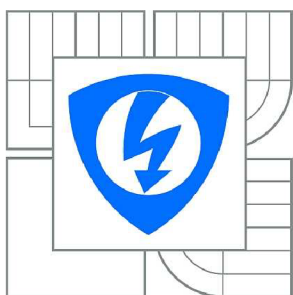


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

ŘÍDICÍ SYSTÉM TVÁŘECÍHO STROJE

FORMING MACHINE CONTROL SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

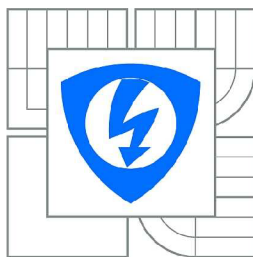
Bc. JIŘÍ VAŠKO

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PAVEL KUČERA, Ph.D.

BRNO 2012



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Jiří Vaško

ID: 106860

Ročník: 2

Akademický rok: 2011/2012

NÁZEV TÉMATU:

Řídicí systém tvářecího stroje

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte hardware a software pro řízení tvářecího stroje na výrobu cukrářských formiček. Koncepte obsahuje návrh řídicí desky pro stávající sběrnici a I/O karty OMRON SYSMAC, programové vybavení řídicího CPU a rozhraní pro komunikaci s PC. Nastavení řídicího systému bude prováděno aplikací v PC. Při návrhu řídicího systému věnujte pozornost problematice bezpečnosti a diagnostice tvářecího stroje.

DOPORUČENÁ LITERATURA:

[1] Barr, M., Massa, A.: Programming Embedded Systems: With C and GNU Development Tools. O'Reilly Media; 2 edition (October 1, 2006). ISBN-13: 978-0596009830.

[2] Prosis, J.: "Programování ve Windows pomocí MFC." 2. vydání. Computer Press, Praha, 2002. ISBN 80-7226-309-9.

Termín zadání: 6.2.2012

Termín odevzdání: 21.5.2012

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

Konzultanti diplomové práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Tato diplomová práce se zabývá realizací řídicího systému pro tvarovací stroj na výrobu cukrářských forem. Úvodní část představuje výrobní stroj a popisuje jeho důležité prvky z hlediska řízení. Druhá část se zabývá návrhem hardware řídicího automatu, popisem funkčních celků řídicí desky a ostatních desek. Následující pasáže představují komunikační protokol mezi automatem a PC, a poté firmware CPU řídicí desky. V závěrečné části je provedena charakteristika obslužné PC aplikace pro spojení s automatem.

KLÍČOVÁ SLOVA

automat, tvarovací stroj, ARM, Cortex M3, C#/.net, JTAG, SD karta, FAT32

ABSTRACT

This master thesis deals with the implementation of the control system for the forming machine used in producing of confectionery forms. The introductory part describes the forming machine from the point of important control elements. The second part is associated with the hardware design of the control automat and description of main functional units on the control PCB, as well as on other PCBs. The following text presents the communication protocol between the control automat and PC and after that there is a description of the CPU firmware. The final section brings the characteristics of the PC application for a connection with the automat.

KEYWORDS

automat, forming machine, ARM, Cortex M3, C#/.NET, JTAG, SD card, FAT32

VAŠKO, Jiří *ŘÍDICÍ SYSTÉM TVÁŘECÍHO STROJE*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2012. 138 s. Vedoucí práce byl Ing. Pavel Kučera, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „ŘÍDICÍ SYSTÉM TVÁŘECÍHO STROJE“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce panu Ing. Pavlu Kučerovi, Ph.D. za vypsání tohoto tématu, cenné rady a pomoc, jež vedla k současné podobě této práce.

Brno

.....

