

Západočeská univerzita v Plzni

Fakulta elektrotechnická

**KATEDRA APLIKOVANÉ ELEKTROTECHNIKY A
TELEKOMUNIKACÍ**

DIPLOMOVÁ PRÁCE

Inovace řídicí jednotky RCX

Autor práce: Bc. Rostislav Jandejsek

Vedoucí práce: Ing. Petr Weissar, Ph.D.

PLZEŇ 2013

Anotace diplomové práce

Jandejsek, R. - Inovace elektronické řídicí jednotky RCX.

Katedra aplikované elektroniky a telekomunikací, Západočeská univerzita v Plzni –
Fakulta elektrotechnická, 2013, 43 s., vedoucí: Ing. Petr Weissar, Ph.D.

Práce se zabývá popisem elektronické jednotky RCX a návrhem nové jednotky, která bude kompatibilní s původním příslušenstvím. Návrh schématu a plošného spoje je realizován v návrhovém systému Altium Designer. Software je psaný v programu KEIL μ Vision.

Klíčová slova:

LEGO, RCX, CMSIS, C, STM32, Altium Designer

Abstract

Jandjsek, R. - RCX control unit redesign.

Department of applied electronics and telecommunication, University of West Bohemia in Pilsen – Faculty of electrical engineering, 2013, 43 p., head: Ing. Petr Weissar, Ph.D.

Thesis deals with describing the RCX unit and its innovation. Innovated unit should be compatible with accessories of original unit. Hardware is designed in the Altium Designer. Design of software is done in the KEIL μ Vision in C language.

Keywords:

LEGO, RCX, CMSIS, C, STM32, Altium Designer

Prohlášení:

Předkládám tímto k posouzení a obhajobě diplomovou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.

v Plzni, dne

.....

Podpis diplomanta

Poděkování

Rád bych poděkoval svému vedoucímu Ing. Petru Weissarovi, Ph.D., za jeho ochotu, cenné rady a vzorné vedení práce. Dále děkuji Ing. Vladimíru Pavlíčkovi, Ph.D., a Ing. Kamilu Kosturikovi, Ph.D., za pomoc při řešení práce. Na závěr děkuji Katedře aplikované elektroniky, jejíž laboratoře jsem mohl využívat.

Obsah

1. Úvod.....	1
2. Rozbor původní koncepce jednotky	2
2.1. Uživatelské rozhraní RCX.....	2
2.2. Napájení.....	2
2.3. Senzory RXC.....	3
2.3.1. Pasivní senzory.....	4
2.3.2. Senzor otáček	5
2.3.3. Světelný senzor	6
2.4. Výstupy.....	7
2.5. Programování jednotky	8
2.5.1. RCX code	8
2.5.2. RoboLab	9
2.5.3. BrickOS	11
3. Návrh nové jednotky	12
3.1. Výběr mikroprocesoru.....	12
3.1.1. Texas Instruments	12
3.1.2. Atmel.....	13
3.1.3. STMicroelectronics	14
3.2. Uživatelské rozhraní	17
3.3. Napájení.....	19
3.4. Rozhraní senzorů	21
3.5. IR komunikace.....	22
3.6. Obsluha senzorů	24
3.7. Programování jednotky	26
3.7.1. CMSIS.....	26
3.7.2. Maple IDE.....	28
3.8. Návrh DPS.....	30
4. Závěr	31

1. Úvod

Jednotka RCX je součástí stavebnice LEGO Mindstorms. RCX je řídicí jednotka, která je pomocí svých vstupů a výstupu schopna ovládat například robota z LEGO dílů. Stavebnice je určena zejména pro nejmladší začínající programátory. Tomu jsou přizpůsobeny i nástroje pro vývoj softwaru, které se snaží naučit základní programátorské návyky jednoduchou formou. Software je nahráván pomocí tzv. *IR věže*, která se do počítače připojuje pomocí USB. Jednotka již není podporována výrobcem, protože se vyrábí nová verze.

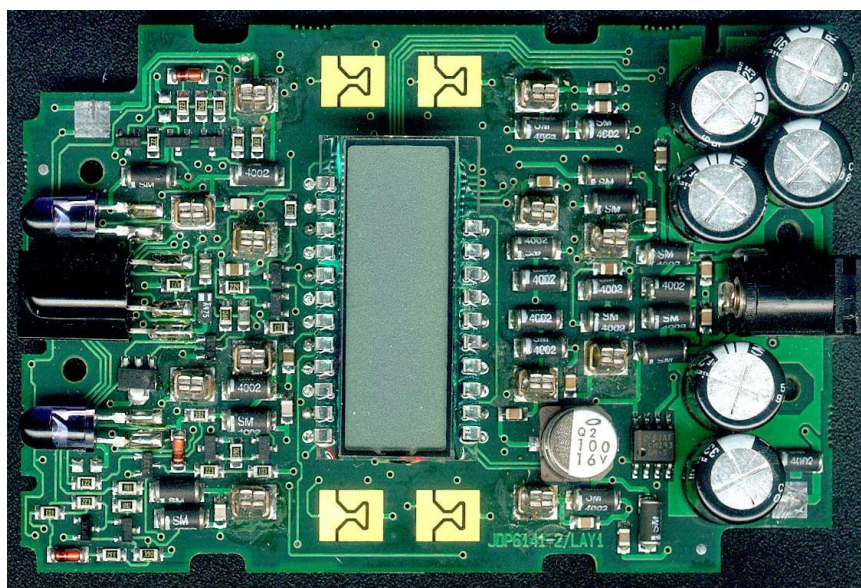
Cílem práce je navrhnout novou koncepci jednotky se zachováním kompatibility s původním příslušenstvím. Nová jednotka by měla být programovatelná v jazyce C nebo Wiring¹. Dále by měla být schopná komunikovat s novou jednotkou LEGO. Obsahem práce je nejprve rozbor původní jednotky včetně příslušenství. Na základě toho provést návrh nového schématu a DPS nové jednotky.

Práci jsem si vybral, protože mě zaujala možnost navrhovat hardware, u kterého byly předem dané rozměry DPS, pozice konektorů a displeje, vstupy a výstupy atd. Také jsem měl možnost se seznámit s různými mikropočítači a vybrat ten, který se nejlépe hodí pro tuto aplikaci.

¹ Variace jazyka C pro Arduino (otevřená platforma založena na MCU Atmega)

2. Rozbor původní koncepce jednotky

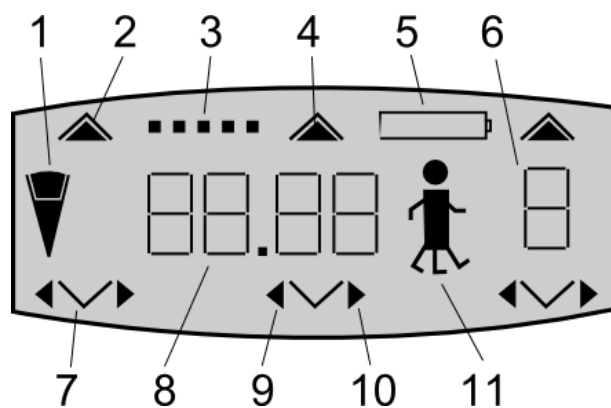
Aby se dala původní jednotka inovovat, bylo třeba se podrobně seznámit se vstupy, výstupy, uživatelským rozhraním, mechanickou konstrukcí a dalšími důležitými parametry. V následujících kapitolách jsou popsány vlastnosti jednotky a příslušenství.



Obr. 1 DPS inovované jednotky [1]

2.1. Uživatelské rozhraní RCX

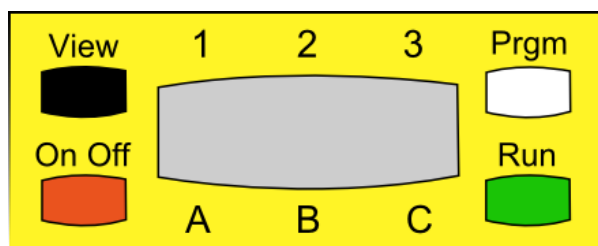
K zobrazování údajů o aktuálním stavu řídicí jednotky je použit monochromatický LCD. Jednotlivé segmenty jsou vyrobeny pevně, k zobrazení nestandardní informace může uživatel použít pouze 4 sedmisegmentové číslice. Na následujícím obrázku (Obr. 2) jsou vidět všechny segmenty.



Obr. 2 Schéma displeje

- | | |
|--|--|
| 1. IrDA aktivní / dosah vysílače | 7. Vybrán vstupní port (použit v programu) |
| 2. Vybrán vstupní port (použit v programu) | 8. Zobrazení měřené veličiny nebo doba běhu programu |
| 3. Ukazatel stavu programování jednotky | 9. Výstupní port ve zpětném režimu |
| 4. Vstupní port aktivní | 10. Výstupní port v dopředném režimu |
| 5. Ukazatel stavu nabití akumulátorů | 11. Stav programu – spuštěný/zastavený |
| 6. Číslo aktivního programu | |

Vedle displeje jsou umístěna 4 tlačítka.

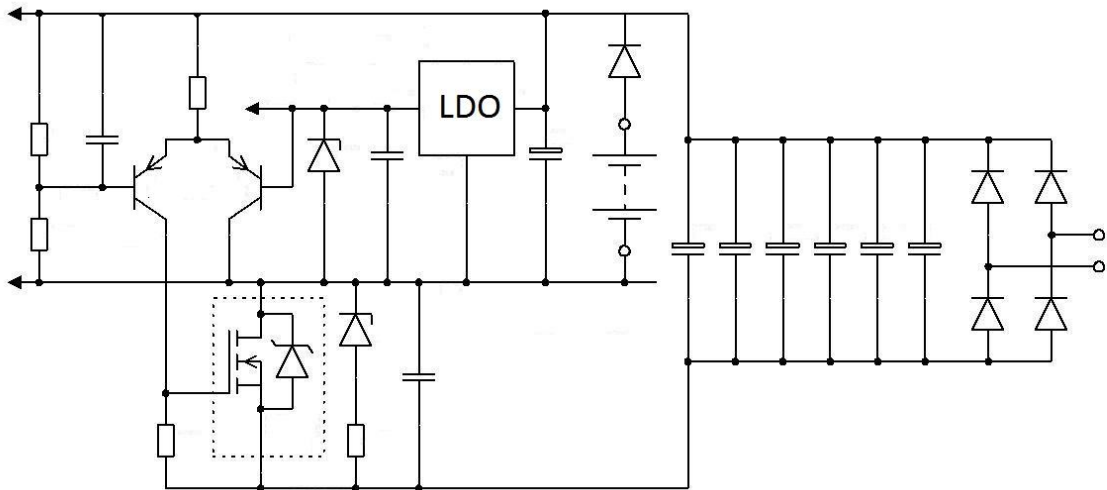


Obr. 3 Rozložení tlačítek

- View přepíná zobrazování informace o vstupech, výstupech a systémového času
- Prgm přepíná mezi programy
- Run pozastavuje a spouští vybraný program
- On Off zapíná/vypíná jednotku

2.2. Napájení

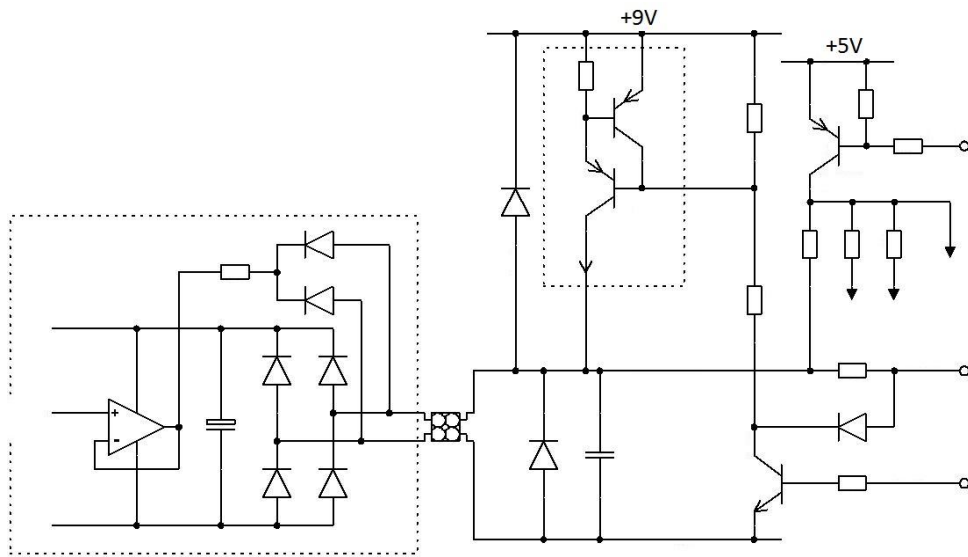
Jednotka RCX je napájena bateriově nebo ze zdroje střídavého napětí. Bateriové napájení je zajištěno šesti akumulátory zapojených do série s diodou. Ta je zde zapojena kvůli ochraně proti přepólování a nabíjení akumulátorů při připojení adaptéru. Vstupní střídavé napětí je usměrněno a vyhlazeno šesti kondenzátory s kapacitou $330\ \mu\text{F}$. Napájení logických obvodů je zajištěno LDO regulátorem napětí.



Obr. 4 Schéma napájení [2]

2.3. Senzory RXC

Jednotka může využívat různé senzory. Základní rozdělení senzorů je na pasivní a aktivní. Pasivní senzory jsou dotykové a teplotní. Mezi aktivní patří světelný a rotační. Jednotlivé senzory budou popsány podrobněji dále. Následující schéma zobrazuje rozhraní pro připojení senzorů.



Obr. 5 Rozhraní senzorů [2]

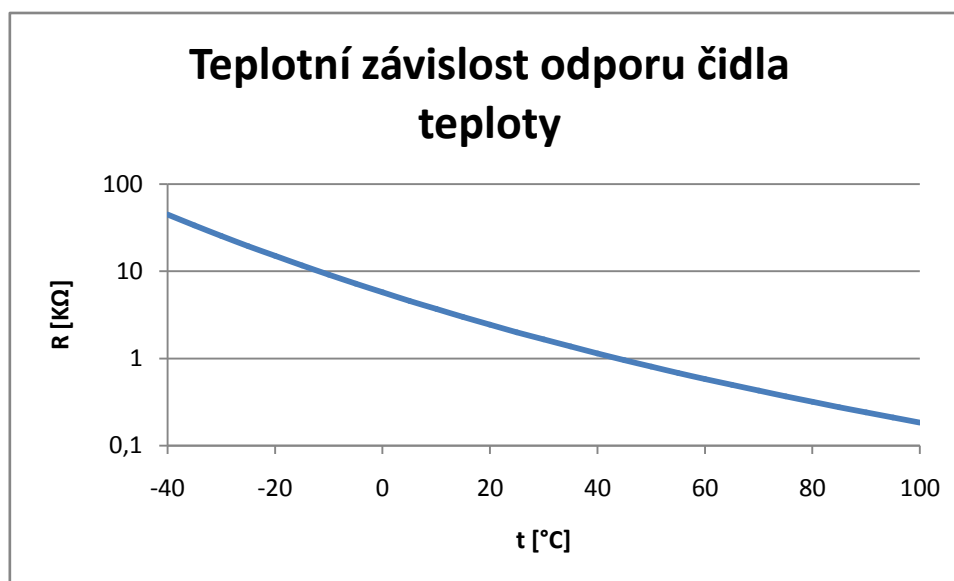
Senzor je napájen pomocí proudového zdroje, který omezuje výstupní proud na 15 mA a tím je zajištěna ochrana proti zkratu. Konektor je však vyroben tak, aby při běžném používání ke zkratu dojít nemohlo. Proudový zdroj se zapíná v případě, že je připojen aktivní senzor (nastavení se musí provést programově). V době vypnutí je připojován pull-up rezistor, který na výstupu udržuje napětí 5 V. Výstup je chráněn dvěma omezujícími diodami. Aktivní senzory mají na vstupu graetzův můstek, díky kterému nezáleží na směru připojení konektoru.

