex, čo znamená, že v jednej chvíli môže iba jedno zariadenie odosielať alebo prijímať dáta. Protokol neobsahuje výberový vodič, preto zariadenia na zbernici musia byť rozlišované pomocou adries. Táto adresa je buď daná výrobcom, alebo je možné ju definovať dodatočne buď programovo alebo hardwardovo. Adresa musí byť jedinečná, aby nenastávali kolízie na zbernici. Obe linky SDA a SCL sú pripojené k napájaciemu napätiu pomocou pullup tranzistoru, ustálený stav na log 1. Každé zariadenie môže tento stav meniť na log 0 pomocou tranzistorov s otvoreným kolektorom, vidíme na obr. 1.13.



Obr. 1.13: HW architekúra I2C.

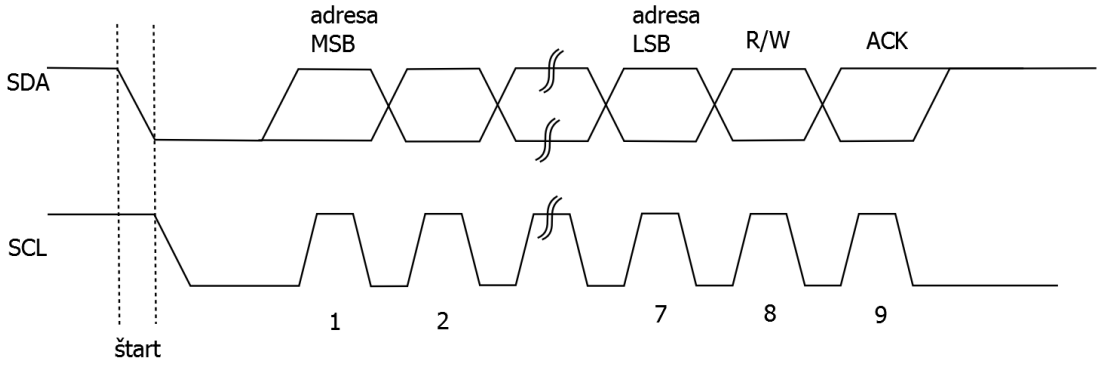
1.6.1 Komunikácia I2C

Oba vodiče sú v stave log 1, čo značí neaktívny prenos. Inicializácia komunikácie začína štartovacou podmienkou. Táto podmienka sa realizuje stavom, kedy je SCL v log 1 a dátový vodič SDA sa preklopí do log 0 za istý krátky čas, vidíme na obr. 1.14.



Obr. 1.14: Komunikácia štart a stop

Po inicializácii vysielania nastane vyslanie 7 bitového čísla, čo je adresa SLAVE zariadenia s ktorým chceme komunikovať. K tejto adrese je potrebné pridať najnižší bit, ktorý značí či budeme dáta do zariadenia zapisovať alebo chceme z neho čítať. Zapisovanie je definované log 0 a čítanie log 1 viz obr. 1.15.



Obr. 1.15: Posielanie adresy

Za jeden časový impulz môže byť odoslaný iba jeden bit. Zmena vysielaného bitu môže nastať iba vo chvíli, kedy je SCL v stave $\log 0$, inak by sa mohlo stať, že tento prechod bude vyhodnotený ako štart alebo stop komunikácie. Pre správnu funkciu sa toto nesmie na linke stať. Po každom odoslaní 8 bitov je potrebné toto odoslanie potvrdiť ACK bitom. Slave zariadenie tento ACK bit vygeneruje a odošle. Celá komunikácia končí vyslaním stop bitu. [12]

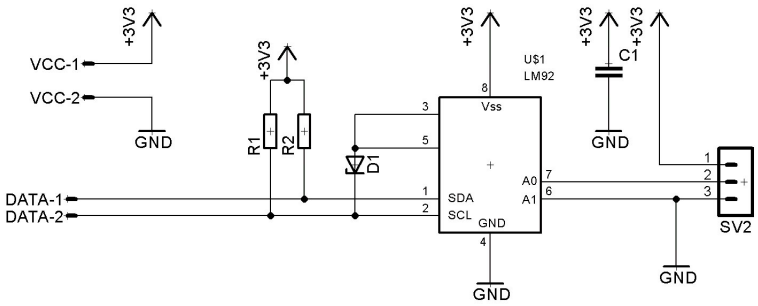
1.6.2 LM 92 - teplotné čidlo

Ako demonštráciu fungovania rozhrania I2C bol zvolený obvod LM92. Ide o teplotný senzor komunikujúci na rozhraní I2C. Zapojenie obvodu viz obr. 1.16. Špecifikácie obvodu:

- napájacie napätie od $2,7\,V$ po $6,5\,V$,
- nízky napájací prúd, typicky $350\mu\,A$ v režime spánku $5\mu\,A$,
- rozsah merania od $-65^{\check{r}}\,C do + 125^{\check{r}}\,C$,
- 12 Bitový prevodník,
- lineárna presnosť $\pm 0,5^{\check{r}}\,C$,
- rozlíšenie $0,0625^{\check{r}}\,C$,
- možnosť nadstavenia adresy každého senzoru,
- jednoduchý dizajn,

Je možné na jednu zbernicu zapojiť až 4 senzory LM92. Adresu je možné nedefinovať pomocou pinou A0 a A1. Tieto dva piny určujú posledné dva bity v adresovaní. Senzory môžu nadobúdať hodnotu:

- 0x48 hexadecimálne: 01001000 binárne,
- 0x49 hexadecimálne: 01001001 binárne,
- 0x4A hexadecimálne: 01001010 binárne,
- 0x4B hexadecimálne: 01001011 binárne,



Obr. 1.16: Zapojenie obvodu LM92.

Tab. 1.3: Popis jednotlivých pinov obvodu.

Názov	Pin	Funkcia
SDA	1	dátová linka zbernice I2C
SCL	2	synchronizačná linka I2C
T CRIT A	3	teplotný alarm
GND	4	napájanie zem
INT	5	prerušenie zo senzor
A1	6	definovanie adresy
A0	7	definovanie adresy
Vss	8	napájacie napätie kladné

Formát dát odosielaných zo senzoru je 13 Bitová binárna postupnosť. Prvý Bit určuje znamienko teplotného prevodu. V prípade log 1 je teplota záporná a pri log 0 je teploty kladná. Zvyšných 12 Bitov je hodnota načítanej teploty. Hodnota sa prevedie do decimálneho tvaru. Výsledná hodnota sa podelí číslom 16 a výsledok je teplota v stupňoch celzia. Príklad:

```
0 1000 0010 0000 b => 2080 /16 = 130°C
```

Senzor má niekoľko registrov. Tieto registre mu určujú jeho fungovanie aj konfiguráciu uvedené v tab. 1.4. Bunky P7 až P3 musia byť nulové a posledné tri bunky P2, P1, P0 vyberajú funkciu, presné funkcie sú uvedené v tab. 1.5. [4]

1.6.3 Teplotné čidlo DS18S20

Toto teplotné čidlo je jedno z najpopulárnejších pre svoju jednoduchú konštrukciu, presnosť a zapojenie iba jedným vodičom. Využíva zbernicu One - Wire. Vyrába ho

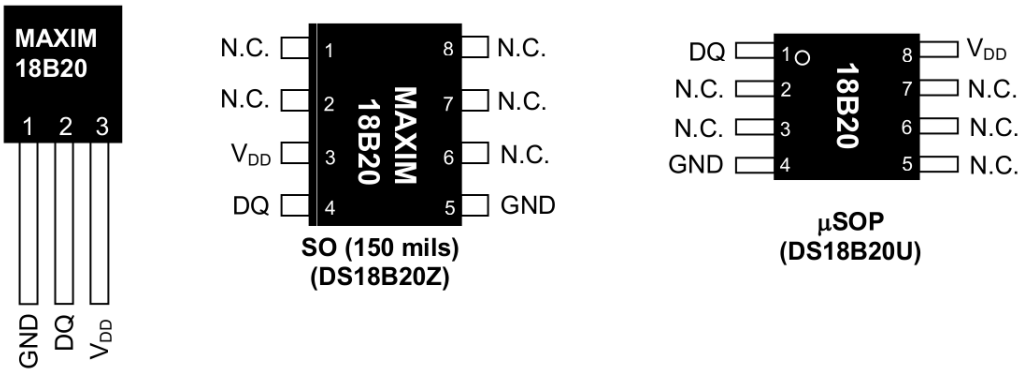
Tab. 1.4: Jednotlivé funkcie.

P7	P6	P5	P4	P3	P2	P1	D0
0	0	0	0	0	Výber funkcie		

Tab. 1.5: Hlavný register.

P2	P1	P0	Funkcia
0	0	0	vyčítanie teploty, štandardne zvolené
0	0	1	konfiguračný mód
0	1	0	T_{HYST}
0	1	1	T_{CRIT}
1	0	0	T_{LOW}
1	0	1	T_{HIGH}
1	1	1	výrobné číslo

firma Dallas, ktorá ho dodáva v troch rôznych púzdrach a to: TO - 92, SO - 8 a μ SOP - 8 vidíme na obr. 1.17.



Obr. 1.17: Púzdra vyhotovenia DS18B20

Tento senzor je možné napájať dvoma spôsobmi a to klasickým napájaním zo zdroja alebo napájaním po dátovom vodiči. Na zbernicu One - Wire je možné zapojiť niekoľko senzorov. Vyberanie medzi týmito senzormi je možné pomocou jedinečného kódu, označený ako "ROM CODE ". Veľkosť tohto senzoru je 64 bitov, z toho posledných 8 bitov je kód rodiny. Pri týchto obvodoch je to pevne dané číslo 28h. Ďalších

48 bitov je unikátne sériové číslo, zvyšných 8 bitov je CRC, čo je kontrola cyklickým kódom, to znamená zabezpečenie proti chybám.

Každé čidlo obsahuje pamäť, nazývanú SCRATCHPAD o veľkosti 8 bajtov.

- **bajt 0** - teplota LSB,
- **bajt 1** - teplota MSB,
- **bajt 2** - T_H alarm horný,
- **bajt 3** - T_L alarm dolný,
- **bajt 4** - nastavovacie registre,
- **bajt 5** - rezervované,
- **bajt 6** - rezervované,
- **bajt 7** - rezervované,
- **bajt 8** - CRC.

Bajt 0 a 1 obsahuje teplotu, presnejšie vysvetlenie je v tab. 1.6 a tab. 1.7. Prevod na teplotu sa vytvára z týchto bajtov tak, že hodnoty I sú nad desatinnou čiarkou a čísla D sú desatinné miesta. Hodnoty bitov S sú podľa presnosti prevodu a určujú alarmy a hodnotu teploty - kladná alebo záporná.

Tab. 1.6: LSB bajt 0

LSB bajt 0							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I^3	I^2	I^1	I^0	D^{-1}	D^{-2}	D^{-3}	D^{-4}

Tab. 1.7: MSB bajt 1

MSB bajt 1							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
S	S	S	S	S	I^6	I^5	I^4

1.6.4 Komunikácia s čidlom

Komunikácia s čidlom má tri základné kroky. Prvé je potrebné inicializovať, teda zistiť, koľko zariadení je na zbernici a zistiť ich adresy. Nasleduje vybratie správneho zariadenia podľa adresy. V tejto chvíli komunikujeme iba s vybratým zariadením, ostatné čakajú na linke na svoj čas. Posledný krok je vyslanie príkazu pre senzor.

