

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Petr ŠIMEK**
Osobní číslo: **E10N0167P**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Dopravní elektroinženýrství a autoelektronika**
Název tématu: **Analyzátor komunikace CAN**
Zadávající katedra: **Katedra aplikované elektroniky a telekomunikací**

Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se se specifikacemi ISO 11898 - High-Speed CAN 2.0A a CAN 2.0B a ISO 11519.
2. Na základě požadovaných vlastností navrhnete blokovou strukturu zařízení a vytipujete vhodný řídicí mikrokontrolér na jádře ARM. Současně navrhnete nejvhodnější způsob uživatelského rozhraní z hlediska zadávání a vizualizace hodnot a vytipujete odpovídající komponenty.
3. Realizujte hardwarový návrh zařízení.
4. Do řídicího mikrokontroléru implementujte software realizující požadavky zadání.
5. Zvažte a diskutujte možnost propojení zařízení s osobním počítačem, výměnu dat mezi počítačem a interní pamětí zařízení a případnou interakci s nadřazenou vizualizační a konfigurační aplikací na PC. Diskutujte možnosti konektivity zařízení v rámci sítě internet.

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah pracovní zprávy: **30 - 40 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.

Vedoucí diplomové práce: **Ing. Petr Krist, Ph.D.**

Katedra aplikované elektroniky a telekomunikací

Konzultant diplomové práce: **Ing. Petr Krist, Ph.D.**

Katedra aplikované elektroniky a telekomunikací

Datum zadání diplomové práce: **17. října 2011**

Termín odevzdání diplomové práce: **11. května 2012**

Doc. Ing. Jiří Hammerbauer, Ph.D.

děkan



Doc. Dr. Ing. Vjačeslav Georgiev

vedoucí katedry

V Plzni dne 17. října 2011

Západočeská univerzita v Plzni
Fakulta elektrotechnická
Katedra aplikované elektroniky a telekomunikací

DIPLOMOVÁ PRÁCE

Analyzátor komunikace CAN
CAN communication analyzer

Anotace

Cílem této práce bylo vytvořit funkční prototyp zařízení pro monitorování, ukládání a generování zpráv na sběrnici CAN odpovídající normě ISO 11898-1,2,3. Osvojit si práci s mikropočítači obsahující jádro ARM a naučit se, jak jej reálně použít v zapojení.

Klíčová slova

ARM coretex-M3, CAN, Ethernet, USB-OTG, SD/MMC-flash, Nokia LCD.

Annotation

The purpose of this work is a functional prototype of the CAN-bus analyzer and learn to work on ARM microcomputer implemented in a real circuit. The major idea of this equipment is save and generate CAN messages. This device must comply with ISO 11898-1,2,3.

Key words

ARM coretex-M3, CAN, Ethernet, USB-OTG, SD-flash, Nokia LCD.

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr magisterského 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 práce.

V Plzni dne 8. května 2012

Petr Šimek

Poděkování

Rád bych tímto poděkoval vedoucímu diplomové práce panu Ing. Petru Kristovi za poskytnuté rady a odborné vedení celé práce.

Obsah

1	Úvod.....	9
2	CAN – Controller Area Network.....	10
2.1	Spojová vrstva.....	11
2.1.1	LLC (Logical Link Control)	11
2.1.2	MAC (Medium Access Control).....	12
2.1.3	Kódování vysílaných dat.....	12
2.1.4	Přístupová metoda CSMA/AMP.....	13
2.1.5	Druhy chyb.....	14
2.1.6	Rámec CAN 2.0A & B	14
2.2	Fyzická vrstva	15
2.2.1	Rozhraní pro připojení budiče linky	15
2.2.2	Budič high speed	16
2.2.3	Budič low speed	17
3	Realizace	19
3.1	Blokové zapojení.....	19
3.2	Volby jednotlivých komponent.....	20
3.2.1	Volba Mikropočítače.....	20
3.2.2	Volba LCD	20
3.2.3	Volba paměťového média	21
3.2.4	Volba napájecího zdroje.....	21
3.2.5	Volba budičů CAN.....	22
3.3	Návrh.....	23
3.3.1	CAN	23
3.3.2	Ethernet	24
3.3.3	USB	25
3.3.4	Displej	26
3.3.5	SD/MMC.....	27
3.3.6	Napájecí zdroj	27
3.4	Plošný spoj	28
3.5	Návod k obsluze.....	29
3.5.1	Nákresy zařízení.....	29
3.5.2	CAN	31
3.5.3	Ethernet	31
3.5.4	USB.....	32
3.5.5	SD/MMC.....	32
3.5.6	USART.....	32
3.5.7	SPI.....	33
3.5.8	I2C.....	33
3.5.9	JTAG/SWD.....	33
3.5.10	Rozšiřující port EXP	34
3.5.11	Analogový port ANALOG.....	34
3.5.12	Propojky	35
3.5.13	Hlavní přepínač	36
3.5.14	Obsluha programu.....	36

3.6	Program.....	42
3.6.1	Konfigurační soubory	42
3.6.2	Datové soubory	46
3.6.3	Editační program.....	48
3.6.4	Struktura projektu	48
3.6.5	Popis funkcí.....	50
3.7	Vývojové prostředí.....	54
3.7.1	Nastavení parametrů kompilátoru.....	54
3.7.2	Keil.....	54
3.7.3	Altium	55
3.7.4	Raid	55
3.7.5	Atollic.....	55
3.7.6	Eclipse + GCC + OpenOCD.....	55
4	Pokračování vývoje	56
5	Ekonomické zhodnocení.....	57
6	Závěr	60
7	Literatura.....	61
8	Přílohy	67

Seznam použitých zkratek

- CAN – Controller Area Network (komunikační sběrnice)
- LIN – Local Interconnect Network (komunikační sběrnice)
- RAM – Random Access Memory (paměť)
- USART – Universal Synchronous Asynchronous Receiver Transmitter
- SPI – Serial Peripheral Interface (synchronní sériová sběrnice)
- MOSI – Master Output Slave Input (signál rozhraní SPI)
- MISO – Master Input Slave Output (signál rozhraní SPI)
- SCLK – Serial Clock (hodinový signál)
- I2S – Integrated Interchip Sound (komunikační sběrnice pro zvukové kodeky)
- I2C – Inter Integrated Circuit (komunikační sběrnice, také možno značit IIC)
- MCK – Master Clock (hodinový signál pro sběrnici I2S)
- MII – Media Independent Interface (komunikační rozhraní budiče Ethernet)
- RMII – Reduced Media Independent Interface (komunikační rozhraní budiče Ethernet)
- MMC – Multi Media Card (paměťové médium)
- SD – Secure Digital (paměťové médium)
- SDIO – Secure Digital Input Output (komunikační rozhraní pro karty SD)
- DMA – Direct Memory Access (komunikační kanál pro přímý přístup do paměti)
- JTAG – Joint Test Action Group (sériové programovací a ladící rozhraní)
- SWD – Serial Wire Debug (sériové programovací a ladící rozhraní)
- FSMC - Flexible static memory controller (komunikační rozhraní pro externí paměť)
- PLL – Phase Lock Loop (smyčka fázového závěsu)
- T_q – Time quantum (časová úseka)
- USB – Universal Serial Bus (sériová sběrnice)
- OTG – On-The-Go (funkční režim USB)
- UTMI – USB 2.0 transceiver macrocell interface (rozhraní pro budič USB 2.0)
- HID – human interface device (třída zařízení USB)
- LCD – Liquid Crystal Display (zobrazovací jednotka)
- SD – Secure Digital (paměťové médium)
- MMC – Multi Media Card (paměťové médium)
- LI-ION – Lithium-Ion (druh baterie)
- LAN – Local Area Network (počítačová síť)
- CRC – Cyclic redundancy check (zabezpečovací kód)

A/D – Analog / Digital (analogově – digitální)
ARM – Advanced Risc Machine (druh jádra mikropočítače)
RISC – Reduced Instruction Set Computer (architektura procesoru)
FAT – File Allocation Table (druh souborového systému)
PC – Personal Computer (osobní počítač)
ISO – International Standard Organisation (organizace vytvářející normy)
DLL – data link layer (vrstva modelu ISO-OSI)
LLC – logical link layer (vrstva modelu ISO-OSI)
MAC – medium access kontrol (vrstva modelu ISO-OSI)
MDI – medium dependant interface (vrstva modelu ISO-OSI)
CSMA/AMP – Carrier Sense Multiple Access – Arbitration on Message Prioriti
ART – Adaptive real-time memory Accelerator (urychlovač přístupu do paměti)
GPL – General Public License (licence pro volné šíření informací také GNU)
SEPIC – Single Ended Primary Inductor Converter (topologie pulzního zdroje)
ID – Identification (identifikátor zprávy)
LDO – Low-dropout regulator (napěťový stabilizátor)
LED – Light-Emitting Diode (zdroj světla)
DPS – Deska Plošného Spoje
GCC – GNU Compiler Collection (volný překladač kódu C/C++)
OCD – On-Chip Debugger (nástroj pro ladění kódu uvnitř mikropočítače)

1 Úvod

S rozvojem elektroniky v užitkových a osobních automobilech dochází k požadavku členit systémy v automobilech do jednotlivých sekcí podle důležitosti a požadavků na spolehlivost. Zde vyvstává nutnost komunikace mezi dílčími systémy za účelem monitorování jejich funkce, nastavování parametrů kalibrace a v první řadě distribuce proměnných hodnot. Dnešní dopravní prostředky, ale i distribuované průmyslové řídicí systémy, nejsou již dávno jen o předávání samostatných jednoduchých binárních a analogových hodnot. Trendem je systém rozčlenit do dílčích částí, které se samostatně starají o daný úsek na základě vnějších podnětů. Zde nastává nutnost komunikace mezi částmi takovéhoto systému. Pro tyto účely slouží nej-různější datové sběrnice, po nichž dochází k distribuci potřebných hodnot. Z tohoto důvodu je třeba klást důraz i na spolehlivost a rychlost těchto komunikačních tras. Dnes již nestačí znát spolehlivost dané části systému, protože je pouhou dílčí částí celé technologie, kterou řídí. S masivním rozvojem sběrnice CAN je třeba monitorování přenášených dat a četnost chyb. Takováto síť je páteří celého systému a narušení její integrity ve většině případů vyvolá kolizi cele řízené technologie a to je v mnoha případech nepřijatelné, jedná-li se o bezpečnostní složku celého systému. V mnoha případech je velice těžké odhalit špatnou funkci technologie způsobenou chybou komunikace. Za tímto účelem vznikla moje diplomová práce. Jejím zá-měrem je zaznamenávat datovou komunikaci na sběrnici CAN v reálném čase. Jedná se o samostatné zařízení s dvojicí komunikačních linek CAN podporující režim high speed a low speed. Takto získaná data lze posléze použít pro vyhodnocení funkce celého řídicího systému a k jeho opravě. Tato jednotka by měla být schopna odpovídat na předem připravené dotazy nebo libovolně generovat zprávy na danou sběrnici CAN, popřípadě reprodukovat komunika-ci celého systému. Tato funkce je například vhodná k ověření stability již funkčního systému, v němž se vyskytne poškozená jednotka.

V první fázi vývoje je třeba vytvořit blokové schéma dle zadaných parametrů práce a vybrat a vytipovat vhodné komponenty pro konstrukci celého systému.

V druhé fázi je třeba vytvořit fyzické schéma, podle kterého vytvoříme návrh plošného spoje.

V třetí fázi je třeba vytvořit programové vybavení mikropočítače, které bude implemen-tovat jednotlivé funkce dle zadaných požadavků.

2 CAN – Controller Area Network

Jedná se o sériovou sběrnici vyvinutou primárně pro automobilový průmysl společností Robert BOSCH GmbH. Je navržena s ohledem na vysokou spolehlivost požadovanou pro základní bezpečnostní funkce vozu jako jsou air-bagy, řízení motoru, trakce a brzd. Postupem času se tato komunikační sběrnice rozšířila i do jiných odvětví, zejména průmyslové výroby, a ve spojení s nadstavbovými vyššími protokoly, jako je CANopen nebo DeviceNet, se stává vhodnou a dobře zabezpečenou variantou pro komunikaci mezi dílčími částmi obecného distribuovaného řídicího systému. Podpora režimu časového spouštění umožňuje tuto sběrnici využívat ve spojení s výše uvedeným protokolem CANopen i pro řízení v reálném čase.

Sběrnice CAN je definována normou ISO 11898, jež je rozdělena do 5 dílčích částí, v nichž jsou specifikovány jednotlivé vlastnosti, funkce a konfigurace. Popis norem je uveden v Tab. 2-1. Čerpáno ze zdroje [22].

ISO 11898	
Část 1 Poslední revize 2003	Specifikace vrstev DLL, LLC a MAC
Část 2 Poslední revize 2003	Specifikace CAN high speed, MDI
Část 3 Poslední revize 2006	Specifikace CAN low speed, fault-tolerant MDI
Část 4 Poslední revize 2004	Specifikace Time-triggered provozu sběrnice
Část 5 Poslední revize 2007	Specifikace řízení spotřeby pro CAN high speed

Tab. 2-1-Seznam ISO norem

Dále následuje popis jednotlivých vrstev uvedených v normách ISO11898-1 společných pro specifikaci CAN 2.0 A i B v provedení jak low speed tak high speed.

2.1 Spojová vrstva

Sem patří dílčí vrstva LLC a MAC. Jejich úkolem je sestavit a zabezpečit rámeček obsahující informace určené k přenosu, který se pomocí nižších vrstev dále zpracovává.

2.1.1 LLC (Logical Link Control)

Tato část má za úkol vytvářet datový rámeček obsahující potřebné náležitosti dle normy. Takto sestavený rámeček se dále předává vrstvě MAC, jež ho dále zpracovává. Vrstva LLC vytváří 2 základní druhy rámečků (znázorněny na Obr. 2.1.1-1 a Obr. 2.1.1-2). Při činnosti tato vrstva generuje nadřazeným vrstvám příznak jako je probíhající přenos, přijat datový rámeček nebo dotaz na data atd....

Rámeček musí obsahovat následující části podle jeho druhu:

- ID - Identifikátor
- DLC - Délka datového pole v bajtech
- DATA - Datové pole 1-8 bajtů



Obr. 2.1.1-1-Datový rámeček



Obr. 2.1.1-2- Rámeček s žádostí o data

Pole **ID** obsahuje identifikátor zprávy určující její prioritu. Podle verze CAN 2.0A je identifikátor dlouhý 11bitů. Pro CAN 2.0B je dlouhý 29bitů a je rozdělen na 2 části po 11bitech a 18bitech.

Pole **DLC** udává počet datových bajtů ve zprávě. Je dlouhý 4 bity.

Pole **Data** obsahuje přenášenou informaci a je dlouhý 1-8 bajtů.

2.1.2 MAC (Medium Access Control)

Jedná se o část zpracovávající data z vrstvy LLC, která jsou zde přetvářena do podoby, v níž jsou potom interpretována fyzickou vrstvou. Probíhá zde zpracování a zabezpečování dat. Informace přijaté z vrstvy LLC se rozšíří o redundantní bezpečnostní a řídicí složku a vznikne tak nový rámec, který je na Obr. 2.1.2-1 a musí obsahovat níže uvedené části.

- SOF – Startovací bit rámce
- AF – Arbitrážní pole
- CF – Kontrolní pole
- DATA – datové pole
- CRC – kontrolní zabezpečovací součet
- ACK – potvrzení korektního příjmu protistanic
- EOF – Zakončení rámce

SOF	AF	CF	DATA	CRC	ACK	EOF
-----	----	----	------	-----	-----	-----

Obr. 2.1.2-1-Rámec vrstvy MAC

Startovací bit rámce je reprezentován dominantní (reprezentováno logickým stavem 0) úrovní a vnitřní časovací obvod jej využívá k synchronizaci.

Pole **arbitrážní** určuje, která stanice má právo na vysílání. Používá se metoda CSMA/AMP (Multiple Access with Arbitration on Message Priority). Součástí pole je identifikační bit IDE, který určuje, zda se jedná o standardní (0 = dominant) nebo prodloužený (1 = recessive) rámec.

Pole **kontrolní** obsahuje údaj o délce přenášených dat.

Pole **datové** obsahuje 1-8 bajtů datové zprávy.

Pole **kontrolního součtu** obsahuje 16-bitové číslo zabezpečující datové kontrolní a arbitrážní pole. Generující polynom má tvar $X^{15}+X^{14}+X^{10}+X^8+X^7+X^4+X^3+1$

Pole **potvrzovací** je dlouhé 2 bity a na vysílací straně je reprezentováno recesivní úrovní. Je-li zpráva přijata alespoň jedním uzlem správně, tento uzel nastaví v potvrzovacím poli dominantní úroveň, a tím vysílači potvrdí správnost příjmu.

Pole **zakončovací** je reprezentováno 7 bity recesivní úrovní.

2.1.3 Kódování vysílaných dat

Používá se zde metoda bit stuffing. Tento způsob kódování přenášených dat, je zvolen z důvodu nutnosti re-synchronizace časovacích obvodů přijímače. Jedná se o vkládání inverz-

ních bitů do vysílaného řetězce v případě 5 a více po sobě jdoucích stejných bitů. Tato metoda je znázorněna v Tab. 2.1.3-1.

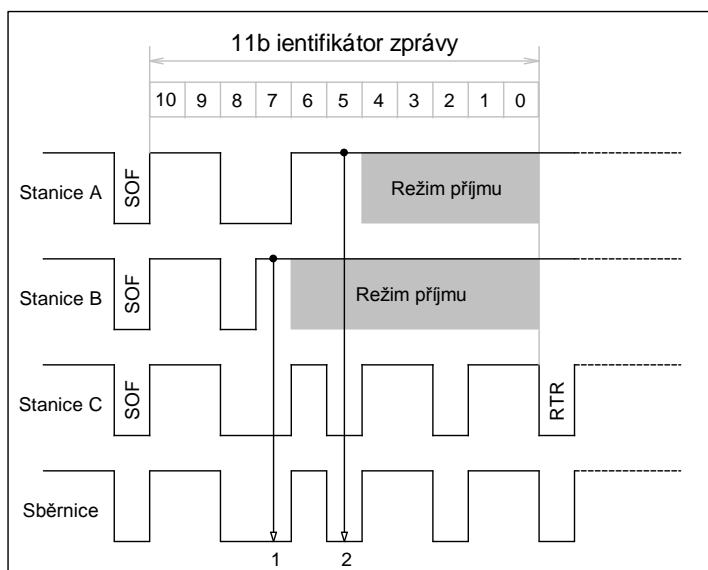
Destuffed	00011111 10001011111 100000 000
Stuffed	000111111 0 10001011111 0 100000 1 000

Tab. 2.1.3-1-Ukázka stuffingu

O toto kódování se stará vrstva MAC a v tomto formátu již generuje data na výstupní bránu Tx. Po přijetí zprávy jsou bity automaticky odstraněny vrstvou MAC.

2.1.4 Přístupová metoda CSMA/AMP

CAN využívá bezkolizního přístupu k přenosovému médium, což zaručuje minimální prodlevy v komunikaci. Tato přístupová metoda zaručuje, že stanice vysílající zprávu s nejvyšší prioritou bude moci odeslat data. Ostatní stanice počkají, než komunikace skončí. Poté opět přistupují ke sběrnici a stanice vysílající zprávu s nejvyšší prioritou má opět přednost a ostatní uzly komunikaci na sběrnici ukončí. Ukázka principu arbitráže je znázorněna na Obr. 2.1.4-1.



Obr. 2.1.4-1-Princip CSMA-AMP

nicí přebírá uzel C. Po ukončení přenosu na sběrnici přistupuje stanice A a nakonec stanice B. To platí za předpokladu, že na sběrnici nepřistoupí jiná stanice a nezačne vysílat zprávu s vyšší prioritou.

Obr. 2.1.4-1 znázorňuje provoz na sběrnici v případě pokusu o komunikaci 3 uzlů najednou. Stanice identifikátor vysílá od bitu s nejvyšší vahou. Zpráva stanice B má nejnižší prioritu a při detekci dominantního stavu v době, kdy se má vysílat recesivní úroveň, přechází do režimu odposlechu sběrnice. To samé platí pro stanici A, komunikaci na sběr-

2.1.5 Druhy chyb

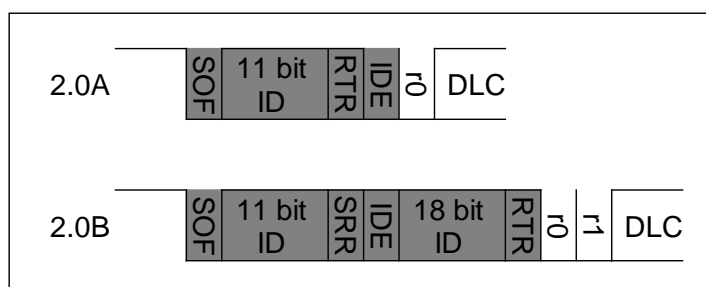
Při komunikaci však může docházet k chybám a vrstva MAC rozlišuje chyby uvedené v Tab. 2.1.5-1.

Chyba (error)	Význam
Bit	V případě, že je rozpor mezi vysílanou a monitorovanou hodnotou.
Stuff	Pokud je víc jak 5 stejných za sebou jdoucích bitů.
CRC	Generuje přijímač v případě, že je poškozený rámeček a neshoduje se CRC.
Form	Generuje se v případě chybného tvaru některého z fixních polí v rámci.
Acknowledgement	Generuje řadič, pokud v poli ACK nepotvrdí žádná stanice příjem.

Tab. 2.1.5-1-Druhy chyb

2.1.6 Rámec CAN 2.0A & B

Jedná se o rámce s odlišnou specifikací arbitrážního pole. Standardní specifikace 2.0A s identifikátorem dlouhým 11bitů a specifikace 2.0B s prodlouženým 29bitovým identifikátorem. Rozdíl je znázorněn na Obr. 2.1.6-1 .



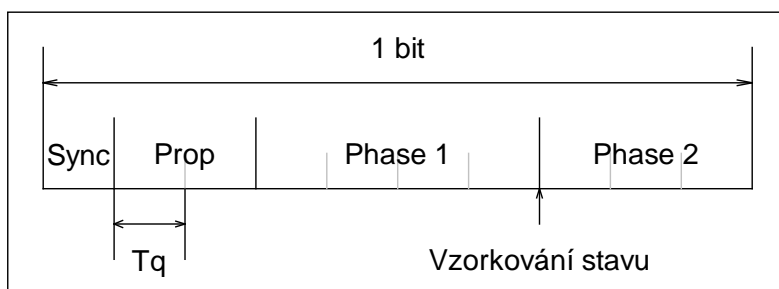
Obr. 2.1.6-1-Ukázka rámce 2.0A & B čerpáno z [19].

- SOF – (Start Of Frame) Startovací bit.
- ID – Pole identifikátoru.
- RTR – (Remote Transmission Request) Rozlišovací bit určující, zda se jedná o data nebo o dotaz na data.
- SRR – (Substitute Remote Request) Přenáší se jako recesivní z toho důvodu, že standardní rámeček CAN2.0A má vyšší prioritu než rozšířený rámeček CAN 2.0B.
- IDE – (Identifier Extension) Bit rozlišující standardní a prodloužený rámeček.
- r0-1 – rezervní bity pro budoucí použití.
- DLC – (Data Length) Pole určující délku přenášených dat.

Zbytek rámce je stejný pro oba standardy.

2.2 Fyzická vrstva

Jejím úkolem je příjem a vysílání datových bytů, jejich přesné časování a synchronizace. Datové rámce vytvořené a zakódované předchozí vrstvou se vloží do zásobníku, odkud jsou v přesné časové okamžiky vysílány na výstupní bránu řadiče. O posuv vysílaných dat se starají časovací obvody. Při příjmu je postup opačný a časovací obvod je složitější, je zde třeba synchronizovat vzorkovací obvod, aby nedocházelo ke čtení hodnoty v nevhodný okamžik.



Obr. 2.2-1-Časování

Při tvorbě nebo příjmu datového bitu vrstvou PLS se bit skládá z časových kvant T_q (maximum 25). Z těchto částí jsou tvořeny 4 hlavní segmenty bitu.

- Sync – Časový úsek sloužící pro synchronizaci časovacích obvodů ($1T_q$).
- Prop – Kompenzuje časové zpoždění při fyzickém šíření signálů sběrnic ($1-8T_q$).
- Phase 1 & 2 – Kompenzují fázovou chybu. Velikost segmentů se mění při re-synchronizaci a posouvá polohu vzorkování Phase1 ($1-8T_q$), Phase2 ($Phase1+2t_q$).

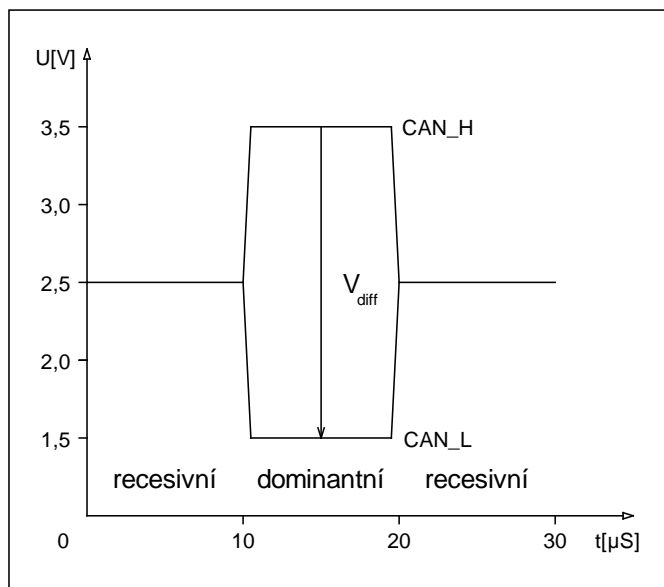
2.2.1 Rozhraní pro připojení budiče linky

Datový výstup řadiče CAN je reprezentován linkami Rx a Tx. Datové toky jsou zde rozděleny a jsou pouze jednosměrné. Napěťová reprezentace úrovní datových bitů je shodná s použitou technologií mikro počítače.

Na toto rozhraní se připojují budiče datového vedení, které na daném vedení generují napěťové úrovně odpovídající dominantnímu a recesivnímu stavu. Norma ISO 11898-2 a 3 stanovuje napěťové poměry a maximální přípustné komunikační rychlosti pro daný typ budiče. Definiuje parametry vedení a jejich zakončení.

2.2.2 Budič high speed

Jedná se o diferenční budič sériové sběrnice. Svorčky pro připojení k vodiči sběrnice mají označení high a low. Diferenční napětí mezi CAN_H a CAN_L určuje recesivní a dominantní stav sběrnice a je znázorněno na Obr. 2.2.2-1. Norma stanovuje délku sběrnice na 40 m při komunikační rychlosti 1 Mbit/s a maximální délce odbočky 30 cm. Čerpáno ze zdroje [2]



Obr. 2.2.2-1-Napět'ové reprezentace stavů high speed budiče

Na Obr. 2.2.2-1 je znázorněná napět'ová reprezentace stavů na sběrnici. V recesivním stavu je na obou linkách vedení shodné napětí. Ve stavu dominantním je na lince diferenční napětí $V_{diff} > 1,2V$. Toto jsou ideální hodnoty toleranční pásma. Pro recesivní a dominantní stav jsou uvedeny v Tab. 2.2.2-1 a Tab. 2.2.2-2.

Recesivní stav						
Parametr	Napětí	Jednotka	Hodnota			Poznámka
			Min.	Nom.	Max.	
Souhlasné	V_{CAN_H}	V		2,5	7,0	Měřeno proti zemi každého z uzlů.
	V_{CAN_L}	V	-2,0	2,5		
Diferenční	V_{diff}	mV	-120	0	12	Měřeno na linkách vedení uzlů.

