



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

NÁVRH EMBEDDED SYSTÉMU

EMBEDDED SYSTEM DESIGN

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. TOMÁŠ GÁBOR

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. TOMÁŠ GÖTTHANS, Ph.D.

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Tomáš Gábor

ID: 134304

Ročník: 2

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Návrh embedded systému

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s principy činnosti vestavěných systémů. Zaměřte se především na systémy s linuxovým jádrem. Pokuste se na testovací platformě vytvořit grafické rozhraní pro připojenou periférii. Na základě získaných zkušeností navrhnete prototyp vlastního počítače.

Navržený prototyp realizujte a otestujte. Vytvořte dokumentaci navrženého systému.

DOPORUČENÁ LITERATURA:

[1] RAGHAVAN, P., LAD, A., NEELAKANDAN, S. Embedded Linux System Design and Development, CRC Press, 2005.

[2] Olimex [online]. [cit. 10.5.2014]. Dostupné na: https://www.olimex.com/wiki/Main_Page.

Termín zadání: 9.2.2015

Termín odevzdání: 21.5.2015

Vedoucí práce: Ing. Tomáš Götthans, Ph.D.

Konzultanti diplomové práce:

doc. Ing. Tomáš Kratochvíl, Ph.D.

Předseda oborové rady

UPOZORNĚNÍ:

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

Abstrakt

Tato práce se zabývá použitím vývojové desky s procesorem s architekturou ARM jako platformou pro vzdálenou interakci s připojenou periferií. Popisuje praktické zkušenosti s vývojovou deskou A20-OLinUXino-Micro-4GB a její vlastnosti. Dále rozebírá použitý hardware a vyvinutou testovací periferii. Následující část je věnována programování softwaru použitým při komunikaci mezi vývojovou deskou a periferií. Práce pokračuje částí zabývající se návrhem vlastního embedded systému od blokového schématu přes výběr komponent a návrh zapojení až k návrhu DPS. V textu jsou popsány praktické kroky návrhu DPS a shrnutí návrhu propojení procesoru a operačních pamětí. Závěr práce je věnován popisu nastavení a kalibrace operační paměti a návrhu rozšiřujících modulů.

Klíčová slova

Allwinner A20, ARM, embedded, Linux, OLinUXino

Abstract

This work deals with application of ARM based development board as platform for remote controlling of attached peripherals. It describes practical experiences with A20-OLinUXino-Micro-4GB development board and its available options. Used hardware and testing peripherals are described in the next part along with programming of the software for web interface and for communication in between development board and peripherals. The second half of this document describes design process of own embedded system beginning with schematic, continuing with PCB wiring. Each step of PCB wiring process is described with required details in text, including crucial connections between processor and memory chips. The last part contains details about setting a calibration of memory chips and designing optional peripheral modules

Keywords

Allwinner A20, ARM, embedded, Linux, OLinUXino

GÁBOR, T. *Návrh embedded systému*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2015. 41 s., 7 s. příloh. Diplomová práce. Vedoucí práce Ing. Tomáš Götthans, Ph.D.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma „Návrh embedded systému“ jsem vypracoval samostatně pod vedením vedoucího ročníkové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne

.....
podpis autora

Na tomto místě bych rád poděkoval svému vedoucímu, rodině, přátelům a všem dalším, bez kterých by tato práce nemohla vzniknout.

Obsah

1	Úvod	1
2	Embedded systém	2
3	Hardware	4
3.1	Vývojová deska OLinuXino	4
3.2	Relé deska	5
3.3	Deska s převodníkem	6
3.4	UART	6
3.5	RS-485	7
3.6	Budič symetrické sběrnice SN65176B	7
4	Software	8
4.1	Operační systém	8
4.2	Webový server	9
4.3	Python	9
4.3.1	Syntaxe	9
4.3.2	Python framework Flask	10
4.3.3	Práce se sériovým rozhraním	10
4.3.4	Práce s GPIO piny	10
4.3.5	Práce s UDP sokety	11
4.4	Webové vývojové prostředí	11
4.5	Protokol pro komunikaci po sériové sběrnici	11
5	Sestava s připojenou periférií	12
5.1	Měření odezvy GPIO pinu	13
6	Přehled existujících řešení	15
7	Použité komponenty	16
8	KiCad	18
8.1	Eeschema	18
8.2	Pcbnew	19
8.3	Správce knihovny	20
8.4	GerbView, Bitmap2Component, Pcb calculator	21
9	Návrh embedded systému	22
9.1	Návrh zapojení	22
9.2	Návrh DPS	23
9.2.1	Vodiče s řízenou impedancí	24
9.2.2	Korekce délky vodičů	26
10	Nastavení parametrů systému	29
10.1	Kalibrace DRAM	29
10.2	Bootloader	30

11 Rozšiřující moduly	31
11.1 SATA	31
11.2 HDMI	32
11.3 USB	32
11.4 SD paměťová karta	32
11.5 Audio	32
11.6 Ethernet	32
11.7 UART, TWI, GPIO	33
11.8 Napájení	33
12 Realizace	34
13 Závěr	36
14 Přílohy	42

Seznam obrázků

1	Embedded systém v automobilu [3]	2
2	ESP-01 [4]	3
3	Vývojová deska OLinuXino A20 [5]	4
4	DPS relé desky	5
5	DPS desky s převodníkem	6
6	Schéma připojení relé desky	12
7	Náhled webového ovládacího rozhraní	13
8	Měření času přepnutí GPIO pinu, opakovaný děj	13
9	Měření času přepnutí GPIO pinu, jednotlivý děj	14
10	Blokové schéma navrženého embedded systému	15
11	Blokové schéma SoC A20 [29]	16
12	Blokové schéma AXP209 [31]	17
13	Vnořené listy ve schématu	18
14	Nástroj pro úpravu délek vodičů	19
15	Režim vysokého kontrastu	20
16	Procházení knihovny footprintů	21
17	Výpočty na mikropáskovém vedení v Pcb calculatoru	21
18	Zapojení spodní (vlevo) a vrchní (vpravo) propojovací lišty	22
19	Detail návrhu DPS pod BGA pouzdrům	23
20	Vizualizace navržené DPS, vrchní strana	24
21	Vizualizace navržené DPS, spodní strana	25
22	Motiv propojení procesoru a operačních pamětí	27
23	Zobrazení délky vodiče v KiCadu	28
24	Vizualizace výsledků testování DRAM [40]	30
25	Rozměry DPS a poloha propojovacích lišt	31
26	DPS před osazením	34
27	Hlavní schéma	42
28	Schéma zapojení operační paměti	43
29	Schéma zapojení napájení	44
30	Schéma zapojení USB a SD karty	45
31	Schéma zapojení Ethernetu	46
32	Schéma zapojení SATA, HDMI a USB	47
33	Schéma zapojení napájení SoC a krystalů	48

Seznam tabulek

1	Parametry vývojové desky [7]	4
2	Struktura komunikačního paketu	11
3	Přehled existujících embedded systémů	15
4	Návrhová pravidla pro DPS	24
5	Vlastnosti DPS	25
6	Přehled DRAM vodičů [37]	26
7	Srovnání vyvinutého embedded systému s Raspberry Pi 2 [28, 45]	35

1 Úvod

V dnešní době existuje mnoho zařízení, které pohání embedded počítač. Toto řešení umožňuje měnit chování zařízení za chodu, vylepšovat nebo opravovat chyby, přidávat nové vlastnosti a to vše bez jakéhokoliv fyzického zásahu do zapojení zařízení. Cena použitého SoC procesoru se pohybuje pod 10 \$ za kus [1]. To umožňuje použití i v jednodušších aplikacích a tím napomáhá většímu rozšíření této platformy.

Prvním cílem práce je oživení vývojové desky, což zahrnuje výběr a instalaci vhodného operačního systému, nejspíše Linuxu. Výstupem by měla být hotová aplikace, pomocí které půjde vzdáleně ovládat připojenou periferii. Aplikace by měla být multiplatformní a snadno modifikovatelná pro další možné úpravy. Mimo aplikaci je také třeba zvolit či vyvinout vhodnou testovací periferii. S tím souvisí i volba vhodného komunikačního protokolu mezi vývojovou deskou a periferií a jeho zapracování do ovládací aplikace. Dalším krokem bude návrh prototypu vlastního embedded systému a vytipování klíčových komponent. Následovat bude kompletní návrh zapojení systému závislý na vybraných komponentách a také návrh DPS. Celé snažení bude zakončeno výrobou DPS a osazením prototypu navrženého embedded systému.

2 Embedded systém

Embedded systém (nebo také vestavěný systém) je zjednodušeně řečeno počítač navržený pro jednu konkrétní úlohu. V porovnání s osobním počítačem je embedded systém neplnohodnotný, obsahuje jen potřebné části a i software může být specializovaný či upravený pro danou aplikaci. [2]

Asi nejrozšířenějším embedded systémem dnešní doby je mobilní telefon. V porovnání se stolním počítačem je menší a kompaktnější. Většinou nemá klávesnici, ovládá se dotykem. Z výstupů u něj najdeme většinou pouze USB konektor pro nabíjení a synchronizaci s PC, případně audio konektor pro připojení sluchátek. Velké rozdíly jsou také v UI, které je přizpůsobeno dotykovému ovládání a celkově odlišným požadavkům na snadnou a rychlou ovladatelnost přístroje. Jako příklad lze uvést optimalizace systému pro co nejdélší výdrž na baterii, automatické uzamykání obrazovky z důvodu nechtěných stisknutí při nošení v kapse a další.

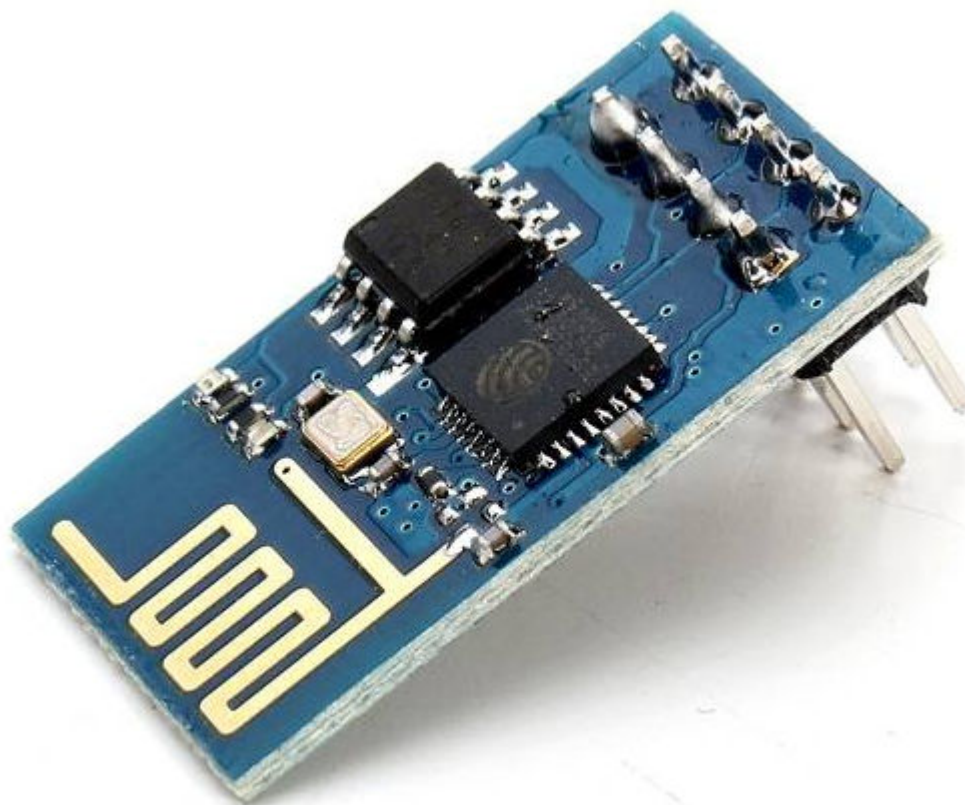
S pokračujícím vývojem v oblasti spotřební elektroniky se hranice mezi embedded a klasickými počítačovými systémy postupně stírají. Některé mobilní telefony a tablety mohou být připojeny do dokovacích stanic, která rozšiřuje jejich možnosti o připojení externího monitoru a jiných periférií přes USB. Tato konfigurace může sloužit jako náhrada osobního počítače s možností mobilní telefon odpojit a rozdělanou práci si vzít s sebou.



Obrázek 1: Embedded systém v automobilu [3]

V nedávné minulosti se na trhu objevila řada zajímavých systémů, s procesory ARM nebo podobnými, za velmi zajímavé ceny. Jako příklad uveďme vývojovou desku s procesorem ESP8266. Procesor přímo integruje Wi-Fi standardy 802.11b/g/n, pracuje v režimu AP i klient a umí komunikovat přes sběrnice UART a SPI. Deska má rozměry 24,5x14,2 mm a obsahuje nejčastěji 1 MB SPI FLASH paměť. Ovládá se přes UART

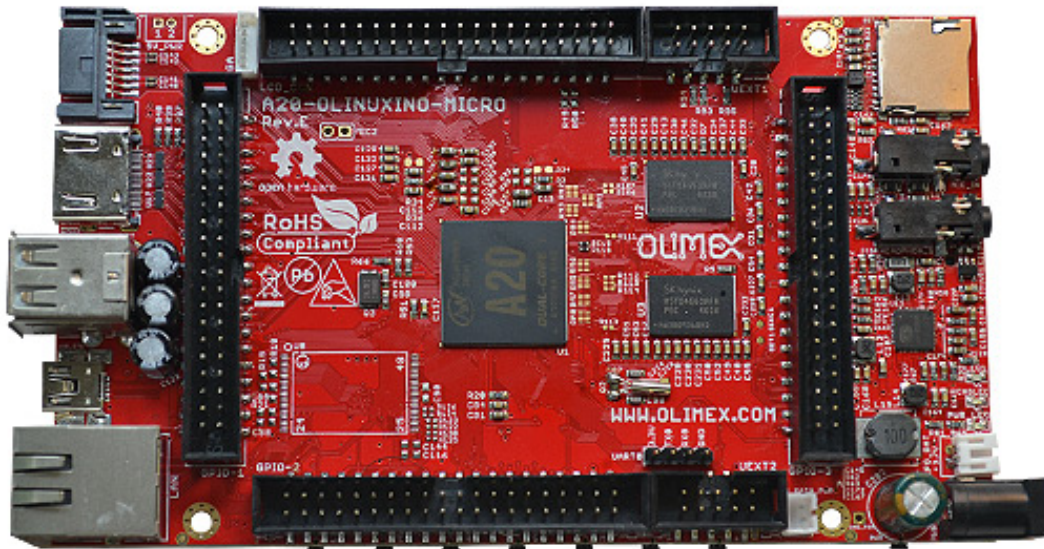
pomocí tzv. AT příkazů, podobně jako modemy. K dispozici je dokumentace a několik alternativních systémů včetně zdrojových kódů, které lze nahrát do FLASH paměti a celek dále programovat pomocí skriptů v Perlu. Na trhu lze sehnat několik různých verzí desky s cenami začínajícími na \$ 3,5 (cca 90 Kč). Toto všechno z desky činí ideálního kandidáta pro vývoj tzv. IoT (Internet of Things) zařízení. [4]



Obrázek 2: ESP-01 [4]

3 Hardware

Tato kapitola se zabývá fyzickými zařízeními. První část přiblíží vývojovou desku, další části potom periferie – relé desku a desku pro převod logických úrovní. Poslední část popisuje sběrnice RS-485 a UART a také integrovaný obvod SN65176B.



