

## **Na tomto místě bude oficiální zadání vaší práce**

- Toto zadání je podepsané děkanem a vedoucím katedry,
- musíte si ho vyzvednout na studijním oddělení Katedry počítačů na Karlově náměstí,
- v jedné odevzdané práci bude originál tohoto zadání (originál zůstává po obhajobě na katedře),
- ve druhé bude na stejném místě neověřená kopie tohoto dokumentu (tato se vám vrátí po obhajobě).

## **Official thesis specification should be here**

- Ask study office on our department to obtain it!
- You have to post two copies – original and copy of your thesis.
- Official thesis specification has to be placed in each of them (original and copy).
- You receive back the thesis copy after succesful defension of your thesis.
- The original of your thesis is hold on departmnet after the defension.

## **Pokyny**

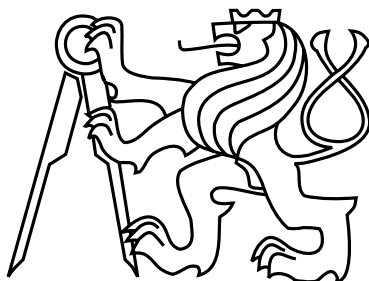
- Projednejte osnovu práce se svým vedoucím! Každému vedoucímu a typu práce nemusí níže prezentovaná osnova vyhovovat.
- Čtete pokyny v komentářích zdrojého souboru (.tex). Je tam mnoho užitečných informací.

## **Instruction**

- Discuss the intended structure of your thesis with your supervisor! Not each supervisor is satisfied with structure suplied in this document!
- Read insructions in comments in the source code of this document (.tex file). Many usefual additional instructions are included.



České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra počítačů



Bakalářská práce

## **Inteligentní zařízení pro získávání informací z internetu**

*Petr Čermák*

Vedoucí práce: Ing. Pavel Kubalík, Ph.D.

Studijní program: Elektrotechnika a informatika, strukturovaný, Bakalářský

Obor: Výpočetní technika

24. května 2010



## Poděkování

Rád bych poděkoval mému vedoucímu práce Ing. Pavlu Kubalíkovi. Dále děkuji své rodině a všem ostatním, kteří mi pomohli ke zdárnému dokončení práce.



## Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Nové Pakce 24. 5. 2010

.....





# Abstract

This work deals with the realization of a hardware device that will allow easy retrieval of data from the Internet. Contains a description of the software libraries and documentation portions of facilities has been implemented. In this work are presented and discussions addressing various components of the device.

# Abstrakt

Práce se zabývá realizací hardwarového zařízení, které bude umožňovat jednoduché získávání dat z internetu. Obsahuje popis softwarových knihoven a dokumentaci k implementovaným částem výsledného zařízení. V práci jsou uvedeny diskuze a řešení jednotlivých částí tohoto zařízení.



# Obsah

<b>Seznam obrázků</b>	<b>xv</b>
<b>Seznam tabulek</b>	<b>xvii</b>
<b>1 Úvod</b>	<b>1</b>
1.1 Specifikace cíle	2
<b>2 Rešerše dostupných zařízení</b>	<b>3</b>
2.1 PC Engines ALIX 1C1	3
2.2 Vývojový kit Olimex PIC-MAXI-WEB	4
2.3 Vývojový kit Web51	4
2.4 Srovnání a závěr	5
<b>3 Analýza</b>	<b>7</b>
3.1 Hardware	8
3.1.1 Mikroprocesor	8
3.1.2 Řadič ethernetu	9
3.1.3 Paměťové úložiště	10
3.1.3.1 Secure Digital Card	10
3.1.3.2 Compact Flash	10
3.1.3.3 MultiMedia Card	10
3.1.4 USB	11
3.1.5 RTC	11
3.1.6 EEPROM	11
3.1.7 RS-232	11
3.1.8 Rozšiřující konektory	12
3.1.8.1 IDE40	12
3.1.8.2 Displej	12
3.1.9 Přepínače a tlačítka	12
3.1.10 Napájení	13
3.1.11 Použité sběrnice	13
3.1.11.1 SPI	13
3.1.11.2 I2C	14
3.2 Firmware	15
3.2.1 TCP/IP Stack	15
3.2.1.1 Microchip TCP/IP stack	15
3.2.1.2 uIP	15
3.2.2 Souborový systém	16

3.3	Náklady na výrobu . . . . .	17
<b>4</b>	<b>Realizace</b>	<b>19</b>
4.1	Hardware . . . . .	19
4.1.1	Napájení . . . . .	19
4.1.2	Mikroprocesor . . . . .	20
4.1.3	Ethernet . . . . .	21
4.1.4	SD karta . . . . .	22
4.1.5	RTC . . . . .	22
4.1.6	Rozšiřující konektory . . . . .	23
4.1.7	RS-232 . . . . .	24
4.1.8	Přepínače a propojky . . . . .	24
4.1.9	Návrh schématu a DPS . . . . .	24
4.1.10	Chyby a problémy při řešení . . . . .	25
4.2	Software . . . . .	25
4.2.1	Vývojové prostředí MPLAB IDE . . . . .	25
4.2.2	Knihovny . . . . .	26
4.2.3	Úprava a konfigurace knihoven . . . . .	26
4.2.3.1	HardwareProfile.h . . . . .	26
4.2.3.2	TCPIPConfig.h . . . . .	26
4.2.3.3	Úprava knihoven . . . . .	26
4.2.4	Obsluha zařízení . . . . .	27
4.2.4.1	Připojení do ethernetu . . . . .	27
4.2.4.2	Fungování skriptovacího jazyka . . . . .	27
4.2.4.3	Úprava skriptů . . . . .	27
4.2.4.4	Vzdálené restartování . . . . .	27
<b>5</b>	<b>Testování</b>	<b>29</b>
5.1	Ožívování a testování periférií . . . . .	29
5.2	FAT16 . . . . .	29
5.3	Ethernet . . . . .	30
5.3.1	Přenosové vlastnosti . . . . .	30
5.4	Ostatní . . . . .	30
5.5	Ukázkový skript . . . . .	30
<b>6</b>	<b>Závěr</b>	<b>31</b>
	<b>Literatura</b>	<b>33</b>
<b>A</b>	<b>Seznam použitých zkratk</b>	<b>35</b>
<b>B</b>	<b>Referenční příručka skriptovacího jazyka</b>	<b>37</b>
B.1	Podporované datové typy, operátory a klíčová slova . . . . .	37
B.2	Podporované příkazy . . . . .	39
B.3	Ukázka skriptu . . . . .	48

<b>C Schéma zapojení a DPS</b>	<b>49</b>
C.0.1 Celkové schéma zapojení . . . . .	50
C.1 Deska plošných spojů . . . . .	51
C.1.1 Vrchní vrstva . . . . .	51
C.1.2 Spodní vrstva . . . . .	52
C.1.3 Osazení desky plošných spojů . . . . .	53
C.2 Seznam součástek . . . . .	54
<b>D Obsah příloženého CD</b>	<b>55</b>



# Seznam obrázků

1.1	Typické použití vytvořeného zařízení . . . . .	1
2.1	Embedded zařízení PC Engines ALIX 1C1 . . . . .	3
2.2	Vývojový kit Olimex PIC-MAXI-WEB . . . . .	4
2.3	Vývojový kit Web51 . . . . .	4
3.1	Blokové schéma zařízení . . . . .	7
3.2	Použití zařízení s wifi USB adaptérem . . . . .	11
3.3	Rozložení funkčních celků rozšiřujícího konektoru . . . . .	12
3.4	Sběrnice SPI . . . . .	14
3.5	Komunikace na sběrnici I2C . . . . .	15
3.6	Blokové schéma vrstev TCP/IP stacku . . . . .	16
4.1	Schéma zapojení napájení navrhovaného zařízení . . . . .	19
4.2	Schéma zapojení procesoru . . . . .	20
4.3	Schéma zapojení řadiče ethernetu . . . . .	21
4.4	Schéma zapojení konektoru ethernetu . . . . .	21
4.5	Schéma zapojení SD karty . . . . .	22
4.6	Schéma zapojení RTC . . . . .	22
4.7	Schéma zapojení rozšiřujícího konektoru . . . . .	23
4.8	Schéma zapojení konektoru pro připojení displeje . . . . .	23
4.9	Schéma zapojení sériového rozhraní . . . . .	24
4.10	MPLAB IDE . . . . .	25
5.1	Průběh příkazu ping . . . . .	30
C.1	Celkové schéma zapojení . . . . .	50
C.2	Vrchní vrstva DPS . . . . .	51
C.3	Spodní vrstva DPS . . . . .	52
C.4	Osazení DPS . . . . .	53





# Seznam tabulek

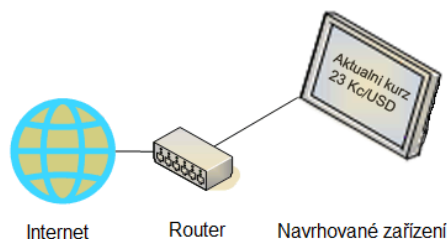
2.1	Srovnání zařízení uvedených v rešerši . . . . .	5
3.1	Maximální odběr proudu pro jednotlivé části navrhovaného systému (pro 3.3 V) . . . . .	13
3.2	Používané signály sběrnice SPI . . . . .	14
3.3	Náklady na výrobu zařízení . . . . .	17
C.1	Seznam použitých součástek a jejich hodnot . . . . .	54



# Kapitola 1

## Úvod

Oblast výpočetní techniky a informačních technologií je již nějakou dobu velmi populárním tématem. Dnešní člověk žije v digitalizovaném světě a velmi rychle si zvyká na další novinky, které mu usnadňují život. V této bakalářské práci, která aw zabývá tvorbou hardwarového zařízení pro získávání informací z internetu, se pokusím vytvořit zařízení, spočívající v návržení hardwaru a naprogramování odpovídajícího softwaru. Jako typický příklad použití takového zařízení lze uvést možnost rychlého získávání informací z internetu, jakými mohou být např. čas odjezdu vlaků, autobusů, aktuální počasí, aktuální čas nebo jiné podobné informace a jejich následné distribuování dále, typicky výpisem na LCD displej.



Obrázek 1.1: Typické použití vytvořeného zařízení  
(Zdroj vlastní)

Čtenář se dozví, na jakém základě zvolil jsem jednotlivé prvky a součástky, nutné k tvorbě vlastního zařízení.

Důkladným studiem sekundárních zdrojů v podobě různých odborných publikací a internetových pramenů bude napsána teoretická část této bakalářské práce, která je členěna následovně.

V rešerši, tedy ve druhé kapitole, se snažím o vytvoření stručného přehledu v současnosti dostupných prostředků, na jejichž základě by mohlo dojít k implementaci požadovaného zařízení.

Navazující třetí kapitola se zabývá analýzou možných řešení, kde je popsán výběr komponent pro stavbu vlastního zařízení a použití vývojového a obslužného softwaru.

V další kapitole je rozebrána konkrétní implementace a zapojení dílčích částí zařízení.

Předposlední kapitola se věnuje testování zařízení a shrnutí jeho zjištěných parametrů.

## 1.1 Specifikace cíle

Cílem této práce je navrhnout, realizovat a implementovat potřebný obslužný software do zařízení tak, aby umožňovalo stahování webových stránek a následné zobrazování vybraných informací na LCD panelu. Těmito informacemi je myšleno aktuální počasí, kurz koruny vůči dolaru apod. Informace budou stahovány ze sítě internet skrze komunikační rozhraní ethernet.

### Co bude zařízení umožňovat

- připojit se do sítě internet skrze rozhraní ethernet
- stáhnout jakýkoliv dostupný soubor<sup>1</sup> z libovolného webového serveru a uložit ho na paměťové médium
- ze staženého souboru na paměťovém médiu získat požadované informace
- připojit periferní zařízení (LCD displej) pomocí „standardizovaného“ konektoru<sup>2</sup>
- zobrazit obsah libovolného souboru na LCD displeji

Zařízení bude koncipováno jako univerzální deska, která bude dále obsahovat konektory pro USB (s možností použití jako USB host a USB device), PS2 a rozšiřující konektor pro napojení dalších periferních zařízení.

---

<sup>1</sup>V této konkrétní implementaci se jedná o HTML stránku

<sup>2</sup>Zapojení tohoto konektoru bude odpovídat zapojení v jiných bakalářských pracích, ve kterých je použit (detailnější popis konektoru je uveden v kapitole 3.1.8 )

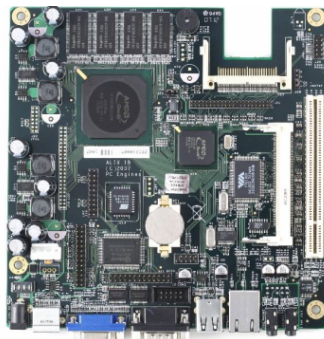
## Kapitola 2

# Rešerše dostupných zařízení

Potřebný typ zařízení se běžně přímo v požadované funkčnosti na trhu nenalzá. O danou funkčnost je potřeba doplnit jakékoliv zařízení. Následující kapitola uvádí několik variant základních zařízení.