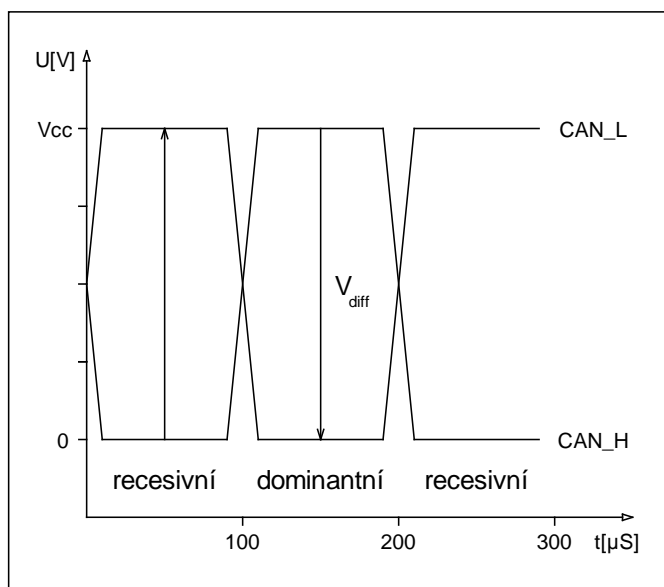
Tab. 2.2.2-1-Definice recesivního stavu high speed budiče

Dominantní stav						
Parametr	Napětí	Jednotka	Hodnota			Poznámka
			Min.	Nom.	Max.	
Souhlasné	V_{CAN_H}	V		3,5	7,0	Měřeno proti zemi každého z uzlů.
	V_{CAN_L}	V	-2,0	1,5		
Diferenční	V_{diff}	V	1,2	2,0	3,0	Měřeno na linkách vedení uzlů.

Tab. 2.2.2-2-Definice dominantního stavu high speed budiče

2.2.3 Budič low speed

V poslední revizi normy ISO 11898-3 je tento budič označován jako necitlivý vůči chybám sběrnice (fault tolerant). Budič je konstruován tak, že i při přerušení, zkratu na kostru nebo na plusové napětí baterie jedné z linek, musí zůstat zachována schopnost komunikace. Budič by měl tento chybový stav linky indikovat. Čerpáno ze zdroje [3]



Obr. 2.2.3-1 znázorňuje reprezentaci stavů sběrnice. V Tab. 2.2.3-1 jsou toleranční pásma stavů. V Tab. 2.2.3-2 jsou stavy pro přechod do režimu snížené spotřeby a normálního režimu. V Tab. 2.2.3-3 je režim pro detekci dotazu na přechod do režimu snížené spotřeby.

Obr. 2.2.3-1-Napět'ová reprezentace stavů low speed budiče

			Recesivní stav			Dominantní stav		
Parametr	Napětí	Jednotka	Hodnota			Hodnota		
			Min.	Nom.	Max.	Min.	Nom.	Max.
Sběrnice	V_{CAN_L}	V	$V_{CC} - 0,3$	-	-	-	-	1,4
	V_{CAN_H}	V	-	-	0,3	$V_{CC} - 1,4$	-	-
Diferenční	V_{diff}	V	$-V_{CC}$	-	$-V_{CC}+0,6$	$V_{CC} - 2,8$	-	V_{CC}

Tab. 2.2.3-1-Definice dominantního a recesivního stavu low speed

			Režim snížené spotřeby energie			Detekce normálního režimu D → R		
Parametr	Napětí	Jednotka	Hodnota			Hodnota		
			Min.	Nom.	Max.	Min.	Nom.	Max.
Sběrnice	V_{CAN_L}	V	$V_{Bat} - 0,3$	-	-	2,8	-	3,5
	V_{CAN_H}	V	-	-	0,3	1,5	-	2,3
Diferenční	V_{diff}	V	-	-	-	-3,5	-	-2,6

Tab. 2.2.3-2- Napět'ové hladiny v různých stavech low speed budiče

			Detekce probuzení jednotky		
Parametr	Napětí	Jednotka	Hodnota		
			Min.	Nom.	Max.
Hladina pro probuzení	$V_{th(wake)L}$	V	2,5	3,2	3,9
	$V_{th(wake)H}$	V	1,1	1,8	2,5
Diferenční	V_{diff}	V	0,8	1,4	-

Tab. 2.2.3-3-Napět'ové hladiny v různých stavech low speed budiče pokračování

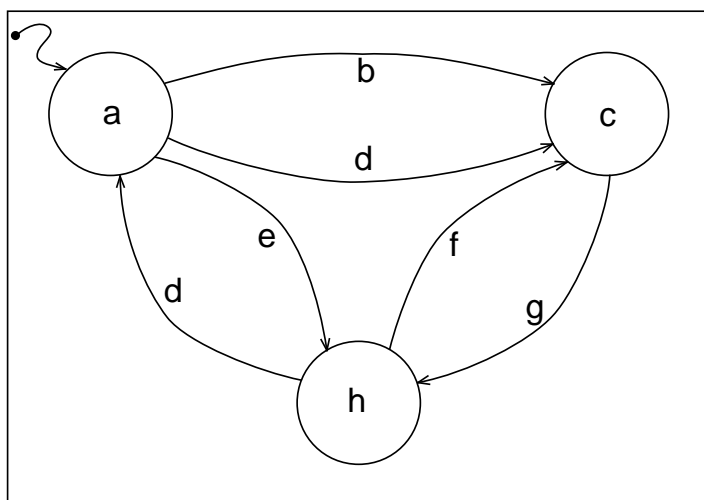


Diagram popisující funkci přechodů mezi jednotlivými stavy budiče je znázorněn na Obr. 2.2.3-2. Tab. 2.2.3-4 obsahuje popis jednotlivých stavů. V Tab. 2.2.3-5 jsou uvedeny komunikační režimy.

Obr. 2.2.3-2- Stavový diagram low speed budiče

Stav	a/0	Normální režim bez chyb na vedení sběrnice	e	$CAN_H \rightarrow VCC_{BATN/LP}$ nebo $UBATN/LP$
	b	$CAN_L \rightarrow U_{BATN}$ nebo $GNDN/LP$	f	Nenastal ani stav b nebo e
	c/E1	Detekována kolize na CAN_L	g	$CAN_H \rightarrow VCC_{BATN/LP}$ nebo $UBATN/LP$
	d	Bez kolize	h/E2	Detekována chyba na CAN_H

Tab. 2.2.3-4-Popis stavů low speed budiče

Stav		Řízení	Řadič	Zakončení
Normální režim	0	Vše sepnuto	Diferenciální režim	$H \rightarrow GND, L \rightarrow V_{CC}$
	E1	CAN_L vypnuto	Jednovodičový režim CAN_H	$H \rightarrow GND, L \rightarrow WV_{CC}$
	E2	CAN_H vypnuto	Jednovodičový režim CAN_L	$H \rightarrow WGND, L \rightarrow V_{CC}$
Režim snížené spotřeby	0	Vše sepnuto	Redukce rozpoznání chyby	$H \rightarrow GND, L \rightarrow V_{Bat}$
	E1	Vše rozepnuto		$H \rightarrow GND, L \rightarrow \text{odpoj.}$
	E2	Vše rozepnuto		$H \rightarrow \text{odpoj.}, L \rightarrow V_{CC}$

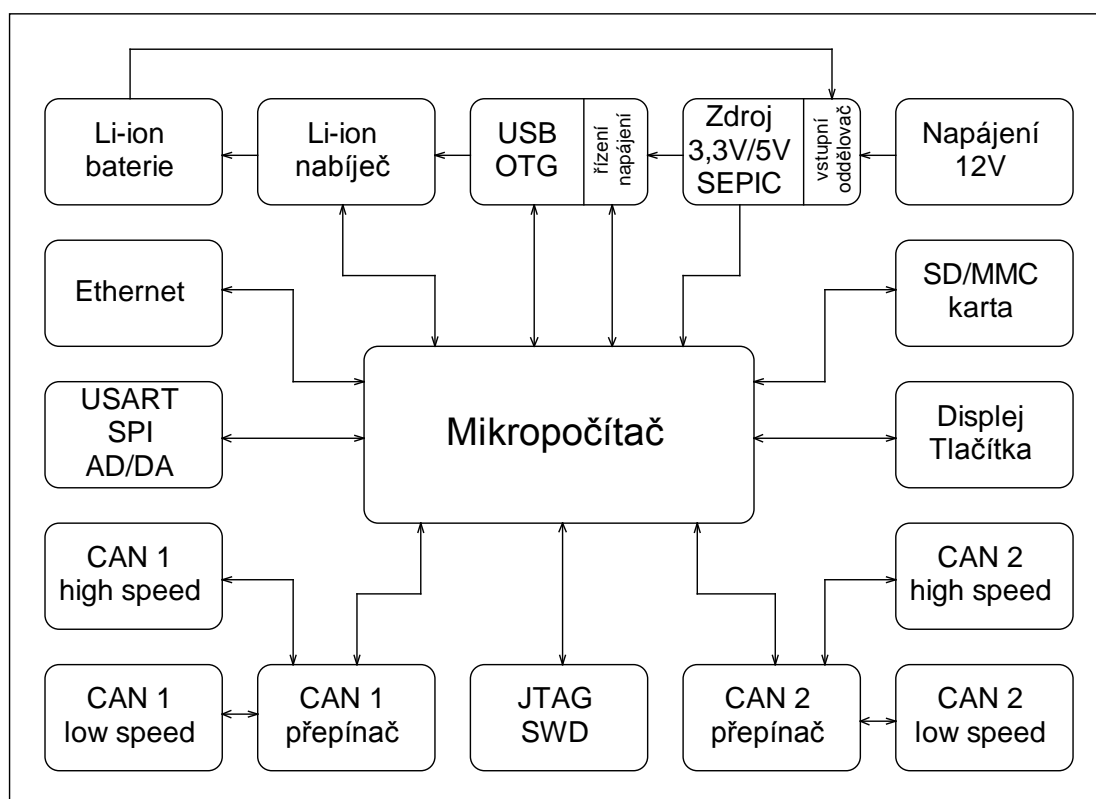
Tab. 2.2.3-5-Seznam komunikačních režimů low speed budiče

3 Realizace

V tomto bodu bylo úkolem vytvořit funkční vzorek. Nyní je třeba zvolit potřebné součásti, vytvořit základní blokové schéma, podle něj navrhnout reálné zapojení, prakticky realizovat konstrukční uspořádání a implementovat obslužný program.

3.1 Blokové zapojení

Na Obr. 3.1-1 je znázorněno úplné blokové schéma, které má 4 základní části: napájecí, komunikační, uživatelskou a řídicí. V sekci návrhu budou podrobněji popsány jednotlivé části a detailněji rozkresleny jednotlivé bloky.



Obr. 3.1-1-Blokové schéma zapojení

3.2 Volby jednotlivých komponent

Na počátku celé práce byla provedena volba všech součástí, které budou potřeba pro realizaci celého zařízení. Tato část by měla rozhodnout o celkovém uspořádání, tvaru celého zařízení a o součástkách použitých pro realizaci. Bude také určovat konečnou cenu celého zařízení.

3.2.1 Volba Mikropočítače

Při volbě mikropočítače byl brán ohled na prvotní požadavek, a to osvojení si práce s mikropočítači s jádrem ARM. Z tohoto důvodu byla zvolena společnost STMicroelectronics, s níž spolupracuje univerzita i vedoucí této práce. Tato společnost nabízí širokou řadu mikropočítačů s jádrem ARM. Pro realizaci byla zvolena řada Cortex-M3 F2, která je primárně určena pro aplikace v reálném čase a za tímto účelem má speciálně navržené jádro a řadič přerušení.

Konkrétní mikropočítač nese označení STM32F217ZG. Disponuje 1 MB programové paměti flash a 128 kB paměti RAM. Čip je vyroben 90 nm technologií a je v pouzdře LQFP144. Pro urychlení běhu programu je jádro vybaveno technologií ART. Mikropočítač pracuje na taktovací frekvenci 120MHz při zpracování 1,25 instrukce na takt. Toto odpovídá teoretickému výpočetnímu výkonu 150 milionů instrukcí za sekundu.

Mikropočítač je vybaven celou řadou komunikačních rozhraní. Důležité jsou zde dva kanály CAN, rozhraní MII/RMII pro budič Ethernetu, rychlé kanály sběrnici USART, SPI, IIC a I2S s podporou DMA, USB OTG nebo rozhraní UTMI pro budič USB high speed a spoustu dalších periférií. Viz dokumentace daného mikropočítače. Naleznete na přiloženém CD v adresáři \\Katalogové listy.

3.2.2 Volba LCD

Součástí zadání je realizace vhodného uživatelského rozhraní a jeho součástí je zobrazovací jednotka. Vzhledem k cenám již hotových řešení, jsem zvolil konstrukci vlastní jednotky. Jako zobrazovač jsem použil náhradní LCD pro NOKIA E51 s rozlišením 320x240 bodů a 24bit hloubkou barev. Tento panel pracuje na jiné napěťové logice než je použitý mikropočítač, proto bylo nutné jej doplnit externími obvody pro převod napěťových úrovní a zdrojovou částí pro napájení řadiče a podsvětlení. Čerpáno ze zdroje [4].

3.2.3 Volba paměťového média

Zařízení je koncipováno jako záznamová jednotka, a protože v této práci nebude navázána na vyšší vizualizační a konfigurační systém, je zde potřeba zachovat kompatibilitu datového úložiště s tímto vyšším systémem. Z tohoto důvodu muselo být médium voleno tak, aby bylo možné uložená data přenést do osobního počítače k dalšímu zpracování. Dalším důvodem volby daného media byla předchozí zkušenost s jeho obsluhou. Proto jako paměť pro data celého zařízení byla zvolena flash paměť SD/MMC, která disponuje dostatečnou rychlostí a kapacitou pro ukládání potřebných dat.

Toto paměťové médium má konfigurovatelné datové rozhraní do dvou režimů. Prvním režimem je 4 bitové paralelní rozhraní SDIO doplněné o řídicí signály. Druhým režimem je sériové rozhraní SPI, které jsem použil ve své práci. Důvodem je jeho jednodušší implementovatelnost. I přesto, že daný mikropočítač disponuje i rozhraním SDIO, bylo z důvodu jednodušší přenositelnosti použito sériové rozhraní SPI. Rozhraní je nakonfigurováno do 8-bitového plně duplexního režimu. Jako první se vysílá bit s nejvyšší hodnotou. Komunikační rychlost je nastavena okolo 5Mb/s. Informace byly čerpány ze zdroje [5].

3.2.4 Volba napájecího zdroje

Napájecí zdroj byl volen s ohledem na požadovaný rozsah napájecího napětí 3 – 16 V. Původně měl zdroj mít výstupní napětí +3,3 V. Toto napětí se změnilo na +5 V z důvodu použití galvanické izolace CAN a USB sběrnice v režimu OTG, kdy je potřeba napájet připojené hostující zařízení.

Typ zdroje je pulzní. Použitá architektura s ohledem na rozsah napájecího napětí je SEPIC. Funkčnost a chování tohoto zdroje byla ověřena v rámci semestrální práce z předmětu KAE/ENZ. Jako regulátor byl použit obvod LM3478, který pracuje v rozsahu 2,98 – 40 V. Důvodem použití této topologie je možnost zdroje přecházet z režimu zvyšujícího do režimu snižujícího pouze změnou střídání spínání. Takovéto chování je nutné z toho důvodu, že generované napětí 5 V se nachází ve zvoleném rozsahu 3-16 V. Tento rozsah je zvolen z důvodu možnosti napájet zařízení z jednoho článku baterie Li-Ion nebo z palubní sítě automobilu. Informace byly čerpány ze zdroje [6].

3.2.5 Volba budičů CAN

Zadáním práce je analyzátor sběrnice CAN, proto jednou z hlavních částí je volba budičů sběrnice. Na trhu je velká škála výrobců polovodičů zabývajících se touto problematikou. Jako první byl vybírán budič high speed a volba padla na obvod MCP 2551 od společnosti Microchip pro jeho cenu, parametry a vyšší dostupnost. Výhodou těchto obvodů je kompatibilita pouzder, to dává možnost přejít na jiného výrobce v případě nevyhovujícího obvodu. Jako budič low speed, byl vybrán obvod TJA 1054 od společnosti NXP. Důvodem je poměrně malá nabídka obvodů na trhu. V Tab. 3.2.5-1 je srovnání jednotlivých budičů high speed.

Budič	TJA1050	SN65HVD251D	MCP2551
Povolené napětí na CAN_H a L	-27 až +40 [V]	-36 až +36 [V]	-42 až +42 [V]

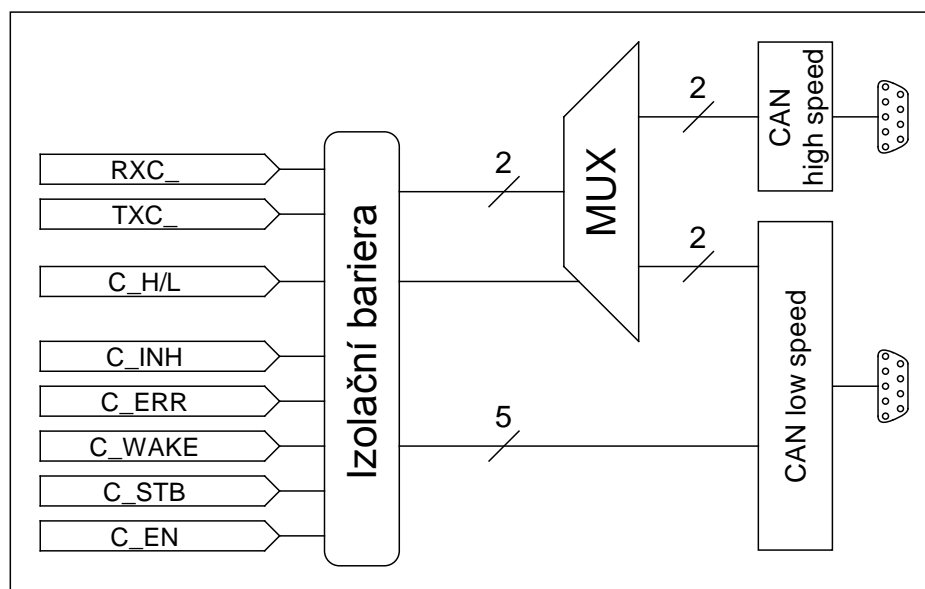
Tab. 3.2.5-1-Porovnání budičů CAN high speed

3.3 Návrh

Dle blokového zapojení vytvoříme reálné zapojení obvodu, podle kterého bude následně navržen plošný spoj. V této části bude komentováno zapojení celého zařízení. Popíše zde konfiguraci a možná nastavení. Konkrétní zapojení níže uvedených vývodů mikropočítače je k nalezení v části 8 Přílohy Hlavní deska sheet 1.

3.3.1 CAN

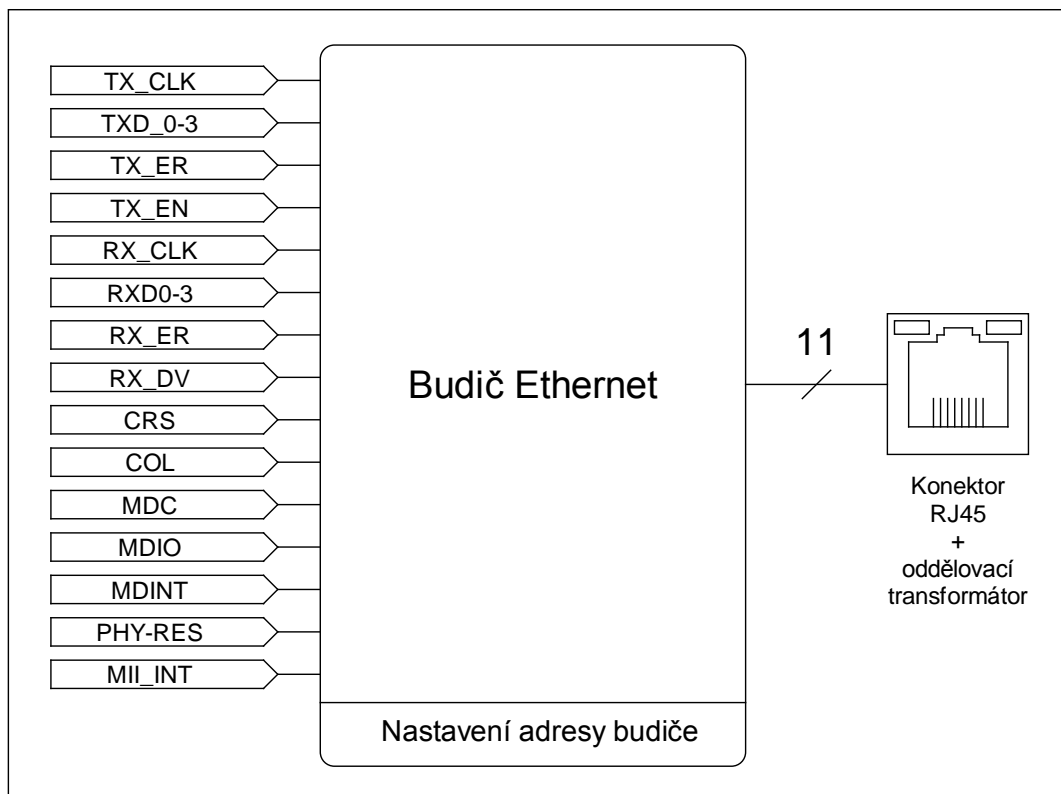
Jedná se o důležitou část celého zapojení. Součástí zadání je kompatibilita se standardy ISO11898-2 a -3. To vyžadovalo nutnost implementace dvou budičů na každý kanál CAN. Při návrhu byl brán i ohled na prostředí, v němž zařízení bude provozováno, a z toho důvodu bylo provedeno galvanické oddělení obou kanálů. Pro plnou kompatibilitu těchto norem, bylo třeba obsluhovat i řídicí signály budiče low speed. Důvodem je umožnění případného přechodu sběrnice do úsporného režimu nebo detekce této žádosti a další stavy rozlišující případné chybové stavy na vedení jako je zkrat nebo rozpojení. V blokovém zapojení na obrázku Obr. 3.3.1-1 je znázorněna principiální funkce a konkrétní zapojení je pak v části 8 Přílohy Hlavní deska – sheet 3-4. Multiplexer slouží pro přepínání signálu Rx a Tx pro budič high a low speed a je ovládán pomocí signálu C_H/L. Ostatní signály pro řízení budiče low speed jsou detailněji popsány v katalogovém listu obvodu TJA 1054 na přiloženém CD v adresáři \\Katalogové listy.



Obr. 3.3.1-1-Blokové schéma CAN

3.3.2 Ethernet

Součástí méj práce je také teoretická rozvaha nad možností připojení tohoto zařízení do sítě Internet. Z tohoto důvodu jsem pro budoucí praktickou realizaci modul navrhnul i s rozhraním Ethernet 10/100 Mb/s. Blokové schéma je na obrázku Obr. 3.3.2-1. Jako budič je použit obvod STE100P, připojen přes rozhraní MII. Vývody připojené k mikropočítači jsou znázorněny v blokovém zapojení na Obr. 3.3.2-1. Konkrétní zapojení je uvedeno v části 8 Přílohy Hlavní deska – sheet 2. Adresa MII budiče je 31 => 0x1F.



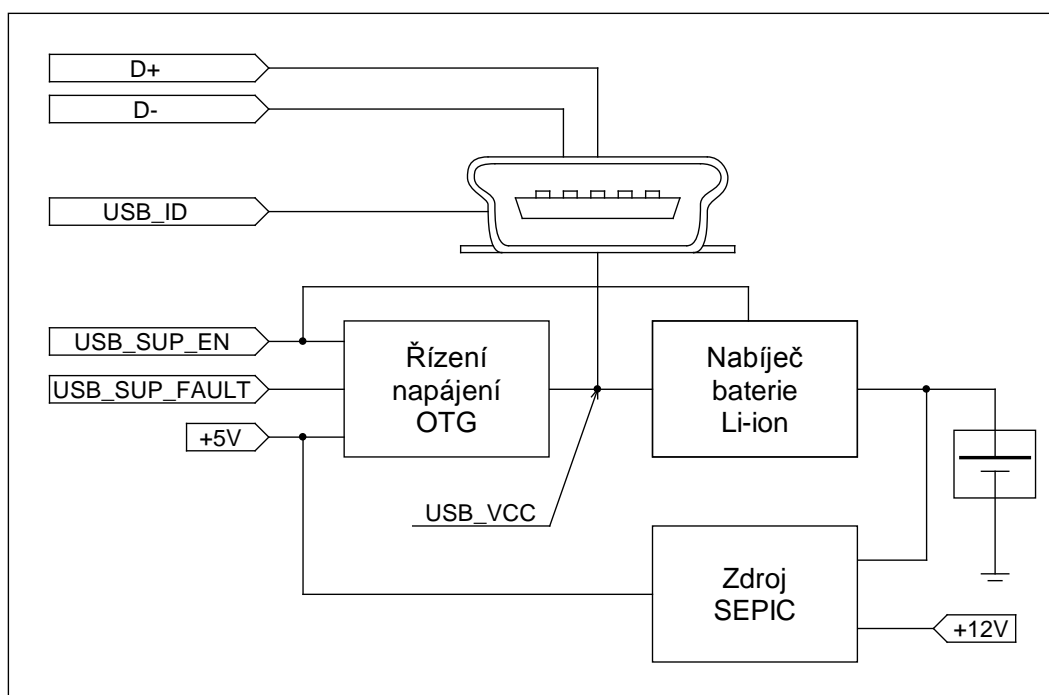
Obr. 3.3.2-1-Blokové schéma Ethernet

3.3.3 USB

Součástí budoucího vývoje je možnost spojení s nadřazenou vizualizační a konfigurační jednotkou a možnost připojování jiných druhů paměťových médií. K tomuto účelu je zvolena sběrnice USB, která je v dnešní době standardem a je na každém osobním počítači. Její blokové zapojení je na Obr. 3.3.3-1 a konkrétní zapojení v části 8 Přílohy Hlavní deska – sheet 5. Použitý mikropočítač podporuje standart OTG (On-The-Go), který je zde implementován. Tento režim provozu umožňuje podle způsobu zapojení kabelu do 5-ti pinového konektoru fungovat v režimu HOST nebo DEVICE. Zařízení, které tuto specifikaci splňuje, musí umožňovat napájet připojené zařízení nebo být schopno provozu při napájení nadřazenou jednotkou. Informace byly čerpány ze zdroje [7].

- HOST – Mikropočítač je jako master a lze k němu připojit libovolné USB zařízení, s kterým umíme komunikovat, jako je standardní protokol mass-storage HID atd.... V tomto režimu je pátý pin ID připojen na zemní potenciál GND.
- DEVICE – Mikropočítač je jako podřízený a může být připojen k nadřazenému systému jako je osobní počítač nebo jiné zařízení v režimu HOST. Pin ID je v tomto režimu jako plovoucí, není zapojen.

Mikropočítač podle stavu pinu ID pozná, zda se má nakonfigurovat jako HOST nebo DEVICE a podle toho zvolí napájecí režim.