Obrázek 3: Vývojová deska OLinuXino A20 [5]

3.1 Vývojová deska OLinuXino

K dispozici jsme měli vývojovou desku OLinuXino, konkrétně verzi A20-OLinuXino-Micro-4GB. Její základní parametry jsou uvedeny v tabulce 1. Celá vývojová deska je licencována jako Open Source – na stránkách výrobce jsou ke stažení podklady pro výrobu i zdrojové kódy potřebné pro kompilaci vlastního linuxového systému. [6]

CPU	A20 dvoujádrový ARM Cortex-A7
GPU	Mali 400
Paměť RAM	1GB DDR3
Paměť FLASH	4GB NAND
SATA, 2x USB, USB-OTG, HDMI, VGA, Ethernet 100 Mbit/s, konektor pro připojení akumulátoru, audio výstup a mikrofonní vstup, 2x UEXT konektor, 160 GPIO pinů, konektory pro připojení SD/MMC a MicroSD karty, debugovací UART konektor, GPIO LED, 10 tlačítek, napájení 6 - 16 V	

Tabulka 1: Parametry vývojové desky [7]

Procesor Allwinner A20 byl navrhnout k použití například v tabletech a je doporučován pro použití s operačním systémem Android ve verzi 4.1 a vyšší. Umožňuje plynulé hardwarové dekódování videa formátu H.264 s rozlišením 2160p a kódování Full HD videa 1080p při 30 snímcích za vteřinu. [8]

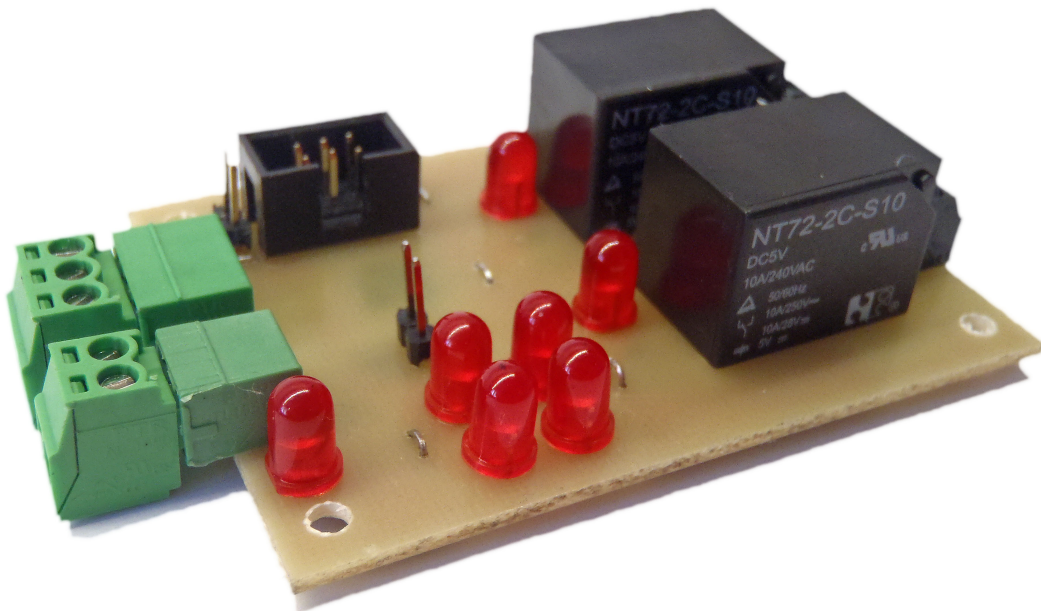
Vývojová deska disponuje řadou vstupně/výstupních konektorů, včetně 2 USB konektorů, Ethernetu a 3 univerzálních sériových linek. Konektory UEXT obsahují napájení 3,3 V, SPI a UART sběrnice a umožňují přímé připojení různých rozšiřujících modulů. Přímou podporu od výrobce, v podobě knihoven pro operační systém Linux, mají různé desky se senzory, převodníky na průmyslové sběrnice, LCD displeje, RFID čtečky apod. Všechny aktuálně nabízené rozšiřující moduly lze nalézt na webových stránkách výrobce. [7, 9]

Ve vnitřní FLASH paměti je ve výchozím stavu nahrán operační systém Android. Na spodní hraně desky se nachází několik tlačítek, pomocí kterých lze Android ovládat. Systém při startu kontroluje přítomnost zavaděče na MicroSD kartě, a ta má přednost před FLASH pamětí. Z druhého slotu, na paměťové karty SD, systém zavést nelze, slouží pouze k rozšíření paměťové kapacity systému. [7]

Desku lze napájet ze stejnosměrného zdroje napětí 6 až 16 V a výkonem alespoň 5 W. Na desce se také nachází konektor pro připojení 3,7 V Li-Po akumulátoru, který může desku napájet. Po připojení externího napájení se Li-Po akumulátor nabíjí. [7]

3.2 Relé deska

Jako periferie byla vyvinuta jednoduchá testovací deska s mikroprocesorem AVR ATmega8 a dvojicí relé. Deska dále obsahuje budič sériové sběrnice RS485 a několik LED. Všechny vstupy a výstupy jsou přivedené na univerzální svorkovnice.



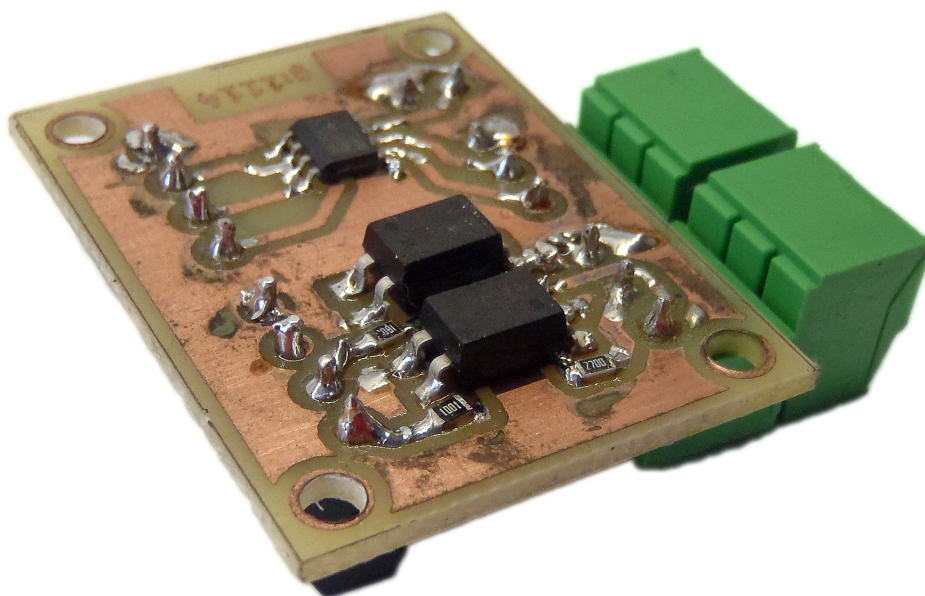
Obrázek 4: DPS relé desky

Napájecí napětí desky je 5 V, na vstupu je připojená ochranná dioda, filtrační kondenzátor a rezistor v sérii s indikační LED. Programování mikroprocesoru je možné

pomocí ISP programátoru (např. BiProg) přes standardní 6 pinový konektor. Komunikace s mikroprocesorem je možná přes budič RS-485, případně přímo pomocí UART rozhraní. Výstupní relé jsou spínána pomocí tranzistorů, paralelně ke spínacím cívkám jsou připojené ochranné diody. Stav každého výstupu je indikován samostatnou LED.

3.3 Deska s převodníkem

Mezi vývojovou desku OLinuXina a desku s relé je zařazena deska s převodníkem. Obvod Texas Instruments SN65176B zajišťuje převod úrovní signálu UART na RS-485 a zpět. Pro testovací účely deska obsahuje také dva optočlony pro oddělení RX a TX vodičů sériové sběrnice z důvodu použití rozdílných napěťových úrovní na vývojové desce. Napájecí napětí je 5 V, proto musí být převodník zapojen přes optočlony.



Obrázek 5: DPS desky s převodníkem

3.4 UART

UART (Universal Asynchronous Receiver and Transmitter) je protokol umožňující komunikaci mezi dvěma body. Komunikace probíhá duplexně a asynchronně – přijímací strana synchronizuje přijatý paket podle jeho celkové délky. Proto se na začátek a konec paketu přidává jeden bit navíc, tzv. start bit a stop bit. Paket obvykle obsahuje 5 až 8 datových bitů a volitelně také paritní bit pro kontrolu chyb. Výsledný paket může být dlouhý 7 až 11 bitů. Přenosová rychlost konzole v operačním systému Linux je standardně nastavena na 115200 Bd. [10]

3.5 RS-485

RS-485 je specifikace fyzické vrstvy sériové sběrnice. Poloduplexní přenos probíhá po dvou kroucených vodičích, které přenášejí symetrický signál, volitelně může být připojen zemní vodič. Na jedné sběrnici může být připojeno až 32 zařízení, maximální délka sběrnice je 1200 m (rychlost přenosu 100 kbit/s), při délce do 10 m je maximální přenosová rychlost až 10 Mbit/s. [11]

3.6 Budič symetrické sběrnice SN65176B

Obvod SN65176B od Texas Instruments funguje jako vysílač a přijímač na symetrické sběrnici. Obvod je napájen 5 V, na sběrnicevých vodičích vytváří symetrický signál. V jednom okamžiku může buď vysílat nebo přijímat, datový provoz je tedy poloduplexní. Aktuální režim se ovládá pomocí vstupů /RE a DE. Obvod umožňuje komunikovat rychlostí až 10 Mb/s a obsahuje ochrany proti přehřátí a nadproudům na sběrnici. [12]

4 Software

V první části této kapitoly je stručně popsána práce s operačním systémem Linux. V další části je rozebrán programovací jazyk Python spolu s krátkými ukázkami zdrojových kódů. V další části jsou zmíněny knihovny použité v rámci programování aplikace. Konec kapitoly se zabývá komunikačním protokolem použitým při komunikaci mezi vývojovou deskou a periferií.

4.1 Operační systém

Výrobce k vývojové desce poskytuje obraz distribuce Android, který je standardně nahrán ve FLASH paměti, a linuxovou distribuci Debian. K dispozici jsou také skripty pro kompilaci vlastního linuxového jádra. [5]

Jako nejvhodnější se jevila distribuce Arch Linux ARM. Jde o odlehčený a minimalistický systém, vhodný pro použití v embedded zařízeních. Používá vlastní balíčkovací systém Pacman, k dispozici je také další uživatelský repozitář AUR. [13] Obraz distribuce je k dispozici na on-line fóru Olimexu. [14] Instalace spočívá v rozbalení obrazu z archivu a nakopírování na MicroSD kartu, například pomocí příkazu `dd` v Linuxu, ve Windows pomocí programu Win32 Disk Imager. Po vložení MicroSD karty do příslušného slotu a připojení napájení systém nastartuje. Interakce probíhá pomocí standardního unixového shellu nebo vzdáleně přes SSH. Po nakonfigurování systému byl celý systém překopírován na interní FLASH paměť. Nejprve jsou ve FLASH paměti vytvořeny tři oddíly, po restartu následuje jejich formátování a kopírování bootladeru a kořenového systému. Celý postup je popsán na internetovém fóru Olimexu. [15]

Pokud chceme na FLASH paměť nahrát původní OS Android, lze tak učinit pomocí aplikace *PhoenixSuit* přes USB-OTG. Program i obraz Androidu lze stáhnout na Wiki Olimexu. [7, 16] Od vývojové desky nejprve odpojíme napájení, následně podržíme tlačítko RECOVERY a napájení opět připojíme. Následně uvolníme tlačítko RECOVERY a vývojovou desku připojíme k PC. Operační systém se zeptá na instalaci ovladačů, nasměrujeme ho do složky s nainstalovaným programem *PhoenixSuit*. Následuje volba obrazu k zapsání do paměti. Po potvrzení se obraz zapíše a o úspěšném dokončení nás program informuje.

Nastavení OS

V této části se budeme věnovat nastavení linuxové distribuce Arch Linux ARM, nastavení jiných distribucí se může lišit. Postup instalace na SD kartu lze nalézt na webu. [17]

Po vložení SD karty a připojení napájecího zdroje by mělo OLinuXino nabootovat. Celý postup bootování můžeme pozorovat na debugovací konzoli, která je ve výchozím nastavení připojená na UART0 (115200 Bd, 8-N-1). Po úspěšném startu se můžeme do systému přihlásit přes debugovací terminál nebo přes výchozí terminál, po připojení klávesnice a monitoru, a nebo vzdáleně přes SSH pomocí přihlašovacích údajů jméno: „root“, heslo: „root“ (oboje bez uvozovek). V našem případě nefungoval po prvním spuštění Ethernet, proto jsme vytvořili soubor `/etc/interfaces/eth0` a přidali do něho nastavení:

```
Interface=eth0
Connection=ethernet
IP=dhcp
```

Zapnuli jsme síťové rozhraní

```
$ netctl start eth0
```

Po pár sekundách byla IP adresa korektně přidělena DHCP serverem. Nastavení síťové adresy můžeme ověřit příkazem

```
# ip addr show
```

Aby se profil síťového rozhraní načítal po každém startu spustili jsme příkaz

```
$ netctl enable eth0
```

Jako další jsme provedli aktualizaci celého systému

```
$ pacman -Syu
```

a instalaci balíčků Pythonu

```
$ pacman -S python2 python2-pip
```

Framework Flask a knihovnu pro ovládání GPIO pinů nainstalujeme pomocí programu pip

```
$ pip install Flask
```

```
$ pip install pyA20
```

4.2 Webový server

Webový server je zodpovědný za přijímání HTTP požadavků od klientů a jejich zpracování. To zahrnuje odpovídání klientům a odesílání dat v různých formátech. V projektu byl vyzkoušen `lighttpd` a `Apache`, nakonec byl použit server integrovaný v Python frameworku `Flask`.

V případě obecného webového serveru jde komunikovat se systémem pomocí spouštění CGI skriptů. Server v tomto případě vystupuje jako prostředník mezi klientem a další aplikací spuštěnou na serveru. Toto řešení funguje, ale v našem případě se ukázalo jako zbytečně složitě a robustní. [18]

4.3 Python

Python je interpretovaný objektový programovací jazyk, který klade důraz na jednoduchou a čitelnou syntaxi. Interpretery jsou k dispozici pro většinu rozšířených platforem, kód jde taktéž zkompilovat a pouštět bez potřeby instalace interpreteru. [19]

4.3.1 Syntaxe

Největší odlišností od „běžných“ programovacích jazyků je formátování kódu pomocí odsazení:

```
for i in range(1,10)
    print i
    print " Hello World!\n"
print "End"
```

V kódu nejsou použity složené závorky, jako například v jazyce C/C++. Kompilátor se orientuje podle počtu mezer a tabulátorů na řádku před každým výrazem. Počet mezer (tabulátorů) může být libovolný, ale stejný pro každý řádek v rámci jedné sekce. V opačném případě nebude program fungovat. Princip členění kódu pomocí odsazování vede k jeho lepší čitelnosti a přehlednosti, ale je potřeba si na něj zvyknout. [19]

Syntaxe se také mírně liší pro verze 2.x a 3.x, podrobnější vysvětlení lze nalézt na Wiki Pythonu [19]

Přehlednosti kódu napomáhá také objektovost jazyka, například práce se souborem vypadá takto:

```
file = open("text.txt", "rw") # načtení souboru do objektu
file.write("Hello World!")   # zapsání textového řetězce
print( file.read(20) )       # vypsání dvaceti znaků souboru
file.close()                 # zavření souboru
```

4.3.2 Python framework Flask

Flask je minimalistický framework určený k tvorbě webových aplikací. Implementuje v sobě webový server a debugger. Podporuje šablony a k dispozici jsou různá rozšíření. Následující kód v Pythonu ilustruje vytvoření webové stránky s nápisem „Hello World!": [20]

```
from flask import Flask app = Flask(__name__)

@app.route("/") def hello():
    return "Hello World!"

if __name__ == "__main__":
    app.run()
```

4.3.3 Práce se sériovým rozhraním

K ovládání sériového rozhraní je použita knihovna *pySerial*. Po instalaci (například pomocí programu pip) se do programu importuje příkazem

```
import serial
```

Knihovna umožňuje jednoduše nastavit přenosovou rychlost, timeout a další parametry. Čtení a zápis probíhá pomocí metod

```
object.read(data)
object.write(size)
```

Jako parametr u čtení lze vložit počet přečtených bajtů. [21]

4.3.4 Práce s GPIO piny

K ovládání GPIO pinů je k dispozici knihovna *pyA20*. Importuje se příkazem

```
import pyA20
```

a obsahuje metody pro nastavování jednotlivých pinů a čtení jejich stavů. Názvy portů lze vkládat celé např. *PH22* a nebo pomocí názvu portu a čísla pinu např. *gpio1p10*, *uext2p9* atd. Knihovna také obsahuje metody pro práci s I2C a SPI sběrnici. [22]

4.3.5 Práce s UDP sokety

V rámci projektu byla vyzkoušena i možnost komunikace po Ethernetu pomocí protokolu UDP. V Pythonu je k dispozici knihovna *socket*, pomocí které lze vysílat i přijímat UDP pakety. Příklady implementace lze najít na Wiki Pythonu v článku o UDP komunikaci [19].

Pomocí UDP paketů je možné v případě potřeby ovládat připojenou periférii. V tom případě by nebylo potřeba použít webové rozhraní, ale například jednoduchou knihovnu nebo aplikaci pro odesílání paketů na straně klienta.

4.4 Webové vývojové prostředí

Pro zjednodušení práce byl na vývojovou desku nainstalováno webové vývojové prostředí (IDE) *Codiad*. Výhoda tohoto typu vývoje spočívá v přístupnosti zdrojových kódů a možnosti programování ze skoro libovolného zařízení s webovým prohlížečem, bez nutnosti instalace dalšího podpůrného softwaru. K provozu vývojového prostředí je potřebný webový server (například Apache) a PHP ve verzi 5 nebo vyšší. Instalace se provede pouhým stažením a rozbalením do složky přístupné z webového serveru. [23]

K vývojovému prostředí existuje spousta doplňků, pomocí kterých ho můžeme rozšířit o další funkce. Jako příklad uveďme rozšíření *Terminal*, které nám umožní skripty naprogramované v Pythonu spouštět přímo z webového prostředí. [23]

4.5 Protokol pro komunikaci po sériové sběrnici

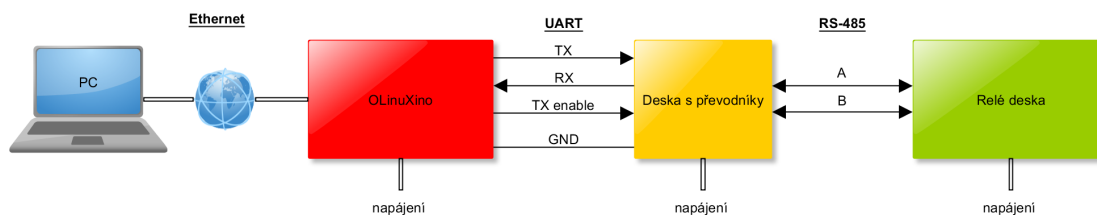
Pro účely projektu byl použit jednoduchý komunikační protokol. Jeden komunikační paket má délku tři bajty (znaky). První bajt slouží jako uvozovací a je pokaždé stejný. Druhý představuje adresu cílového zařízení. První čtyři bity z posledního bajtu přenášejí typ příkazu, který se má vykonat a ve druhých čtyřech bitech se přenáší případná data. Struktura paketu je názorně vidět v tabulce 2.

Uvozovací znak [8 bitů]	Adresa [8 bitů]	Příkaz [4 bity]	Data [4 bity]
0x3E	0x1A	0x2	0x4

Tabulka 2: Struktura komunikačního paketu

Kvůli ověření bezchybného přenosu odesílá SLAVE zařízení přijatý paket zpět k MASTER zařízení. V paketu případně změní pouze poslední čtyři bity které představují data.