### 2.1 PC Engines ALIX 1C1

Zařízení PC Engines ALIX 1C1[1] je embedded systém, realizovaný jednodeskovým PC, který je přímo uzpůsoben pro konkrétní potřeby zařízení. Deska (obrázek 2.1) obsahuje AMD procesor taktovaný na 500 MHz, 256 MB SDRAM, 1 linku na ethernet, 2 COM porty, sběrnici I2C a další. Cena takovýchto zařízení se pohybuje od 3 500 Kč a výše (dle konfigurace). Výhodou těchto embedded zařízení je "pohodlnost" tvorby obslužné aplikace, kde na desce běží většinou speciálně upravený operační systém Linux (existují i verze MS Windows, ale ty nejsou zdarma) a vývoj aplikace probíhá velmi podobným způsobem na obdobných nástrojích jako vývoj aplikace pro běžné PC. Hlavním nedostatkem tohoto systému je nutnost připojit jakékoliv periferní zařízení pomocí standardních rozhraní a nemožnost použít přímo vývody procesoru, což zvyšuje nároky na periferní zařízení o nutnost řídit jejich komunikaci s tímto modulem.



Obrázek 2.1: Embedded zařízení PC Engines ALIX 1C1  
(Zdroj [1])

## 2.2 Vývojový kit Olimex PIC-MAXI-WEB

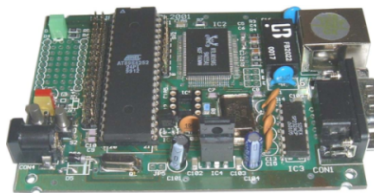
Tento vývojový kit Olimex PIC-MAXI-WEB[8] (obrázek 2.2) je postaven na mikroprocesoru PIC18F97J60 od firmy Microchip. Deska obsahuje rozhraní RS-232, LCD displej, konektor s vývody přímo na mikroprocesor pro připojení dalších periférií, a.j. Hlavní nevýhodou zmíněného kitu je malá RAM (pouze 3.8 kB), neobsažení obvodu reálného času, USB a slotu na SD kartu. Naproti tomu obsahuje dostatek volných vývodů, na které lze tyto požadované obvody napojit. Cena tohoto kitu se pohybuje kolem 2 300 Kč.



Obrázek 2.2: Vývojový kit Olimex PIC-MAXI-WEB  
(Zdroj [8])

## 2.3 Vývojový kit Web51

Jedná se o kompletní vývojový kit (obrázek 2.3) pro práci s ethernetovým rozhráním, postaveným na mikroprocesoru ATMEL AT89S8252 a ethernet řadiči RTL8019AS. Existuje pro něj přímo od vývojářů několik ukázkových firmwarů. Obsahuje komunikační rozhraní ethernet, RS-232 a 16 I/O pinů. Nevýhodou je absence LCD displeje, RTC, slotu na SD kartu a USB. Cena tohoto zařízení se pohybuje od 2 200 Kč.[4]



Obrázek 2.3: Vývojový kit Web51  
(Zdroj [4])

## 2.4 Srovnání a závěr

Z tabulky 2.1 je patrné, že žádné z navrhovaných zařízení plně nesplňuje konkrétní požadavky na funkčnost. Požadované řešení tedy nebude zrealizováno na základě výše popsaných možností, ale bude navrženo a vytvořeno „na míru“ požadované aplikaci. Tento způsob představuje i cenově nejpříjemnější variantu.

	<b>ALIX 1C1</b>	<b>PIC-MAXI-WEB</b>	<b>Web51</b>
Velikost FLASH	dle CF karty	128kB	8kB/64kB
Velikost RAM	256 MB	3.8 kB	256 B/1 kB
RS232	2x	1x	1x
USB	4x	-	-
SD/MMC slot	1x	1x	-
LCD	-	Textový 2 x 16	-
RTC	Ano	Ne	Ne
Rozšiřující I/O	-	26	16
Rozměry	170 x 170 mm	120 x 108 mm	?
Cena	3 500 Kč	2 300 Kč	od 3 500 Kč

Tabulka 2.1: Srovnání zařízení uvedených v rešerši  
(Zdroj vlastní)



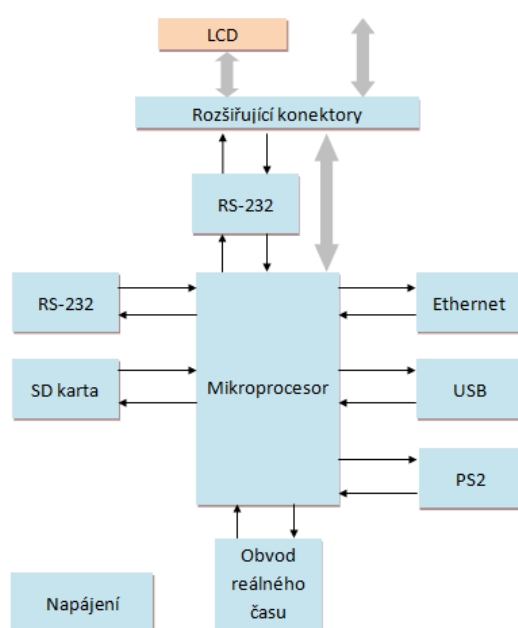


# Kapitola 3

## Analýza

V této kapitole bude čtenář seznámen s analýzou dílčích součástí, které jsou potřeba pro návrh daného zařízení. Jedná se hlavně o výběr a popis jednotlivých součástek a výběr existujících softwarových knihoven, z kterých se bude v projektu vycházet.

Zařízení se skládá z několika základních částí. Hlavní komponentou je centrální mikroprocesor komunikující s okolními periferiemi. Souhrnné blokové schéma je vidět na obrázku 3.1.



Obrázek 3.1: Blokové schéma zařízení  
(Vlastní zdroj)

Součástí softwarového vybavení jsou ovladače všech periferií, se kterými zařízení komunikuje (SD karta, Ethernet, displej, RTC...). V následujících podkapitolách budou uvedeny návrhy realizace jednotlivých bloků.

## 3.1 Hardware

### 3.1.1 Mikroprocesor

Mikroprocesor tvoří základní stavební prvek většiny systémů, nejinak tomu je i zde. Na trhu existuje mnoho výrobců mikroprocesorů. Největší a nejznámější z nich jsou Microchip[5] s procesorovými řadami PIC, Atmel[2] s procesory AVR a Texas Instrument[11] s procesory ARM. Všichni výrobci nabízejí celou škálu mikroprocesorů s různě širokou datovou sběrnici<sup>1</sup>, s různými periferiemi a obecně odlišnými charakteristikami.

Na základě dosavadních zkušeností s procesory PIC a možnosti přístupu k ladícím prostředkům této technologie, jsem se rozhodl pro použití mikroprocesoru této značky. Microchip nabízí mikroprocesory od šířky datové sběrnice 8 bitů, přes 16 bitů až po 32 bitové mikroprocesory. Jednotlivé řady mikroprocesorů mají své specifické vlastnosti, které hrají roli při výběru vhodného mikroprocesoru. Hlavní požadavky na mikroprocesor, určený do tohoto zařízení jsou

- napájení 3.3 V <sup>2</sup>
- co nejvíce požadovaných periferií přímo na čipu <sup>3</sup>
- dostatečně velká paměť programu a dat

V sekci 8-bitových procesorů se nacházejí řady PIC10, PIC12, PIC16, PIC18. První tři jsou pro tuto aplikaci nevhodné. Uplatnění těchto mikrokontrolérů se nachází v jednodušších aplikacích, kde není potřeba mnoho vnějších rozhraní a není kladen velký důraz na paměťové nároky. Některé mikroprocesory z řady PIC18 mají podporu USB 2.0 a Ethernetu 10 BaseT (nikdy ne současně). Opět je zde problém s paměťovými nároky, kde nejvybavenější mikroprocesor obsahuje pouze 3.8 kB RAM, proto je pro naši aplikaci nevhodný.

Sekce 16-bitových mikroprocesorů nabízí řady PIC24F, PIC24H, dsPIC30 a dsPIC33. Jediná řada PIC24F obsahuje mezi periferiemi podporu USB 2.0 rozhraní. Ethernetové rozhraní nenabízí žádný mikroprocesor z těchto řad. Mikroprocesory s největší pamětí mají 16 kB RAM a 256 kB Flash. Procesory řady PIC24H, dsPIC30 a dsPIC33 nemají podporu USB a proto jsou pro účely tohoto zařízení nevhodné.

V sekci 32-bitových mikroprocesorů se nalézá pouze jediná řada a to PIC32, která má podporu USB 2.0, v nejvyšší verzi má paměť programu 512 kB a paměť RAM činí 32 kB.

---

<sup>1</sup>8, 16, 32 bitů

<sup>2</sup>Více o důvodech v kapitole 3.1.10

<sup>3</sup>Aby výsledné zařízení bylo co nejmenší

V úvahu tak připadá některý mikroprocesor z řady PIC24F nebo PIC32. Byl zvolen mikroprocesor PIC24FJ256GB108, který se vyznačuje těmito vlastnostmi

- 16 MIPS, 32 MHz
- napájecí napětí 2.3 - 3.3 V (vstupní piny tolerantní 5.5 V)
- 80 pinů, z toho 68 I/O
- 256 kB Flash paměti + 12 kB bootovací paměti
- 16 kB paměti RAM
- USB 2.0 (Master, Host)
- 5 x 16b + 1 x 32 b Timer
- 4 x RS-232 modul
- 4 x SPI modul
- 4 x I2C modul
- 5 x PWM modul

Tento procesor má podporu tzv. Peripheral Pin Select, což umožňuje napojit většinu periférií (SPI, PWM, ...) na téměř libovolný I/O pin. To ve výsledku umožní optimalizovat návrh na případné jednodušší zapojení DPS.

### 3.1.2 Řadič ethernetu

Jelikož mikroprocesor neobsahuje přímo na čipu řadič ethernetu, je potřeba ho k němu připojit externě.

Ke zvolení vhodného typu ethernetového řadiče je dobré se podívat, co zařízení bude umožňovat. Jedná se o stahování webových stránek, které dosahují velikost typicky desítek až stovek kB dat. Pro přenos takové množství dat je dostačující ethernet typu 10BaseT, který dosahuje rychlosti až 10 MBit/s. V případě použití zařízení pro stahování velkých objemů dat, je tento typ ethernetu nevhodný, poněvadž by působil jako omezující prvek celého systému.

V zásadě lze říci, že na trhu se ethernet řadičů mnoho nenalézá<sup>4</sup>. Budou zde uvedeny řadiče typu ENC28J60 a RTL8019.

Řadič ethernetu ENC28J60 je produktem společnosti Microchip. Obsahuje fyzickou a linkovou vrstvu síťového modelu TCP/IP. Pracuje s plně duplexním ethernetem 10BaseT, je ale kompatibilní i s 100BaseT a 1000BaseT. Obsahuje 8 kB FIFO vyrovnávací paměti pro příchozí a odchozí data. Výhodou tohoto čipu je jeho velikost (pouzdro s 28 vývody) a softwarová podpora od výrobce<sup>5</sup>. Komunikace s tímto modulem probíhá pomocí sběrnice SPI (kapitola 3.1.11.1).

---

<sup>4</sup>Běžně dostupných v ČR

<sup>5</sup>Společnost Microchip k většině svých produktů volně nabízí knihovni funkce. Zde k tomuto řadiči nabízí TCP/IP stack (více v kapitole 3.2.1)

Řadič RTL8019 je od společnosti Realtek. Obsahuje stejné vrstvy síťového modelu a spolupracuje se stejnými typy ethernetů jako ENC28J60. Obsahuje 16 kB vyrovnávací paměti a 9 kB paměti programu pro jednoduché obslužné programy. Výhodou je větší vyrovnávací paměť. Nevýhodou je velikost čipu, který čítá 100 vývodů a způsob připojení do systému pomocí ISA sběrnice.

Pro naše zařízení je řadič RTL8019 je zbytečně velký, většina jeho pinů by zůstala nevyužita. Jeho vnitřní programovací část není třeba používat a implementace komunikace na sběrnici ISA je složitá. Naproti tomu ENC28J60 je malý, co se funkčnosti týká téměř totožný a navíc výrobce nabízí softwarovou podporu. Z výše uvedených důvodů jsem se rozhodl pro použití řadiče ENC28J60.

### 3.1.3 Paměťové úložiště

Paměťové médium bude sloužit pro ukládání libovolných dat, např.: konfigurace zařízení, stahované soubory, a.j. Nejlépe takové, které bude vyjimatelné a bude umožňovat jednoduché připojení k zařízení třetích stran (PC, čtečka paměťových médií, ...). K dispozici je několik alternativ. Jako možnost připadá realizace pomocí USB flash disku, běžného pevného disku nebo paměťové karty. U USB flash disku je zapotřebí mít realizovanou podporu USB Host pro zvládnutí komunikace. Běžný pevný disk je rovněž velmi složitý na připojení. Analýza a návrh implementace komunikace s pevným diskem by byly na samostatnou bakalářskou práci. Proto se zaměřím na paměťové médium v podobě paměťové karty.

