



Středoškolská technika 2013

Setkání a prezentace prací středoškolských studentů na ČVUT

Správa I/O po LAN

Michal Drahoš

Střední průmyslová škola elektrotechnická
a Vyšší odborná škola, Pardubice, Karla IV. 13

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Správa I/O po LAN

Michal Drahoš

Pardubice 2013

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor SOČ: 18. Informatika

Správa IO po LAN

Autor: Michal Drahoš

Škola: Střední průmyslová škola elektrotechnická a
Vyšší odborná škola
Karla IV. 13
531 69 Pardubice

Pardubice 2013

Prohlášení:

Prohlašuji, že jsem svou práci vypracoval samostatně, použil jsem pouze podklady (literaturu, SW atd.) uvedené v příloženém seznamu a postup při zpracování a dalším nakládání s prací je v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Pardubicích dne 15. května 2013

podpis:

Anotace

Práce se zabývá vývojem aplikace, která je schopna přenášet informace ze vstupní jednotky po síti. Též dokáže nastavovat po síti výstupní jednotku, která je tvořena několika relé. Aplikace je určena spíše ke studijním účelům, než pro konkrétní nasazení do praxe. Tomu odpovídá zvolený hardware a celá filozofie programu.

Klíčová slova

vstup, výstup, LAN, Ethernet, IP, ARP, ICMP, UDP, ARM procesor, posuvný registr, programování, studijní účely

1	Úvod.....	- 6 -
2	Mikroprocesor.....	- 6 -
3	Blokové schéma aplikace.....	- 7 -
4	Vývojové prostředí	- 7 -
5	Ethernetový modul.....	- 8 -
5.1	SPI komunikace	- 9 -
6	Výstupní deska.....	- 10 -
6.1	Indikační LED bargraf	- 11 -
7	Vstupní deska.....	- 12 -
7.1	Konfigurační DIP přepínače	- 13 -
8	Implementace síťových protokolů	- 14 -
8.1	Ethernetový rámec	- 14 -
8.2	IPv4	- 15 -
8.3	ARP.....	- 16 -
8.4	ICMP.....	- 17 -
8.5	UDP	- 18 -
8.6	Vlastní protokol aplikace	- 19 -
9	Firmware	- 20 -
10	Software	- 21 -
11	Konstrukce	- 22 -
12	Závěr	- 22 -
	Seznam použitých značek a symbolů	- 23 -
	Seznam obrázků.....	- 23 -
	Seznam příloh	- 23 -
	Seznam použité literatury a zdrojů	- 24 -

1 Úvod

Cílem projektu bylo vytvořit aplikaci, která bude komunikovat se vzdáleným zařízením. Hlavním principem je přenos vstupní informace ze zařízení a nastavení výstupních portů (na kterých jsou připojena relé), po síti Internet.

Po určitých úpravách může tato aplikace sloužit jako jednoduché ovládání I/O v LAN síti, jak v domácnosti, tak v průmyslu (avšak pro ovládání sekundárních aplikace, nikoliv k řízení provozu). Po přepnutí do autonomního módu jsou aktivní pouze časovače, které nejsou závislé na síti. Tudíž lze aplikaci využít i například jako rozšíření časové relé, ovládané po síti.

Zařízení bylo sestrojeno především za účelem studia programování 32-bitového ARM mikroprocesoru a základní síťové komunikace (implementace nejzákladnějších složek LAN komunikace).

Součástí aplikace je samotné hardwarové řešení (několik desek plošných spojů umístěných na základně z plexiskla) a obslužný program psaný v programovacím jazyce Java.

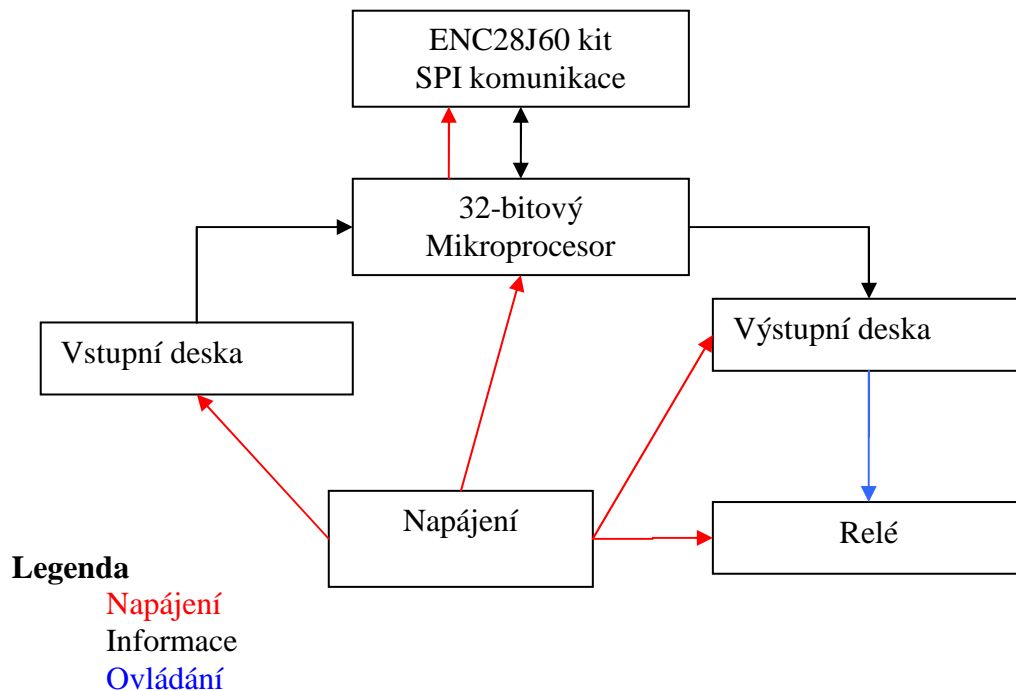
2 Mikroprocesor

Pro řízení celé aplikace byl zvolen mikroprocesor STM32F100RBT6B firmy ST Electronics. V jednočipu je zahrnuto výkonné jádro ARM 32-bit Cortex™-M3 CPU, s hodinovou frekvencí 24 MHz (1,25 DMIPS/MHz), 128 KB Flash paměti, 8 KB RAM paměti, interní PLL obvod, RTC, DMA, 51 I/O portů a 12 časovačů. Z těchto zmíněných periférií je v aplikaci použit obvod RTC (včetně zálohovací baterie) pro generování hodin reálného času (použito v časovém spínání výstupu), SysTimer (implementován v jádru ARM) pro přesné odměření zpožděvacích smyček a časovače TIM2, TIM3 a TIM4.

Jelikož je daný mikroprocesor vyráběn pouze v SMD provedení, je jeho použití v amatérském prostředí komplikované. Proto byl zvolen výukový kit STM32VLDISCOVERY, též od firmy ST Electronics, na němž je tento čip osazen včetně vývojových nástrojů (ST-Link debugger) a několik základních periférií (uživatelské tlačítko, reset, externí krystal 8MHz, a dvě uživatelské LED).

Vybraný mikroprocesor je pro danou aplikaci značně předimenzovaný. I přesto, že by celý algoritmus byl schopen řídit i osmibitový jednočip, například značky Atmel, čistě ze studijních důvodů byl vybrán 32-bitový mikroprocesor.

3 Blokové schéma aplikace



Obr. 1 Blokové schéma aplikace

4 Vývojové prostředí

Při vývoji byl použit software Atollic TrueSTUDIO v2.3.0 Lite. Jeho hlavní výhodou je možnost použít zabudovaný ST-Link debugger v Discovery Kitu. Tudiž programátor je schopen ihned po zkompilování projektu, bez jakéhokoliv zásahu do zapojení, nahrát program do Flash paměti mikroprocesoru a začít s laděním.

Pokročilý debugger používá všechny důležité nástroje. Programátor může krokovat jak program v jazyce C, tak v assembleru. V Lite verzi, která je po registraci zdarma, je k dispozici jeden breakpoint a několik watchpointů. Také lze v mikroprocesoru kdekoliv změnit data v uživatelské paměti.

Nevýhodou Lite verze TrueSTUDIA je, již zmíněný, jeden breakpoint a nemožnost sledovat registry periférií za běhu. Též limit 30kB zkompilevaného kódu je pro jisté aplikace omezením.

Další vývojové prostředí je například komerční μ Vision4 od výrobce Keil, popř. freeware software CooCox Builder.

5 Ethernetový modul



Obr. 2 Ethernetový kit

Pro připojení aplikace do sítě Internet byl použit jednočip firmy Microchip – ENC28J60. Tento čip je plně kompatibilní s normou 10/100/1000Base-T, má integrovanou podporu MAC, 10Base-T PHY, komunikaci half-duplex i full-duplex, interní DMA pro rychlou komunikaci s 8 KB bufferem. Obsahuje řadu paketových filtrů a jednotkou pro výpočet kontrolního součtu (CRC).

S mikroprocesorem je připojen pomocí SPI standartu, napájen z STM32VLDISCOVERY kitu.

Jedinou a nepostradatelnou funkcí modulu je přijmout z Internetu ethernetový paket a celý ho předat po sběrnici mikroprocesoru. Tento proces musí být funkční i reverzně – mikroprocesor odešle celý paket do kitu, ten dopočítá CRC součet a odešle ho do sítě.

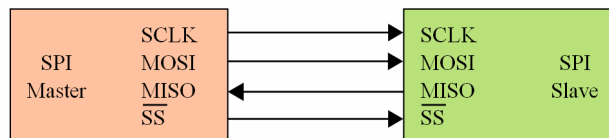
Veškerá kontrola paketu je tedy řízena mikroprocesor, který se na základě vstupní kontroly rozhodne, jestli bude paket zahozen, nebo zpracován. Interní filtry ENC28J60 nejsou použity.

Knihovna pro komunikaci s ENC28J60 byla stažena z webových stránek uvedených v závěru dokumentace.

Na trhu jsou k dispozici mnohem propracovanější obvody pro ethernetovou komunikaci. Spousta výkonných mikroprocesorů již obsahuje jednotku pro podporu ethernetu. Nicméně byla snaha se těmto obvodům vyhnout z důvodů vysoké ceny a složitosti konfigurace.

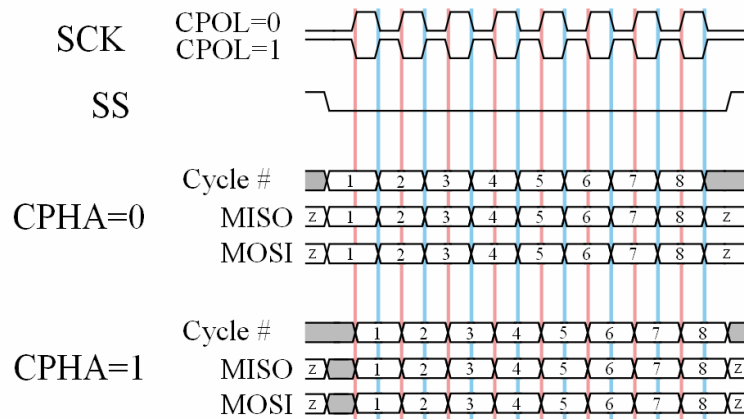
5.1 SPI komunikace

Serial Peripheral Interface je seriová sběrnice, která se používá pro komunikaci mezi mikroprocesorem a dalšími periferiemi. Komunikace je realizována pomocí společné sběrnice, výběr zařízení je realizován pomocí zvláštního vodiče SS (Slave Select), popř. CS (Chip Select).