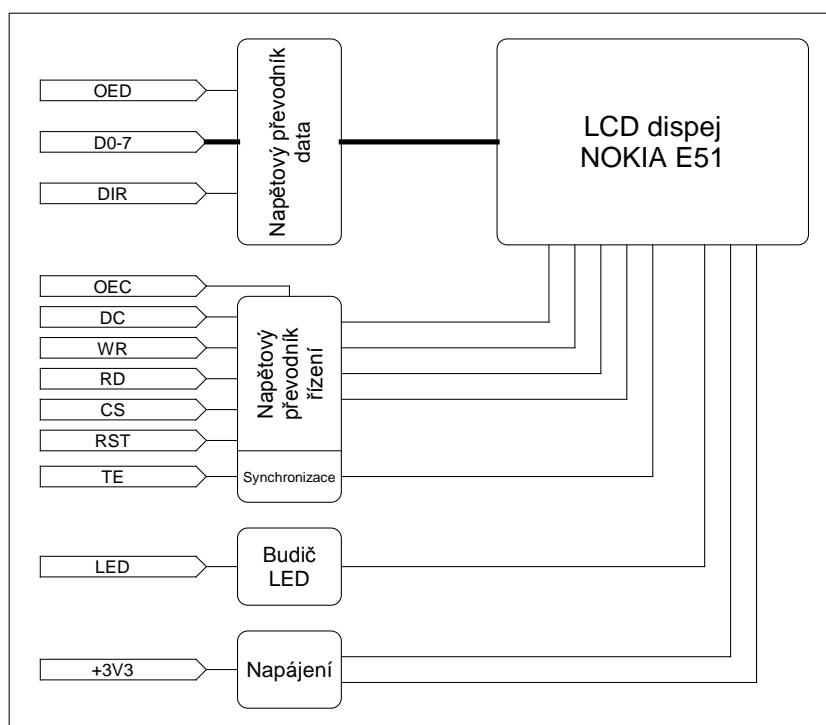


Obr. 3.3.3-1-Blokové schéma USB a dobíjení

3.3.4 Displej

Další částí zadání je vytvořit vhodnou lokální vizualizaci. K tomuto účelu jsem použil barevný LCD panel z telefonu NOKIA E51. Ten pracuje na 1,8 V CMOS logice a analogová část je napájena napětím 2,5 V. Pro displej bylo třeba použít úrovnňový převodník a dva lineární LDO stabilizátory. Dalším potřebným blokem byl obvod pro napájení podsvětlení panelu. Osvětlení je realizováno pomocí 3 vysoce svítivých led diod spojených do série, což vyvolává potřebu vyššího napájecího napětí okolo 10-12 V. Tento problém je řešen pomocí speciálního pulzního regulátoru pro proudové buzení vysoce svítivých LED diod, který zároveň umožňuje pomocí PWM signálu měnit jas.

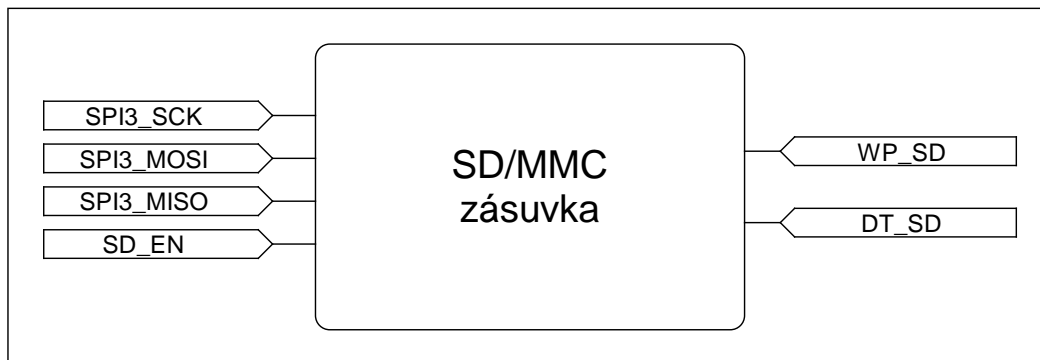
Komunikace s displejem je po 8-mi bitové paralelní sběrnici INTEL 8080. Do tohoto režimu lze nakonfigurovat jednotka FSMC na použitém mikropočítači. Toto zapojení však není použito z důvodu sdílení stejných vývodů pouzdra řadičem CAN a Ethernet, a proto je protokol emulován programově a je o něco pomalejší. Mikropočítač v tomto pouzdru bohužel neumožňuje souběžnou činnost 2 kanálů CAN, Ethernet a FSMC v 8-mi bitovém režimu. Rozhraní pro komunikaci je navíc doplněno o dva signály pro povolení napěťových převodníků a jeden pro řízení datového toku pro zápis a čtení. Konkrétní signály viz blokové zapojení na obrázku Obr. 3.3.4-1 a přesné zapojení je v části 8 Přílohy Displej – sheet 1 – 2. Informace byly čerpány ze zdroje [4].



Obr. 3.3.4-1-Blokové schéma LCD

3.3.5 SD/MMC

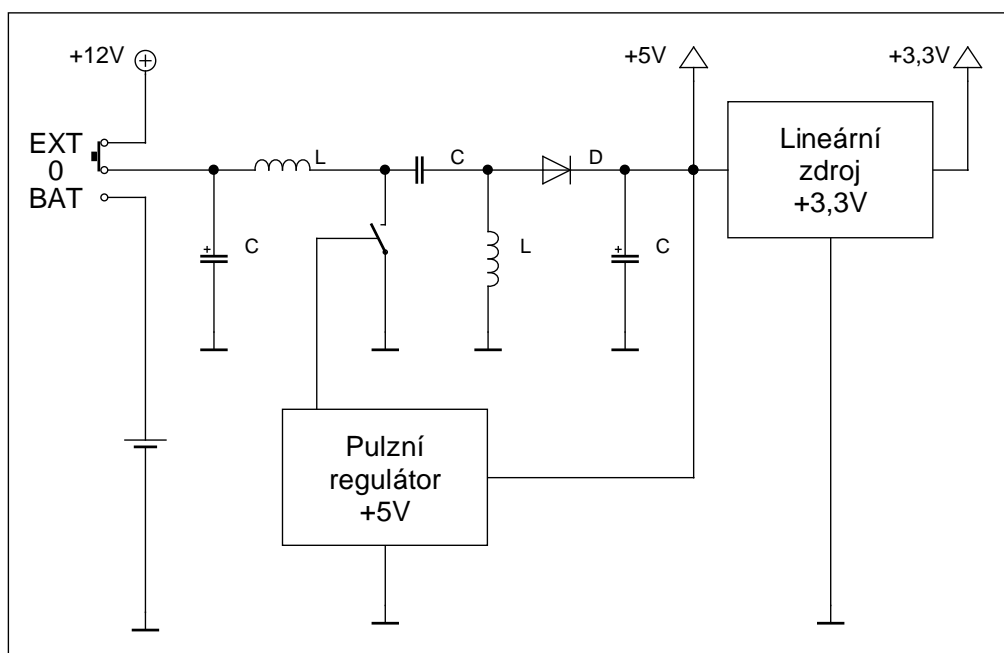
V zadání práce je požadavek na možnost ukládání a čtení dat z paměťového media. K tomuto účelu slouží paměť flash typu SD nebo MMC pracující v komunikačním režimu po sběrnici SPI. Blokové schéma je na Obr. 3.3.5-1, konkrétní zapojení naleznete v části 8 Přílohy Hlavní deska – sheet 6. Informace byly čerpány ze zdroje [8].



Obr. 3.3.5-1-Blokové schéma SD/MMC

3.3.6 Napájecí zdroj

Je zde použit pulzní regulátor s topologií SEPIC. Jeho vstupem je napájení z palubní sítě automobilu a z baterie Li-ion, které jsou od sebe oddělené pomocí přepínače na přední straně zařízení, viz Obr. 3.5.1-2. Pracovní rozsah je 3-16 V. Zdrojová jednotka dává napětí +3,3 V a +5 V. Hlavní části jsou v blokovém zapojení na Obr. 3.3.6-1. Podrobné zapojení naleznete v části 8 Přílohy Hlavní deska – sheet 10. Informace byly čerpány ze zdroje [6].



Obr. 3.3.6-1-Blokové schéma Zdroje

3.4 Plošný spoj

Součástí práce je výroba celého zařízení, které je rozděleno na 3 části, z nichž každá má svoje dílčí zapojení a plošný spoj. Obrazce a osazovací plány spojů naleznete v části 8 Přílohy Hlavní deska DPS + osazovací plán, Panel DPS + osazovací plán a displej DPS + osazovací plán. Podrobnější popis konektoru a nastavení propojek je uveden v sekci 3.5 Návod k obsluze. Pro návrh plošného spoje a tvorbu schémat byl použit program Eagle verze 5.7.0. Výrobní data všech plošných spojů jsou k nalezení na přiloženém CD v adresáři \\Výrobní data.

Nejdůležitější, je hlavní deska, na které jsou komunikační periferie a řídicí mikropočítač. Spoj je navržen jako dvouvrstvý. Minimální izolační vzdálenosti jsou větší než 150 μ m a vrtané otvory větší než 0,3mm. Hodnoty čerpány ze zdroje [9]. Splněním těchto podmínek bylo možné nechat plošný spoj vyrobit jako prototyp u společnosti PragoBoard s. r. o.. Spoj má zelenou nepájivou masku, bílý potisk z obou stran a jako jádro je použit materiál FR4.

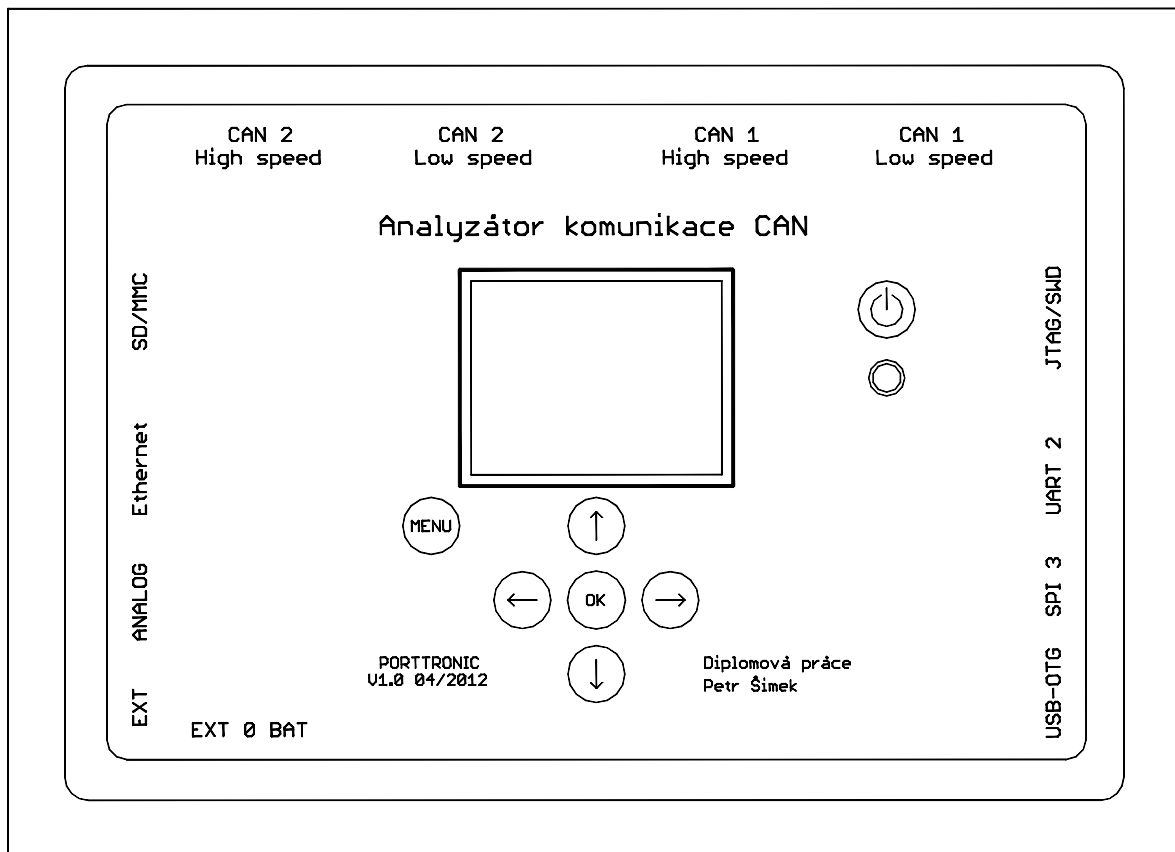
Deska displej je nosičem zobrazovacího panelu LCD a obsahuje napájecí a přizpůsobovací obvody. Výrobní parametry a technologie je obdobná jako u hlavní desky, ale má popis pouze na spodní straně.

Deska panelu je nosičem LCD a tlačítek pro obsluhu programu. S ohledem na výrobní náklady je pouze jednostranná. Vyrobená byla pomocí modré nažehlovací fólie na jádru FR4.

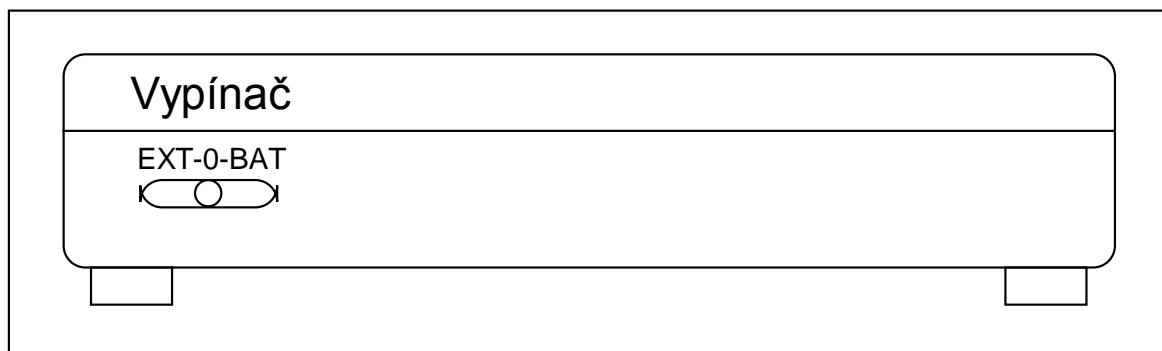
3.5 Návod k obsluze

3.5.1 Nákrezy zařízení

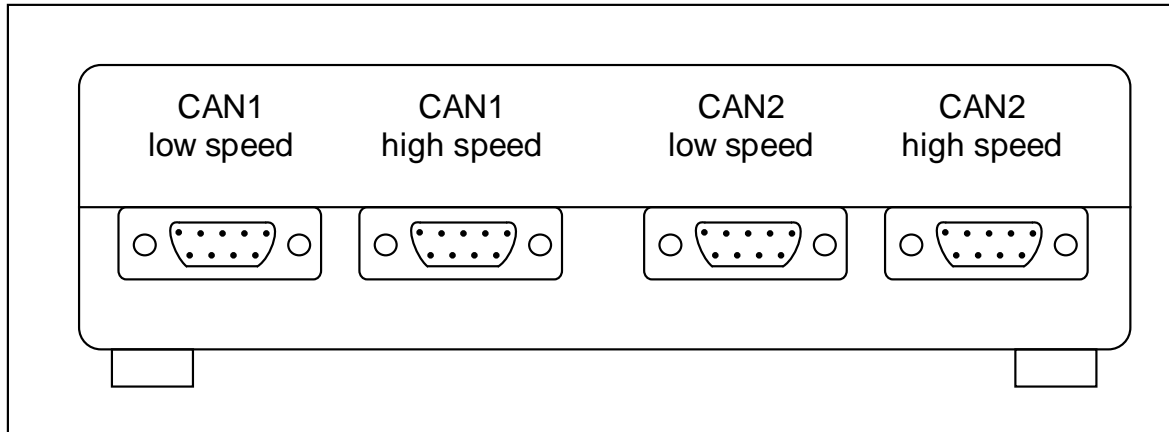
Zde jsou obrázky jednotlivých pohledů na zařízení.



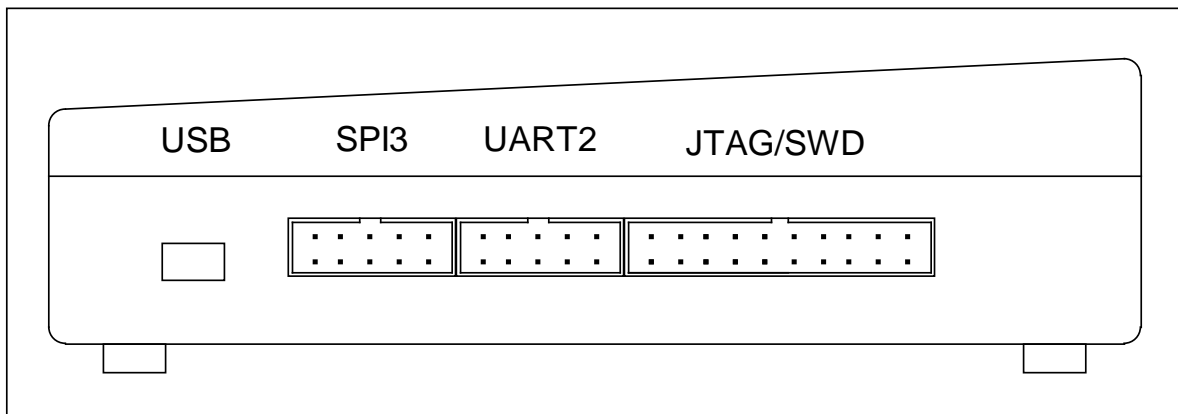
Obr. 3.5.1-1-Pohled shora



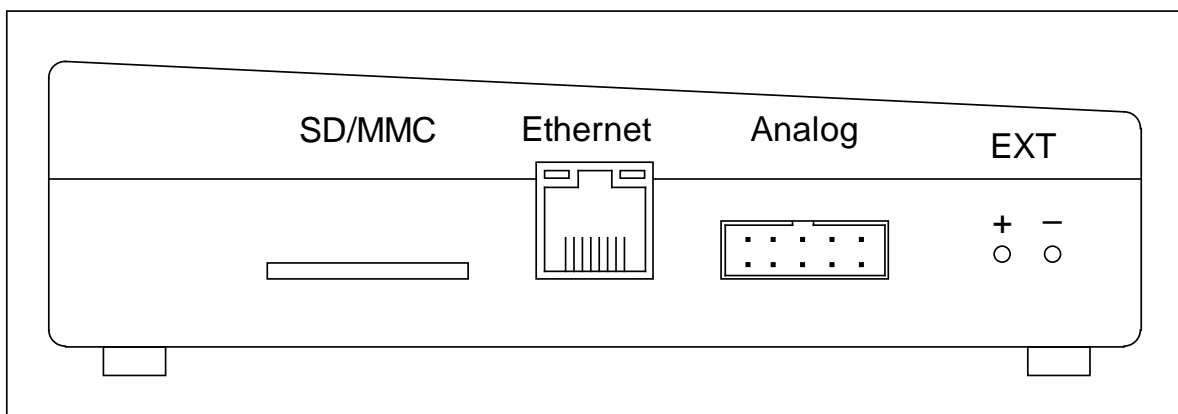
Obr. 3.5.1-2-Pohled zepředu



Obr. 3.5.1-3-Pohled zezadu



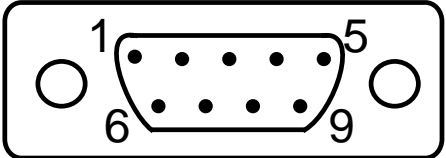
Obr. 3.5.1-4-Pohled z pravého boku



Obr. 3.5.1-5-Pohled z levého boku

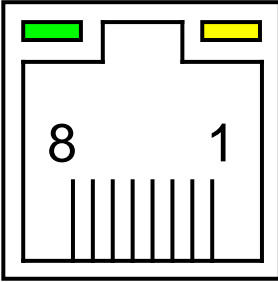
V následujících sekcích jsou popsána zapojení jednotlivých konektorů, které jsou použity pro zjednodušení obsluhy.

3.5.2 CAN

	Pin	High speed	Low speed
	1	-	-
	2	CAN_L	CAN_L
	3	GND	GND
	4	-	-
	5	-	-
	6	GND	GND
	7	CAN_H	CAN_H
	8	-	-
	9	-	V _{BAT+}

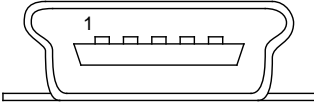
Tab. 3.5.2-1-Popis konektorů CAN, čerpáno ze zdroje [10]

3.5.3 Ethernet

	Pin	Funkce
	1	T _{XD+}
	2	T _{XD-}
	3	R _{XD+}
	4	-
	5	-
	6	R _{XD-}
	7	-
	8	-
	led	Funkce
Zelená	Detekce linky svítí při spojení s proti-stanicí	
Žlutá	Indikace přenosu bliká max. 10 Hz	

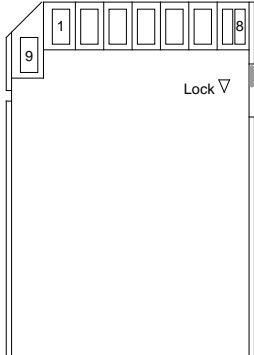
Tab. 3.5.3-1-Popis konektoru Ethernet, čerpáno ze zdroje [11]

3.5.4 USB

	Pin	Funkce
	1	V_{BUS}
	2	D-
	3	D+
	4	ID
5	GND	

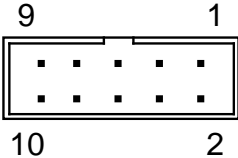
Tab. 3.5.4-1-Popis konektoru mini USB, čerpáno ze zdroje [7]

3.5.5 SD/MMC

	Pin	SD Mode	SPI Mode
	1	CD/DAT3	CS
	2	CMD	MOSI
	3	GND	GND
	4	V_{CC}	V_{CC}
	5	CLK	SCLK
	6	GND	GND
	7	DAT0	MISO
	8	DAT1	-
9	DAT2	-	

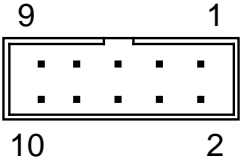
Tab. 3.5.5-1-Popis konektoru SD/MMC, čerpáno ze zdroje [8]

3.5.6 USART

	Pin	Funkce	Pin	Funkce
	1	+3,3 V	2	GND
	3	CK	4	GND
	5	R_{XD}	6	GND
	7	T_{XD}	8	GND
9	RTS	10	CTS	


Tab. 3.5.6-1-Popis konektoru USART 2

3.5.7 SPI

	Pin	Funkce	Pin	Funkce
	1	+3,3 V	2	GND
	3	EN	4	GND
	5	MOSI	6	GND
	7	MISO	8	NSS/TDI
	9	SCLK	10	MCK

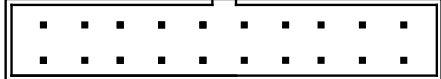
Tab. 3.5.7-1-Popis konektoru SPI 3

3.5.8 I2C

	Pin	Funkce
	1	+3,3 V
	2	GND
	3	SMBA
	4	SCL
5	SDA	

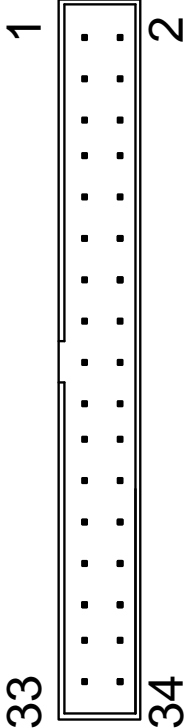
Tab. 3.5.8-1-Popis konektoru I2C 2 (interní)

3.5.9 JTAG/SWD

	Pin	Funkce		Funkce
	1	+3,3 V	2	+3,3 V
	3	NTRST	4	GND
	5	TDI	6	GND
	7	TMS/SWDIO	8	GND
	9	TCK/SWCLK	10	GND
	11	RTCK	12	GND
	13	TDO	14	GND
	15	NSRST	16	GND
	17	DBGRRQ	18	GND
	19	DBGACK	20	GND

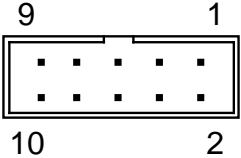
Tab. 3.5.9-1-Popis konektoru JTAG/SWD

3.5.10 Rozšiřující port EXP

	Pin	Funkce		Funkce
	1	LCD WR	2	LCD RD
	3	LCD DC	4	LCD CS
	5	LCD RST	6	LCD DIR
	7	LCD OED	8	LCD OEC
	9	LCD D0	10	LCD D1
	11	LCD D2	12	LCD D3
	13	LCD D4	14	LCD D5
	15	LCD D6	16	LCD D7
	17	LCD TE	18	LCD LED
	19	MENU	20	LEFT
	21	DOWN	22	OK
	23	UP	24	RIGHT
	25	STANDBY	26	LED RED
	27	LED GREEN	28	SPI2/3 MOSI
	29	SPI2/3 MISO	30	SPI2/3 SCLK
31	+3,3 V	32	+5 V	
33	GND	34	GND	

Tab. 3.5.10-1-Popis konektoru externích zařízení



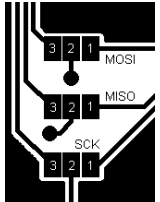
3.5.11 Analogový port ANALOG

	Pin	Funkce	Pin	Funkce
	1	+3,3 V	2	GND
	3	ADP(CH2)	4	GND
	5	ADL(CH1)	6	GND
	7	DAP(3CH8)	8	GND
	9	DAL(2CH6)	10	GND

Tab. 3.5.11-1-Popis konektoru analogových vstupů a výstupů

3.5.12 Propojky

Níže uvedené propojky se nachází v sekci 8 Přílohy Hlavní deska osazovací plán vrchní strana. Výjimkou jsou propojky MOSI, MISO, SCK, ty se nalézají v sekci 8 Přílohy Hlavní deska osazovací plán spodní strana.

Název	Poloha		Funkce
BOOT0 BOOT1		BOOT 0 : 1-2 BOOT 1 : 1-2	Start programu z interní flash paměti.
		BOOT 0 : 2-3 BOOT 1 : 2-3	Start programu z externí SRAM.
	BOOT 0 : 2-3 BOOT 1 : 1-2	Start programu z bootloadeu (UART,CAN...).	
VBAT		1-2	Napájení zálohované paměti z hlavního zdroje.
		2-3	Napájení zálohované paměti z baterie.
VBUS	Rozpojeno		Zakázání autodetekce režimu USB host.
	Spojeno		Povolení autodetekce režimu USB host.
JP2,7	Rozpojeno		Režim řízení strmosti hrany signálu CAN.
	1-2		Režim high speed.
	2-3		Režim standby budiče CAN.
JP3,4,6,8	Rozpojeno		Terminace 120 Ω odpojena.
	Spojeno		Terminace 120 Ω zapojena.
OTGEN	Rozpojeno		Napájení externího USB zařízení zakázáno.
	Spojeno		Napájení externího USB zařízení povoleno.
MOSI		1-2	SPI 2 MOSI přivedeno na konektor EXP.
MISO		2-3	SPI 3 MOSI přivedeno na konektor EXP.
		1-2	SPI 2 MISO přivedeno na konektor EXP.
SCK		2-3	SPI 3 MISO přivedeno na konektor EXP.
		1-2	SPI 2 SCK přivedeno na konektor EXP.
USB_CHG		2-3	SPI 3 SCK přivedeno na konektor EXP.
	1-2	Samostatné řízení nabíjení.	
		2-3	Společné řízení nabíjení.