#### 3.1.3.1 Secure Digital Card

Secure Digital Card je v současnosti nejvíce využívaný typ paměťové karty. Data se ukládají do flash paměti. Vyrábí se ve 2 variantách, SD a SDHC. Verze SD nabízí maximální kapacitu 2 GB, SDHC až 32 GB. Připojení je možné realizovat třemi způsoby

- 1 bitový SD mód
- 4 bitový SD mód
- SPI mód (pouze verze SD)

#### 3.1.3.2 Compact Flash

Compact Flash karty dosahují kapacity až 64GB a připojují se pomocí IDE rozhraní.

#### 3.1.3.3 MultiMedia Card

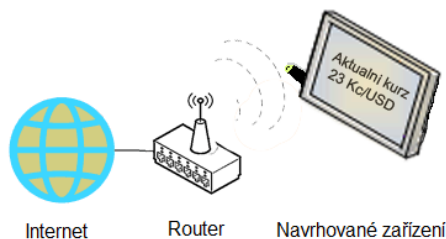
MultiMedia Card ukládá data do NAND flash. Připojení je možné realizovat 1-bitovým sériovým rozhraním nebo 8-bitovým paralelním rozhraním.

Z výše uvedených skutečností paměťová karta Compact Flash nepřipadá v úvahu pro složitou realizaci komunikačního rozhraní (podobné jako u pevného disku). Rozdíly mezi MultiMedia Card a Secure Digital Card nejsou velké a v jistém ohledu mohou být zaměnitelné (používají velmi podobný mód komunikace).

V našem zařízení bude použita Secure Digital karta, připojená pomocí SPI sběrnice (o SPI více v kapitole 3.1.11.1).

### 3.1.4 USB

USB je univerzální sériová sběrnice, která je v současné době nejpoužívanějším rozhraním pro připojování periferních zařízení k počítači. Cílem této bakalářské práce není obsluha USB. Konektor tohoto rozhraní je vyveden a připraven na budoucí možné použití. Proto zde nebude podrobně rozebírán. Mikroprocesor podporuje jak režim host, tak i device. Obrázek 3.2 uvedí příklad využití USB v této aplikaci. Místo připojení po ethernetovém kabelu by se využívalo bezdrátového připojení pomocí USB WIFI klíčenky.



Obrázek 3.2: Použití zařízení s wifi USB adaptérem  
(Vlastní zdroj)

### 3.1.5 RTC

Mikroprocesor obsahuje hardwarovou podporu reálného času (RTC), ale neumožňuje jej jednoduše zálohovat po výpadku napájení. To je nutné realizovat napájením celého mikroprocesoru a uspaním všech ostatních periférií, čímž dojde k poklesu spotřeby na minimální hodnotu. Jednodušší je připojení externího obvodu, který lze jednoduše zálohovat pomocí připojené baterie. Jako tento obvod byl zvolen DS1338-33 [3]. Zapojení bude realizováno dle doporučení v datasheetu. Obvod navíc obsahuje zálohovanou RAM o velikosti 56 B. Komunikace bude probíhat po sběrnici I2C (více o této sběrnici v kapitole 3.1.11.2).

### 3.1.6 EEPROM

Paměť EEPROM, v tomto zařízení nevyužitá, je do zařízení zahrnuta pro případné budoucí použití. Důvod navržení dalšího paměťového média, ačkoliv zařízení již obsahuje SD kartu, je prostý. Toto úložiště je pevné a nevýměnné. Typickým využitím je ukládání konfigurace zařízení. Jako rozhraní pro připojení je použito I2C (Popis rozhraní I2C je uveden v kapitole 3.1.11.2).

### 3.1.7 RS-232

Sériové rozhraní RS-232 je zde obsaženo proto, že se jedná o komunikační rozhraní, které je jednoduše nastavitelné a snadno provozovatelné. Hlavním důvodem je možnost výstupu ladících zpráv při vytváření obslužného firmwaru.

Vybraný mikroprocesor obsahuje 4 UART moduly, které mají hardwarovou podporu pro toto rozhraní. Jeden modul je použit pro výstup ladících zpráv, druhý je vyveden na

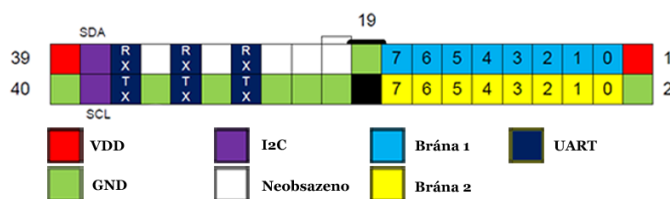
rozšiřující konektor (více v kapitole 3.1.8). Zbývající dva moduly jsou v našem případě nevyužité.

### 3.1.8 Rozšiřující konektory

Rozšiřující konektory umožňují připojení dalších periférií k tomuto zařízení. Jejich zapojení jsem realizoval po dohodě s vedoucím práce, který k těmto konektorům dodal podklady.

#### 3.1.8.1 IDE40

Na obrázku 3.3 je zobrazeno rozložení funkčních částí na hlavním rozšiřujícím konektoru. Celková funkčnost se mi podařila zachovat, a proto byl konektor implementován zcela dle požadované specifikace.



Obrázek 3.3: Rozložení funkčních celků rozšiřujícího konektoru (Vedoucí práce)

#### 3.1.8.2 Displej

Tvorba displeje není cílem této bakalářské práce, proto je zde použit displej, který je výsledkem bakalářské práce Armena Harapetjana na téma Programovatelný řadič displeje[?]. Dokumentace k tomuto displeji a komunikačnímu protokolu je uvedena v jeho práci a proto zde nebude dále rozebírána.

V případě použití jiného typu displeje, který nebude podporovat určený komunikační standard, je třeba změnit firmware zařízení.

### 3.1.9 Přepínače a tlačítka

Při návrhu je přihlíženo k tomu, aby byla vyvíjená deska co možná nejvíce konfigurovatelná. Proto se na ní nachází řada propojek, pomocí nichž lze měnit některé vlastnosti zařízení.

### 3.1.10 Napájení

Pro zařízení je třeba dvou úrovní napětí. Veškeré hardwarové součásti jsou vybrány a navrženy na 3.3 V logiku. Druhou úrovní napájení je 5 V, které je jednak vyvedeno na rozšiřující konektor a dále použito na konektoru PS/2. Důvodem je možnost připojení běžné počítačové klávesnice, která vyžaduje pro svoji činnost 5 V napájení.

Napájecí obvody musí být navrženy s ohledem na předpokládaný odběr zařízení s určitou rezervou. Počítat se musí i s napájením zařízení, připojených pomocí rozšiřujícího konektoru.

Periferie	Maximální odběr
Mikroprocesor	210 mA
Řadič ethernetu	160 mA
SD Karta	110 mA
Sériová linka	2 mA
RTC, EEPROM	10 mA
<b>CELKEM</b>	492 mA

Tabulka 3.1: Maximální odběr proudu pro jednotlivé části navrhovaného systému (pro 3.3 V)

(Zdroj vlastní)

V tabulce 3.1 jsou uvedeny jednotlivé maximální proudy pro dílčí části navrhovaného zařízení. Nikdy ale nepoběží všechny současně a ve většině případů se jedná pouze o špičkové proudy. Proto bude zdroj dimenzován na 0.5 A.

Pro napájení napěťovou úrovní 5 V je využita periferie pro připojení rozšiřujícího zařízení pomocí konektoru PS/2 (klávesnice, myš, ...). Dále je možné toto napětí připojit pomocí přepínačů na rozšiřující konektor.

Pro napájení budou využity 2 spínané zdroje MC34063A ?? . Jeden nastavený na 3.3 V, druhý na 5 V. Zapojení napájecích obvodů je znázorněno na obrázku (??). Vstupní napájecí napětí může být v rozsahu 5 V až 35 V.

### 3.1.11 Použité sběrnice

V této části práce budou uvedeny sběrnice, které se v tomto zařízení využívají.

#### 3.1.11.1 SPI

Zkratka SPI znamená Serial Peripheral Interface. Hlavní použití tohoto rozhraní je mezi řídicím systémem (typicky mikroprocesor) a ostatními obvody. Jedná se o komunikační model Master-Slave. Komunikace je synchronní a sériová po společné sběrnici.

**Master** se stará o řízení komunikace pomocí hodinového signálu a o výběr zařízení, se kterým bude komunikovat. K tomuto účelu slouží signál SS<sup>6</sup> nebo CS<sup>7</sup>, který master aktivuje nastavením do logické "0".

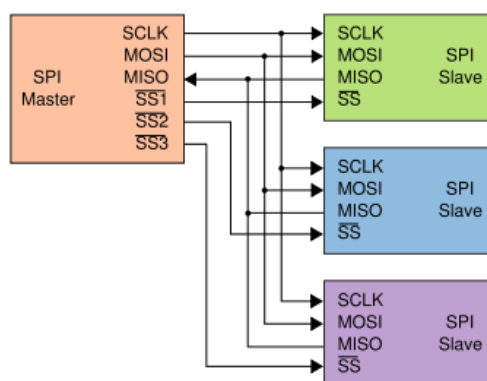
**Slave** vysílá odpověď podle hodinového signálu od mastera, pouze pokud je vybrán pomocí signálu SS nebo CS.

V tabulce 3.2 jsou uvedeny signály, které se používají na sběrnici SPI.

Signál	Popis signálu
SCK	Hodinový signál
SDO	Sériová výstupní data
SDI	Sériová vstupní data
SS, CS	Výběr slave zařízení

Tabulka 3.2: Používané signály sběrnice SPI

Na obrázku 3.4 je znázorněna sběrnice SPI se 3 slave zařízeními.



Obrázek 3.4: Sběrnice SPI  
[10]

### 3.1.11.2 I2C

I2C (Inter-Integrated Circuit) je sériová, synchronní, poloduplexní sběrnice. Z čehož vyplývá složitější vnitřní struktura připojených zařízení, protože příslušné piny musí být schopny přepínat mezi vstupním a výstupním režimem. Výběr zařízení, se kterým se bude komunikovat, se zde neprovádí pomocí zvláštních signálů jako u SPI, protože každé zařízení na této sběrnici má svoji vlastní jednoznačnou adresu a výběr zařízení je realizován přesně stanoveným komunikačním protokolem.

Veškeré řízení komunikace na sběrnici má na starosti zařízení typu Master. V jednom okamžiku může být masterem pouze jedno zařízení. Ostatní zařízení jsou typu slave, které

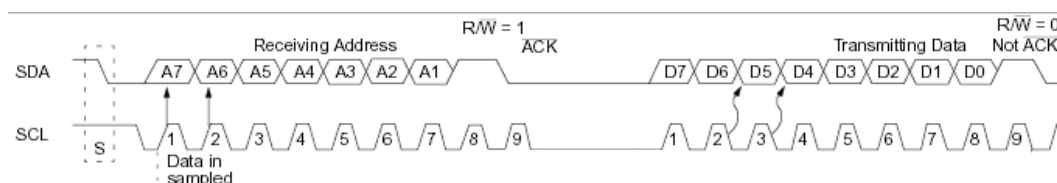
<sup>6</sup>Slave select

<sup>7</sup>Chip select



nemohou řídit komunikaci a ani nemohou inicializovat komunikaci. Přenos dat probíhá po dvou vodičích, datovém SDA a hodinovém SDL.

Po zachycení podmínky START porovnávají všechny obvody svou adresu s adresou s adresou vysílanou na sběrnici. Zjistí-li některý z obvodů shodu, je vysílání určeno právě jemu a musí přijetí adresy potvrdit bitem ACK. Potom přijímá nebo vysílá další data. Následující obrázek 3.5 znázorňuje komunikace na sběrnici I2C. [12]



Obrázek 3.5: Komunikace na sběrnici I2C  
[10]

## 3.2 Firmware

### 3.2.1 TCP/IP Stack

Realizovat jednotlivé vrstvy TCP/IP by bylo velmi časově náročné, a proto jsem se rozhodl pro využití z některé z knihoven třetích stran. Při hledání vhodné knihovny připadaly v úvahu následující balíky, které splňovaly požadavek na minimální velikost přeloženého programu a minimální obsazení RAM.

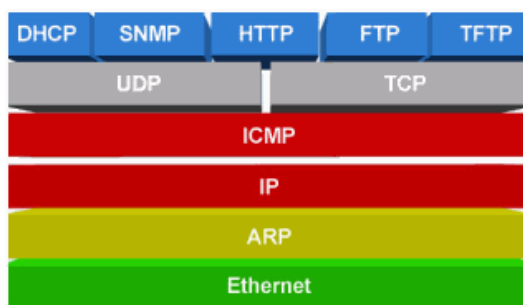
#### 3.2.1.1 Microchip TCP/IP stack

Komplexní řešení TCP/IP stacku od firmy Microchip je optimalizované na většinu procesorových řad tohoto výrobce. Zásobník je modulární, je napsán v programovacím jazyku C. Efektivní implementace může být provedena ve zhruba 28 až 34 kB kódu, v závislosti na použitých modulech. Podporuje řadiče ENC28J60, ENC624J600 a ZeroG ZG2100M. Na obrázku 3.6 jsou znázorněny vrstvy tohoto TCP/IP stacku.

#### 3.2.1.2 uIP

Open zdrojový software uIP je napsán v programovacím jazyce C. Dokumentace a zdrojový kód je volně k dispozici. Je portován na širokou škálu mikroprocesorů a používán v mnoha embedded produktech. Podporuje řadič ethernetu Realtek 8019AS.