Obr. 3 Blokové zapojení SPI komunikace

Z obrázku č. 3 je patrné, že přenos je synchronní – master generuje hodinový signál. Přenos je též full duplexní – obě zařízení mohou v jednom čase vysílat i přijímat.

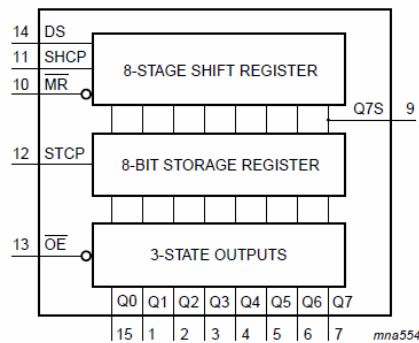


Obr. 4 Konfigurace SPI komunikace

Z obrázku č. 4 je patrné, že je třeba rozlišovat polaritu klidových signálů a hranu, při níž je signál přečten z datového vodiče. V tomto případě je CPOL = 0 a CPHA = 0. Frekvence hodin je 3 MHz.

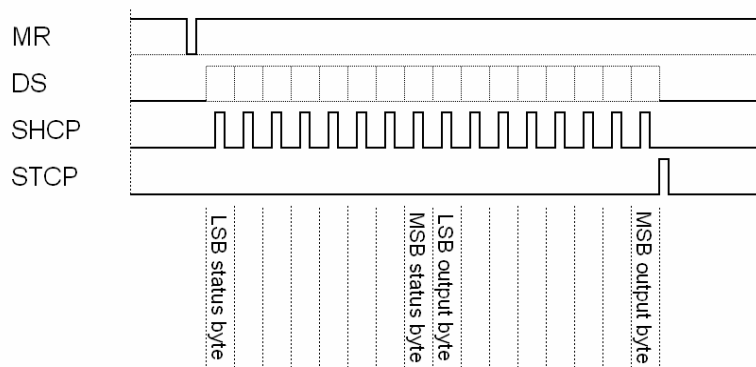
6 Výstupní deska

Základem výstupní desky jsou dva obvody 74HC595. Jedná se o převodníky SIPO (vstupní seriová informace - Serial In - je převedena na výstupní paralelní informaci - Paralel Out).



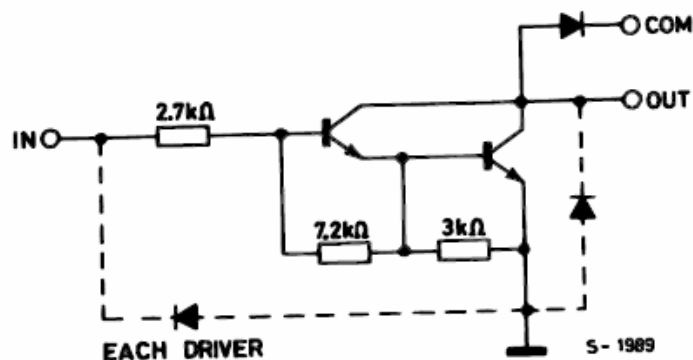
Obr. 5 Blokové schéma 74HC595

Deska je k mikroprocesoru připojena pomocí čtyř vodičů. Vodič SHCP přenáší hodinový signál z mikroprocesoru. Vodič DS posílá 16-ti bitovou informaci do prvního čipu. Vodič MR se stará o zresetování posuvných registrů. Poslední vodič STCP předá informaci z posuvných registrů do klopných obvodů na výstupu. Časový diagram je zobrazen na následujícím obrázku.



Obr. 6 Časový diagram komunikace s 74HC595

První posuvný registr má za úkol ovládat samotnou výstupní jednotku. K jeho výstupům je připojeno sedm vstupů integrovaného obvodu ULN2003A. Osmý bit je nezapojen, avšak softwarově přístupný. Obvod ULN2003A je pole sedmi budičů (konkrétně se jedná o tranzistory zapojené v Darlingově zapojení). Vnitřní schéma jednoho budiče je na obrázku č. 7.



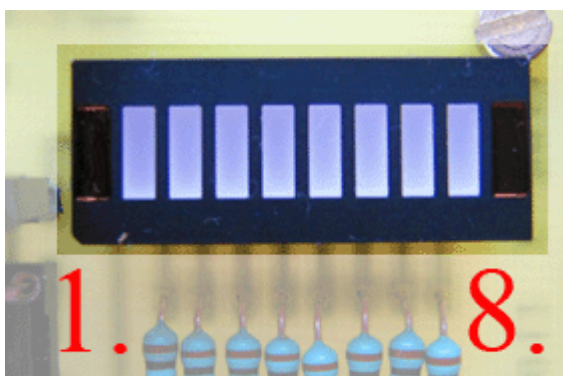
Obr. 7 Vnitřní schéma budiče ULN2003A

Z vnitřního schématu je patrné, že v klidovém stavu je výstup kanálu ULN2003A připojen (přes diodu) na napájecí napětí (COM). Po přivedení napětí na bázi prvního tranzistoru, tranzistory sepnou a výstup se uzemní. Z toho vyplývá, že koncové relé je spínáno proti zemi. Druhý kontakt relé musí být připojen ke kladnému pólu napájecího napětí (5V).

Na druhém obvodu 74HC595 jsou připojeny indikační LED diody. Jejich popis je v následující podkapitole.

Deska má vlastní stabilizaci napájení. Napájení je nutné udržet v rozsahu od 7 V do 30 V stejnosměrného napětí. Použitá logika je TTL pro 3,3 V. Deska je připojena k mikroprocesoru přes piny PB6, PB7, PB8, PB9. Konkrétní připojení pinů včetně nastavení GPIO v mikroprocesoru je uvedeno v souboru main.c.

6.1 Indikační LED bargraf



Obr. 8 Pohled na LED bargraf na desce

První indikační dioda signalizuje pohotovostní stav. Pokud je zařízení připojeno a je funkční, pouze problikává.

Následující dvě diody signalizují dostupnost routeru a serveru. V pravidelném intervalu aplikace odesílá ARP žádost na router a server. V případě, že nedostane odpověď, rozsvítí tyto diody.

Čtvrtá dioda se rozsvítí v případě přijetí ICMP paketu o nedostupnosti cílového zařízení, kam byl odeslán poslední paket. ICMP paket o nedostupnosti cíle může (ale nemusí!) být odeslán, tudíž se na tuto informace nedá stoprocentně spoléhat.

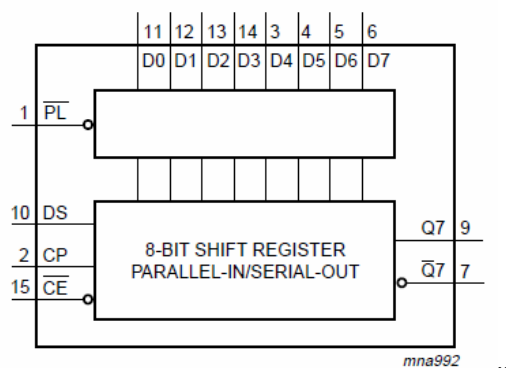
Pátá dioda svítí pouze v případě, že je alespoň jeden z časovačů aktivovaný.

Šestá a sedmá dioda informuje o chybových stavech. Pokud jedna z nich svítí, aplikace je v chybovém stavu a je třeba ji zresetovat (černé tlačítko na STM32VLDISCOVERY kitu).

Poslední osmá dioda informuje o nepřipojeném kabelu. V případě že svítí, kabel není připojen do obou stanic, nebo je přerušovaný.

7 Vstupní deska

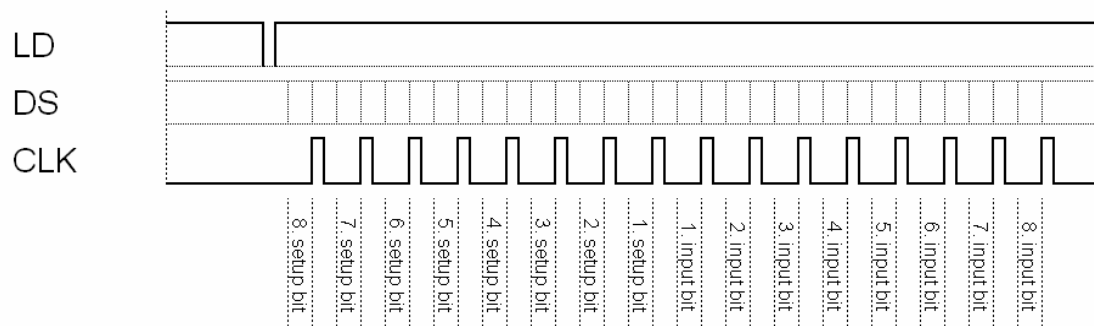
Stejně jako u výstupní desky, je zde základem posuvný registr, avšak 74HC165. Oproti výstupní desce má však opačnou funkci, tudíž je použit registr PISO (paralelní vstupní informace, seriová výstupní informace).



Obr. 9 Blokové schéma 74HC165

Tyto posuvné registry slouží jako oddělení externí aplikace od mikroprocesoru. V praxi se používá výhradně galvanické oddělení, avšak pro účely této aplikace je to zbytečné (připojená zařízení jsou známá a předpokládá se ověření kompatibility před připojením).

Vodič LD (PL) načte informaci ze vstupních pinů do vnitřních klopných obvodů. Na vodič CLK (CP) je připojen hodinový signál. Na posledním vodiči, DS, je konkrétní logická hodnota pinu.

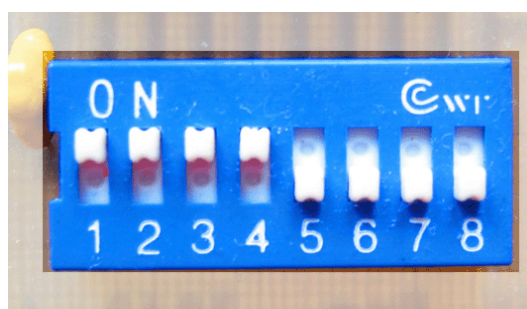


Obr. 10 Časový diagram komunikace s 74HC165

První čip se stará o přenos vstupní informace (8 digitálních vstupů, 3 V TTL logika). Druhý čip má na starost konfigurační DIP přepínače. Popis jednotlivých funkcí přepínačů je popsán níže v textu.

Deska má vlastní stabilizaci napájení. Napájení je nutné udržet v rozsahu od 4,5 V do 20 V stejnosměrného napětí. Použitá logika je TTL pro 3,3 V. Deska je připojena k mikroprocesoru přes piny PA8, PA9 a PA10. Konkrétní připojení pinů včetně nastavení GPIO v mikroprocesoru je uvedeno v souboru main.c.