V případě zapojení pasivních senzorů vznikne měřené napětí na odporovém děliči, který tvoří pull-up rezistor spolu s vnitřním odporem senzoru. Zde je možné vzorkovat v libovolných časových intervalech. Aktivní senzory využívají konektor jak pro napájení, tak pro vyčtení příslušné analogové hodnoty. Proudový zdroj se sepne a po tuto dobu se nabíjí kondenzátor uvnitř senzoru. Poté se zdroj vypne, připojí se pull-up rezistor, který s proměnným odporem senzoru vytvoří odporový dělič. Po navzorkování se odpojí pull-up rezistor a opět se aktivuje proudový zdroj, tento proces musí být dostatečně krátký, aby nedošlo k vybití kondenzátoru a následnému naměření chybných hodnot. Způsob vzorkování je popsán u jednotlivých senzorů.

2.3.1. Pasivní senzory

Nejjednodušší typ senzoru je dotykový. Uvnitř senzoru je zapojen rezistor v sérii se spínačem, který se při stlačení sepne. Rezistor vytváří napěťový dělič s pull-up rezistorem RCX jednotky a toto napětí je vzorkováno. K dispozici jsou dva senzory s odpory 500Ω a $2,5 \text{ k}\Omega$ v sepnutém stavu.

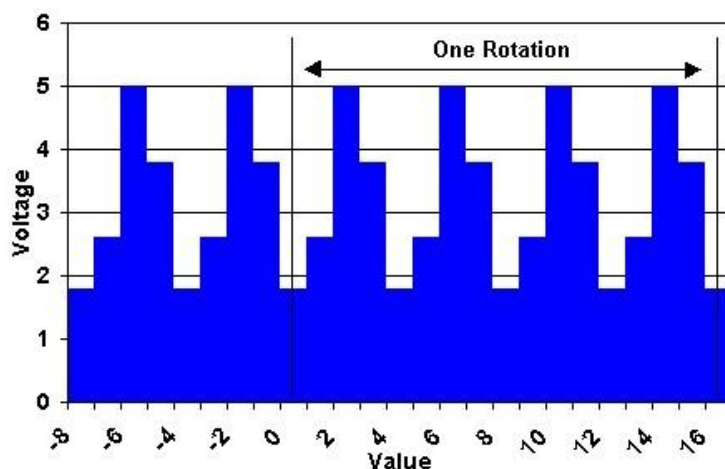
Druhým typem pasivního senzoru je teplotní. Uvnitř senzoru je zapojen NTC termistor. Hodnotu z pasivních senzorů je možno vyčítat kdykoliv. U jednotky RCX je vyčítáno po několika milisekundách.



Obr. 6 Závislost odporu na teplotě

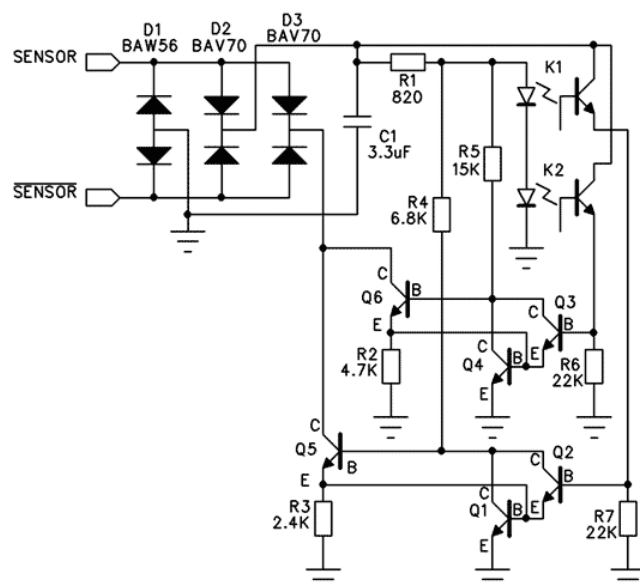
2.3.2. Senzor otáček

Ke snímání rychlosti otáčení se používá rotační senzor. Princip činnosti spočívá ve využití dvou světelných závor tvořených páry IR dioda–fototranzistor. Diody jsou zatmívány lopatkami, které jsou připevněny na osičku senzoru. Protože jsou zde 2 závory a 4 lopatky, je možné rozlišit 16 poloh osičky senzoru a z této informace zjistit i natočení. Jednotlivé úrovně popisuje Obr. 7. Je patrné, že již ze dvou po sobě jdoucích hodnot lze určit směr otáčení.



Obr. 7 Napětí z rotačního senzoru [3]

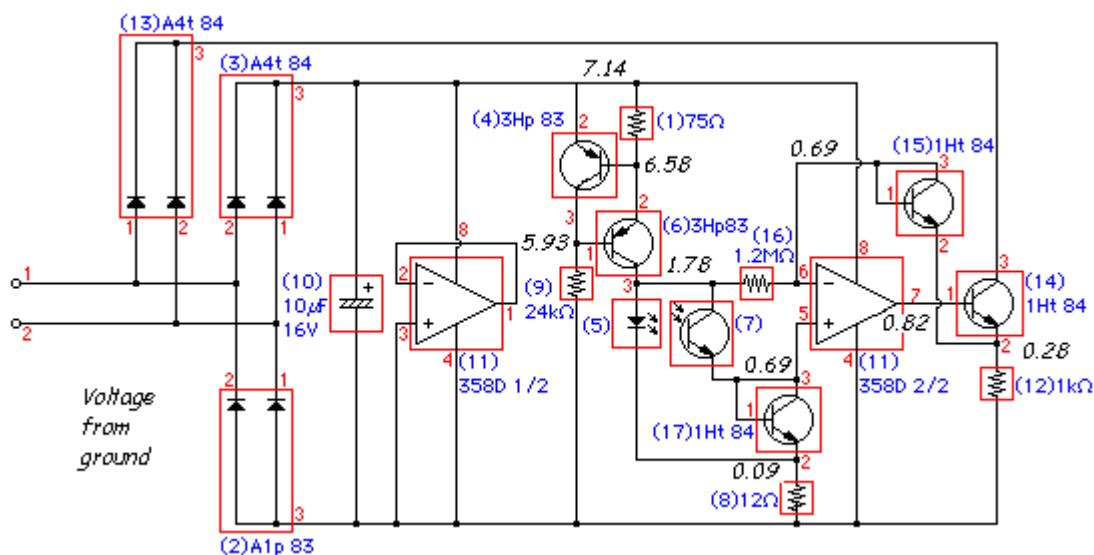
Schéma (Obr. 8) zobrazuje vnitřní zapojení senzoru. V bodě mezi R1 a K1 je udržováno napětí přibližně 2,2 V, které vzniká na IR diodách (napětí se mění v závislosti na stavu nabití baterií). Optickými závory je řízen proud do bází tranzistorů Q3 a Q2 v darlingtonovo zapojení s Q4 a Q1. Jejich výstup řídí báze tranzistorů Q5 a Q6, které mají připojené předpětí 2,2 V. Podle stavu optických závor jsou připojovány rezistory R2 a R3, které tvoří napěťový dělič s pull-up rezistorem v RCX jednotce. U tohoto senzoru je napájení zapnuto po dobu 2,9 ms a vzorkování trvá 100 μ s.



Obr. 8 Vnitřní zapojení senzoru otáček [4]

2.3.3. Světelný senzor

Světelný senzor má dvě základní funkce: snímá intenzitu okolního světla a je schopen sám osvětlovat okolí a snímat odražené světlo. Dá se tedy použít například pro sledování čáry robotem nebo rozpoznání dvou kontrastních barev. To lze využít při rozpoznání lego kostek (černá/žlutá) nebo pro měření rychlosti pomocí dvoubarevného rotujícího kotoučku. Zapojení senzoru zobrazuje Obr. 9.



Obr. 9 Vnitřní zapojení světelného senzoru [5]

Vstupní napětí je „usměrněno“ graetzovým můstkem, z kterého je napájen kondenzátor (10). Jako u předchozího senzoru zajišťuje napájení v době čtení analogové hodnoty. První operační zesilovač (11 1/2), který je v pouzdře dvakrát, je nevyužit. Součástky (1, 4, 6, 9) tvoří proudový zdroj, který napájí LED (5) a fototranzistor (7) proudem 7,5 mA. Tranzistor (17) je zapojen jako dioda a slouží ke stabilizaci pracovního bodu fototranzistoru. Operační zesilovač (11 2/2) je zapojen jako logaritmický zesilovač, který linearizuje charakteristiku fototranzistoru. Odpor tranzistoru (14), který je buzen operačním zesilovačem, je úměrný osvětlení fototranzistoru. Společně s pull-up rezistorem jednotky RCX se vytvoří napěťový dělič, z kterého se čte výsledná měřená veličina. Princip čtení je obdobný jako u senzoru otáček.

2.4. Výstupy

K jednotce RCX lze připojit řadu zařízení. Z originálního příslušenství jsou to motory, světelné prvky a zvukové měniče. Výstup je buzen obvodem LMX10402, který je určen pro buzení DC motorů a LED. Obvod je ovládán dvěma vstupy podle logiky uvedené v tabulce.

IN1	IN2	M1	M2	Režim výstupu
1	0	1	0	dopředný
0	1	0	1	zpětný
1	1	0	0	brzdový
0	0	Z	Z	plovoucí

Tab. 1 Ovládání budičů motorů

Budič je do zátěže schopen dodat proud 500 mA s úbytkem 1,2 V. Obvod je vybaven proudovým omezením a ochranou proti přehřátí. Řízení je realizováno pomocí čítače v mikropočítači s maximální frekvencí 1 kHz.

2.5. Programování jednotky

Jednotka se dá programovat ve více vývojových prostředích. LEGO oficiálně podporovalo 2 nástroje, existuje však několik dalších. Následující kapitoly stručně popisují některé tyto programy.

2.5.1. RCX code

Grafický programovací jazyk, který je dodáván ke stavebnici Robotics Invention System™ (RIS) vznikl za účelem co nejjednoduššího programování RCX. Programuje se pomocí předem vytvořených příkazů (ikon), které se skládají pod sebe. Výsledný program je prováděn postupně odshora dolů. RCX code umožňuje multitasking, takže je možné provádět mimo hlavního bloku i několik dalších souběžně viz Obr. 10. Uvedené bloky jsou určeny pro obsluhování přerušení od senzorů.



Obr. 10 Ukázka programu RCX Code [6]

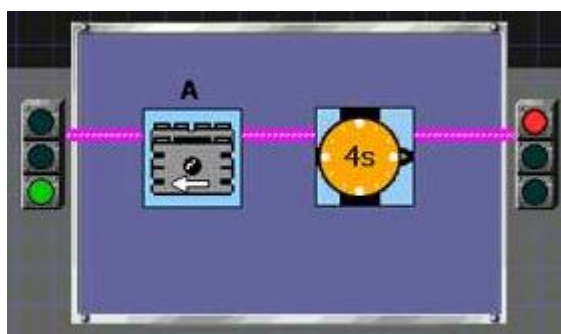
Na obrázku je vidět hlavní blok a 2 bloky přerušení od senzorů dotyku. Jednotlivé příkazy jsou v menu zařazeny do skupin dle významu a barevně označeny. Jsou zde základní programové příkazy jako smyčky, čekání, nebo větvení. Větvit se dá logickými podmínkami jako například údajem ze senzoru tlaku nebo světla.

RCX code není nejúčinnější způsob programování a chybí zde některé funkce klasického C. Program má ovšem základní programové funkce a odpovídá zásadám strukturovaného programování. Je určen pro děti a tímto způsobem se naučí základní programátorské návyky. Pro pokročilé programování se dají využít dále popsané způsoby.

2.5.2. RoboLab

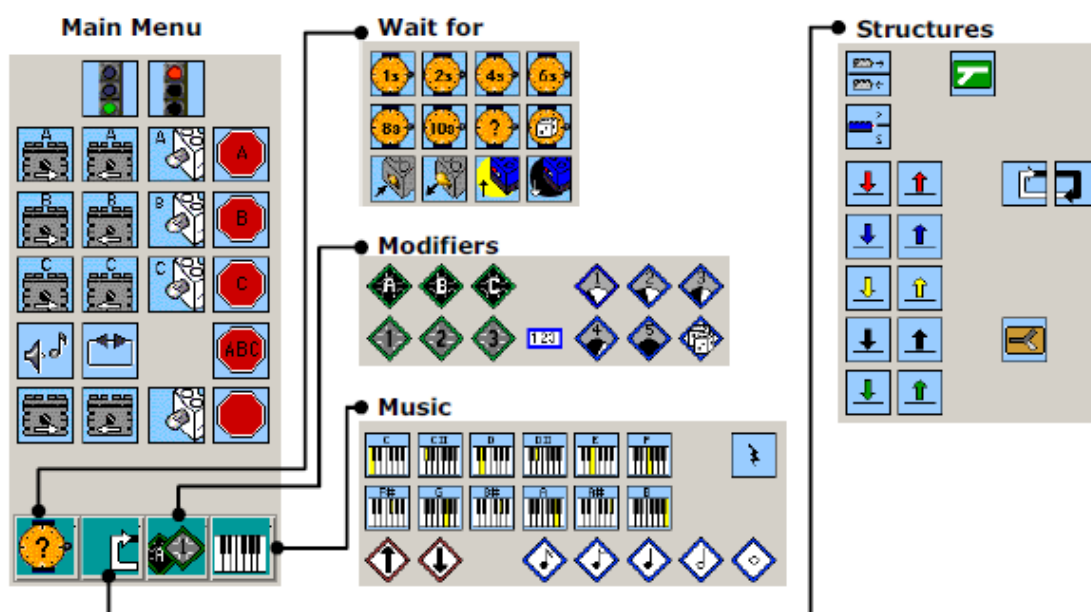
Dalším program s grafickým programováním je RoboLab založený na LabVIEW od firmy National Instruments. Software je přímo zaměřen na výuku programování. Jsou zde dvě úrovně obtížnosti (*Pilot* a *Inventor*), které se dále dělí do čtyř úrovní. Dělení na různé obtížnosti odpovídá postupnému učení začínajícího programátora. Zatímco základní úroveň je založená na příkladech a má omezené možnosti, ve druhé je již možné využít veškeré funkce RoboLabu. V programu lze přehledně programovat i složité funkce.

Na Obr. 11 je vidět první program základní úrovně. Tento program spustí motor připojený na port A po dobu 4 sekund. Pomocí těchto jednoduchých kódů se uživatel postupně naučí používat čekající smyčky, větvení programu nebo čekání na hodnotu ze senzoru.