Tab. 3.5.12-1-Seznam a nastavení propojek

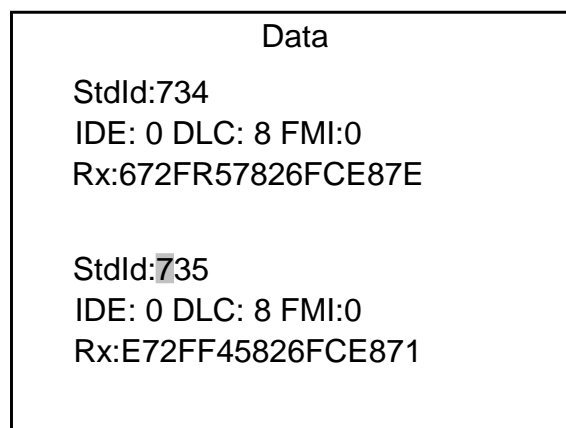
3.5.13 Hlavní přepínač

Jedná se o přepínač určený k úplnému vypnutí zařízení, který je znázorněn na Obr. 3.5.1-2, lze pomocí něj vybrat zdroj napájení. V poloze EXT je do zařízení dodávána energie z externího zdroje, například z palubní sítě automobilu v rozsahu 3 – 16 V. V poloze BAT je energie dodávána z vnitřního akumulátoru.

3.5.14 Obsluha programu

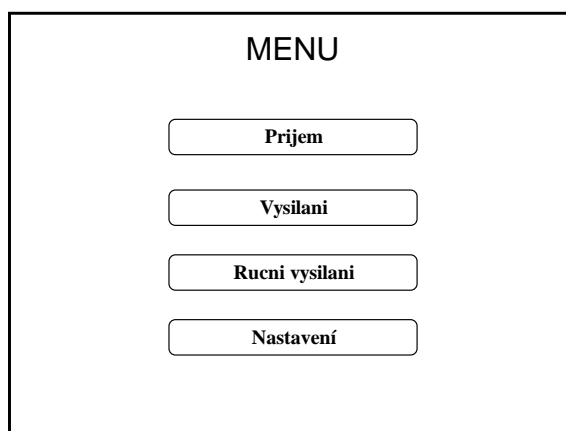
Tato kapitola obsahuje návod k obsluze programového vybavení navrženého zařízení a ukázky obrazovek s jejich popisem.

Při zapnutí zařízení se zobrazí uvítací obrazovka a program přejde na obrazovku ručního vysílání, viz Obr. 3.5.14-1. V tomto režimu není používán klasický kurzor šipky, ale k nastavení hodnot vysílané zprávy slouží ovládací šipky. Pravá a levá šipka vybírají nastavovaný znak a šipka nahoru zvyšuje hodnotu a šipka dolů snižuje.



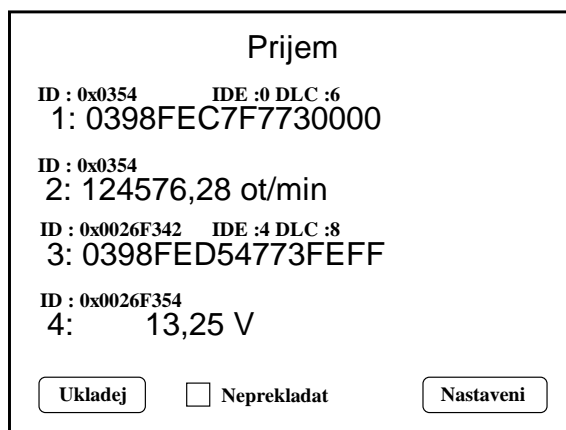
Obr. 3.5.14-1-Obrazovka Rucni vysilani

Po stisku tlačítka **MENU** (viz Obr. 3.5.1-1) dojde k zobrazení obrazovky MENU znázorněné na Obr. 3.5.14-2, v níž je možno vybrat jinou z funkčních obrazovek nebo přejít do nastavení zařízení.



Obr. 3.5.14-2-Obrazovka MENU

Obrazovka Prijem na Obr. 3.5.14-3 slouží k zobrazování dat z předem nastavených čtyř příjmových boxů. Režim zobrazení a nastavení těchto parametrů je popsán v sekci 3.6.1.2 Nastavení přijímacích boxů. V režimu příjmu proměnných se zobrazuje pouze ID zprávy, v režimu příjmu dat i pomocné informace jako je délka zprávy a druh rámce. Při povolení **Neprekladat** se zobrazují celé datové zprávy. V tomto režimu obrazovky Prijem se jedná o programový filtr hodnot sloužící pro zobrazení a převod dat přenášených ve zprávě lze kombinovat s filtry řadiče.



Obr. 3.5.14-3-Obrazovka Prijem

Po stisknutí tlačítka **Nastavení** dojde k přechodu na obrazovku Příjem nastavení na Obr. 3.5.14-4. Na této obrazovce je možnost zjištění podrobnějšího nastavení parametrů jednotlivých přijímacích boxů a nastavení druhu souborů, do něhož budou ukládána přijatá data. Výchozí nastavení souboru je binární s příponou **.DAT**, který umožňuje i pozdější reprodukci provozu na sběrnici, nebo textový s příponou **.CSV**. Konverze těchto souborů se provádí pomocí níže popsaného programu s názvem **KonvertorV1.0.exe**. Pomocí šipek ve výběrovém boxu se zobrazují hodnoty nastavení jednotlivých přijímacích boxů, popis hodnot naleznete v Tab. 3.6.1-3.

Prijem nastaveni

Box_2

Min : -1500000
Max : -1500000
Offset : 150000
Bajt : 0
Typ : uint32_t
ID : 0x00000354

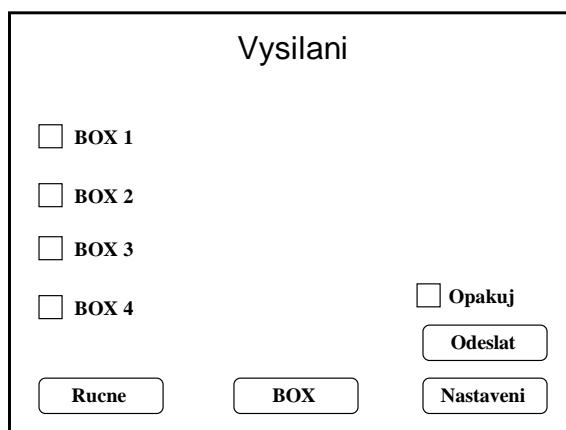
Ramec : 0
Kanal : 1
Jednotka : ot/min
Rezim : 0

.DAT
Zpet

Obr. 3.5.14-4-Obrazovka Prijem nastaveni

Obrazovka Vysilani na Obr. 3.5.14-5 zajišťuje nastavení základních funkcí a povolení vysílacích boxů. Přepínač **BOX** má stavy **BOX / DATA**, přepínač **Rucne** má stavy **Rucne / Automat**.

- **BOX**
 - **Rucne**
 - ✓ **Opakuj** – Při stisku tlačítka **Odeslat** cyklicky odesílá data z povolených boxů s definovanou periodou.
 - Opakuj** – Při stisku tlačítka **Odeslat** odešle jednou data z povolených boxů zpožděné o jednu periodu.
 - **Automat**
 - ✓ **Opakuj** – Při příchodu zprávy určené v daném boxu nastavením filtrovací zprávy dojde k odeslání datové zprávy zpožděné o jednu periodu po každém příjmu.
 - Opakuj** – Zpráva se odešle pouze po prvním příjmu
- **DATA** – Jméno datového souboru je definováno v souboru VYSILANI.INI.
 - **Rucne**
 - ✓ **Opakuj** – Při stisku tlačítka **Odeslat** se odešle jedna položka z datového souboru, položky se odesílají opakovaně.
 - Opakuj** – Data z datového souboru se odešlou pouze jednou.
 - **Automat**
 - ✓ **Opakuj** – Data z datového souboru se odesílají automaticky s uloženou periodou. Data v souboru se odesílají dokola.
 - Opakuj** – Data z datového souboru se odešlou pouze jednou.
- **Nastaveni** – Přepne na obrazovku Vysilani nastaveni.



Obr. 3.5.14-5-Obrazovka Vysilani

Obrazovka Vysilani nastaveni na Obr. 3.5.14-6 plní obdobnou informační funkci jako obrazovka Prijem nastaveni.

Vysilani nastaveni

Box_2

Data H : 0x76FED443
Data L : 0x7633D434
Data ID : 0x00000564
Filtr H : 0x76F63F31
Filtr L : 0x767756F3
Filtr ID : 0x00000564

Data DLC : 8
Data ide : 0
Filtr DLC : 8
Filtr ide : 0

Perioda : 155438
Kanal : 1
Rezim : 0

Zpet

Obr. 3.5.14-6-Obrazovka Vysilani nastaveni

Obrazovka Nastaveni na Obr. 3.5.14-7 je rozcestník pro nastavování jednotlivých funkcí zařízení.

- **CAN 1** – Otevře obrazovku pro nastavení řadiče CAN 1
- **CAN 2** – Otevře obrazovku pro nastavení řadiče CAN 2

Nastaveni

CAN 1

CAN 2

Obr. 3.5.14-7-Obrazovka Nastaveni

Obrazovka nastavení CAN 1 a 2 na Obr. 3.5.14-8. Je určena pro nastavení režimu a přenosové rychlosti řadičů CAN.

- **High** – Přepíná budič sběrnice dle požadované normy.
- **Expert** – Podrobnější nastavení režimu řadiče.
- **Uloz** – Přepíše nastavení v konfiguračním souboru CAN.INI
- **Zpet** – Provede nastavené změny a vrátí se na obrazovku Nastavení.
- + a – – Zvyšuje a snižuje hodnotu parametru v daném řádku.
- **Povoleni** – Aktivuje a deaktivuje příslušný řadič CAN.
- **Rezim** – Nastavuje parametr MODE v souboru CAN.INI

CAN x		Rezim	
<input type="checkbox"/> Povoleni		<input type="button" value="^"/> <input type="button" value="v"/> <input type="text" value="Silent"/>	
DIV	<input type="button" value="-"/> <input type="button" value="+"/>	12	<input type="button" value="High"/>
BS1	<input type="button" value="-"/> <input type="button" value="+"/>	4 tq	<input type="button" value="Expert"/>
BS2	<input type="button" value="-"/> <input type="button" value="+"/>	5 tq	<input type="button" value="Uloz"/>
SJW	<input type="button" value="-"/> <input type="button" value="+"/>	1 tq	<input type="button" value="Zpet"/>

Obr. 3.5.14-8-Obrazovka CANx

3.6 Program

V této sekci bude popsán program, jeho dílčí části, konfigurační a výstupní datové soubory. V části popisu funkcí budou popsány pouze důležité procedury, zbylé části programu jsou komentovány ve zdrojovém kódu – vstupní a výstupní proměnné funkcí jsou vždy popsány v hlavičkových souborech s komentářem účelu funkce, k nalezení na CD : \\Projekt.

3.6.1 Konfigurační soubory

Zařízení je koncipováno, tak že jeho konfigurace se provádí pomocí INI souborů. Zde je uloženo nastavení řadičů CAN, filtrů a zpráv, určených k příjmu a vysílání. Všechny potřebné soubory a podpůrné programy jsou uloženy na kartě SD/MMC.

3.6.1.1 Nastavení řadiče CAN

Soubor je umístěn ve složce \\CONFIG na kartě SD/MMC a jmenuje se CAN.INI – jsou zde hodnoty pro nastavení řadiče a filtrů podle standardních ovladačů periférií dodávaných od ST. Popis souboru je uveden v Tab. 3.6.1-1 a Tab. 3.6.1-2 Konkrétní informace jsou k nalezení v souboru s nápovědou dodávanou společně s ovladači. Naleznete na [20].

Pole	Popis
[CANx]	Popis sekce, v níž jsou konfigurační data CAN1 a CAN2.
ACTIV=1	0 = Kanál CAN zakázán, 1 = Kanál CAN povolen.
NORMA=1	1 = High speed, 0 = Low speed
TTCM=0	Zakázání time – triggered režimu.
ABOM=0	Zakázání Automatického bus-off režimu.
AWUM=0	Zakázání režimu automatického probuzení.
NART=0	Zakázání automatického přenosu.
RFLM=0	Zakázání uzamčení příjmové paměti FIFO.
TXFP=0	Nastavení priority vysílací paměti FIFO.
MODE= 0	Nastavení režimu řadiče CAN. 0 – Normal, 1 – LoopBack, 2 – Silent, 3 – Silent LoopBack.
SJW=0	Nastavení velikosti re-synchronizačního skoku hodnota 0 – 3.
BS1=3	Nastavení velikosti Základního segmentu 1 hodnota 0 – 15.
BS2=4	Nastavení velikosti Základního segmentu 2 hodnota 0 – 7.
PRESCALER=16	Nastavení děličky. Určuje velikost časového kvanta hodnota 0 – 1024.

Tab. 3.6.1-1-Popis konfiguračního souboru CAN – nastavení řadiče

$$V_p = \frac{f_{APB1} / \text{PRESCALER}}{BS1 + BS2 + 1} = \left[\frac{b}{s} \right]$$

Rovnice 3.6.1-1- Vzorec pro výpočet přenosové rychlosti

Při nastavování přenosové rychlosti se vychází ze základního vstupního kmitočtu $f_{APB1} = 30\text{MHz}$, ten se vydělí hodnotou děličky a celkovým počtem časových kvant viz Rovnice 3.6.1-1. Zde se automaticky připočítává jedno časové kvantum, jako úsek Sync popsany na Obr. 2.2-1.

Pole	Popis
[FLT _x]	Popis sekce, v níž jsou konfigurační data filtru FLT0-27.
ACTIV=1	0 = Filtr zakázán, 1 = Filtr povolen.
MODE=0	0 = Odfiltrování nastavené hodnoty 1 = Propouštění.
SCALE=0	0 = 16bit 1 = 32bit.
ID_HIGH=0x0000	Hodnota filtrovaného ID horních 16bit.
ID_LOW=0x0000	Hodnota filtrovaného ID dolních 16bit.
MASK_ID_HIGH =0xFFFF	Maska filtru horních 16bit.
MASK_ID_LOW=0x0000	Maska filtru dolních 16bit.
FIFO=0	Výběr, do jaké paměti FIFO bude filtrováno 0 nebo 1.

Tab. 3.6.1-2-Popis konfiguračního souboru CAN – nastavení filtrů

V šestnáctibitovém režimu filtr pracuje jako dva, což při použití 11 – bitových identifikátorů zvyšuje počet filtrů na dvojnásobek. Hodnotu 11 – bitového identifikátoru je třeba v 16bit registru posunout o 5 bitů doleva (vynásobit 32). Filtr pro identifikátor 0x723 má hodnotu 0xE460. Pro 29 – bitový identifikátor se použije režim filtru 32 – bitový a posouvá se o 3bity, hodnota se rozdělí a vloží do 2 16 – bitových registrů. Pro masku je postup obdobný. Filtry jsou rozděleny na 2 sekce. Filtr 0 – 13 pro CAN 1 a 14 – 27 pro CAN 2.

3.6.1.2 Nastavení přijímacích boxů

Soubor je umístěn ve složce \\CONFIG na paměťové kartě SD/MMC a jmenuje se PRIJEM.INI. Slouží pro nastavení programového filtrování a převod proměnných ve zprávě. Soubor obsahuje data pro 4 boxy, v nichž jsou zprávy zpracovány a zobrazovány na zařízení. Pro ukládání pouze těchto zpráv, je třeba nakonfigurovat filtr řadiče daného kanálu CAN na stejné identifikátory. Při spuštění záznamu, jsou zaznamenány všechny zprávy, které projdou přes filtr daného řadiče CAN. Filtry boxů jsou pouze pro účel zobrazování proměnných nebo celých zpráv.

Pole	Popis
[BOXx]	Popis sekce přijímacího boxu.
MIN=-1000	Minimum hodnoty maximálně 10 pozic (celé číslo).
MAX=1000	Maximum hodnoty maximálně 10 pozic (celé číslo).
OFFSET=1000	Posunutí hodnoty v proměnné. 10 pozic (celé číslo).
BAJT=0	První bajt proměnné.
TYP=uint16_t	Udává druh proměnné.
ID=0x00000278	Identifikátor zprávy.
RAMEC=0	Druh rámce 0 = StdId, 4 = ExtId.
KANAL=1	Kanál, ze kterého se přijímá 1 = CAN1, 2 = CAN2.
TEXT=mA	Jednotka proměnné maximálně 6 znaků (text).
REZIM=0	Určuje druh zobrazení 0 = Překlad proměnné, 4 = data.

Tab. 3.6.1-3-Popis konfiguračního souboru PRIJEM.INI

Výše uvedená konfigurace definuje bezznaménkovou proměnnou s rozlišením 16 – bitů umístěnou ve zprávě v nultém (dolních 8 bitů) a prvním (horních 8 bitů) bajtu. Hodnota má rozsah 0 – 2000, proměnná 0 – 65536. Z tohoto plyne, že váha nejnižšího bitu je 0,030517578125. Z důvodu bezznaménkové proměnné je zde zaveden offset, který se od přijaté hodnoty odečte, a tím se docílí i záporných hodnot. Převodní vztah viz Rovnice 3.6.1-2. Takto interpretovaná hodnota se zpracovává dále jako reálné číslo typu float.

$$Hodnota = \left(promenna \times \frac{MAX - MIN}{2^{8 \times sizeof(TYP)}} \right) - OFFSET$$

Rovnice 3.6.1-2-Vztah pro převod proměnné

Takto je definován TYP: uint8_t, uint16_t, uint24_t, uint32_t, jako další možností je přímé zobrazení přenášené hodnoty float jako TYP: real32_t, nebo zobrazení jednotlivých

bitů daného bajtu proměnné jako TYP: binar. Detaily převodní funkce jsou popsány v sekci 3.6.5 funkce `Tisk_hodnoty(BOXx)` ; .

3.6.1.3 Nastavení vysílacích boxů

Soubor je umístěn ve složce `\\CONFIG` na paměťové kartě SD/MMC a jmenuje se `VYSILANI.INI`. Slouží pro nastavení obsahu datové a filtrovací zprávy.

Pole	Popis
[BOXx]	Popis sekce vysílacího boxu.
DATA_H=0xF567CD5E	Datové bajty 4 – 7.
DATA_L=0x234ED7F8	Datové bajty 0 – 3.
DATA_ID=0x00000348	Identifikátor datové zprávy.
DATA_DLC=8	Délka datového pole.
DATA_IDE=0	Pokud má parametr hodnotu : 0 = StdId, 4 = ExtId..
DATA_RTR=0	Hodnota udává, zda se jedná o 0 = Data, 2 = Dotaz.
FILTR_H=0x00000000	Datové bajty filtru 4 – 7.
FILTR_L=0x00000087	Datové bajty filtru 0 – 3.
FILTR_ID=0x00000349	Identifikátor zprávy filtru.
FILTR_DLC=1	Délka datového pole filtru.
FILTR_IDE=0	Pokud má parametr hodnotu : 0 = StdId, 4 = ExtId.
FILTR_RTR=0	Hodnota udává, zda se jedná o 0 = Data, 2 = Dotaz.
KANAL=1	Kanál, s kterým se pracuje 1 = CAN1, 2 = CAN2.
PERIODA=123000	Perioda odesílání zpráv nebo zpoždění v μ s.
[SOUBOR]	Popis sekce pro soubor, z něhož se čtou reprodukováná data.
NAZEV=DATA.DAT	Řetězec s názvem souboru datových zpráv.

Tab. 3.6.1-4-Popis konfiguračního souboru VYSILANI.INI

Položky DATA slouží pro nastavení vysílané zprávy. Položky FILTR slouží pro nastavení filtrační zprávy. Při přijetí stejné zprávy, která je nastavena v položkách FILTR, jednotka odešle zprávu definovanou v položkách DATA zpožděnou o počet mikrosekund uvedených v položce PERIODA. Podrobnější popis funkcí viz sekce 3.5.14, strana 39.

3.6.2 Datové soubory

Textový soubor je ukládán v kořenové složce na paměťové kartě SD/MMC a jmenuje se 00000000.CSV. Při každém spuštění záznamu, je vytvořen nový, jeho název je o jedničku vyšší číslo. Na každém řádku je uložena jedna zpráva v textové podobě.

000001,0000000001,2A3F,87,FF,FF,FF,FF,FF,FF,FF,FF;	
Sekce	Popis
1	Číslo zprávy 1 – 999999. (dekadicky)
2	Čas v μ s od počátku záznamu v době stisknutí tlačítka Ukladej .
3	Identifikátor zprávy. (Hexadecimálně)
4	Datové pole. (Hexadecimálně)
5–12	Přijatá data podle délky zprávy.

Tab. 3.6.2-1.Popis zprávy v datovém souboru

87 -> 10000111	
Bit	Popis
0	Druh rámce 1 = 2.0B, 0 = 2.0A
1	Dotaz na data 1 = dotaz, 0 = data
2	Kanál 1 = CAN2, 0 = CAN1
3	Rezerva
4	Rezerva
5 – 7	Délka datové části 0-7 bajtů. V případě dotazu neplatné.

Tab. 3.6.2-2-Popis datového pole ve zprávě

Binární soubor je ukládán v kořenové složce a jmenuje se 00000000.DAT. Vytváření nových souborů je obdobné jako u textového formátu. Zaznamenávají se zde přímo data nepřevedená na text. Tvar dat je udán strukturou **Zasobník**. V Tab. 3.6.2-3, v níž je použita struktura **CanRxMsg** uvedená v Tab. 3.6.2-4. Pro jejich zobrazení a převod do formátu .CSV slouží program KonvertorV1.0.exe. Další popis programu je v kapitole 3.6.3 Editační program.

Typ	Název	Popis
uint32_t	StdId	11 bitový identifikátor 0x0000 - 0x07FF
uint32_t	ExtId	29 bitový identifikátor 0x00000000 - 0x1FFFFFFF
uint8_t	IDE	Pokud má parametr hodnotu platí : 0 = StdId, 4 = ExtId.
uint8_t	RTR	Pole udává, zda se jedná o 0 = Data, 2 = Dotaz.
uint8_t	DLC	Udává počet platných bajtů datového pole.
uint8_t	Data[0-7]	Přijatá data.
uint8_t	FMI	Stav pinu budiče low-speed bit 1 – INH, bit 2 – ERR .

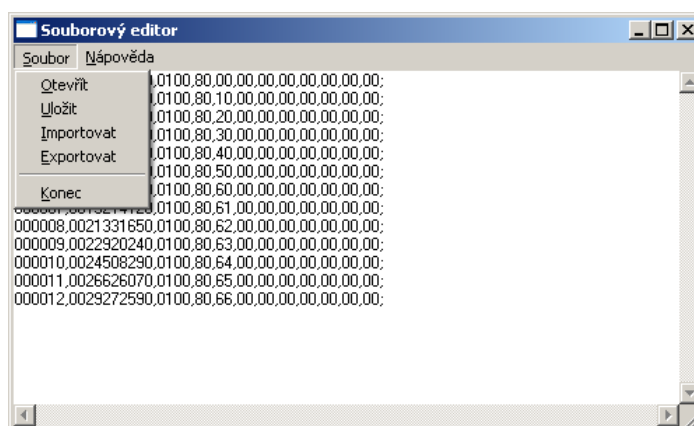
Tab. 3.6.2-3-Struktura datové zprávy

Typ	Název	Popis
uint32_t	PORADI	Určuje pořadí zpráv, jak byly přijaty.
uint32_t	CAS	Čas v μ s od počátku záznamu v době stisknutí tlačítka Ukladej .
uint8_t	KANAL	Kanál CAN, na který zpráva přišla.
CanRxMsg	ZPRAVA	Přijatá zpráva.

Tab. 3.6.2-4-Struktura zásobníku

3.6.3 Editační program

Z důvodu jednoduššího reprodukování datových zpráv se data primárně ukládají jako binární, a proto je třeba konverzní program. Jeho účelem je vzájemný převod datových souborů, možnost prohlížení a editace. Dále program může sloužit pro tvorbu a editaci .INI konfiguračních souborů. Po stažení datového souboru z paměťové karty SD/MMC je soubor otevřen nebo importován a může být dále zpracován nebo exportován či uložen. Program je napsán jako aplikace pro MS Windows a je uložený ve složce \\Program na přiloženém CD. Informace byly čerpány z ukázek ze zdroje [12].



Obr. 3.6.3-1-Ukázka konverzního programu

Obsluha aplikace je pomocí záložky **Soubor** v liště menu okna programu.

- **Otevřít** – slouží pro načtení .CSV souboru a jeho kontrolu.
- **Uložit** – Uloží otevřená data do souboru .CSV.
- **Importovat** – Otevře a převede binární soubor .DAT na textový .CSV
- **Exportovat** – převede otevřená data do binárního souboru .DAT.

3.6.4 Struktura projektu

Vzhledem k rozsahu projektu je třeba dělit dílčí části do knihoven. Z tohoto důvodu zde popíši základní strukturu programu a obsah jednotlivých částí. Projekt je uložen ve složce \\Projekt\Can_logger_V1.0 a v jeho podsložkách na přiloženém CD. Každá knihovna má podsložku \src obsahující soubory .c se zdrojovým kódem a podsložku \inc obsahující hlavičkové soubory. Jako základ pro interakci s uživatelem je použita grafická komponenta společnosti ST s názvem Embedded GUI, čerpáno ze zdroje [21], která zajišťuje vykreslování jednotlivých objektů a reakci na jejich změnu stavu. Pomocí aplikace Resource Editor For Embedded GUI Library čerpáno ze zdroje [21] jsou sestaveny jednotlivé obrazovky interpretované uživa-

teli a je pro ně vygenerován kód pro vykreslování objektů a funkce prováděné jako reakce na změnu stavu jednotlivých vykreslených objektů.

3.6.4.1 Knihovny

Složka : \\Projekt\Can_logger_V1.0\Libraries\

Podsložka : analyzer

analyzer.c a .h – Obsahuje funkce pro práci s daty, jejich zpracování, vykreslování a dále také nastavení řadiče CAN a obsluhu displeje.

Podsložka : CMSIS

Obsahuje soubory zavaděče a konfiguraci jádra ARM. Podrobnější informace najdete na přiloženém CD ve složce : \\Projekt\Libraries\CMSIS.

Podsložka : Embedded_GUI_HAL

LcdHal.c a .h – Obsahuje funkce pro vykreslování objektů na LCD.

JoyHal.c a .h – Funkce pro čtení stavu tlačítek, pro obsluhu kurzoru.

Podsložka : Embedded_GUI_Library

uiframework.c a .h – Funkce pro vykreslování obrazovek.

uiappuser.c a .h – Reakce na obsluhu objektů.

pictures.c a ,h – Definice konvertovaných obrázků.

graphicObject.c a .h – Funkce vyšších objektů.

gl_fonts.c a .h – Definice textových fontů.

cursor.c a .h – Vykreslování kurzoru.

Podrobnější informace najdete na přiloženém CD ve složce:

\\Projekt\Libraries\Embedded_GUI_Library

Podsložka : fat_sd

sd_spi_stm32.c – Zprostředkovává komunikaci FAT s SD/MMC kartou.

Podrobnější informace najdete na přiloženém CD ve složce:

\\Projekt\Libraries\ fat_sd

Podsložka : lcd_driver

lcd_driver.c a .h – Obsahuje základní moduly pro obsluhu a inicializaci paralelního LCD.

lcd_cmd.h – Obsahuje definice příkazu pro paralelní LCD.

lcd_SPI_driver.c – Obsahuje základní moduly pro obsluhu a inicializaci sériového LCD.

Podsložka : minIni

minIni.c a .h – Pracuje s konfiguračními .INI soubory.

Podsložka : STM32F2xx_StdPeriph_Driver

Obsahuje knihovny pro obsluhu periférií daného mikropočítače.