7.1 Konfigurační DIP přepínače



Obr. 11 Pohled na konfigurační DIP přepínače na desce

První přepínač zajistí, že po resetu se načte defaultní nastavení sítě, tj. IP kitu 192.168.1.25, IP routeru 192.168.1.1, IP serveru 192.168.1.1, MAC adresa 00:1D:72:D2:35:1B, port serveru 81 a port aplikace 80.

Druhý přepínač povoluje konfiguraci nastavení po síti (mód administrátor). Bez jeho nastavení, není možné měnit IP adresy a nastavení sítě. Lze pouze používat časovače, číst vstupy a nastavovat výstupy.

Po nastavení třetího přepínače se aplikace přepne do read-only módu. Tento mód je autonomní. Není možné v něm cokoli měnit. Funkční zůstávají pouze časovače. Tento mód je také nejvíce bezpečný – ignoruje veškeré konfigurační pakety, které k němu přijdou.

Čtvrtý přepínač aktivuje alarm na vstupní jednotku. Po jeho aktivaci jsou periodicky čteny vstupy a v případě změny je odeslán paket (na IP a port serveru), který informuje o změně. Garantovaná doba pro zaregistrování změny je 1 sekunda (pokud by byla potřeba doba nižší, je vhodné doplnit vstupní jednotku monostabilním klopným obvodem).

Pátý až sedmý přepínač je vyhrazen pro uživatelské zpracování.

Poslední, osmý, přepínač nastavuje alternativní nastavení MAC adresy. Defaultní MAC je 00:1D:72:D2:35:1B, alternativní MAC adresa má hodnotu 00:16:EA:A0:0B:5F.

8 Implementace síťových protokolů

V aplikaci je implementována pouze nezbytná část několika síťových protokolů.

8.1 Ethernetový rámeček

Jedná se o nejnižší protokol před hardwarovou vrstvou. Data pro ethernetový paket jsou odeslána po SPI sběrnici do ENC28J60. Ten se dále postará o komunikaci s hardware a o předání informace.

Ethernetový rámeček							
Preamble	SFD	MAC cíle	MAC zdroje	Typ/délka	Data a výplň	CRC32	Mezera mezi rámečky
7× oktet 10101010	1× oktet 10101011	6 oktetů	6 oktetů	2 oktety	46-1500 oktetů	4 oktety	12 oktetů
					64-1518 oktetů		
72-1526 oktetů							

Obr. 12 Ethernetový rámeček

Úvodní Preamble a SFD (Start of Frame) obstará ethernetový kit. Jedná se čistě o hardwarovou záležitost. Preamble obsahuje střídavě binární „1” a „0”, pro synchronizování hodin u koncového zařízení.

MAC cíle je 48 bitová adresa koncového zařízení v síti. Tato adresa může být individuální (unicast), skupinová (multicast) nebo všeobecná (broadcast). MAC adresa musí být v lokální síti LAN unikátní, avšak nemusí být unikátní v celém Internetu (jako adresa IP).

MAC zdroje je adresa kitu (která se dá do jisté míry nastavit konfiguračními DIPy).

Položka Protokol specifikuje protokol v následující vrstvě (který se nachází v datové části rámce). V této aplikaci se jedná buď o protokol IPv4 (kód 0x0800) nebo ARP (0x0806).

Datová položka obsahuje samotné informace, které má rámec přenést.

Kontrolní součet CRC je 32-bitový kontrolní kód, který slouží ke kontrole odeslaných dat. Kód je spočítán ze všech položek rámce, mimo preamble a SFD. Koncová stanice spočítá stejným algoritmem přijatá data. V případě, že se oba CRC součty nerovnají, paket je zahozen jako vadný.

K výpočtu CRC32 je použita funkce implementovaná uvnitř ENC28J60. Do kitu jsou odeslány pouze MAC adresy, typ přenášených dat a samotná data.

8.2 IPv4

IPv4 Header Format																																	
Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	<i>Version</i>				<i>IHL</i>				<i>DSCP</i>				<i>ECN</i>				<i>Total Length</i>															
4	32	<i>Identification</i>												<i>Flags</i>				<i>Fragment Offset</i>															
8	64	<i>Time To Live</i>				<i>Protocol</i>				<i>Header Checksum</i>																							
12	96	<i>Source IP Address</i>																															
16	128	<i>Destination IP Address</i>																															
20	160	<i>Options (if IHL > 5)</i>																															

Obr. 13 IP datagram

Zatímco Ethernet měl za úkol přepravit data mezi jednotlivými hardwary, úkolem IP je přepravovat data v celém Internetu. Jako hlavní identifikace je použita IP adresa, která se skládá ze 4 bajtů. Spolu s maskou sítě rozhoduje o tom, jestli je cílová stanice v lokální síti, nebo ne.

Položka Version obsahuje verzi IP protokolu, v tomto případě číslo 0x04. IHL je velikost hlavičky vydělená čtyřmi. Jelikož je typická a minimální velikost hlavičky 20 bajtů, tato položka nabývá obvykle hodnoty 0x05. Maximální hodnota je stanovena na 0x0F, tudíž na 60 bajtů.

DSCP a ECN vyjadřují ideální podmínky přenosu paketu v Internetu. V praxi však k realizaci nedošlo. Momentálně jsou bity použity pro služby QoS (Quality of Service).

Total Length vyjadřuje celkovou délku paketu v bajtech.

Položky Identification, Flags a Fragment Offset se týkají fragmentace paketu. Jelikož paket v této aplikaci je příliš malý pro fragmentaci, není třeba se jím v dokumentaci zabývat.

Time To Live je důležitý prvek pro stabilitu Internetu. Před odesláním do sítě je definován počet směrovačů, přes které může paket projít. Je-li hodnota TTL rovna nule, paket je zahozen – vypršela jeho životnost. Tímto je zabráněno cyklickému pohybu paketu v Internetu.

Protocol, stejně jako u Ethernetu vyjadřuje protokol dat vyšší vrstvy, který paket nese.

Header Checksum je kontrolní součet. Je spočítán z hlavičky IP. V případě, že nesouhlasí, je paket zahozen.

Dále následují IP adresy zdrojového a cílového zařízení a samotná data.

8.3 ARP

Address Resolution Protocol je další ze síťových protokolů. Jeho úkolem je k určité IP adrese obstarat MAC adresu. Tento proces je nezbytný, neboť i přesto, že zařízení zná IP adresu cíle, musí vyplnit MAC adresu cíle v ethernetovém rámci.

Pokud je cílová stanice mimo LAN (IP adresa neodpovídá masce sítě), jako MAC adresa je použita adresa brány, která zajišťuje komunikaci s Internetem.

bit offset	0 – 7	8 – 15
0	Hardware type (HTYPE)	
16	Protocol type (PTYPE)	
32	Hardware address length (HLEN)	Protocol address length (PLEN)
48	Operation (OPER)	
64	Sender hardware address (SHA) (first 16 bits)	
80	(next 16 bits)	
96	(last 16 bits)	
112	Sender protocol address (SPA) (first 16 bits)	
128	(last 16 bits)	
144	Target hardware address (THA) (first 16 bits)	
160	(next 16 bits)	
176	(last 16 bits)	
192	Target protocol address (TPA) (first 16 bits)	
208	(last 16 bits)	

Obr. 14 ARP datagram

Hardware type a Protocol type specifikují typ hardwaru (pro Ethernet je to 0x0001, a pro IP 0x0800).

Hardware address length a Protocol address length obsahuje hodnotu 0x06 (pro Ethernet, 48-bitová MAC adresa má 6 bajtů). IP adresa má 4 bajty, proto Protocol address rovno 4.

Operation je kód operace, která se má provést. Pro žádost o překlad IP adresy na MAC je použita hodnota 1, pro odpověď je to hodnota 2.

Dále existují hodnoty 3 a 4, proto tzv. RARP – Reverse ARP. Toto nastavení však není implementováno.

Sender hardware/protocol address je pro adresy odesílatele. Totéž pro Target hardware/protocol address.

8.4 ICMP

V aplikaci jsou implementovány pakety Echo Request, Echo Reply (známé jako ping – pong) a Destination Unreachable.

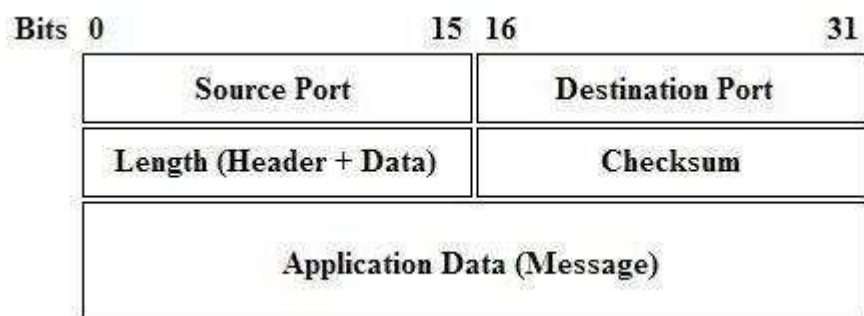
Echo Request (ping) je prostředek pro zjištění, zda dané zařízení je připojeno v síti a komunikuje. To je velmi důležité při testování a ožívování aplikace v síti. I přesto, že to je velmi dobrý nástroj, bývají tyto pakety filtrovány z důvodu útoků na síť.

Echo Reply (pong) je odpověď na ping. Pokud zařízení obdrží ping paket, je žádoucí, aby v co nejkratší době odpovědělo pong paketem.

Destination Unreachable je paket, který informuje, že cílová stanice, kam byl odeslán paket, je nedostupná. Tudiž paket nebyl zpracován. I tento paket bývá často filtrován z důvodů zneužití narušiteli sítě.

8.5 UDP

User Data Protocol je odlehčenou formou TCP protokolu. Obsahuje jen nejnútnejší informace pro komunikaci. Narozdíl od TCP, neobsahuje žádné mechanismy pro kontrolu doručení dat (handshake, integritu, ani pořadí). Tento krok si musí programátor, pokud ho vůbec potřebuje, zajistit na vyšší vrstvě.



Obr. 15 UDP datagram

Zdrojový (source) a cílový (destination) port aplikace jsou 16-bitová čísla. Spolu s IP adresou v protokolu IPv4 definují přesnou pozici zařízení v Internetu.

Délka (length) zahrnuje jak hlavičku, tak data. Minimální hodnota je 8.

Kontrolní součet (checksum) je spočítán jak z hlavičky, tak z datové části paketu.

V datové části je vlastní protokol aplikace.

8.6 Vlastní protokol aplikace

Při návrhu vlastního protokolu bylo dbáno na jednoduchost. Primárně bylo použito vztahu master - slave, avšak v případě resetu systému, nebo odeslání alarmu o změně vstupů, byla udělena výjimka – slave (aplikace) může odeslat informace masterovi bez předchozího zažádání.