Obr. 11 Ukázka jednoduchého programu v RoboLab [7]

V úrovni *Inventor* jsou k dispozici panely s příkazy (Obr. 12). Vedle základních programových příkazů jsou k dispozici pokročilé funkce jednotky. RoboLab umožňuje ovládat IR komunikaci, tónový generátor nebo například ovládání jednotky přes internet. Manuály k softwaru RoboLab, kde jsou podrobně popsány všechny funkce, jsou k dispozici na internetových stránkách².



Obr. 12 Příkazová paleta RoboLab [8]

² http://legoengineering.com/library/cat_view/31-programming-references/46-robolab.html

2.5.3. BrickOS

Jedná se o open-source operační systém a vývojové prostředí, které umožňuje programovat jednotku v jazyce C/C++. Původně byl tento nástroj vyvíjen pro Linux, ale je ho možné provozovat i pod systémem Windows.

BrickOS poskytuje alternativní operační systém pro jednotku a ukázkové příklady psané v C a C++. Dále jsou k dispozici nástroje, pomocí kterých se do jednotky nahraje operační systém a následně i zkompilovaný program.



Obr. 13 Ukázka programu BrickOS [9]

3. Návrh nové jednotky

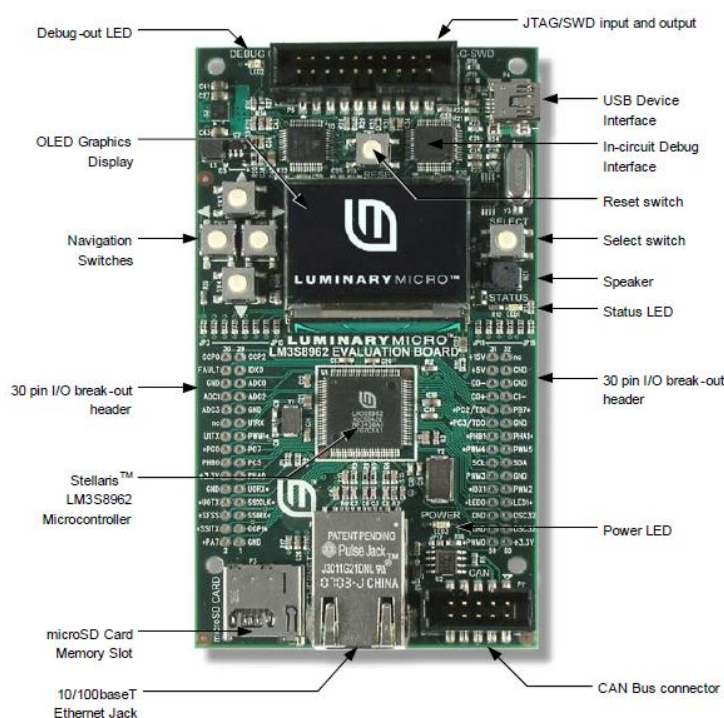
Po prostudování všech periférií a vlastností jednotky RCX jsem mohl začít navrhovat novou jednotku. Vedle inovace jsem se snažil zachovat zpětnou kompatibilitu s původní jednotkou. Následující kapitoly popisují postup návrhu nové jednotky.

3.1. Výběr mikroprocesoru

Vzhledem k tomu, že novější verze jednotky – NXT využívá 32-bit mikroprocesor Cortex-M3, vybíral jsem z řady 32-bit procesorů. Ve škole jsem se setkal s vývojovými kity od výrobců ST, Atmel a TI a vybíral jsem tedy z jejich nabídky. Následující odstavce jsou psány z mého pohledu, na základě vlastních zkušeností.

3.1.1. Texas Instruments

Nejprve jsem vyzkoušel vývojovou desku (Obr. 14) s MCU LM3S9862 od firmy Texas Instruments. Kit je určen především k práci se sběrnici CAN a Ethernet. Tyto periférie jsem nevyužil, ale na kitu se dají vyzkoušet všechny potřebné periférie. Výrobce má na svých stránkách velmi podrobné manuály a instruktážní videa, která umožňují rychlé seznámení s perifériemi a způsobem programování. MCU je z řady Stellaris, pro kterou jsou k dispozici knihovny Stellarisware. Pomocí těchto knihoven lze programovat pokročilejší MCU podobným způsobem jako 8-bit MCU. Výrobce nabízí programovací prostředí Code Composer Studio, které je nástavbou na Eclipse. Díky tomu je možné využít zvyklosti z vyšších programovacích nástrojů, jako například automatické doplňování, snipety, propracovaný debug a různé konfigurace generování výstupů. Nevýhodou je jeho rychlost, protože je psané v interpretovaném jazyce Java.



Obr. 14 Vývojový kit EKS-LM3S8962 [10]

3.1.2. Atmel

Od výrobce Atmel jsem vyzkoušel vývojový kit (Obr. 15) s mikroprocesorem SAM3SD8. Jedná se o 32-bit MCU s jádrem ARM. Zaznamenal jsem mnohem menší podporu než u výrobce Texas Instruments. Z mého pohledu jsou vzorové příklady špatně pochopitelné a začátky programování složité.

Atmel má však velmi dobré vývojové prostředí Atmel Studio. To je založeno na softwarovém nástroji Visual Studio. Zachovává jeho hlavní výhody a dodává navíc další funkce pro vývoj mikropočítačů. Podporovány jsou jak mikropočítače typu AVR, tak i MCU řady ARM Cortex -M. Atmel Studio 6 je poskytováno zdarma s Atmel Software Framework – knihovna obsahující příklady pro ARM i AVR. Dále je k dispozici *Atmel Gallery*, kde lze stáhnout další kompilátory, ladící nástroje, RTOS, komunikační systémy a jiná rozšíření.

Vývojový kit ATSAM3S-EK2 obsahuje TFT LCD display s touch panelem, rozhraní USART, konektory BNC a ZigBEE. Dále lze vyzkoušet technologii QTouch - ovládání kapacitních tlačítek a sliderů. K programování kitu lze využít rozhraní JTAG, nebo ICE.



Obr. 15 Vývojový kit ATSAM3S-EK2 [11]

3.1.3. STMicroelectronics

MCU od STMicroelectronics lze programovat například v prostředích CoCoX, Atollic True Studio nebo KEIL. Poslední jmenované prostředí je vyvíjeno společností ARM. Umožňuje spravování projektů, editování zdrojových kódů, ladění programu a simulace. Prostedí je distribuováno v různých verzích a konfiguracích dle architektury programovaného MCU. Hlavní předností tohoto prostředí je kompilátor s velice dobrou optimalizací kódu a debugger. Hlavní nevýhodou je nízká přehlednost kódu v editoru a absence snippetů, automatického doplňování kódu a dalších vlastností, které nabízejí prostředí ostatních výrobců. V dubnu 2013 byla vydána nová verze prostředí s automatickým doplňováním kódu, avšak zatím je ve fázi vývoje a obsahuje chyby. Z pohledu přívětivosti je toto prostředí nejvíce nepřívětivé z výše popisovaných.

Programuje se buď pomocí bitových zápisů do registrů nebo lze využít knihovny CMSIS, viz kapitola 2.7.1. Výrobce poskytuje knihovny a příklady ke každé periférii. Firma STMicroelectronics je orientovaná na velké odběratele, a proto je zde velmi slabá podpora koncového uživatele. Po pochopení programování s využitím CMSIS je však programování velice efektivní.

Kit STM32F10x – EVAL (Obr. 16) je osazen mikroprocesorem STM32F103RBT6. Na desce se dají vyzkoušet všechny periferie, které mikroprocesor obsahuje. Z desky jsou vyvedeny sběrnice USB, CAN, USART, I²C a SPI. Dále je osazen slot pro SD kartu, mikrofon a audio výstup.



Obr. 16 Vývojový kit STM32F10x-EVAL [12]

Kvůli požadavku na jednoduchost programování jednotky jsem se rozhodl pro MCU STM32F103RB. Pro tento mikroprocesor je napsaný USB bootloader, který volně poskytuje komunita Leaflabs [16]. Leaflabs prodává vývojové kity s tímto MCU, které lze programovat v prostředí Maple-IDE v jazyce Wiring. K dispozici jsou také knihovny k většině periférií. Ty jsou psané v jazyce C a C++. Následuje přehled některých vlastností použitého MCU.

- Maximální frekvence jádra - 72 MHz
- 128 KB Flash paměti
- 20 KB SRAM paměti
- Napájecí napětí 2 – 3,6 V
- Vnitřní 8 MHz a 30 kHz oscilátor
- Sleep, Stop a Standby režimy
- 2 x 12-bit A/D převodník – až 16 kanálů
- Sedmikanálový DMA řadič s podporou čítačů, ADC, SPI, I²C a USART
- 26/27/51/80 I/O vývodů – mapovatelné do 16ti vektorů vnějšího přerušení, většina jsou 5 V tolerantní
- 7 čítačů
 - Tři 16-bit čítače s IC/OC/PWM
 - 16-bit čítač s řízením motorů – PWM s programovatelným dead-time a nouzovým vypnutím
 - Dva watchdog čítače
 - 24-bit watchdog čítač
- Komunikační rozhraní – I²C, USART, IrDA, CAN, USB 2.0
- Jednotka pro výpočet CRC

3.2. Uživatelské rozhraní

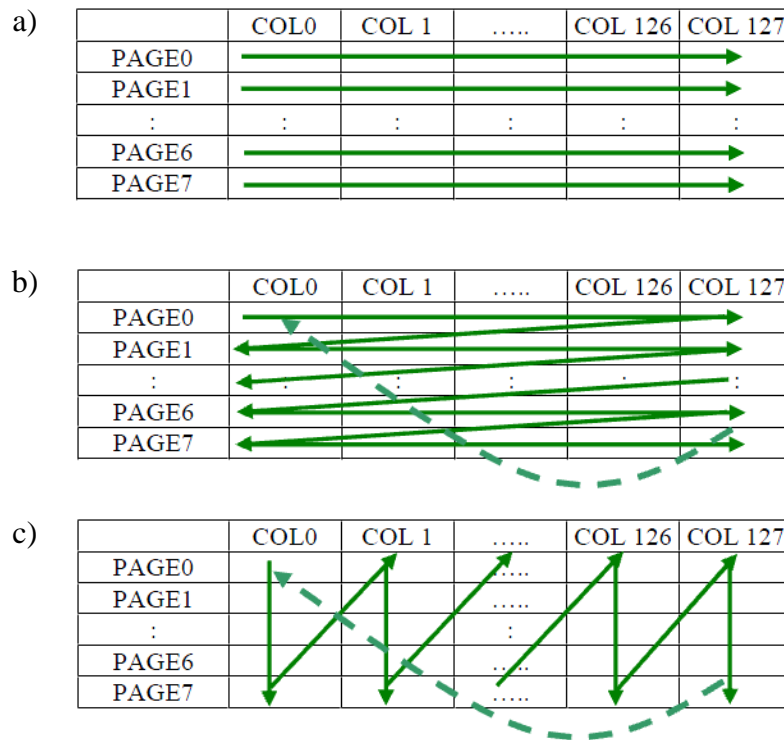
Jelikož nový hardware využívá starou kryt, bylo třeba dodržet rozměry i rozmístění tlačítek a displeje. Originální tlačítka jsou opatřena vodivou pryží, která při stisku tlačítka propojí vodivý motiv na DPS (Obr. 17). Tato tlačítka mají oproti běžným mikrosplínačům tu výhodu, že při sepnutí nevznikají zákmity.



Obr. 17 Motiv tlačítka na DPS

Průhled displeje v krytu je 10 mm x 32 mm veliký. Těmto rozměrům bylo nutno přizpůsobit nový display. Aby se co nejlépe nahradil původní zobrazovač s hotovými segmenty, byl zvolen grafický monochromatický OLED display s úhlopříčkou 0,9“. Jeho aktivní zobrazovací plocha je 5.5 mm x 22.3 mm. Poskytuje rozlišení 128 x 32 pixelů a je vybaven vlastním řadičem SSD1306.

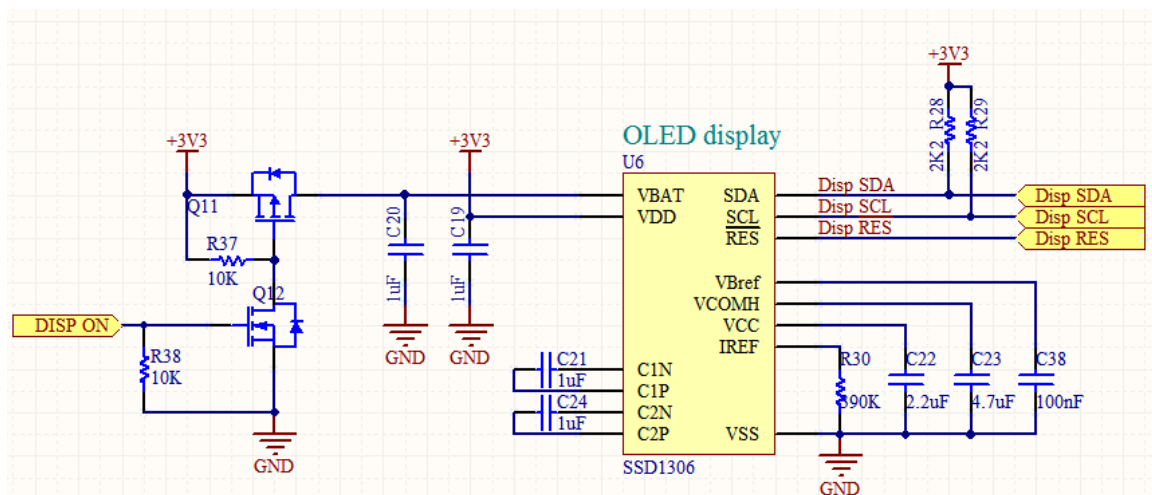
Univerzální řadič SSD1306 podporuje displeje s rozlišením maximálně 128 x 64 pixelů. Řadič integruje paměť RAM a oscilátor, což omezuje spotřebu zařízení a počet vnějších součástek. Komunikace probíhá pomocí jednoho ze čtyř komunikačních rozhraní – paralelní 6800/8000, sériové a I²C. Na OLED modulu je vyvedeno rozhraní I²C. Rozlišení příkazů a dat se provádí pomocí přepínacího bajtu. Zápis do paměti se provádí třemi způsoby (Obr. 18).



Obr. 18 Režimy plnění paměti [13]

Řádky jsou sloučeny po osmi bitech do stránek. Při každém zápisu bajtu do RAM se ukazatel automaticky inkrementuje dle zvoleného režimu. Koncové souřadnice odpovídají ploše displeje, nebo je lze nastavit. Ukazatel v režimu a) se inkrementuje směrem doprava a při dosažení koncového sloupce se zde zastaví a je nutno ho ručně posunout na další stránku. V režimu b) je tento posuv proveden automaticky. V režimu c) je ukazatel automaticky inkrementován směrem dolů a při dosažení poslední stránky je posunut o jeden sloupec doprava. Koncové stránky a sloupce jsou po resetu nastaveny na rozměry displeje, lze je však změnit a zapisovat na menší oblast.