Podrobnější informace najdete na příloženém CD ve složce:

\\Projekt\Libraries\STM32F2xx_StdPeriph_Driver

3.6.4.2 Inicializace

Složka : \\Projekt\Can_logger_V1.0\Utilities

Podsložka : CAN_LOGGER

Hardware_profile.h – Obsahuje deklarace všech použitých rozhraní a jejich konkrétní mapování na daný mikropočítač.

can_logger.c a .h – Obsahují základní inicializační funkce a makra.

3.6.4.3 Základní soubory

Složka : \\Projekt\Can_logger_V1.0

main.c – Hlavní základní soubor.

stm32f2xx_it.c a .h – Obsluha přerušení.

system_stm32f2xx.c – Základní konfigurace sběrnic a rozvodu hodin.

stm32f2xx_conf.h – Konfigurace základního projektu a výběr ovladačů periférií.

3.6.5 Popis funkcí

Umístění souborů je popisováno z hlavní projektové složky nazvané \\Can_loggerV1.0 umístěné na příloženém CD. Hlavičkové soubory jsou vždy ve stejném podadresáři ve složce \\inc.

3.6.5.1 Hlavní část

int main (**void**)

- Funkce hlavní programové smyčky provádí počáteční inicializaci a poté cyklickou obsluhu grafického rozhraní, vysílání a ukládání zpráv z řadiče CAN.

- Umístění \\src\main.c

void init_prog (**void**)

- Funkce inicializace zařízení konfiguruje používané periferie mikropočítače.

- Umístění \\src\main.c

void idle_loop (void)

- Funkce pro obsluhu grafického rozhraní a práce s kartou SD/MMC načítání a ukládání datových zpráv.
- Umístění \\src\main.c

3.6.5.2 Důležité funkce

void Load_CAN_INI (void)

- Funkce pro načtení inicializačního souboru pro řadič CAN. Zpracovává soubor CAN.INI. Voláno při inicializaci zařízení.
- Umístění \\Libraries\analyzer\src\analyzer.c

void Load_CAN_X_Config (char CAN_X)

- Pomocná funkce pro načtení parametru konfigurace ze souboru CAN.INI. Využívá se při ručním nastavení přenosové rychlosti řadiče CAN. Volá se při stisku CAN 1 a CAN 2 v obrazovce Nastavení.
- Vstupní parametry 1 nebo 2 podle nastavovaného kanálu.
- Umístění \\Libraries\analyzer\src\analyzer.c

void Load_BOX_data (void)

- Naplnění proměnných BOX_1–4. Voláno při vstupu na obrazovku Prijem a Nastavení příjem.
- Umístění \\Libraries\analyzer\src\analyzer.c

void Screen_select (void)

- Provádí vykreslování obrazovek obsluhu kurzoru a periodickou obnovu dat pro vybrané obrazovky. Voláno v idle_loop ();
- Umístění \\Libraries\analyzer\src\analyzer.c

void Load_prijem_nastaveni (Prijem_BOX *BOX)

- Provádí výpis hodnot nastavených v souboru PRIJEM.INI.
- Vstupní parametr je ukazatel na proměnnou BOX_1–4, v níž jsou umístěny parametry načtené předchozí funkcí Load_box_data();
- Umístění \\Libraries\analyzer\src\analyzer.c

void Tisk_hodnoty (Prijem_BOX *BOX)

- Provádí převod přijatých dat podle nastavení jednotlivých boxů. Voláno v Screen_select();
- Vstupní parametr je ukazatel na proměnnou BOX_1–4.
- Umístění \\Libraries\analyzer\src\analyzer.c

void Uloz_data_prijem (**void**)

- Plnění kruhového vyrovnávacího zásobníku pro jeho pozdější zpracování funkcí `Nacti_data_prijem()`; Voláno v přerušení CAN kanálu 1 a 2.
- Umístění `\\Libraries\analyzer\src\analyzer.c`

void Nacti_data_prijem (**void**)

- Provádí vyčítání, zpracování a ukládání přijatých dat na paměťové médium. Voláno v `idle_loop()`;
- Umístění `\\Libraries\analyzer\src\analyzer.c`

void Uloz_data_vysilani (**void**)

- Plnění kruhového vyrovnávacího zásobníku pro jeho pozdější zpracování funkcí `Nacti_data_vysilani()`; Voláno cyklicky v případě uvolnění místa v zásobníku. Provádí předpřípravu dat pro jejich přesné odeslání.
- Umístění `\\Libraries\analyzer\src\analyzer.c`

void Nacti_data_vysilani (**void**)

- Provádí vyčítání a zpracování dat a nastavování časové spoušti. Voláno v přerušení systémového časovače, CAN 1 a CAN 2.
- Umístění `\\Libraries\analyzer\src\analyzer.c`

uint8_t Porovnej_zpravy (CanRxMsg Data, CanRxMsg Filtr)

- Provádí porovnání 2 přijatých zpráv.
- Vstupní parametry struktury zpráv `CanRxMsg`.
- Výstupní parametr Zprávy jsou stejné = 1, rozdílné = 0.
- Umístění `\\Libraries\analyzer\src\analyzer.c`

uint8_t CAN_Transmit_RxMsg(CAN_TypeDef* CANx, CanRxMsg* RxMes)

- Odesílá zprávy ze struktury `CanRxMsg` pro možnost využití přijímacího zásobníku i pro vysílání zpráv voláno funkcí `Nacti_data_vysilani()`;
- Vstupní parametry Kanál CAN a struktura zprávy.
- Výstupní parametr číslo odesílacího boxu řadiče CAN.
- Umístění `\\Libraries\analyzer\src\analyzer.c`

void Uprava_zpravy (**void**)

- Nastavuje vysílací zprávu na obrazovce Rucni nastaveni voláno ve funkci `Screen_select()`;
- Umístění `\\Libraries\analyzer\src\analyzer.c`

3.6.5.3 Přerušení

void SysTick_Handler(**void**)

- Obsluhuje přerušení od systémového časovače, který pracuje synchronně s taktem jádra a přerušuje každých 10 μ s. Obsluhuje čekací smyčky pro Souborový systém, měření času při ukládání zpráv a přesné odesílání zpráv na CAN.
- Umístění `\\src\stm32f2xx_it.c`

void CAN1_RX0_IRQHandler(**void**)

- Obsluha přerušení od řadiče CAN 1. Po příchodu zprávy provede její zpracování nebo uložení. Zprávy jsou ukládány do paměti FIFO 0 a mohou využívat filtry řadiče 0–13.
- Umístění `\\src\stm32f2xx_it.c`

void CAN2_RX0_IRQHandler(**void**)

- Obsluha přerušení od řadiče CAN 2. Po příchodu zprávy provede její zpracování nebo uložení. Zprávy jsou ukládány do paměti FIFO 1 a mohou využívat filtry řadiče 14 – 27.
- Umístění `\\src\stm32f2xx_it.c`

Celý program probíhá v hlavní smyčce `main()`, kde se volají funkce obsluhy grafického prostředí jako je funkce `Screen_select()`, která zajišťuje zobrazování grafických objektů na displej. Funkce zajišťující zápis a čtení dat z karty SD/MMC jako je `Nacti_data_prijem()` a `Uloz_data_vysilani()`. Další důležité a méně časově náročné funkce jako je příjem nebo vysílání rámců jsou volány v přerušeních řadiče CAN 1 a 2 a systémového čítače.

Zbytek funkcí je popsán a komentován přímo ve zdrojovém kódu umístěném na příloženém CD v adresáři `\\Projekt`. Členění a obsah jednotlivých souborů je popsán v sekci 3.6.4 Struktura projektu.

3.7 Vývojové prostředí

Mikropočítače s jádrem ARM jsou univerzální platformou jakou je architektura x86 nebo x64. Z tohoto důvodu je na trhu velké množství vývojových prostředí jak volně šiřitelných pod licencí GPL až po plně komerční. Výhodou použití jádra ARM je jeho standardní základní sada instrukcí, která umožňuje přenositelnost kódu od jednoho výrobce mikropočítačů k jinému, bez nutnosti zásadní změny kódu. Nyní zde popíši několik nejznámějších prostředí, která se vyskytují na trhu, a projekt s nimi lze přeložit. Pro ladění a programování jsem použil adapter ST-Link s rozhraním SWD. Jako vývojové prostředí, byla použita školní verze KEIL omezená na kompilaci 256KB kódu.

3.7.1 Nastavení parametrů kompilátoru

Projekt je možno zkompileovat i s nejvyšší úrovní optimalizace (-O3). Do projektu je třeba vložit cesty ke všem hlavičkovým souborům – ty se vždy nalézají ve složce \inc.

Seznam cest k souborům:

```
..\inc
..\Libraries\CMSIS\CMS3\DeviceSupport\ST\STM32F2xx
..\Libraries\CMSIS\CMS3\CoreSupport
..\Libraries\STM32F2xx_StdPeriph_Driver\inc
..\Utilities\CAN_LOGGER\inc
..\Libraries\fat_sd\inc
..\Libraries\lcd_driver\inc
..\Libraries\analyzer\inc
..\Libraries\minIni\inc
..\Libraries\Embedded_GUI_HAL\inc
..\Libraries\Embedded_GUI_Library\inc
```

Jako použitý mikropočítač se musí zvolit obvod STM32F217ZG. Podle tohoto si kompilátor nastaví velikosti paměti počáteční adresy a umístění systémových zásobníků.

3.7.2 Keil

Jedná se o plně komerční prostředí se spoustou komponent pro vývoj a ladění programu. Společnost také vyvíjí svůj vlastní překladač, ale umožňuje použití i překladače gcc. Je zde podpora širokého množství programovacích adapterů. Základní zkušební verze Lite je omezena na 32Kb kódu. Podrobnější informace na [13].

3.7.3 Altium

Jedná se pouze o modul pro vývojové prostředí, které poté umožňuje kompletní vývoj elektronického zařízení od návrhu zapojení až po vytvoření výrobní dokumentace a podkladu pro výrobu zařízení včetně možnosti softwarové simulace zařízení jako celku. Toto prostředí je víceúčelové a je zaměřeno na komplexní vývoj zařízení. Podrobnější informace na [14].

3.7.4 Raid

Jedná se o prostředí vyvíjené společností Raisonance. Jeho starší verze je zdarma dostupná ke stažení. Prostředí používá volný kompilátor gcc a nahrávání programu probíhá pomocí adapteru R-Link, který podporuje jak rozhraní SWD tak JTAG. Základní verze tohoto adapteru bez klíče umožňuje ladění kódu pouze do velikosti 64Kb, zato však kompilaci kódu neomezené velikosti. Podrobnější informace na [15].

3.7.5 Atollic

Jedná se o prostředí založené na Eclipse používající překladač gcc. Volná verze Lite není omezena velikostí kódu, ale neobsahuje spoustu podpůrných komponent pro vývoj jako plná placená verze. Podrobnější informace na [16].

3.7.6 Eclipse + GCC + OpenOCD

Jedná se o využití programu Eclipse jako standardního prostředí IDE. Je zde třeba vše nastavit, jaký se použije kompilátor a programovací rozhraní. Pracuje se zde s gcc kompilátorem a programovacím a ladícím serverem Open OCD. Výhodou je, že vše je plně zdarma šířeno pod licencí GPL. Nevýhodou je nutná podrobná znalost funkce všech komponent a jejich nastavení. Tuto nevýhodu se snaží řešit projekt CooCox, který provede konfiguraci za vás. Podrobnější informace na [17], [18].

4 Pokračování vývoje

Jako další rozvoj tohoto projektu bude zdokonalování obslužného programu a implementace nových funkcí, pro zpracování a generování dat. Další potřebnou implementací je sběrnice USB, vytvoření nadřazeného programu pro obsluhu a integrace mass-storage protokolu. V další generaci obvodového zapojení bude třeba doplnit systémové programové řízení některých funkcí jako je například terminace sběrnice CAN a vylepšení řízení napájení. Zde je třeba doplnit vypínání měničů pro napájení jednotlivých budičů sběrnice CAN, upravit a překreslit řízení napájení hlavního zdroje. V další generaci, by byl využit mikropočítač s vyšším počtem vývodů z důvodu možnosti využití sběrnice FSMC pro připojení displeje a SDIO pro zrychlení ukládání dat. Jako další často používanou komunikační linkou je LIN a v další generaci by proběhla jeho implementace přímo do zařízení, ne pouze možnost jako přídatný modul. Další zvažovanou možností je analýza Ethernetu nebo jeho případné využití pro spojení více zařízení a vytvořit tak most mezi více sběrnici CAN. Tento režim může být vhodný pro analýzu sběrnic v průmyslové technologii a svedení několika analyzátorů do centrální jednotky, kde může probíhat komplexní analýza celé technologie.

5 Ekonomické zhodnocení

Zde je provedena finanční rozvaha celého zařízení, ceny byly čerpány z online katalogů společností GM electronic, TME Electronic Component a Mobil-Partner.cz v prvním pololetí roku 2012. V Tab. 5-1 je provedená cenová rozvaha pro hlavní desku. V Tab. 5-2 je cenová rozvaha panelu a v Tab. 5-3 displeje. Rozvaha ceny pro celkové zařízení je uvedena v Tab. 5-4. Jedná se o náklady pro vytvoření prototypu v množství jednoho kusu. V zapojení by se dalo nalézt ještě spoustu míst k podstatným úsporám výrobních nákladů zejména na výrobě plošných spojů a použitím levnějších LCD panelů. Vzhledem k účelu daného zařízení se nedá předpokládat jeho velkosériová výroba, proto množstevní slevy nebudou tak značné. V celkovém soupisu je 700 Kč jako náklady na dopravu, tato položka by se dala podstatně snížit provedením 1 kompletní objednávky, což není úplně možné, protože neexistuje společnost, která by dodala vše. Velká část nákladů na dopravu, vyvstala z toho, že některé díly bylo potřeba objednat přednostně před finálním návrhem z důvodu testování zapojení a před-přípravy programových komponent.

Hlavní deska			
Množství	Hodnota	Název	Cena
6	1N4148	1N4148SMD	0,6
2	74HCT157D	74HCT157D	3,5
2	AM1DTHT	AM1DTHT	122
15	100n	C-EUC0603	0,15
14	100nF	C-EUC0603	0,15
6	100n	C-EUC0603K	0,15
2	2u2	C-EUC0805K	2
2	10pf	C-EUC0805K	2
4	22pf	C-EUC0805K	2
1	100n	C-EUC0805K	2
2	220nF	C-EUC0805K	2
1	320n	C-EUC0805K	2
1	2u2	C-EUC1206K	2
1	8MHz	CON-SMD4XA	33
2	10u	CPOL-EUA/3216-18R	5
3	10uF	CPOL-EUA/3216-18R	5
1	47uF	CPOL-EUC/6032-28R	5
3	100uF	CPOL-EUC/6032-28R	7
1	10uF	CPOL-EUCT3528	5
1	CR2032H	CR2032H	25
1	25MHz	CRYTALSM49	16
1	3,2768KHz	CRYTALTC26H-SMD	33
2	SM6T5CA	DIODE_SCHOTTKY2-SMB	8
6	SM6T15CA	DIODE_SCHOTTKY2-SMB	8
1	MBRS2040LT3G	DIODE-SMB	4,5
1	ICTE15SMD	ICTE5SMD	10
2	ILD206TSMD	ILD206TSMD	12

1	IRLL2705	IRLL2705SMD	12
2	ISO7221SMD	ISO7221SMD	84
4		JP1E	
4		JP1Q	
3		JP2E	
1	LM1117SMD	LM1117SMD	14,7
1	LM3478SMD	LM3478SMD	66
1	1uH	L-US1210	6,4
2	12uH	L-USDE1207	8
4		M09HP	
1		MA05-1	
1	MAX1811ESASMD	MAX1811ESASMD	76
2	MCP2551	MCP2551SMD	19,4
3		ML10L	12,3
1		ML20L	12,6
1		ML34	16
1		MTA02-100	
2	10k	R-EU_M0805	10
1	47K	R-EU_M0805	10
1	470R	R-EU_M0805	10
1	0R	R-EU_R0603	10
5	1k5	R-EU_R0603	10
1	4k7	R-EU_R0603	10
8	10k	R-EU_R0603	10
2	50R	R-EU_R0603	10
1	100R	R-EU_R0603	10
2	240R	R-EU_R0603	10
1	0R	R-EU_R0805	10
4	1K	R-EU_R0805	10
1	1k	R-EU_R0805	10
2	2K4	R-EU_R0805	10
2	2k2	R-EU_R0805	10
14	4K7	R-EU_R0805	10
6	10K	R-EU_R0805	10
15	10k	R-EU_R0805	10
4	120R	R-EU_R0805	10
10	470R	R-EU_R0805	10
1	0R22	R-EU_SMD	0
1	RJ45(J3011)	RJ45SMD	149
1	SDSD	SDSD	41,7
2	SFH6916SMD	SFH6916SMD	35
2		SJ2P	
7		SJ3P	
1	STE100P	STE100PSMD	0
1	STM32F217ZGT6	STM32F217SMD	0
1	STMPS2141STRSMD	STMPS21X1STRSMD	0
2	TJA1054SMD	TJA1054SMD	35
1	25k	TRIM_EU-T53Y	18
1	USB-SHIELD	USB-SHIELD	10
1		W237-102	3,3
1	CAN_Loger	CAN_Loger	1744
223			3950,75

Tab. 5-1-Cenová rozvaha hlavní desky

Panel			
Množství	Hodnota	Název	Cena
1		DUOLED5MM	5
1		ML34L	16
1	NOKIA_LCD_HEADERH	NOKIA_LCD_HEADERH	10
1	510R	R-EU_R0805	2
1	680R	R-EU_R0805	7
7	KS01-B	SWITCHB	13,1
1	PANEL	PANEL	100
13			231,7

Tab. 5-2-Cenová rozvaha panelu

LCD			
Množství	Hodnota	Název	Cena
2	74LVC4245ADSOT137-1	74LVC4245ADSOT137-1	13
1	TC1185-1.8VCT	ADP33091	12,3
1	TC1185-2.5VCT	ADP33091	12,3
5	100n	C-EUC0603K	0,15
2	1uF	C-EUC0805K	2
1	4,7uF	CPOL-EUA/3216-18R	2
3	10uF	CPOL-EUA/3216-18R	2
1	LT1932ES6	LT1932ES6SMD	77
1	4,7uH	L-USL1210	3
1	MMBT3904	MMBT3904	4
1	NOKIA_E51_LCD	NOKIA_LCD_HEADERH	269
1	NOKIALCD	NOKIALCDSMD	70
2	1k	R-EU_R0805	2
1	10k	R-EU_R0805	2
1	750R	R-EU_R0805	2
1	SS14	SCHOTTKY-DIODESMA	1,3
1	LCD_E51	LCD_E51	393
26			888,65

Tab. 5-3-Cenová rozvaha displeje

CAN logger			
Množství	Hodnota	Název	Cena
1		Hlavní deska	3950,75
1		Panel	231,7
1		LCD	888,65
1		Obal	100
1	ML34F-ML34F	Propojovací kabel	50
8		Doprava	100
			6021,1

Tab. 5-4-Cenová rozvaha celého zařízení

6 Závěr

Prvním bodem práce bylo seznámit se s normou ISO 11898. Její výtah je uveden v sekci 2 CAN – Controller Area Network.

Druhým bodem je vytvoření blokové struktury, která je znázorněna v sekci 3.1 Blokové zapojení. A v dalších sekcích bloku 3 je struktura podrobněji komentována a rozkreslena.

Třetím bodem byla fyzická realizace celého zařízení. Tento postup je popsán taktéž v sekci 3. Obrázky a fotky zařízení jsou připojeny v sekci 8 Přílohy Obrázky a fotky zařízení.

Čtvrtým bodem práce je vytvoření řídicího programu. Řídicí program je popsán v sekci 3.6 Program a jeho obsluha v sekci 3.5.14 Obsluha programu.

Pátým a posledním bodem, je rozvaha nad použitím dalších komunikačních periférií jako je USB a Ethernet. Tato diskuze je řešena v sekci 4 Pokračování vývoje.

Nejvíce práce a času na celém projektu zabrala jeho fyzická realizace, což způsobilo snížení nároku na množství funkcí implementovaných do zařízení pouze na ty stanovené v zadání práce. Při následném vývoji programového vybavení vyvstaly nové požadavky na zapojení systému, které jsou popsány v sekci 4 Pokračování vývoje.

7 Literatura

- [1] Norma : ISO11898-1
- [2] Norma : ISO11898-2
- [3] Norma : ISO11898-3
- [4] Informace o komunikaci z LCD panelem z NOKIA E51. Dostupné na:
< <http://gobotronics.wordpress.com/tag/e51/>>
- [5] Informace o souborovém systému FAT. Dostupné na:
< http://elm-chan.org/fsw/ff/00index_e.html >
- [6] Informace a ukázkové aplikace zdrojů SEPIC. Dostupné na:
< <http://www.ti.com/product/LM3478> >
- [7] Konsorcium USB. Dostupné na:
< <http://www.usb.org/developers/onthego/> >
- [8] Informace o paměťových médiích SD/MMC. Dostupné na:
< <http://www.hw.cz/navrh-obvodu/reseni-pro-zapis-na-pametove-karty-sd-mcu-atmega16.html>>
- [9] Prago Board. Dostupné na:
< <http://www.pragoboard.cz>>
- [10] Informace o CAN a zapojení konektorů. Dostupné na:
< <http://www.interfacebus.com/CAN-Bus-Description-Vendors-Canbus-Protocol.html> >
- [11] Zapojení zásuvky RJ-45. Dostupné na:
< <http://www.hw.cz/navrh-obvodu/rozhrani/zapojeni-konektoru-a-kabelu-tp.html> >
- [12] Vývojové prostředí C++ pro platformu x86. Dostupné na:
< <http://www.bloodshed.net/devcpp.html> >
- [13] KEIL. Dostupné na:
< <http://www.keil.com> >
- [14] Altium. Dostupné na:
< <http://www.altium.com> >

- [15] Raisonance. Dostupné na:
< <http://www.raisonance.com> >
- [16] Atolic. Dostupné na:
< <http://www.atollic.com> >
- [17] CooCox. Dostupné na:
< <http://www.coocox.org> >
- [18] Eclipse. Dostupné na:
< <http://www.eclipse.org> >
- [19] Kosturik, K.: Controller Area Network, 2007
- [20] Knihovna ovladačů STM32F2xx . Dostupné na:
<<http://www.st.com/internet/mcu/product/250173.jsp>>
- [21] Grafické rozhraní Embedded GUI. Dostupné na:
<<http://www.st.com/internet/mcu/product/250173.jsp>>
- [22] Seznam norem ISO 11898. Dostupné na:
<http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=33422>

Seznam obrázků

Obr. 2.1.1-1-Datový rámeček	11
Obr. 2.1.1-2- Rámeček s žádostí o data	11
Obr. 2.1.2-1-Rámeček vrstvy MAC	12
Obr. 2.1.4-1-Princip CSMA-AMP	13
Obr. 2.1.6-1-Ukázka rámečku 2.0A & B čerpáno z [19].	14
Obr. 2.2-1-Časování	15
Obr. 2.2.2-1-Napětíové reprezentace stavů high speed budiče.....	16
Obr. 2.2.3-1-Napětíová reprezentace stavů low speed budiče	17
Obr. 2.2.3-2- Stavový diagram low speed budiče	18
Obr. 3.1-1-Blokové schéma zapojení	19
Obr. 3.3.1-1-Blokové schéma CAN	23
Obr. 3.3.2-1-Blokové schéma Ethernet.....	24
Obr. 3.3.3-1-Blokové schéma USB a dobíjení.....	25
Obr. 3.3.4-1-Blokové schéma LCD	26
Obr. 3.3.5-1-Blokové schéma SD/MMC	27
Obr. 3.3.6-1-Blokové schéma Zdroje	27
Obr. 3.5.1-1-Pohled shora	29
Obr. 3.5.1-2-Pohled zepředu.....	29
Obr. 3.5.1-3-Pohled zezadu.....	30
Obr. 3.5.1-4-Pohled z pravého boku.....	30
Obr. 3.5.1-5-Pohled z levého boku	30
Obr. 3.5.14-1-Obrazovka Rucni vysilani.....	36
Obr. 3.5.14-2-Obrazovka MENU	37
Obr. 3.5.14-3-Obrazovka Prijem	37
Obr. 3.5.14-4-Obrazovka Prijem nastaveni	38

Obr. 3.5.14-5-Obrazovka Vysilani	39
Obr. 3.5.14-6-Obrazovka Vysilani nastaveni.....	40
Obr. 3.5.14-7-Obrazovka Nastaveni	40
Obr. 3.5.14-8-Obrazovka CANx	41
Rovnice 3.6.1-1- Vzorec pro výpočet přenosové rychlosti	43
Rovnice 3.6.1-2-Vztah pro převod proměnné	44
Obr. 3.6.3-1-Ukázka konverzního programu	48

Seznam tabulek

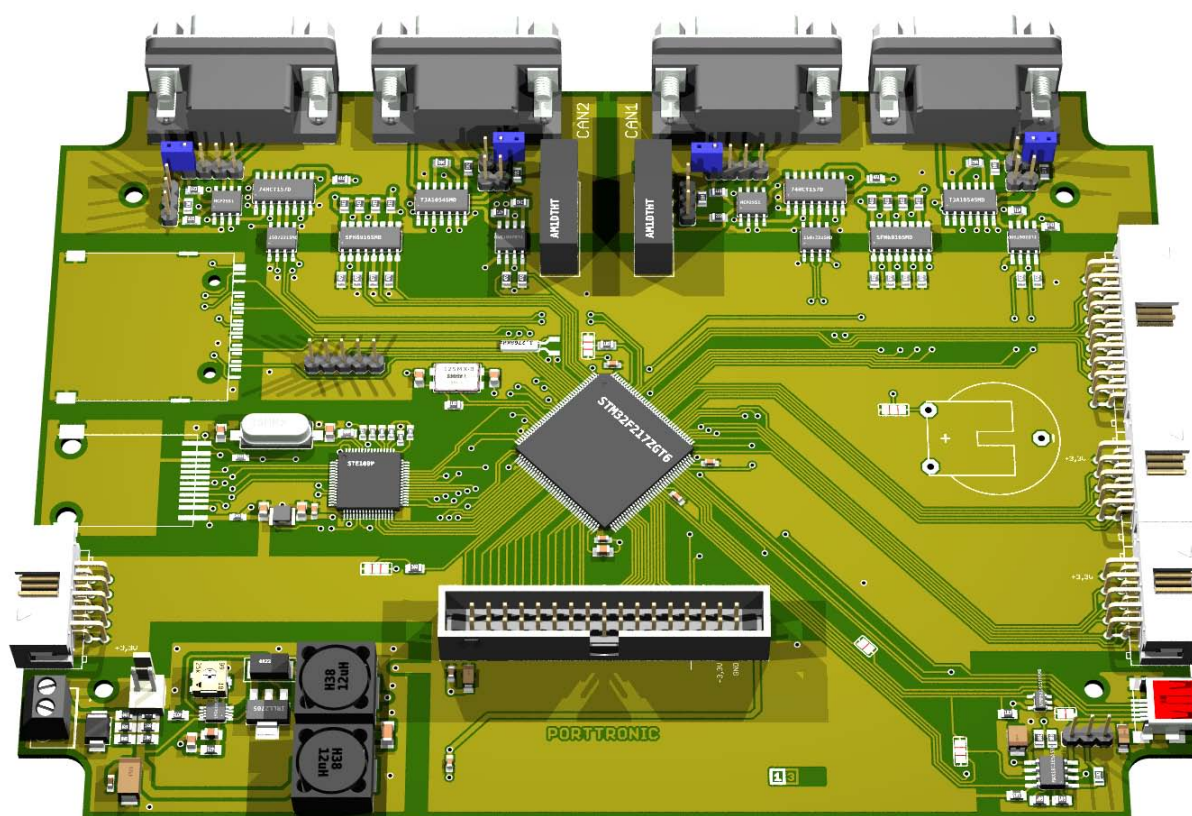
Tab. 2-1-Seznam ISO norem	10
Tab. 2.1.3-1-Ukázka stuffingu	13
Tab. 2.1.5-1-Druhy chyb	14
Tab. 2.2.2-1-Definice recesivního stavu high speed budiče	16
Tab. 2.2.2-2-Definice dominantního stavu high speed budiče.....	16
Tab. 2.2.3-1-Definice dominantního a recesivního stavu low speed	17
Tab. 2.2.3-2- Napět'ové hladiny v různých stavech low speed budiče	17
Tab. 2.2.3-3-Napět'ové hladiny v různých stavech low speed budiče pokračování.....	18
Tab. 2.2.3-4-Popis stavů low speed budiče	18
Tab. 2.2.3-5-Seznam komunikačních režimů low speed budiče.....	18
Tab. 3.2.5-1-Porovnání budičů CAN high speed	22
Tab. 3.5.2-1-Popis konektorů CAN, čerpáno ze zdroje [10].....	31
Tab. 3.5.3-1-Popis konektoru Ethernet, čerpáno ze zdroje [11]	31
Tab. 3.5.4-1-Popis konektoru mini USB, čerpáno ze zdroje [7]	32
Tab. 3.5.5-1-Popis konektoru SD/MMC, čerpáno ze zdroje [8].....	32
Tab. 3.5.6-1-Popis konektoru USART 2.....	32
Tab. 3.5.7-1-Popis konektoru SPI 3	33
Tab. 3.5.8-1-Popis konektoru I2C 2 (interní).....	33
Tab. 3.5.9-1-Popis konektoru JTAG/SWD	33
Tab. 3.5.10-1-Popis konektoru externích zařízení.....	34
Tab. 3.5.11-1-Popis konektoru analogových vstupů a výstupů.....	34
Tab. 3.5.12-1-Seznam a nastavení propojek	35
Tab. 3.6.1-1-Popis konfiguračního souboru CAN – nastavení řadiče	42
Tab. 3.6.1-2-Popis konfiguračního souboru CAN – nastavení filtrů	43
Tab. 3.6.1-3-Popis konfiguračního souboru PRIJEM.INI	44