5 Sestava s připojenou periférií



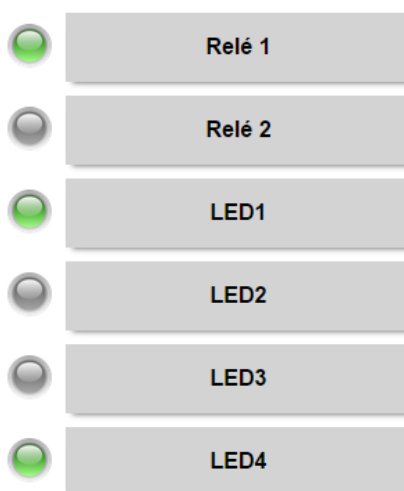
Obrázek 6: Schéma připojení relé desky

K vývojové desce je připojena deska s převodníkem přes sběrnici UART. Protože je na vývojové desce použito napájecí napětí 3,3 V, je na desce s převodníkem datový signál nejprve oddělen dvojicí optočlenů a následně přiveden do budiče symetrické sběrnice RS-485. Na desku je taktéž přiveden ovládací signál TX_enable pro ovládání stavu budiče a signálová zem. Mezi deskou s převodníky a relé deskou je signál veden po dvou signálových vodičích A a B. Protože je signál symetrický, není potřeba propojit signálové země obou desek, ale je to možné. Relé deska obsahuje další budič, který převádí symetrický signál na UART sběrnici pro mikroprocesor Atmel AVR. V případě potřeby lze vynechat budiče a propojit přímo UART sběrnici za optočleny s UART sběrnici u mikroprocesoru. Vývojová deska je připojena do počítačové sítě a jde ovládat z internetového prohlížeče pomocí webového rozhraní.

Na testovací desce je spuštěn webový server s ovládací aplikací. Po připojení z webového prohlížeče je vidět aktuální stav všech výstupů na relé desce. Klikáním na příslušná tlačítka lze stavy výstupů libovolně měnit.

Celé jádro aplikace včetně webového serveru je vyvinuto v Pythonu. Po kliknutí na tlačítko změny výstupu ve webovém rozhraní se vytvoří paket a odešle přes UART sběrnici z vývojové desky do převodníku, který zajistí jeho převedení na symetrický signál RS-485. Po sběrnici signál putuje do dalšího převodníku na relé desce, kde je převeden zpět na UART, a dále do mikroprocesoru AVR. Mikroprocesor přijatý paket zpracuje, zajistí příslušné nastavení výstupů a odešle odpověď s aktuálním stavem výstupů zpět do vývojové desky. Program na vývojové desce přijme odpověď a zajistí zobrazení aktuálního stavu na webové stránce.

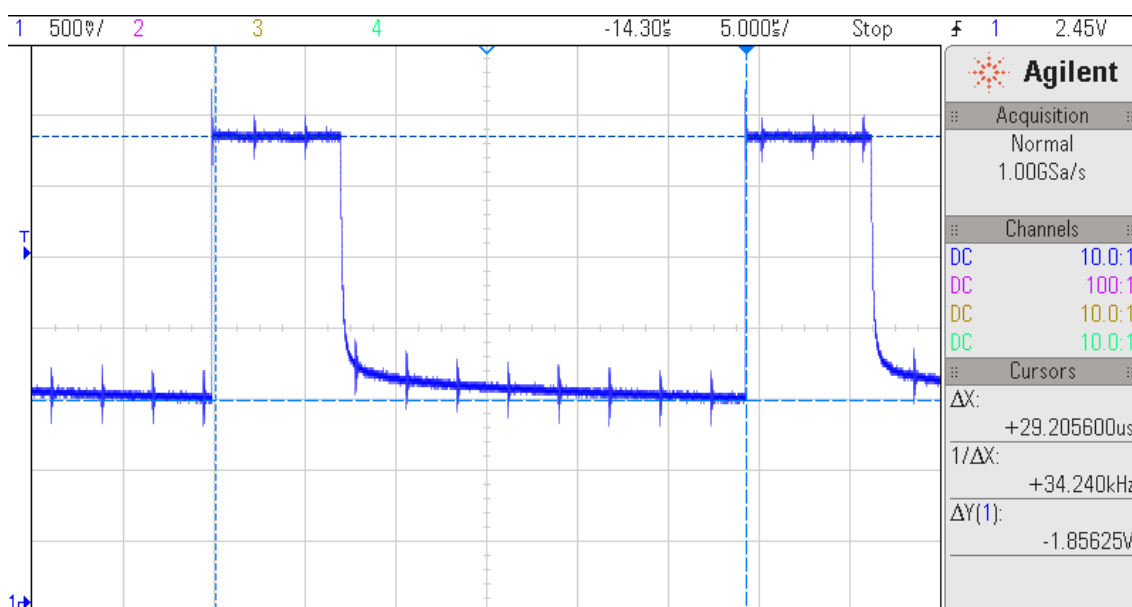
GPIO control



Obrázek 7: Náhled webového ovládacího rozhraní

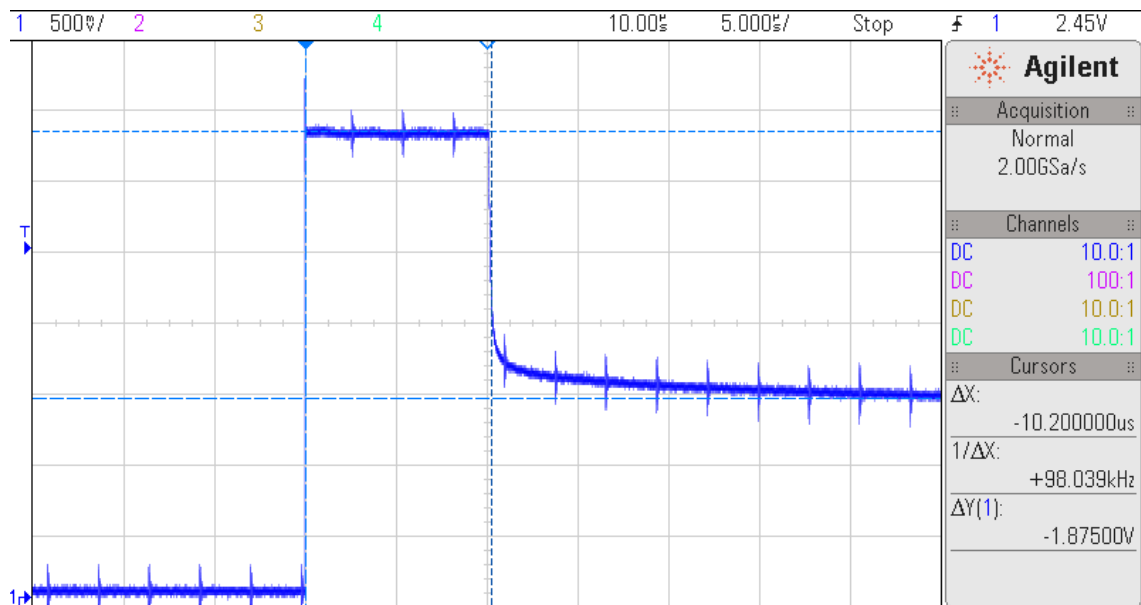
5.1 Měření odezvy GPIO pinu

V prvním testu byl GPIO pin nastavován v nekonečném cyklu střídavě do logické nuly a jedničky. Průběh napětí na pinu byl sledován osciloskopem. Z oscilogramu 8 je patrné, že perioda přepnutí je asi $29,2 \mu\text{s}$, což odpovídá frekvenci $34,2 \text{ kHz}$.



Obrázek 8: Měření času přepnutí GPIO pinu, opakovaný děj

V druhém testu byl GPIO pin trvale nastaven do logické nuly. Spuštěním programu se nastavil do logické jedničky a ihned zpět do logické nuly. Celý děj byl opět sledován osciloskopem, důležitým parametrem je doba trvání logické jedničky. Z oscilogramu 9 lze odečíst, že půlka periody přepnutí trvá $10,2 \mu\text{s}$. Perioda potom trvá $20,4 \mu\text{s}$, čemuž odpovídá frekvence přepínání asi 49 kHz .



Obrázek 9: Měření času přepnutí GPIO pinu, jednotlivý děj

V průběhu testování stouplо vytížení jednoho jádra procesoru na 100 %. Pravidelnost spínání se v čase nepatrně změnila, stejně tak perioda. Frekvence spínání 34,2 kHz se jeví jako poměrně nízká. V případě potřeby rychlejší odezvy by bylo možné použít real-time jádro, které zrychlí odezvy systému.

Jako další možnost se jeví vynechání ovladače napsaného v Pythonu a ovládání portu přímo z operačního systému. Více informací lze nalézt na internetu, například v dokumentu <http://www.py6zgp.com/download/A20-GPIO.pdf>.

6 Přehled existujících řešení

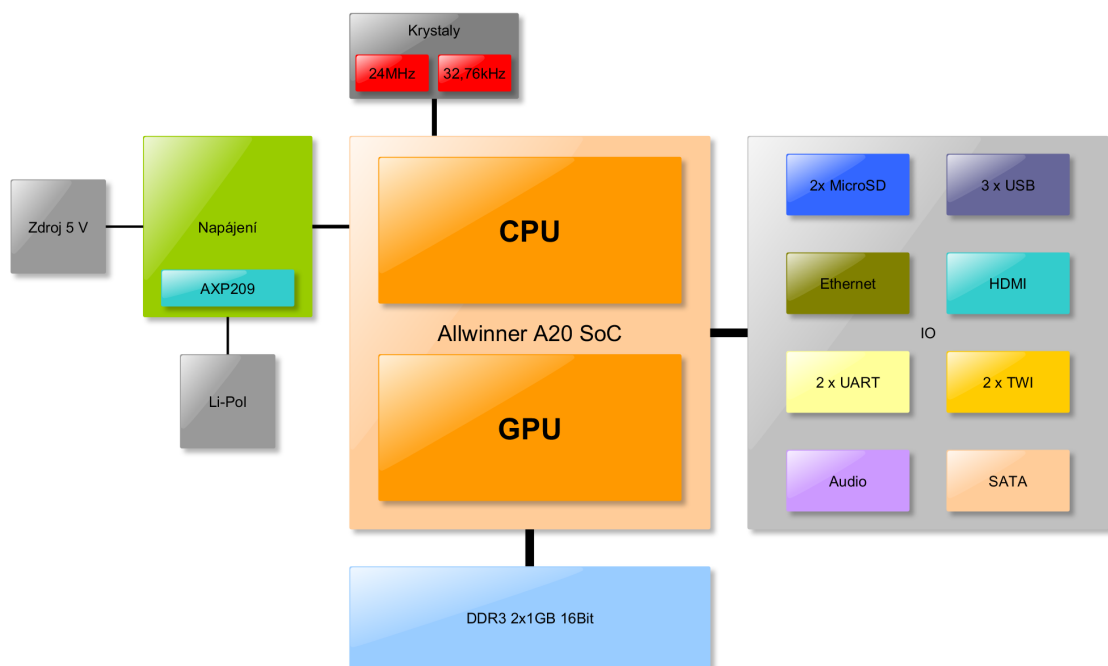
Kromě již zmiňované desky od Olimexu se se SoC A20 vyrábí také další. V tabulce 3 je uveden stručný přehled některých z nich. Další lze nalézt na Wiki projektu *sunxi*. [24, 25]

Název	Podklady k dispozici	SoC
A20-OLinuXino-MICRO	Schéma, DPS	Allwinner A20
Banana Pi R1	Schéma	Allwinner A20
Cubieboard 1-3	Schéma	Allwinner A20
LinkSprite pcDuino	Schéma	Allwinner A20
MarsBoard A20	Schéma	Allwinner A20
Orange Pi A20	Schéma	Allwinner A20
expeyes-sbc	Schéma, DPS	Allwinner A10

Tabulka 3: Přehled existujících embedded systémů

Existujících řešení se SoC A20 je ještě více, k většině je však k dispozici pouze schéma zapojení v PDF. Výjimku tvoří vývojová deska od Olimexu, k níž jsou k dispozici i zdrojové soubory ve formátu Eagle. Další deskou se zdrojovými kódy ve formátu KiCad je *expeyes-sbc*, která je ale osazená SoC A10. [26, 27]

Na základě zapojení výše uvedených vývojových desek byly vytipovány komponenty a hrubé rysy zapojení vlastního embedded systému. Blokové schéma navrženého zapojení je na obrázku 10.

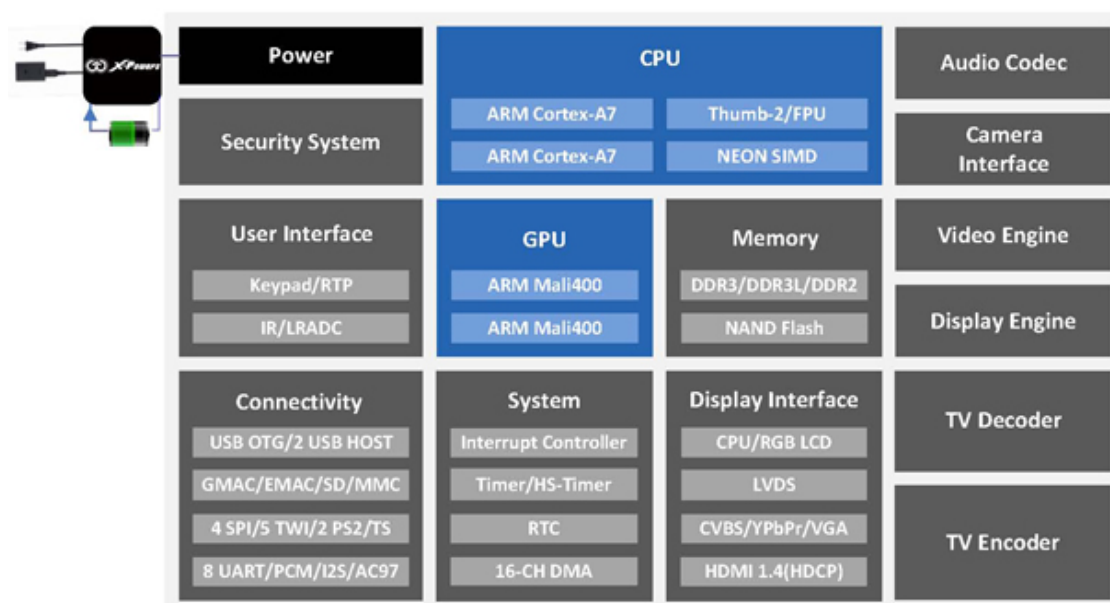


Obrázek 10: Blokové schéma navrženého embedded systému

7 Použité komponenty

SoC Allwinner A20

SoC Allwinner A20 je dvoujádrový ARM Cortex-A7 procesor navržený pro použití v mobilních aplikacích, jakými jsou například tablety nebo chytré televizory. Architektura Cortex-A7 byla poprvé představena v roce 2011. Procesor začleňuje také ARM Mali400 MP2 GPU, která umožňuje plynulé dekódování 2160p videa a také kódování 1080p videa z kamery. A20 obsahuje mnoho dalších periférií, které budou zmíněny dále v textu, a je vývodově kompatibilní se svým předchůdcem, procesorem A10. Dodává se v pouzdře FBGA 441. [28]



Obrázek 11: Blokové schéma SoC A20 [29]

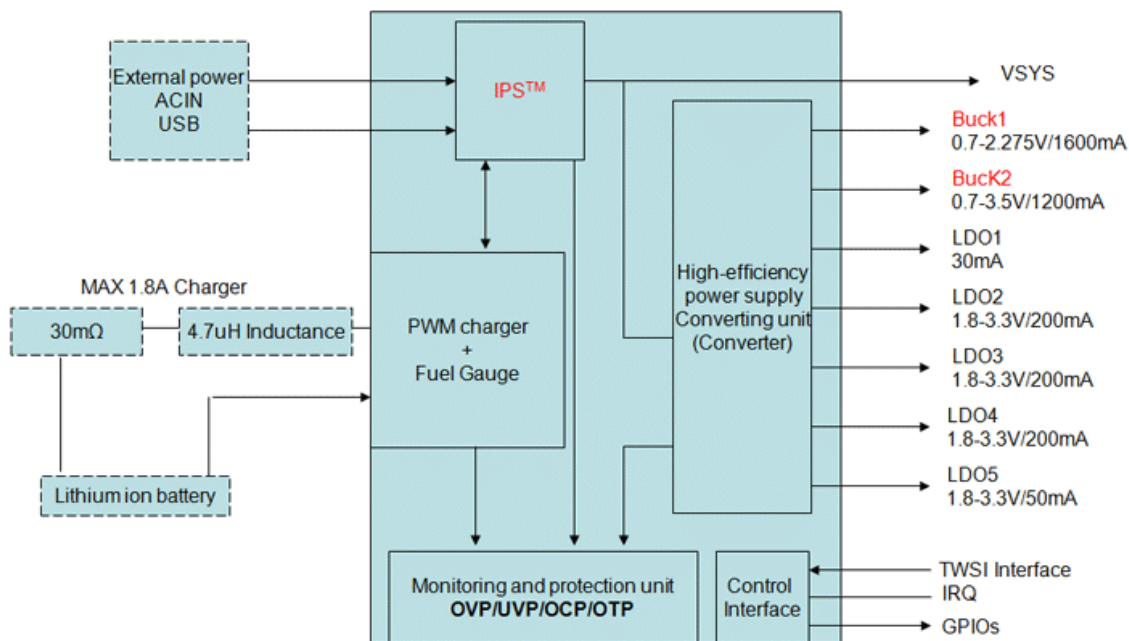
Operační paměť

V zapojení jsou použity DDR3 paměti v pouzdře FBGA 96. Paměťové čipy mají šířku datové sběrnice 16 bitů, proto jsou v zapojení použity dva. Adresní sběrnice má šířku 15 bitů; protože musí být propojena s oběma paměťovými čipy, bylo použito zapojení v topologii T. Všechny datové, adresní, hodinové a řídicí vodiče mají řízenou impedanci a musí být délkově přizpůsobeny. [36, 37]

Napájecí napětí je 1,5 V. Referenční napětí 0,75 V je tvořeno napěťovým děličem.