Napájecí napětí pro displej je 7 V, to lze připojit buď externě, nebo je možné využít vnitřní DC/DC měnič, který je napájen ze zdroje napětí 3,3 – 4,2 V. Následuje schéma zapojení displeje na DPS.



Obr. 19 Schéma zapojení displeje

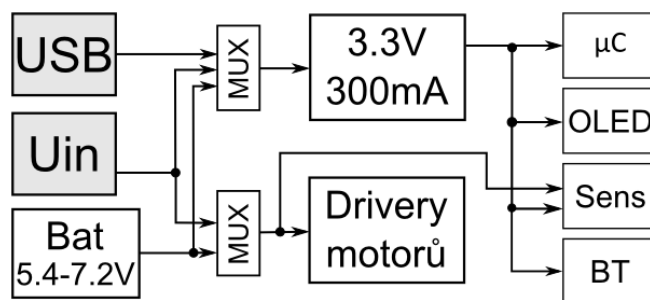
Napájení displeje je z důvodu ochrany a prodloužení životnosti spuštěno až po inicializaci řadiče. K tomu slouží tranzistory Q11 a Q12. Pull-up rezistory R28 a R29 nastavují klidové napětí na I²C sběrnici a R30 nastavuje referenční proud pro proudové zdroje řadiče. Nábojová pumpa využívá kondenzátory C21 a C24. Ostatní kondenzátory jsou stabilizační, nebo blokovací.

3.3. Napájení

Původní externí napájení ze střídavého adaptéru je nahrazeno stejnosměrným adaptérem s napětím 9 V. Původní zdroj totiž vyžadoval zapojení elektrolytických vyhlazovacích kondenzátorů. Jejich odstraněním se uvolnilo místo pro USB konektor, konektor rozhraní SWD³ a piezoměnič.

³ Serial Wire Debug

Bateriové napětí je původní – tedy 6 zinko-uhlíkových akumulátorů, které dodávají napětí 9 V. Uvažoval jsem i o možnosti nabíjení akumulátorů přímo v jednotce. Vzhledem k tomu, že jsou akumulátory zapojeny v sérii, bylo by nutné zasáhnout do mechanické konstrukce a nabíjet vždy po dvojici. Jiná možnost by byla použít Li-Pol akumulátory, které by se daly nabíjet pomocí integrovaného nabíjecího kontroléru. Tuto možnost jsem také nakonec zamítl kvůli nutnosti změnit prostor pro ukládání baterií.

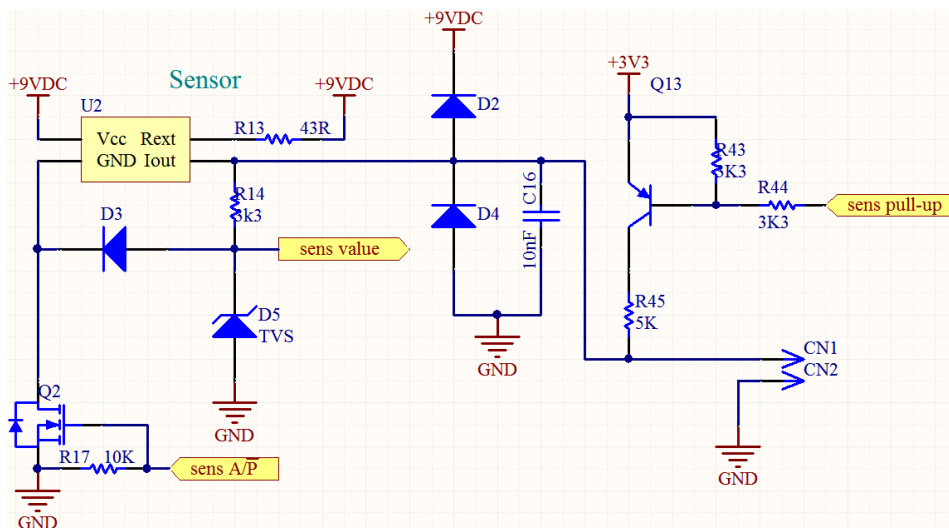


Obr. 20 Blokové schéma napájení

Jednotku je možno napájet pomocí tří zdrojů napětí. Původní dvě možnosti jsou rozšířeny o napájení z USB. Výběr napětí je realizován pomocí diod v zapojení výběru nejvyššího napětí. Pro napájení logických obvodů je zde LDO regulátor napětí, který je použit i v původní jednotce. Pokud jsou odpojeny ostatní zdroje, je napětí s USB přivedeno na budiče motorů. Napětí zdrojů je však snímáno a funkce motorů lze v tomto případě blokovat. Na výstup budiče lze připojit i další příslušenství a poté je možné podle odhadované spotřeby napájet i z USB. Budiče senzorů jsou napájeny střídavě ze dvou zdrojů.

3.4. Rozhraní senzorů

Protože nová jednotka používá původní senzory, je schéma pro obsluhu senzorů obdobné jako v původní jednotce (Obr. 5).



Obr. 21 Schéma rozhraní senzorů

Integrovaný obvod U2 je proudový zdroj PSSI2021Y. Rezistor R13 nastavuje maximální výstupní proud zdroje na 15 mA. Zdroj se aktivuje sepnutím tranzistoru Q2. Aby se na vstupním obvodu AD převodníku mikropočítače nemohlo objevit napětí 9 V, je zde dioda D3, která při sepnutém zdroji svede napětí na zemní potenciál. Jako dodatečná ochrana AD vstupu je zde zapojen transil. Diody D2 a D4 jsou zde kvůli ochraně přepětí nebo podpětí z výstupních svorek. Při čtení hodnoty ze senzoru je připojen pull-up rezistor pomocí tranzistoru Q13. Kondenzátor C16 je zde zapojen kvůli potlačení rušení ze senzoru. Z důvodu pomalého poklesu napětí při přepnutí na režim čtení byla hodnota kondenzátoru C16 snížena na 100 μ F.

3.5. IR komunikace

K dálkovému ovládní LEGO zařízení se používá infračervená komunikace. K tomuto účelu byl vytvořen protokol, který podporuje celou řadu režimů ovládní. Základní rámec se skládá ze start a stop bitu, tří datových bajtů a jednoho bajtu kontrolního součtu. Následuje popis jednotlivých bitů rámce.

<i>start</i>	<i>T</i>	<i>E</i>	<i>C</i>	<i>C</i>	<i>a</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>D</i>	<i>D</i>	<i>D</i>	<i>D</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>stop</i>
--------------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	-------------

Start/Stop	Synchronizační bity
T	Tento bit je negován při každé následující zprávě
E	Přepíná mezi PWM a ostatními režimy
CC	Výběr jednoho ze čtyř kanálů
a	Vždy vyslán jako nula, rezerva do budoucna
MMM	Výběr režimu řízení
DDDD	Tyto bity mění význam dle vybraného režimu
LLLL	Kontrolní součet

IR přijímač LEGO je schopen ovládat 4 výstupní vývody samostatně nebo je sdružit do dvou portů a ovládat jimi motory v různých režimech. Následuje stručný popis jednotlivých režimů, kterými jsou ovládní výstupy. Detailní popis komunikace lze najít v odkazu [15].

- Kombinovaný přímý – ovládá samostatně dva porty v plovoucím, dopředném, obráceném a brzděném módu - musí se obnovovat.
- Samostatný nepřetržitý výstup – nastavuje, resetuje nebo neguje výstupní pin.
- Samostatný přerušovaný výstup – stejné možnosti jako u předchozího režimu, ale musí se obnovovat.
- Samostatný port – zde je možné ovládat jak motor pomocí PWM, tak jednotlivé piny portu. Některé z režimů se musí obnovovat.
- Kombinovaný PWM – ovládá samostatně dva porty pomocí PWM v sedmi krocích.

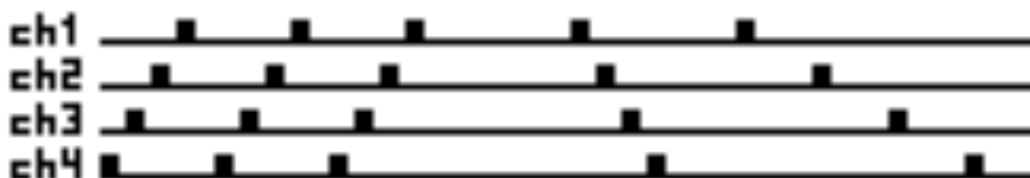
Obnovování příkazu se musí opakovat minimálně jednou za 1,2 vteřiny, jinak dojde k odpojení výstupu. Používá se obecně u maximální střední hodnoty PWM nebo u pinu nastaveného na vysokou úroveň (v tomto případě +9 V).

Kódování a časování

Ke komunikaci se využívá infračervené záření o vlnové délce 950 nm. Na něj je modulován obdélníkový signál o frekvenci 38 kHz. Každý bit, včetně start a stop bitu, je uvozen šesti impulzy, po kterých následuje proměnná doba nízké úrovně. Podle této doby jsou rozlišeny jednotlivé bity:

- Log. 1 – 6 impulzů + 10 časových intervalů nízké úrovně - 421 μ s
- Log. 0 – 6 impulzů + 21 časových intervalů nízké úrovně - 711 μ s
- Start/stop bit – 6 impulzů + 39 časových intervalů nízké úrovně - 1183 μ s

Při dekódování se počítá s určitou časovou tolerancí na každý bit, maximální čas potřebný pro odeslání jedné zprávy je tak přibližně 18 ms. Protože protokol umožňuje vysílat na čtyřech kanálech, je nutno dodržet rozložení zpráv v čase (obr. 22), aby nedošlo ke vzájemnému rušení. Každá zpráva je odeslána pětkrát po sobě v intervalech, které jsou určeny dle příslušného kanálu.



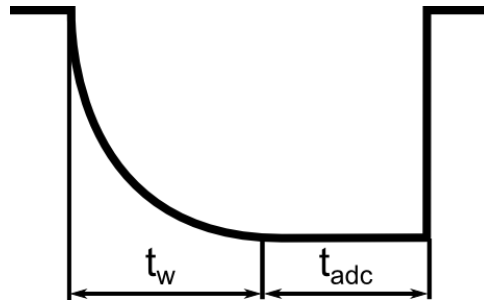
Obr. 22 Rozložení zpráv v čase

Pro zjištění bezchybného vyhodnocení zprávy se přenáší čtyřbitový kontrolní součet (LLLL – viz str. 22). Tento součet se generuje před vysláním následujícím způsobem:

$$LLLL = 0xF \text{ xor (první nibble) xor (druhý nibble) xor (třetí nibble)}$$

3.6. Obsluha senzorů

Jak již bylo popsáno v kapitole o senzorech, většinu času je senzor napájen a během krátkého okamžiku vypnutí napájení je přečtena analogová hodnota. Následující obrázek zachycuje přibližný průběh napětí po vypnutí napájení senzoru.



Obr. 23 Průběh napětí na senzoru

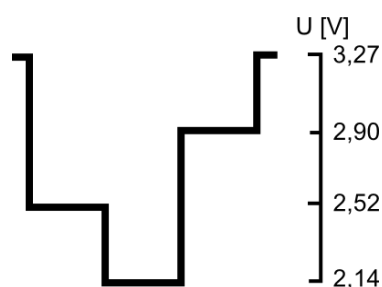
Měření senzorů ukázalo, že doba potřebná k ustálení měřeného napětí t_w se pohybuje okolo 40 μs . Po vypnutí napájení a připojení pull-up rezistoru se počká po dobu 50 μs a spustí se AD převodník. Doba jednoho cyklu převodu je 12,5 taktu + doba vzorkování, kterou lze nastavit podle impedance měřeného zdroje signálu. Taktovací frekvence AD převodníku je maximálně 14 MHz. Ta se odvozuje od hodinového signálu systému, který je zde 72 MHz. Dělicí poměr děličky je 2, 4, 6 a 8, nevyšší možná frekvence AD převodníku je tedy 12 MHz. Já jsem zvolil dobu vzorkování 7,5 taktu a to odpovídá 1,6 μs na celý převod. Používám zde regulární mód ve kterém je možné opakovaně vzorkovat až 16 kanálů a každému kanálu lze nastavit rozdílnou dobu vzorkování.

Po každém převodu se aktivuje DMA řadič, který přenesení výsledek převodu do pole v paměti. V době t_{adc} se navzorkuje každý senzor dvakrát a po přerušení od DMA se znovu zapne napájení a odpojí pull-up rezistor. Zároveň se zavolá funkce na přepočítání hodnoty dle zapojení senzoru. Tento proces se opakuje každé 3 ms.

Reprezentace hodnot

Napětí ze světelného senzoru se pohybuje v rozmezí 2,7 – 2,95 V v závislosti na okolním osvětlení a odrazivosti osvětlovaného materiálu. Jelikož se senzor využívá zejména na detekci rozhraní dvou kontrastních barev, je zapotřebí zjistit odpovídající napěťovou diferenci. Při vzdálenosti 2 cm od snímaného povrchu se napětí mění o více než 100 mV. Funkce pro vyhodnocení světlého/tmavého povrchu si ukládá předchozí hodnotu napětí a porovnává ji s aktuální. Při častém volání funkce je však diference velmi malá. Zde bylo třeba zvolit kompromis mezi rychlou odezvou senzoru a schopností rozeznat změnu odrazivosti povrchu i při pomalých změnách.

Senzor rychlosti poskytuje napětí ve čtyřech úrovních, které jsou znázorněny na následujícím obrázku.



Obr. 24 Napěťové úrovně senzoru otáček

Napěťová diference mezi hodnotami je 370 - 380 mV, takže je možné dobře rozlišit jednotlivé úrovně napětí. Funkce *Speed_Sens_Conversion* převádí napěťové hodnoty na jednotlivé stavy a počítá jejich změny. Přepočet na otáčky za minutu je proveden jednou za sekundu.

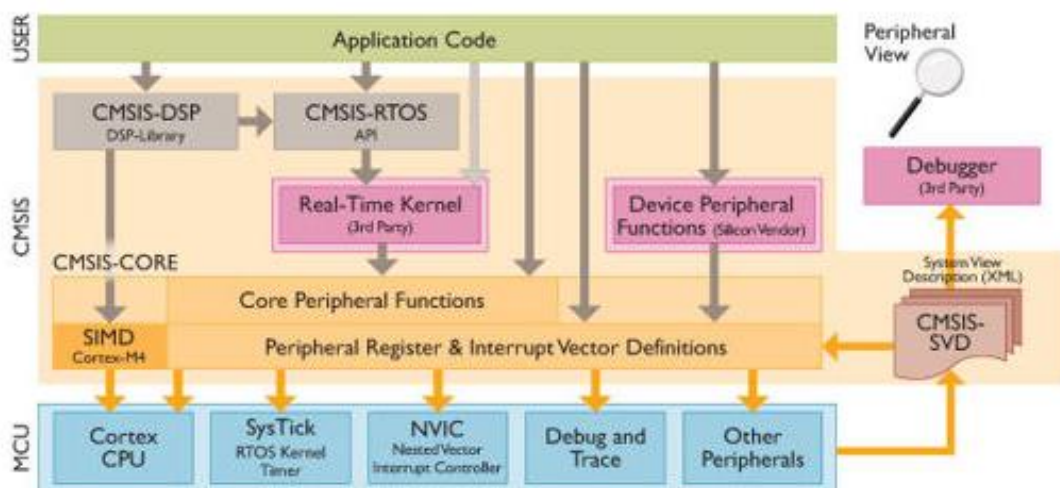
3.7. Programování jednotky

Jednotku lze programovat několika způsoby. MCU má již z výroby naprogramovaný bootloader, pomocí kterého lze program nahrát přes sériovou linku. Na desce jednotky jsem však sériovou linku nevyvedl.