(podpis autora)

OBSAH

1	Úvod	11
2	Popis zařízení	12
2.1	Základní přehled	12
2.2	Prvky zařízení	13
2.2.1	Napájení	13
2.2.2	Obvod CentralSTOP	14
2.2.3	Obvod řídicího napětí	14
2.2.4	Hydraulický agregát	15
2.3	Funkce zařízení	15
2.3.1	Inicializace zařízení	15
2.3.2	Start cyklu	16
2.3.3	Běh cyklu	17
2.3.4	Události, Poruchy	18
3	Návrh hardware řídicího modulu	21
3.1	Základní přehled	21
3.2	Napájení	22
3.3	CPU	27
3.4	Připojení na sběrnici, USB	29
3.5	Komunikační obvody	32
3.6	Analogové vstupy	35
3.7	CPU WatchDog obvod	38
3.8	Připojení SD karty	40
4	Návrh hardware 8bitové 24VDC DO karty	42
4.1	Základní přehled	42
4.2	Logické obvody	42
4.3	Výstupy	44
5	Komunikační protokol	48
5.1	Fyzická vrstva - RS-232	48
5.2	Linková vrstva	50
5.3	Síťová vrstva	52
5.4	Aplikační vrstva	53

6 Firmware CPU řídicí desky	59
6.1 Architektura CORTEX-M3	59
6.2 Vytvořené zdrojové soubory	61
6.3 RTOS CooCOX CoOS	62
6.4 Implementace firmware	64
7 PC aplikace	72
7.1 Základní charakteristika	72
7.2 Programové řešení	74
7.3 Grafické rozhraní (GUI)	80
8 Závěr	88
Literatura	89
Seznam příloh	91
A Elektronická příloha	92
B Řídicí modul	93
B.1 Kompletní schéma zapojení	93
B.2 DPS	102
B.3 Seznam součástek	107
B.4 Fotodokumentace osazených desek	112
C 8bitová 24VDC DO karta	115
C.1 Schéma zapojení	115
C.2 DPS	119
C.3 Seznam součástek	123
D Sestava automatu	125
E Řídicí rozvaděč	127

SEZNAM OBRÁZKŮ

2.1	Blokové schéma tvarovacího stroje	12
3.1	Blokové schéma obvodů hardware řídicího modulu	21
3.2	Funkce a zapojení obvodu MC34161 v režimu zpožděného resetu (lit.[5])	24
3.3	Řídicí modul 1/8 - přívodní napájecí část, napájení vany a CPU . . .	25
3.4	Řídicí modul 2/8 - napájení komunikačních obvodů a analog. vstupů .	26
3.5	Řídicí modul 3/8 - připojení CPU, připojení kontrolky	28
3.6	Zápis na výstupní kartu, čtení ze vstupní karty	30
3.7	Řídicí modul 4/8 - připojení CPU na sběrnici ve vaně, USB(rezerva) .	31
3.8	Řídicí modul 5/8 - galvanicky oddělená komunikace RS-232, RS-485 .	34
3.9	SPI pro komunikaci s ADC převodníkem (lit.[7])	36
3.10	Řídicí modul 6/8 - galvanicky oddělené analogové vstupy 0 - 20 mA .	37
3.11	Opatření při resetu CPU	38
3.12	Řídicí modul 7/8 - obvod CPU WatchDogu pro obvod CentralSTOP .	39
3.13	Řídicí modul 8/8 - připojení SD karty pomocí softwarového SPI . . .	41
4.1	Blokové schéma výstupní karty	42
4.2	Zakázané pásmo budících LED optočlenů	43
4.3	Výstupní karta 1/3 - ovládací logika karty	45
4.4	Výstupní karta 2/3 - zapojení výstupů DO0 - 3	46
4.5	Výstupní karta 3/3- zapojení výstupů DO4 - 7, LED pro indikaci stavu	47
5.1	Propojení mezi PC a řídicím modulem pomocí 9pinových D-Sub ko- nektorů	48
5.2	Odeslání bajtu o hodnotě 61h pomocí UART při nastavení 8N1 (lit.[9])	49
5.3	Transakce iniciovaná PC aplikací a CPU řídicího modulu	51
6.1	Pipelining (lit.[3])	59
6.2	Paměťový prostor Cortex-M3 (lit.[10])	60
6.3	Procesy běžící v systému	64
6.4	Inicializace prostředků a RTOS CoOS Main()	65
6.5	Obslužná rutina TIMER0_Handler() přerušení časovače Timer0 , hlí- dání přívodního napájení	66
6.6	Obslužná rutina WDT_IRQHandler() při přetečení WatchDog	66
6.7	Úloha taskControl obsluhující řízení procesu	67
6.8	Úloha taskUART_receive pro příjem dat ze sériové linky (UART3)	68
6.9	Úloha taskUART_send odeslání rámců po sériové lince (UART3) . .	69
6.10	Úloha taskLog pro zápis logu na SD kartu	70
6.11	Průběh tlaku v hydraulickém obvodu ke konci kroku výrobního cyklu	71
7.1	Blokové schéma odeslání dat (výrobního cyklu)	77
7.2	Blokové schéma příjmu dat	78