Na základě zvoleného řadiče ethernetu bylo vybráno řešení TCP/IP stacku od stejného výrobce, který tuto knihovnu neustále vyvíjí.



Obrázek 3.6: Blokové schéma vrstev TCP/IP stacku  
[?]

### 3.2.2 Souborový systém

Souborovým systémem se označuje způsob organizace dat ve formě souborů a adresářů tak, aby k nim byla možnost snadného přístupu. Existuje mnoho typů souborových systémů, které jsou odvozené v závislosti na systému, na kterém jsou provozovány (FAT, NTFS, UFS, ...).

Výrobce použitého mikroprocesoru nabízí vlastní implementaci souborového systému FAT12, FAT16 a FAT32 v podobě knihovny MDD File System. Tato knihovna realizuje rozhraní mezi hardwarovou implementací a souborovým systémem. Knihovna poskytuje metody pro čtení a zápis dat pomocí mikroprocesoru.

Omezením této knihovny je maximální velikost souboru na 2 GB, maximální délka souboru stanovená na 8 znaků. Tento balík je koncipován jako modulární knihovna, ve které lze jednoduše pomocí direktiv v programu při překladu aktivovat či deaktivovat jednotlivé funkční moduly.

### 3.3 Náklady na výrobu

V následující tabulce je uveden rozpis cen za jednotlivé funkční bloky desky. Ceny jsou použity maloobchodní z internetových obchodů [www.gme.cz](http://www.gme.cz) a [www.tme.cz](http://www.tme.cz).

Součást	Cena při nákladu 1 kus [Kč]	Cena při nákladu 100 kusů [Kč]
Výroba DPS	520	210
Základní osazení <sup>1</sup>	235	175
Napájení 5V	28	24
Ethernet komunikace	180	120
SD karta	27	21
USB	20	11
PS2	13	10
RS232	67	44
RTC	75	57
Rozšiřující konektor	38	29
<b>Výsledná cena</b>	<b>1203</b>	<b>701</b>

Tabulka 3.3: Náklady na výrobu zařízení

Při výrobě prototypu této desky byla použita jiná technologie výroby, při které nebyly zhotoveny filmové podklady pro výrobu. Cena výroby DPS uvedená v tabulce počítá s již připravenými filmovými podklady. Vytvoření filmových podkladů je jednorázová investice a slouží právě pro levnější opakovatelnost výroby.

V porovnání s cenami zařízení uvedených v rešerši (kapitola 2), se navržené zařízení dostalo cenou nejnižší a to o více než polovinu (V ceně zařízení nejsou promítnuty náklady za čas strávený při vývoji zařízení).

<sup>1</sup>Napájení 3.3 V, procesor a součástky kolem něho



# Kapitola 4

## Realizace

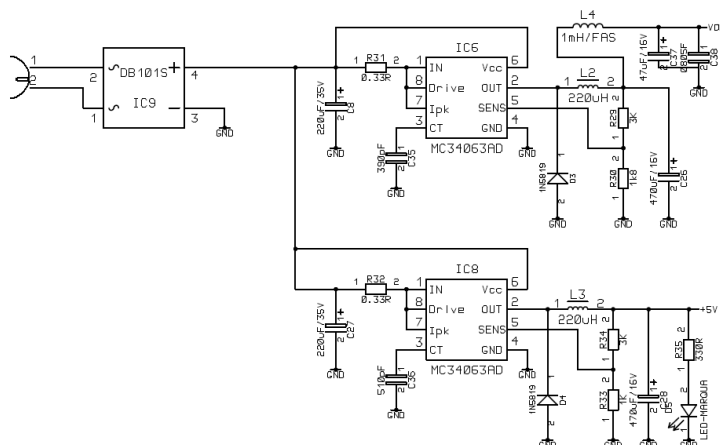
Kapitola realizace se zabývá zapojením jednotlivých, v analýze uvedených, součástí zařízení, konfigurací použitých knihoven a vývojem vlastního obslužného firmwaru.

### 4.1 Hardware

Tato část práce se věnuje problematice zapojení funkčních bloků po hardwarové stránce.

#### 4.1.1 Napájení

Jak již bylo uvedeno v kapitole 3.1.10, bude použito dvou spínaných zdrojů. Na obrázku 4.1 je znázorněno konkrétní zapojení. Vstup je připojen přes usměrňovací můstek pro ochranu proti přepólování vstupního napětí. Zapojení obou větví je téměř totožné, pouze jsou jiné hodnoty zpětnovazebních odporů, pomocí nichž se určuje výstupní napětí. Na výstupu napájení 3.3 V je navíc přidán filtrační LC prvek pro vyhlazení signálu.



Obrázek 4.1: Schéma zapojení napájení navrhovaného zařízení (Zdroj vlastní)

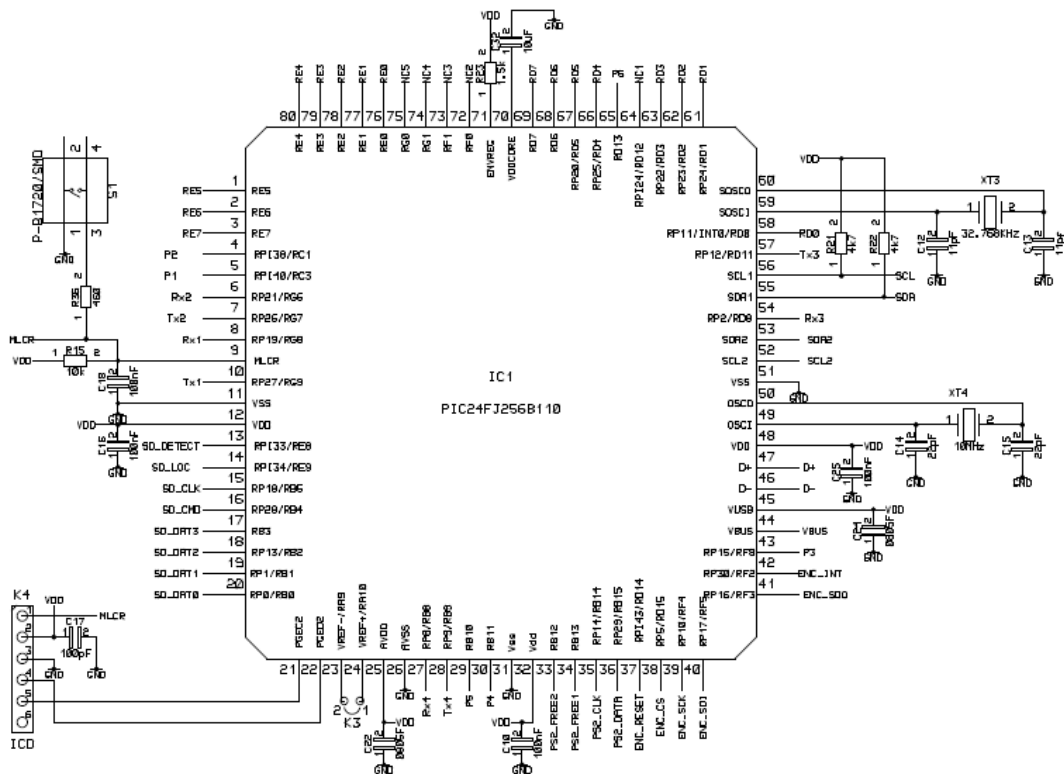
### 4.1.2 Mikroprocesor

Na mikroprocesor jsou připojeny všechny rozšiřující periferie. Většina signálů na obrázku 4.2 je popsána symbolickou adresou, která se rovněž nachází u dané periferie.

Ačkoliv procesor obsahuje vlastní zdroj hodinového signálu, je zde připojen externí oscilátor, který lze nataktovat vnitřními obvody až na 32 MHz. Naproti tomu vnitřní oscilátor může pracovat maximálně na frekvenci 8 Mhz.

Ke všem napájecím vstupům jsou připojeny boostovací kondenzátory, které zajistí přísun dostatečného množství proudu v případě krátkodobého špičkového odběru.

Na konektor K4 jsou vyvedeny programovací vstupy mikroprocesoru.

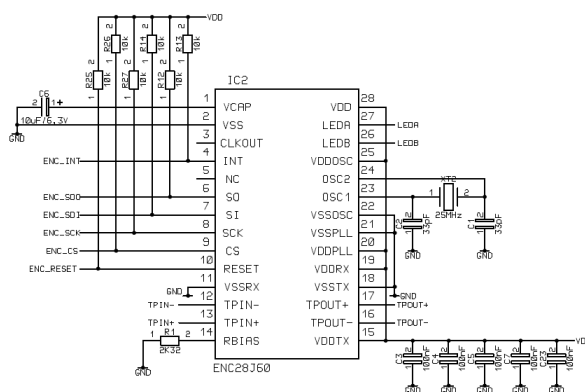


Obrázek 4.2: Schéma zapojení procesoru  
(Zdroj vlastní)

### 4.1.3 Ethernet

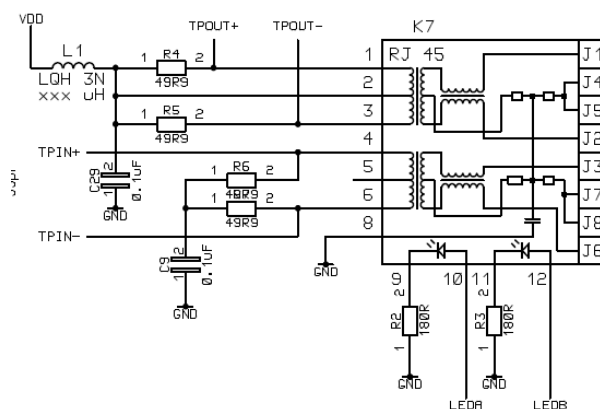
Připojení ethernetového rozhraní k mikroprocesoru (obrázek 4.3) je realizováno pomocí SPI sběrnice (kapitola 3.1.11.1) po vodičích ENC\_SCK, ENC\_SDI, ENC\_SDO. Navíc je z mikroprocesoru do řadiče ethernetu veden signál ENC\_RESET, umožňující reset řadiče.

Zapojení samotného řadiče je provedeno dle specifikace vyžadované v datasheetu. Zvedací odpory na komunikaci mezi řadičem a mikroprocesorem se později ukázaly jako přebytečné, protože je mikroprocesor má v sobě obsažené.



Obrázek 4.3: Schéma zapojení řadiče ethernetu  
(Zdroj vlastní)

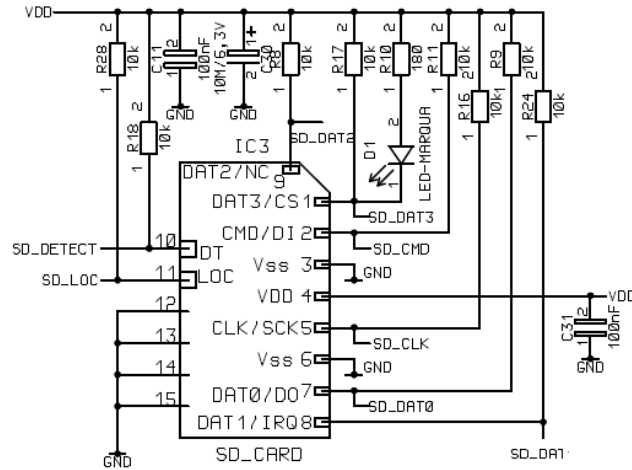
Konektor pro ethernetové rozhraní byl použit MIC24311-0101T-LF3 od firmy Midcom[6], který v sobě obsahuje oddělovací transformátory. Tím pádem bylo možné celé připojení konektoru, které je znázorněné na obrázku 4.4, k řadiči zjednodušit. Zapojení je provedeno dle doporučení v datasheetu.



Obrázek 4.4: Schéma zapojení konektoru ethernetu  
(Zdroj vlastní)

#### 4.1.4 SD karta

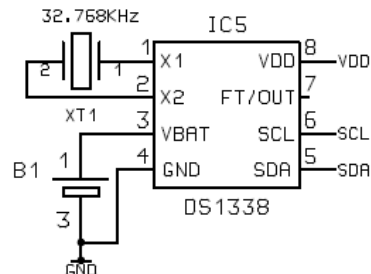
SD karta je připojena pomocí SPI rozhraní. Na signálový vodič SD\_DAT3, který aktivuje tuto periférii pro komunikaci po SPI rozhraní, je dále připojena paralelně signalizační LED dioda. Ta zobrazuje průběh přístupu k SD kartě. Signál SD\_DETECT zjišťuje, zda-li je SD karta zasunuta ve slotu. Signál SD\_LOC detekuje zamknutí SD karty. Ostatní signálové vodiče jsou použity na SPI rozhraní.



Obrázek 4.5: Schéma zapojení SD karty  
(Zdroj vlastní)

#### 4.1.5 RTC

Obvod reálného času je připojen k mikroprocesoru pomocí sběrnici I2C (kapitola 3.1.11.2). Na obrázku 4.6 je zobrazeno konkrétní zapojení obvodu DS1338. K obvodu je nutné připojit vnější oscilátor na 32.768 kHz a baterii pro zálohování pro případ výpadku napájení. Výstup FT/OUT slouží ke generování výstupního hodinového signálu (1 Hz / 4.096 kHz / 8.192 kHz / 32.768 kHz), který ale v této aplikaci nebyl využit.