AXP209

AXP209 je mikrokontrolér pro správu napájení, zkráceně PMU (Power Management Unit). Obvod je vyráběn společností X-Powers, sesterskou společností Allwinneru. Byl



Obrázek 12: Blokové schéma AXP209 [31]

primárně vyvinut pro napájení SoC z řad A10, A13 a A20. Obvod umožňuje napájení systému z jednočládkové Li-Ion nebo Li-Po baterie a obsahuje několik DC-DC měničů. PMU umí plynule přecházet mezi zdroji napájení v závislosti na přítomnosti napájecích napětí. Pro nabíjení baterie je vyhrazen jeden z PWM měničů s výstupním proudem až 1,8 A.

Nabíjení je plně automatické, s volbou výsledného napětí baterie po nabití, a s indikací nabíjení LED. PMU také omezuje nabíjecí proud z hlediska zatížení ostatních větví tak, aby nedocházelo k přetížení napájecího zdroje. Podporováno je také připojení další záložní baterie pro obvod reálného času s volitelným nabíjecím proudem. PMU dále obsahuje dvojici spínaných zdrojů s rozsahem napájecích napětí 0,7 V až 2,275 V nebo 3,5 V a proudovou zatížitelností 1,6 A a 1,2 A. Obvod vyžaduje napájecí napětí 2,9 V - 6,3 V. Dále je přítomna pětice nízkoúbytkových nastavitelných regulátorů, z nichž tři jsou také nízkošumové. Proudová zatížitelnost je různá a pohybuje se v rozsahu 30 mA až 200 mA. Pro měření napětí a proudů je určen 12 bitový ADC převodník. Obvod dokáže komunikovat přes I2C sběrnici a předávat informace o své vnitřní teplotě, odběru v mA nebo mW, zbývající kapacitě baterie v % nebo mAh a také o zbývajícím odhadovaném času do úplného nabití. PMU taktéž obsahuje řadu ochran: proti zkratu, přehřátí, přepětí i podpětí. Mikrokontrolér obsahuje i čtyři programovatelné GPIO piny; dodává se ve 48 pinovém QFN pouzdře. Podrobnější informace lze nalézt v katalogovém listu. [30, 31]

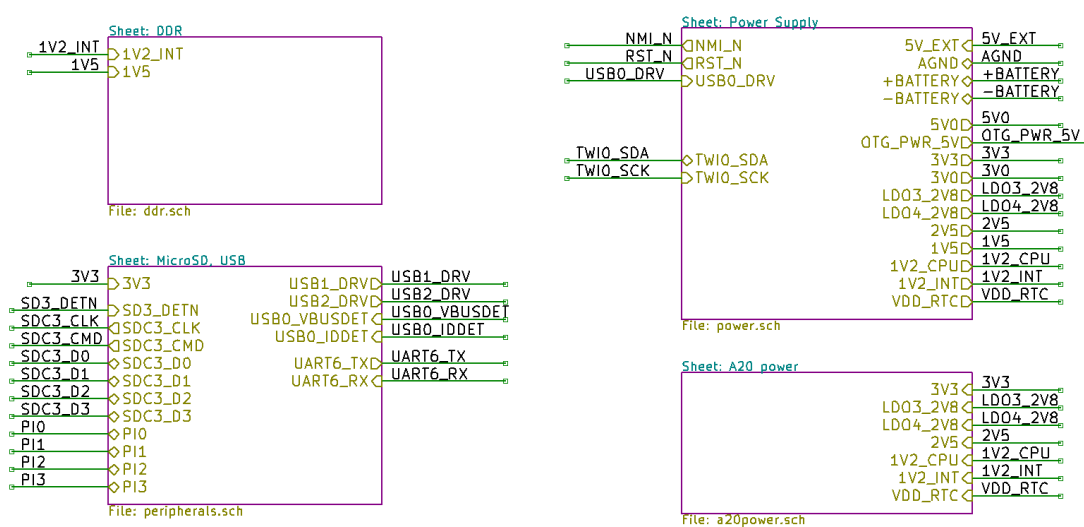
8 KiCad

Celým názvem *KiCad EDA Software Suite* je programové prostředí pro vývoj elektronických systémů licencované pod svobodnou licenci GNU GPL v2.0. Zdrojové kódy programu jsou volně k dispozici na stránkách projektu. Odtud lze také stáhnout nejnovější i stabilní verzi programu, případně si můžeme zkompileovat vlastní verzi. Vývoj programu začal v roce 2007, na projekt aktuálně finančně přispívá *Evropská organizace pro jaderný výzkum* (CERN). Prostedí se skládá z několika programů pro správu projektu, kreslení schémat, práci s knihovnami součástek, tvorbu plošných spojů a prohlížení souborů ve formátu Gerber. V posledních verzích se v editoru plošných spojů objevili i pokročilé nástroje pro routování diferenciálních párů, korekce délek vodičů a „push and shove“ routování, i když zatím jen v experimentálním režimu. Pro použití těchto funkcí je nutné program přepnout do režimu vykreslování v OpenGL (výchozí klávesová zkratka je F11). [32]

Po spuštění programu se ocitneme v okně správce projektů. V levé části vidíme všechny soubory, které jsou v projektu obsaženy a se kterými můžeme pracovat. Z vrchní lišty můžeme spustit jednotlivé nástroje, které software obsahuje. V následující části popíšeme jednotlivé programy KiCadu a uvedeme nejmarkantnější rozdíly v porovnání s programem Eagle.

8.1 Eeschema

Eeschema je program pro kreslení a editaci schémat. Ze zajímavých funkcí uvedeme nástroj pro automatickou anotaci součástek a práci s hierarchií jednotlivých listů schématu. Listy lze libovolně vnořovat, v nadřazené úrovni jsou reprezentovány jako blok s vstupními a výstupními porty, které můžeme určit. Další užitečnou funkcí je kontrola chyb v zapojení (ERC). U každého portu součástky lze definovat jeho typ (vstupní, výstupní, napájecí atd.), následně můžeme v matici nastavit, které kombinace spojení portů jsou žádoucí a které budou při kontrole generovat chyby.



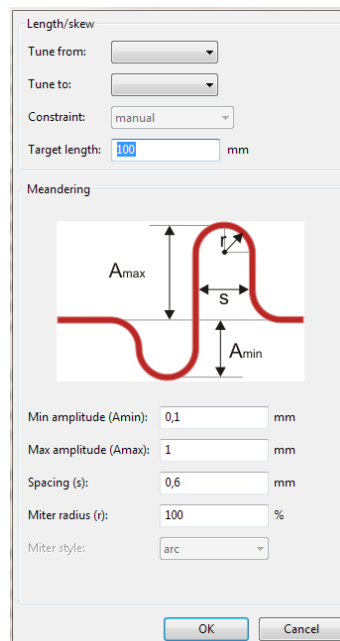
Obrázek 13: Vnořené listy ve schématu

8.2 Pcbnew

slouží k návrhu desek plošných spojů. Umožňuje práci až s 16 vrstevnými DPS a nechybí export dat do různých formátů včetně Specctra DSN. Výrobní data lze generovat ve formátu Gerber RS-274X. V posledních verzích byla přidána i skriptovací konzole PyCrust umožňující automatizaci procesů pomocí skriptů napsaných v programovacím jazyce Python.

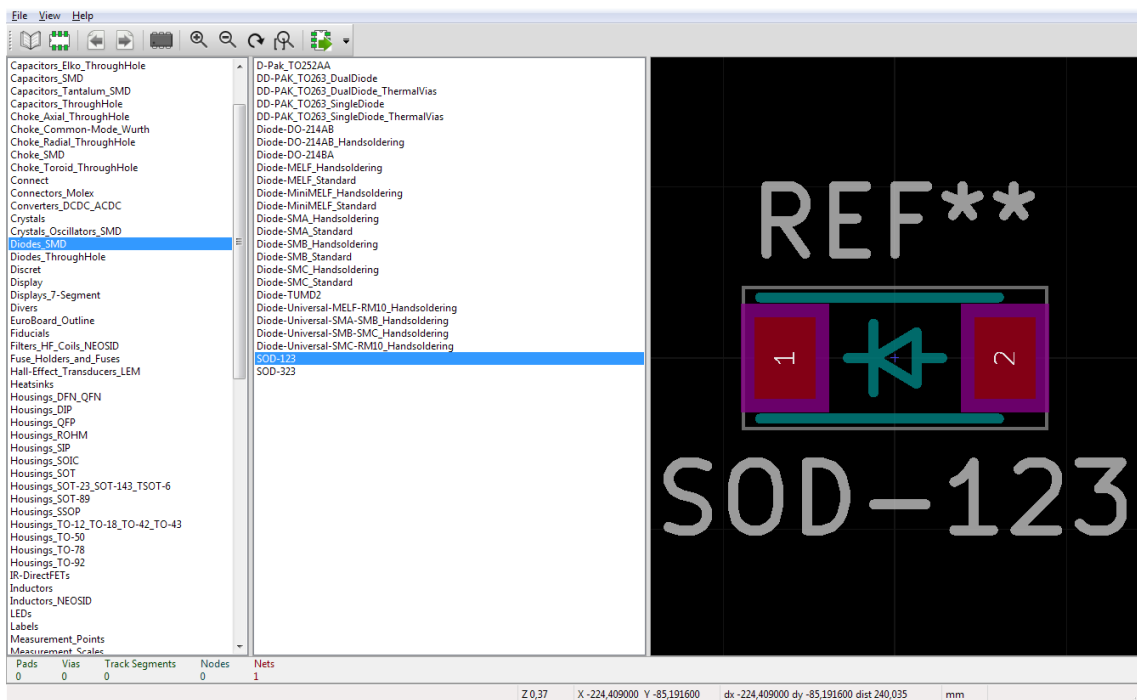
Užitečnou funkcí, která je ale ještě ve vývojovém režimu, je interaktivní routování. Pokud tento režim aktivujeme, při táhnutí nové cesty nám spoje a prokvy podle možností „odskakují“ a odpadá tím nutnost ručního posouvání cest po jedné. Router v tomto módu zachovává nastavené izolační vzdálenosti, můžeme také nastavit usilovnost (effort) posouvání nebo obcházení překážek. Na výběr jsou také dva další módy - zvýrazňování kolizí a obcházení překážek. V programu je přítomen i modul automatického routování.

Další používanou funkcí byla úprava délek vodičů. Nejdůležitějším parametrem je délka, na kterou budou vodiče prodlužovány. Klávesami „1“ a „2“ můžeme zmenšovat a zvětšovat vzdálenost mezi smyčkami, klávesami „3“ a „4“ můžeme zmenšit nebo zvětšit jejich amplitudu. Dalším podobným nástrojem můžeme srovnat délku vodičů diferenciálního páru. K dispozici je i nástroj pro úpravu délky diferenciálních párů, který ale v aktuální verzi nefungoval.



Obrázek 14: Nástroj pro úpravu délek vodičů

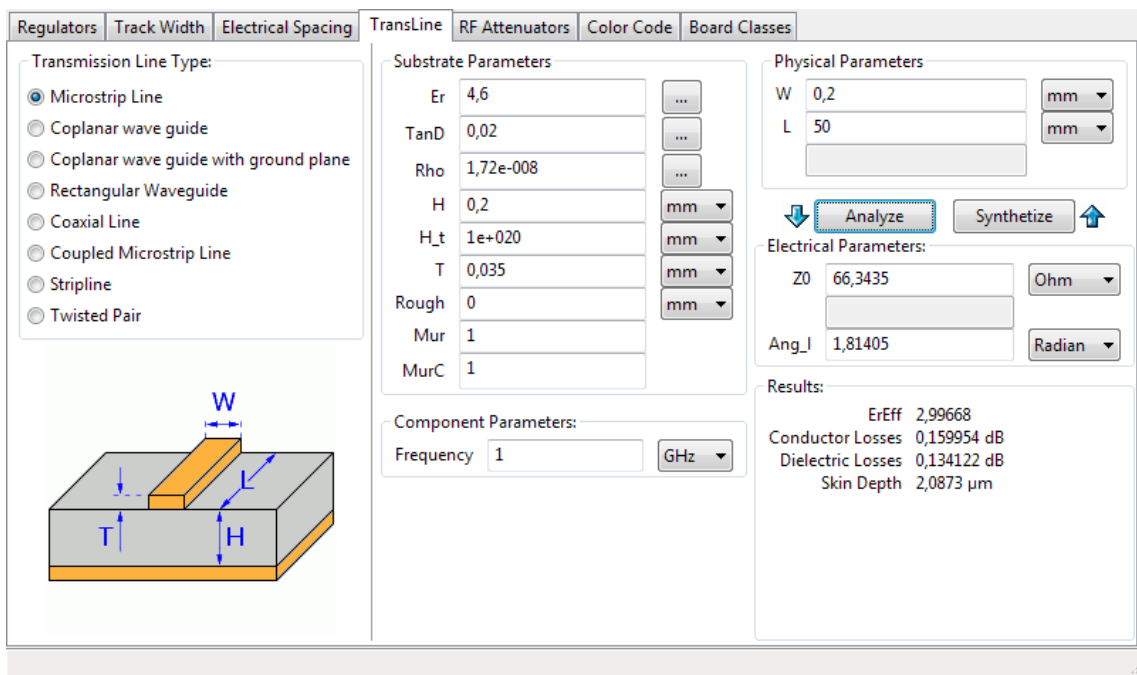
S výhodou jsme také použili rozsáhlé možnosti nastavení zobrazení, například skrytí ploch vylitých mědí a tzv. „režim vysokého kontrastu“, kdy jsou všechny aktuálně neaktivní vrstvy podbarveny šedě.



Obrázek 16: Procházení knihovny footprintů

8.4 GerbView, Bitmap2Component, Pcb calculator

součástí balíku jsou i další pomocné nástroje. GerbView umožňuje prohlížení výrobních dat ve formátu Gerber, pomocí Bitmap2Component můžeme převádět bitmapové obrázky na schématické symboly nebo footprinty a přímo je ukládat do knihoven. Pcb calculator obsahuje užitečné nástroje například pro výpočty na mikropáskových vedeních, útlumových člancích aj.



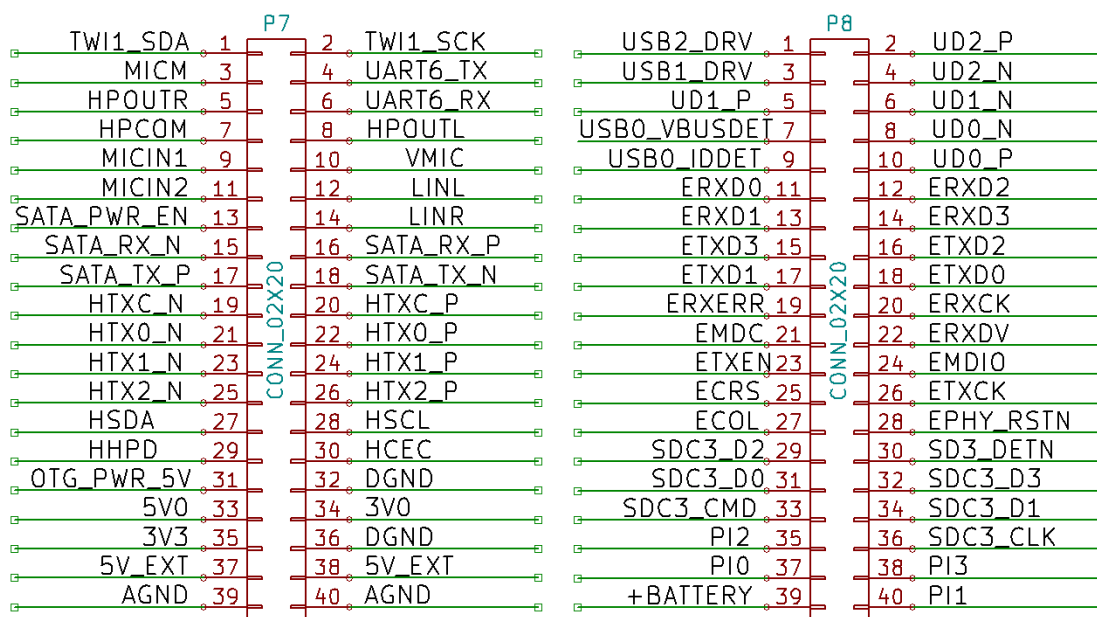
Obrázek 17: Výpočty na mikropáskovém vedení v Pcb calculatoru

9 Návrh embedded systému

9.1 Návrh zapojení

Po vytipování komponentů následovala tvorba příslušných schématických značek. Většinu z nich KiCad již obsahuje, další byly získány z webu a přizpůsobeny. [27, 33] Zapojení vychází z doporučeného zapojení SoC A20 a experimentálních desek A20-OLinuXino-MICRO, Cubieboard1, Cubieboard2 a Cubietruck (Cubieboard3).

Návrh je pro větší přehlednost hierarchický, na základním listu jsou umístěny vstupní a výstupní konektory, listy s podrobnými zapojeními jsou vnořeny. První list obsahuje zapojení operačních pamětí DRAM a napěťový dělič pro získání referenčního napětí. V dalších listech najdeme zapojení MicroSD karty, USB, SATA, HDMI a TWI portů. Vodiče všech těchto portů jsou vyvedené na dvě propojovací pinové lišty v konfiguraci 2 x 20. Pomocné obvody (napájení SATA, přepětové ochrany) budou řešeny na přídatných deskách. Na propojovací lištu jsou připojeny i další signály: Ethernet, UART, Audio a další SD karta. Signály jsou sdruženy do skupin, navíc jsou na lištu vyvedena všechna napájecí napětí. To v případě potřeby umožňuje napájení celého modulu z přídatné desky, například po Ethernetovém kabelu (PoE) po přidání regulátoru na 5 V. Na samostatné lišty je vyvedena i debugovací sériová linka a pin pro připojení IR přijímače.