V první tabulce jsou znázorněny funkce aplikace. Nulový bajt v protokolu slouží k výběru funkce, první bajt je náhodné číslo přiřazeno pro paket. Dále následují volitelná data.

Zcela stejný formát je u odpovědi na paket. Nulový bajt má však nastavený nejvyšší bit, pro signalizace odpovědi. ID paketu je stejné jako u přijatého paketu – podle toho se aplikace orientuje, zda přijala správné pakety ve správném pořadí.

Příkaz	CMD_TIME_GET	CMD_TIME_SET	CMD_KIT_TEST	CMD_SETP_GET	CMD_SETP_SET	CMD_ALRM_SET	CMD_ALRM_GET	CMD_INPT_GET	CMD_OUTP_GET	CMD_OUTP_SET	CMD_DBG_STAT_SET	CMD_DIP_GET	CMD_RST	CMD_RST_REQUEST
0. byte	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D
1. byte	ID paketu (náhodně generováno)													
2. - n. byte	-	čas 4B	-	-	IP apl. 4B	timer ID 1B	timer ID 1B	-	-	OR 1B	OR 1B	-	reset ID 1B	-
					IP router 4B	opak. 1B				AND 1B	AND 1B			
					IP server 4B	OR 1B								
					Maska 4B	AND 1B								
					MAC 6B	čas 4B								
					Port apl. 2B									
					Port server 2B									

Obr. 16 Protokol aplikace, požadavek

Odpovědi z aplikace														
0. byte	0x80	0x81	0x82	0x83	0x84	0x85	0x86	0x87	0x88	0x89	0x8A	0x8B	-	0x8D
1. byte	ID paketu													
2. - n. byte	čas 4B	"0"	verze 3B	IP apl. 4B	"0"	"0"	timer ID 1B	vstup 1B	vystup 1B	"0"	"0"	DIP 1B	-	reset ID 1B
		"k"	timer status 1B	IP router 4B	"k"	"k"	opak. 1B			"k"	"k"			
		"l"	config 1B	IP server 4B	"l"	"l"	OR 1B			"l"	"l"			
				Maska 4B			AND 1B							
				MAC 6B			čas 4B							
				Port apl. 2B										
				Port server 2B										
				config 1B										

Obr. 17 Protokol aplikace, odpověď

9 Firmware

Celý firmware je napsán v programovacím jazyce ANSI C. Firmware je vyvíjen v prostředí Atollic TrueStudio (postaveno na Eclipse), verze 2.3.0 Lite. Firmware byl, vyjma knihovny pro ENC28J60 a driverů pro STM32VLDISCOVERY kit, kompletně vyvíjen autorem. Veškeré zdroje, včetně internetových stránek, jsou vypsány na konci dokumentace.

Ačkoliv vysvětlit celý program je náročné, jsou zde popsány hlavní bloky kódu, včetně vysvětlení úseků, kde není smysl na první pohled patrný.

Ihned po resetu je provedena inicializace a nastavení všech periférií. Proběhne nastavení periférií kitu, ENC28J60, časovačů a konfigurace přerušení.

Po inicializaci je odeslán packet na zjištění MAC adresy serveru a routeru. Tímto začíná první komunikace se sítí. Zjišťování dostupnosti serveru a routeru se cyklicky opakuje. Nedostupnost signalizuje LED dioda.

Následně se program dostane do nekonečné smyčky. V této smyčce čeká buď na příchod paketu, nebo na přerušení. Ačkoliv rutiny přerušení se standardně zpracovávají ihned po vyvolání přerušení, v této aplikaci bylo třeba postupovat odlišně. Důvodem bylo to, že není možné připustit přerušení v době komunikace se sítí (odeslání paketu a čekání

na odpověď). Proto se při příchodu přerušení uloží adresa přerušovací rutiny do proměnné a po ukončení úlohy, kdy se program vrátí do nekonečné smyčky, se obslouží přerušení. Tento poněkud nestandardní krok bylo možné použít, neboť nebylo třeba zpracovávat informace ihned po příchodu – přerušení není natolik důležité, aby nemohlo počkat pár milisekund.

Dalším krokem je RTC. Jádro RTC u tohoto mikroprocesoru neobsahuje registry zvlášť pro hodiny, minuty, sekundy, .. ale obsahuje jeden 32-bitový registr, který se každou sekundu inkrementuje o jedničku – tudíž obsahuje sekundy. Jelikož by byl přepoččet, zejména na měsíce a roky, náročný, je v registru uložen UNIX time. Jedná se o číslo, které vyjadřuje počet sekund od 1.1.1970. Tento čas je třeba synchronizovat s počítačem, který bude s aplikací komunikovat.

Další mikroprocesory od firmy ST Electronics založené na ARM mají v RTC pro každý časový údaj zvláštní registr. To je pro programátora značné ulehčení.

Soubor INTERNET.c obsahuje funkce pro komunikaci s protokoly Ethernet, IP, ARP a ICMP. Soubor protokol.c obsahuje funkce pro komunikaci podle protokolu aplikace. Dále io.c obsahuje funkci pro ovládání periférií vstupu a výstupu, rtc.c se stará o hodiny reálného času, viz. výše, backup.c má na starosti uložení nastavení (IP, porty, maska).

10 Software

Obslužný program pro PC byl napsán v programovacím jazyce Java, vyvíjen v prostředí NetBeans. Hlavním důvodem byla relativní jednoduchost a přístupnost programovacích prostředků. Pro vývoj nebylo třeba znát struktury Windows API, ani řešit problémy debugingu při ožívování.

Program se skládá celkem ze sedmi tříd. Každá třída plní specifickou funkci.

Třída Core, obsahující „entry point“ se stará o GUI (vytvořené v designeru NetBeans prostředí) a o hlavní větvení programu. Pokud bychom měli funkci přirovnat k aplikaci Windows API, jednalo by se o analogii k funkci WindowProc. Dále obsahuje metody pro zaznamenávání událost do log souboru.

Třída GUI.java zpracovává ručně vytvořené grafické prvky. Jedná se především o prvky, které ovládají výstupní a vstupní jednotky, překreslení prvků.

Ini.java se stará o načtení konfigurace ze souboru config.ini. Popis a nastavení jednotlivých položek je uveden ve zdrojovém kódu.

Net.java navazuje a ukončuje spojení s aplikací. Stará se též o chybové hlášky a výjimky. Tato třída slouží jako brána pro tok informací do Internetu a zpět.

Protokol.java je vlastní implementace protokolu pro celou aplikaci. Přijatá data jsou předána metodě zpracovatPrichozi, která se postará o parsifikaci dat a vykoná požadované úkony.

Vlakno.java a VlaknoServeru.java jsou analogie k WinAPI Threads. Vlakno.java se stará o předání přijatých odpovědí ohledně konfigurace aplikace. Vlákno běží v nekonečné smyčce. Pokud přijde paket na port, který je vyhrazen pro konfigurační pakety, toto vlákno je zachytí a zpracuje.

VlaknoServeru.java se stará o pakety, které jsou aplikací odeslány k serveru. Převážně se jedná o pakety, které signalizují změnu stavu vstupní jednotky, popř. reset systému.

11 Konstrukce

Desky plošných spojů byly v amatérských podmínkách vyrobeny metodou nažehlení toneru na kuprexitovou desku. Alternativou k metodě nažehlení toneru, je výroba desky fotocestou. Avšak pro tuto aplikaci byla přesnost a preciznost nažehlení naprosto dostačující. Několik podleptaných míst bylo opraveno a na funkčnosti desky to nemá žádný vliv.

Pro mechanickou konstrukci bylo použito plexisklo tloušťky 5 mm. Úprava plexiskla byla provedena na vibrační brusce a díry byly vyvrtány na stolní vrtačce. Pro oddělení dolní a horní desky byla použita závitová tyč se závitem M5.

Desky plošných spojů jsou do plexiskla přimontovány na distančních sloupcích 10 mm vysokých, se závitem M3.

12 Závěr

Cílem práce bylo vytvořit zařízení pro správu vstupních a výstupních jednotek na vzdáleném zařízení v síti LAN.

Zařízení bylo vyvíjeno průběžně jako hardware a software – tudíž po konstrukci elektroniky byl ihned napsán testovací program, který ověřil funkčnost zapojení. Při vývoji se nevyskytly žádné velké chyby ani nepředpokládané stavy. Při vývoji byl použit program Wireshark, pro analýzu příchozích paketů a jejich ověření (kontrolní součty).

Aplikaci se podařilo bez větších problémů vytvořit a je zcela funkční. Dále jsou připraveny další rozšíření a konkrétní přizpůsobení pro ovládání spotřebičů v domácnosti.

Seznam použitých značek a symbolů

Jednotka

[V]

[Hz]

[B]

Význam

volt

frekvence

množství dat v informatice

Seznam obrázků

Obr. 1	Blokové schéma aplikace
Obr. 2	Ethernetový kit
Obr. 3	Blokové zapojení SPI komunikace
Obr. 4	Konfigurace SPI komunikace
Obr. 5	Blokové schéma 74HC595
Obr. 6	Časový diagram komunikace s 74HC595
Obr. 7	Vnitřní schéma budiče ULN2003A
Obr. 8	Pohled na LED bargraf na desce
Obr. 9	Blokové schéma 74HC165
Obr. 10	Časový diagram komunikace s 74HC165
Obr. 11	Pohled na konfigurační DIP přepínače na desce
Obr. 12	Ethernetový rámec
Obr. 13	IP datagram
Obr. 14	ARP datagram
Obr. 15	UDP datagram
Obr. 16	Protokol aplikace, požadavek
Obr. 17	Protokol aplikace, odpověď