7.3	Blokové schéma uložení programu do XML souboru a načtení programu	79
7.4	Jednotlivé položky menu aplikace (MDIMain)	81
7.5	Hlavní okno aplikace (MDIMain)	82
7.6	Okno nastavení COM portu (FormConnectSetting)	83
7.7	Okno zobrazení převážně statistických hodnot (FormStatistics)	84
7.8	Historie procesních událostí (FormProcessEvent)	85
7.9	„Vyskakovací“ okno procesní události (FormProcessEventInfoWindow)	85
7.10	Vizualizace procesních událostí	86
7.11	Okno nastavení CPU dat (FormControlCPUDataSettings)	87
7.12	Informace o aplikaci (AboutBox1)	87
B.1	Řídicí modul - sestava řídicí desky a přídavných desek	113
B.2	Řídicí modul - čelní panel	114
D.1	Sestava automatu	125
D.2	Vana pro modulární automat	126

SEZNAM TABULEK

2.1	Podmínky běhu zařízení	16
2.2	Režimy běhu zařízení	16
2.3	Podmínky ukončení kroku	18
2.4	Přehled událostí a poruch	20
5.1	Napěťové úrovně linky RS-232 (lit.[9])	49
5.2	Rámec linkové vrstvy	52
6.1	Seznam vytvořených zdrojových a hlavičkových souborů	61

1 ÚVOD

V automatizaci výrobních strojů a linek se v současnosti nejvíce využívá programovatelných logických automatů (PLC), které mohou být v kompaktní nebo modulární formě. Kompaktní automat obvykle disponuje všemi základními částmi (napájecí část, řídicí CPU jednotka, I/O porty, komunikační porty) na jedné nerozebíratelné chassis. Koncepce modulárního automatu, jak už vyplývá z názvu, umožňuje sestavit a jednotlivě propojit jednotlivé části neboli moduly dle použití v konkrétní aplikaci. K tomuto účelu slouží tzv. vana (rack), do které se jednotlivé moduly vkládají. Vana obsahuje obvykle komunikační sběrnici a napájecí vodiče.

Pro základní funkci modulárního automatu postačí přítomnost napájecího modulu, řídicího modulu a I/O karet pro spojení se samotným řízeným procesem. Nepostradatelnou součástí dnešních PLC je taktéž komunikační rozhraní, které lze přiřadit jako součást řídicího modulu, nebo ve formě modulu s vlastním komunikačním procesorem. Ovšem kterak souvisí předchozí řádka s touto diplomovou prací?

Cílem předkládané diplomové práce bylo vytvoření řízení tvářecího stroje pro výrobu vykrajovacích forem za určitých požadavků a prerekvizit. Na základě funkce stroje (kapitola 2) bylo pro vypracování diplomové práce předloženo zapojení řídicího rozvaděče, spektrum čidel a akčních prvků. Pro využití byly přímo určeny vana a část I/O karet dnes již zastaralého modulárního automatu OMRON SYSMAC SCY-M1 z éry 80.let minulého století. V rámci této práce byl tedy proveden návrh hardware řídicího modulu pro ovládání karet a implementace firmware řídicího CPU. Dále bylo nutné vytvořit chybějící I/O karty. Vznikl tak řídicí automat na míru požadavkům tvářecího stroje jako jednoúčelového zařízení.