Tento príkaz priamo ovládajú senzor, napr. spustenie konverziu, prečítanie scratch-pad a mnoho iných. [14]

1.7 SPI zbernica

Ide o synchrónnu štvorvodičovú zbernicu. Pracuje v režime takzvanom "full duplex", čo znamená obojstranný prenos dát. Obvykle sa využíva ako prepojenie rôznych periférií s mikropočítačom. Protokol bol definovaný firmou Motorola v roku 1982. Zriadenia na zbernici sa označujú ako jedno master zariadenie a viacero slave zariadení. Táto zbernica sa nazýva aj takzvaná "four wire", môžeme ju vydieť na obr. 1.18, kde v ľavo vidíme zapojenie z jedným zariadením slave a na pravo je vidieť zapojenie z viacerými slave zariadeniami. [8] [9]

Obr. 1.18: SPI zapojenie

1.7.1 Popis pinov

- **SCLK** - sériové hodiny, výstup z master,
- **MOSI/MISO** - master výstup, slave vstup,
- **MISO/MOSI** - master vstup, slave výstup,
- **SS** - výber slave, výstup z master.

Alternatívne značenie, záleží od výrobcu ktoré z týchto označovaní použije.

- **SCK, CLK** - sériové hodiny, výstup z master,
- **SDI, DI, SI** - sériové dáta vstup,
- **SDP, DO, SO** - sériové dáta výstup,
- **CS, CSB, nSS, STE** - výberový vodič.

1.7.2 Zahájenie komunikácie

Pokiaľ je na zbernici iba jedno zariadenie môžeme výberový vodič vynechať a na zariadení môžeme rovno tento vodič pripojiť na $\log \mathrm{O}$, čo znamená že toto zariadenie bude jediné na zbernici a vždy bude vybraté na komunikáciu. Pri niektorých zariadeniach toto prepojenie nie je možné vykonať pretože podriadený obvod čaká na nábežnú hranu a tým nastane že zariadenie nereaguje, akoby nebolo vybraté. Pre zbernice na ktorých

1.7.3 Prenos dát

Pred samotným prenosom dát je potrebné nadstaviť hodinové impulzy. Frekvencia týchto impulzov musí byť nižšia alebo maximálne zhodná z frekvenciou ktorú podporuje slave zariadenie. Táto frekvencia môže byť v rozsahu od 1 MHz až po 70 MHz. Nasleduje vybratie zariadenia privedením $\log 0$ na výberový vodič správneho zariadenia. Master pošle jeden bit po linke MOSI kde sa zapíše do registra z kade si ho môže slave zariadenie prečítať, súčastne ale môže zariadenie slave poslať tiež jeden bit po linke MISO do master zariadenia kde sa zapíše do registra. Po prenose si tieto zariadenia môžu prečítať prijaté bity vo svojich registroch. Počet takto prenesených bitov musí byť definovaný ešte pred prenosom ale zvyčajne to je 8 bitov alebo 16 bitov. Mnoho zariadení nepotrebuje ani pin MOSI, napríklada sa jedná o digitálne teplotné čidlá ktoré hneď po vybratí výberovým pinom posielajú aktuálnu teplotu po pine MISO. Aby prenos správne fungoval je potrebné správne nadstaviť polaritu a fázu hodinového signálu, to sa nadstavuje obvykle pomocou registrov z označením CPOL a CPHA. Nadstavenia sú závislé od slave zariadenia. SPI neobsahuje opravný mechanizmus.

Klady rozhrania SPI.

- komunikácia full duplex,
- možnosť posielať aj viac bitov ako 8,
- nízke energetické nároky,
- nie je potrebné adresovanie,
- je možné signálové vodiče galvanicky izolovať.

Nevýhody komunikácie SPI.

- využíva minimálne 4 piny,
- nie je definovaný protokol kontroly chýb,
- neobsahuje potvrdenia o prijatí,
- žiadna kontrola toku dát.

Tento štandard je veľmi rozšírený používajú ho najrôznejšie zariadenia a snímače od snímačov tlaku, teploty, dotykové obrazovky a AD prevodníky. Pamäte typu flash alebo EEPROM obvykle používajú tento štandard. Vo veľkom sa používajú aj na

prepojenie obvodov ktoré udržujú čas aj počas vypnutého zariadenia, pripojenie SD alebo MMC karty je obvykle taktiež realizované týmto štandardom.

1.8 AD/DA prevodník

Jedno z úloh bolo navrhnúť analógovo digitálny a digitálno analógový prevodník. Z dostupných obvodov som zvolil obvod AMC 7823 od výrobcu Texas Instrument. Zvolený obvod obsahuje ako AD tak aj DA prevodník. Komunikuje po zbernici SPI. Na Beagleboarde je možnosť využiť až 3 zariadenia na tejto zbernici. Obvod má veľmi špecifické púzdro bol vytvorený samostatný plošný spoj. Špecifikácie obvodu. [6]

- 8 kanálový a 12 Bitový AD prevodník,
- rýchlosť vzorkovania až 200 kSPS,
- 8 kanálový a 12 Bitový DA prevodník,
- rýchlosť nadstavenia napätia do 2 μ s,
- 6 analógových vstupov z alarmom,
- interné referenčné napätie,
- možnosť pripojenie externej referencie,
- obvod obsahuje teplotný senzor,
- SPI reaguje na logickú úroveň od 3 V,
- možnosť prepnúť obvod do úsporného módu,
- miniatúrne púzdro typu QFN-40 o veľkosti 6 x 6 mm.

Maximálne rozsahy obvodu.

Tab. 1.8: Povolené rozsahy obvodu

Parameter	Rozsah	Jednotka
Napájacie napätie	$-0,3$ do $+6$	V
Analógové vstupné napätie	$-0,3$ do $+0,3$	V
Vstupný prúd	± 20	mA
Rozsah pracovnej teploty	-40 do $+105$	℃
Deštrukčné teploty	-65 do $+150$	℃

1.8.1 Komunikácia z obvodom

Komunikácia pozostáva z posielaním 16 bitových príkazov. Najvýznamnejší bit určuje typ príkazu či ide o zápis do registra alebo ide o čítanie z registra. V prípade

že je tento bit v hodnote log 0 ide o čítanie z registra, v prípade log 1 ide o zápis od registra. Pri čítaní z registra sa nadstavuje začiatočný register a konečný register. V pamäti sú uložené dáta o jednotlivých AD prevodoch. Obvod po prijatí takéhoto príkazu zasiela dáta z pamäte, ktoré sú ohraničené v príkaze. V prípade zaslania príkazu o zápis do registra je tento príkaz podobný ako príkaz pri čítaní. Zasiela sa začiatočný register a konečný register. Po taktom príkaze nasleduje dátový príkaz z hodonotou aká bude zapísaná do registra. Od tohoto registra sa odvíja výstupná hodnota pri DA prevodníku. Presnejšie informácie sú uvedené v dokumentácií obvodu. [6]

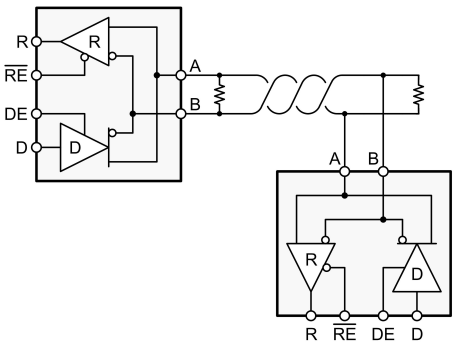
1.9 Budič linky

Toto rozhranie rozširuje možnosti beagleboardu. Na základnej doske je umiestnený 9 pinový D - sub konektor, ktorý obsahuje rozhranie RS - 232. Preto som zvolil budič linky na štandarde RS - 485. Táto zbernica má veľmi podobný štandard ako vyššie spomínaný RS - 232, ale líši sa hlavne inou napäťovou úrovňou, možnosťou vytvorenia siete až z 32 zariadeniami, možnosť komunikovať na 1200 m. Tento štandard definovalo EIA - 485 v roku 1983 ako štandard sériovej komunikácie. Navrhnutý je aby umožnil vytvorenie polo-duplexného dvojvodičového viacbodového sériového spoja. Vodiče v tomto vedení majú označenie A a B, alebo sa označujú ako $+$ a -. Prenosová rýchlosť je nepriamo úmerná dĺžke vedenia a to do vzdialenosti 10 m, čo môže byť až 10 Mbit/s. Pri dlhších vedeniach je odporúčané zakončiť vedenie na oboch stranách rezistor o hodnote 110 Ω. Tento rezistor zabráni odrazom signálu na konci vedenia, taktiež napomáha zvýšeniu odolnosti proti rušeniu. Impedancia linky je v tom prípade, budú paralelne zapojené ukončovacie rezistory o hodnote 110 Ω, bude výsledná impedancia linky len 55 Ω viz obr. 1.19.

Reprezentácia logických úrovní na linke sa určuje ako rozdiel medzi oboma vodičmi. Pri RS - 232 sa tento rozdiel vzťahuje k referenčnej zemi. Detekcia logického stavu z rozdielu medzi vodičmi je veľmi odolná proti indukcii rušivého signálu, keďže v prijímači sa toto indukované napätie odčíta a výsledkom je správna reprezentácia logického stavu. Prijímač rozlišuje log 1, ak je rozdiel na vodičoch A a B väčší ako +200 mV. Pri log 0 je rozdiel medzi A a B menší ako -200 mV. []

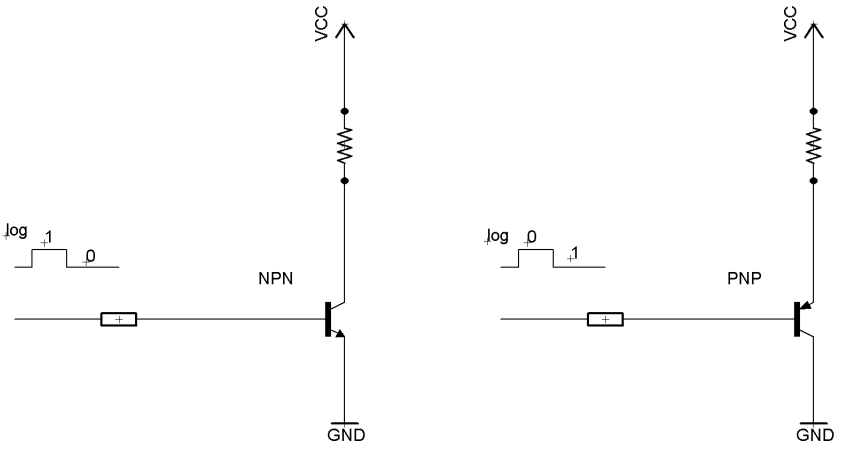
1.10 Výkonové spínače

Rozšírenie o tento modul je potrebné v prípade, že by sme pomocou beagleboardu chceli ovládať komponenty z väčším odberom ako je povolený. Tento problém je možné riešiť viacero spôsobmi, v nasledujúcej časti si niektoré popíšeme.



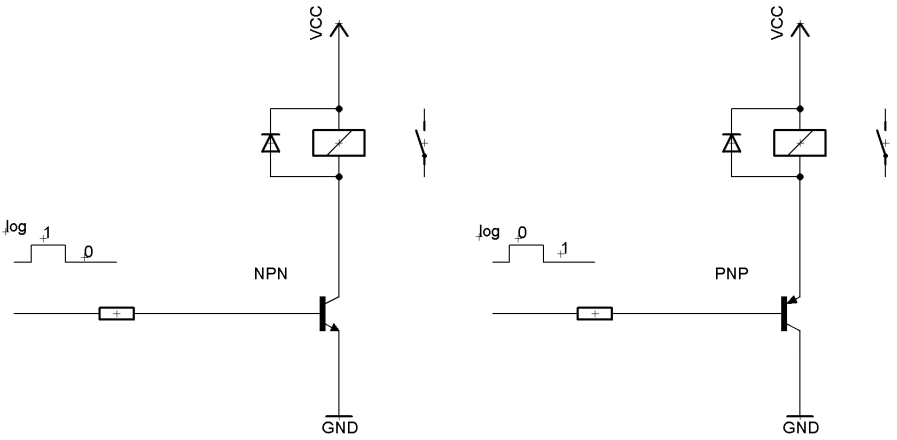
Obr. 1.19: Zbernica rozhrania RS - 485

Riešenie je posilniť výstupný pin pomocou tranzistoru. Možné je využiť tranzistory typu NPN aj PNP, zapojenie je vidieť na obr. 1.20. Tento spôsob je jednoduchý a lacný, ale nie moc bezpečný v prípade prerazenia tranzistora, môže sa cez ochranný rezistor dostať vysoké napätie na pin, kde môže zničiť výstup. Toto riešenie je vhodné na spínanie nižších jednosmerných napätí do 40 V.