Obrázek 18: Zapojení spodní (vlevo) a vrchní (vpravo) propojovací lišty

Napájení je postaveno okolo obvodu AXP209, který komunikuje s procesorem po TWI sběrnici. Vstupní napětí 5 V je jištěno transilem, napájecí obvod ho dále snižuje na napětí 1,2 V pro napájení procesoru a 2,8 V a 3 V pro napájení oscilátorů. Další napětí 1,5 V a 3,3 V jsou generována step-down regulátory SY8009A. Požadované napětí je u nich nastaveno poměrem odporů na napěťovém děliči R_a a R_b . Velikost výstupního napětí počítáme podle vztahu

$$V_{OUT} = 0,6 \cdot \left(1 + \frac{R_a}{R_b}\right) \quad [34]$$

Napětí 5 V je generováno zvyšujícím regulátorem SY7208. Výstupní napětí spočítáme podle stejného vztahu jako u regulátoru S8009A. Napětí 2,5 V je určeno pro napájení části procesoru která se stará o rozhraní SATA a je generováno nízkošumovým regulátorem MCP1702. Velikost tohoto výstupního napětí je pevně nastavena.

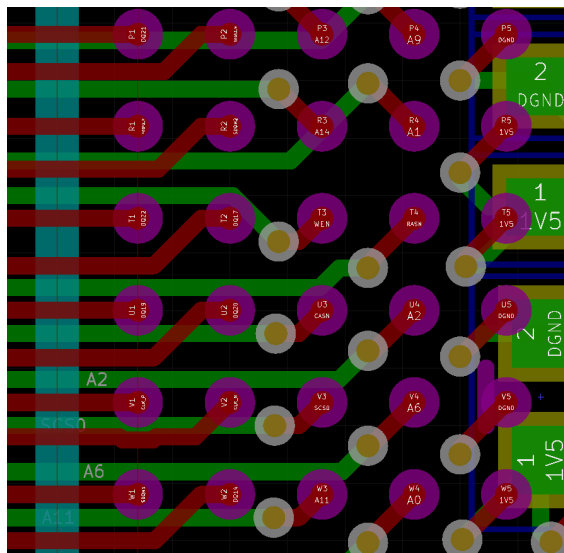
Všechny regulátory jsou napájené z pinů IPSOUT, na které se přenáší aktuální vstupní napětí. Může se jednat o výchozí 5 V, nebo také napětí připojeného akumulátoru. Velikost napětí se pohybuje v rozmezí 2,9 V až 5 V. [35]

Připojené napájecí napětí a nabíjení akumulátoru je indikováno LED. Přítomna jsou též tlačítka pro zapnutí, restart a pro přístup do tzv. recovery režimu. Pro běh procesoru je nezbytný krystal, konkrétně jeden s frekvencí 24 MHz a druhý s frekvencí 32768 Hz pro zajištění běhu obvodu reálných hodin (RTC). Většina použitých součástek je v SMD provedení, výjimku tvoří jen propojovací pinové lišty. Rezistory jsou v pouzdrech 0603, odrušovací kondenzátory jsou keramické, převážně v pouzdrech 0402, kvůli co možná nejmenší parazitní induktanci.

9.2 Návrh DPS

Jako první byl na desku umístěn procesor A20 a paměti DRAM. Následovalo propojení datových a adresních pinů. Propojit všechny adresní vodiče se však napoprvé nepodařilo, proto následovalo několik posunutí pamětí DRAM směrem od sebe a také od procesoru a nové vzájemné propojování. Následovala korekce délky vodičů a routování zbytku desky.

Jako jednodušší se jevílo propojení datových vodičů, jelikož se jedná o nevětvící se spoje. Příslušné piny na procesoru se nachází v prvních dvou řadách, proto ve většině případů nebylo nezbytné s vodiči přecházet do druhé vrstvy. Vodiče se také poměrně snadno daly posouvat do stran, abychom kolem nich vytvořili místo pro následující úpravy délek jiných vodičů.



Obrázek 19: Detail návrhu DPS pod BGA pouzdrem

Routování adresní sběrnice bylo náročnější. Příslušné piny se na pouzdře procesoru nacházejí ve třetí a čtvrté řadě od kraje. Proto bylo potřebné se nejprve pomocí prokovů dostat do spodní vrstvy. Následují prokovy zpět do vrchní vrstvy a rozdělení vodiče k horní a dolní paměti. Ve dvou vnitřních vrstvách je v oblasti mezi procesorem a paměťmi rozlitá

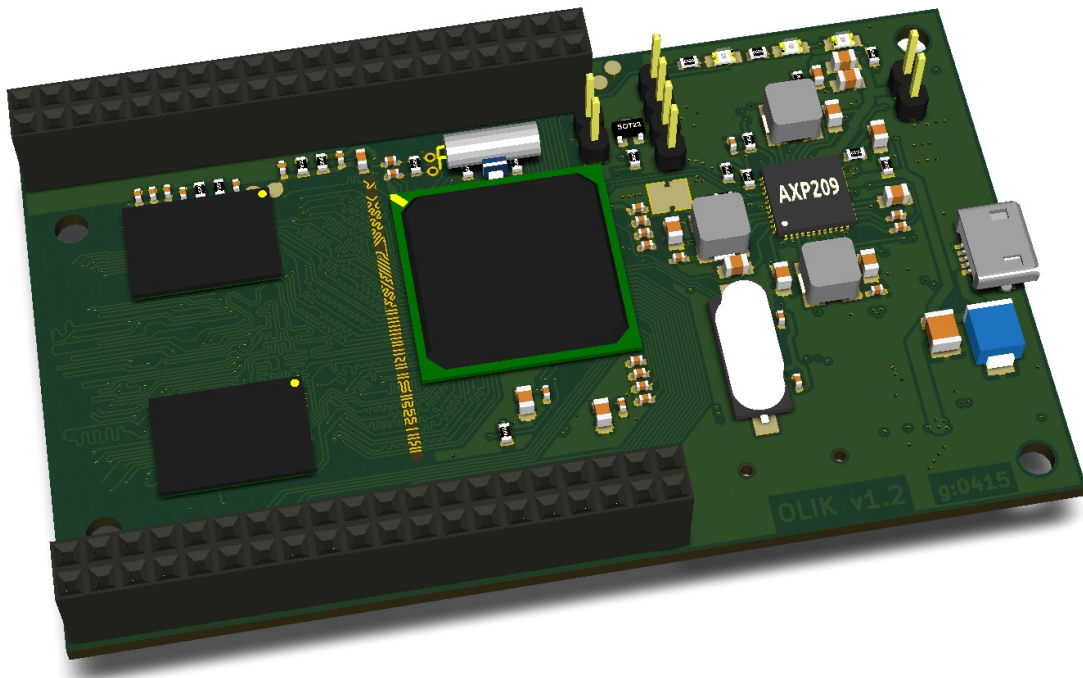
měď tvořící zemní plochu pro mikropásková vedení. Pro horní vrstvu je tato plocha uzemněna, pro spodní vrstvu je na plochu přivedeno napájecí napětí 1,5 V pro operační paměti.

Výzvou bylo i samotné sladění velikosti prokůvů, mezikruží a šířky vodivých cest pod BGA pouzdry vzhledem k možnostem a ceně výroby DPS.

Pozice na propojovacích lištách byly optimalizovány pro co nejjednodušší propojení s procesorem. Země jsou pro digitální a napájecí část oddělené z důvodů minimalizace rušení a spojené pouze v jednom bodě na vrchní straně DPS. Návrhová pravidla jsou uvedena v tabulce 4.

Šířka vodiče / izolační mezery	min. 0,15 mm
Průměr otvoru	min. 0,2 mm
Šířka mezikruží	min. 0,075 mm

Tabulka 4: Návrhová pravidla pro DPS

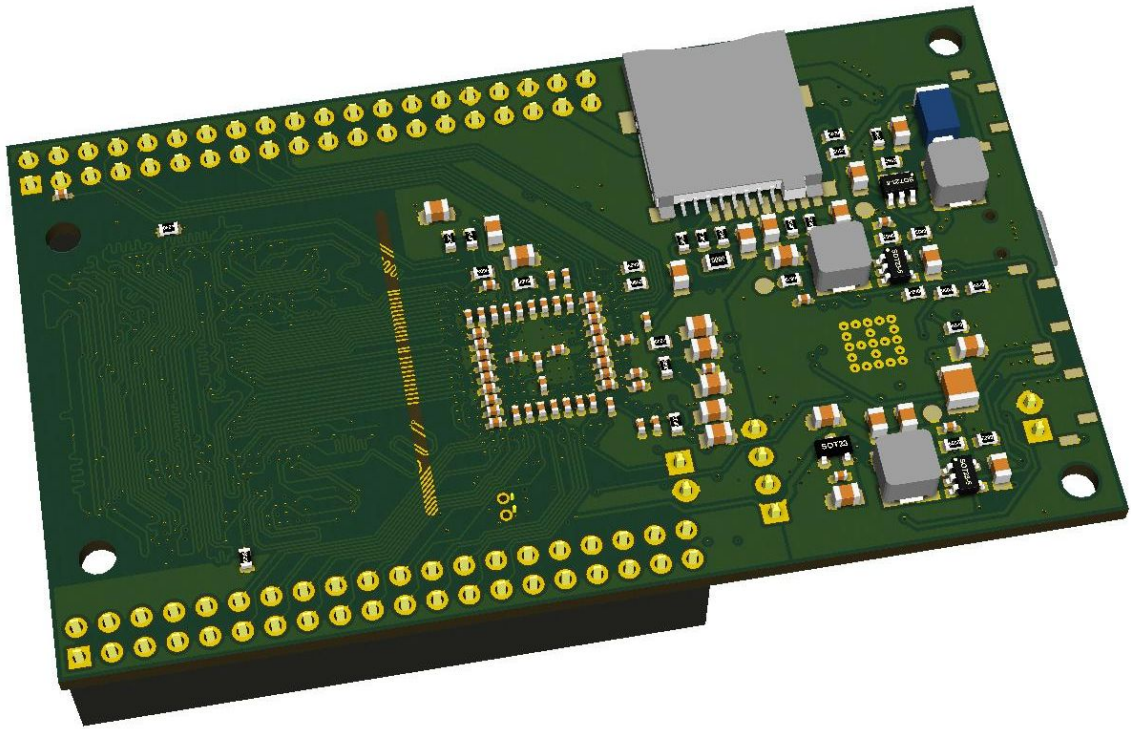


Obrázek 20: Vizualizace navržené DPS, vrchní strana

9.2.1 Vodiče s řízenou impedancí

Adresní a datové vodiče mezi procesorem a operačními paměťmi mají mít, z důvodu co nejmenšího zkreslení signálu, definovanou impedanci. Pro jednoduché signály je doporučena impedance 50Ω , pro diferenciální páry 100Ω pro každý z vodičů, jejich paralelní kombinací tedy dostaneme impedanci 50Ω . Na DPS je tato situace řešena použitím mikropáskového vedení nad rozlitou měď ve vnitřních vrstvách. [36]

Vlastnosti substrátu jsou uvedeny v tabulce 5, hodinová frekvence je 480 MHz. Impedanci mikropáskového vedení jsme vypočítali vestavěným kalkulátorem. Pro impedanci mikropáskového vedení $Z_0=50 \Omega$ na daném substrátu jsme vypočítali jeho šířku



Obrázek 21: Vizualizace navržené DPS, spodní strana

Materiál	FR4
Relativní permitivita	$\epsilon_r = 4,8$
Tloušťka Cu fólie	18 μm
Tloušťka jádra	1,2 mm
Tloušťka prepregu	0,15 mm
Celková tloušťka	1,5 mm
Povrchová úprava	nepájivá maska, HASL

Tabulka 5: Vlastnosti DPS

$w_{t50}=0,25$ mm. Pro diferenciální pár s impedancí jednoho mikropásku $Z_0=100 \Omega$ vyšla jeho šířka jako $w_{t100}=0,045$ mm.

Šířka vodiče 0,045 mm je z výrobního hlediska nerealizovatelná. Jistou možností by byla změna materiálu za jiný s nižší relativní permitivitou, nebo zvýšení tloušťky dielektrika, abychom mohli udělat vodič širší. V takovém případě bychom ale museli neúměrně zvýšit šířku ostatních nediferenciálních vodičů. Například pro materiál s teoretickou relativní permitivitou $\epsilon_r = 1,6$ by byla ideální šířka vodiče diferenciálního páru 0,152 mm. Přitom by ale šířka vodiče s impedancí 50Ω vzrostla na 0,53 mm. Protože těchto vodičů je v zapojení podstatně více (65 oproti 10), výrazně by narostla velikost desky i složitost routování, což je nežádoucí. Z těchto důvodů jsme přistoupili ke kompromisu, a šířku vodičů diferenciálních párů snížili na 0,15 mm, pár jako celek má tedy výslednou charakteristickou impedanci $32,5 \Omega$. Šířku nediferenciálních vodičů jsme snížili také na 0,15 mm, jejich charakteristická impedance je 65Ω .

9.2.2 Korekce délky vodičů

Protože rychlost šíření signálu skrze vodiče je konečná, v případě různých délek vodičů signál cestuje různě dlouhou dobu. V případě paralelního přenosu dat vysokými rychlostmi mohou různé délky vodičů sběrnice v krajním případě zapříčinit i nefunkčnost celého zapojení. V našem projektu byla provedena korekce všech datových a adresních a řídicích spojů mezi procesorem a operačními paměťmi. Po hrubém pospojování na DPS byly změřeny délky vodičů a zapsány do tabulky. V dalším kroku byly rozděleny do skupin, které budou později upravovány na stejnou délku: [36]

- Skupiny 1-4: Datové signály po osmi bitech spolu s příslušnými řídicími signály, například: DQ0 až DQ7, SDQS0_P a SDQS0_N spolu s SDQM0.
- Skupina 5: Adresní, hodinové a řídicí signály: A0 až A14, CLK_P a CLK_N, BA0 až BA2 a další.

Označení schéma	Katalogový list	Skupina	Popis
DQ0 - DQ7	$DQ0 - DQ7$	1	Datová sběrnice, 1. bajt
DQ8 - DQ15	$DQ8 - DQ15$	2	Datová sběrnice, 2. bajt
DQ16 - DQ23	$DQ16 - DQ23$	3	Datová sběrnice, 3. bajt
DQ24 - DQ31	$DQ24 - DQ31$	4	Datová sběrnice, 4. bajt
SDQS0_P, SDQS0_N	$DQSL, \overline{DQSL}$	1	Časovací signál pro data
SDQS1_P, SDQS1_N	$DQSU, \overline{DQSU}$	2	Časovací signál pro data
SDQS2_P, SDQS2_N	$DQSL, \overline{DQSL}$	3	Časovací signál pro data
SDQS3_P, SDQS3_N	$DQSU, \overline{DQSU}$	4	Časovací signál pro data
SDQM0	DML	1	Datový vstup / výstup
SDQM1	DMU	2	Datový vstup / výstup
SDQM2	DML	3	Datový vstup / výstup
SDQM3	DMU	4	Datový vstup / výstup
A0 - A14	$A0 - A14$	5	Adresní sběrnice
BA0 - BA2	$BA0 - BA2$	5	Adresa banky
WEN, CASN, RASN	$\overline{WE}, \overline{CAS}, \overline{RAS}$	5	Prováděný příkaz
SCS0, SCS1	$\overline{CS0}, \overline{CS1}$	5	Chip Select
DDR_ODT	$ODT0$	5	Vnitřní zakončovací Z_0
DDRRES	\overline{RESET}	5	Reset
CLK_P, CLK_N	CK, \overline{CK}	5	Hodinový signál
CKE	\overline{CKE}	5	Clock Enable

Tabulka 6: Přehled DRAM vodičů [37]

Dále byla stanovena optimální délka pro každou skupinu, na kterou se budou kratší vodiče prodlužovat. Pro datové skupiny se tyto délky pohybují od 21 mm do 30 mm, pro adresní skupinu je výsledná délka 55 mm.