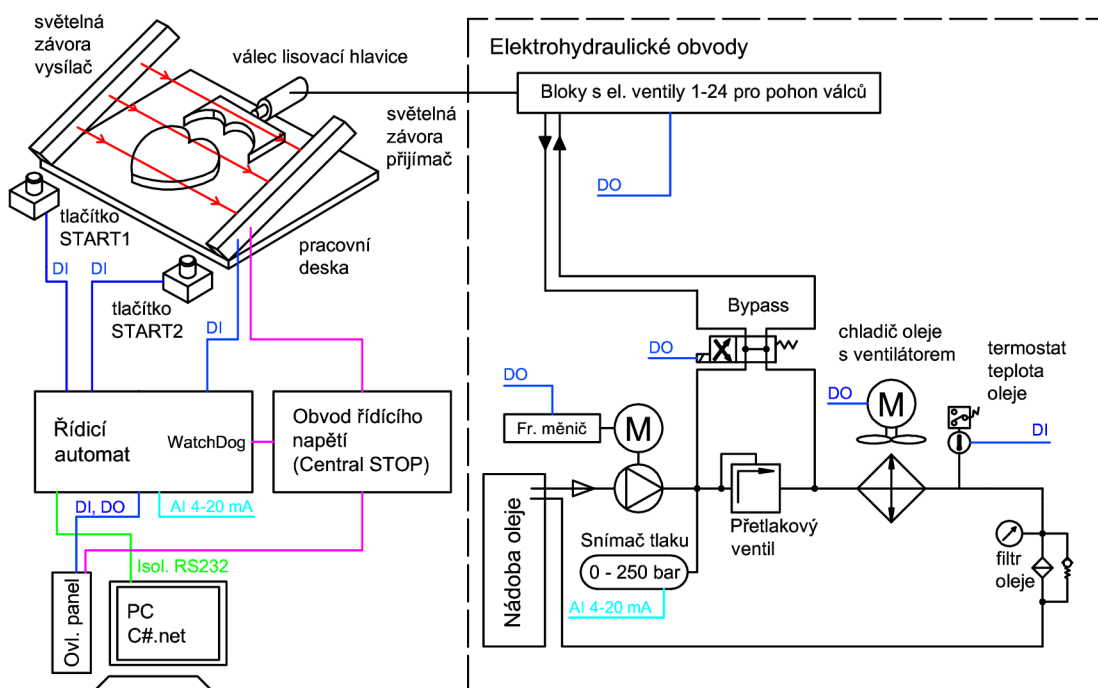
Navržený řídicí automat musí splňovat požadavky na efektivitu výroby a hlavně bezpečný provoz pro obsluhu. Nastavení podoby a parametrů výrobního procesu, diagnostika procesních poruch, stejně jako záznam o provedené výrobě musí být zajištěny pomocí PC aplikace. K tomuto účelu byla realizována komunikace s řídicím modulem pomocí galvanicky oddělené RS-232 a pro PC byla vytvořena aplikace na platformě C#/.NET běžící pod operačním systémem MS Windows XP (kapitola 7). Je nutné zdůraznit, že veškeré kritické řízení je realizováno přímo v CPU řídicího modulu automatu, jakékoli selhání PC neovlivní tedy bezpečné odstavení stroje.

2 POPIS ZAŘÍZENÍ

2.1 Základní přehled

Tvarovací stroj se jako komplexní zařízení skládá z několika funkčních celků, což je schematicky zobrazeno na obrázku 2.1: pracovní deska, elektrohydraulické obvody, řídicí obvody, ovládací panel a PC.