Obr. 1.20: Zapojenie s tranzistormi

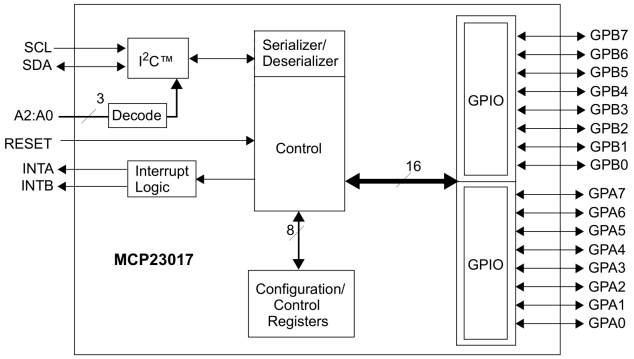
Preto toto zapojenie vylepšíme elektromagnetickým spínacím prvkom. Toto relé má väčší odber ako je povolený výstup beagleboardu, preto je potrebné využiť zapojenie s tranzistorom. Takéto zapojenie vidíme na obr. 1.21. Toto zapojenie je omnoho bezpečnejšie, pretože spínané napätie nezasahuje do obvodu. Relé môže spínať aj striedavé aj jednosmerné napätie. Výstupný výkon je obmedzovaný použitým relé. K cievke relé je paralelne pripojená univerzálna dióda, ktorá odstraňuje impulz, ktorý vzniká pri odpojení cievky relé. Tento impulz by inak mohol zničiť spínací tranzistor. Tento jav je spôsobovaný indukčnosťou cievky.



Obr. 1.21: Zapojenie s tranzistorom a relé

1.11 Expandér

Tento obvod slúži k rozšíreniu samotného beagleboardu o ďalšie vstupno výstupné piny. Tieto expandéry sa vyrábajú v rôznych veľkostiach a rôznych typov komunikácie. Zvolil som obvod, ktorý obsahuje 16 pinov. Tieto piny je možné využiť ako vstup alebo výstup, ich funkcia sa dá programovo zmeniť. Obvod má možnosť pomocou 3 bitov zmeniť adresu, preto je takýchto obvodov možné zapojiť paralelne až 8, čo by rozšírilo beagleboard o 128 pinov. Expandér pracuje na napájacom napätí $+5\,V$. Prúdové zaťaženie je maximálne 25 mA na pin alebo 125 mA na celý obvod. Keďže obvod pracuje na napätí 5 V je nebezpečné ho pripájať na beagleboard, ktorý má 1,8 V logiku. Preto aj rozhranie I2C je potrebné previesť cez menič úrovne, týmto ochránime beagleboard pred zničením. Vnútornú štruktúru obvodu vidíme na obr. 1.22.



Obr. 1.22: Vnútorná štruktúra MCP23017

2 VÝSLEDKY ŠTUDENTSKEJ PRÁCE

2.1 Inštalácia systému

Zo všetkých možných dostupných linuxových distribúcií bol vybraný ako operačný systém Ubuntu. Nainštalovať tento systém je zložitejšie ako na normálnom počítači, pretože je potrebné celú inštaláciu vykonávať na počítači kde už je nainštalovaná linuxová distribúcia operačného systému. Prvým krokom je nainštalovať do ubuntu nástroj na formátovanie a prácu z SD kartou. Toto sa vykoná príkazom.

```
sudo apt-get install wget pv dosfstools parted
```

Druhým krokom je stiahnutie pred inštalovaného balíčka, to sa vykoná nasledujúcim príkazom.

```
wget http://rcn-ee.net/deb/rootfs/quantal/
                ubuntu-12.10-console-armhf-2013-09-26.tar.xz
```

Ďalším krokom je rozbalenie tohoto balíčka. Súbor má koncovku .tar, čo je vlastne linuxová alternatíva windovsovského .rar.

```
tar xJf ubuntu-12.10-console-armhf-2013-04-26.tar.xz
```

Presunieme sa do zložky kde bol rozbalený balík.

```
cd ubuntu-12.10-console-armhf-2013-04-26
```

Nasleduje krok v ktorom zistíme ako je zobrazená naša SD karta. Všetky zariadenia vypíšeme príkazom.

```
df -h
```

Zobrazia sa nám všetky zariadenia a diskové jednotky.

```
Filesystem            Size  Used Avail   Use\% Mounted on
/dev/sda1             10G   1G    9G     10\%  /
none                  943M  652K  942M   1\%   /dev
none                  949M  1.1M  948M   1\%   /dev/shm
none                  949M  100K  949M   1\%   /var/run
none                  949M     0  949M   0\%   /var/lock
/dev/sdb1             7.2G   32K  7.2G   1\%   /media/
```

Správne formátovanie a presunutie na SD kartu vykoná príkaz uvedený nižšie. Aby fungovalo formátovanie je potrebné nainštalovať program krotý je v prvok kroku popísaný.

```
sudo ./setup_sdcard.sh --mmc /dev/sdb --uboot beagle_xm
```

Vykonanie toho príkazu bude chvíľku trvať. Skript si bude sťahovať zavádzače z internetu. Nasleduje skopírovanie rozbaleného balíka na SD kartu. Po úspešnom

vykonaní týchto príkazov vypíše do konzole nasledujúce riadky, súčasťou je aj meno a heslo na prihlásenie do systému.

```
Finished populating rootfs Partition
-----------------------------
setup_sdcard.sh script complete
-----------------------------
The default user:password for this image:
ubuntu:temppwd
```

Môžeme odpojiť kartu zo systému, vybrať z čítačky, vložiť do Beagleboardu, zapnúť napájanie. Systém by sa mal do niekoľkých minút na monitore spustiť terminál. V prípade že sa tak nestane je potrebné kartu vrátiť do čítačky a upraviť konfiguračný súbor. Názov toho súboru je *uEnv.txt* v nom je potrebné povoliť grafický výstup, pretože štandardne je zakázaný. Povolenie tohoto portu je uvedené nižšie, je dobré definovať aj rozlíšenie monitora na ktorý sa bude potom beagleboard pripájať. V mojom prípad to bolo plné HD rozlíšenie. [17]

```
#Video: Uncomment to override U-Boots value:
defaultdisplay=dvi
dvimode="omapfb.mode=dvi:1920x1080"
vram=12MB\textbf{}
```

2.2 Nastavenie statickej IP adresy

Konfigurovanie beagleboardu je možné priamo pomocou pripojenej klávesnice. Výsledok je vidieť priamo na obrazovke alebo displeji. Keď sa chceme vyhnúť pripájaním periférií je možnosť sa na beagleboard pripojiť pomocou SSH protokolu. Tento protokol dokáže plnohodnotne ovládať priamo celý operačný systém. Aby sme sa mohli vždy pripojiť na rovnakú IP adresu. Je potrebné nastaviť statickú IP adresu zariadenia. Inak by sme vždy museli pátrať akú adresu pridelil router beagleboardu. Otvoríme konzolu a celé nastavenie prevedieme cez konzolu. Nasledujúcim príkazom zobrazíme všetky rozhrania.

```
cat /etc/network/interfaces
```

Ak je nastavený beagleboard na použitie DHCP serveru ktorý prideľuje IP adresu automaticky vypíše sa nasledovný riadok.

```
iface eth0 inet dhcp
```

Pokiaľ je nastavená statická IP adresa vypíše nám konzola nasledovný riadok.

```
iface eth0 inet static
```

Vytvoríme si zálohu konfiguračného súboru, to prevedieme nasledujúcim príkazom. Tento príkaz ale potrebuje zadanie hesla root.

```
sudo cp /etc/network/interfaces /etc/network/interfaces.backup
```

Môžeme pristúpiť k samotnej konfigurácií. Otvoríme súbor v editore ja som vybral editor *nano*. Uskutočníme to nasledovným príkazom.

```
sudo nano /etc/network/interfaces
```

Editujeme súbor do nasledujúceho tvaru.

```
iface eth0 inet static
    address 192.168.1.104
    netmask 255.255.255.0
    gateway 192.168.1.1
```

Uložíme konfiguračný súbor a ukončíme editor. Zariadenie reštartujeme aby sa nadstavenia zaviedli do operačného systému, tento krok prevedieme nasledovným príkazom.

```
sudo shutdown -r now
```

Po nabehnutí zariadenia môžeme dané nadstavenia overiť pomocou jednoduchého príkazu.

```
ifconfig eth0
```

Do konzole sa nám vypíše aktuálne používaná IP adresa a maska siete. [17]

2.3 Nastavenie rozširujúceho portu

Tento port slúži beagleboardu k rozšíreniu svojich možností. Pinov je dokopy 28, z toho je 22 signálových, ostatné slúžia ako výstup napätia alebo zeme, poprípade ako reset pin. Tieto signálové cesty sú smerované rovno z procesora. Procesor má funkciu multiplexovania, preto môže rôzne signáli smerovať na rôzne výstupné piny. Každý pin je možné multiplexovať do siedmych režimov, všetky tieto režimy sú uvedené v tab. D.1. V tabuľke sú uvedené iba prvé štyri režimy, zvyšné pre moju prácu nie sú potrebné. Prvý stĺpec tejto tabuľky určuje poradie pinu na rozširujúcom konektore. Druhý stĺpec označuje konektor na procesore. Zvyšné štyri rozlišujú mód daného pinu, takzvaný MUX. Nadstavením prislúchajúcej hodnoty v riadiacom registry bude vybraný typ signálu smerovaný na zvolený pin. Mód 0 je štandardne zvolený. Tento mód je možné zmeniť dvoma spôsobmi. Prvý spôsob je predefinovať mód ešte pred tým ako preložíme jadro operačného systému. Upravením konfiguračného súboru *mux.c* je možné dopredu predefinovať všetky piny. Príklad takejto definície je nižšie.

```
MUX_CFG_34XX("AE4_34XX_GPIO136_OUT",
        0x164, OMAP34XX_MUX_MODE4 | OMAP34XX_PIN_OUTPUT)
MUX_CFG_34XX("AF6_34XX_GPIO140_UP",
        0x16c, OMAP34XX_MUX_MODE4 | OMAP34XX_PIN_INPUT_PULLUP)
```

Druhá možnosť je definovať mód pinu za behu operačného systému. Je potrebné konfigurovať súbor *beagle.h*. Jeho umiestnenie je vždy trochu iné, pretože je závislé od typu operačného systému. Uvediem príklad pre operačný systém Angstrom, ktorý je nahratý na pamäťovej karte ako úvodný operačný systém.

```
tmp/work/beagleboard-angstrom-linux-gnueabi/u-boot
                /git/board/ti/beagle/beagle.h
```

V tomto konfiguračnom súbore nájdeme každý pin definovaný v móde 0. Takúto definíciu vidíme nižšie. Presnejšie je to pin 24.

```
MUX_VAL(CP(I2C2_SCL), (IEN|PTU|EN|M0)) /*I2C2_SCL*/\
```

Tento kód určuje že pin 24 je v móde 0, čo je patrné z posledného parametra ktorý je nadstavený na M0. V prípade, že by sme chceli mód zmeniť riadok, by sme prepísali takto.

```
MUX_VAL(CP(I2C2_SCL), (IEN|PTU|EN|M4)) /*I2C2_SCL*/\
```

V tejto chvíli sme zmenili mód na režim 4, čo z tabuľky označuje GPIO 168. Tento výstup je univerzálny vstup alebo výstup. Následne tento upravený súbor uložíme a musíme ho skompilovať aby sa vytvorené zmeny prejavili. V tejto chvíli môžeme pin vyskúšať. Príkazom

```
cd /sys/class/gpio/
echo 168 > export
cd gpio168
```