Tab. 3.6.1-4-Popis konfiguračního souboru VYSILANI.INI	45
Tab. 3.6.2-1.Popis zprávy v datovém souboru	46
Tab. 3.6.2-2-Popis datového pole ve zprávě	46
Tab. 3.6.2-3-Struktura datové zprávy	47
Tab. 3.6.2-4-Struktura zásobníku	47
Tab. 5-1-Cenová rozvaha hlavní desky	58
Tab. 5-2-Cenová rozvaha panelu	59
Tab. 5-3-Cenová rozvaha displeje.....	59
Tab. 5-4-Cenová rozvaha celého zařízení	59

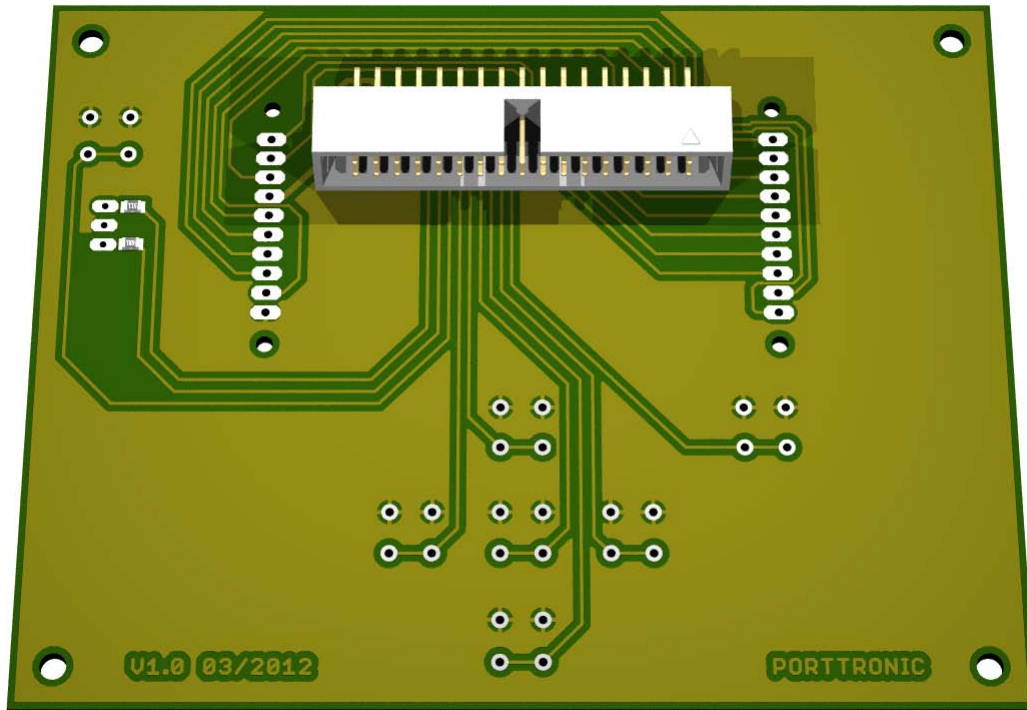
8 Přílohy

Seznam příloh

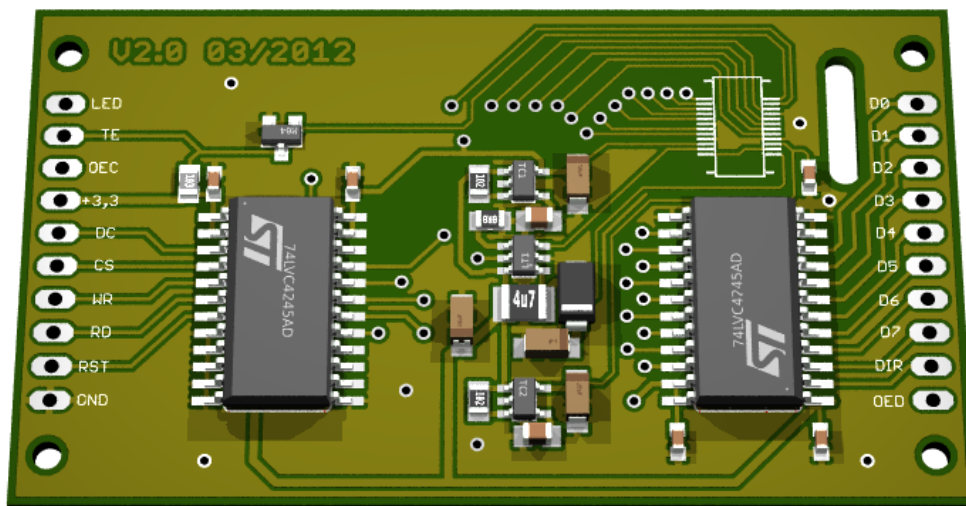
- Obrázky a fotky zařízení
- Hlavní deska (zapojení)
- Panel (zapojení)
- Displej (zapojení)
- Hlavní deska DPS + osazovací plán
- Panel DPS + osazovací plán
- Displej DPS + osazovací plán



Obrázek 8-1-Hlavní deska



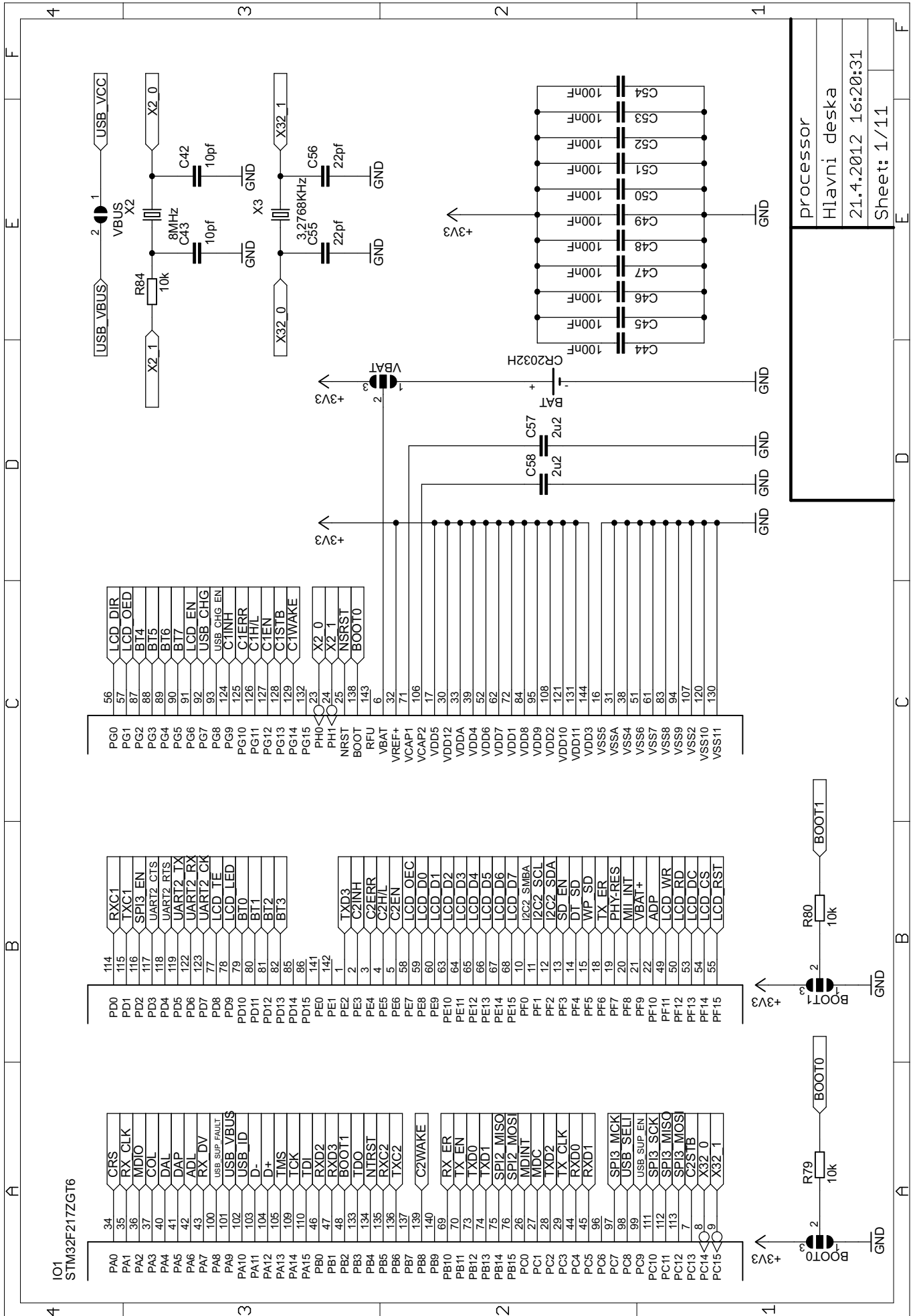
Obrázek 8-2-panel



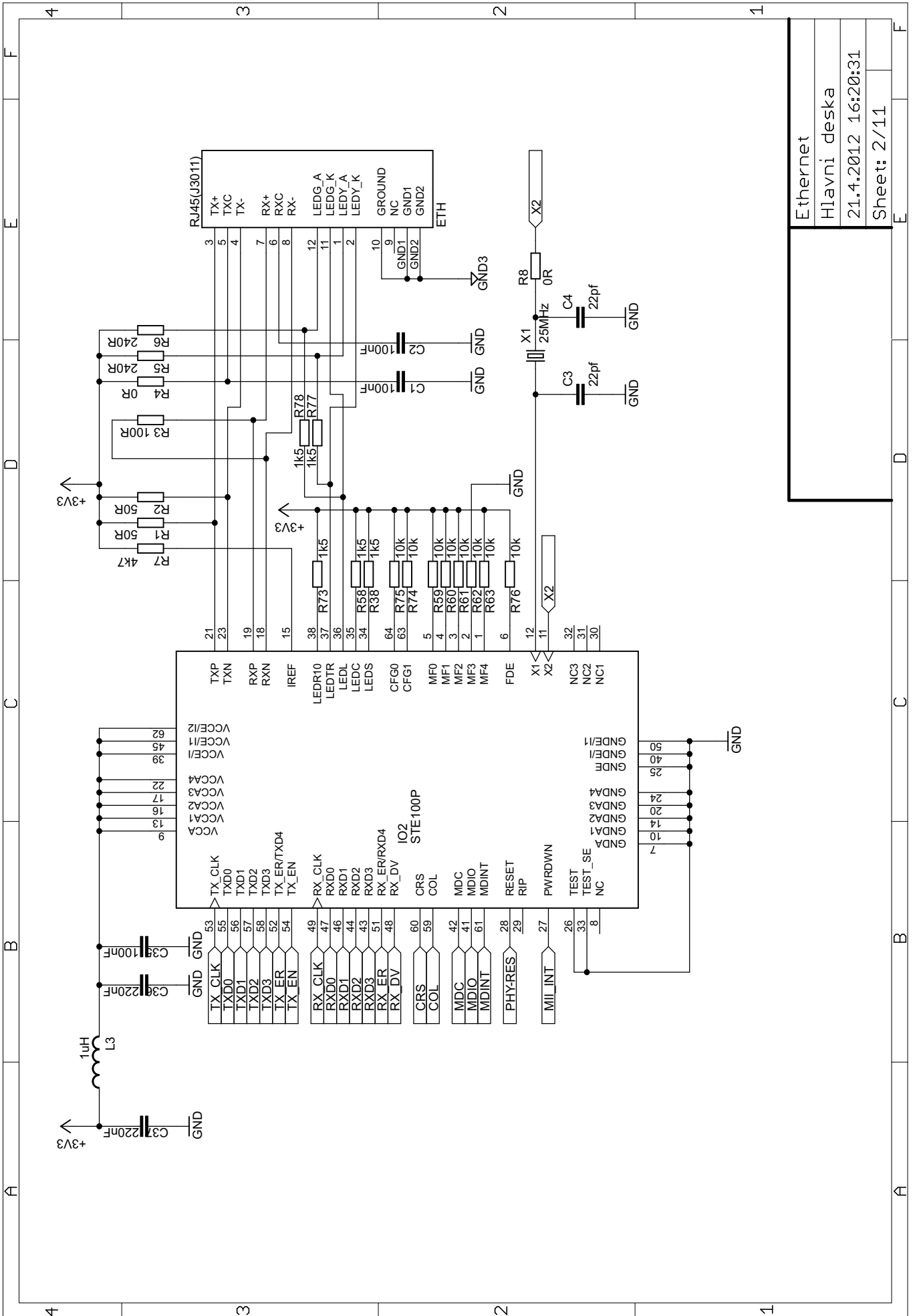
Obrázek 8-3-displej



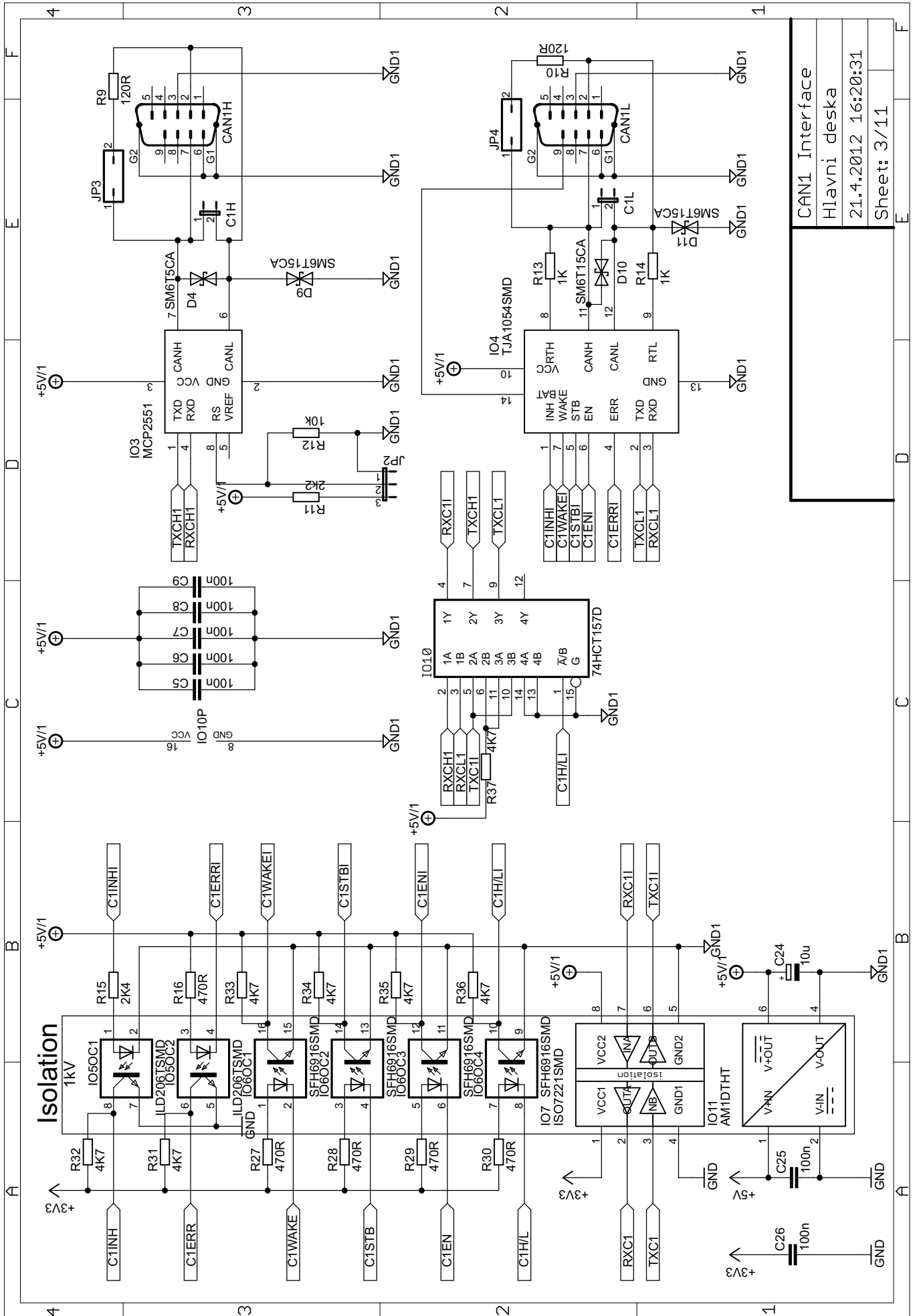
Obrázek 8-4-Fotky zařízení



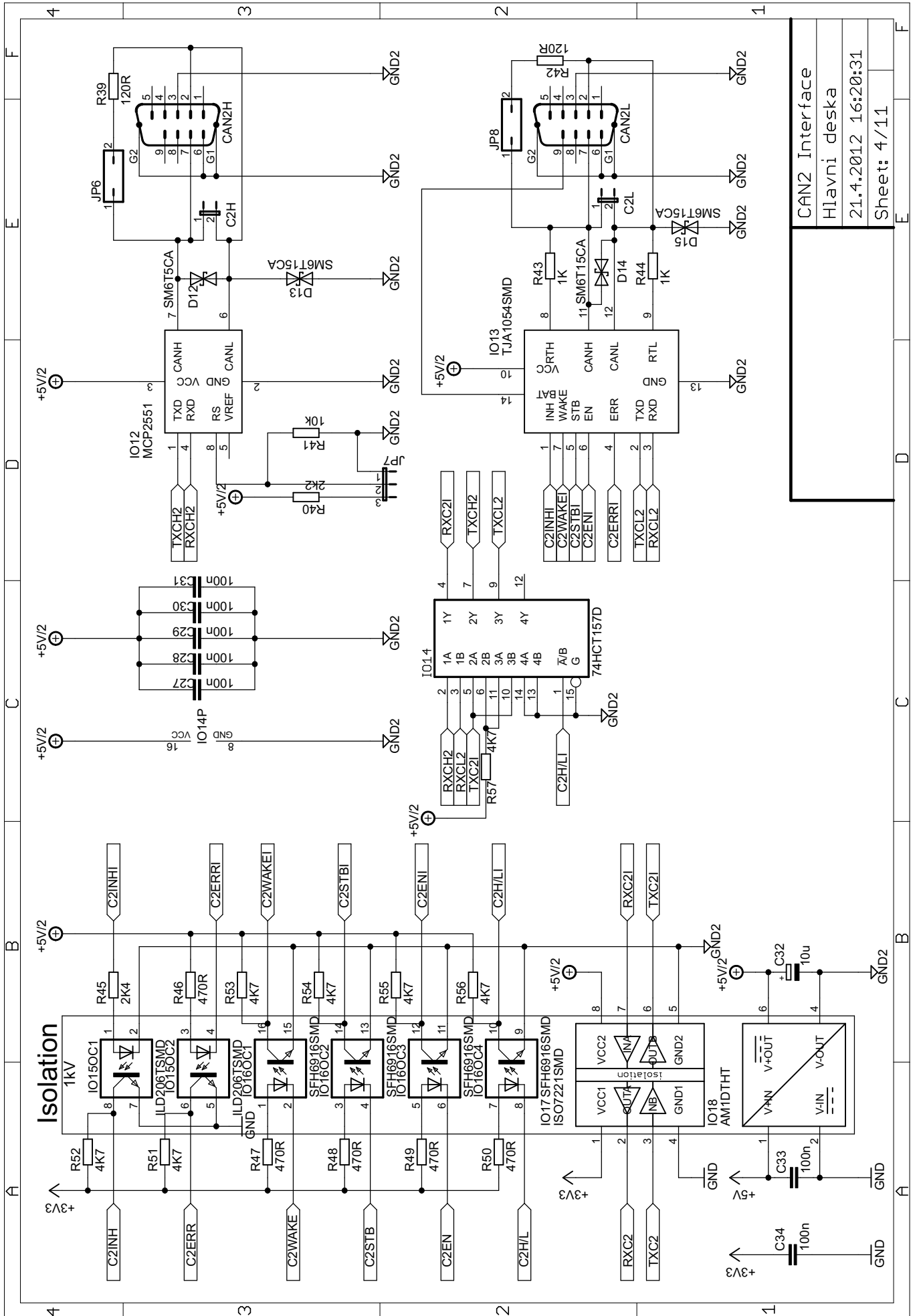
processor
Hlavní deska
21.4.2012 16:20:31
Sheet: 1/11