Další způsob programování je přes rozhraní SWD (serial wire debug). Zde se ke komunikaci využívají 2 vodiče – datový a hodinový. Dále je třeba propojit země programátoru a cílové desky. Jednotku jsem programoval pomocí programátoru ST-Link, který umožňuje ladění programu i napájení. Nahrání programu je možné buď přímo z programu KEIL μ Vision nebo lze nahrát bin soubor pomocí STM32 ST-Link Utility. Periferie MCU a obvody na desce jsem programoval v programu μ Vision pomocí knihoven CMSIS. Poslední možnost programování je přes USB. K tomu je zapotřebí nejprve nahrát USB bootloader od výrobce STMicroelectronics nebo již zmiňovaný bootloader od komunity Leaflabs.

3.7.1. CMSIS

CMSIS je zkratka z anglického *ATM Cortex Microcontroller Software Interface Standart*. Jedná se o softwarovou vrstvu pro mikroprocesory řady Cortex-M, která je nezávislá na výrobci mikroprocesoru. Knihovny CMSIS umožňují snadnější používání periférií a zlepšují přenositelnost software. Schéma CMSIS popisuje následující obrázek.



Obr. 25 Struktura CMSIS [14]

- Jádru CMSIS definuje rozhraní pro všechny procesory řady Cortex-M. Poskytuje start-up metody a funkce pomocí kterých lze přistupovat k perifériím a ke specifickým funkcím mikroprocesoru. Všechna zařízení, která jsou podporovaná knihovnou CMSIS, mají jasně danou strukturu registrů periférií a přerušení.
- CMSIS-DSP knihovny podporují vektorové a maticové operace, komplexní čísla, filtrační a kontrolní funkce, fourierovy transformace a další často používané DSP funkce. Většina těchto funkcí pracuje s plovoucími nebo jinými formáty pevné desetinné čárky. Knihovna DSP je napsaná kompletně v jazyce C a je poskytována se zdrojovými kódy, takže je možné upravovat algoritmy dle konkrétní aplikace.
- CMSIS-RTOS rozhraní sjednocuje přístup k operačním systémům reálného času a rozšiřuje tak výhody používání CMSIS.
- CMSIS-SVD popisuje mikroprocesor z hlediska programátora včetně registrů periférií. SVD soubory umožňují vytvořit hlavičkový soubor, který obsahuje registry periférií a definice přerušení. Tyto informace jsou využívány také pro ladění programů.

Převzato z [14].

3.7.2. Maple IDE

V minulosti byla práce s procesory ARM v neprofesionálních programech nepřívětivá kvůli proprietárním nástrojům a složitým instrukčním sadám. Kvůli tomu byly málo srozumitelné žákům a amatérským programátorům. Proto vznikl projekt Maple, který má za cíl zpřístupnit procesory ARM a poskytnout jednodušší programovací prostředí. Komunita LeafLabs dává k dispozici zdrojové kódy k programům a schémata k vývojovým kitům, které prodává.

Velkou výhodou programovacího prostředí Maple IDE je, že je podobné s Arduinem⁴, je dobře známé mezi amatérskými programátory. Prostedí zachovává práci s kódem i označení vývodů.

Rozvržení vývodů MCU na desce jednotky je kompatibilní s kitem od LeafLabs. Z toho důvodu ji lze programovat stejným způsobem. Na následujícím obrázku je ukázka prostředí Maple IDE.

⁴ Otevřená platforma založená na MCU ATmega.

```
Blink | Maple IDE v0.0.12 (Arduino 0018)
File Edit Sketch Tools Help

Blink $

Turns on the built-in LED on for one second, then off for one second
repeatedly.

Ported to Maple from the Arduino example 27 May 2011
By Marti Bolivar
*/

void setup() {
  // Set up the built-in LED pin as an output:
  pinMode(D10, OUTPUT);
}

void loop() {
  digitalWrite(D10 , HIGH); // sets the LED on
  delay(1000); // waits for a second
  digitalWrite(D10, LOW); // sets the LED off
  delay(1000); // waits for a second
}

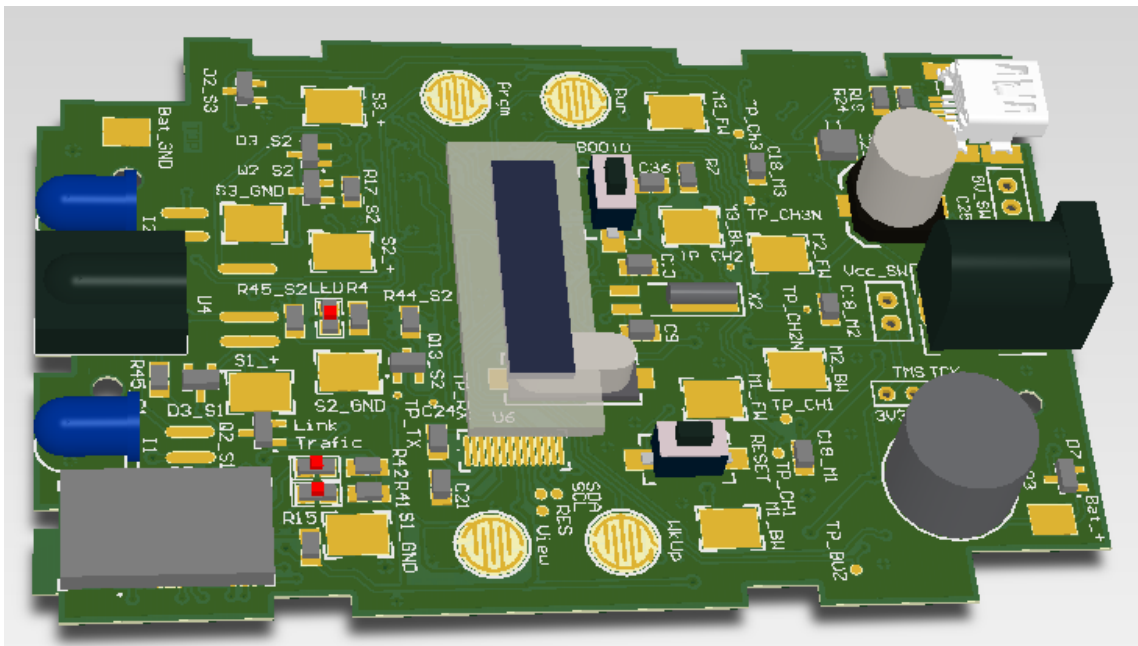
Done uploading.
bytes_per_flash=100
Starting download:
[#####] finished!
state(8) = dfuMANIFEST-WAIT-RESET, status(0) = No error condition
is present
13
```

Obr. 26 Ukázka programu Maple IDE

Zde může uživatel buď použít již hotové knihovny pro MCU, nebo si napsat vlastní. Program obsahuje knihovny *libmaple*, které pracují s registry MCU a uživatel se tak může soustředit na vývoji aplikace. Každá dílčí knihovna má v hlavičkové funkci struktury obsahující definice názvu registrů a jejich bitů. Výsledná struktura má v sobě odkaz na tyto struktury a některé další proměnné např. informace o nastavení hodinového signálu. S těmito strukturami pak pracují funkce pro práci s periferiemi. Ve své podstatě se jedná o podobné knihovny, jako jsou CMSIS. Další informace o projektu Maple jsou k dispozici na stránkách komunity LeafLabs [16].

3.8. Návrh DPS

Jelikož je nová deska umístěna do krytu staré jednotky, bylo nutné zachovat stejný tvar a rozměry. Jednotku RCX jsem měl k dispozici, takže jsem mohl změřit všechny rozměry. Dalším požadavkem bylo zachovat umístění napájecího konektoru, konektoru vstupů a výstupů, IR součástek a displeje. Jediný zásah do mechanické konstrukce bylo vyvrtání otvoru pro USB konektor, který jsem umístil na místo, kde předtím byly vyhlazovací kondenzátory. DPS jsem navrhl v návrhovém systému Altium Designer.



Obr. 27 3D model navržené DPS

4. Závěr

Při ožívování DPS jsem zjistil jisté nedostatky. Protože je jednotka v původním krytu, musel jsem zajistit, aby žádná ze součástek nepřekážela. Tlačítko RESET jsem však umístil tak, že by bylo krytem stále zmáčknuté a musel jsem ho otočit o 90°. Dalším problémem je nevhodně zvolený budič motoru. Pokud je motor připojen na jeho výstup při napájení 9 V, výstupní napětí klesne téměř na 0 V. Motor je pak zapotřebí roztočit ručně a po té se již dá řídit pomocí PWM. Při nižším napájení (např. z USB) tyto problémy nenastávají. V další revizi je třeba zvolit jiný budič.

Funkčnost jednotky jsem ověřoval pomocí programů psaných s využitím knihoven CMSIS. Napsal jsem knihovny pro display, motory, senzory a akustický měnič. Dále jsem naprogramoval funkce pro vysílání IR rámců, avšak zatím jsem neměl k dispozici hardware, na kterém bych otestoval jejich příjem. Do jednotky jsem nahrál USB bootloader a otestoval programování jednotky v prostředí Maple-IDE [16]. Díky podobnému zapojení vývodů MCU jako na kitu Maple, lze jednotku programovat stejným způsobem. Na ukázkou jsem do jednotky nahrál program, který vyčítá hodnoty ze světelného senzoru a senzoru otáček a zobrazuje měřené hodnoty na display. Projekt z programu μ Vision a všechny použité knihovny jsou k dispozici na CD.

Původní jednotka se programovala pomocí IR rozhraní, zatímco inovovaná jednotka se dá programovat přes USB nebo SWD. Tím se rozšířily i možnosti použití různých programovacích prostředí. Jednotka disponuje hardwarem pro IR komunikaci, pomocí které může jednotka komunikovat s novou jednotkou NXT a jejími IR perifériemi. Další výhodou je možnost komunikace přes Bluetooth modul. Dále je plánováno napsat knihovny pro příjem IR rámců a Bluetooth. Pro případ, že by se vyráběla nová DPS, je na CD uložen opravený projekt z programu Altium Designer.

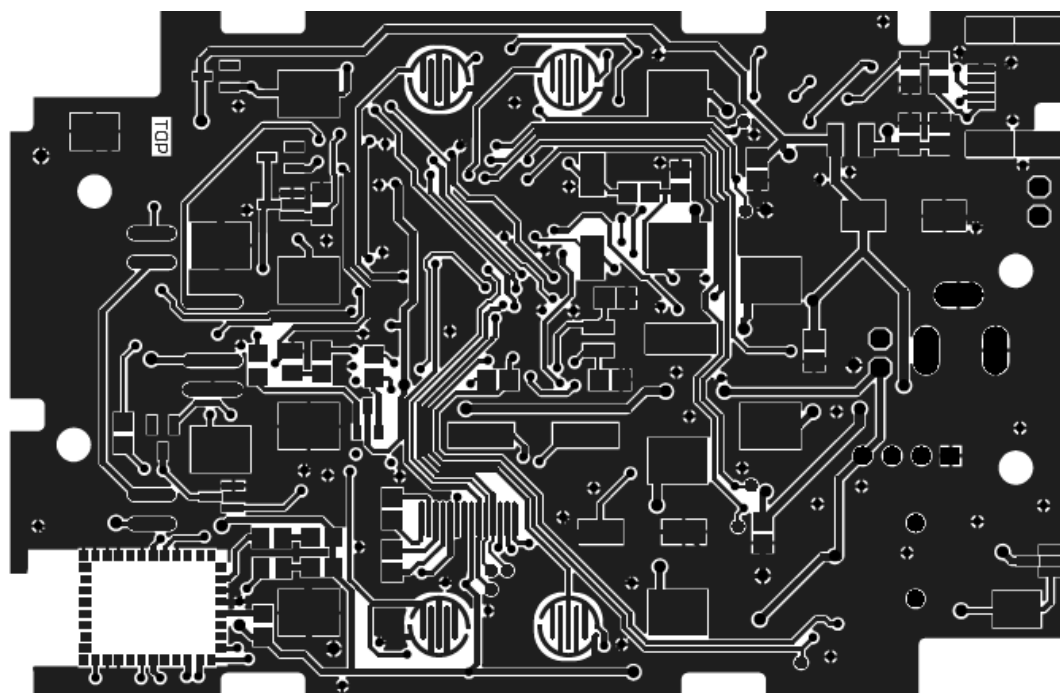
Seznam Obrázků

Obr. 1 DPS inovované jednotky [1].....	2
Obr. 2 Schéma displeje	3
Obr. 3 Rozložení tlačítek	3
Obr. 4 Schéma napájení [2]	2
Obr. 5 Rozhraní senzorů [2]	3
Obr. 6 Závislost odporu na teplotě.....	4
Obr. 7 Napětí z rotačního senzoru [3].....	5
Obr. 8 Vnitřní zapojení senzoru otáček [4].....	6
Obr. 9 Vnitřní zapojení světelného senzoru [5]	6
Obr. 10 Ukázka programu RCX Code [6]	8
Obr. 11 Ukázka jednoduchého programu v RoboLab [7]	10
Obr. 12 Příkazová paleta RoboLab [8]	10
Obr. 13 Ukázka programu BrickOS [9].....	11
Obr. 14 Vývojový kit EKS-LM3S8962 [10]	13
Obr. 15 Vývojový kit ATSAM3S-EK2 [11]	14
Obr. 16 Vývojový kit STM32F10x-EVAL [12].....	15
Obr. 17 Motiv tlačítka na DPS.....	17
Obr. 18 Režimy plnění paměti [13]	18
Obr. 19 Schéma zapojení displeje.....	19
Obr. 20 Blokové schéma napájení	20
Obr. 21 Schéma rozhraní senzorů.....	21
Obr. 22 Rozložení zpráv v čase	23
Obr. 23 Průběh napětí na senzoru	24
Obr. 24 Napěťové úrovně senzoru otáček	25
Obr. 25 Struktura CMSIS [14].....	27
Obr. 26 Ukázka programu Maple IDE.....	29