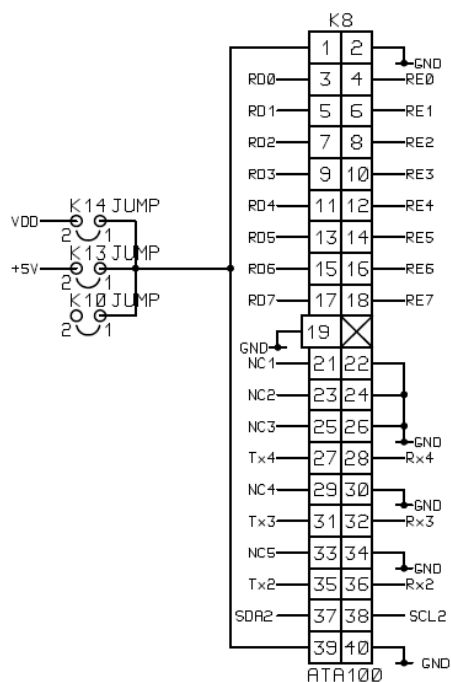


Obrázek 4.6: Schéma zapojení RTC  
(Zdroj vlastní)



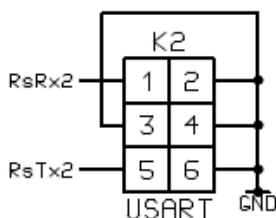
### 4.1.6 Rozšiřující konektory

Na obrázku 4.7 je zobrazeno fyzické zapojení rozšiřujícího konektoru IDE40. Povedlo se připojit bránu 1 a 2 na celý port procesoru, což bude umožňovat jednoduchou správu těchto bran. Ostatní periferie tohoto konektoru jsou připojeny na univerzální I/O piny procesoru, které umožňují volbu použité periferie (Peripheral Pin Select).



Obrázek 4.7: Schéma zapojení rozšiřujícího konektoru  
(Vlastní zdroj)

Další rozšiřující částí je konektor pro připojení displeje. Jeho zapojení je zobrazeno na obrázku 4.8. Signály RsRx2 a RsTx2 jsou použity pro přenos po RS-232. Jejich napěťová úroveň bude rozebrána v kapitolách 4.1.7 a 4.1.8.



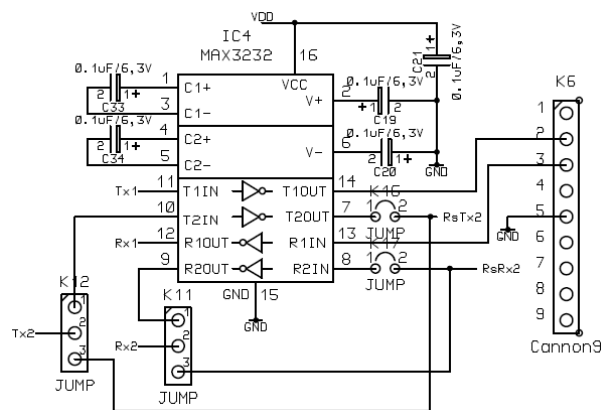
Obrázek 4.8: Schéma zapojení konektoru pro připojení displeje  
(Vlastní zdroj)

Zbývající volné piny procesoru jsou vyvedeny na konektor K15, pro možné budoucí použití. Pro případ použití některého vstupu jako analogový (v tomto zařízení se využí-

vají všechny digitálně) je vyveden na konektor K3 vstup referenčního napětí. Ten není rovněž v současnosti použit.

#### 4.1.7 RS-232

Připojení rozhraní RS-232 je realizováno pomocí konvertoru sběrnice MAX3232, který je zapojen dle specifikace uvedené v datasheetu. Tento konvertor obsahuje 2 obousměrné budiče mezi úrovní TTL a UART. Jeho zapojení je znázorněno na obrázku 4.9.



Obrázek 4.9: Schéma zapojení sériového rozhraní  
(Zdroj vlastní)

#### 4.1.8 Přepínače a propojky

Na obrázku 4.9 jsou použity přepínače K11, K12, K16 a K17. Pomocí těchto propojek se určuje úroveň logických hodnot napětí na rozšiřujícím konektoru pro displej. Lze zvolit mezi úrovněmi pro rozhraní RS-232 (-3 až -15 V pro logickou "1" a 3 až 15 V pro logickou "0") nebo úrovní TTL.

Na obrázku 4.7 jsou k volbě připojeného napájecího napětí použity propojky K10, K13 a K14. Možno je zvolit napájení 3.3 V, 5 V nebo žádné napájení (napájení je fyzicky odpojeno).

Pro provedení resetu mikroprocesoru (tudíž i celého zařízení - protože dojde k reinitializaci veškerých periférií) slouží tlačítko S1. Jeho zapojení je dle doporučení od výrobce.

#### 4.1.9 Návrh schématu a DPS

Návrh schématu a DPS byl proveden v programu Rangel XL od společnosti Seetrax[9]. Plošný spoj je proveden jako dvouvrstvý. Spodní vrstva je převážně použita jako podzemění, z důvodu EMC. Veškeré signály jsou vedeny v horní vrstvě, pouze v případě křížení jsou taženy ve vrstvě spodní.

### 4.1.10 Chyby a problémy při řešení

V průběhu vývoje se objevilo několik nedostatků při návrhu schématu zapojení. Nejzásadnějším byl otočený konektor K6 pro linku RS-232. Tento problém byl vyřešen proškrcnutím spojení na DPS a vyvedením propojky. Dalším problémem byla otočená polarita kondenzátoru C34.

Při samotném vývoji obslužného firmwaru nastal největší problém při odhalování chyby v komunikaci mezi řídicím procesorem a radičem ethernetu. Řešení tohoto problému bylo po několika týdnech zkoumání, přepajování objeven. Chyba byla v mikroprocesoru, který při určitém nastavení SPI sběrnice pracoval nekorektně a komunikace na tomto rozhraní byla v některých okamžicích nesprávná. Tento problém je popsán v revizi datasheetu mikroprocesoru.

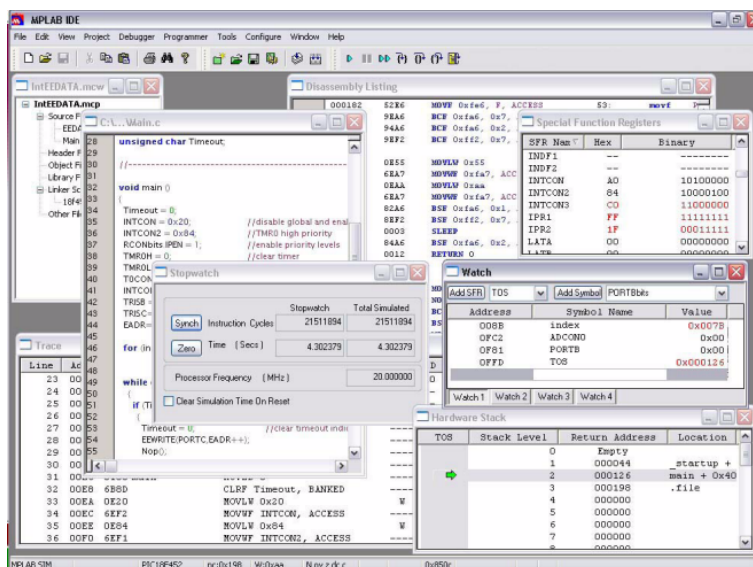
Veškeré výše uvedené chyby jsou již ve zdrojových podkladech opravené.

## 4.2 Software

Obsahem této sekce je popis softwarové části vyvíjeného zařízení.

### 4.2.1 Vývojové prostředí MPLAB IDE

Toto vývojové prostředí je zdarma dodáváno firmou Microchip. Umožňuje integrovat překladače na různé typy mikroprocesorů a poskytuje větší komfort při vývoji aplikací. Současně obsahuje nástroje pro programování, simulaci a ladění aplikací<sup>1</sup>.



Obrázek 4.10: MPLAB IDE  
(Zdroj [7])

<sup>1</sup>Ladění aplikací v reálném běhu je možné pouze za použití debuggeru - např. ICD2, ICD3 nebo Real-ICE

Při vývoji firmwaru jsem měl možnost vyzkoušet a použít debugery ICD2, ICD3 a Real-ICE. Nejnovější z výše jmenovaných ICD3, oproti svým předchůdcům, disponoval rychlejším nahráváním firmwaru do procesoru a lepší ladící funkcí.

### 4.2.2 Knihovny

Základními kameny celého obslužného softwaru jsou knihovny dodávané výrobcem mikroprocesoru. Konkrétně se jedná o knihovnu TCP/IP Stacku a MDD File System. Použití těchto knihoven umožňuje soustředění pozornosti pouze na samotnou realizaci funkčních částí obslužného firmwaru.

### 4.2.3 Úprava a konfigurace knihoven

Knihovny dodané výrobcem mikroprocesu je nutné nakonfigurovat. Konfigurací se myslí aktivace či deaktivace jednotlivých bloků knihoven, mapování na porty procesoru a nastavení ostatních parametrů (např. frekvence mikroprocesoru apod.)

#### 4.2.3.1 HardwareProfile.h

Prvním krokem je třeba nastavení konfiguračního souboru `HardwareProfile.h`, ve kterém je definováno rozhraní mezi knihovnamy a fyzickým hardwarovým zapojením. Jedná se hlavně o nastavení SPI pro řadič ethernetu, SPI pro SD kartu a rozhraní UART.

#### 4.2.3.2 TCPIPConfig.h

V souboru `TCPIPConfig.h` jsou zdefinovány možnosti nastavení TCP/IP Stacku, mezi nimiž je (de)aktivace jednotlivých modulů, které bude TCPIP Stack podporovat, nastavení velikosti bufferů, nastavení sítě, apod.

#### 4.2.3.3 Úprava knihoven

V době, kdy vznikal obslužný firmware tohoto zařízení, neexistovalo propojení knihoven TCP/IP Stacku a MDD File System a proto bylo nutné si provázání obou knihoven dopsat.

Provázal tyto 2 knihovny na úrovni FTP protokolu, kde v případě původní knihovny byl implementován pouze příkaz GET pro nahrání souboru do EEPROM. Tento protokol jsem rozšířil o další příkazy (LIST, PWD, CWD, MKD, RMD, DELE, RETR) čímž jsem vytvořil téměř plnohodnotný FTP server.

Dále jsem vytvořil obecný program pro stažení libovolného souboru z internetu, který jsem zakomponoval do použitých knihoven.

V současné době (cca 3 měsíce od napsání obslužného firmwaru) již výrobce těchto knihoven implementuje mnou použité knihovny v jednom balíčku a částečně již zrealizoval jejich propojení. FTP server není dostupný na takové úrovni, v jaké je naimplementován v této práci.

#### 4.2.4 Obsluha zařízení

Do zařízení jsem naimplementoval vlastní jednoduchý skriptovací jazyk, pomocí něhož lze ovlivňovat chování daného zařízení. Samotný skript musí být uložen v souboru na SD kartě v adresáři `/scripts/boot/`.

##### 4.2.4.1 Připojení do ethernetu

Po připojení zařízení do ethernetu se nejprve snaží získat IP adresu z DHCP serveru. Pokud se přiřazení IP adresy nezdaří, zařízení si nastaví defaultní adresu 10.0.0.120 a masku sítě 255.255.255.0. Defaultní bránu v tomto případě bude mít nastavenou na 10.0.0.1

##### 4.2.4.2 Fungování skriptovacího jazyka

Standardně po zapnutí zařízení se systém pokusí najít implicitní skript, který je spouštěn po startu. Ten musí být uveden v souboru `/scripts/boot/init.src`. Pokud systém tento soubor nenalezne, neproběhne spuštění žádného skriptu (pro opětovné spuštění je zapotřebí zařízení restartovat).

Po spuštění a nalezení implicitního souboru se skriptem si systém vytvoří jeho kopii do adresáře `/script/run/`. Tím je možné měnit konfiguraci startovacího skriptu za běhu (pro provedení změn je vždy potřeba provést restart zařízení).

Detailní popis jednotlivých funkcí, které implementuje tento skriptovací jazyk, je uveden v příloze [B](#).

##### 4.2.4.3 Úprava skriptů

Úpravu skriptů, které se budou v zařízení spouštět, je možné provést 2 způsoby. Prvním je vyjmutí SD karty ze zařízení a nahrání nového skriptu pomocí počítače. Dalším způsobem je připojení se k zařízení pomocí vestavěného FTP serveru a nahrání nového skriptu do zařízení či přímo jeho online úprava.

##### 4.2.4.4 Vzdálené restartování

Z předchozí kapitoly [4.2.4.3](#) již víme, jak do zařízení nahrát nový skript či upravit stávající. Pokud se tato změna provádí pomocí FTP přenosu, je zapotřebí pro nové načtení takto upravených skriptů provést restart zařízení. Proto jsem částečně využil implementaci HTTP serveru, na kterém běží stránka, pomocí níž je možné provést restart zařízení na dálku.



# Kapitola 5

## Testování

Testování celého zařízení probíhalo průběžně s vývojem v několika fázích. V této kapitole bude zmíněno, jak probíhaly testy jednotlivých částí.