Výsledná největší odchylka vodiče od ideální délky je cca 1 mm. Rychlost šíření vlny na vedení lze vypočítat ze vztahu

$$v_p = \frac{c}{\sqrt{\epsilon_{reff}}} = \frac{3 \cdot 10^8}{\sqrt{3,24}} = 1,67 \cdot 10^8 \text{ m/s [38]}$$

Čas šíření získáme převrácením hodnoty

$$t_p = \frac{1}{v_p} = \frac{1}{1,67 \cdot 10^8} = 6 \text{ ns/m} = 6 \text{ ps/mm}$$

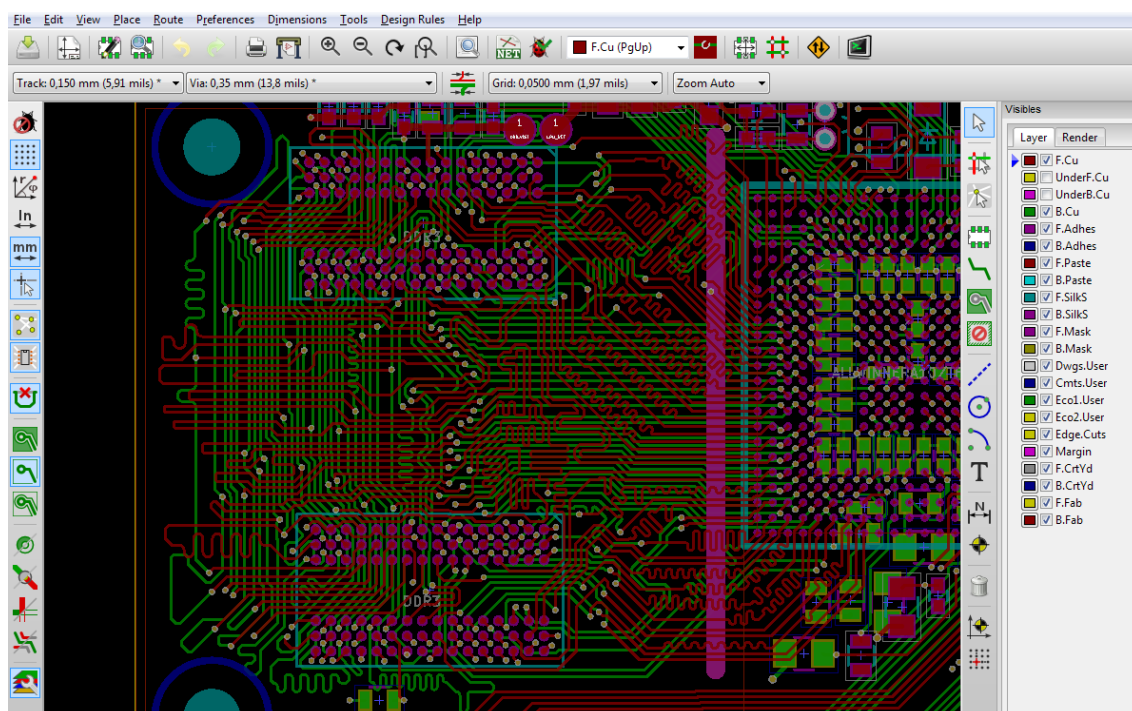
Signál se tedy na o 1 mm delším vodiči zpozdí o 6 ps. Perioda signálu s frekvencí 480 MHz je rovna

$$t_s = \frac{1}{f} = \frac{1}{480 \cdot 10^6} = 2,08 \text{ ns}$$

Zpoždění signálu mezi nejdelším a nejkratším vodičem představuje procentuální rozdíl

$$\Delta = \frac{6 \text{ ps}}{2,08 \text{ ns}} = 0,29 \%$$

V první fázi byly příslušné vodiče pospojovány bez ohledu na jejich délky. Po úspěšném vyřešení prostorových dispozic na DPS bylo přistoupeno na měření délek vodičů v aktuálním stavu. KiCad zobrazuje délku aktuálně označeného segmentu ve stavovém řádku okna. Zde je také zobrazena celková délka vodiče až k případnému rozvětvení (viz obrázek 23).



Obrázek 22: Motiv propojení procesoru a operačních pamětí

Změřené délky jsme zanesli do tabulkového procesoru společně se zvolenými konečnými délkami skupin vodičů. Následovalo samotné prodlužování. Začali jsme od adresních a řídicích vodičů ležících ve spodní vrstvě uprostřed. Nejprve byla dorovnáována délka horní a spodní části po rozvětvení, nakonec celková délka prodloužením vodiče před rozvětvením. Každá úprava délky byla opět zanesena do tabulkového procesoru pro následnou kontrolu. Jako další přišla na řadu úprava především datových vodičů v horní vrstvě. Protože se tyto spoje nevětví, šlo o jednodušší práci. Délka vodičů byla upravována

vestavěným nástrojem KiCadu, nebo v některých případech ručně, vytvořením různých smyček a záhybů. Celý proces byl několikrát opakován, protože posouváním spojů uprostřed DPS vedlo k rozbíjení okolních již přizpůsobených vodičů. KiCad v tomto případě nerespektoval smyčky vytvořené na vodičích a i v případě nastavení nejmenší usilovnosti (effort) routeru tyto smyčky mazal. Bylo by zajímavé zjistit, jaké jsou v tomto ohledu možnosti u jiného SW s interaktivním routováním, jakým je například Altium. Nakonec byly délky upravovaných vodičů přeměřeny, opět zaneseny do tabulkového procesoru a naposled zkontrolovány. Úprava délek vodičů byla velmi zdlouhavá a zabrala nejvíce času z práce na tomto projektu.

Type	NetName	NetCode	Status	Layer	Width	Segment Length	Length
Track	/DDR/DDR_ODT	29.1	..	B.Cu	0,150 mm	2,578 mm	43,200 mm

Obrázek 23: Zobrazení délky vodiče v KiCadu

10 Nastavení parametrů systému

Nastavení parametrů, jako například hodinovou frekvenci operačních pamětí aj., procesor parsuje ze souboru *script.bin* z bootovacího oddílu. Pro snadnější úpravu lze soubor převést do textové podoby (*script.fex*) programem *bin2fex* a po úpravě zpět do binární podoby programem *fex2bin*. Soubor používá strukturu podobnou konfiguračním souborům operačního systému Linux a rozlišuje velká a malá písmena. Každá sekce začíná názvem v hranatých závorkách, každý další řádek obsahuje dvojici „klíč = hodnota”. Vše, co je na řádku uvedeno za středníkem, překladač ignoruje; této skutečnosti je využito pro psaní komentářů.

Úvodní část souboru obsahuje verzi a název produktu. V sekci *target* je nastavená výchozí frekvence procesoru a napětí zdrojů. V dalších sekcích můžeme nastavit frekvence oscilátorů, adresu zaváděcího sektoru a především namapování všech portů na fyzické piny procesoru. Definice GPIO portu vypadá následovně

```
port:<název><funkce><pullup/pulldown><výstupní_proud><výchozí_úroveň>
```

Názvy portů a jejich funkce jsou přehledně uvedené v tabulce na webu <http://linux-sunxi.org/A20/PIO>. Při úpravách souboru je potřebné dbát na zvýšenou opatrnost. Špatné nastavení například výstupních napětí měničů může vést k nevratnému zničení komponent na DPS. [39]

10.1 Kalibrace DRAM

Po vodičích SDQ probíhá komunikace v obou směrech mezi procesorem a paměťmi. Proto musí DRAM přepínat porty mezi módy čtení a zápisu ve správný časový okamžik. Po odeslání příkazu do DRAM dorazí odpověď až po určitém časovém úseku. Pro větší odolnost vůči chybám jsou před a za každý příkaz přidány ochranné intervaly v délkách 0,9 a 0,3 celkové délky cyklu. Důležitým parametrem je také zpoždění mezi odesláním příkazu a přijetím odpovědi. Tato doba může být automaticky změřena, což je naneštěstí v některých případech nespolehlivé. Jedná se zejména o studený start systému a vyšší taktovací frekvence pamětí. Proto se jako vhodnější jeví změření a nastavení těchto parametrů manuálně v konfiguračním souboru (*script.fex*).

Dalším parametrem, který ovlivňuje integritu signálu, je správné impedanční zakončení. Impedance vodivých cest je pevně dána, může být ovlivněna pouze při návrhu. Zakončovací impedance vodičů na straně DRAM paměti jde konfigurovat pomocí proměnných *zq*, *odt_en* a *emr1* v konfiguračním souboru. Tyto proměnné určují velikost zakončovací impedance v poměru ke kalibračním rezistorům připojeným k pinům SZQ na straně procesoru a ZQ na straně DRAM. Následujícím nastavovaným parametrem je zpoždění signálu mezi hodinovými (CLK) a ovládacími signály (SDQ). Tato zpoždění neumí hardware nijak změřit, proto je musíme zjistit a nastavit ručně v proměnné *tpr3* v konfiguračním souboru. Zde můžeme nastavit zpoždění zvlášť pro čtení a zápis všech čtyř osmibitových skupin datových vodičů. Hodnoty časování jsou specifické pro každý typ DRAM, výrobce je uvádí v katalogovém listu. Tyto hodnoty je potřeba zapsat do proměnných *tpr0*, *tpr1* a *tpr2* v konfiguračním souboru. [39]

Optimální parametry pro konkrétní DPS lze nalézt i „hrubou silou”. Potřebujeme k tomu linuxový systém s jádrem *sunxi-3.4* a doporučuje se nastavit síťové bootování ze vzdáleného NFS. Celý postup lze nalézt na webové stránce http://linux-sunxi.org/A10_DRAM_Controller_Calibration. Tímto způsobem lze vyzkoušet různé kombinace nastavení parametrů a vybrat z nich nejvhodnější pro náš návrh. [40]

<pre> dcdc3_vol = 1325 dram_clk = 648 mbus_clk = 600 dram_type = 3 dram_rank_num = 1 dram_chip_density = 4096 dram_io_width = 8 dram_bus_width = 32 dram_cas = 9 dram_zq = 0x2c (0x10d3900) dram_odt_en = 3 dram_tpr0 = 0x429899b4 dram_tpr1 = 0xa0a0 dram_tpr2 = 0x2c200 dram_tpr3 = 0x182222 dram_emr1 = 0x42 dram_emr2 = 0x10 dram_emr3 = 0x0 dqs_gating_delay = 0x07070707 active_windowing = 1 </pre>	<pre> mfxdly phase=36 phase=54 phase=72 phase=90 phase=108 phase=126 0x07 0x073333 0x072222 0x071111 0x070000 0x07EEEE 0x07DDDD 0x06 0x063333 0x062222 0x061111 0x060000 0x06EEEE 0x06DDDD 0x05 0x053333 0x052222 0x051111 0x050000 0x05EEEE 0x05DDDD 0x04 0x043333 0x042222 0x041111 0x040000 0x04EEEE 0x04DDDD 0x03 0x033333 0x032222 0x031111 0x030000 0x03EEEE 0x03DDDD 0x02 0x023333 0x022222 0x021111 0x020000 0x02EEEE 0x02DDDD 0x01 0x013333 0x012222 0x011111 0x010000 0x01EEEE 0x01DDDD 0x00 0x003333 0x002222 0x001111 0x000000 0x00EEEE 0x00DDDD 0x08 0x083333 0x082222 0x081111 0x080000 0x08EEEE 0x08DDDD 0x10 0x103333 0x102222 0x101111 0x100000 0x10EEEE 0x10DDDD 0x18 0x183333 0x182222 0x181111 0x180000 0x18EEEE 0x18DDDD 0x20 0x203333 0x202222 0x201111 0x200000 0x20EEEE 0x20DDDD 0x28 0x283333 0x282222 0x281111 0x280000 0x28EEEE 0x28DDDD 0x30 0x303333 0x302222 0x301111 0x300000 0x30EEEE 0x30DDDD 0x38 0x383333 0x382222 0x381111 0x380000 0x38EEEE 0x38DDDD </pre>	<pre> Lane phase adjustments: [0, 0, 0, 0] Error statistics from memtester: [bitflip=19, solidbits=12] Total number of successful memtester runs: 235 Best luminance at the height 0.5 is above 0x081111, score = 0.742 Best luminance at the height 1.0 is above 0x081111, score = 0.634 Best luminance at the height 2.0 is above 0x081111, score = 0.512 Best luminance at the height 3.0 is above 0x081111, score = 0.440 Read errors per lane: [3, 0, 12, 2]. Lane 1 is the most noisy/problematic. Errors from the lane 0 are not intersecting with the errors from the worst line 1. Errors from the lane 3 are not intersecting with the errors from the worst line 1. Write errors per lane: [1, 1, 2, 11]. Lane 0 is the most noisy/problematic. Errors from the lane 1 are 50.0% eclipsed by the worst lane 0. Errors from the lane 2 are not intersecting with the errors from the worst line 0. Errors from the lane 3 are not intersecting with the errors from the worst line 0. </pre>
--	--	--

Obrázek 24: Vizualizace výsledků testování DRAM [40]

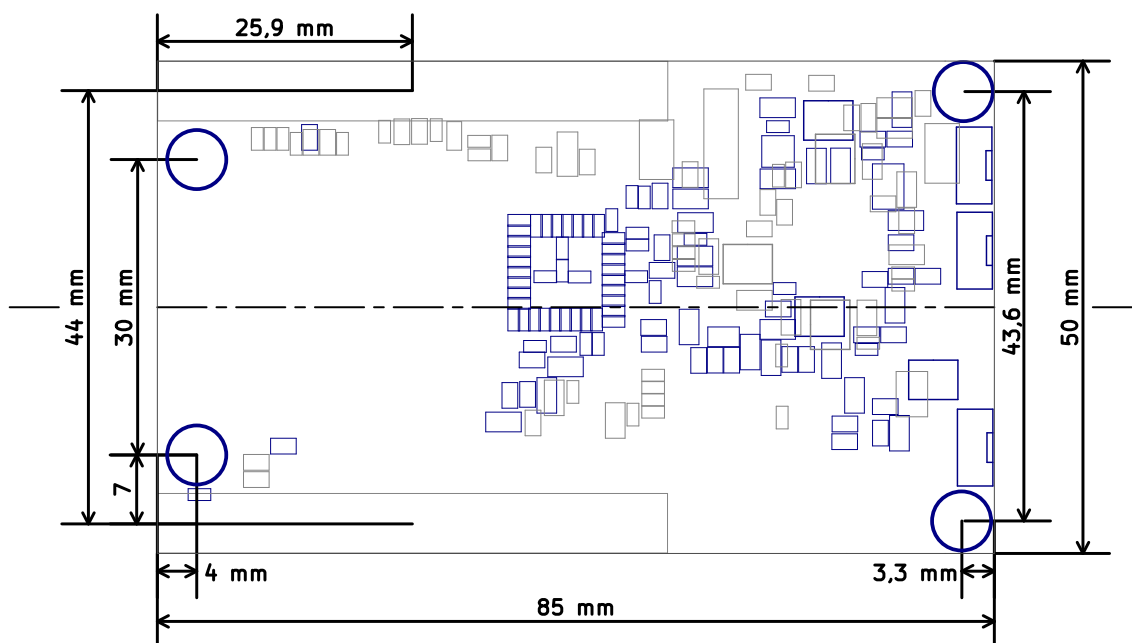
10.2 Bootloader

Procesor se pokouší ve výchozím stavu nahrát bootloader ze zařízení v následujícím pořadí: SDC0 → NAND FLASH → SDC2 → SPI0. Pokud se mu nepodaří nabootovat ani z jednoho zařízení, přepne se do režimu bootování z USB. [28]

Jako bootloader v linuxových systémech je použit U-boot. [41] První část MicroSD karty obsahuje 16 MB velký oddíl formátovaný jako FAT, ve kterém je nahané linuxové jádro a konfigurační soubor, obojí v binární podobě. Druhá část karty obsahuje oddíl s programy a soubory operačního systému. U-boot podporuje také bootování ze sítě nebo pevného disku, připojeného přes SATA rozhraní. [41]

11 Rozšiřující moduly

Vlastní deska systému obsahuje jen základní části nutné pro běh systému. Funkčnost může být rozšířena připojením přídatného modulu pomocí pinových lišt. Předpokládaná velikost modulu je stejná jako vlastní desky, tedy 85 x 50 mm, s konektory v levé části u delších hran. Každý z konektorů obsahuje celkem 40 pinů rozdělených do dvou řad, zapojení obou lišt je zakresleno na obrázku 18 na straně 22. Pro pevnější mechanické spojení rozšiřujícího modulu a desky lze použít například distanční sloupky ve spojení s montážními otvory v rozích desky.



Obrázek 25: Rozměry DPS a poloha propojovacích lišt

Na pinové lišty je vyvedeno několik skupin vodičů. V následující části bude rozebráno, jaké další obvody budou na přídatném modulu potřeba, abychom mohli jednotlivé periferie provozovat.

11.1 SATA

K přidání SATA rozhraní na rozšiřující modul je potřebné vyvést symetrické páry datových sběrnic SATA_RX a SATA_TX, celkem 4 vodiče. Před konektor je vhodné na každý datový vodič přidat sériový blokující kondenzátor o hodnotě například 10 nF. Pro 2,5" pevné disky potřebujeme napájecí napětí 5 V z vodiče 5V0. V případě problémů s vysokým odběrem můžeme disk napájet přímo ze vstupního napětí 5V_EXT, toto napájení ale není zálohované z připojeného akumulátoru. V případě použití 3,5" pevného disku musíme řešit potřebu napájecího napětí 12 V. Protože jsou tyto disky původně určené do stolních počítačů, mívají také větší odběr než 2,5" disky. Možným řešením je napájení celého systému 12 V a přidáním měniče na 5 V na rozšiřující modul. Od procesoru je také vyveden signál SATA_PWR_EN, který ovládá připojení napájecího napětí k pevnému disku. [7]

11.2 HDMI

Pro vyvedení HDMI je potřebných celkem 12 vodičů. Jedná se o čtyři symetrické datové páry a další čtyři datové vodiče. Je doporučeno na všechny vodiče přidat přepětové ochrany. Na vývojové desce A20-OLinuXino-MICRO je použit obvod RClamp0524P. Jedná se o čtyřnásobné pole TVS diod v kompaktním pouzdře (2,5 x 1 x 0,58 mm), s malou kapacitancí přechodu, navržené pro použití v aplikacích jako je DVI, DisplayPort, PCI Express, eSATA, nebo právě HDMI. [42] Pro správnou funkci rozhraní je nutné na vodiče HSCL a HSDA připojit pullup rezistory na napětí 5 V. [7]

11.3 USB

Na lišty jsou vyvedena dvě USB rozhraní. Každé z nich obsahuje symetrický datový pár a jeden řídicí vodič pro zapnutí napájení. Ke správné funkci je také nutné přivést napájecí napětí 5V0 s vhodně vřazenou proudovou ochranou, například obvodem SY6280, u kterého můžeme nastavit omezovací proud pomocí jednoho rezistoru. Tento obvod také obsahuje ochranu proti přehřátí a je dodáván v pouzdře SOT23.