Prepneme do zložky, kde je možné pracovať s rozširujúcim portom. Nasledujúcim príkazom sa prepneme na nami zvolený port. Nadstavíme pin ako výstupný.

```
echo out > direction
```

A samotná zmena výstupu.

```
echo 1 > value  // pre hodnotu 1
echo 0 > value  // pre hodnotu 0
```

Pre potrebu tohoto projektu je potrebné správne predefinovať skoro všetky piny. V tab. 2.1 je uvedené správne definovanie. Ďalšou z možností ako definovať piny je priamo za behu operačného systému. Je to možné uskutočniť nasledovným príkazom. Táto zmena sa prejaví okamžite bez potreby reštartovať operačný systém alebo prekladať jadro. [10]

```
/sys/kernel/debug/omap_mux # echo 0x0001 > uart2_tx
```

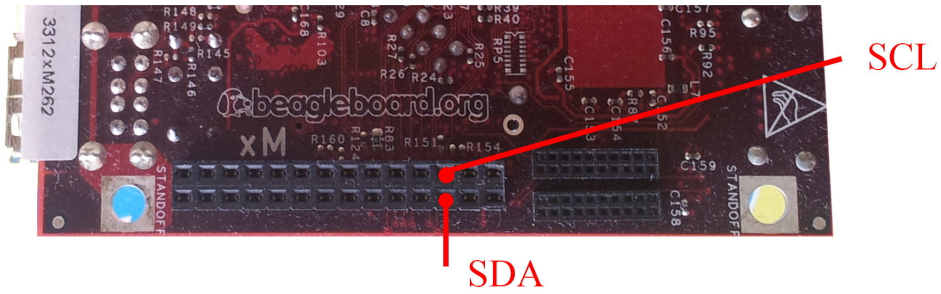
Tab. 2.1: Nastavenie módov rozširovacieho portu

PIN	Proc	0	1	2	3	4
1		Vcc $+1,8\,V$				
2		Vcc $+5\,V$				
3	AE39	-	-	-	-	**GPIO_139**
4	AB26	-	-	**GPT9 PWMEVT**	-	-
5	AF3	-	-	-	-	**GPIO_138**
6	AA25	**UART2_TX**	-	-	-	-
7	AH3	-	-	-	-	**GPIO_137**
8	AE5	-	**UART2_RX**	-	-	-
9	AE4	-	-	-	-	**GPIO_136**
10	AB25	**UART2_RTS**	-	-	-	-
11	AF4	-	**McSPI3_CS0**	-	-	-
12	V21	-	**McSPI4_SIMO**	-	-	-
13	AG4	-	**McSPI3_CS1**	-	-	-
14	W21	-	-	-	-	**GPIO_162**
15	AH4	-	-	-	-	**GPIO_133**
16	K26	-	**McSPI4_CS0**	-	-	
17	AH5	-	**McSPI3_SOMI**	-	-	-
18	U21	-	**McSPI4_SOMI**	-	-	-
19	AG5	-	**McSPI3_SIMO**	-	-	-
20	Y21	-	**McSPI4_CLK**	-	-	-
21	AE2	-	**McSPI3_CLK**	-	-	-
22	AA21	-	-	-	-	**GPIO_157**
23	AE15	**I2C2_SDA**	-	-	-	-
24	AF15	**I2C2_SCL**	-	-	-	-
25		Regen				
26		Reset				
27		GND				
28		GND				

2.4 I2C rozhranie

Beagleboard XM obsahuje dokopy 3 zbernice I2C. Prvá I2C1 zbernica je využité k riadeniu grafického výstupu DVI. Druhá z názvom I2C2 je vyvedené do rozširujúceho konektoru, toto rozhranie budeme využívať ako testovacie rozhranie. Posledné tretie rozhranie I2C3 slúži na správu a kontrolu sekvenčného napájania. Rozhranie

je na nadstavené na výstup 1,8 V, preto je potrebné použiť menič napäťovej úrovne. Zvolil som špecializovaný obvod od firmy Texas Instrument. Ide o obvod PCA9306, jeho bližší popis je uvedený v teoretickom úvode. Obvod je dimenzovaný na obojsmerný prenos dát po zbernici I2C. Pre fungovanie je potrebné správne predefinovať výstupné piny a to SDA čo je dátový vodič a SCL synchronizačný vodič. Definovanie prebieha pomocou *PIN MUXU*, tiež popísané práci presnejšie môžeme vidieť na obr. 2.1. Aby samotné I2C2 ktoré nás zaujíma pracovalo musí byť v jadre modul ovládačov pre dané rozhranie. V starších moduloch jadra bolo potrebné tento modul inštalovať ale v mnou zvolenom jadre už modul bol inicializovaný. Ako jednoduchý test funkčnosti môžeme skontrolovať či existuje zložka : $-/dev/i2c-2$", ak táto zložka existuje rozhranie I2C2 je pravdepodobne funkčné.



Obr. 2.1: I2C v rozširovacom konektore.

Existuje nástroj ktorým môžeme jednoducho odskúšať funkčnosť rozhrania. Názov nástroja je I^2C tools. Nástroj bol vytvorený pre sondovanie a komunikáciu z rôznymi senzormi na základných doskách a ných zariadeniach. Je volne šíriteľný. Nainštalovať je ho možné pomocou nasledovného príkazu. Zápis príkazu sa líši podľa použitej distribúcie operačného systému.

```
# apt-get install i2c-tools
```

Inštalovaný balíček má mnoho nástrojov jedným z nich je aj detekcia zariadenia na rozhraní. Detekcia sa spúšťa nasledovným príkazom.

```
# i2cdetect -y -r 2
```

Príznak -r znamená čítať a príznak 2 je určenie že sa má čítať zo zbernice I2C-2. Príznak -y znamená YES ktorý definuje že vieme čo robíme. V prípade že bude chýbať príznak -y, nástroj sa opýta či chceme naozaj vypísať adresy. Príkaz vypíše adresy všetkých dostupných zariadení. Výpis vyzerá nasledovne.

```
# i2cdetect -y -r 2
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
```

```
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Týmto výpisom sme dokázali overiť funkčnosť rozhrania. K Beagleboardu nebolo pripojené žiadne zariadenie preto sú všetky riadky prázdne. V prípade že zariadenie bude pripojené vypíše sa nasledovný výsledok.

```
# i2cdetect -y -r 2
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: 50 -- -- -- UU UU UU UU -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Výpis znamená že zariadenie je pripojené a má adrese 0x50. Môžeme ešte overiť či zariadenie komunikuje. Na toto overenie slúži program i2cdump ktorý je súčasťou inštalovaného balíka. Ako predchádzajúci príkaz potrebuje argument povolenia -y, nasleduje argument zbernice pre nás 2, adresa zariadenia pre nás 0x50 a mód prenosu c znamená postupné posielanie po sebe idúcich bytov. Výpis bude nasledovný.

```
# i2cdump -y 2 0x50 c
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  0123456789abcdef
00:4c 4d 39 32 43 49 4d ff ff ff ff ff ff ff ff ff  LM92CIM.........
10:ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
20:ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
30:ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
40:ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
50:ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
60:ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
70:ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
80:ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
90:ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
a0:ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
b0:ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
c0:ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
d0:ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
e0:ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
f0:ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
```

V prípade že zariadenie nie je pripojené a neodpovedá výsledok tohoto príkazu bude nasledovný.

```
# Error: Write start address failed, return code -1
```

2.4.1 Komunikácia cez I2C v jazyku C

Pred zahájením komunikácie je potrebné najprv otvoriť zbernicu. Daný súbor musí mať dostatočne oprávnenie k prístupu, častým problémom je oprávnenie ktoré nie je dostatočné pre komunikáciu na zbernici. Otvorenie zbernice sa vykonáva pomocou jednoduchej štruktúry, tá je vyobrazená nižšie. Kód vypíše chybu v prípade že sa nepodarí otvoriť zbernicu.

```
int file;
char *filename = "/dev/i2c-2";
if ((file = open(filename, O_RDWR)) < 0) {
    /* ERROR */
    perror("Failed to open the i2c bus");
    exit(1);
}
```

Následným krokom je inicializácia zariadenia podla adresy. Adresa je buď uvedená v datasheete použitého obvodu alebo vyhľadať adresu zariadenia podla vyššie uvedeného postupu. Tento postup sa aplikuje napríklad pre drobné obvody akými sú teplomery v prípade obvodov z rozšírenými funkciami je adresa uvedená v datasheete daného obvodu. Inicializácia zariadenia sa vykonáva pomocou nasledujúcej štruktúry. [12]

```
int addr = 01010000;          // adresa zariadenia 0x50
if (ioctl(file, I2C_SLAVE, addr) < 0) {
    printf("Chyba.\n");
    /* ERROR */
    exit(1);
}
```

2.5 Rozhranie 1-WIRE

Na obsluhu tejto zbernice je vyhradený pin z označením HDQ. Do jadra je potrebné zaviesť ovládač. Po úspešnom zavedení ovládačov je možné vypísať na ktorom pine je definovaná zbernica 1-Wire.

```
# dmesg | grep w1
[ 0.208956 ] w1-gpio connected to GPIO_138
```

Pripojením senzoru DS18B20 na tento pin je možné snímať teplotu z tohto senzoru. Výpis zariadení na zbernici.

```
# ls -la /sys/bus/w1/devices
                28-00000256073
```

Vypísanie teploty je možné pomocou príkazu.

```
# cat /sys/bus/w1/devices/28-00000256073a/w1_slave
        b8 01 4b 46 7f ff 08 10 8a : crc=8a YES
        b8 01 4b 46 7f ff 08 10 8a t=27500
```

Teplota je zakódovaná v premennej t=27500, prevod je jednoduchý stačí hodnotu t podeliť 100 a výsledok je teplota = 27,5 °C. Je možné vytvoriť jednoduchý skript ktorý priamo vypíše teplotu na senzore. Príklad je uvedený nižšie.

```
# cat /sys/bus/w1/devices/28-00000256073a/w1_slave | tail -n1
                | awk -F= '{print \$2/1000 " C"}'
        27.500 C
```

2.6 Ovládanie GPIO pinov

2.6.1 LED kontrolka

Uvedený popis je zobrazený na príklade obyčajnej svetelnej diódy. Pokiaľ by sa výstup z Beagleboardu poslinil tranzistorom alebo logickým obvodom bolo by možné spínať relé ktoré by mohlo spínať aj výkonnú záťaž. Príklad budem demonštrovať na pine GPIO 139, ktorý je na rozširovacom konektore umiestnený pod pinom číslo 3. Prvým krokom je pripojiť GPIO pin do systému. To sa vykoná pomocou nasledovného príkazu.

```
# echo 139 > /sys/class/gpio/export
```

Pin je ďalej potrebné nastaviť na výstup. Nasledovným príkazom je to možné uskutočniť.

```
# echo out > /sys/class/gpio/gpio139/direction
```

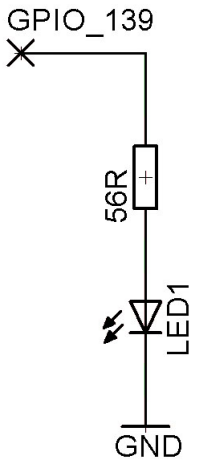
Nadstaviť pin tak aby na pine bola $\log 1$ môžeme príkazom nižšie a druhým riadkom je možné nastaviť $\log 0$.

```
# echo 1 > /sys/class/gpio/gpio139/value
        // privedia ne pin GPIO 139 napetie 1,8V
# echo 0 > /sys/class/gpio/gpio139/value
        // privedie na pin GPIO 139 GND
```

Test týchto príkazov je možný pomocou zapojenia na obr. 2.2.

2.6.2 Snímanie hodnoty zo spínača

Príklad uvádzam taktiež pre pin GPIO 139. Postup je podobný ako tomu bolo pri LED dióde.