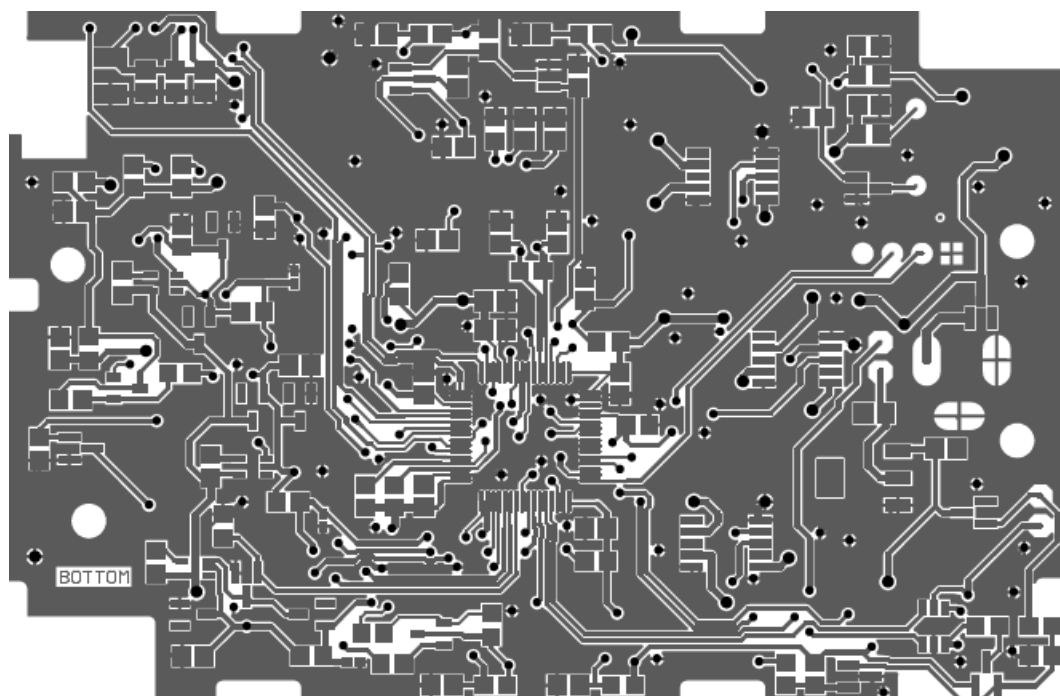
Seznam zdrojů

- [1] Hispabrick Magazine. [online]. [2013-05-06]. Dostupné z: <http://www.hispabrickmagazine.com>
- [2] Mindstorms Internals. [online]. [2013-05-06]. Dostupné z: <http://www.crynwr.com/lego-robotics/>
- [3] ExtremeNXT. [online]. [2013-05-06]. Dostupné z: <http://www.extremenxt.com/rotate.htm>
- [4] Philohome. [online]. [2013-05-06]. Dostupné z: <http://www.philohome.com/sensors/legorot.htm>
- [5] ExtremeNXT. [online]. [2013-05-06]. Dostupné z: <http://www.extremenxt.com/light.htm>
- [6] Robova LEGO® stránka. [online]. [2013-05-06]. Dostupné z: http://seifrob.sweb.cz/lego_robotika/RIS_navod/ris-navod.htm
- [7] CIPCE. [online]. [2013-05-06]. Dostupné z: <http://www.cipce.rpi.edu/programs/robotics/roboLab/lesson1/lesson1.html>
- [8] BPSRobotics. [online]. [2013-05-06]. Dostupné z: <http://bpsrobotics.wikispaces.com/file/view/RoboLabInventor3.pdf>
- [9] Jochen Hoenicke. [online]. [2013-05-06]. Dostupné z: <http://www.jochen-hoenicke.de/rcx/brickemu.html>
- [10] Mouser. [online]. [2013-05-06]. Dostupné z: <http://mouser.com>
- [11] Digikey. [online]. [2013-05-06]. Dostupné z: <http://digikey.com>
- [12] STMicroelectronics. [online]. [2013-05-06]. Dostupné z: <http://st.com>
- [13] Datasheet řadiče SSD1306
- [14] ARM. [online]. [2013-05-06]. Dostupné z: <http://arm.com>
- [15] Philohome. [online]. [2013-05-06]. Dostupné z: http://www.philohome.com/pf/LEGO_Power_Functions_RC.pdf
- [16] LeafLabs. [online]. [2013-05-06]. Dostupné z: <http://leaflabs.com>