Seznam příloh

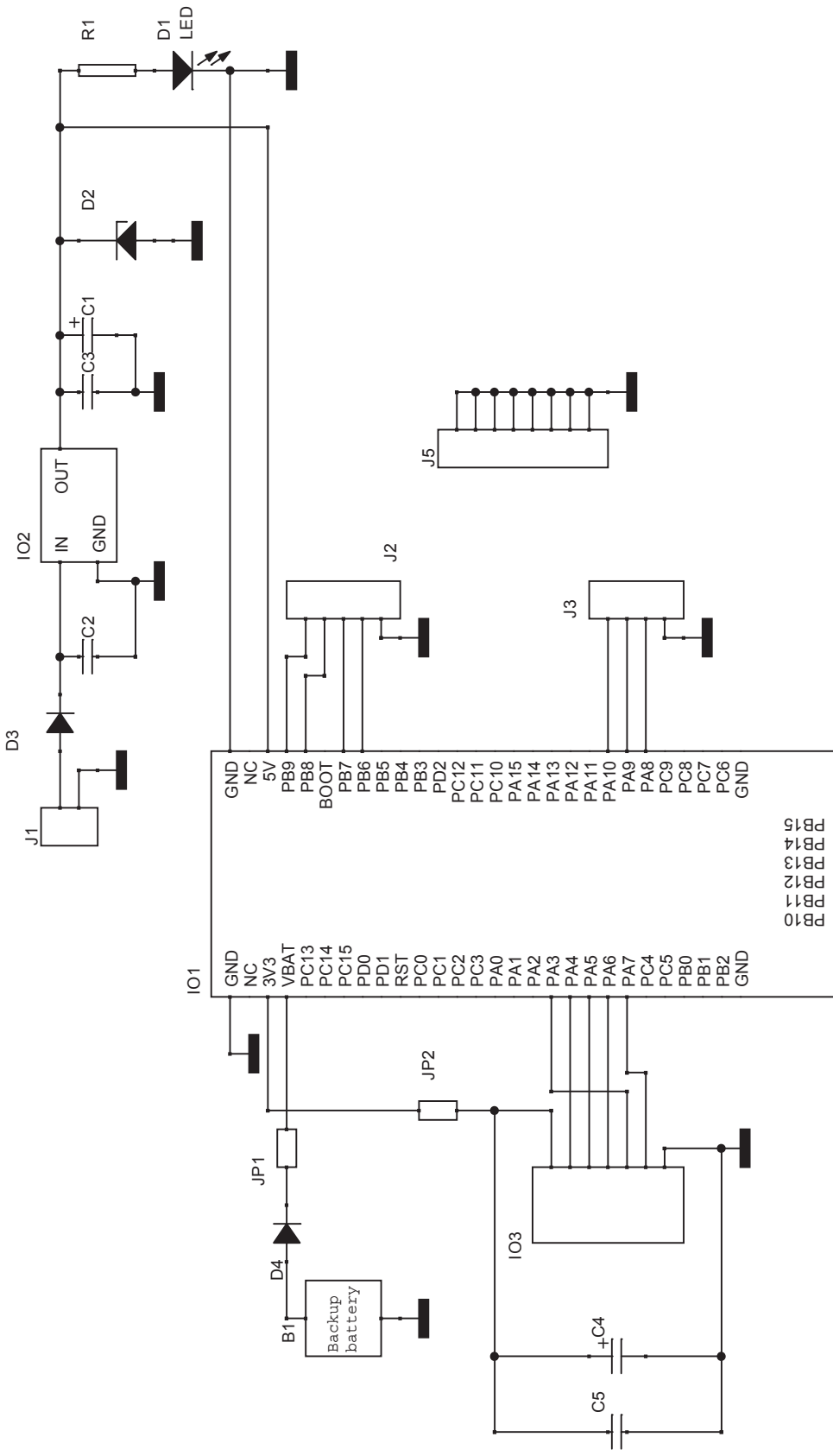
- 1 Schéma hlavní (core) desky, 1 list
- 2 Schéma výstupní desky, 1 list
- 3 Schéma vstupní desky, 1 list

- 4 Schéma relé desky, 1 list
- 5 Schéma napájecí desky, 1 list
- 6 Předloha DPS a osazovací výkres hlavní (core) desky, 1 list
- 7 Předloha DPS a osazovací výkres výstupní desky, 1 list
- 8 Předloha DPS a osazovací výkres vstupní desky, 1 list
- 9 Předloha DPS a osazovací výkres relé desky, 1 list
- 10 Předloha DPS a osazovací výkres napájecí desky, 1 list
- 11 Výkres konstrukce, horní deska, 1 list
- 12 Výkres konstrukce, dolní deska, 1 list
- 13 Výkres konstrukce, bokorys, 1 list
- 14 Rozpis součástí, 2 listy
- 15 Ukázka obslužného programu, 1 list
- 16 Fotografie celého zařízení, 1 list

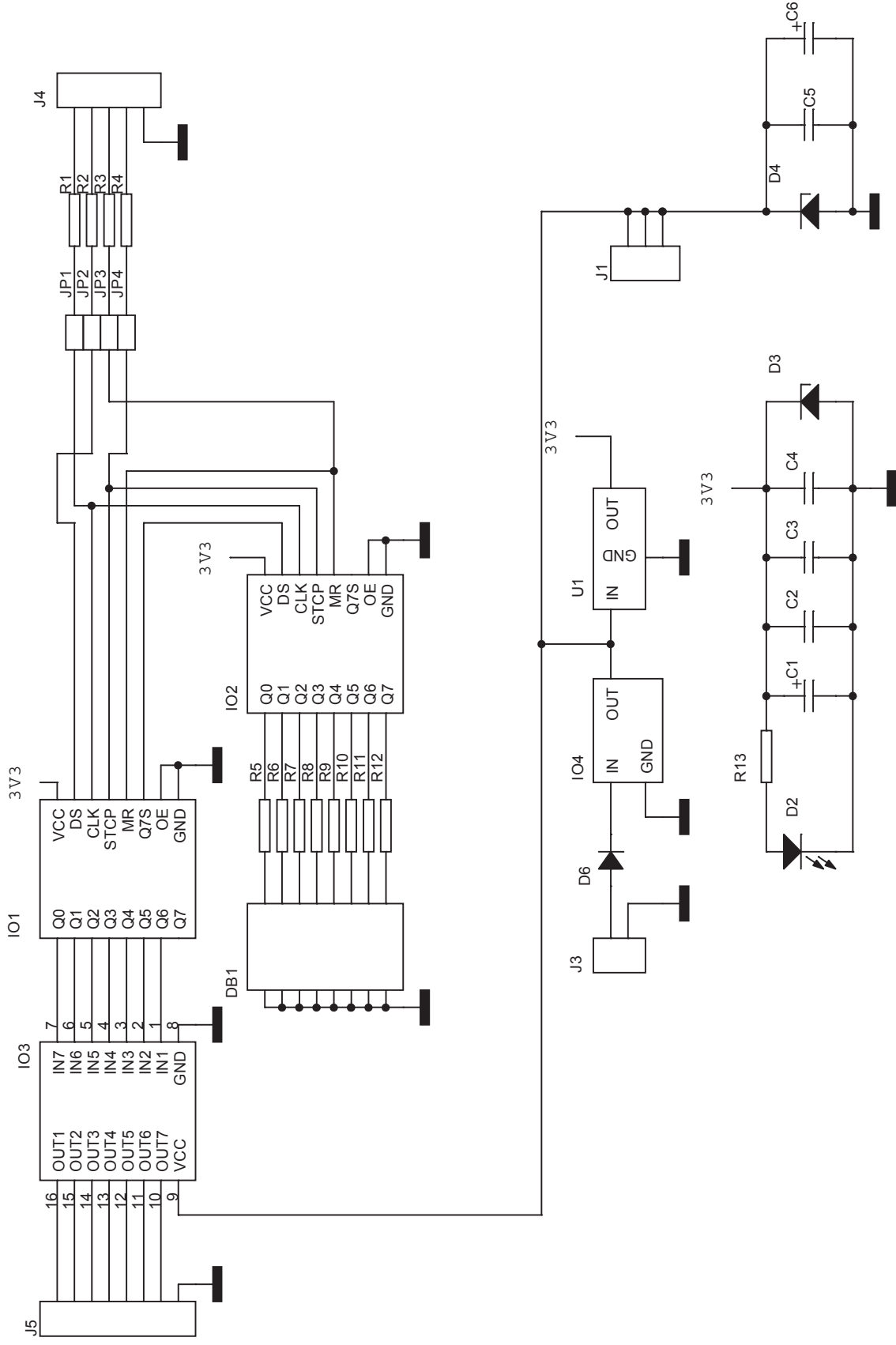
Seznam použité literatury a zdrojů

- [1] MARD, [MCU.CZ]. Začínáme s STM32VL Discovery kitem: 36. - webserver s ENC28J60. [online]. [cit. 2013-03-31]. Dostupné z: <http://mcu.cz/comment-n2893.html>
- [2] KABELOVÁ, Alena a Libor DOSTÁLEK. *Velký průvodce protokoly TCP/IP a systémem DNS*. 5., aktualiz. vyd. Brno: Computer Press, 2008, 488 s. ISBN 978-80-251-2236-5.
- [3] HEROUT, Pavel. *Učebnice jazyka C*. 4., přeprac. vyd. České Budějovice: Kopp, 2004, 271, viii s. ISBN 80-723-2220-6.
- [4] *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-03-31]. Dostupné z: http://cs.wikipedia.org/wiki/Hlavn%C3%AD_strana
- [5] <http://www.st.com> – datasheet k STM32VLDISCOVERY, mikroprocesoru STM32F100RBT6B, STM32F10x Standard Peripheral Library

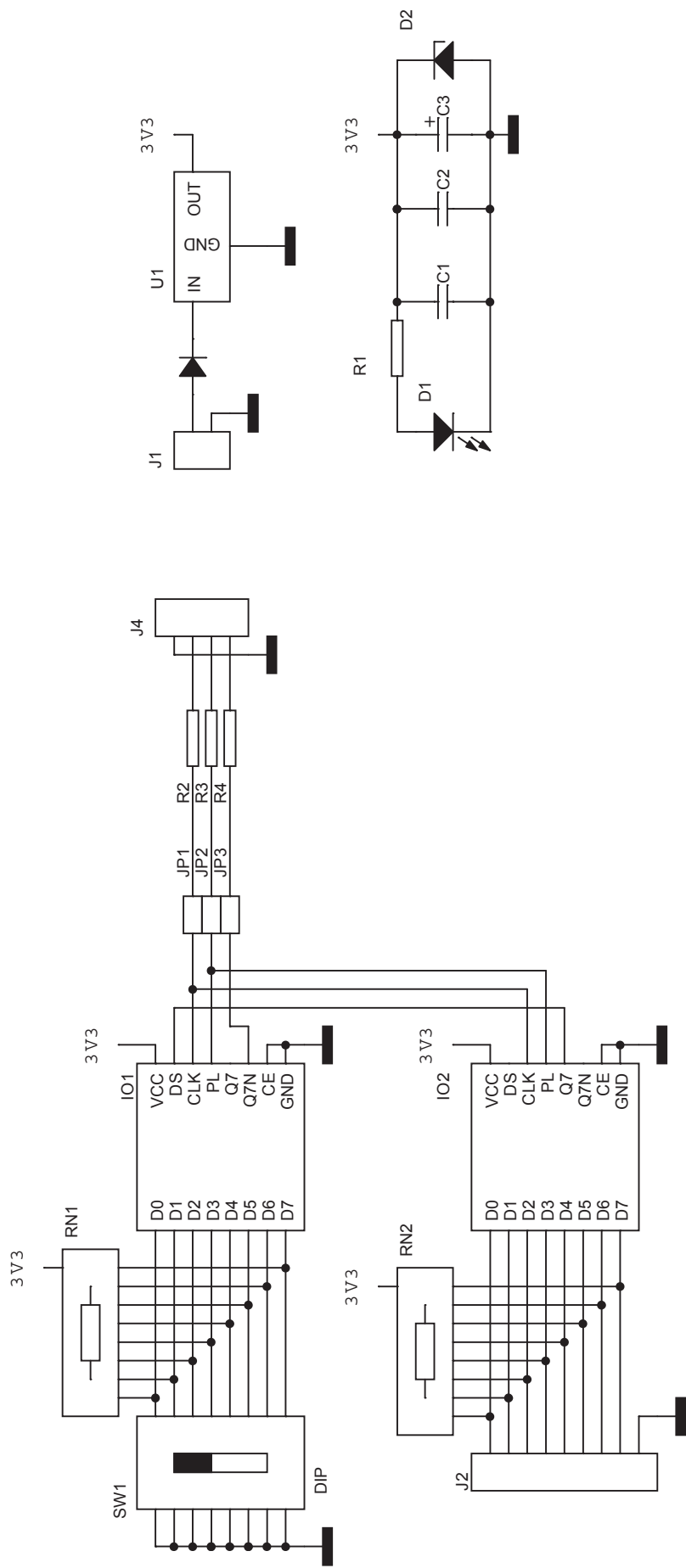
Příloha č. 1 - Schéma hlavní (core) desky, 1 list