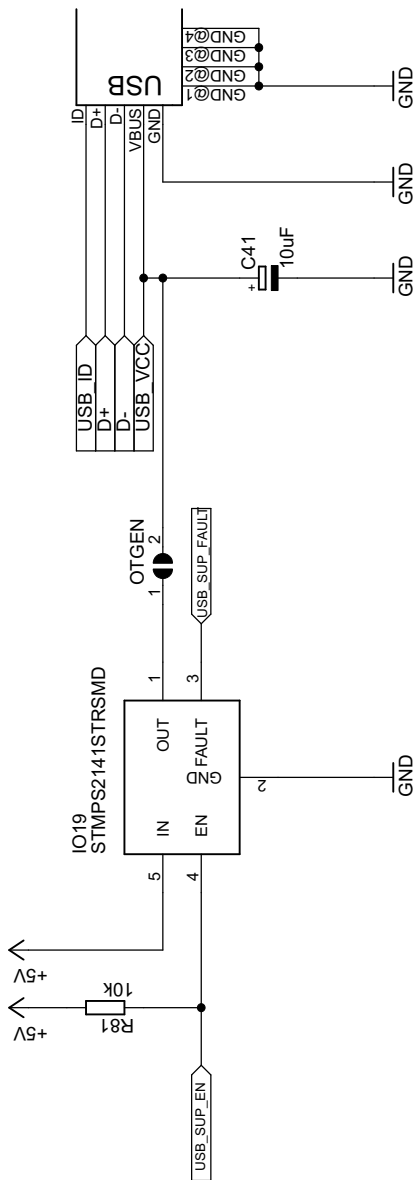


Ethernet
Hlavní deska
21.4.2012 16:20:31
Sheet: 2/11

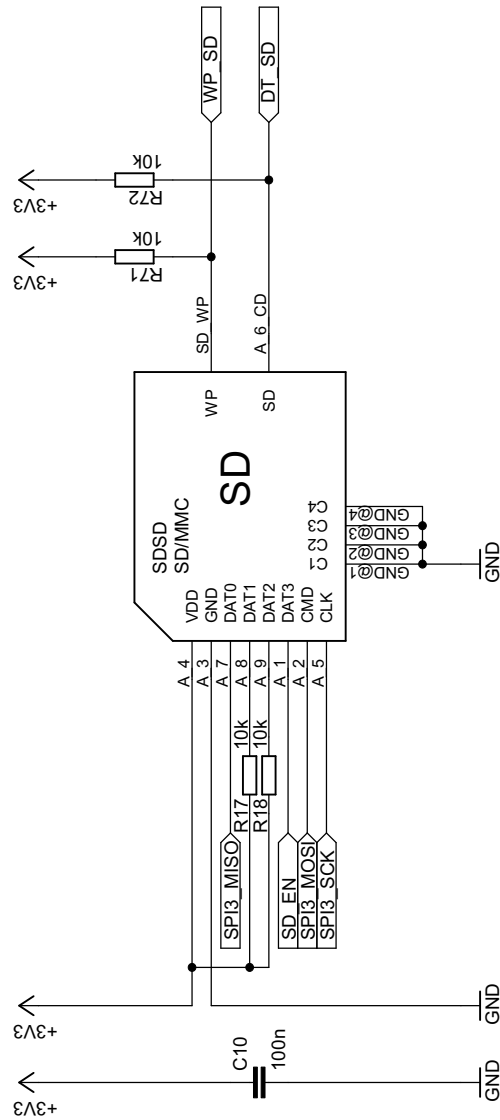


CAN1 Interface
Hlavní deska
21.4.2012 16:20:31
Sheet: 3/11

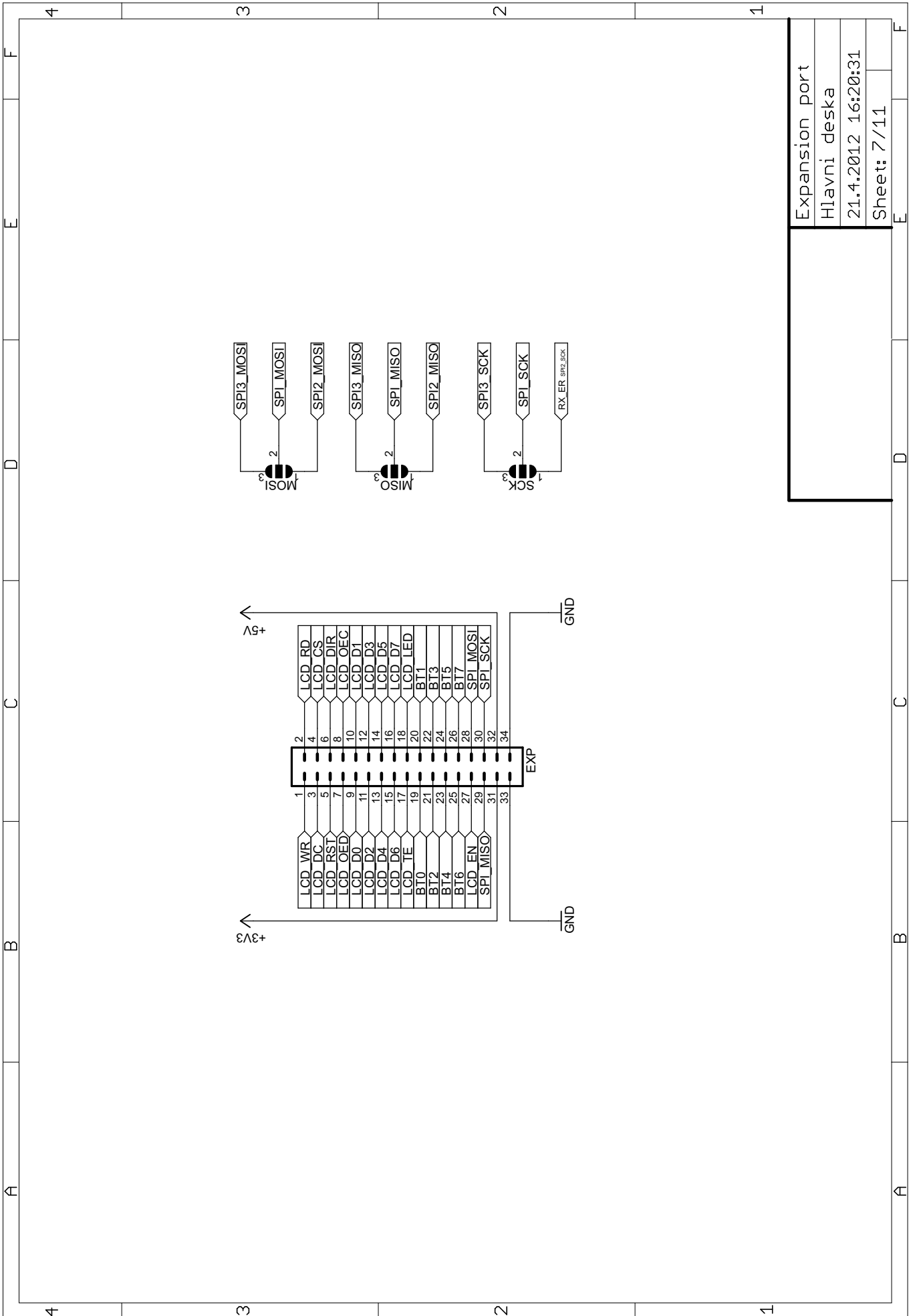




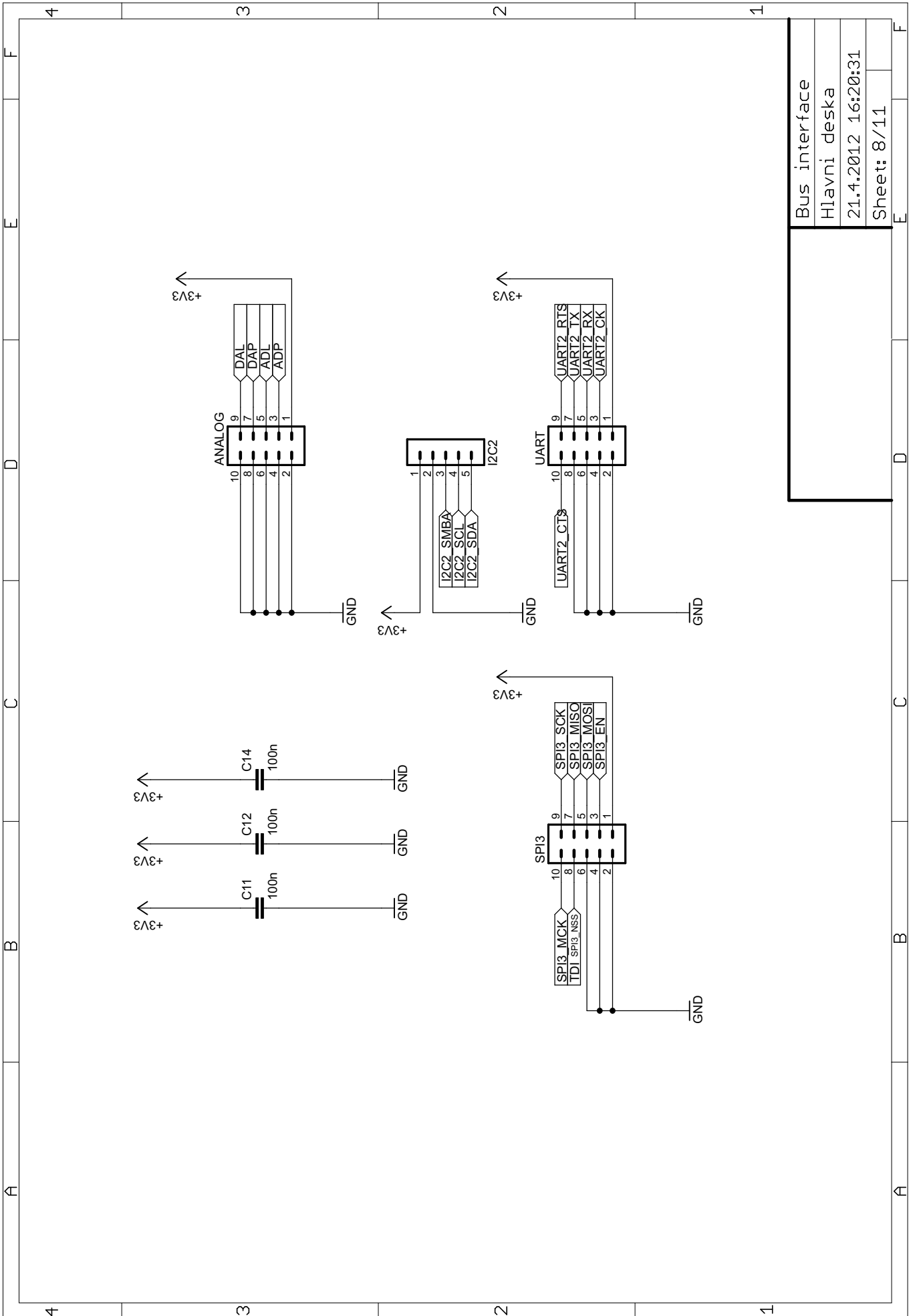
USB
Hlavní deska
21.4.2012 16:20:31
Sheet: 5/11



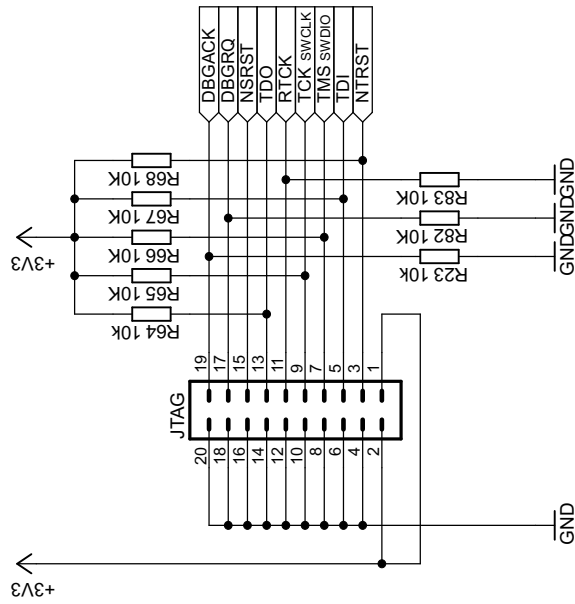
Memory card
Hlavní deska
21.4.2012 16:20:31
Sheet: 6/11



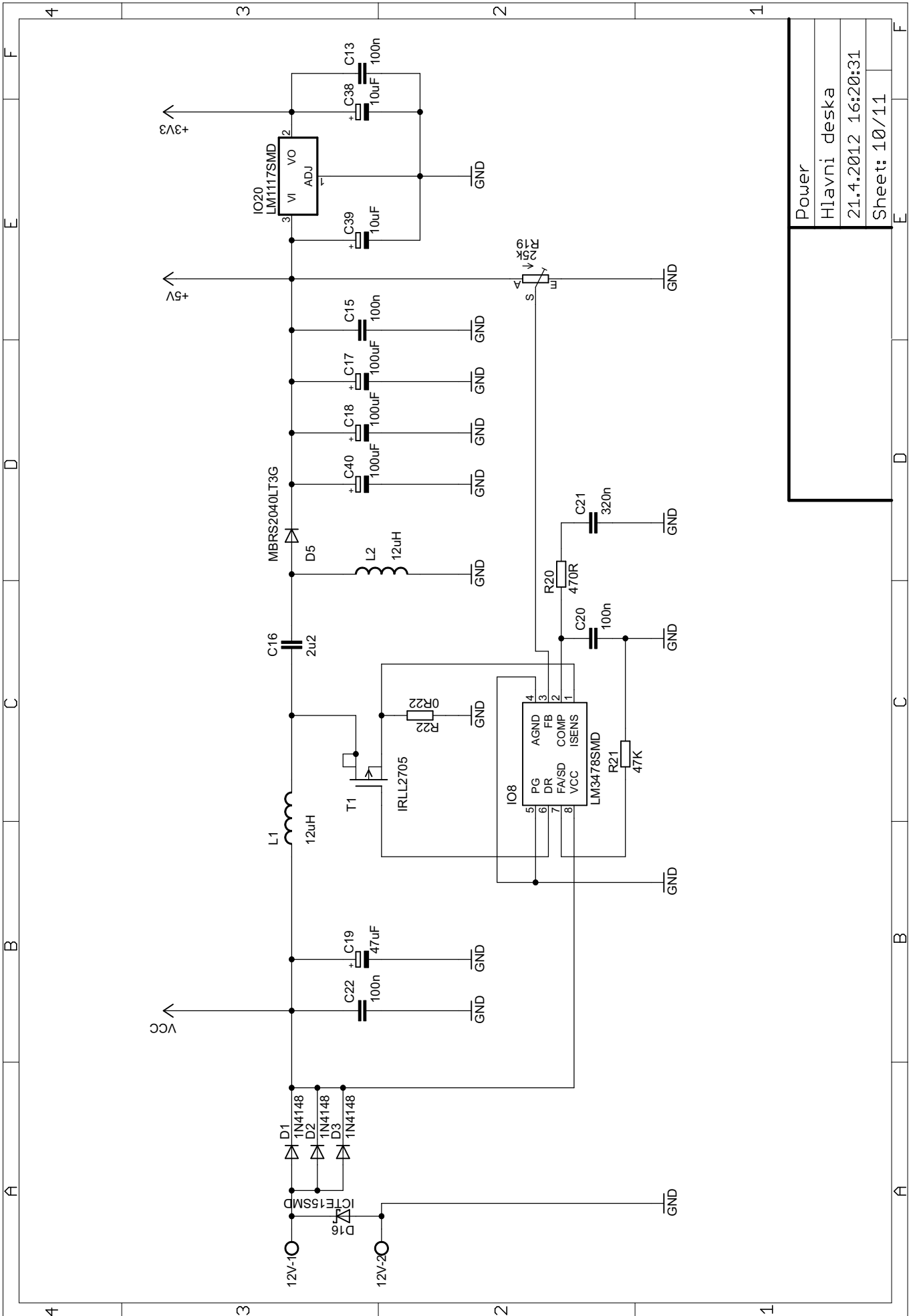
Expansion port
Hlavní deska
21.4.2012 16:20:31
Sheet: 7/11



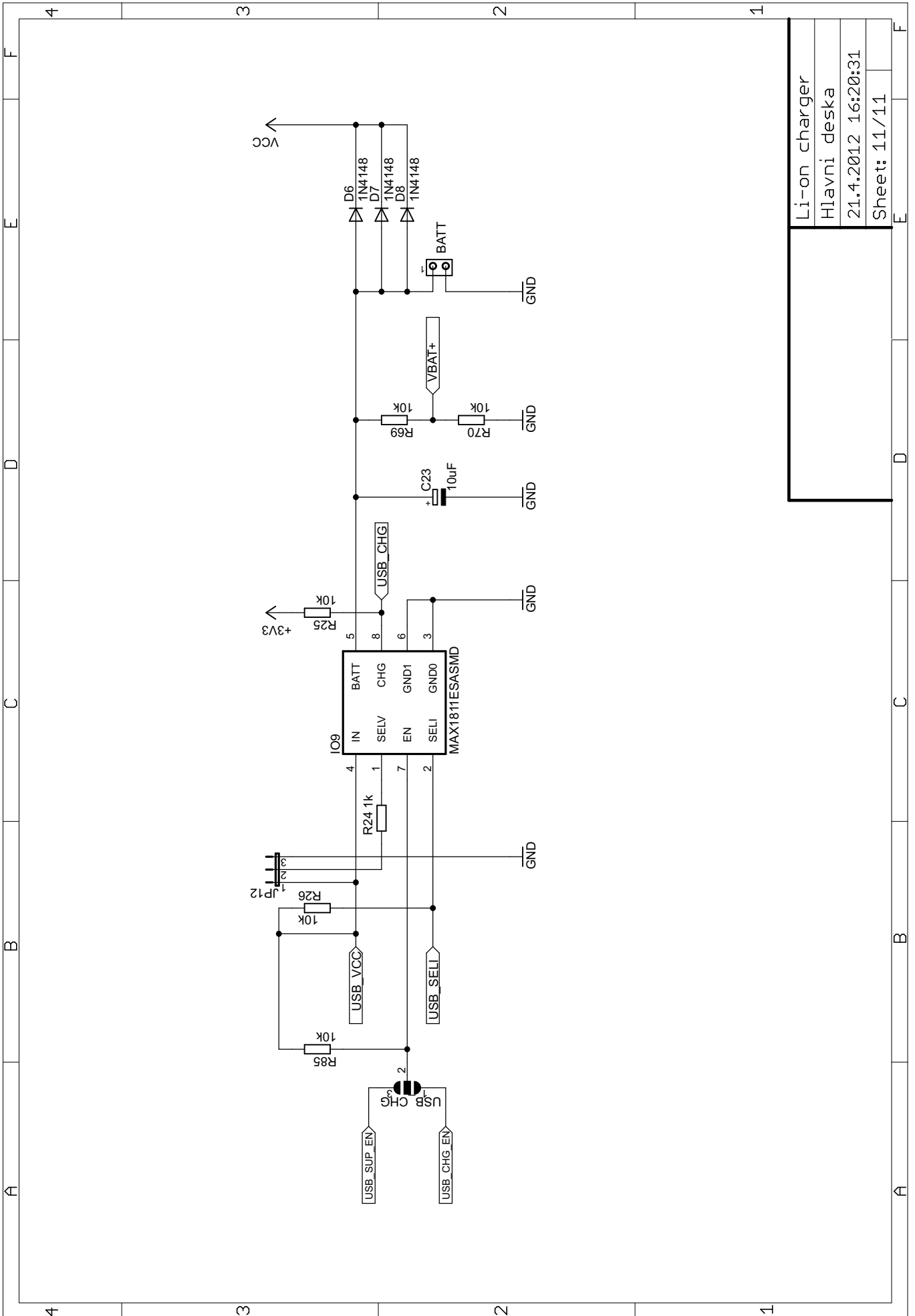
Bus interface
Hlavní deska
21.4.2012 16:20:31
Sheet: 8/11



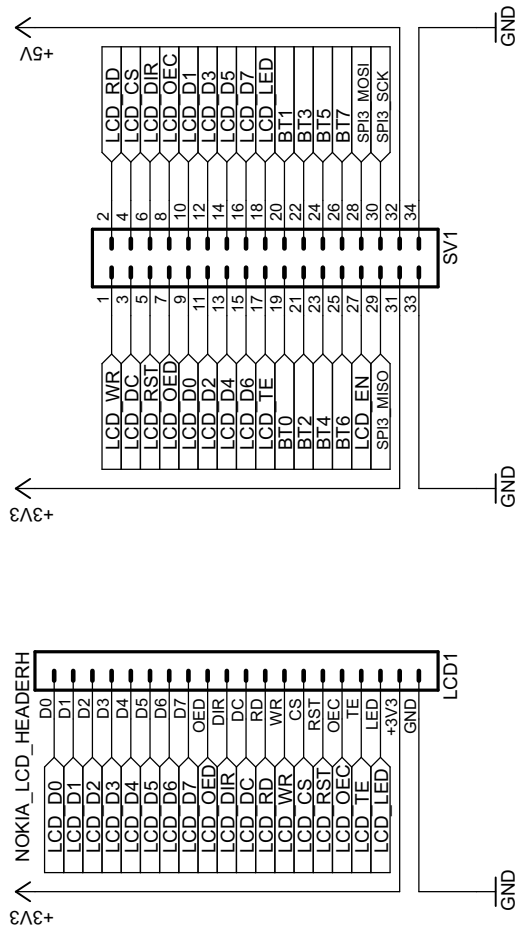
JTAG & SWD
Hlavní deska
21.4.2012 16:20:31
Sheet: 9/11



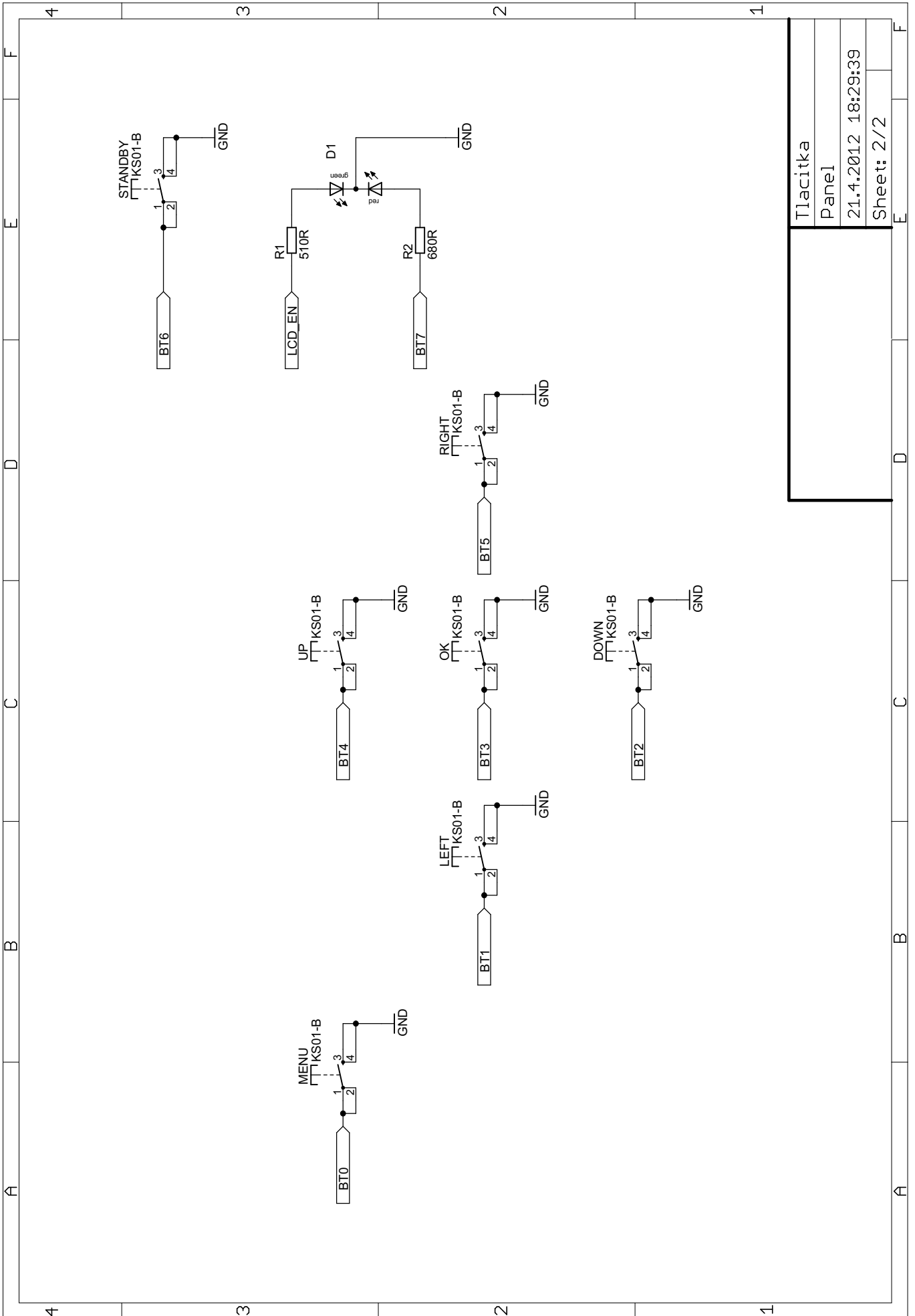
Power
Hlavní deska
21.4.2012 16:20:31
Sheet: 10/11



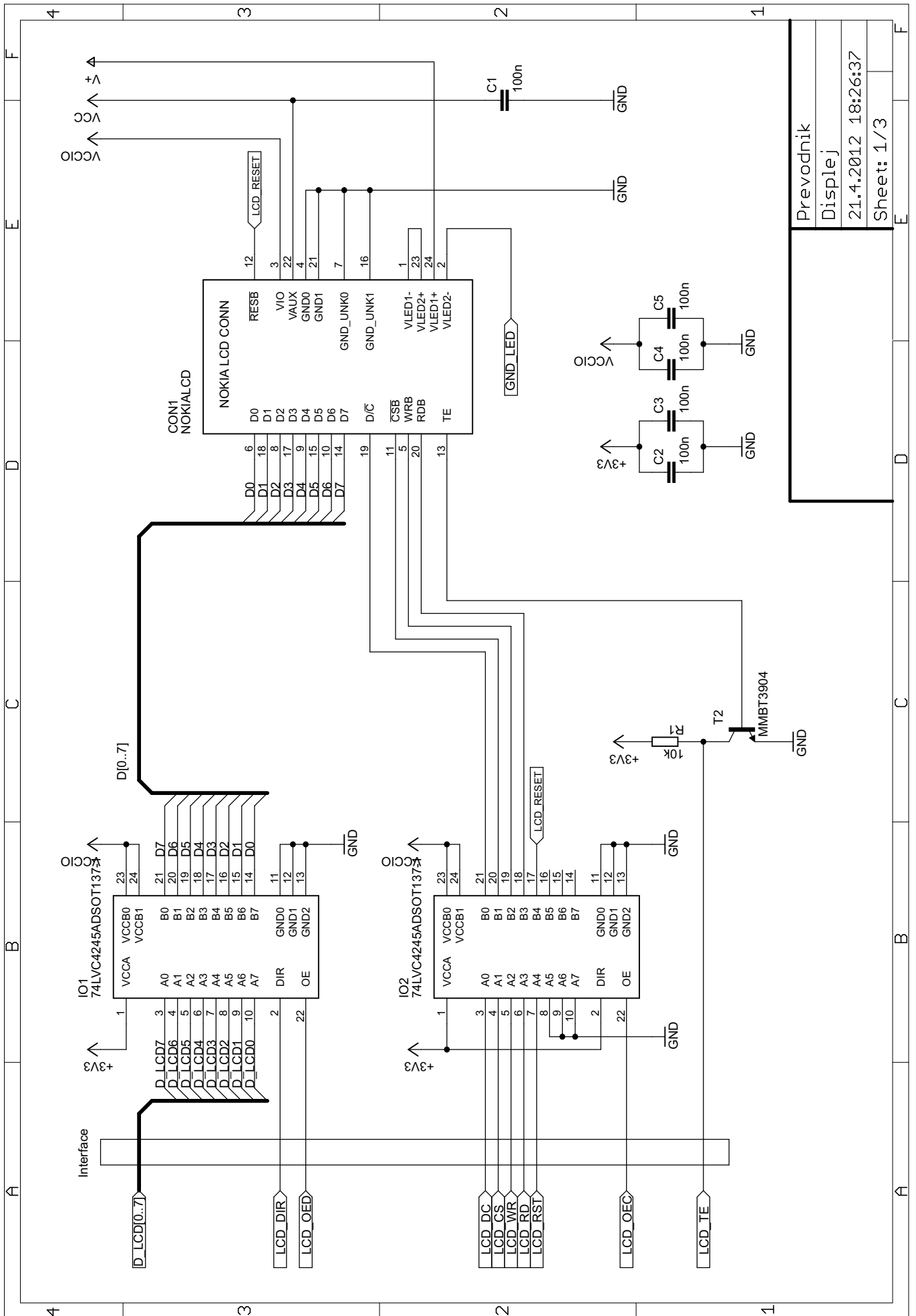
Li-on charger
Hlavní deska
21.4.2012 16:20:31
Sheet: 11/11



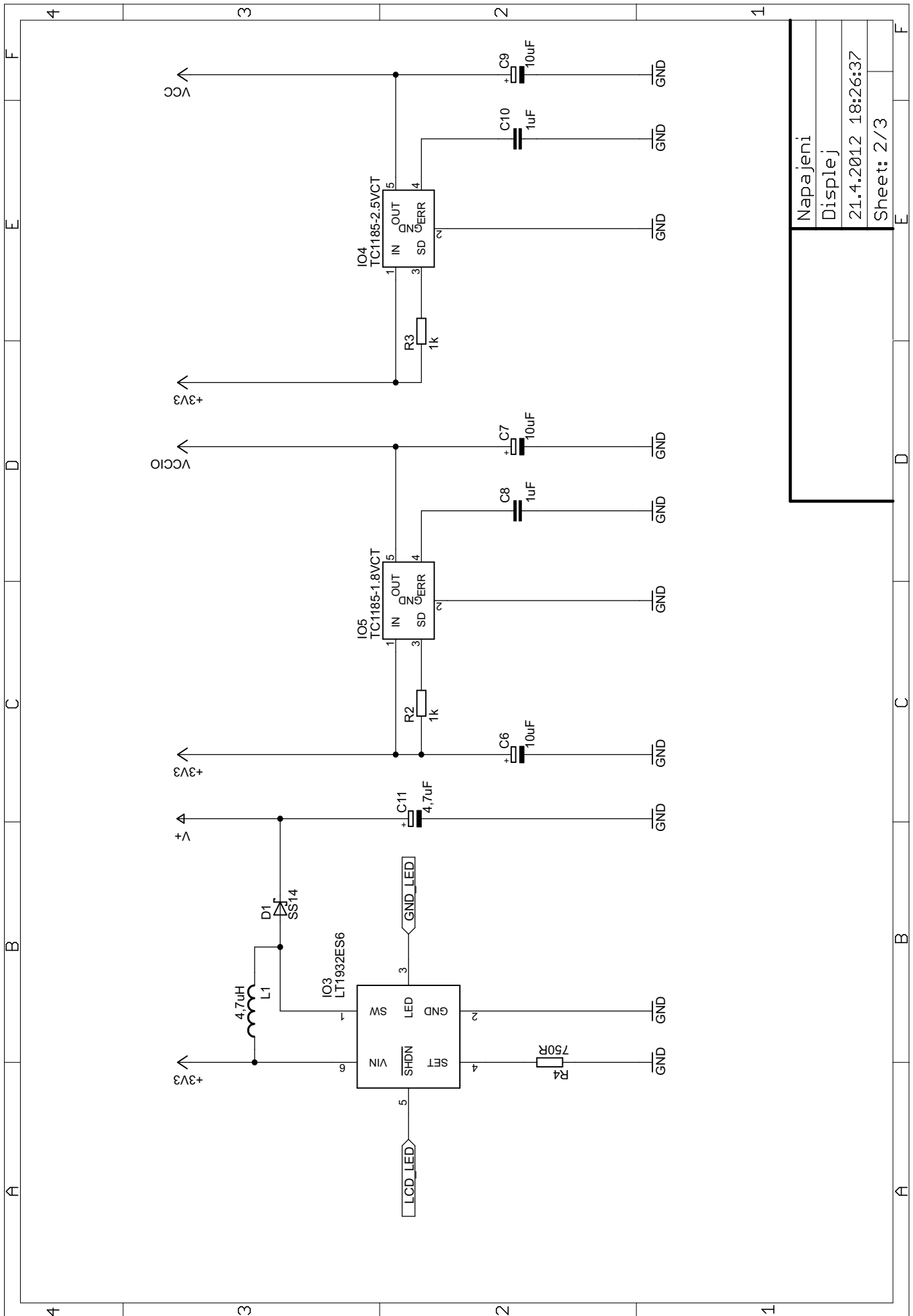
Konektory
Panel
21.4.2012 18:29:39
Sheet: 1/2