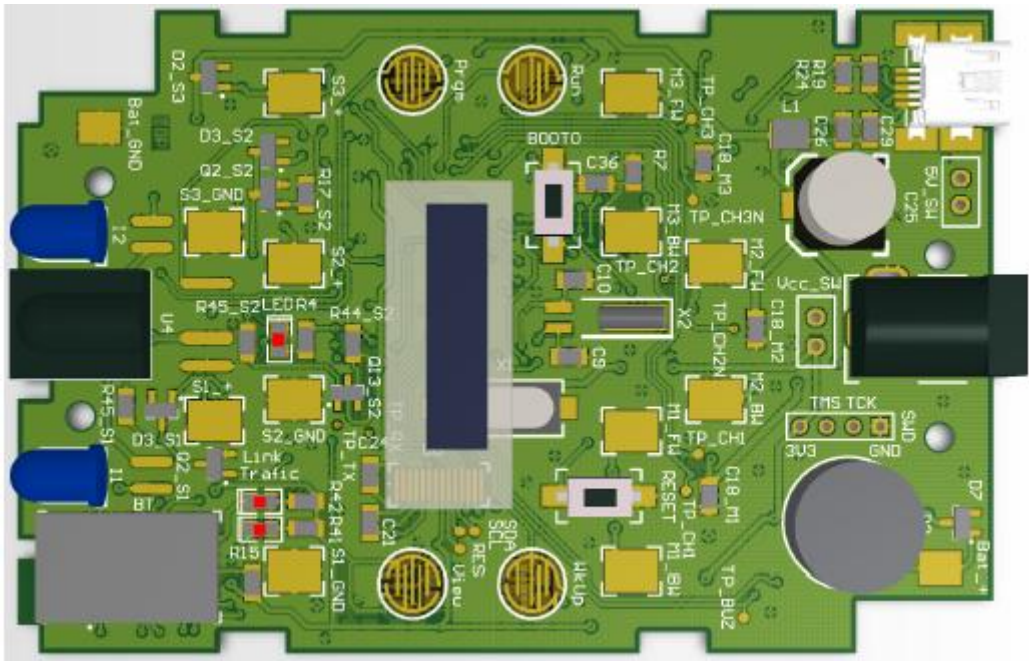
Přílohy



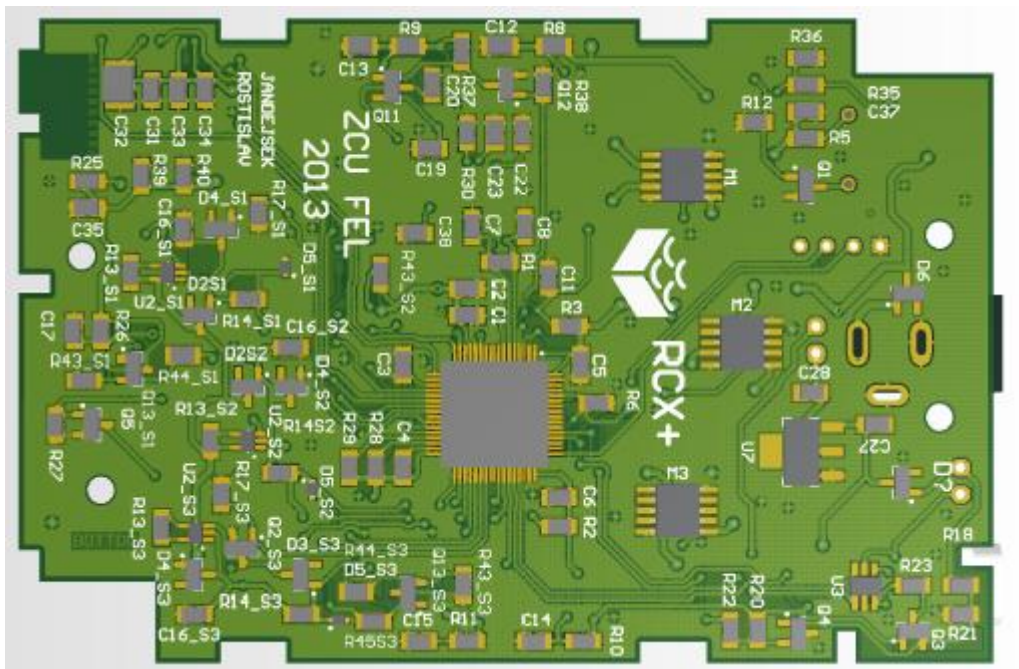
Návrh DPS - horní vrstva



Návrh DPS - spodní vrstva



3D model - horní strana



3D model - spodní strana

A

A

B

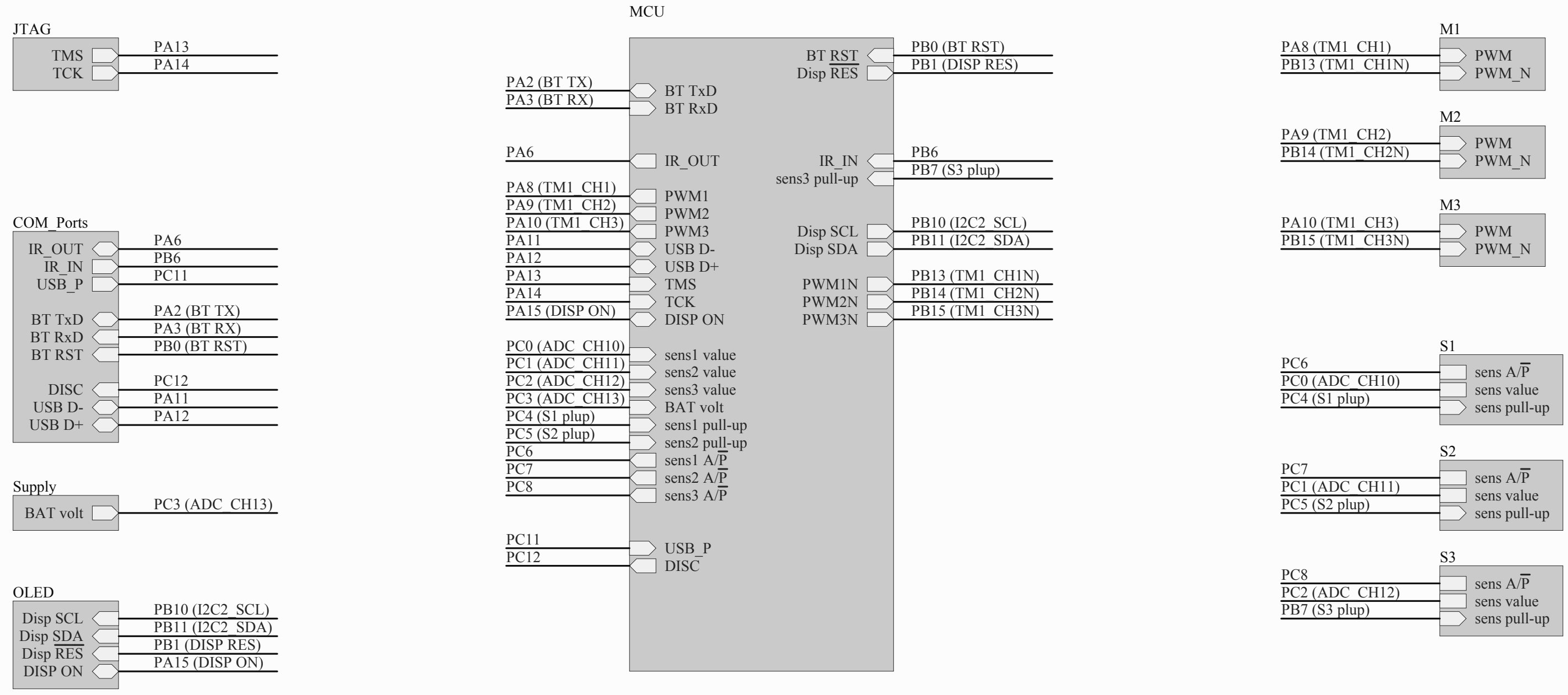
B

C

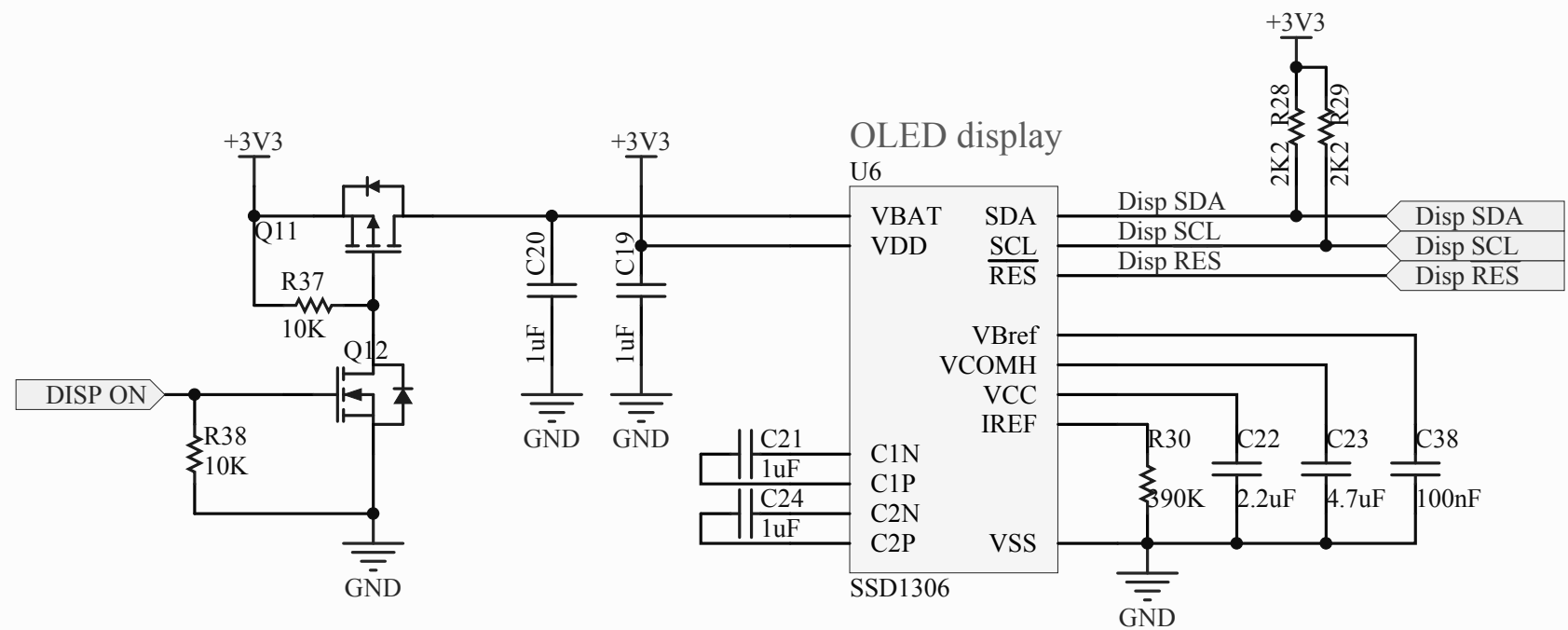
C

D

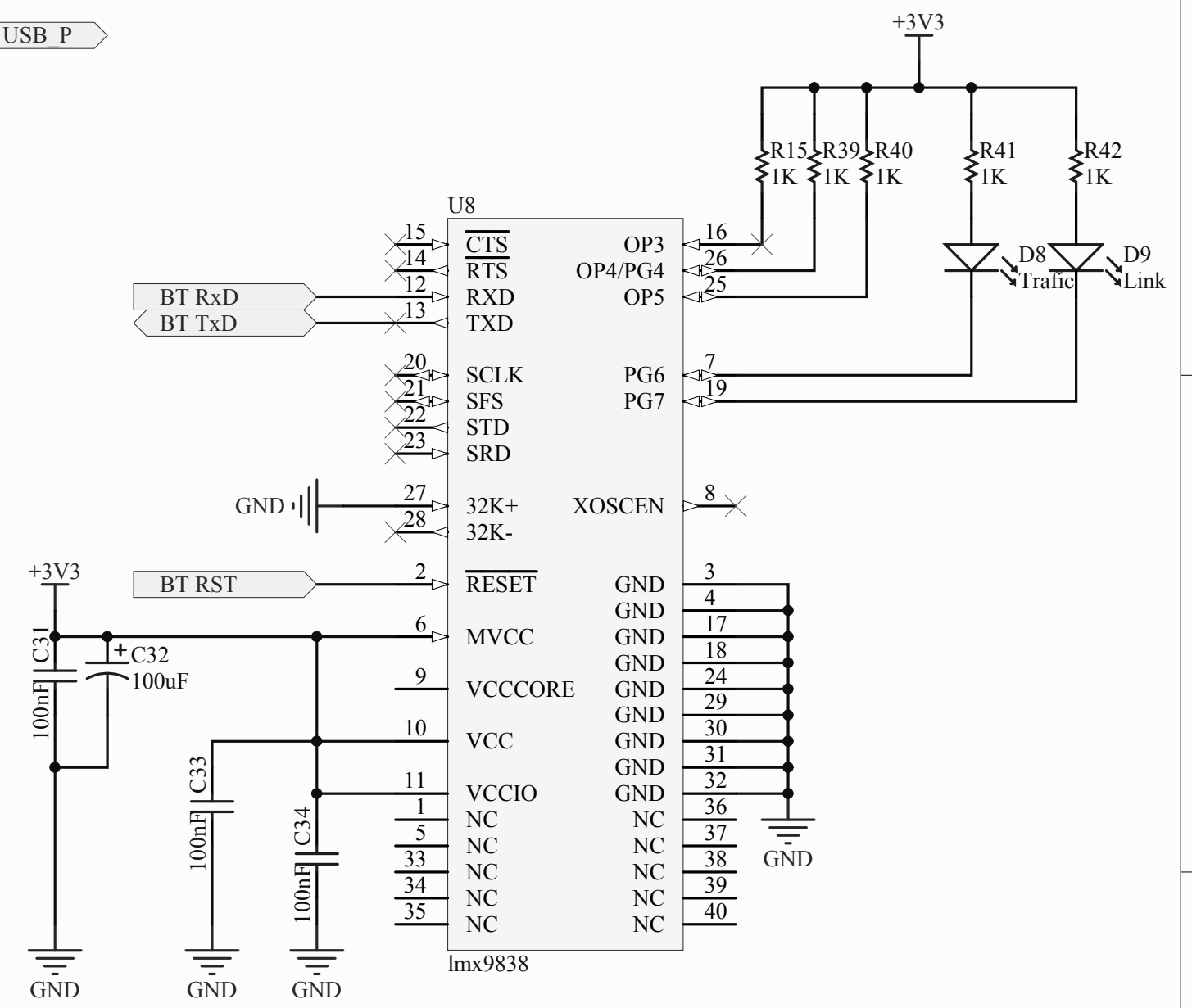
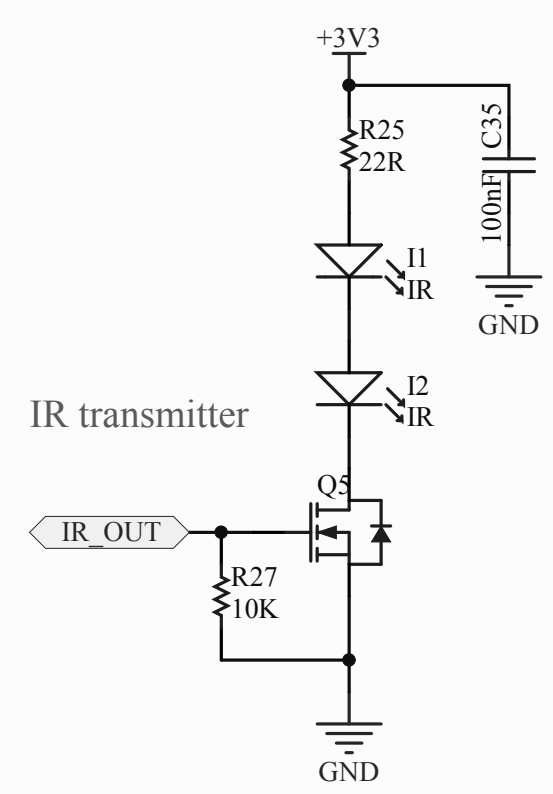
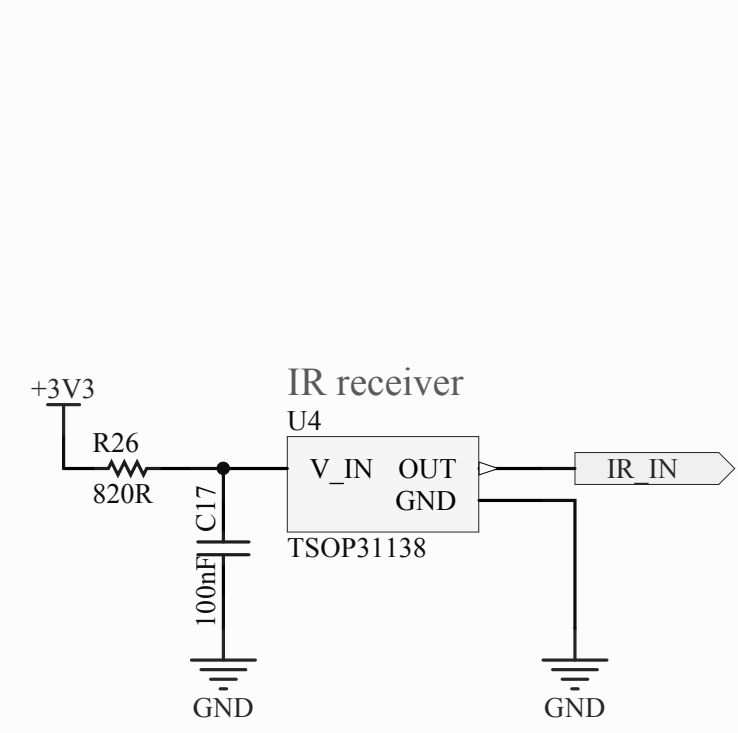
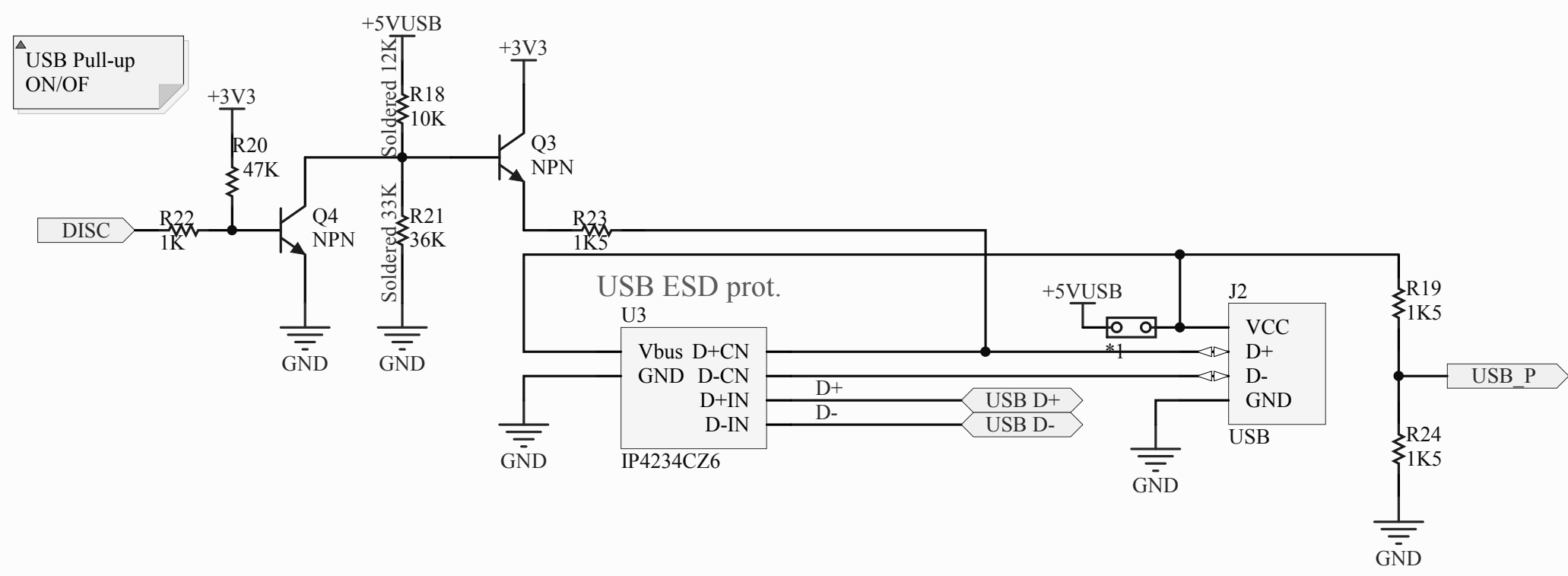
D



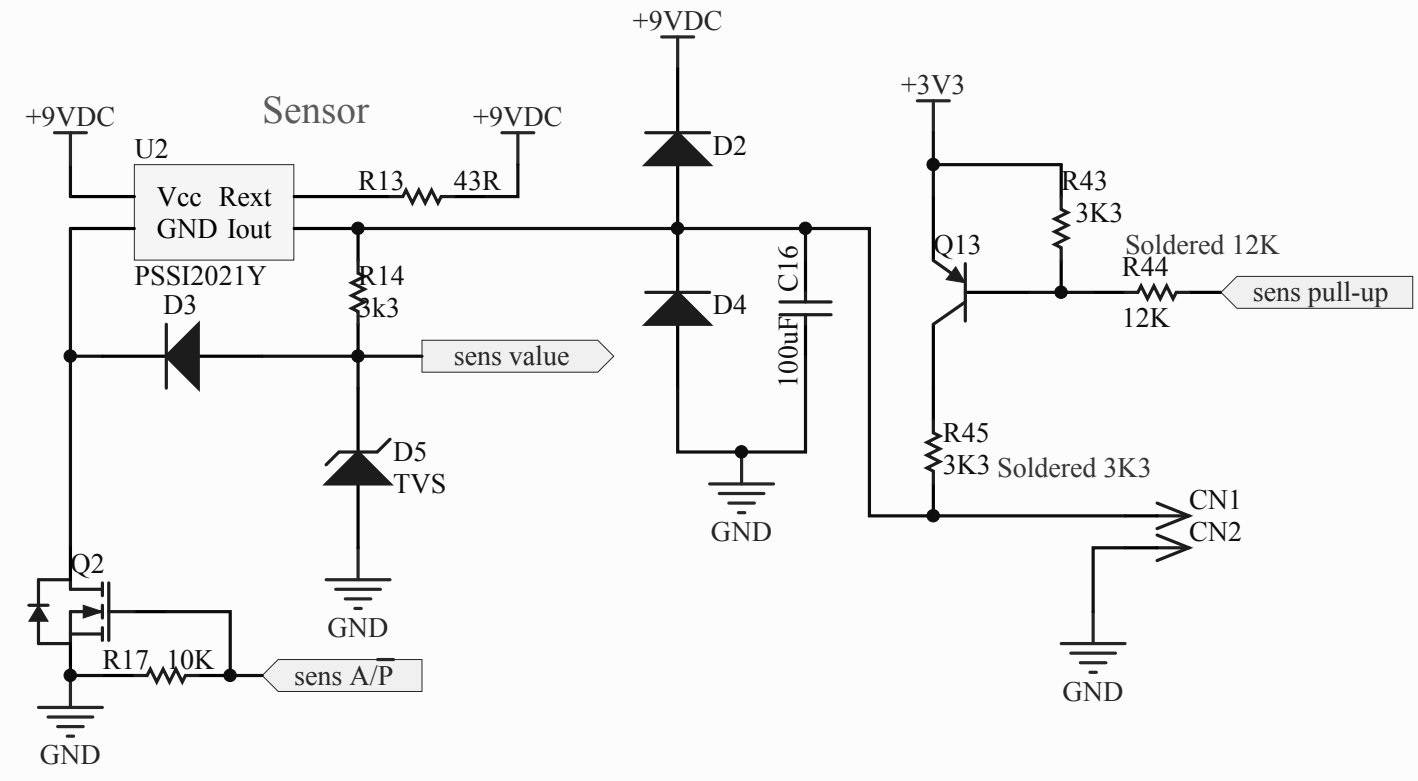
Title TOP		
Size A4	Number	Revision
Date: 6.5.2013	Sheet 1 of 8	
File: D:\Vyuka...\Top.SchDoc	Drawn By: Jandejsek Rostislav	



Title OLED		
Size A4	Number	Revision
Date: 6.5.2013	Sheet 2 of 8	
File: D:\Vyuka\...\OLED.SchDoc	Drawn By: Jandejsek Rostislav	



Title Communication		
Size A4	Number	Revision
Date: 6.5.2013	Sheet 3 of 8	
File: D:\Vyuka...\Communication.SchDoc	Drawn By: Jandejsek Rostislav	



Title		
Sensor		
Size	Number	Revision
A4		
Date:	6.5.2013	Sheet 4 of 8
File:	D:\Vyuka\...\Sensor.SchDoc	Drawn By: Jandejsek Rostislav