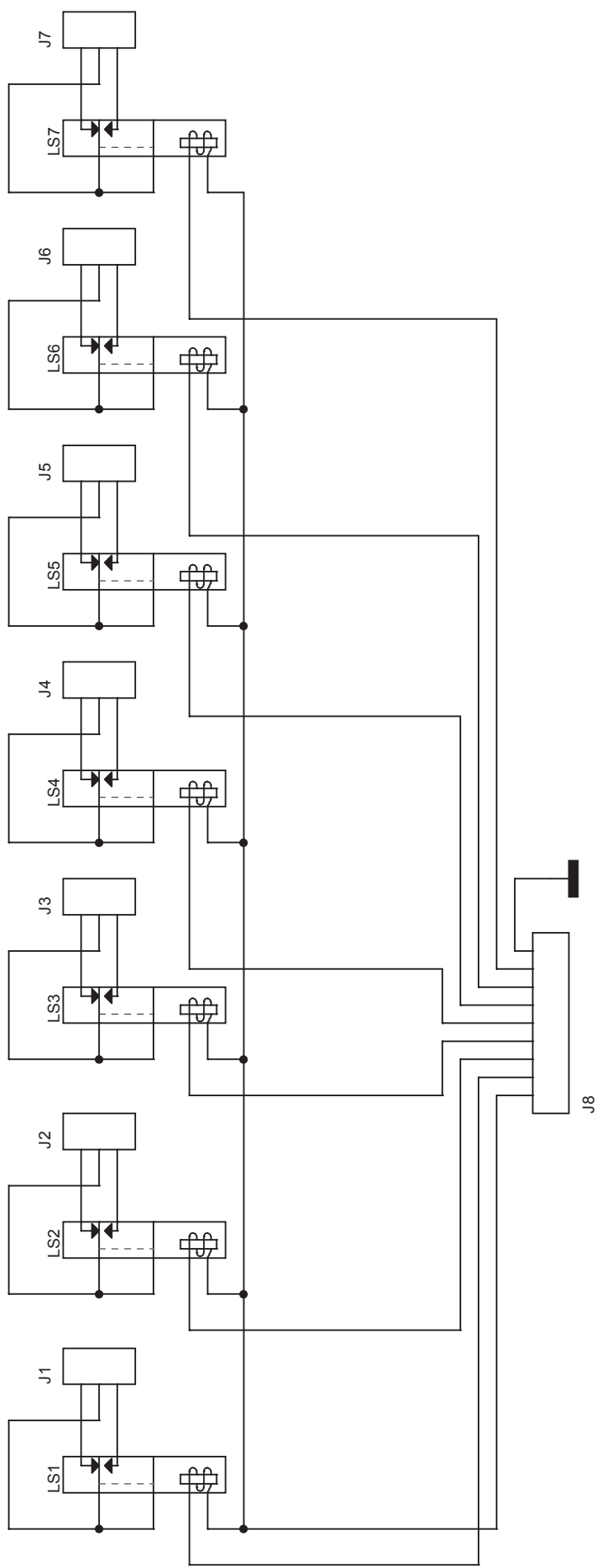
Příloha č. 2 - Schéma výstupní desky, 1 list



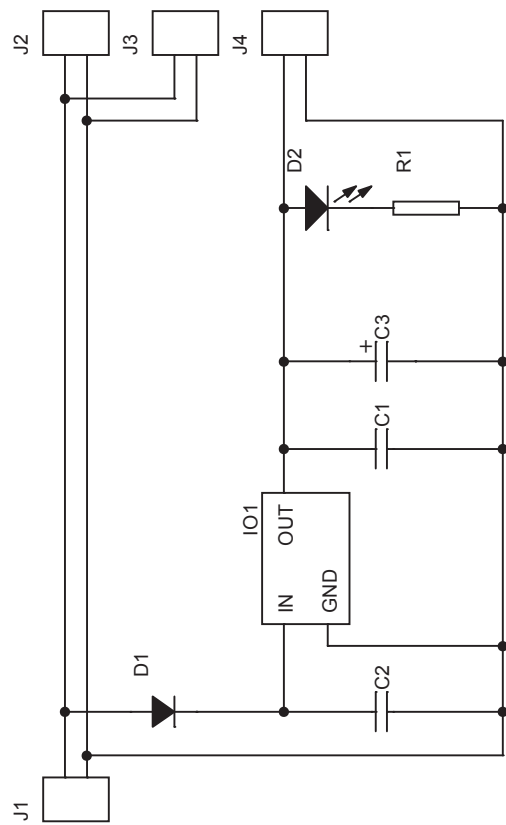
Příloha č. 3 - Schéma vstupní desky, 1 list



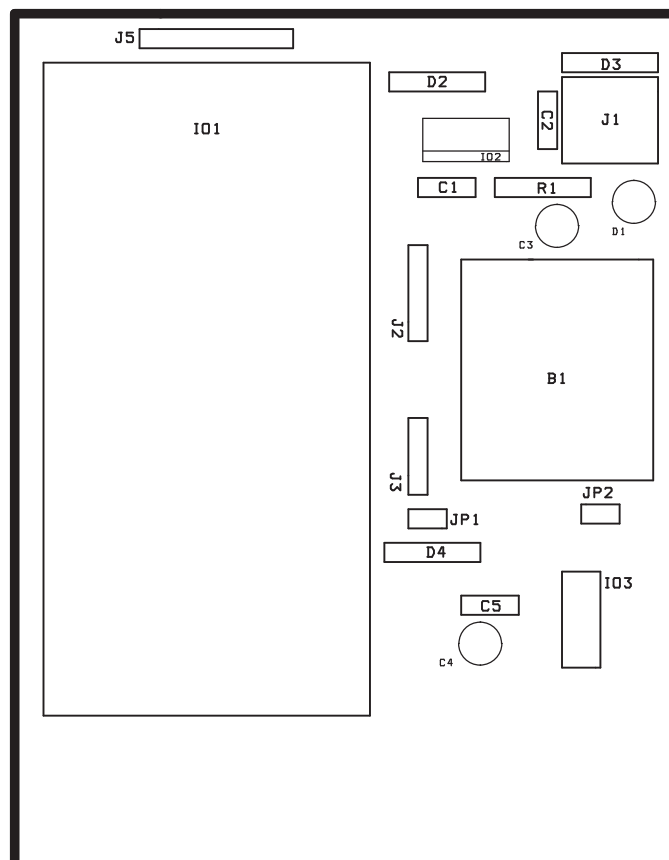
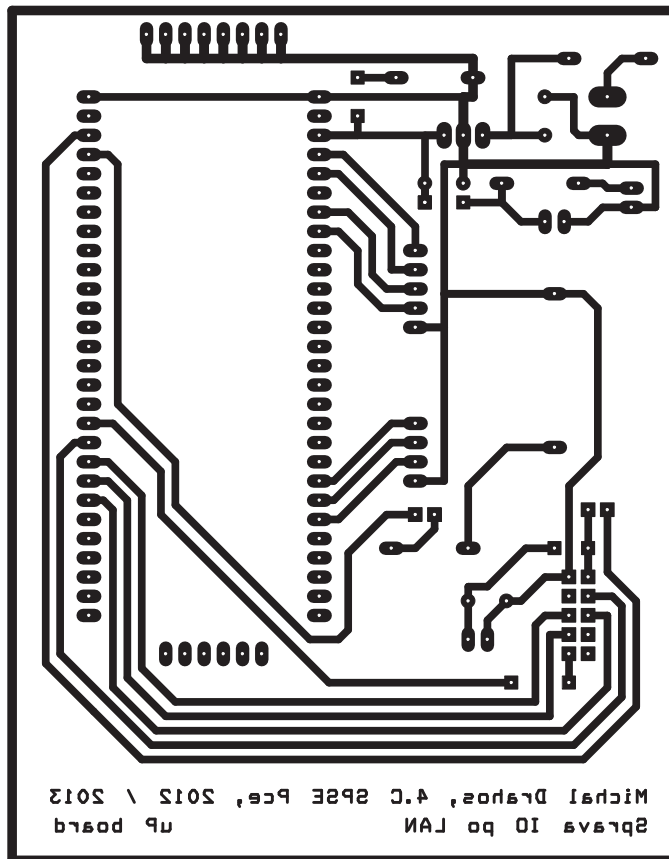
Příloha č. 4 - Schéma relé desky, 1 list



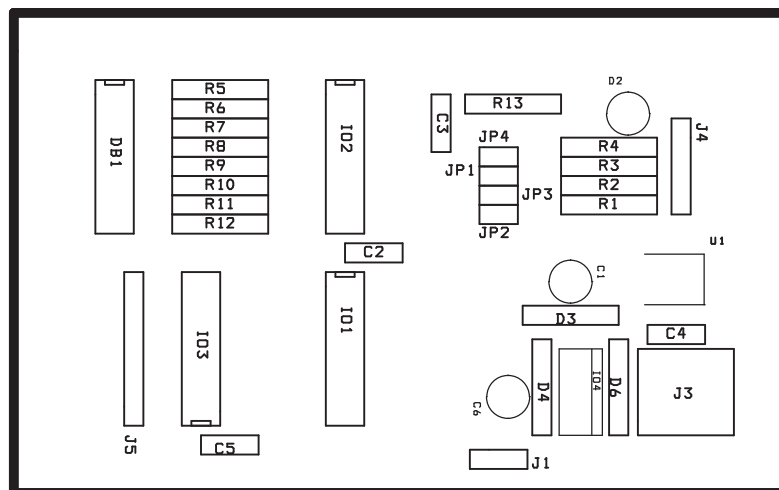
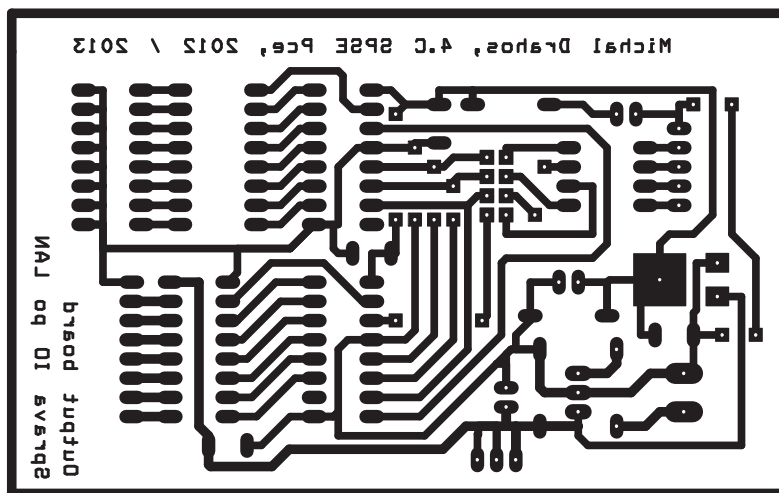
Příloha č. 5 - Schéma napájecí desky, 1 list