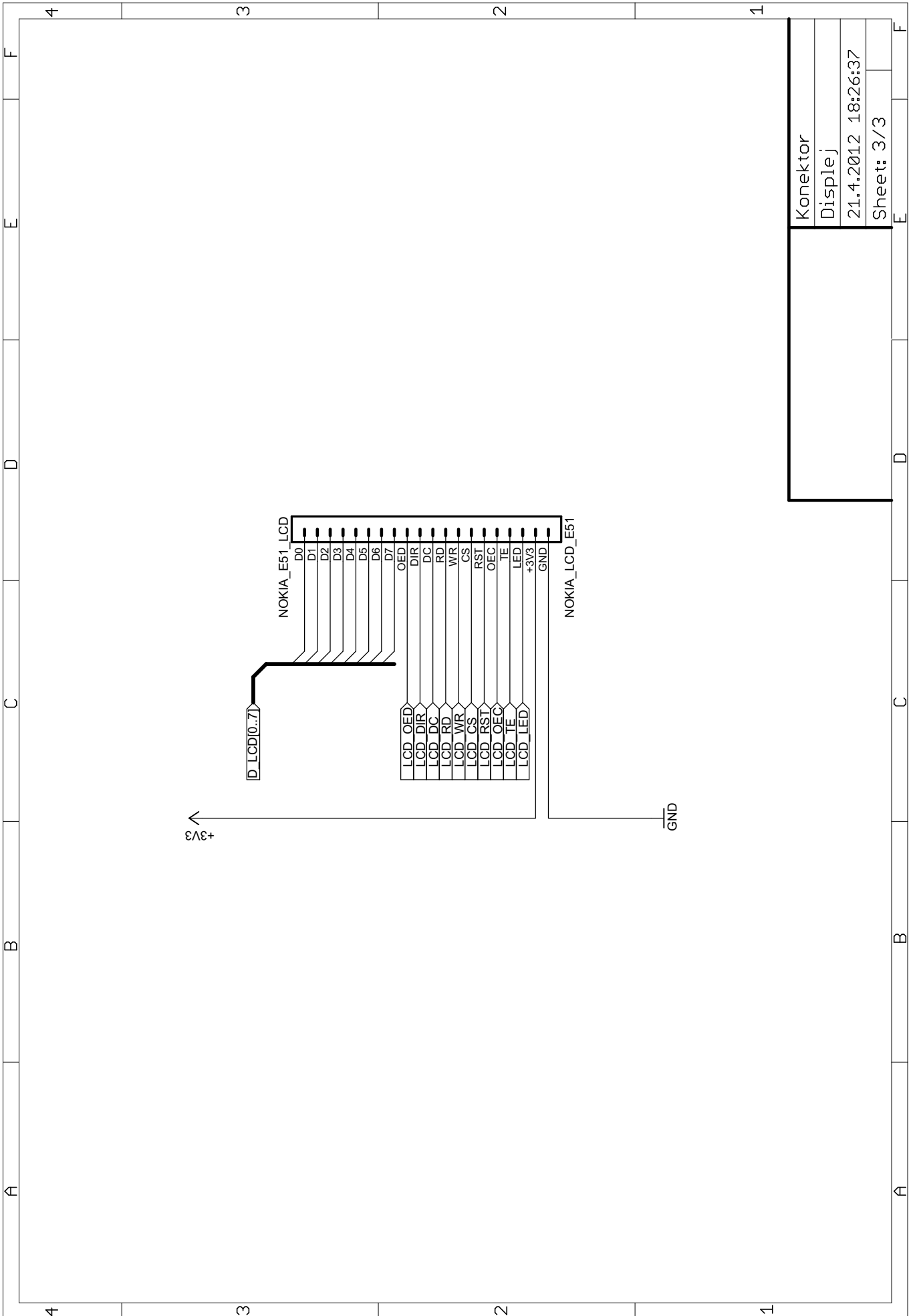
Tlacička
Panel
21.4.2012 18:29:39
Sheet: 2/2



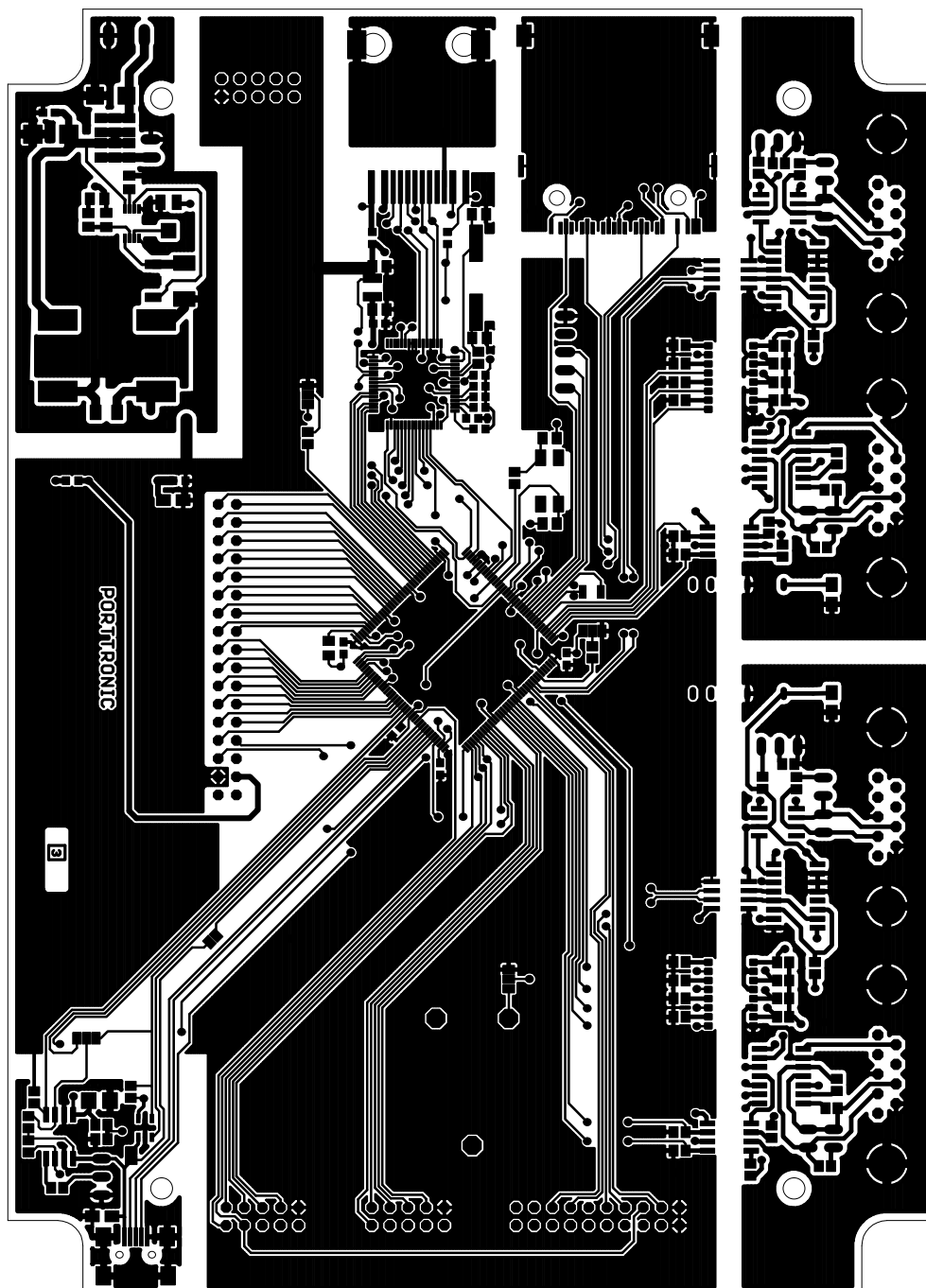
Prevodník
Displej
21.4.2012 18:26:37
Sheet: 1 / 3



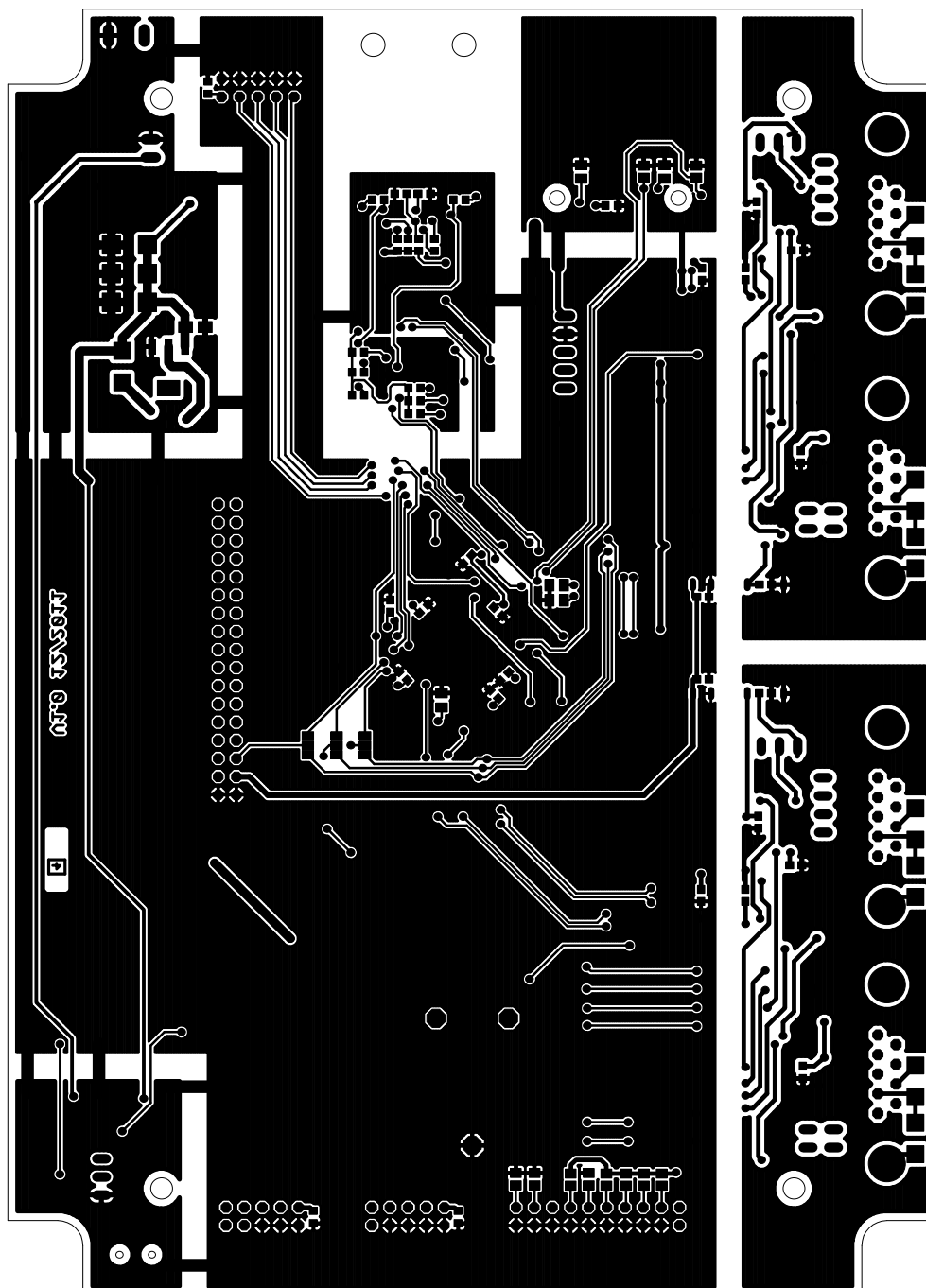
Napajeni
Display
21.4.2012 18:26:37
Sheet: 2/3



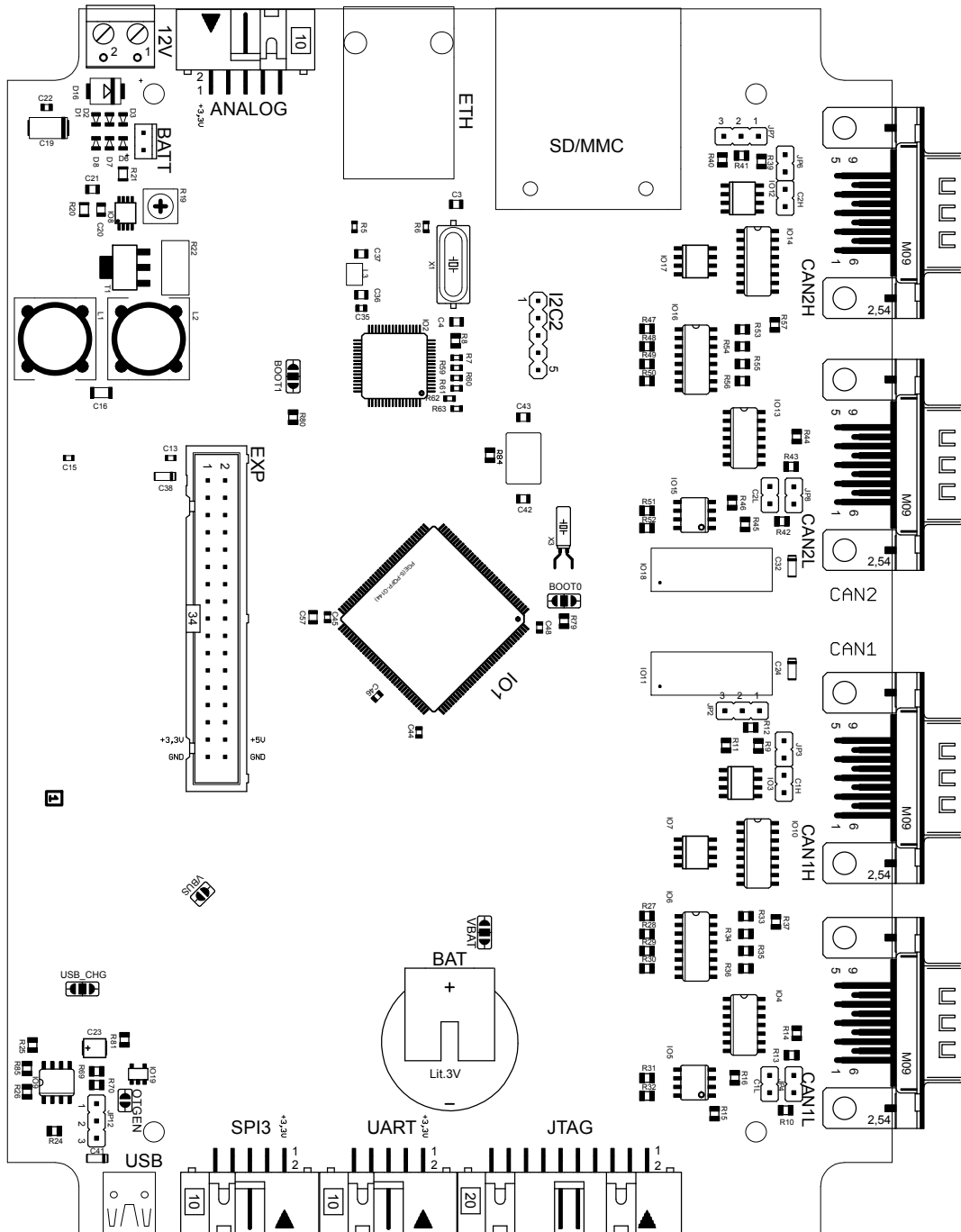
Konektor
Displej
21.4.2012 18:26:37
Sheet: 3/3



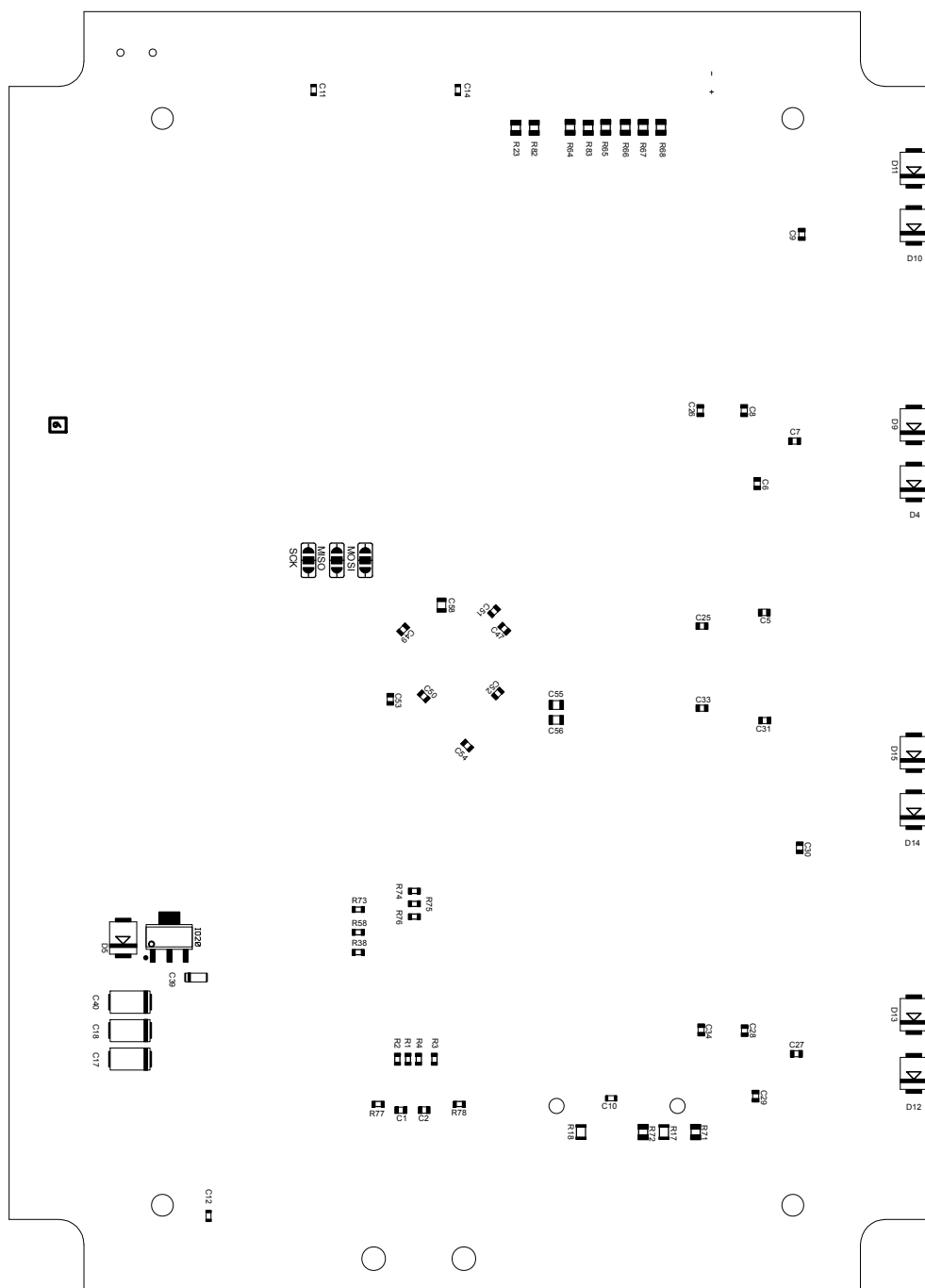
Hlavní deska DPS vrchní strana



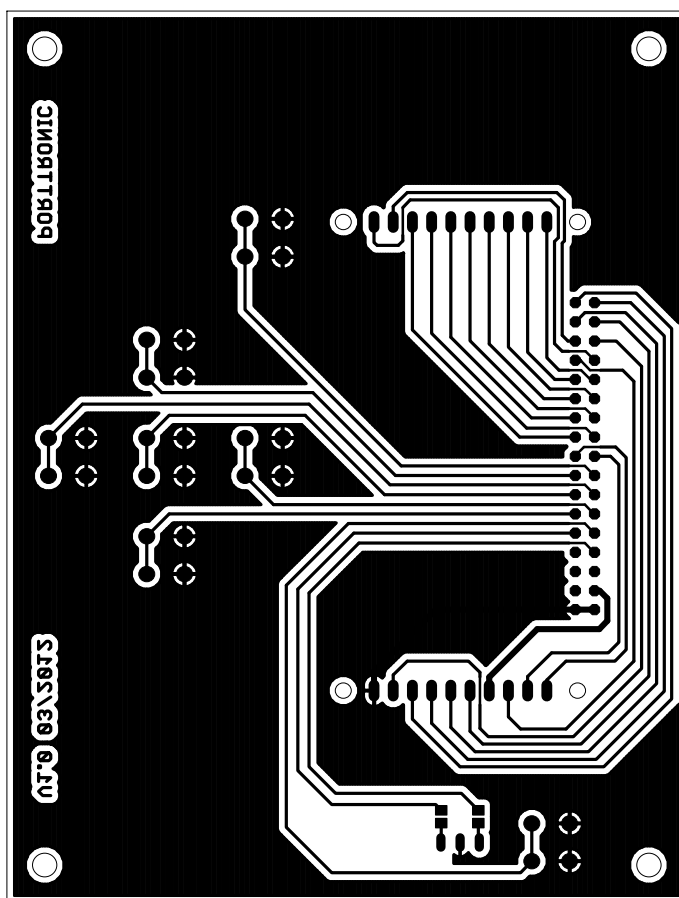
Hlavní deska DPS spodní strana



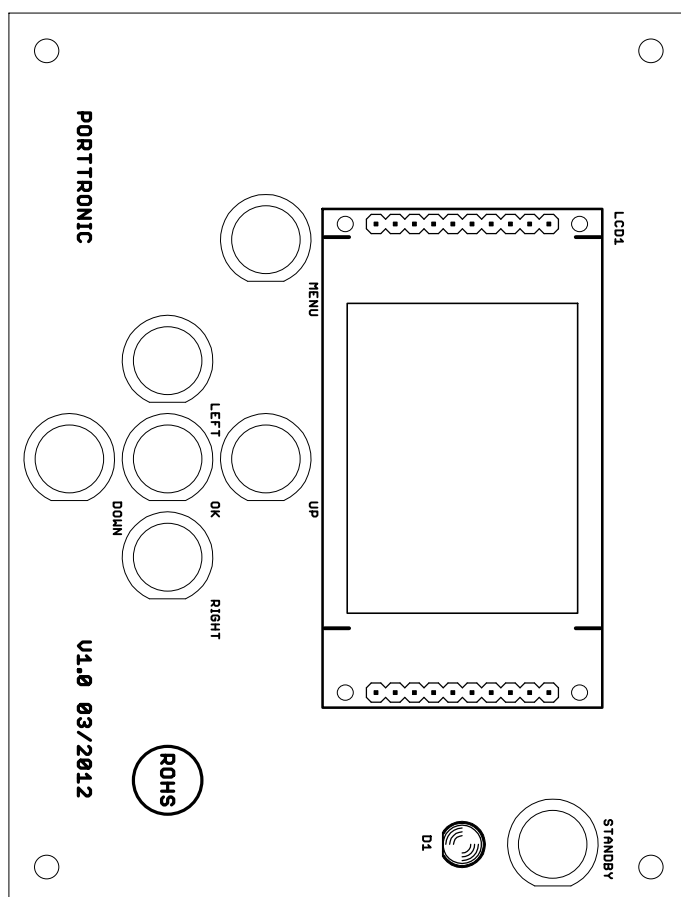
Hlavní deska osazovací plán vrchní strana



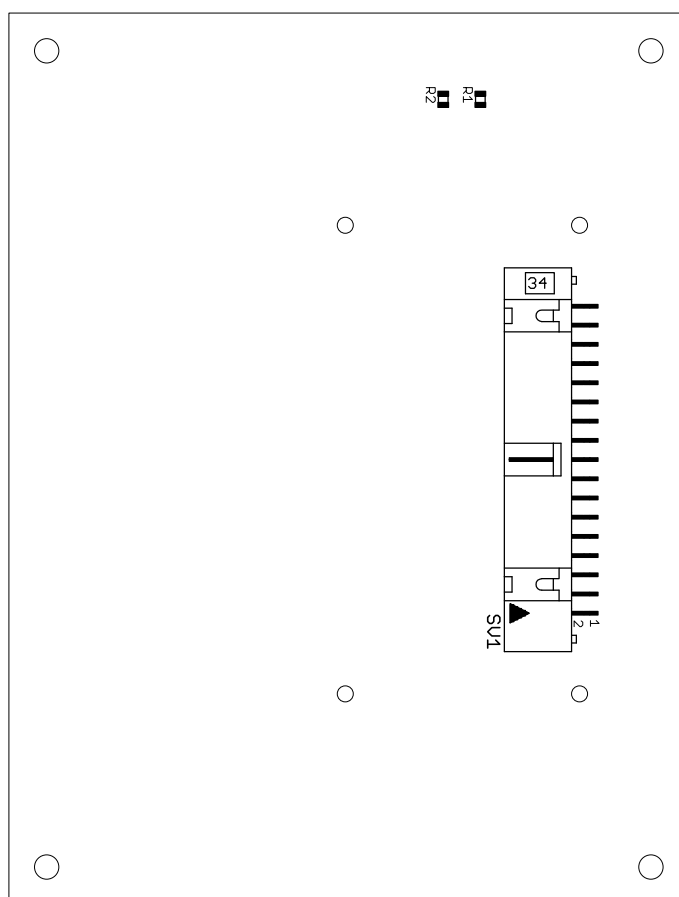
Hlavní deska osazovací plán spodní strana



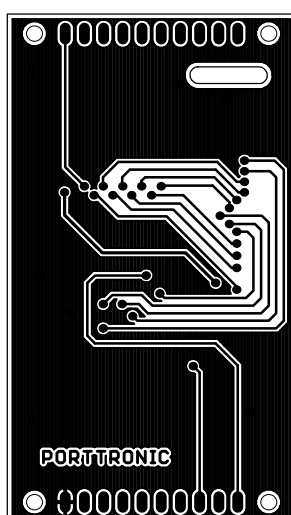
Panel DPS spodní strana



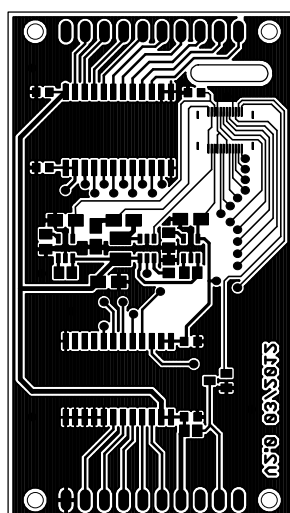
Panel osazovací plán vrchní strana



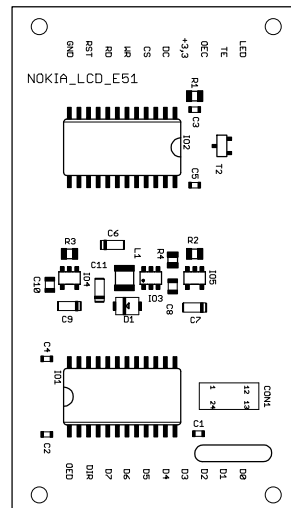
Panel osazovací plán spodní strana



Displej DPS vrchní strana



Displej DPS spodní strana



Displej osazovací plán spodní strana