### 5.1 Oživování a testování periferií

Při sestavování a oživování zařízení byly postupně připojovány další periferie, které byly vždy samostatně otestované, zda-li splňují očekávané předpoklady.

Otestování zdroje proběhlo měřením zvlnění napětí 3.3 V, které se pohybovalo v rozsahu 19 mV (špička - špička). Odběr zařízení při neaktivním vysílání se pohyboval kolem 55 mA, při aktivním vysílání ethernetového rozhraní a aktivní SD kartě byl odběr 210 mA.

### 5.2 FAT16

Na SD kartě, která byla pomocí běžné čtečky karet připojena k počítači, jsem vytvořil náhodnou adresářovou strukturu včetně různě velikých souborů. Takto vytvořený souborový systém se mi pomocí zařízení dařilo procházet a ze souborů číst či do nich zapisovat.

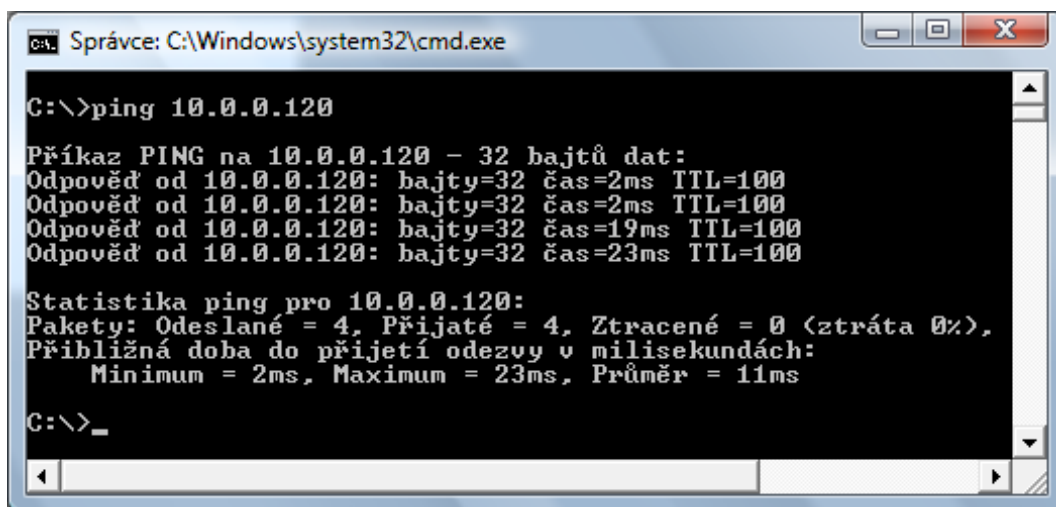
Opačným směrem jsem prováděl obdobné testování, kdy v zařízení byl vytvořen (smazán) souborový systém, který se po přečtení v počítači shodoval.

## 5.3 Ethernet

Při odlaďování a testování ethernetového rozhraní jsem používal program Wireshark[13], který dokáže odchyťvat komunikaci na určeném síťovém rozhraní.

### 5.3.1 Přenosové vlastnosti

Testování odezvy ethernetového rozhraní jsem provedl pomocí příkazu ping. Jeho průběh je znázorněn na obrázku 5.1.



```
ca. Správce: C:\Windows\system32\cmd.exe
C:\>ping 10.0.0.120
Příkaz PING na 10.0.0.120 - 32 bajtů dat:
Odpověď od 10.0.0.120: bajty=32 čas=2ms TTL=100
Odpověď od 10.0.0.120: bajty=32 čas=2ms TTL=100
Odpověď od 10.0.0.120: bajty=32 čas=19ms TTL=100
Odpověď od 10.0.0.120: bajty=32 čas=23ms TTL=100

Statistika ping pro 10.0.0.120:
Pakety: Odeslané = 4, Přijaté = 4, Ztracené = 0 (ztráta 0%),
Přibližná doba do přijetí odezvy v milisekundách:
    Minimum = 2ms, Maximum = 23ms, Průměr = 11ms
C:\>_
```

Obrázek 5.1: Průběh příkazu ping  
(Zdroj vlastní)

Rychlost přenosu dat jsem otestoval na vestavěném FTP serveru. Při přenosu dat z PC do zařízení byla průběrná přenosová rychlost 19 KB/s. Při opačném směru přenosu se rychlost pohybovala v průměru na 27 KB/s.

## 5.4 Ostatní

Rozhraní USB, PS/2 jsem nijak netestoval, protože nebyly náplní této práce.

## 5.5 Ukázkový skript

Do zařízení jsem vytvářel ukázkový skript (je přiložen na CD), který zobrazuje na LCD panelu jednak aktuální čas a dále čas odjezdu nejbližšího spojení mezi stanicemi Karlovo Náměstí a Dejvická. Tyto údaje se v čase aktuálně mění.



# Kapitola 6

## Závěr

Výsledkem této bakalářské práce je plně funkční zařízení, které splňuje veškeré předem stanovené požadavky na funkčnost.

Řešením bylo vytvoření hardwarového přípravku, který narozdíl od zařízení uvedených v rešerši (kapitola 2) obsahuje veškeré potřebné periferie. Do takto navrženého a vyrobeného zařízení jsem naimplementoval obslužný firmware, který dokáže na základě skriptu, uloženém na SD kartě, ovlivňovat chování zařízení. Tímto odpadá přeprogramování firmwaru v případě změny požadavku na chování zařízení.

Rychlost a odezva ethernetového rozhraní se pohybovala v očekávaných hodnotách.

V průběhu práce se objevovaly nové skutečnosti, které by v současnosti vedly k použití jiných dílčích částí tohoto zařízení. Mezi nejvýraznější patří nový ethernetový řadič ENC624J600, který vyvinula a uvolnila do prodeje společnost Microchip asi 2 měsíce po vytvoření tohoto zařízení. Tento řadič disponuje ethernetovým typem 10/100 Base-T. Nebo ještě v pozdější době nový mikroprocesor řady PIC32, který již v sobě má zabudované rozhraní ethernetu a má podporu USB 2.0 a přitom disponuje dostatečnou kapacitou paměti programu a RAM.

Na tuto práci lze navázat mnoho dalších projektů díky zachování kompatibility rozšiřujících konektorů a díky obsazení mnoho periferních obvodů.

Tato práce je pro mne velkým přínosem. Měl jsem možnost pracovat na hardwarovém projektu již od samotného návrhu schématu přes osazování až po realizaci obslužného firmwaru.



# Literatura

- [1] PC Engines — Product file.  
<http://www.pcengines.ch/alix1c.htm>.
- [2] Atmel — Oficiální stránky, 2010.  
<http://www.atmel.com/>.
- [3] DS1338 — Datasheet.  
[http://www.maxim-ic.com/quick\\_view2.cfm/qv\\_pk/3696](http://www.maxim-ic.com/quick_view2.cfm/qv_pk/3696).
- [4] WEB51 — stránky projektu.  
<http://web51.hw.cz/>.
- [5] Microchip — Oficiální stránky, 2010.  
<http://www.microchip.com/>.
- [6] Midcom — Homepage, 2010.  
<http://www.midcom-inc.com/>.
- [7] MPLAB IDE — Product file, 2010.  
<http://www.microchip.com/>.
- [8] Olimex — PIC-MAXI-WEB.  
<http://www.olimex.com/dev/pic-maxi-web.html>.
- [9] Texas Instrument — Oficiální stránky, 2010.  
<http://seetrax.com/>.
- [10] Root.cz — Externí sériová sběrnice SPI a I2C, 2008.  
<http://www.root.cz/clanky/externi-seriove-sbernice-spi-a-i2c/>.
- [11] Texas Instrument — Oficiální stránky, 2010.  
<http://www.ti.com/>.
- [12] Wikipedia — I2C, 2010.  
<http://cs.wikipedia.org/wiki/I2C>.
- [13] Wireshark — Go deep, 2010.  
<http://www.wireshark.org/>.



# Příloha A

## Seznam použitých zkratek

**A/D** Analog/Digital

**ARP** Address Resolution Protocol

**CLK** Clock signal

**CS** Chip Select

**DHCP** Dynamic Host Configuration Protocol

**DPS** Deska plošných spojů

**EMC** Electromagnetic Compatibility

**FAT** File Allocation Table

**FTP** File Transfer Protocol

**HTTP** Hyper-Text Transfer Protocol

**I2C** Inter-Integrated Circuit

**ICMP** Internet Control Message Protocol

**IDE** Integrated Development Environment

**I/O** Input/Output

**LED** Light Emitting Diode

**MCU** Microcontroller Unit, mikroprocesor

**PS/2** Konektor MINI DIN6, dříve používaný pro připojení počítačové klávesnice nebo myši

**RAM** Random Access Memory

**RTC** Real-time clock

**SD** Secure Digital

**SPI** Serial Peripheral Interface

**TCP/IP** Transmission Control Protocol / Internet Protocol

**UDP** User Datagram Protocol

**USB** Universal Serial Bus

## Příloha B

# Referenční příručka skriptovacího jazyka

Pro nastavení chování zařízení je navržen vlastní skriptovací jazyk, který umožňuje ovládání základních periferních zařízení a dovoluje jejich řízení na vyšší vrstvě bez nutnosti zásahu do firmwaru zařízení. Jazyk obsahuje sadu instrukcí pro ovládání sériové linky, komunikaci s HTTP serverem, komunikaci s LCD displejem, soubory uloženými na SD kartě, atd. Aby bylo možné napsat příslušný skript, je třeba uchovávat dílčí výsledky. K tomu je v programu implementována sada proměnných různých datových typů.

### B.1 Podporované datové typy, operátory a klíčová slova

#### Číslo

Zapsané přímo jako celé číslo (i záporné) (např. 10).

#### Text

Zapsané mezi uvozovkami (např. "ukázkový text").

#### Registr

Zapsané ve tvaru #číslo\_registru (např. #3). Počet registrů je omezen na počet 10.

#### Bit registru

Určuje jeden konkrétní bit v registru. Zápis ve tvaru #registr.bit (například #3.1).

#### Buffer

Má význam textového řetězce uloženém v paměti procesoru. Jeho délka je omezená kapacitou paměti RAM a její aktuální obsazení. Zápis ve tvaru @buffer (např. @2). Počet bufferů je omezen na počet 10.

## Soubor

Učtuje název či obsah souboru. Vyjadřuje se vždy absolutní cestou. Zápis ve tvaru ?cesta\_k\_souboru (např. ?data.txt)

## Port

Adresa výstupního portu. Zápis ve tvaru oznaceni\_portu (např. RA - port A). Porty jsou dostupné a označené podle zapojení ve schématu (RA až RG).

## Bit Portu

Adresa pouze jednoho bitu portu. Zápis oznaceni\_portu.bi\_portu (např. RA.2)

## Datum a čas

Určuje proměnou typu datumu či času. Zápis \$proměnná. Dostupné jsou tyto

- \$H - hodina
- \$M - minuta
- \$S - sekunda
- \$D - den
- \$N - měsíc
- \$Y - rok

## Aritmeticko-logické operace

- "+"součet
- "-"rozdíl
- "\*"násobení
- "/"dělení
- "%"zbytek po dělení
- "&"logický součin
- "|"logický součet
- ">>"logický posun vpravo
- "<<"logický posun vlevo



### Srovnávací operandy

- <"menší
- >"větší
- ="rovno
- <="menší rovno
- >="větší rovno
- "!="různé

### Speciální klíčová slova

- CONNECTED
- NOTCONNECTED
- INPUT
- OUTPUT
- BLOCKED
- NOTBLOCKED

## B.2 Podporované příkazy

### Append

Připojí na konec vstupu požadovanou hodnotu.

**Append(výstup,hodnota);**

- výstup
  - cíl, ke kterému se bude požadovaná hodnota připojovat
  - soubor, buffer
- hodnota
  - hodnota, která se bude připojovat
  - soubor, buffer, řetězec, registr, bit registru, port, bit portu, rok, měsíc, den, hodina, minuta, sekunda

## Arith

Matematické operace.

**Arith(výsledný registr, operace, operand, operand);**

- výsledný registr
  - registr, do kterého bude uložen výsledek
  - registr
- operace
  - označení požadované operace
  - sčítání, odčítání, celočíselné dělení, násobení, zbytek po dělení, logický součet, logický součin, logický posun
- operand
  - operand pro aritmetickou operaci
  - číslo, registr

## Call

Spuštění podprogramu.

**Call(návěští);**

- návěští
  - označení podprogramu, který se bude spouštět
  - návěští

## ClearBuffer

Vyprázdnění bufferu. Pro uvolnění zabrané paměti.

**ClearBuffer(buffer);**

- buffer
  - buffer, který bude vyprázdněn a jeho paměť bude dealokována
  - buffer

## ClrLCD

Vymaže celý displej a nastaví kurzor na počátek.

**ClrLCD();**

## ConfigPort

Nastaví port jako vstup nebo výstup.

**ConfigPort(port, typ);**

- port
  - konkrétní port (pouze jeden bit)
  - bit portu
- typ
  - způsob nastavení portu
  - INPUT, OUTPUT

## DeleteFile

Smaže soubor ze souborového systému

**Deletefile(cesta);**

- cesta
  - absolutní cesta souboru, který chcete smazat
  - soubor

## DownloadFile

Umožňuje stažení ze vzdáleného serveru soubor a jeho uložení na SD kartu.