Jako třetí rozhraní je možné na desku umístit USB-OTG. Jedná se o obousměrné USB rozhraní, které lze využívat jako běžný USB port, nebo ho také lze přepnout do režimu, kdy se chová jako host. Tento režim je využíván u chytrých mobilních telefonů, které se po připojení k počítači jeví jako datové úložiště. Oproti běžnému USB portu je přítomna další dvojice vodičů pro detekci režimu portu. [7]

11.4 SD paměťová karta

Paměťová karta je zapojena stejně jako jedna již přítomná, pomocí čtyř datových vodičů, dvěma vodiči s hodinovým a řídicím signálem a jedním vodičem pro rozpoznání přítomnosti karty ve slotu. Tato karta může sloužit jako rozšíření paměťové kapacity zařízení, není možné z ní zavést systém. SoC A20 podporuje paměťové karty standardů SDSC / SDHC / SDXC / MMC s kapacitou až 128 GB. [7, 28]

11.5 Audio

K dispozici je stereo sluchátkový výstup, stereo nebo symetrický linkový vstup a stereo mikrofonní vstup. SoC obsahuje 24 bitové DAC a ADC převodníky se vzorkovací frekvencí až 192 KHz. Na propojovací lištu je přivedeno odpovídající množství vodičů spolu s napájecím napětím pro mikrofony a vlastní audio zemí. [28]

11.6 Ethernet

Pro připojení sítě k procesoru je použito rozhraní MII. Převod na Ethernet zajistí například obvod RTL8201. Na vstupu má 18 vodičů, na výstupu 2 diferenciální páry. Rychlost Ethernetu může být 10 Mbit/s nebo 100 Mbit/s. K obvodu je přes TWI sběrnici připojena EEPROM paměť s uloženou MAC adresou. Napájecí napětí obvodu je 3,3 V. [43]

11.7 UART, TWI, GPIO

Pro připojení dalších komponent jsou na lištu vyvedeny sběrnice UART a TWI na napěťové úrovni 3,3 V. Na oba vodiče TWI, datový i hodinový, jsou připojeny pullup rezistory. Na liště jsou k dispozici i čtyři GPIO piny pro obecné použití, PI0 - PI3.

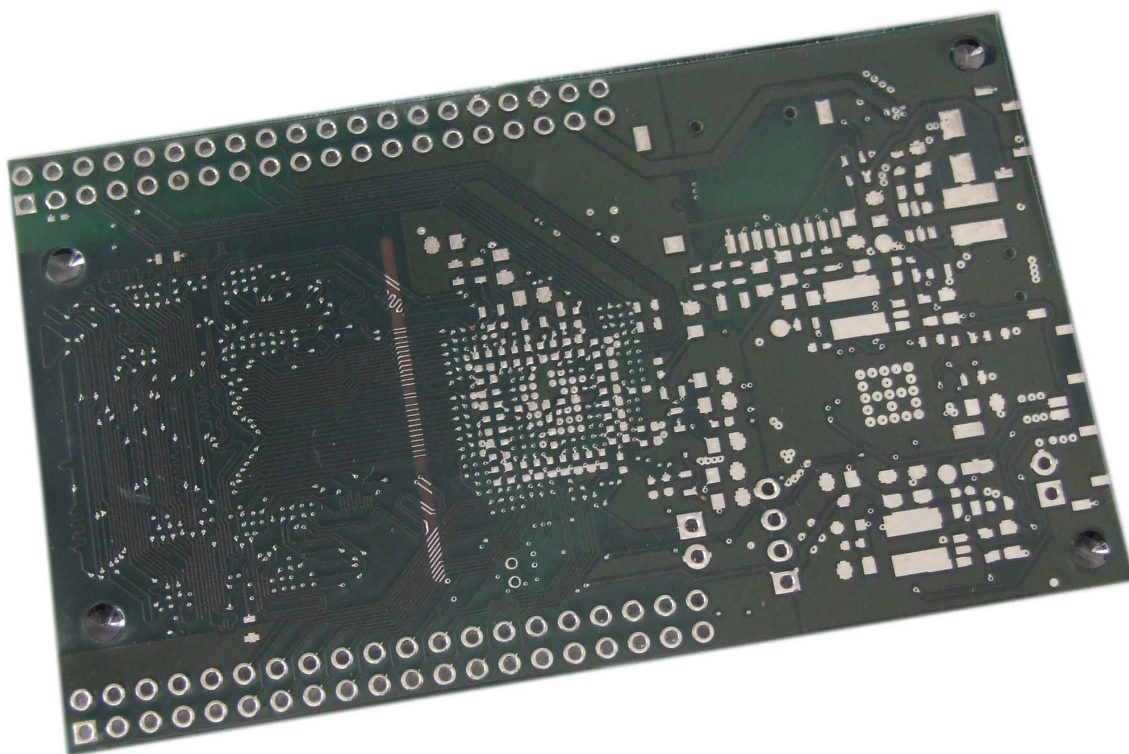
11.8 Napájení

Na přídatné desce můžeme využít několik napájecích napětí ze základové desky, konkrétně 3V0, 3V3, 5V0 a 5V_EXT. Vyvedené jsou také vodiče pro připojení baterie. Této skutečnosti lze využít pro napájení desky z jiného externího zdroje. Na lištu jsou taktéž vyvedeny obě země, analogová i digitální. Spojené jsou v jednom místě na vrchní straně základové desky.

12 Realizace

Hotový návrh DPS byl z KiCadu exportován do formátu Gerber RS-274X. V našem případě se jednalo o čtyři soubory s motivem leptaným do mědi, dva soubory s vrchní a spodní nepájivou maskou a jeden soubor s rozměry pro oříznutí DPS. Jako další jsme vyexportovali dva soubory ve formátu Excellon s daty pro vrtání prokovených a neprokovených otvorů. Data byla následně odeslána do firmy, která podle nich vyrobila DPS.

V následujícím kroku byla vytvořena osazovací dokumentace skládající se ze dvou osazovacích výkresů, jeden pro každou stranu desky, a soupisky všech součástek včetně jejich označení a pouzdra. K dokumentaci byla přiložena i vizualizace DPS ve 3D. Vyrobená DPS byla podle dokumentace osazena v další firmě. Vlastní osazení bylo provedeno ve dvou krocích. V prvním kroku se osazovaly součástky v FBGA pouzdrech, tedy procesor a dvě DRAM. Po přetavení v pájecí peci následovalo ruční osazení zbytku součástek. Na jemném motivu FBGA pouzder došlo v některých částech k nahromadění pájky a naopak na některé z pinů se pájka nechytla vůbec. Jako vhodnější než HASL se tedy jeví použití jiné povrchové úpravy DPS, například pozlacení.



Obrázek 26: DPS před osazením

Celkové náklady na výrobu prvního prototypu se vyšplhaly na cca 5200 Kč. Tato částka byla zhruba stejnými díly rozdělena mezi nákup součástek, výrobu DPS a osazení. Cena vývojové desky A20-OLinuXino-MICRO je asi 65 € (1800 Kč) a známý embedded počítač Raspberry Pi 2 B přijde na zhruba 1100 Kč. [5, 44]. Prototyp je v současné době ve stádiu ožívování.

Pokud bychom měli vyvinutý embedded systém zběžně porovnat s Raspberry Pi 2 B, můžeme říci, že Raspberry bude zřejmě výkonnější, protože má o dvě procesorová jádra více. V ostatních pohledech jsou oba systémy velmi podobné. Výhodou Raspberry je větší

	Vyvinutý systém	Raspberry Pi 2 B
SoC	Allwinner A20, 2 x 1GHz	Broadcom BCM2836, 4 x 900 MHz
Architektura	Cortex-A7	Cortex-A7
GPU	Mali400 MP2	VideoCore IV
RAM	1 GB	1 GB
USB	3x	4x
Rozměry	85 x 50 mm	85,6 x 56,5 mm

Tabulka 7: Srovnání vyvinutého embedded systému s Raspberry Pi 2 [28, 45]

kompaktnost z celkového pohledu, protože na DPS již jsou přítomny USB konektory, HDMI aj. Vyvinutá deska tyto konektory neobsahuje přímo, ale mohou být zahrnuty na přídatném modulu. [45]

Vývoj vlastního embedded systému s plánovaným výrobním objemem v desítkách kusů se z finančního hlediska zřejmě nevyplatí. Výjimku by mohli tvořit velmi malé a jednoduché systémy, u kterých by se podařilo snížit cenu za výrobu DPS a osazení. V opačném případě je výhodnější použít již existující platformu, která se vyrábí ve velkých objemech a cena tedy může být přijatelnější. Vývoj vlastního embedded systému se také může vyplatit v případě speciálních požadavků na systém, které nemůžou uspokojit běžně dostupné systémy.

13 Závěr

V rámci projektu byl na vnitřní FLASH paměť vývojové desky Olimex A20-OLinuXino-Micro-4GB nainstalován operační systém Arch Linux ARM. Dále byla vyvinuta testovací deska obsahující mikroprocesor AVR ATmega, dvojici relé a převodník UART / RS-485 pro její ovládání. Operační systém byl nakonfigurován a opatřen potřebným softwarem pro vývoj v jazyce Python. Následně byl navrhnut a implementován protokol pro komunikaci po sériové sběrnici mezi vývojovou deskou a periferií. Též bylo naprogramováno webové rozhraní, skrz které lze ovládat všechny výstupy na relé desce. V následující části projektu byla změřena rychlost odezvy GPIO pinu. Frekvence spínání se řádově pohybovala v okolí 30 kHz při plném vytížení jednoho jádra procesoru.

V další části byl navržen vlastní systém na platformě Allwinner A20 SoC na základě již existujících řešení. Byly rozebrány hlavní použité součásti a software. Návrh schématu a DPS byl realizován v Open Source programu KiCad. Nejsložitější částí celého projektu byl návrh DPS, konkrétně propojení procesoru a operačních pamětí. Většina vodičů mezi těmito součástmi musí být pro zajištění bezproblémové funkčnosti impedančně a délkově přizpůsobena. Navržená DPS byla vyrobena a osazena, aktuálně se nachází ve stádiu oživování. Poslední část práce se věnuje popisu návrhu rozšiřujícího modulu, který může být podle potřeby upraven.

Literatura

- [1] A20. OLIMEX. Olimex [online]. 2014 [cit. 2014-12-15]. Dostupné z: <https://www.olimex.com/Products/Components/IC/A20/>
- [2] RAGHAVAN, P., LAD, A., NEELAKANDAN, S. Embedded Linux System Design and Development, CRC Press, 2005.
- [3] NVIDIA Powers New Honda Infotainment System. Nvidia Corporation [online]. 2014 [cit. 28.11.2014] Dostupné z: <http://nvidianews.nvidia.com/news/nvidia-powers-new-honda-infotainment-system>
- [4] ESP8266 Community Wiki [online]. 2015 [cit. 2015-05-16]. Dostupné z: <http://www.esp8266.com/wiki>
- [5] A20-OLinuXino-MICRO. OLIMEX. Olimex [online]. 2014 [cit. 25.11.2014]. Dostupné z: <https://www.olimex.com/wiki/A20-OLinuXino-MICRO>
- [6] OLINUXINO GitHub. OLIMEX. Olimex [online] [cit. 25.11.2014]. Dostupné z: <https://github.com/OLIMEX/OLINUXINO>
- [7] OLIMEX. A20-OLinuXino-MICRO USER'S MANUAL [online]. 2014 [cit. 28.11.2014]. Dostupné z: <https://www.olimex.com/Products/OLinuXino/A20/A20-OLinuXino-MICRO-4GB/resources/A20-OLinuXino-Micro.pdf>
- [8] ALLWINNER. A20 Datasheet [online]. 2015 [cit. 16.5.2015]. Dostupné z: <https://github.com/allwinner-zh/documents/tree/master/A20>
- [9] OLIMEX. UEXT Modules [online]. 2014 [cit. 28.11.2014]. Dostupné z: <https://www.olimex.com/Products/Modules/>
- [10] ATMEL. ATmega8/L datasheet [online]. 2014 [cit. 28.11.2014]. Dostupné z: <http://www.atmel.com/devices/atmega8.aspx>
- [11] WIKIBOOKS. RS-485 Technical Manual [online]. 2014 [cit. 28.11.2014]. Dostupné z: http://en.wikibooks.org/wiki/Serial_Programming/RS-485
- [12] TEXAS INSTRUMENTS. SN65176B datasheet [online]. 2014 [cit. 28.11.2014]. Dostupné z: <http://www.ti.com/lit/ds/symlink/sn65176b.pdf>
- [13] VINET, J., GRIFFIN, A. ArchWiki. [online]. 2014 [cit. 25.11.2014]. Dostupné z: <https://wiki.archlinux.org/>
- [14] OLIMEX. Unofficial Arch Linux ARM image for A20-OLinuXino-MICRO [online]. 2014 [cit. 28.11.2014]. Dostupné z: <https://www.olimex.com/forum/index.php?topic=1856.0>
- [15] OLIMEX. How to install Debian on NAND in A20-MICRO? [online]. 2014 [cit. 28.11.2014]. Dostupné z: <https://www.olimex.com/forum/index.php?topic=2028.0>
- [16] OLIMEX. A20-OLinuXino-MICRO Wiki [online]. 2015 [cit. 16.5.2015]. Dostupné z: <https://www.olimex.com/wiki/A20-OLinuXino-MICRO>

- [17] A20 OlinuXino Micro [online]. 2015 [cit. 2015-05-16]. Dostupné z: <http://archlinuxarm.org/platforms/armv7/allwinner/a20-olinuxino-micro>
- [18] THE APACHE SOFTWARE FOUNDATION. Dynamic Content with CGI [online]. 2014 [cit. 28.11.2014]. Dostupné z: <http://httpd.apache.org/docs/current/howto/cgi.html>
- [19] PYTHON SOFTWARE FOUNDATION. Python Wiki [online]. 2014 [cit. 28.11.2014]. Dostupné z: <https://wiki.python.org/moin/>
- [20] RONACHER, A. Flask [online]. 2014 [cit. 25.11.2014]. Dostupné z: <http://flask.pocoo.org/>
- [21] LIECHTI, C. pySerial's documentation [online]. 2013 [cit. 28.11.2014]. Dostupné z: <http://pyserial.sourceforge.net/>
- [22] OLIMEX. pyA20 0.2.0 [online]. 2014 [cit. 28.11.2014]. Dostupné z: <https://pypi.python.org/pypi/pyA20>
- [23] Codiad: Web Based, Cloud IDE [online]. 2015 [cit. 16.5.2015]. Dostupné z: <http://codiad.com/>
- [24] A20 Boards. Wiki of the linux-sunxi community [online]. 2015 [cit. 16.5.2015]. Dostupné z: http://linux-sunxi.org/Category:A20_Boards
- [25] A10 Boards. Wiki of the linux-sunxi community [online]. 2015 [cit. 16.5.2015]. Dostupné z: http://linux-sunxi.org/Category:A10_Boards
- [26] A minimal single board computer using Allwinner A10 SOC [online]. 2015 [cit. 16.5.2015]. Dostupné z: <http://www.iuac.res.in/~elab/phoenix/SBC/>
- [27] System on Module Board 2 [online]. 2015 [cit. 16.5.2015]. Dostupné z: <http://www.gplsquared.com/SoM2/SoM2.html>
- [28] ALLWINNER. A20 User manual [online]. 2015 [cit. 16.5.2015]. Dostupné z: <https://github.com/allwinner-zh/documents/tree/master/A20>
- [29] ALLWINNER. A20 [online]. 2015 [cit. 16.5.2015]. Dostupné z: <http://www.allwinnertech.com/en/clq/processora/A20.html>
- [30] AXP209. Wiki of the linux-sunxi community [online]. 2015 [cit. 16.5.2015]. Dostupné z: <http://linux-sunxi.org/AXP209>
- [31] X-Powers. AXP20X [online]. 2015 [cit. 16.5.2015]. Dostupné z: http://www.x-powers.com/product/AXP20X_en.php
- [32] KiCad EDA Software Suite [online]. 2015 [cit. 16.5.2015]. Dostupné z: <http://www.kicad-pcb.org/display/KICAD/KiCad+EDA+Software+Suite>
- [33] KiCad Electronic CAD Libraries [online]. 2015 [cit. 16.5.2015]. Dostupné z: <http://smisioto.no-ip.org/elettronica/kicad/kicad-en.htm>