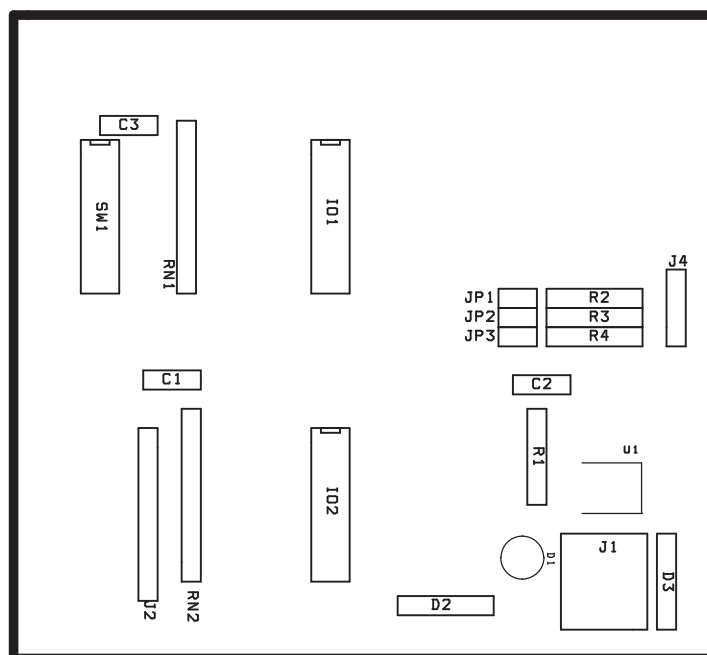
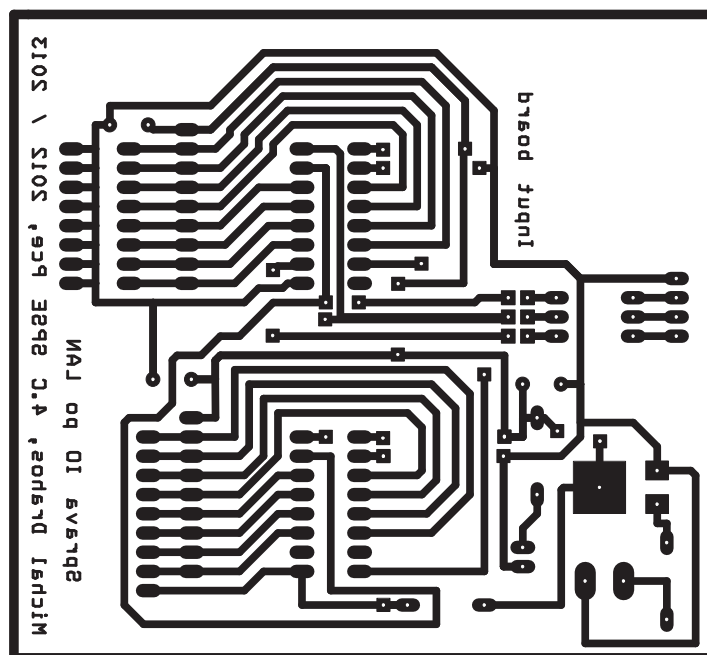
Příloha č. 6 - Předloha DPS a osazovací výkres hlavní (core) desky, 1 list



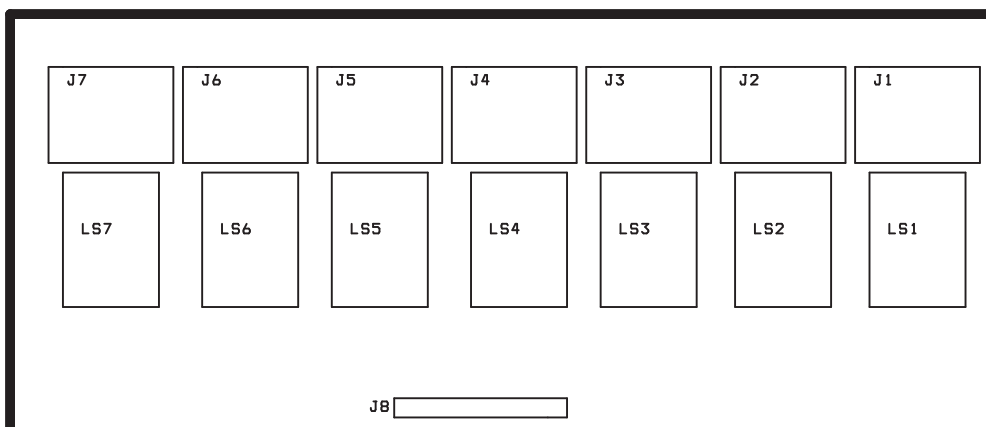
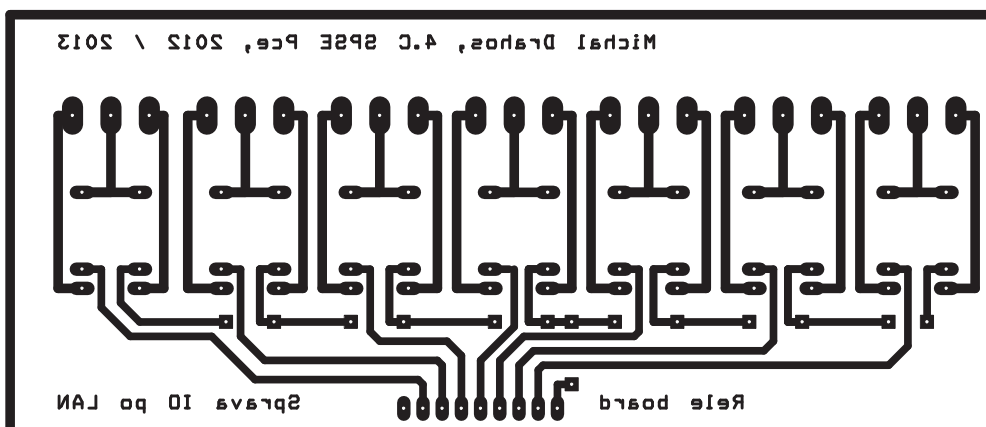
Příloha č. 7 - Předloha DPS a osazovací výkres výstupní desky, 1 list



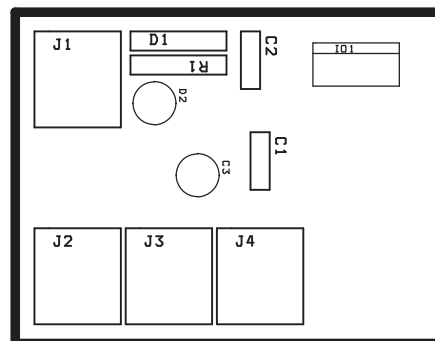
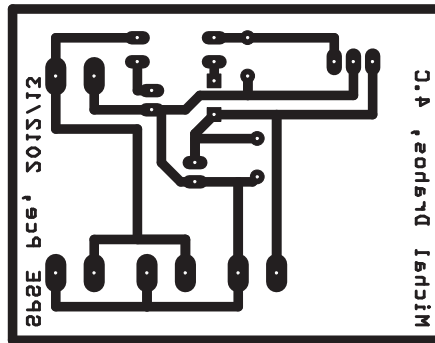
Příloha č. 8 - Předloha DPS a osazovací výkres vstupní desky, 1 list



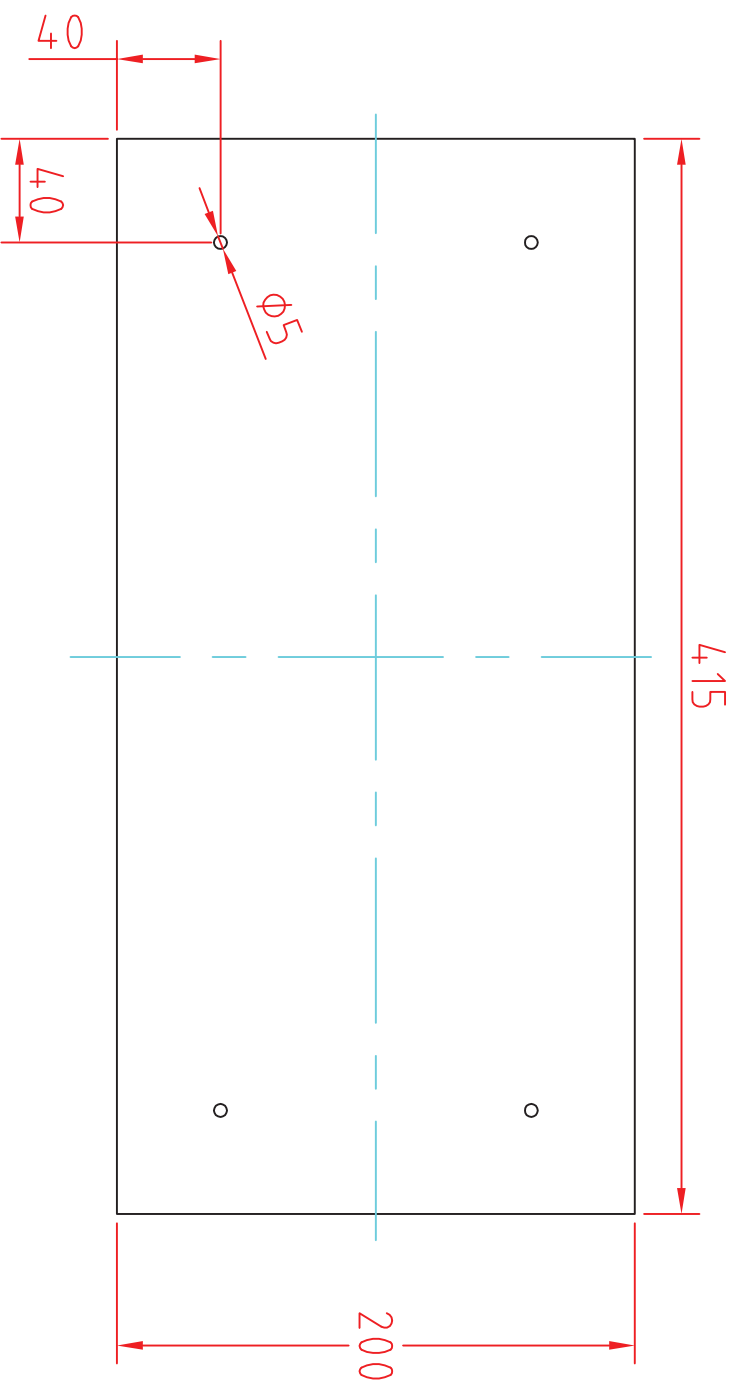
Příloha č. 9 - Předloha DPS a osazovací výkres relé desky, 1 list