Obr. 2.2: Led dióda pripojená na GPIO 139.

```
# echo 139 > /sys/class/gpio/export

# echo in > /sys/class/gpio/gpio139/direction
```

Nasledovný príkaz vypíše hodnotu na pine. V prípade že je spínač rozpojený výstup z príkazu bude $\log 1$ a pri stlačenom spínači vypíše príkaz $\log 0$.

```
# cat /sys/class/gpio/gpio139/value
```

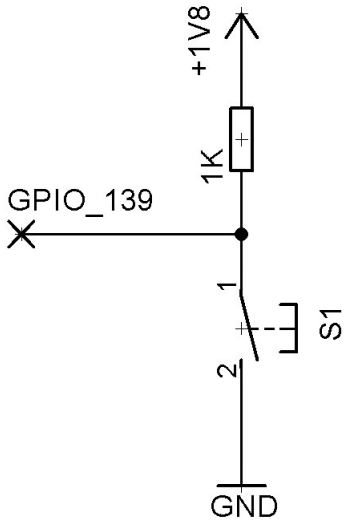
Je možné tento príkaz spustiť v nekonečnom cykle. Skript bude vypisovať každú sekundu stav spínača.

```
# GPIO=139; while true; do sleep 1;
        [ $(cat /sys/class/gpio/gpio$GPIO/value) = "1" ]
                && echo -n -|| echo -n _; done
```

Zapojenie takéhoto spínača je vyobrazené na obr. 2.3.

2.7 Generovanie PWM modulácie

PWM je skratka pre pulznú šírkovú moduláciu, presnejší popis je uvedený v teoretickom úvode práce. Táto modulácia slúži na rôzne účeli od regulácie výkonu až po ovládanie servo motorov. Samotný beagleboard má v rozširovacom konektore tri piny určené pre generovanie pwm modulácie. V nasledujúcich riadkoch bude vysvetlené ako postupovať pokiaľ chceme generovať pwm signál. [11]



Obr. 2.3: Spínač zapojený na GPIO 139.

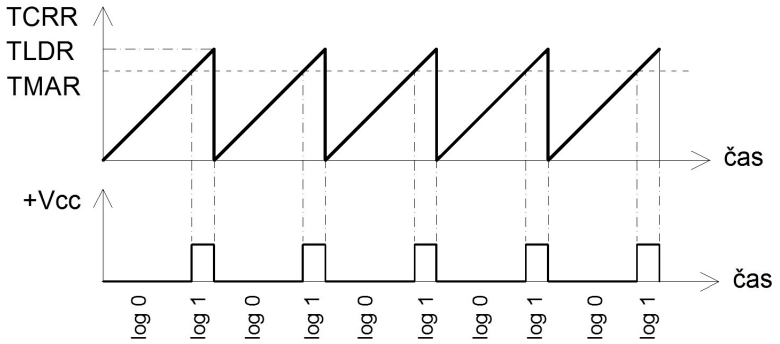
2.7.1 PIN MUX pre PWM

Prvým krokom je potrebné predefinovať v konfigurácií pinov režim generovania PWM. Primárne sú definované inak preto je potrebné piny prepnúť do modu 2. Máme na výber z troch možných pinov. Krok ako predefinovať piny je popísaný v práci taktiež. Tento krok môže byť problematický pre staršie verzie jadra ako je 2.6.32.

```
OMAP3_MUX(DSS_DATA15, OMAP_MUX_MODE2|OMAP_PIN_OUTPUT),
                /* GPT9_PWMEVT, pin 4 */
OMAP3_MUX(UART2_TX, OMAP_MUX_MODE2|OMAP_PIN_OUTPUT),
                /* GPT11_PWMEVT, pin 6 */
OMAP3_MUX(UART2_RTS, OMAP_MUX_MODE2|OMAP_PIN_OUTPUT),
                /* GTP10_PWMEVT, pin 10 */
```

2.7.2 Aktivácia a registre

Generovanie PWM sa vykonáva pomocou čítača ktorý obsahuje samotný procesor. Stručný popis ako funguje čítač. Čítač inkrementuje register každým taktom procesora. Keď sa dosiahne istá definovaná hodnota prepne sa výstup na PWM pine, hodnota v registry sa inkremuntuje ďalej pokiaľ sa nedosiahne hodnota plného registra. Vtedy registre pretečie, výstup na pine PWM sa preklopí a čítanie sa opakuje z tým že hodnota v registry sa vynuluje. Generovanie je lepšie vidieť na obr. 2.4.



Obr. 2.4: PWM a registre.

Tab. 2.2: Adresy v pamäti pre jednotlivé časovače.

	Časovač	Adresa	Pin
1	GPTIMER09	0x4904 0000	4
2	GPTIMER10	0x4808 6000	6
3	GPTIMER11	0x4808 8000	10

Tab. 2.3: Tabuľka registrov.

Názov	Offset	Popis
TCLR	0x024	Riadiaci register
TCRR	0x028	Inkrementačný register
TLDR	0x02C	Maximálna hodnota pretečenia
TMAR	0x038	Hodnota kde pin zmení hodnotu

Každý časovač má vyhradené 4 KBit v pamäti. Začiatok adresy tejto pamäte je uvedený v tab. 2.2.

Prístup k týmto registrom je možný pomocou modulu zavedenom v jadre. Alebo môžeme priamo pristupovať k hodnote registra, priamo v fyzickej pamäti. Náhľad do tejto pamäte je možný v zložke */dev/mem*. Pri tomto prístupe môže nastať problém ak chceme do registra pristúpiť a tento register je systémovo volaný a tým uzavretý pre čítanie a zmenu. Obídenie je možné pomocou funkcie *mmap*(). Vzorový program, definuje PWM generovanie na pin 9 zo striedou 1 : 1. Kód si uložíme ako *demo.c*.

```
#include <glib.h>
#include <errno.h>
```

```
#include "omap3530-pwm.h"

int main(int argc, char **argv) {
    int mem_fd = pwm_open_devmem();
    if (mem_fd == -1) {
        g_error("Nemožno otvoriť /dev/mem,
                        ste root ?: %s", g_strerror(errno));
    }
    pwm_config_clock(mem_fd, TRUE, TRUE);
    guint8 *gpt9 = pwm_mmap_instance(mem_fd, 9);
    guint32 resolution = pwm_calc_resolution
                        (20000, PWM_FREQUENCY_13MHZ);

    pwm_config_timer(gpt9, resolution, 0.5);
    pwm_munmap_instance(gpt9);
    pwm_close_devmem(mem_fd);
}
```

Takto vytvorený program je možné preložiť pomocou nasledujúceho príkazu.

```
#gcc demo.c -o pwm-demo -lglib-2.0 -I/usr/include/glib-2.0
                -I/usr/lib/glib-2.0/include
```

V prípade že tento kód nie je možné spustiť bude problém z prístupom k registrom a hodinám. Výpis chyby je nasledovný.

```
# ./pwm-demo
0
Bus error
# dmesg | tail -1
[  174.893035] Unhandled fault:
        external abort on non-linefetch (0x1818) at 0x4001e024
```

Aby bolo možné spustiť program je potrebné v konfigurácií jadra zakázať položku *CONFIG OMAP RESET CLOCKS* a preložiť jadro. Po úspešnom preložení by mal byť program spustiteľný a mal by na pine 4 (GPT9 PWMEVT) generovať PWM moduláciu zo striedou "50%".

2.8 SPI rozhranie

Procesor doske Beagleboardu xM má možnosť ovládať až 4 SPI rozhrania. Prvé dve rozhrania sú využité na samotnom Beagleboarde, k týmto rozhraniam sa nedostaneme, tým pádom ich nemáme ako využiť. Pre moju prácu sú zaujímavejšie rozhrania SPI 3 a SPI 4. Rozhrania sú vyvedené do rozširovacieho portu, presný popis funkcie a čísla pinu je uvedený v tab. 2.4. [11]

Tab. 2.4: Tabuľka pinou rozhrania SPI.

Pin	Názov pinu na procesore	MUX
11	AF4	$McSPI3_CS0$
12	V21	$McSPI4_SIMO$
13	AG4	$McSPI3_CS1$
16	K26	$McSPI4_CS0$
17	AH5	$McSPI3_SOMI$
18	U21	$McSPI4_SOMI$
19	AG5	$McSPI3_SIMO$
20	Y21	$McSPI4_CLK$
21	AE2	$McSPI3_CLK$

Prvým krokom je potrebné stiahnuť zdrojové súbory jadra ktoré majú zavedené ovládače zbernice SPI. Použijeme jadro Angstrom Linuxu. Následne bude potrebné skompilovať jadro pre danú platformu. Tieto kroky preto vykonávam na stolovom počítači, má omnoho vyšší výkon a preloženie nebude trvať pár minút. Uskutočním to nasledujúci príkaz.

```
# git clone git://gitorious.org/angstrom/angstrom-linux.git
# cd angstrom-linux
# git checkout 12ca45fea91cfbb09df828bea958b47348caee6d
# git checkout -b spi
```

Po úspešnom stiahnutí zdrojových súborov môžeme pokračovať v práci. Je potrebné nadstaviť správne multiplexovanie signálov z procesora. Toto je možné nadstaviť dvoma spôsobmi buď priamo v jadre alebo v súboru U-boot. Popíšem spôsob úpravy v jadre. Do tohoto súboru je potrebné vložiť pár riadkov kódu. Tento krok sa dá uskutočniť aj pomocou pachu, je to súbor ktorý dokáže po spustení modifikovať daný súbor.

```
# nano arch/arm/mach-omap2/board-omap3beagle.c
```

Môžeme pristúpiť ku kroku samotnej kompilácie.

```
# make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- mrproper
# make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- omap3_beagle_defconfig
# make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- menuconfig
```

Počas kompilácie sa zobrazí jednoduché menu kde je potrebné zvoliť preloženie jadra aj z ovládačmi pre SPI rozhranie. Je potrebné povoliť tieto tri položky.

```
Device Drivers > SPI Support > Debug support for SPI drivers
Device Drivers > SPI Support > McSPI driver for OMAP24xx/OMAP34xx
Device Drivers > SPI Support > User mode SPI device driver support
```

Uložiť konfiguráciu a spustiť preklad jadra. Aby bolo možné preložiť jadro, je potrebné nainštalovať knižnicu *ncurses*.

```
apt-get install libghc-ncurses-dev
```

```
# make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- uImage
```

Postup ako otestovať či sme všetko nadstavili správne a rozhranie funguje môžeme jednoducho. V zložke *Documentation/spi/* je súbor z názvom *spidev$_t$est.c*. Súbor preložíme do binárneho tvaru aby bol následne spustiteľný na beagleboarde.

```
# arm-linux-gnueabi-gcc Documentation/spi/spidev_test.c -o spitest
```

Potrebujeme presunúť skompilované jadro na SD kartu, ktorú následne vložíme do beagleboardu. Na kartu skopírujeme súbor *uImage* ktorý bol vygenerovaný preložením jadra. Je umiestnený na karte v oddiely */boot*. Taktiež si musíme na kartu presunúť vytvorený a preložený testovací program. Ten ale musíme skopírovať do zložky kde sú uložené dáta. Rozoznáme ju podla toho, že je naformátovaná do *ext3* formátu. Pre test rozhrania SPI musíme ešte na beagleboarde nakrátko prepojiť piny 12 a 18, tým spojíme vysielací pin z prijímacím pinom. Spustením programu overíme funkčnosť, do konzoly sa vypíše všetko čo odošleme rozhraním SPI. V zložke */dev* si môžeme zobraziť tri zariadenia *spidev3.0 spidev3.1 spidev4.0*.

```
# ./spitest -D /dev/spidev4.0
```

2.9 Testovacie dosky

Pri testovaní jednotlivých rozhraní, nastával problém z prepojovaním jednotlivých periférií na nepájivom poli. Jednoduchšie bolo vyrobiť si jednoduché prípravky ktoré by bolo možné obmeniť do niekoľkých sekúnd, zasunutím do rozširovacieho konektora. Vyrobil som tri rôzne dosky každá určená k trochu inému využitiu.