Vyfrézovaný tvar předlohy výrobku a hydraulické válce s tvarovanými hlavicemi jsou umístěny pomocí přípravků na pracovní desce (12mm ocelový plech). Princip výroby formičky spočívá v ohýbání proužků olemovaného plechu přes protikus specifického tvaru pomocí hlavic na hydraulických válcích. Jednotlivé válce jsou spouštěny v určitém pořadí, aby proces postupoval podélně od jednoho konce plíšku a nedošlo ke zničení materiálu. Proces výroby jedné formy je nazýván jako výrobní cyklus, v rámci této diplomové práce byl proveden návrh výrobního cyklu v podobě popsané níže. Z bezpečnostních důvodů je nad deskou umístěna světelná závora, která přeruší běh zařízení při zakrytí v době výrobního cyklu. Start výrobního cyklu se provádí tlačítky START1/2.



Obr. 2.1: Blokové schéma tvarovacího stroje

Celkově jsou osazeny tři bloky po osmi elektromagnetických ventilech, při sepnutí

napájení se připojený hydraulický válec vysouvá, při vypnutí válec zajíždí. Motor hydraulického agregátu je buzen z frekvenčního měniče, což umožňuje nastavit z automatu frekvenci napětí motoru dle potřebného průtoku oleje. Rozepnutím bypassu dochází k natlakování obvodu na tlak nastavený na přetlakovém ventilu. Tlak je měřen pomocí tlakového snímače připojeného k analogovému vstupu AI1 řídicího modulu automatu. Hydraulický agregát dále obsahuje termostat pro měření teploty oleje, jež je připojen k DI automatu a automat spíná přes stykač motor ventilátoru chladiče oleje. Každý hydraulický obvod musí být opatřen filtrem, obvod filtru je řešen tak, aby upozornil na jeho zanesení pomocí tlakového budíku.

Řídicí rozvaděč obsahuje vytvořený modulární automat, který je pomocí digitálních I/O připojen k ovládacímu panelu, umístěnému bezprostředně vedle monitoru nainstalovaného PC. Připojení řízených prvků k automatu schematicky znázorňuje obrázek 2.1. Jak ale vznikla idea návrhu vlastního řídicího automatu?

Původní myšlenka řízení stroje, která časově předcházela zadání této diplomové práce, počítala s variantou nasazení PLC některého renomovaného výrobce. Objevili se ovšem dva důležité faktory, které vyústili v použití dílů zastaralého modulárního automatu a vývoje řídicího modulu. Nejprve byly plánovány elektromagnetické ventily s jmenovitým napětím 24 VDC, které by musely být buzeny vyšším proudem z výkonnějších DO karet automatu. S ohledem na podstatně vyšší cenu ventilů i DO karet bylo přistoupeno k napětí ventilů 110 VAC a využití 8bitových triakových karet, kterými již bylo disponováno.

Druhým faktorem byl požadavek specifické počítačové aplikace, která bude s řídicím automatem komunikovat. Obsluha musí mít možnost v GUI aplikaci vytvářet nové sekvence, archivovat použité, a hlavně aktuální požadovanou sekvenci nahrát prostřednictvím komunikačního rozhraní do řídicího automatu. To bylo možno jednoduše implementovat v prostředí MS Windows tak, aby komunikace s řídicím automatem byla bezprostředně na míru potřebám aplikace.

2.2 Prvky zařízení

Tato kapitola popisuje funkce a vybavení zařízení, zapojení ovládacích prvků obsahuje schéma řídicího rozvaděče v příloze E na str. 127.

2.2.1 Napájení

Zařízení je napájeno z trojfázové soustavy 3x400 VAC 50Hz bez využití nulového vodiče, a to prostřednictvím hlavního vypínače Q1 (příl. E, str. 128). Vestavěný toroidní oddělovací transformátor T1 s primárním vinutím 400 VAC poskytuje na sekundáru 230 VAC pro napájení PC (zásuvka Z1), monitoru (Z2) a průmyslového

zdroje 24 VDC. Odbočka na sekundárním vinutí dále napájí elektromagnetické ventily napětím 110 VAC (L110). Silový 3f rozvod (L1, L2, L3) určený pro napájení měniče motoru čerpadla hydraulického agregátu a motoru chlazení oleje, je spínán přes redundantní sériovou kombinaci dvou stykačů K2.6 a K2.7 (kap. E, str. 129). Dalších spínacích kontaktů těchto stykačů je využito pro sepnutí přívodu 110 VAC pro napájení elektromagnetických ventilů (výstupy automatu O4.0 - O7.7).