**DownloadFile(adresa serveru, port, vzdálený soubor, cílový soubor, výsledný registr);**

- adresa serveru
  - jmená nebo IP adresa serveru (např. google.com)
  - řetězec, buffer
- port
  - číslo portu, na kterém běží server (typicky 80)
  - číslo, registr
- vzdálený soubor
  - konkrétní soubor ze serveru (zadaná celá cesta), případně hlavička odesílaného protokolu (např. GET google.com HTTP/1.0 ...)

- řetězec, buffer, soubor
- cílový soubor
  - cesta, kam se bude soubor ukládat na SD kartu
  - soubor
- výsledný registr
  - návratová hodnota po ukončení stahování, která nabývá těchto hodnot
    - \* "0" pokud vše proběhlo v pořádku
    - \* "1" pokud vypšel časový limit
    - \* "2" pokud nastala při stahování chyba
  - registr

### **EndProgram**

Ukončení vykonávání všech skriptů.

**EndProgram();**

### **FileExists**

Zjištění, zda-li soubor existuje.

**FileExists(cesta);**

- cesta
  - cesta k souboru, u kterého chceme zjišťovat, zda-li existuje
  - soubor

### **GetDateTime**

Aktualizace datumu a času v datumových registrech.

**GetDateTime();**

### **GetLength**

Vrací délku řetězce nebo velikost souboru.

**GetLength(návratový registr, argument);**

- návratový registr

- registr, do které bude uložen výsledek
- registr
- argument
  - argument, na kterém se bude zjišťovat délka
  - řetězec, buffer, soubor

### GetPosition

Vrací pozici hledaného řetězce v jiném řetězci či v souboru.

**GetPosition(návratový registr, násobnost, posun, text, hledaný text);**

- návratový registr
  - registr, do které se uloží pozice (od začátku) hledaného výrazu
  - registr
- násobnost
  - kolikátá hodnota hledaného výrazu se má použít
  - registr, číslo
- posun
  - posun počátku vyhledávání v textu
  - registr, číslo
- text
  - text, ve kterém se bude hledat pozice
  - buffer, soubor
- hledaný text
  - hledaný text
  - buffer, řetězec

### GetSubString

Vrací část řetězce z jiného vstupu.

**GetSubString(start, počet, výstup, vstup);**

- start
  - místo, od kterého se bude brát část řetězce
  - číslo, registr

- počet
  - počet znaků, které mají být zahrnuty do výběru
  - číslo, registr
- výstup
  - kam má být výsledná část řetězce uložena
  - soubor, buffer
- vstup
  - vstupní text
  - řetězec, soubor, buffer

### GetSystemValue

Vrací vybranou systémovou hodnotu.

**GetSystemValue(návratová hodnota, systémová hodnota);**

- návratová hodnota
  - regist, kam bude uložena vybraná systémová hodnota
  - registr
- systémová hodnota
  - 1 - Nejnižší byte IP adresy
  - 2 - Následující byte IP adresy
  - 3 - Následující byte IP adresy
  - 4 - Nejvyšší byte IP adresy
  - 5 - IP adresa jako jedno 4 bytové číslo
  - 6 - Nejnižší byte gateway
  - 7 - Následující byte gateway
  - 8 - Následující byte gateway
  - 9 - Nejvyšší byte gateway
  - 10 - Gateway jako jedno 4 bytové číslo
  - 11 - Nejnižší byte masky sítě
  - 12 - Následující byte masky sítě
  - 13 - Následující byte masky sítě
  - 14 - Nejvyšší byte masky sítě
  - 15 - Maska jako jedno 4 bytové číslo
  - 16 - Nejnižší byte MAC adresy
  - 17 - Následující byte MAC adresy
  - 18 - Následující byte MAC adresy
  - 19 - Následující byte MAC adresy
  - 20 - Nejvyšší byte MAC adresy
  - 21 - Připojení ethernetového kabelu

## Goto

Skok na určité místo v programu.

**Goto(návěští);**

- návěští
  - označení místa v programu, na které se bude skákat
  - návěští

## If

Podmíněný skok.

**if(operand porovnání, operand porovnání, operace porovnání, návěští);**

**if(speciální operátor, návěští);**

- operand porovnání
  - porovnávaná hodnota
  - registr, číslo
- operace porovnání
  - porovnávací operace
  - menší, větší, rovno, menší rovno, větší rovno, různé
- speciální operátor
  - speciální operátor pro stav připojení ethernetového kabelu ke konektoru
  - CONNECTED,NOTCONNECTED
- návěští
  - místo v programu, kam skákat v případě splnění podmínky
  - návěští

## Return

Návrat z podprogramu.

**Return;**

## RunFile

Spustí vykonávání skriptu v souboru.

**RunFile(cesta);**

- cesta
  - cesta k souboru, který má být spuštěn
  - soubor

## SendSerial

Odeslání dat po sériové lince.

**SendSerial(hodnota);**

- hodnota
  - hodnota, která má být odeslána
  - řetězec, buffer, číslo, registr, soubor, bit registru, port, bit portu, den, měsíc, rok, hodina, minuta, sekunda

## Set

Nastaví příslušnou část na určitou hodnotu.

**Set(co, hodnota);**

- co
  - co má být nastaveno
  - registr, bit registru, buffer, port, bit portu
- hodnota
  - jaká hodnota má být nastavena
  - odpovídá vždy v závislosti na tom, co nastavujeme

## SetMode

Nastavení způsobě vykonávání skriptu.

**SetMode(způsob);**

- způsob
  - určuje, jak bude skrip vykonáván
  - BLOCKED - vždy se vykoná jedna instukce apoté se vykonávání na okamžik přeruší a obslouží se ostatní části systému
  - NOTBLOCKED - vykonává se jedna insturukce za druhou. Po dobu vykonávání jsou ostatní funkce blokovány. V případě zavolání funkce, která má čekající charakter (Sleep, WaitFor) se systém automaticky přepne do módu BLOCKED.



## ShowLCD

Zobrazí na displeji zadaný text od pozice kurzoru.

**ShowLCD(text);**

- text
  - text, který bude zobrazen
  - text, buffer, soubor

## Sleep

Pozastaví vykonávání skriptu na zadanou dobu v ms.

**Sleep(délka);**

- délka
  - délka doby v ms, po kterou bude vykonávání skriptu pozastaveno
  - číslo

## WaitFor

Pozastavení programu do příchodu události.

**WaitFor(port, hodnota);**

**WaitFor(připojení);**

- port
  - port, na kterém bude záviset další pokračování programu
  - bit portu
- hodnota
  - hodnota, kterou musí nabývat port, aby se program dále spustil
  - číslo, bit registru, bit portu
- připojení
  - speciální podmínka pro pokračování v programu
  - CONNECTED, NOTCONNECTED

### B.3 Ukázka skriptu

Následující ukázka skriptu vykonává následující činnost. Snaží se připojit na server novinky.cz a stáhnout soubor /rss/. Po stažení tohoto souboru z něho získává postupně jednotlivé články, které dále přeposílá na sériovou linku.

program:

```

SendSerial("Init\n\r");
SendSerial("Connecting...\n\r");
WaitFor(CONNECTED);
SendSerial("Connected\n\r\n\r");

DownloadFile("www.novinky.cz",80,"/rss/",?NOVINKY.XML,#1);

GetPosition(#1,2,0,?NOVINKY.XML,"<title>");
If(#1,-1,==,"konec");
Arith(#1,+,#1,1);

EndProgram();

```

smyčka:

```

GetPosition(#2,1,#1,?NOVINKY.XML,"<title>");
If(#2,-1,==,"konec");

GetPosition(#3,1,#2,?NOVINKY.XML,"</title>");

Arith(#2,+,#2,7);
Arith(#4,-,#3,#2);

GetSubString(#2,#4,@0,?NOVINKY.XML);
GetPosition(#5,1,#2,?NOVINKY.XML,"<description>");
GetPosition(#6,1,#2,?NOVINKY.XML,"</description>");

Arith(#5,+,#5,13);
Arith(#7,-,#6,#5);

GetSubString(#5,#7,@1,?NOVINKY.XML);

SendSerial(@0);
SendSerial("\n\r");
SendSerial(@1);
SendSerial("\n\r\n\r");
Arith(#1,+,#3,1);

Goto("smycka");

```

konec:

```

EndProgram();

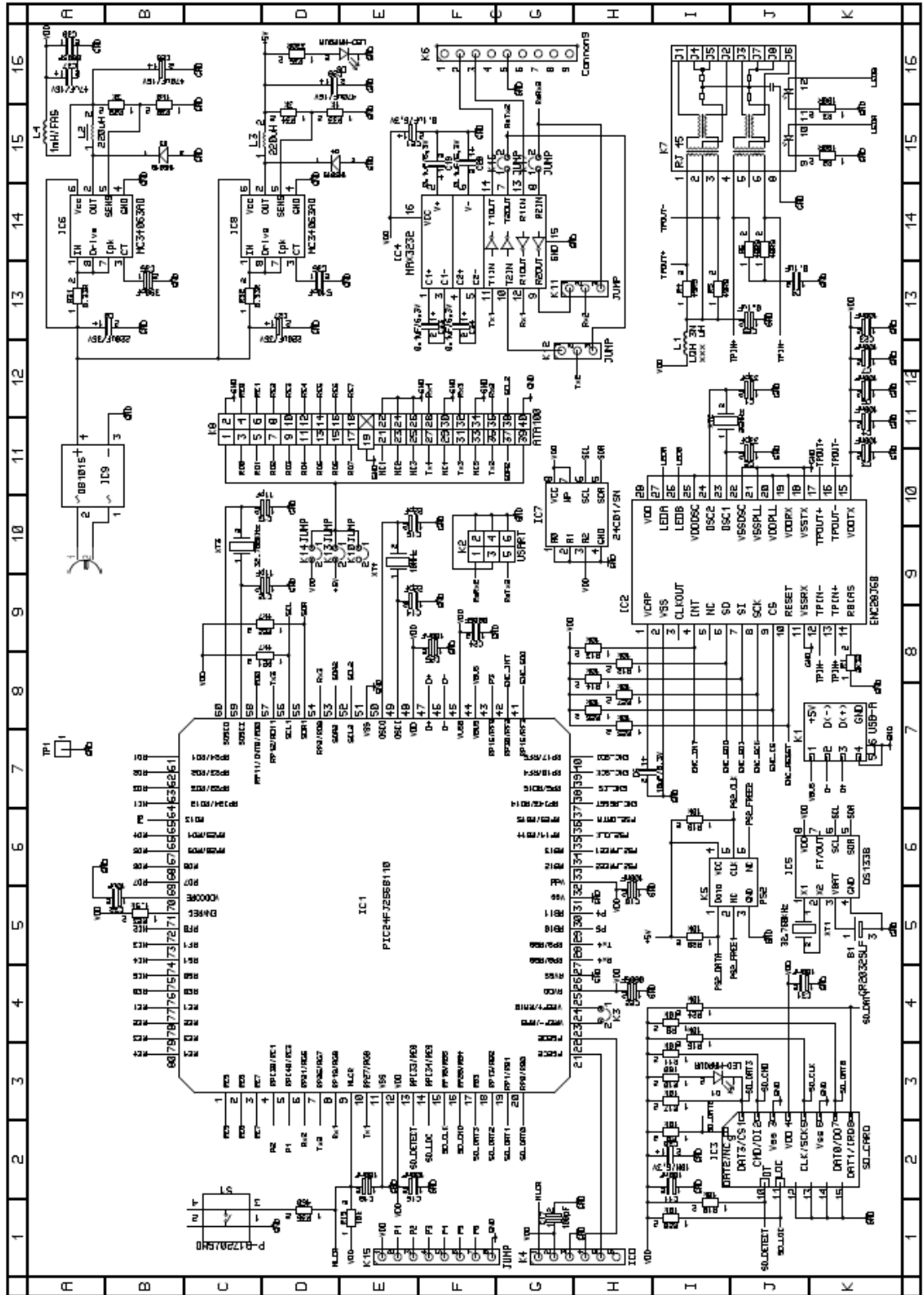
```

## Příloha C

# Schéma zapojení a DPS

V následující kapitole jsou uvedeny obrázky schématu a DPS.