2.9.1 Doska z LED diódami

Táto doska je z nich najjednoduchšia. Obsahuje iba 10 led diód. Doska neobsahuje žiadne prevodníky napäťovej úrovne. Výstup z beagleboardu je 1,8 V čo je dostatočné veľké napätie na rozsvietenia led diódy. Prúdové zaťaženie pri rozsvietenej led dióde je 8 mA čo je v povolenom rozmedzí jednotlivých GPIO pinou. Schému, plošný spoj a predlohu osadzovania môžeme vidieť na obrázku B.1 v prílohe.

2.9.2 Doska z teplotným senzorom LM92

Doska obsahuje jednoduchý prevodník napäťovej úrovne určený pre rozhranie I2C. Doska dokopy obsahuje dva obvody jedným je prevodník úrovne a druhý z dvojice je teplotný senzor. Doska bola zhotovená za účelom jednoduchej demonštrácie rozhrania I2C. Bližší popis obvodov aj rozhrania je rozobratý v práci. Navrhnutú testovaciu dosku je umiestnená v prílohe na obr. B.4.

2.9.3 Doska z prevodníkom napäťovej úrovne

Táto doska je navrhnutá za účelom univerzálnosti, obsahuje dva 8 bitové prevodníky napäťovej úrovne. Touto doskou je možné následne pracovať z 5 V alebo 3,3 V logikou na nepájivom poli. Náhľad tejto dosky je umiestnený v prílohách na obr. B.6. Doska obsahuje konektor kompatibilný z rozširovacím konektorom na Beagleboarde. Následne je táto doska vsunutý do Beagleboardu.

2.10 Návrh a výroba dosky

Návrh finálnej dosky sa odvíjal od možností ktoré som mohol odskúšať na testovacích doskách alebo na nepájivom poli. Doska vznikla podstate spojením testovacích dosiek a pridaním ďalších periférií. Celková schéma zapojenia je vyobrazená na prílohe C.1. Plošný spoj je navrhnutý ako dvojvrstvový. Náhľad týchto vrstiev je možný v prílohe a to na obr C.3 a obr C.2. Obsahuje 4 prevodníky napäťovej úrovne. Jeden z nich je určený pre rozhranie 1-WIRE. Pre toto rozhranie je zvolené teplotné čidlo DS18B20 popísané v práci bližšie. Súčasťou dosky je aj prevodník pre zbernicu I2C. Ide presnejšie o obvod PCA9306 popísaný v teoretickej časti práce. Následne sa rozhranie I2C rozdvojuje do dvoch rôznych zariadení. Jedným z týchto zariadení je teplotný senzor LM92. Druhým obvodom na zbernici je expandér MCP23017. Expandér je 16 bitový. Šesť pinov z expandéru je využitých na obsluhu LCD displeja. Displej obsahuje radič HD44780 tiež popísaný v práci. Osem pinov je použitých ako príprava pre maticovú klávesnicu. Zvyšné zbernice sú vyvedené do konektorov presnejšie ide o zbernicu SPI 3, SPI 4 a UART. Doska ako posledné obsahuje lineárny napäťový stabilizátor. Dimenzovaný na prúd 1000 mA, ten by mal byť dostatočný na napájanie všetkých obvodov a Periférií. V prílohe je vyobrazená predloha pre osádzanie súčiastok viz. C.4, C.5.

3 ZÁVER

Jednodoskové systémy využívajúce procesory architektúry ARM sú v dnešnej dobre veľmi populárnou záležitosťou. Majú obrovský rozsah použitia od prehrávačov k televízorom až po rôzne aplikácie v robotike. Obrovský bum nastal z týmito zariadeniami pár rokov do zadu, kedy táto architektúra procesorov zlacnela na dostupnú úroveň a ich výkon sa vyrovnal skoro na hranicu výkonu stolových počítacou. Pri tom ale zostala cena veľmi nízko oproti stolovým počítačom. Majú mnoho nesporných výhod napríklad nízka spotreba, pasívne chladenie, malé rozmery a iné. Takýchto zariadení sa vyrába v dnešnej dobe veľké množstvo a líšia sa iba drobnými rozdielmi napríklad rozmermi alebo rôznym počtom GPIO pinov.

V úvode práce bolo potrebné naštudovať samotný jedndoskový počítač Beagleboard Xm. Zistiť všetky jeho dostupné periférie, naštudovať možnosti rozšírenia tohto počítača. Náplňou semestrálneho projektu bolo navrhnúť toto rozšírenie, v tejto fáze nenastali žiadne výrazné problémy. Nasledovala inštalácia operačného systému. Z dostupných operačných systémov bol zvolený Ubuntu verzie 12.10. V tejto fáze nastalo pár problémov z napájacím zdrojom, zariadenie potrebuje okrem dostatočnej prúdovej rezervy aj stabilitu napájacieho zdroja. Pokiaľ táto stabilita nebola zabezpečená stávalo sa že operačný systém nebol spustený a vyhlásil chybu vnútorného prerušenia. Poprípade sa stávalo že operačný systém sa spustil ale nespustili sa všetky periférie, presnejšie išlo o USB hub a sieťový adaptér. Výber vhodného operačného systému taktiež zabral istý čas, pretože bolo potrebné zvoliť operačný systém z jadrom ktoré bude mať zavedené ovládače periférií a to hlavne I2C a SPI. Pre rozhranie SPI som si nakoniec musel preložiť vlastné jadro. Pretože žiadny z dostupných operačných systémov tieto ovládače nemal zavedené v module jadra. V ďalšej fáze prác bolo potrebné odskúšať jednotlivé periféria. Aby som eliminoval možnosť hardvérovej chyby ako je napríklad studený spoj, vyrobil som si na tento účel niekoľko testovacích plošných spojov. Pomocou týchto dosiek som overil činnosť rozhraní SPI, I2C a napríklad aj PWM modulácie. Práca za týmto spôsobom uľahčila a urýchlila. Následne bola navrhnutý finálna doska.

Posledná a finálna fáza pozostávala z osadenia navrhnutej dosky. Súčasťou tejto fáze bolo napísanie programy pre overenie funkčnosti jednotlivých periférií. Táto úloha vola jedna z najťažších a zabrala aj veľké množstvo času. Nastalo mnoho problémov a zaseknutí. Nakoniec sa ale podarilo sfunkčniť väčšinu požadovaných periférií. Jediným problematickým rozhraním je AD prevodník, ale na oprave sa stále pracuje. Zriadenie je určené na štúdijné účeli a prezentáciu jednotlivých periférií. Výkon zariadenia je dostatočný. V prípade zložitejších aplikácií už asi nebude výkon Beagleboardu dostatočný. Bolo by potrebné zvoliť niektorú z výkonnejších alternatív podobného zariadenia.

LITERATÚRA

[1] HRBÁČEK, J. *Komunikace mikrokontroléru z okolím - 1. díl*. BEN technická literatúra, Praha 1999. ISBN 80-86056-42-2.

[2] HRBÁČEK, J. *Komunikace mikrokontroléru z okolím - 2. díl*. BEN technická literatúra, Praha 2000. ISBN 80-86056-73-2.

[3] *Technical documentation about DM3730, DM3725 Digital Media Processors* [online]. 2010, posledná aktualizácia 1. 7. 2011 [cit. 20. 10. 2013]. Dostupné z URL: <http://www.ti.com/product/dm3730>.

[4] *Technical documentation about LM92* [online]. 2010, posledná aktualizácia 22. 3. 2013 [cit. 20. 4. 2014]. Dostupné z URL: <http://www.ti.com/product/LM92/technicaldocuments>.

[5] *Texas Instruments* výrobca procesora Omap4460 [online]. 2013, posledná aktualizácia 1. 11. 2013 [cit.4.12.2013]. Dostupné z URL: <http://www.ti.com/product/omap4460>.

[6] *Texas Instruments* výrobca AD a DA prevodníku AMC7823 [online]. 2012, posledná aktualizácia 20. 3. 2012 [cit.4.12.2013]. Dostupné z URL: <http://www.ti.com/product/amc7823>.

[7] *BeagleBoard. In Wikipedia: the free encyclopedia* [online]. Wikipedia Foundations, poslední aktualizace 2. 10. 2012 [cit. 20. 10. 2013]. Dostupné z URL: <http://www.en.wikipedia.org/wiki/BeagleBoard>.

[8] *BeagleBoard a SPI* [online]. 2013, posledná aktualizácia 20. 1. 2013 [cit. 20. 5. 2014]. Dostupné z URL: <http://www.elinux.org/BeagleBoard/SPI>.

[9] *Povolenie SPI na Beagleboarde* [online]. 2013, posledná aktualizácia 7. 10. 2011 [cit. 20. 5. 2014]. Dostupné z URL: <http://linuxdeveloper.blogspot.cz/2011/10/enabling-spi-on-beagleboard-xm.html>.

[10] *BeagleBoard a PinMUX* [online]. 2013, posledná aktualizácia 2. 1. 2014 [cit. 20. 5. 2014]. Dostupné z URL: <http://www.elinux.org/BeagleBoardPinMux>.

[11] *BeagleBoard a PWM* [online]. 2013, posledná aktualizácia 18. 5. 2013 [cit. 20. 5. 2014]. Dostupné z URL: <http://www.elinux.org/BeagleBoardPWM>.

[12] *BeagleBoard a I2C* [online]. 2013, posledná aktualizácia 8. 10. 2012 [cit. 20. 5. 2014]. Dostupné z URL: <http://www.elinux.org/Interfacing_with_I2C_Devices>.

[13] *Raspberry Pi. In Wikipedia: the free encyclopedia* [online]. Wikipedia Foundations, poslední aktualizace 25. 9. 2012 [cit. 20. 10. 2013]. Dostupné z URL: <http://www.cs.wikipedia.org/wiki/Raspberry_Pi>.

[14] *Technical documentation about DS18B20* [online]. Technical documentation, poslední aktualizace 18. 3. 2009 [cit. 20. 11. 2013]. Dostupné z URL: <http://www.maximintegrated.com/app-notes/index.mvp/id/4377>.

[15] STROLENÝ, J. *Znakové LCD displeje* [online]. 2010, posledná aktualizácia 25. 2. 2012 [cit. 4. 11. 2013]. Dostupné z URL: <http://doveda.byl.cz/lcd/index.htm>.

[16] CHYTIL, J. *Maticová klávesnice* [online]. 2010, posledná aktualizácia 11. 9. 2007 [cit. 4. 11. 2013]. Dostupné z URL: <http://www.8bitu.cz/clanek/serial-pic16f84-14-maticova-klavesnice/>.

[17] *Ubuntu* distribucia OS Linux [online]. 2013, posledná aktualizácia 1. 11. 2012 [cit.4.12.2013]. Dostupné z URL: <http://www.ubuntu.com>.

[18] *Raspberry-Pi* mikropočítač [online]. 2013, posledná aktualizácia 28. 5. 2013 [cit.4.12.2013]. Dostupné z URL: <http://www.raspberrypi.org>.