2.2.2 Obvod CentralSTOP

Bezpečnostní relé *Crouzet KNA3* je zapojeno dle doporučení výrobce (příl. E, str. 129). Stykače K2.6, K2.7 připojeny na svorky bezpečnostního relé jsou ponechány napájeny, pokud je obvod CentralSTOP spojený. V překladu to znamená splnění několika podmínek. Bezpečnostní tlačítko S0.0 není zmáčknuto a WatchDog relé řídicího automatu K1.7, K1.8 jsou sepnuta. Další podmínka je uvedena v následujícím odstavci.

Světelná závora disponuje dvěma výstupy, které ovládají pomocná relé K2.3, K2.4, jejichž jedny spínací kontakty jsou zapojeny do obvodu Central STOP a druhé do vstupu automatu I0.1. Zakrytá světelná závora odpojí relé. Protože není žádoucí, aby po dokončení cyklu a vyjímání výrobku proběhlo rozpojení obvodu CentralSTOP (a s tím související odpojení řídicího napětí), je nutné kontakty obou relé závory překlenout. K tomuto účelu slouží relé K14.7 a K14.8. buzena z výstupů O3.6 a O3.7 (příl. E, str. 134). Aby naopak nedošlo k situaci, kdy bude spuštěn pohyb hydraulických válců, aniž by překlenutí K14.7 a K14.8 odpadlo, vyhodnocuje automat na vstupu I0.6 (příl. E, str. 131) zpětné hlášení od rozpínacích kontaktů relé K14.7 a K14.8.

Pokud nějakým způsobem vznikne chyba v běhu aplikace CPU řídicího modulu automatu, dojde na základě resetu vyvolaného obvodem WatchDog v CPU k odpojení relé K1.7, K1.8., která jsou připojena přímo k výstupu „WDOOut“ na řídicím modulu automatu. Rovněž pokud analyzuje automat poruchu, která vyžaduje odpojení silových částí, odpojí WatchDog relé. Pravděpodobnost, že by zůstal obvod CentralSTOP spojen se tímto snižuje.

2.2.3 Obvod řídicího napětí

Při splnění podmínky spojení obvodu CentralSTOP lze provést nahození stykačů K2.6 a K2.7, připojených k bezpečnostnímu relé *Crouzet KNA3*, pomocí tlačítka S1.0 „Řídicí napětí zapnuto“. Spínací kontakty stykačů K2.6 a K2.7 spínají obvod řídicího napětí 24 VA (indikováno signálkou H2.8) pro napájení výstupů (O3.0 -

O3.7) a dále spínají silové části jako pohony (L1, L2, L3) a ventily (L110). Uvedeno zde (příl. E, str. 129).

Řídící modul, vstupní karty I0.0-0.7, I1.0-I1.7 a výstupy O2.0 - O2.7 (signálky) jsou napájeny stále. Automat tak vyhodnotí zdroj rozpojení obvodu CentralSTOP a informuje obsluhu pomocí kontrolky H13.5 o poruše. Rovněž zašle zprávu o poruše PC aplikaci (viz 2.3.4).

2.2.4 Hydraulický agregát

Frekvenční měnič U3.1 (příl. E, str. 130) napájí motor čerpadla hydraulického agregátu (dále jen HA), zpětné hlášení o správné funkci měniče a stavu motorového spouštěče před měničem Q3.1 je přivedeno na vstup I1.0. Výstup O3.0 (příl. E, str. 134) dává měniči povel „Jed“. V měniči je přednastaveno 16 rychlostí (adresa 0 - 15), jež jsou adresovány výstupy O3.1 - O3.5. Řazení konkrétní rychlosti vyplývá z požadovaného průtoku oleje. Motor ventilátoru chlazení oleje je spouštěn výstupem O3.5 přes stykač K14.6 na určitou dobu, pokud termostat na hydraulickém agregátu sepne vstup I1.2. Pro doplnění, výstup O7.7 sepnutím ventilu rozepne bypass hydraulického obvodu.