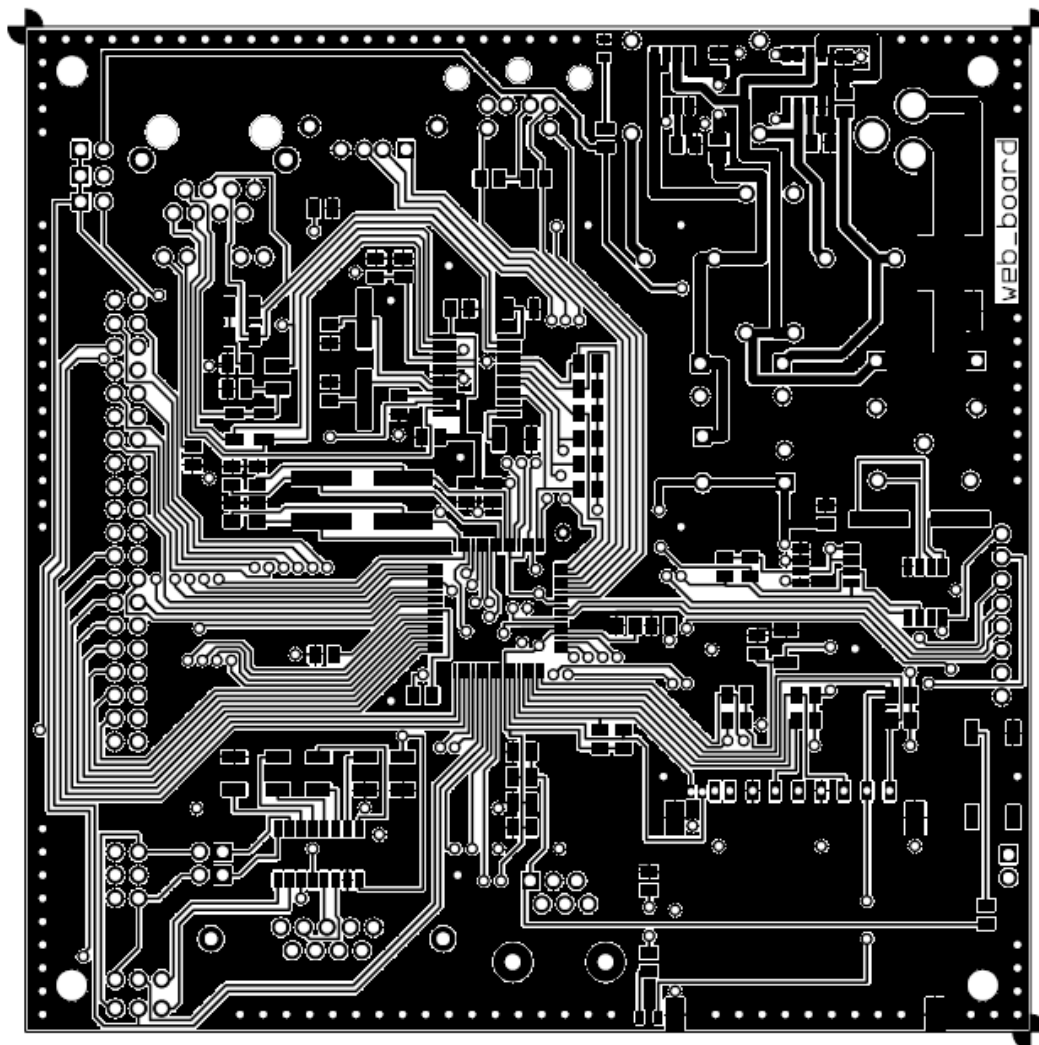
C.0.1 Celkové schéma zapojení



Obrázek C.1: Celkové schéma zapojení  
(Vlastní zdroj)

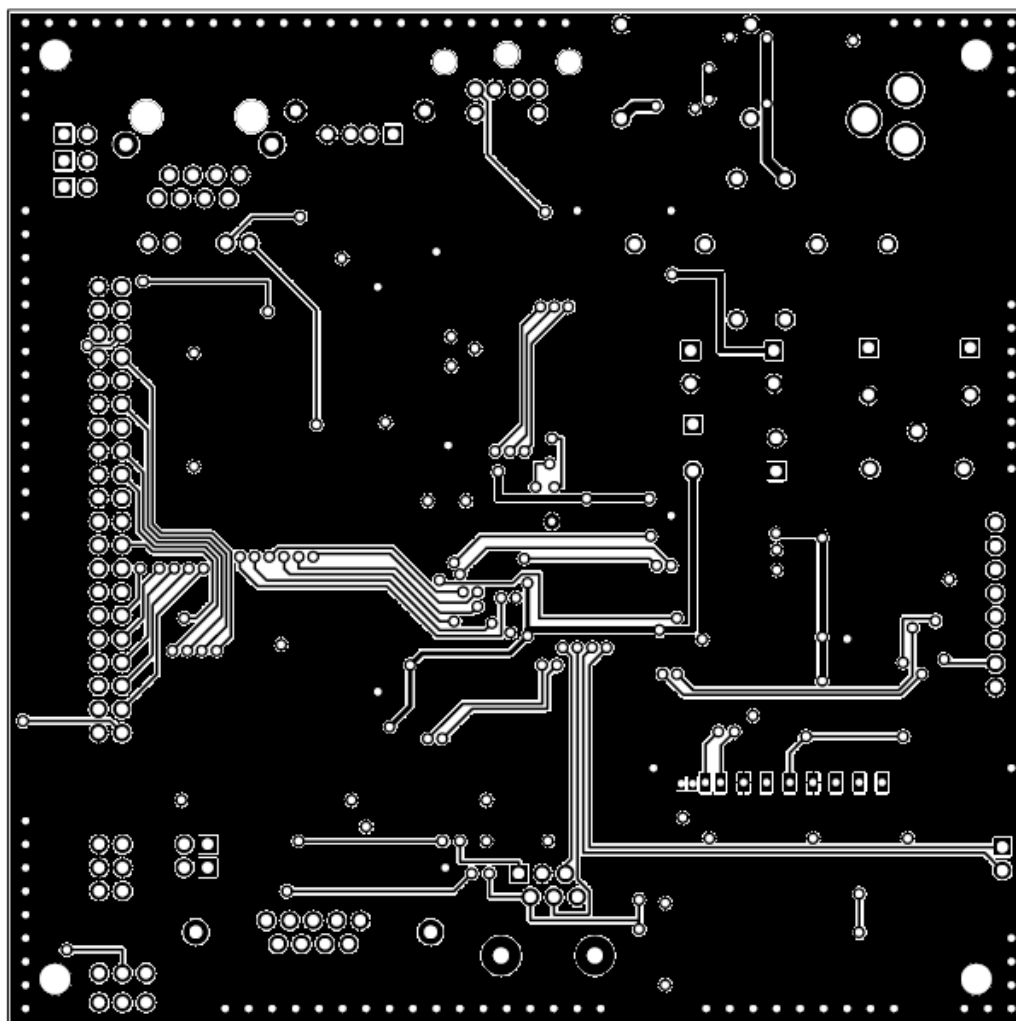
## C.1 Deska plošných spojů

### C.1.1 Vrchní vrstva



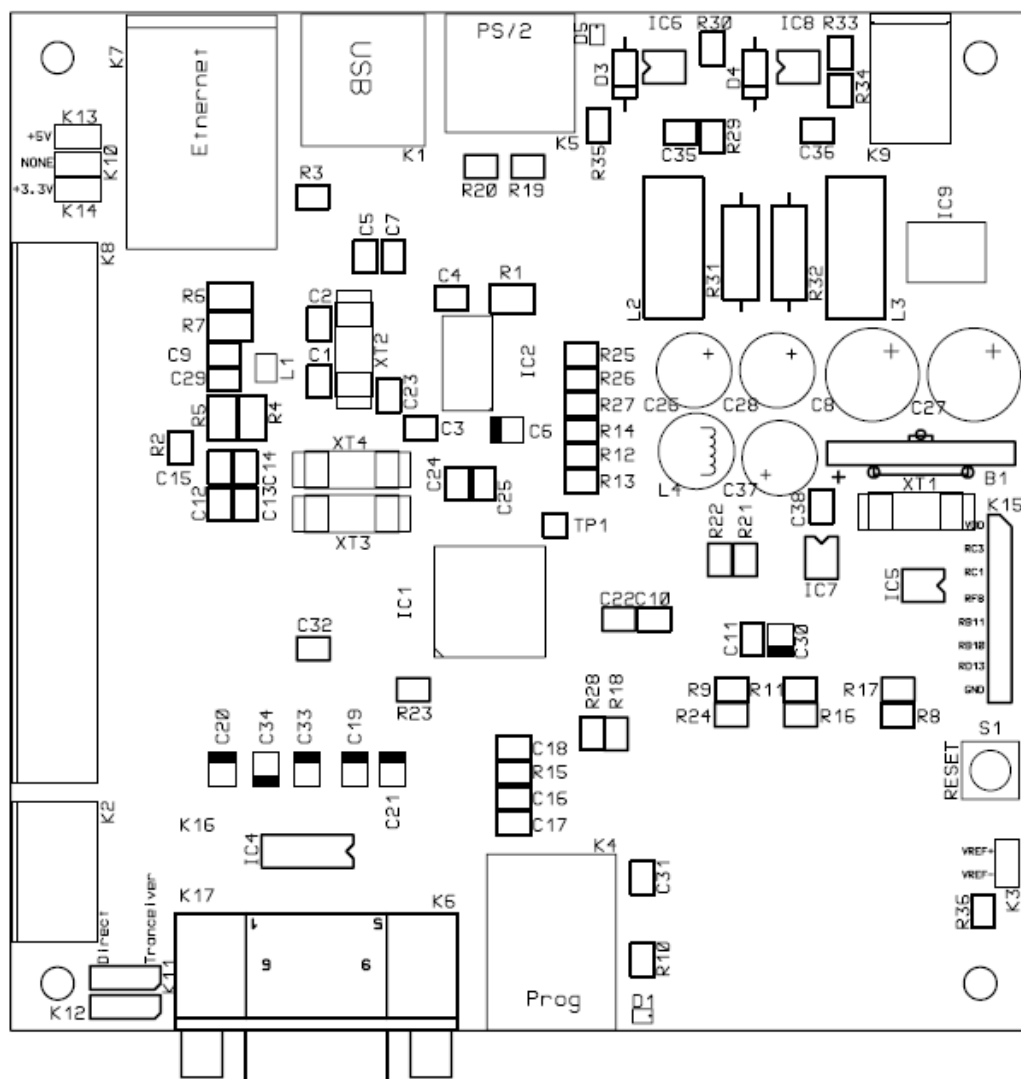
Obrázek C.2: Vrchní vrstva DPS  
(Vlastní zdroj)

## C.1.2 Spodní vrstva



Obrázek C.3: Spodní vrstva DPS  
(Vlastní zdroj)

### C.1.3 Osazení desky plošných spojů



Obrázek C.4: Osazení DPS  
(Vlastní zdroj)

## C.2 Seznam součástek

Součástka	Hodnota	Součástka	Hodnota	Součástka	Hodnota
R1	2k32	C5	100 nF	K2	USART
R2	180R	C6	10 uF (6.3 V)	K3	JUMP2
R3	180R	C7	100 nF	K4	WEBP 6-6 LP
R4	49R9	C8	220 uF (35 V)	K5	PS2
R5	49R9	C9	0.1 uF	K6	CANNON 9
R6	49R9	C10	100 nF	K7	MIC24311
R7	49R9	C11	100 nF	K8	ATA100
R8	10k	C12	11 pF	K9	PC-GK2.5
R9	10k	C13	11 pF	K10	JUMP2
R10	180	C14	22 pF	K11	JUMP2
R11	10k	C15	22 pF	K12	JUMP2
R12	10k	C16	100 nF	K17	JUMP2
R13	10k	C17	100 pF	B1	CR2032SLF
R14	10k	C18	100 nF	S1	P-B1720
R15	10k	C19	0.1 uF (6.3 V)	IC1	PIC24FJ256B108
R16	10k	C20	0.1 uF (6.3 V)	IC2	ENC28J60
R17	10k	C21	0.1 uF (6.3 V)	IC3	
R18	10k	C22	100nF	IC4	ICL232
R19	10k	C23	100 nF	IC5	DS1338
R20	10k	C24	100 nF	IC6	MC34063AD
R21	4k7	C25	100 nF	IC7	24C01C
R22	4k7	C26	470 uF (16V)	IC8	MC34063AD
R23	1k5	C27	220 uF (35V)	IC9	DB101S
R24	10k	C28	470 uF (16V)	D1	LED-MARQUA
R25	10k	C29	0.1 uF	D3	1N5819
R26	10k	C30	10 uF (6.3 V)	D4	1N5819
R27	10k	C31	100 nF	D5	LED-MARQUA
R28	10k	C32	10 uF	XT1	32.768 KHz
R29	3k	C33	0.1uF (6.3 V)	XT2	25 MHz
R30	1k8	C34	0.1uF (6.3 V)	XT3	32.768 kHz
R31	0R33 (1.6 W)	C35	390pF	XT4	10 MHz
R32	0R33 (1.6 W)	C36	510 pF		
R33	1k	C37	47 uF (16 V)		
R34	3k	C38	100 nF		
R35	330R	L1	LQH 3N		
R36	460R	L2	220 uH		
C1	33 pF	L3	220 uH		
C2	33 pF	L4	1 mH/FAS		
C3	100 nF	TP1	TESTPIN		
C4	100 nF	K1	USB-A		

Tabulka C.1: Seznam použitých součástek a jejich hodnot  
(Vlastní zdroj)



## Příloha D

# Obsah příloženého CD

Na příloženém CD je tento text ve formátech PDF a ve zdrojovém textu v Latexu. Dále se CD nachází veškeré podklady pro výrobu včetně dat do editoru a zdrojové kódy implementační části.

CD

- | - Datasheet
- | - Schema a DPS
  - | - Editor
  - | - Vyroba
- | - Text
  - | - Latex
- | - Zdrojove kody