[19] *Raspberry-Pi* príklady použitia arm počítača [online]. 2013, posledná aktualizácia 18. 5. 2014 [cit.20.5.2014]. Dostupné z URL: <http://www.astromik.org/malymenu/menuraspi.htm>.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

ARM procesory s redukovanou inštrukčnou sadou – Acorn RISC Machine

SoC integrované komponenty na jednom čipe – System on chip

RS232 sériová linka – Serial comunication

RS485 sériová linka – Serial comunication

I2C multi-masterová počítačová sériová zbernica – Inter-Integrated Circuit

GPIO všeobecný vstupno výstupný pin – General-purpose input/output

SRAM statická pamäť pre čítanie a zápis – Shadow Random Access Memory

ROM elektronická pamäť iba na čítanie – Read-Only Memory

JTAG štandard na testovanie plošných spojov – Joint Test Action Group

LED dióda emitujúca svetlo – Light-Emitting Diode

LCD displej z tekutých kryštálov – Liquid crystal display

MDDR synchrónna dynamická operačná pamäť – Synchronous dynamic system memory

USB univerzálna sérová zbernica – Universal Serial Bus

SMD technológiu povrchovej montáže súčiastok – Surface Mount Devices

PWM impulzová šírková modulácia – Pulse Width Modulation

SDA dátový kanál – Serial Data Line

SCL hodinový kanál – Serial Clock Line

LSB najmenej významný bit – Least Significant Bit

MSB najvýznamnejší bit – Most Significant Bit

ZOZNAM PRÍLOH

A Tabuľka parametrov **57**

B Testovacia dosky **58**

C Finálna dosky **61**

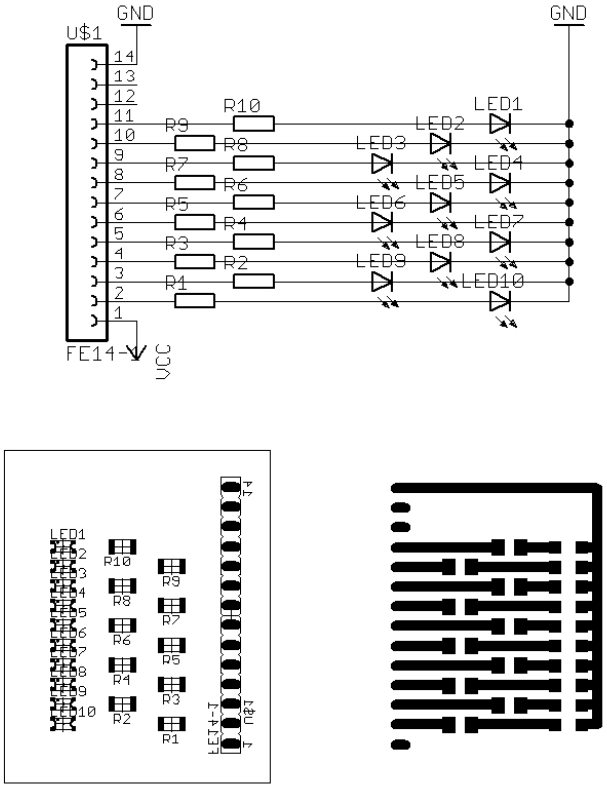
D Tabuľka rozširujúceho konektora **65**

A TABUĽKA PARAMETROV

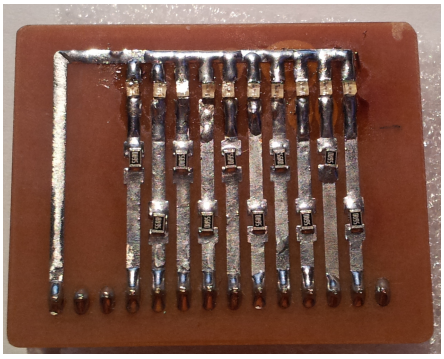
Tab. A.1: Prehľadná tabuľka parametrov

Model	BeagleBoard-xM	BeagleBone	BeagleBone Black
CPU	1GHz ARM, Cortex-A8	720MHz ARM, Cortex-A8	1GHz ARM, Cortex-A8
Označenie chipu	TI DM3730	TI AM3358/9	TI AM3359
GPU	TMS320C64x+	PowerVR SGX530	PowerVR SGX530
Pamäť	512 MB DDR2	256 MB DDR2	512 MB DDR3
Video výstup	DVI-D, S-Video, HDMI	-	microHDMI
Audio výstup	3.5 mm stereo jack	-	microHDMI
Ostatné periférie	McBSP, DSS, I2C, UART, LCD, McSPI, PWM, JTAG, Camera Interface	4x UART, 8x PWM, LCD, GPMC, MMC1, 2x SPI, 2x I2C, A/D Converter, 2xCAN Bus, 4 Timers, FTDI USB to Serial, JTAG via USB	4x UART, 8x PWM, LCD, GPMC, MMC1, 2x SPI, 2x I2C, A/D Converter, 2xCAN Bus, 4 Timers
Veľkosť	76.2 x 76.2 mm	86.4mm x 53.3mm	86.4mm x 53.3mm
Váha	37 g	30 g	30 g
Uvedenie na trh	14.9.2010	31.11.2011	23.4.2013
Štartovacia cena	120 €	70 €	40 €

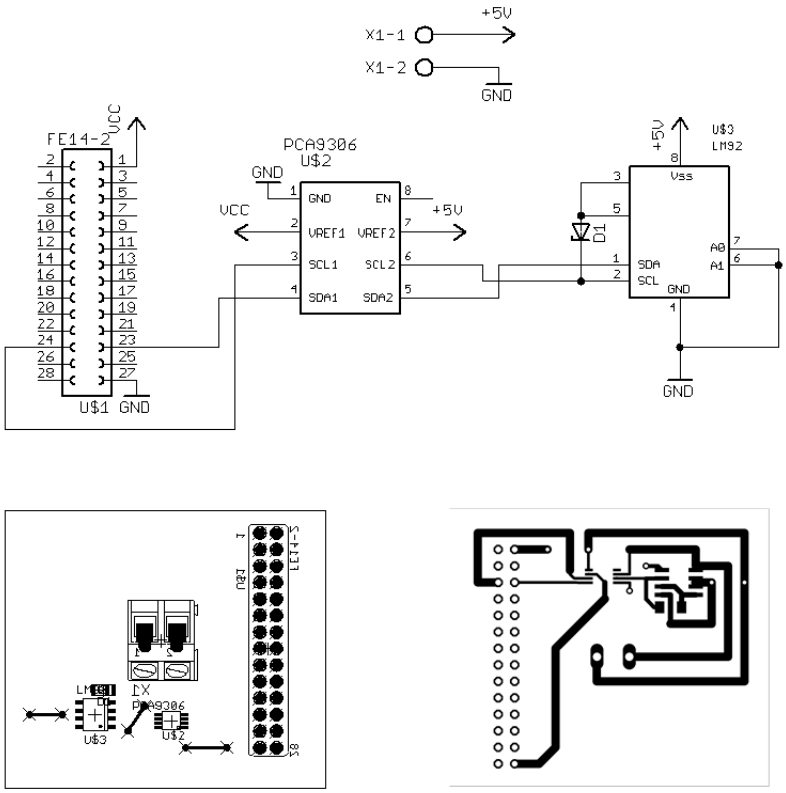
B TESTOVACIA DOSKY



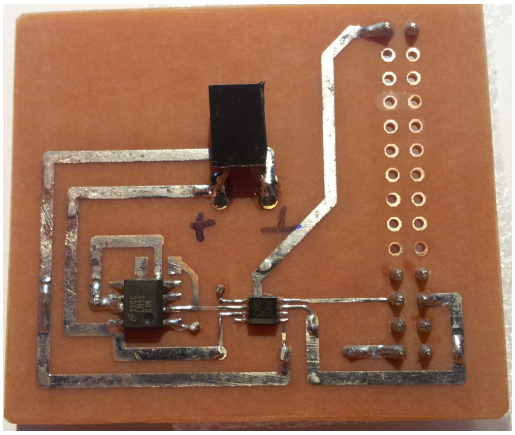
Obr. B.1: Testovacia doska z diódami.



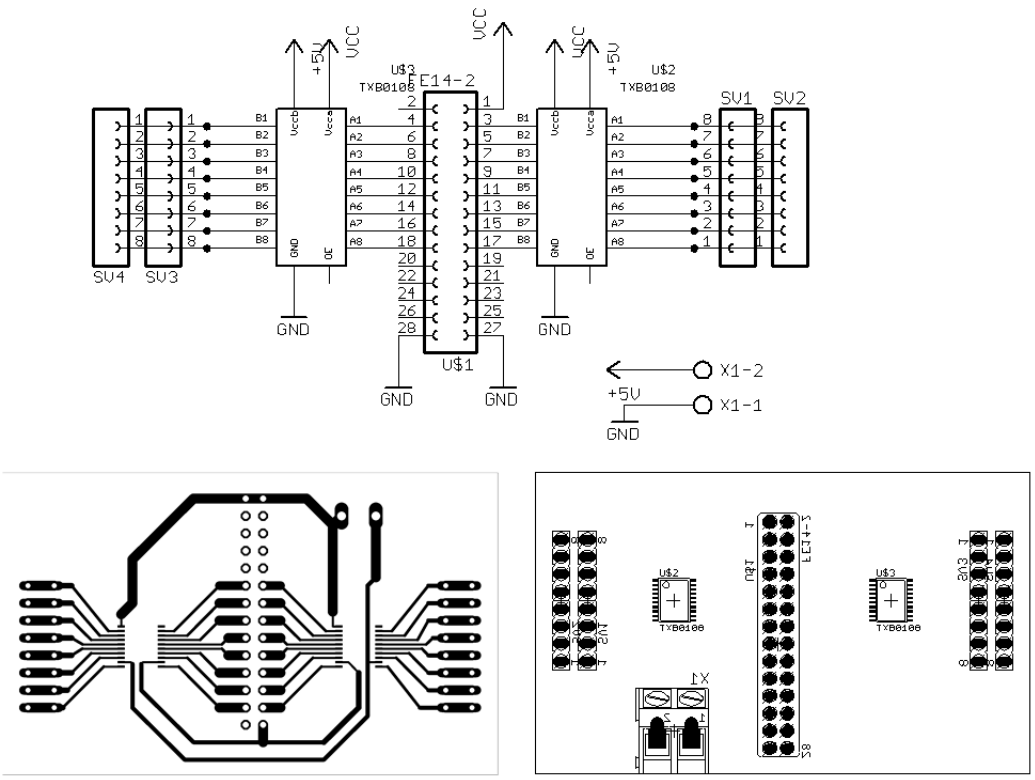
Obr. B.2: Testovacia doska z diódami osadená.



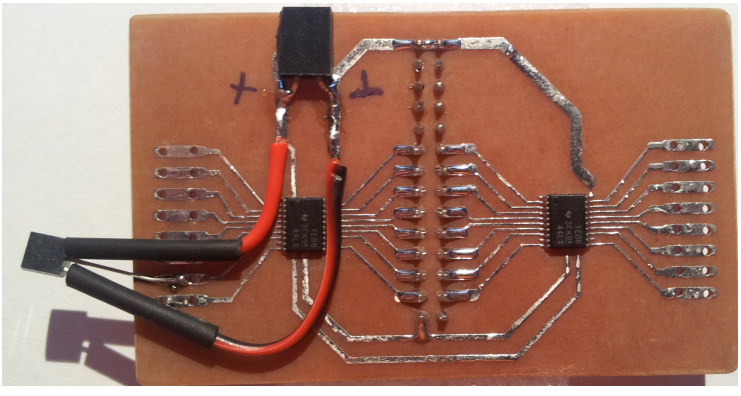
Obr. B.3: Testovacia doska z LM92.



Obr. B.4: Testovacia doska z LM92 osadená.