A

A

B

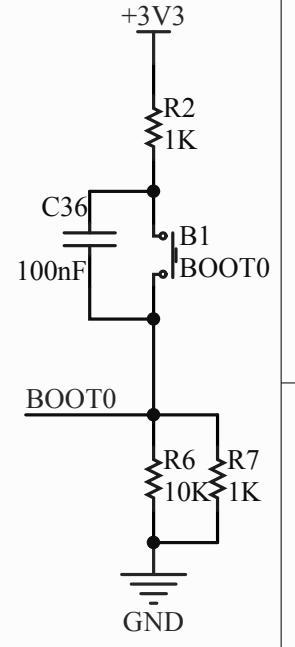
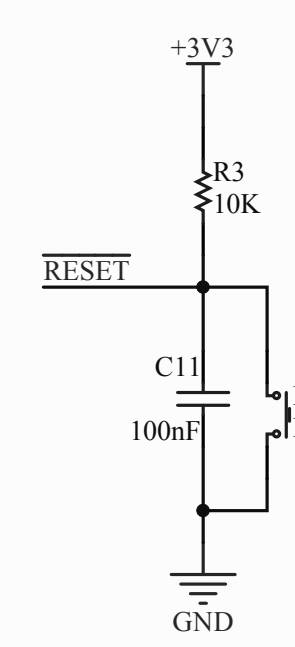
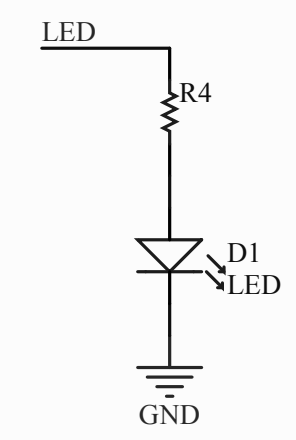
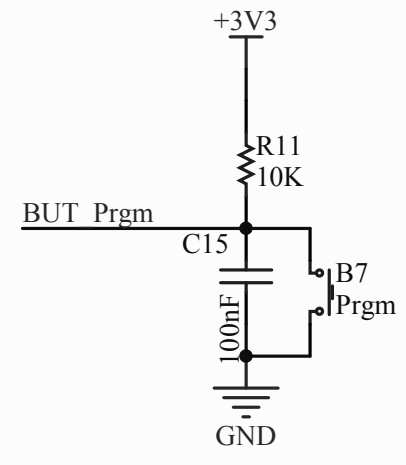
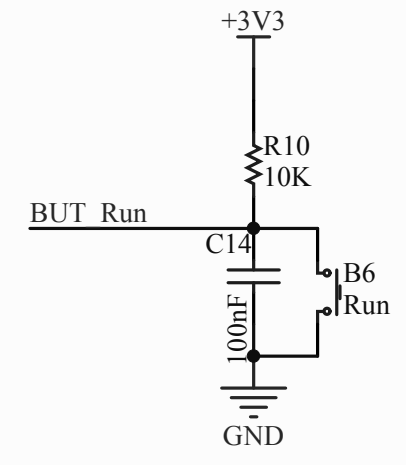
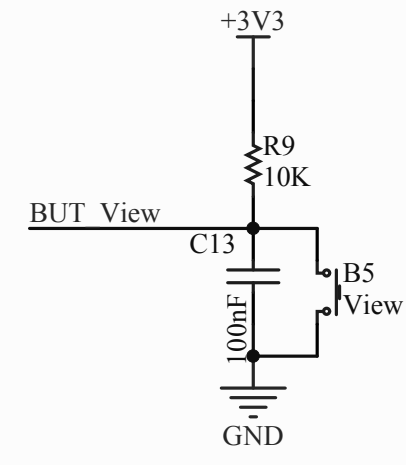
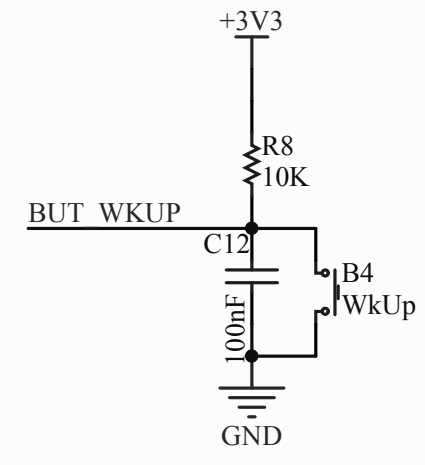
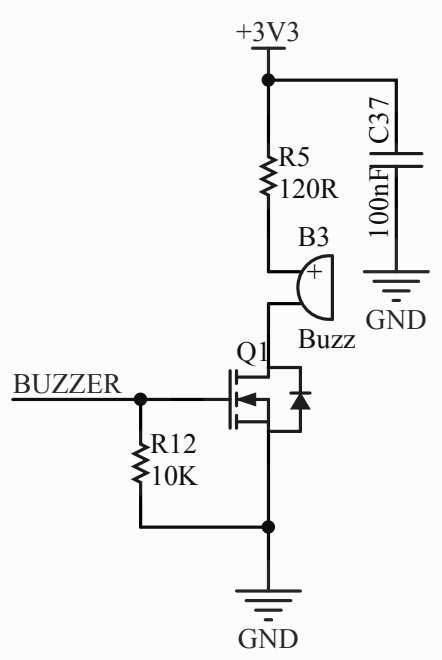
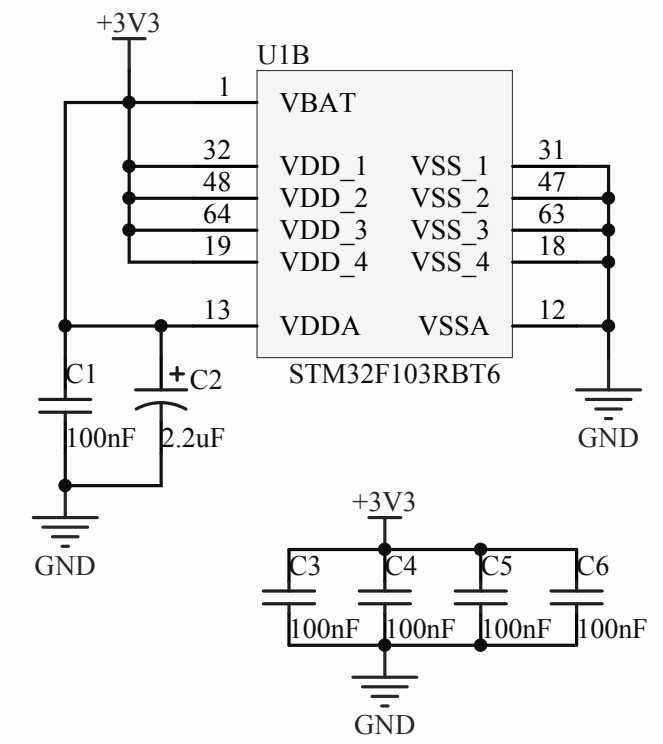
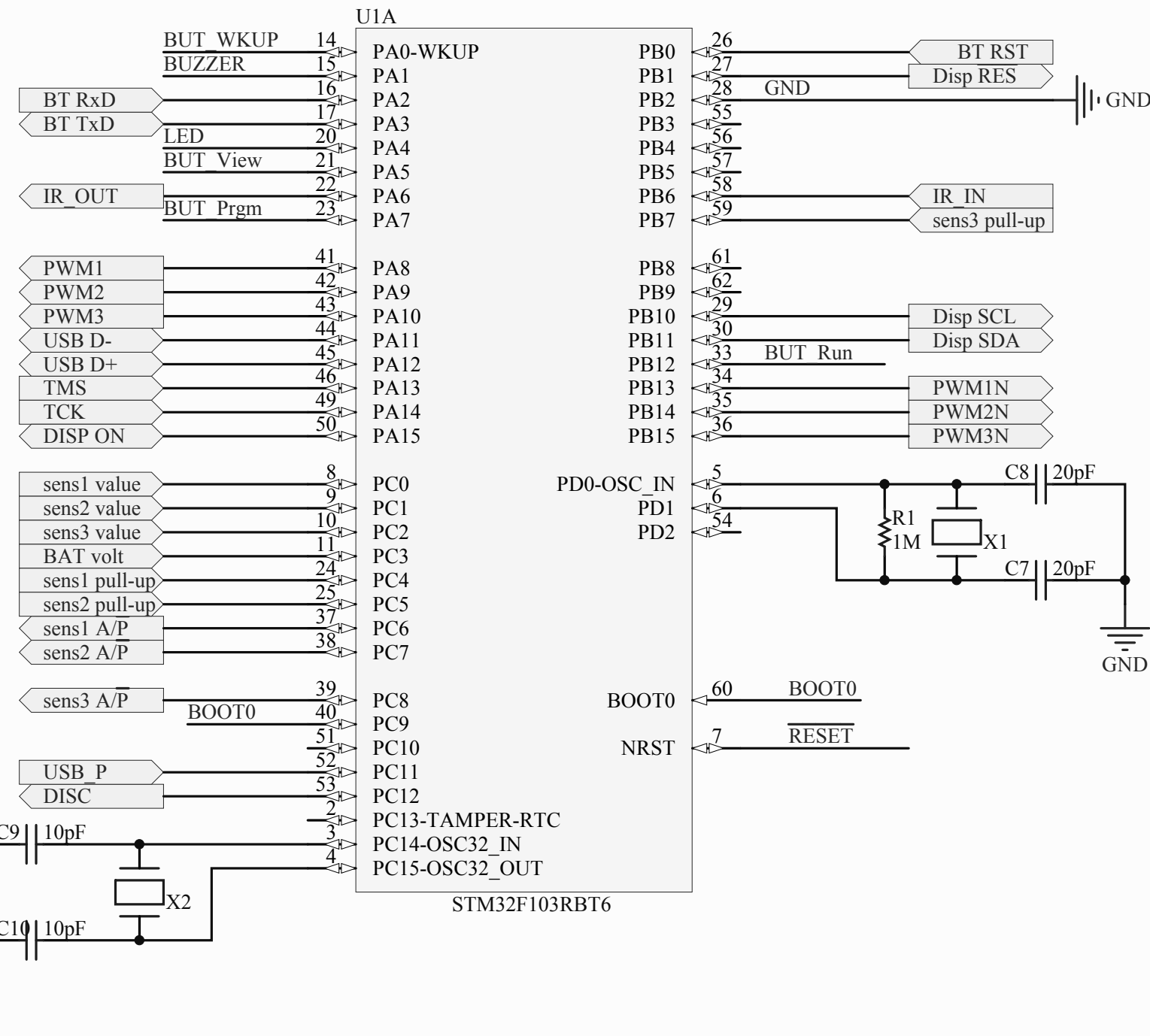
B

C

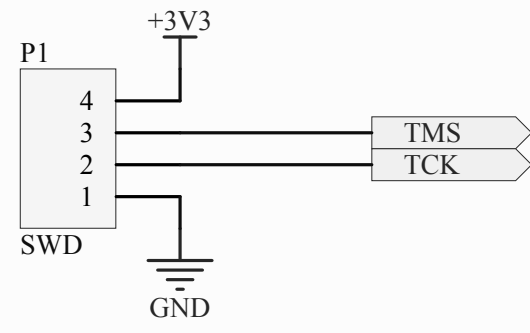
C

D

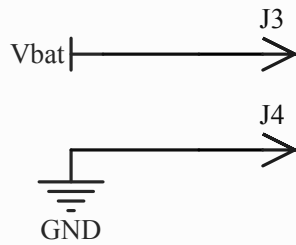
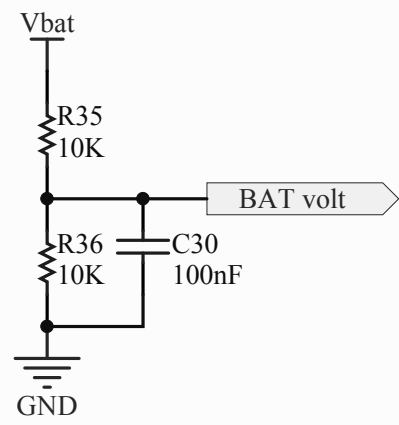
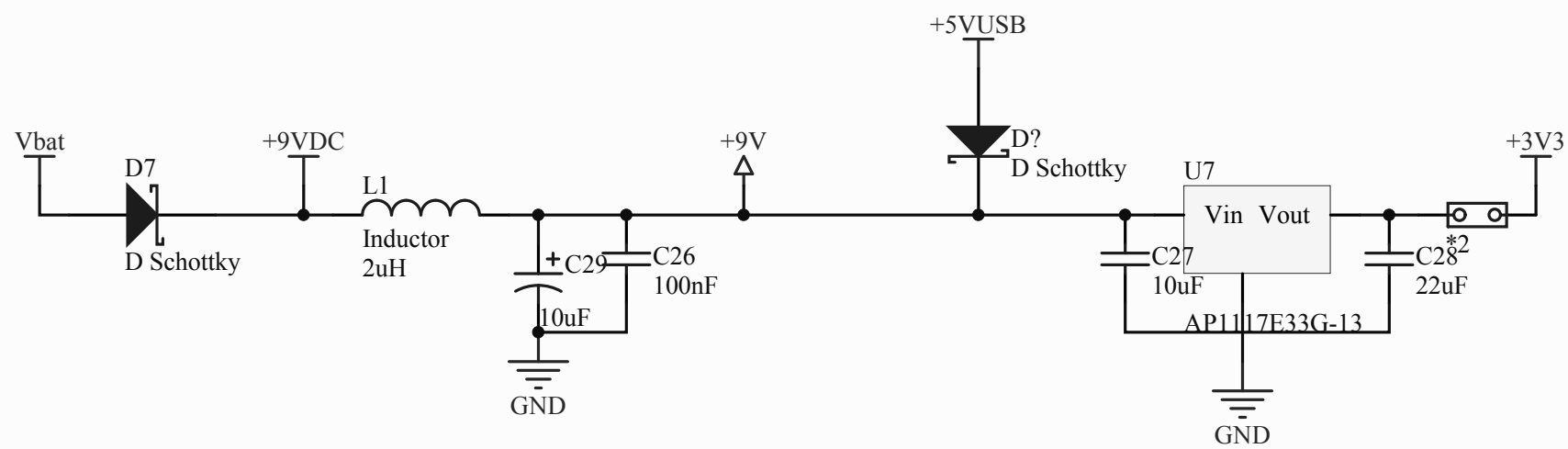
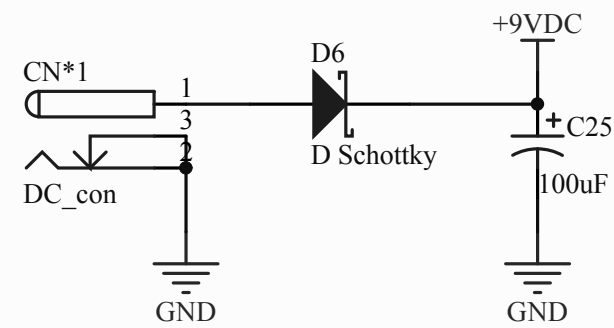
D



Title			MCU		
Size	Number		Revision		
A4					
Date:	6.5.2013		Sheet 5 of		8
File:	D:\Vyuka...\MCU.SchDoc		Drawn By: Jandejsek Rostislav		

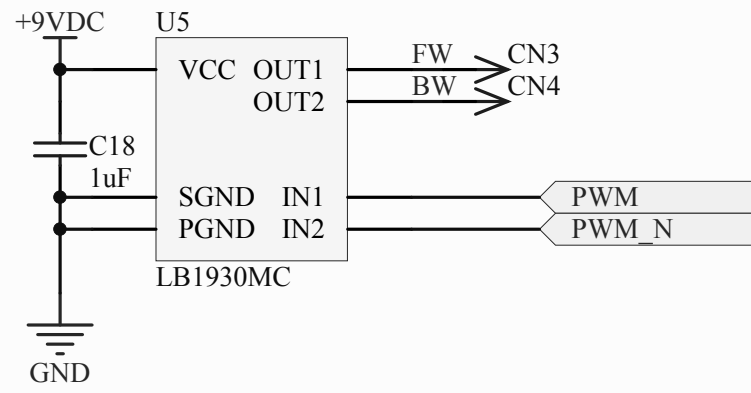


Title SWD Connector		
Size A4	Number	Revision
Date: 6.5.2013	Sheet 6 of 8	
File: D:\Vyuka\...\JTAG.SchDoc	Drawn By: Jandejsek Rostislav	



Title Supply		
Size A4	Number	Revision
Date: 6.5.2013	Sheet 7 of 8	
File: D:\Vyuka...\Supply.SchDoc	Drawn By: Jandejsek Rostislav	

Motor driver



Title		
Motor driver		
Size	Number	Revision
A4		
Date:	6.5.2013	Sheet 8 of 8
File:	D:\Vyuka\...\Motor_driver.SchDoc	Drawn By: Jandejsek Rostislav