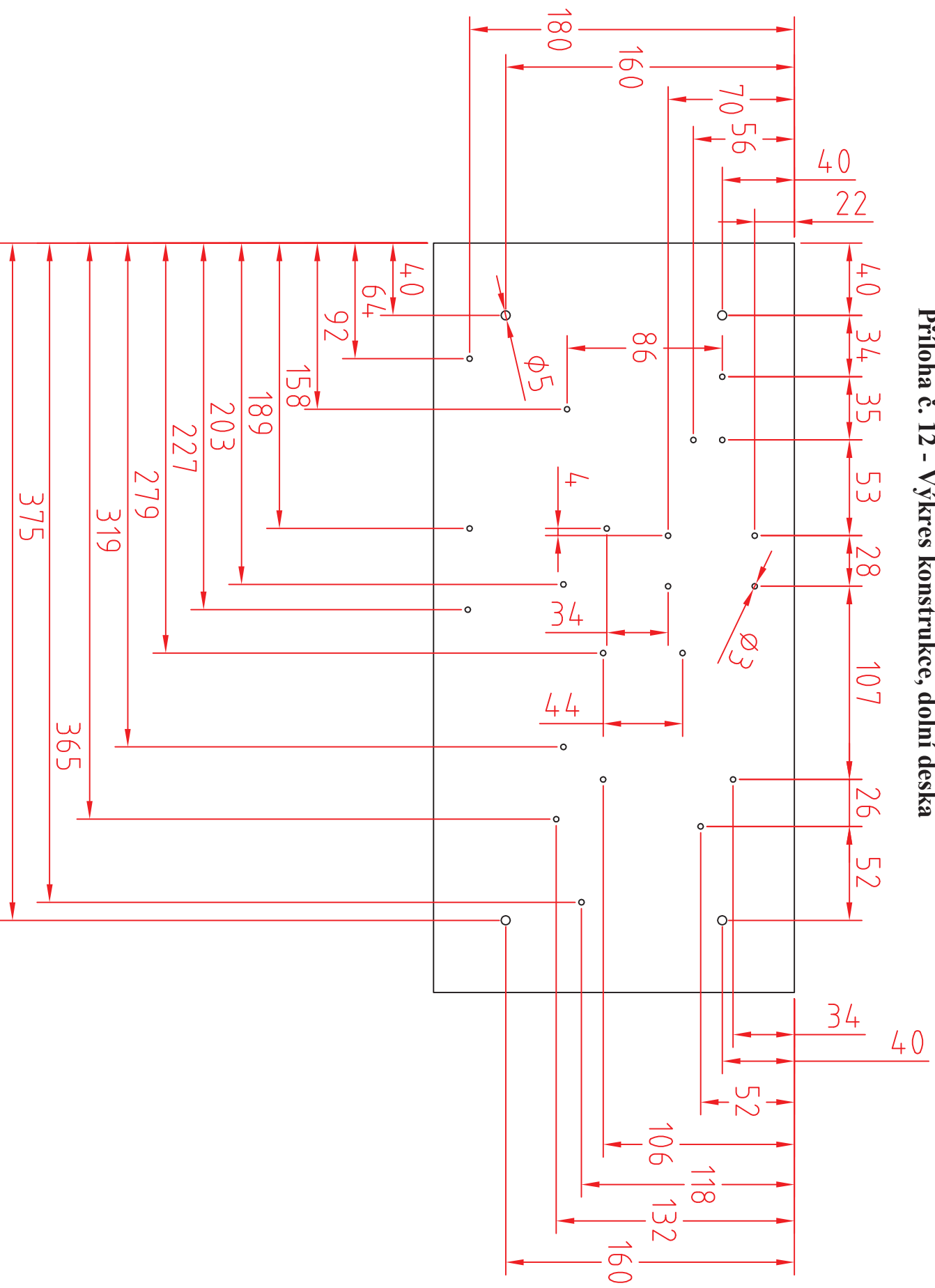
Příloha č. 10 - Předloha DPS a osazovací výkres napájecí desky, 1 list



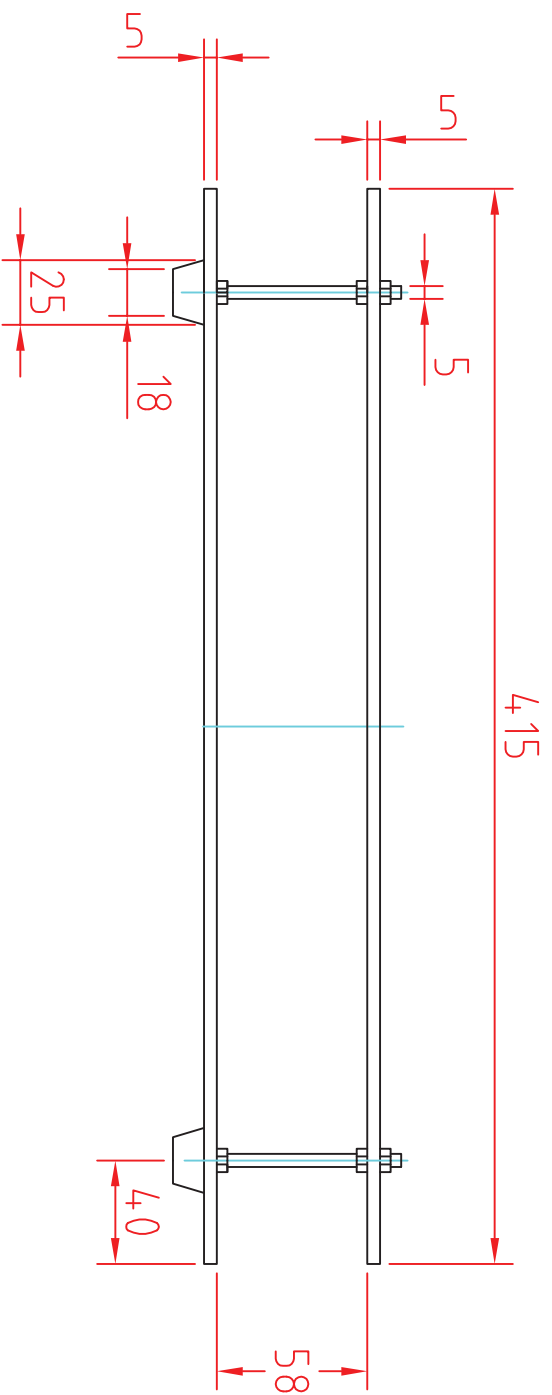
Příloha č. 11 - Výkres konstrukce, horní deska



Příloha č. 12 - Výkres konstrukce, dolní deska



Příloha č. 13 - Výkres konstrukce, bokorys



Příloha č. 14 – Rozpis součástek, list 1/2

Hlavní (core) deska

Označení	Součástka	Hodnota
B1	držák baterie do DPS	BH2032
C1	kondenzátor, elektrolyt.	4u7F
C2	kondenzátor, keramický	330nF
C3, C5	kondenzátor, keramický	100nF
C4	kondenzátor, keramický	470uF
D1	LED	L-53GD
D2	Zenerova dioda	BZX83V005.1
D3	dioda	1N4007
D4	dioda	1N4148
IO1	kit	STM32VLDISCOVERY
IO2	stabilizátor	78S05
IO3	Oboustran. kolík XINYA	-
J1	svorkovnice	-
J2, J3, J5	Oboustran. kolík XINYA	-
JP1, JP2	jumper	-
R1	odpor	330R

Výstupní deska

Označení	Součástka	Hodnota
C1, C6	kondenzátor, elektrolyt.	4u7F
C2 - C5	kondenzátor, keramický	100n
D2	LED	L-53GD
D3	Zenerova dioda	BZX79C3V3
D4	Zenerova dioda	BZX83V005.1
D6	dioda	1N4007
DB1	LED bar	L-BARGRAF G
IO1	Posuvný registr	M74HC595B1
IO2	Posuvný registr	M74HC595B1
IO3	Budič	ULN2003A
IO4	stabilizátor	78S05
J1	Oboustran. kolík XINYA	-
J3	svorkovnice	-
J4	Oboustran. kolík XINYA	-
J5	Oboustran. kolík XINYA	-
JP1 - JP4	jumper	-
R1 - R4	odpor	10k
R5 - R12	odpor	220R
R13	odpor	270R
U1	stabilizátor	MC33269DT-3.3

Příloha č. 14 – Rozpis součástek, list 2/2

Vstupní deska

Označení	Součástka	Hodnota
C1	kondenzátor, elektrolyt.	4u7F
C2	kondenzátor, keramický	100n
C3	kondenzátor, keramický	100n
D1	dioda	1N4007
D2	Zenerova dioda	BZX79C3V3
D3	LED	L-53GD
IO1	posuvný registr, PISO	SN74HC195N
IO2	posuvný registr, PISO	SN74HC195N
J1	svorkovnice	-
J2	svorkovnice	-
J3	Oboustran. kolík XINYA	-
JP1 - JP3	jumper	-
R1 - R3	odpor	10k
R4	odpor	270R
RN1	odporová síť, typ A	RR 8X10K 2%
RN2	odporová síť, typ A	RR 8X10K 2%
SW1	DIP spínač	DIP 08 BLUE
U1	stabilizátor 3V3	MC33269DT-3.3

Relé deska

Označení	Součástka	Hodnota
J1 - J7	svorkovnice	-
J8	Oboustran. kolík XINYA	-
LS1 - LS7	relé	N4100CHS3DC5AC

Napájecí deska

Označení	Součástka	Hodnota
J1 - J4	svorkovnice	-
C3	kondenzátor, elektrolyt.	470uF
C1	kondenzátor, keramický	100nF
C2	kondenzátor, elektrolyt.	330nF
D1	dioda	1N4007
D2	LED	L-53GD
R1	odpor	330R
IO1	stabilizátor	78S05

Příloha č. 15 - Ukázka obslužného programu, 1 list

Server v1.2.4 for I/O LAN controller [DEBUG]

Režim desky : administrátor
Status : Připraven.

Přijaté data : [Sun Mar 31 21:19:13 CEST 2013] Alarm vstup!
 Michal Drahoš, 2012/2013 SPŠE Pardubice, 4.C

Výstupy

N	Motor 1
N	Motor 2
N	Stykac 1
N	Stykac 2
N	Svetlo 1
N	Topeni
N	Vetrak

Vstupy

 	Cidlo dveře
 	Cidlo okno
 	Motor 1
 	Motor 2
 	Koncak 1
 	Koncak 2
 	
 	

Příloha č. 16 - Fotografie celého zařízení, 1 list