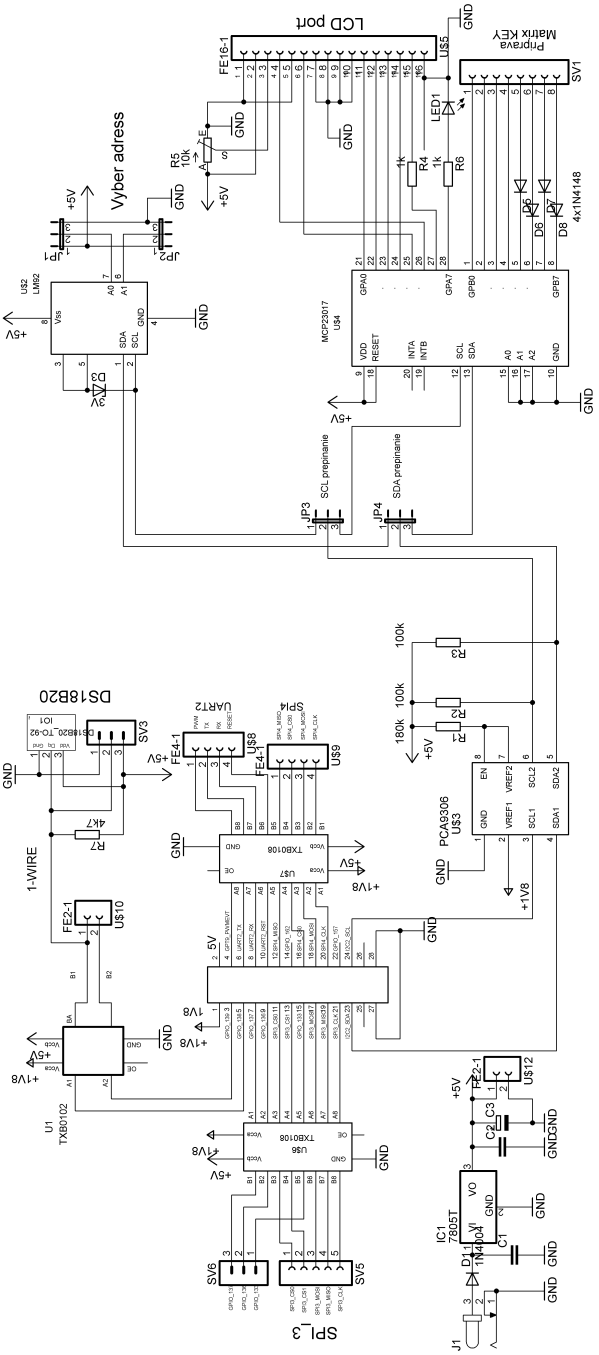


Obr. B.5: Testovacia doska univerzálna.

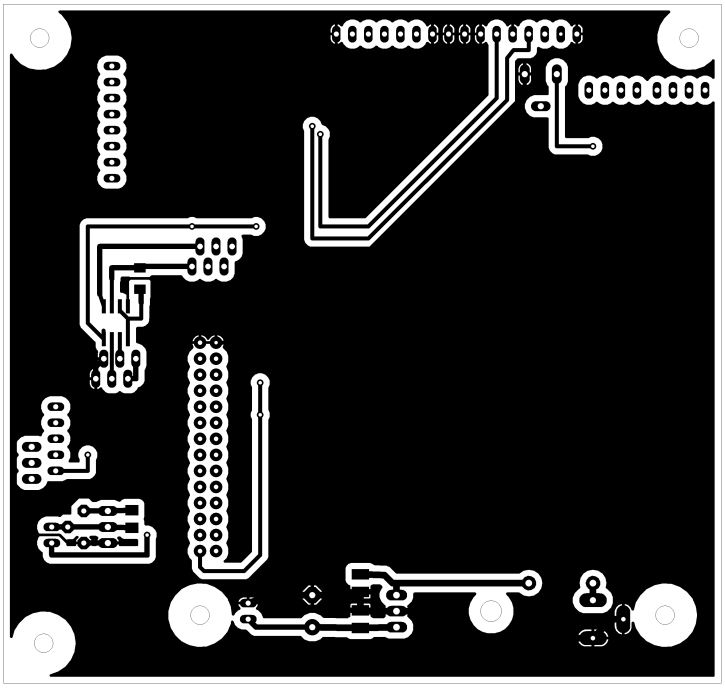


Obr. B.6: Testovacia doska univerzálna osadená.

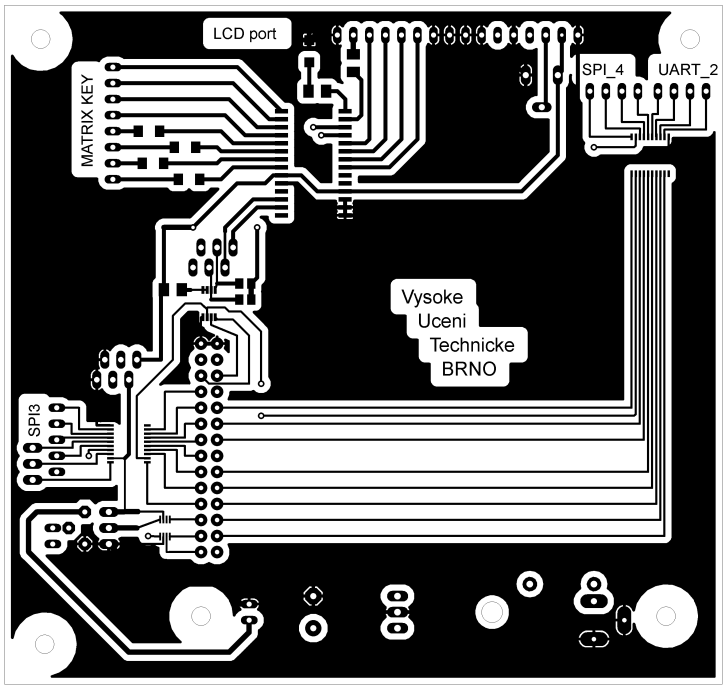
C FINÁLNA DOSKY



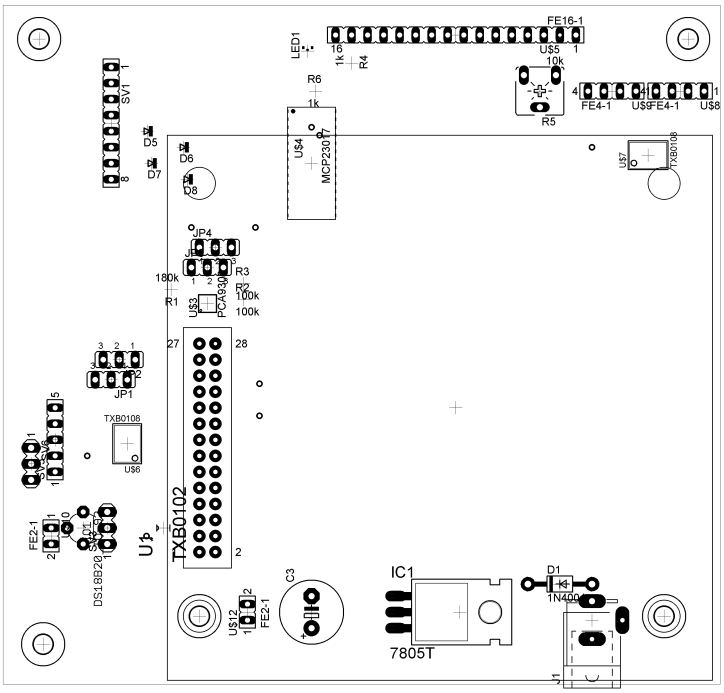
Obr. C.1: Schéma finálne dosky.



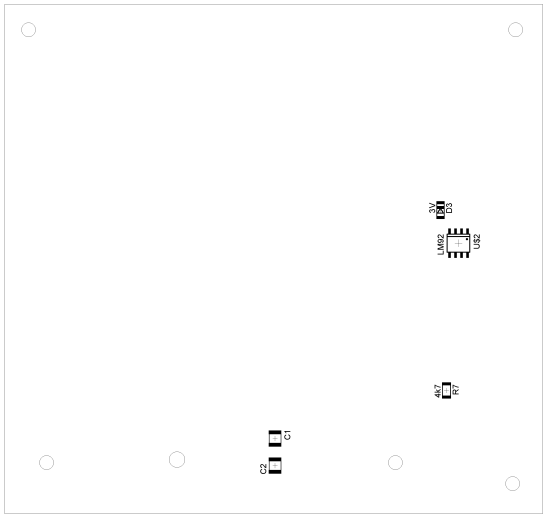
Obr. C.2: Plošný spoj zo spodnej strany.



Obr. C.3: Plošný spoj z vrchnej strany.



Obr. C.4: Rozvrhnutie súčiastok zo spodnej strany.



Obr. C.5: Rozvrhnutie súčiastok z vrchnej strany.

Partlist

EAGLE Version 5.6.0 Copyright (c) 1988-2009 CadSoft

Part	Value	Device	Package	Sheet
C1		C-EUC1210	C1210	1
C2		C-EUC1210	C1210	1
C3		CPOL-EUE5-10.5	E5-10,5	1
D1	1N4004	1N4004	DO41-10	1
D3	3V	ZENER-DIODE	SOD80C	1
D5		DIODE-MINIMELF	MINIMELF	1
D6		DIODE-MINIMELF	MINIMELF	1
D7		DIODE-MINIMELF	MINIMELF	1
D8		DIODE-MINIMELF	MINIMELF	1
IC1	7805T	7805T	TO220H	1
IO1	DS18B20	DS18B20_TO-92	TO-92	1
J1		JACK-PLUG0	SPC4077	1
JP1		JP2E	JP2	1
JP2		JP2E	JP2	1
JP3		JP2E	JP2	1
JP4		JP2E	JP2	1
LED1		LEDCHIPLED_1206	CHIPLED_1206	1
R1	180k	R-EU_M1206	M1206	1
R2	100k	R-EU_M0805	M0805	1
R3	100k	R-EU_M0805	M0805	1
R4	1k	R-EU_M1206	M1206	1
R5	10k	TRIM_EU-CA6V	CA6V	1
R6	1k	R-EU_M1206	M1206	1
R7	4k7	R-EU_M1206	M1206	1
U$2	LM92	LM92	SOIC-8	1
U$3	PCA9306	PCA9306	S-PDSO-G8	1
U$4	MCP23017	MCP23017	SOIC-28	1
U$5	FE16-1	FE16-1	FE16-1	1
U$6	TXB0108	TXB0108	TSSOP20	1
U$7	TXB0108	TXB0108	TSSOP20	1
U$8	FE4-1	FE4-1	FE4	1
U$9	FE4-1	FE4-1	FE4	1
U$10	FE2-1	FE2-1	FE2	1
U$11	AMC7823	AMC7823	40-WQFN	1
U$12	FE2-1	FE2-1	FE2	1
U1	TXB0102	TXB0102DCUR	SOP50P310X90-8N	1

Obr. C.6: Zoznam použitých súčiastok.

D TABUĽKA ROZŠIRUJÚCEHO KONEKTORA

Tab. D.1: Tabuľka módou rozširujúceho konektora

PIN	Proc	0	1	2	3	4
1		Vcc $+1,8\,V$				
2		Vcc $+5\,V$				
3	AE39	MMC2_DAT7	*	*	*	GPIO_139
4	AB26	UART2_CTS	McBSP3_DX	GPT9_PWMEVT	X	GPIO_144
5	AF3	MMC2_DAT6	*	*	*	GPIO_138
6	AA25	UART2_TX	McBSP3_CLKX	GPT11_PWMEVT	X	GPIO_146
7	AH3	MMC2_DAT5	*	*	*	GPIO_137
8	AE5	McBSP3_FSX	UART2_RX	X	X	GPIO_143
9	AE4	MMC2_DAT4	*	X	*	GPIO_136
10	AB25	UART2_RTS	McBSP3_DR	GPT10_PWMEVT	X	GPIO_145
11	AF4	MMC2_DAT3	McSPI3_CS0	X	X	GPIO_135
12	V21	McBSP1_DX	McSPI4_SIMO	McBSP3_DX	X	GPIO_158
13	AG4	MMC2_DAT2	McSPI3_CS1	X	X	GPIO_134
14	W21	McBSP1_CLKX	McBSP3_CLK	X	X	GPIO_162
15	AH4	MMC2_DAT1	X	X	X	GPIO_133
16	K26	McBSP1_FSX	McSPI4_CS0	McBSP3_FSX	X	GPIO_161
17	AH5	MMC2_DAT0	McSPI3_SOMI	X	X	GPIO_132
18	U21	McBSP1_DR	McSPI4_SOMI	McBSP3_DR	X	GPIO_159
19	AG5	MMC2_CMD	McSPI3_SIMO	X	X	GPIO_131
20	Y21	McBSP1_CLKR	McSPI4_CLK	X	X	GPIO_156
21	AE2	MMC2_CLKO	McSPI3_CLK	X	X	GPIO_130
22	AA21	McBSP1_FSR	X	*	Z	GPIO_157
23	AE15	I2C2_SDA	X	X	X	GPIO_183
24	AF15	I2C2_SCL	X	X	X	GPIO_168
25		Regen				
26		Reset				
27		GND				
28		GND				