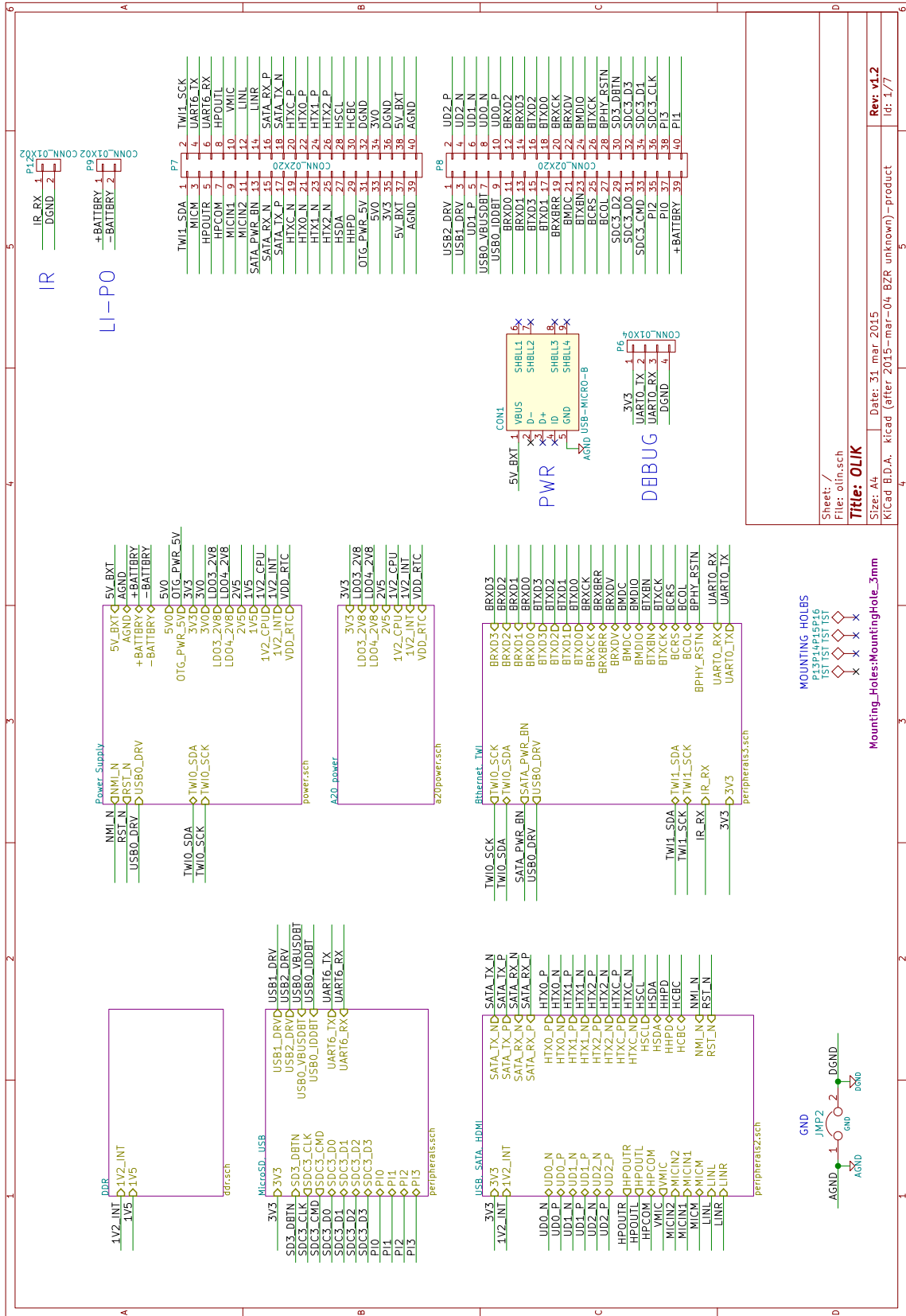
- [34] SILERGY. SY8009A datasheet [online]. 2015 [cit. 16.5.2015]. Dostupné z: <http://www.applelogic.org/files/SY8009A.pdf>
- [35] AXP 209 Datasheet [online]. 2015 [cit. 16.5.2015]. Dostupné z: http://linux-sunxi.org/File:AXP209_Datasheet_v1.0en.pdf
- [36] A10 Cubieboard Schematics [online]. 2015 [cit. 16.5.2015]. Dostupné z: https://www.openhacks.com/uploads/productos/a10_cubieboard_120808.pdf
- [37] HYNIX. 8Gb DDR3 SDRAM [online]. 2015 [cit. 16.5.2015]. Dostupné z: [https://www.skhynix.com/inc/pdfDownload.jsp?path=/datasheet/pdf/dram/Computing_DDR3_H5TQ8G4\(8_6\)3AMR\(Rev0.1\).pdf](https://www.skhynix.com/inc/pdfDownload.jsp?path=/datasheet/pdf/dram/Computing_DDR3_H5TQ8G4(8_6)3AMR(Rev0.1).pdf)
- [38] TEXAS INSTRUMENTS. High-Speed Layout Guidelines [online]. 2015 [cit. 16.5.2015]. Dostupné z: <http://www.ti.com/lit/an/scaa082/scaa082.pdf>
- [39] Fex Guide. Wiki of the linux-sunxi community [online]. 2015 [cit. 16.5.2015]. Dostupné z: http://linux-sunxi.org/Fex_Guide
- [40] A10 DRAM Controller Calibration. Wiki of the linux-sunxi community [online]. 2015 [cit. 16.5.2015]. Dostupné z: http://linux-sunxi.org/A10_DRAM_Controller_Calibration
- [41] U-Boot [online]. 2015 [cit. 16.5.2015]. Dostupné z: <http://www.denx.de/wiki/U-Boot>
- [42] SEMTECH. RClamp0524P Datasheet [online]. 2015 [cit. 16.5.2015]. Dostupné z: http://www.semtech.com/images/datasheet/rclamp0522p_0524p.pdf
- [43] REALTEK. RTL8201 Datasheet [online]. 2015 [cit. 16.5.2015]. Dostupné z: <http://realtek.info/pdf/rtl8201.pdf>
- [44] RASPBERRY PI 2 MODEL B 1GB [online]. 2015 [cit. 16.5.2015]. Dostupné z: <http://eshop.minidroid.cz/vsechno-pro-raspberry-pi/335-raspberry-pi-2-model-b-1gb.html>
- [45] RASPBERRY PI 2 MODEL B [online]. 2015 [cit. 16.5.2015]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

Seznam symbolů, veličin a zkratk

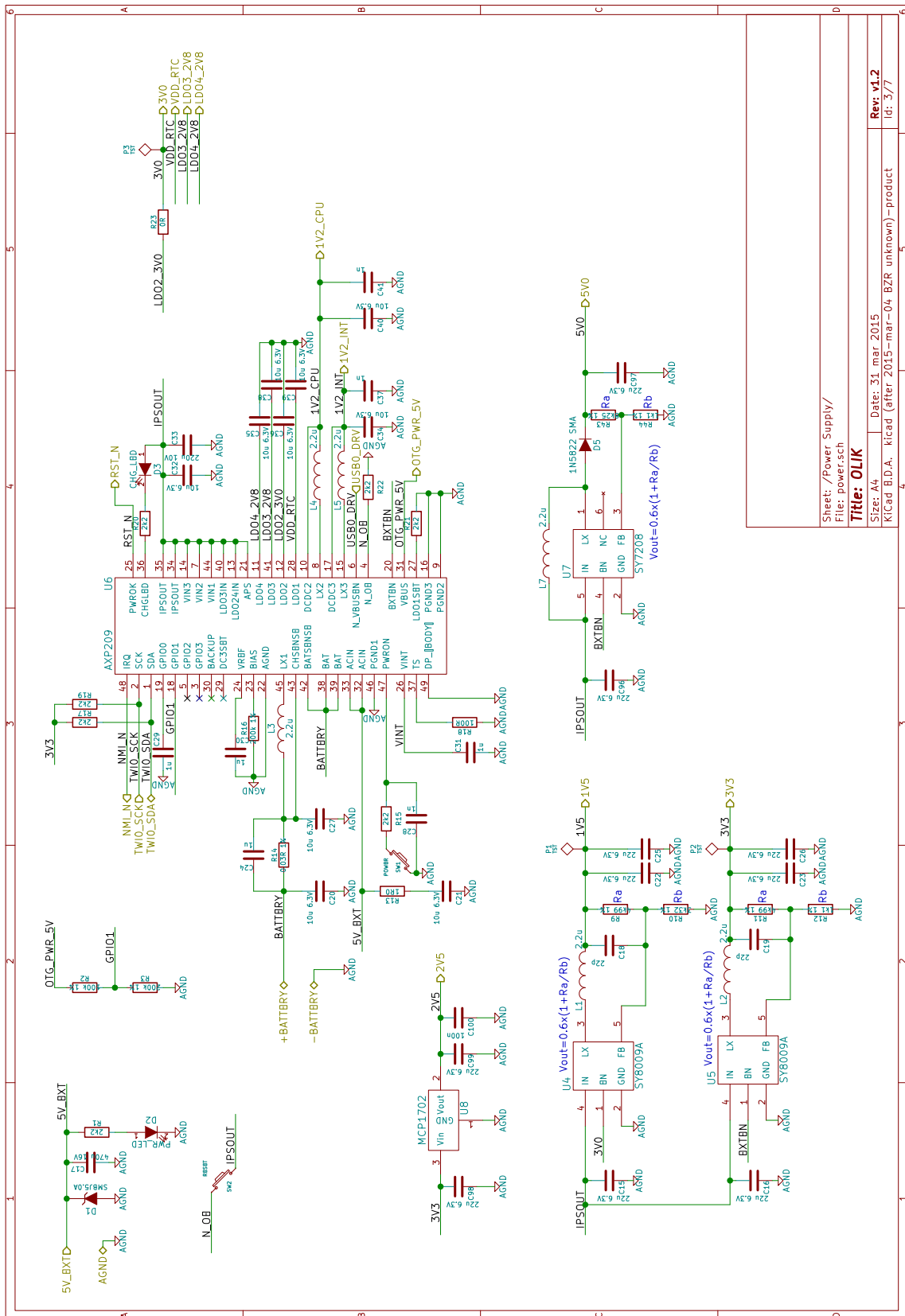
API	Application Programming Interface
ARM	Advanced RISC Machine
CGI	Common Gateway Interface
CPU	Central Processing Unit
DPS	Deska Plošných Spojů
DVI	Digital Visual Interface
DRAM	Dynamic Random Access Memory
eSATA	external SATA
ESD	Electrostatic Discharge
FBGA	Fine pitch Ball Grid Array
GPIO	General-Purpose Input/Output
GPU	Graphics Processing Unit
HDMI	High-Definition Multimedia Interface
IDE	Integrated Development Environment
IR	Infrared Radiation
MAC	Media Access Control
MII	Media-independent interface
MMC	MultiMediaCard
PCI	Peripheral Component Interconnect
PMU	Power Management Unit
PoE	Power over Ethernet
PWM	Pulse Width Modulation
QFN	Quad Flat No-leads package
RTC	Real-Time Clock
SATA	Serial Advanced Technology Attachment
SD	Secure Digital
SDHC	SD High-Capacity
SDSC	SD Standard-Capacity

SDXC	SD eXtended-Capacity
SoC	System on a Chip
SOT	Small Outline Transistor
SSH	Secure Shell
TVS	Transient Voltage Suppression
TWI	Two-Wire Interface
UART	Universal Asynchronous Receiver and Transmitter
UEXT	Universal Extension
UI	User Interface
USB	Universal Serial Bus
USB-OTG	USB On-The-Go
VGA	Video Graphics Array

14 Přílohy

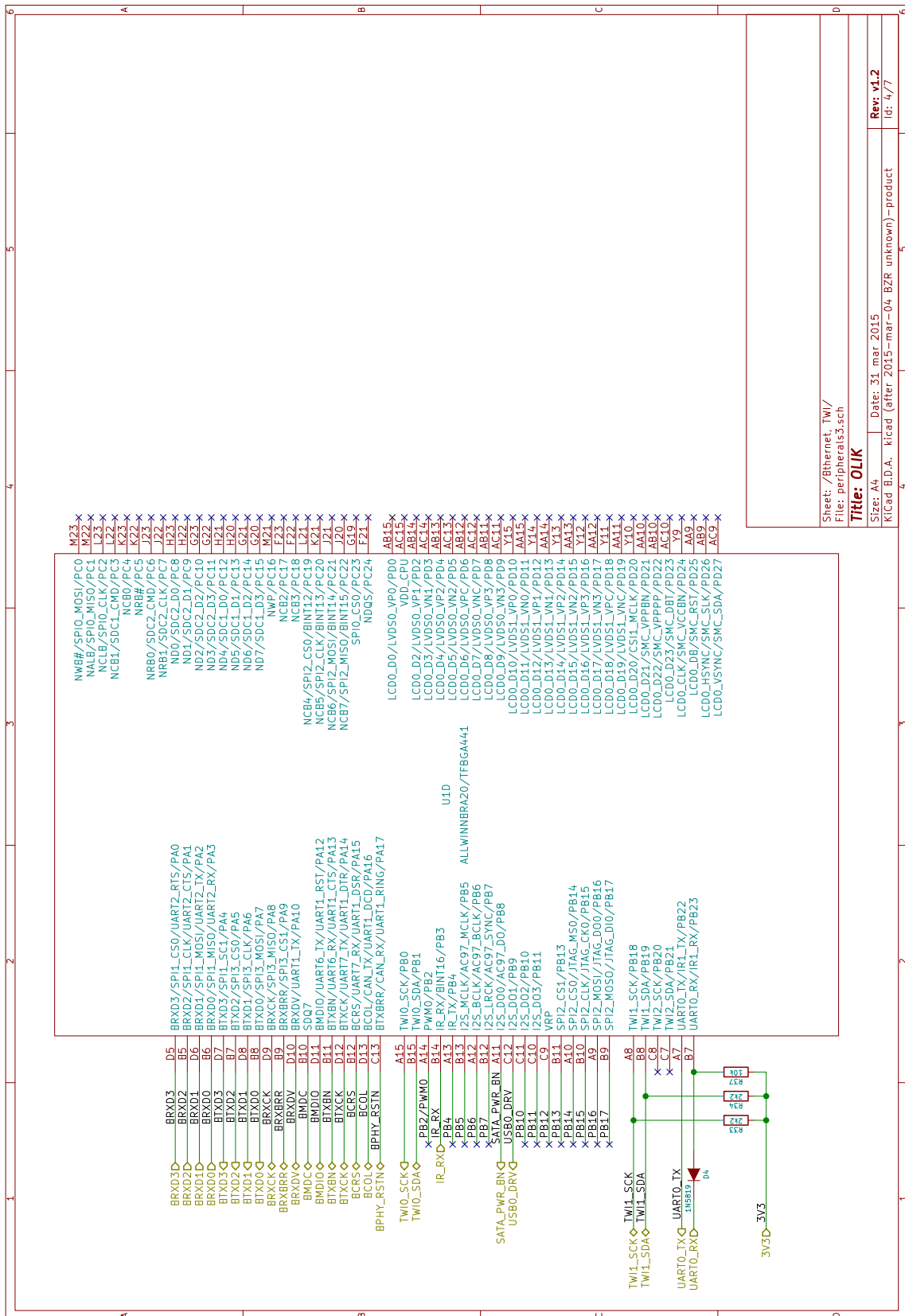


Obrázek 27: Hlavní schéma

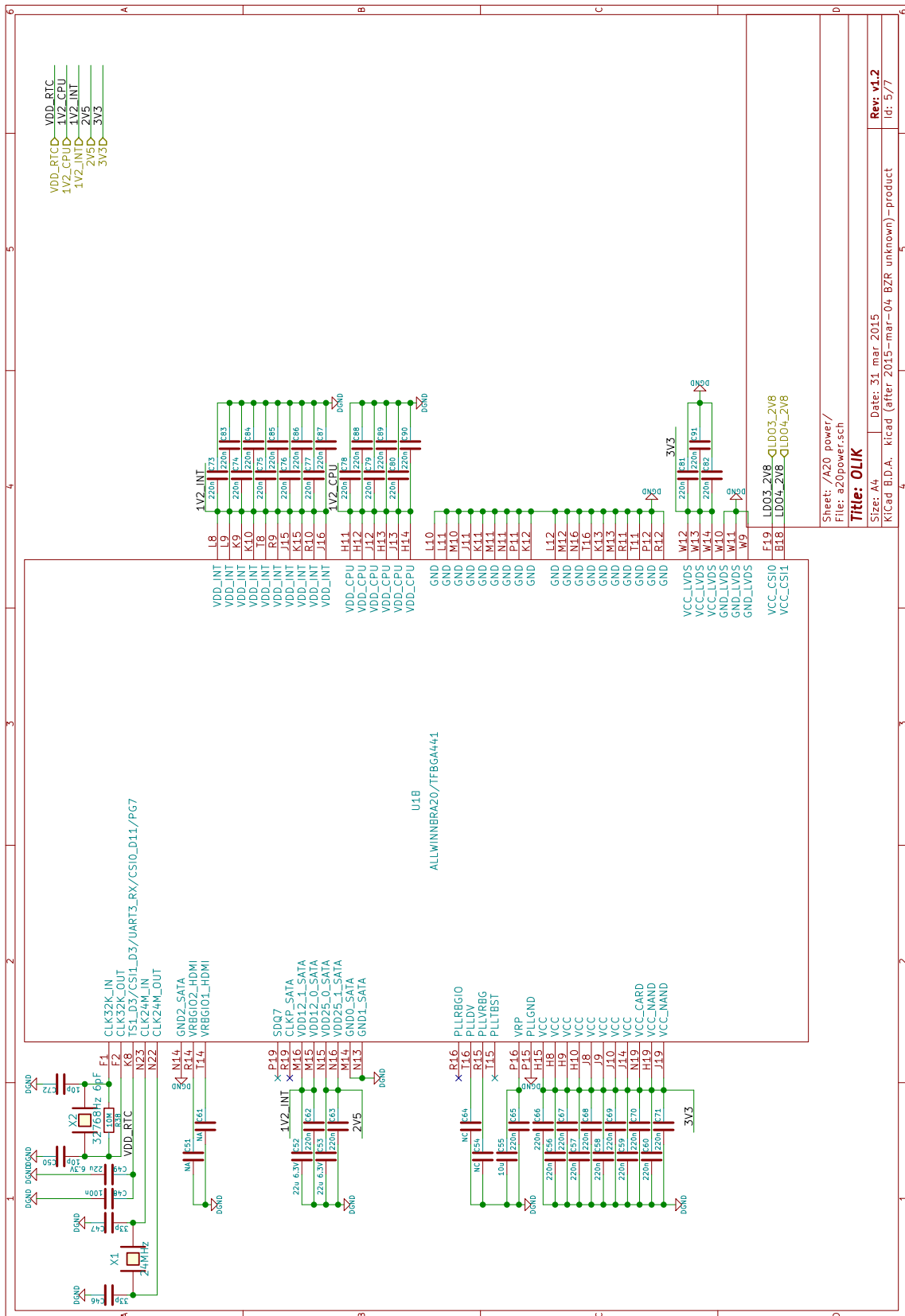


Sheet: /Power Supply/
File: power.sch
Title: OLIK
Size: A4
Date: 31. mar 2015
KiCad B.D.A. kicad (after 2015-mar-04 BZR unknown)-product
Rev: v1.2
Id: 3/7

Obrázek 29: Schéma zapojení napájení



Obrazek 31: Schéma zapojení Ethernetu



Sheet: /A20 power/
File: a20power.sch
Title: OLİK
Size: A4 Date: 31. mar 2015
Kicad B.D.A. kicad (after 2015-mar-04 BZR unknown)-product
Rev: v1.2
Id: 5/7

Obrázek 33: Schéma zapojení napájení SoC a krystalů