2.3 Funkce zařízení

2.3.1 Inicializace zařízení

Po spuštění napájení nastartuje řídicí automat, je nutné spustit zabudované PC. Podoba výrobního cyklu není ukládána v permanentní paměti, je nutné ji po startu zařízení nahrát prostřednictvím PC aplikace do automatu. Toto řešení je implementováno, aby nedošlo po startu k případnému spuštění jiného cyklu, než na který je osazena pracovní deska. PC aplikace taktéž umožňuje nahrát do automatu tzv. manuální krok, při němž dojde pouze k vysunutí určitého výrobního kroku (bude vysvětleno dále).

Nutnou podmínkou běhu stroje je nahozený obvod řídicího napětí (viz 2.2.2) a s tím spojená přítomnost napájení silových částí (HA, ventily). Rovněž musí být vybrán jeden z dostupných režimů běhu zařízení (viz Tab. 2.2). Shrnutí základních podmínek běhu stroje tak, jak jsou vyhodnocovány řídicím automatem, obsahuje Tab. 2.1.

Během výrobního cyklu nereaguje automat na přepnutí režimu, přenastavení režimu probíhá po skončení cyklu. Po změně režimu nebo nahrání nového výrobního cyklu proběhne reinicializace, při níž se zasunou všechny hydraulické válce (musí být splněny všechny podmínky běhu vyjma poslední).

Logický vstup	Logická hodnota	Popis
I0.0	0	Bezpečnostní tlačítko S0.0 uvolněno
I1.0	1	Měnič U3.1 připraven
I1.3	1	Ochrana Q3.7 ventilátoru oleje v pořádku
I0.5, I0.4		Vybrán dostupný pracovní režim pomocí přepínače S11.4 (viz Tab. 2.2)
		Hydraulický agregát spuštěn (spustit/vypnout pomocí vstupu I1.1)
		Výrobní cyklus / Manuální krok nahrán (dle režimu)

Tab. 2.1: Podmínky běhu zařízení

Režim	Režim běhu přepínač S11.4 (I0.5, I0.4)	Spouštění cyklu tlačítka START2(I0.3), START2(I0.2)	Popis
M1	10	START1 a START2	Režim obouručního automatického ovládání výrobního cyklu (v provozu uzamčen S11.4 v této poloze - běžný personál)
M2	00	START1	Režim jednoručního automatického ovládání výrobního cyklu, cyklus spuštěn náběžnou hranou START1 (pověřené osoby)
M3	01	START1	Režim krokování výrobního cyklu, vyjetí kroku na náběžnou hranu START1 (pověřené osoby)
M3	01	START2	Režim manuálního kroku, aktivní na úroveň START2 (1 - vyjeto, 0 - zajeto) (pověřené osoby)
	11	XX	nesmyslný režim - porucha

Tab. 2.2: Režimy běhu zařízení

2.3.2 Start cyklu

Provoz zařízení v režimech M1, respektive M2 provádí obsluhu výrobního cyklu. Start je možný při splnění výše uvedených podmínek běhu. V běžném režimu M1 lze cyklus startovat pouze dvouručním ovládáním tlačítka START1 a START2 (viz Tab. 2.2), přičemž maximální možná prodleva mezi zmáčknutím tlačítek odpovídá

době 500 ms. Režim M2 může být provozován pouze pověřenými pracovníky, jinak se obsluha obou režimů neliší, pro další účely popisu cyklu bude uvažován režim M1.

Výrobní cyklus volí obsluha následovně: „S výchozí polohou“, respektive „Bez výchozí polohy“, dle požadovaného chování prvního kroku. V prvním případě automat pomocí dvojice relé realizuje „překlenutí“ kontaktů relé světelné závory, aby zakrytím světelné závory nebyl rozpojen obvod CentralSTOP. Toto umožňuje, v případě výroby určitých tvarů, zaaretování, chce-li přimáčknutí, plíšku hydraulickým válcem při přidržení tohoto plíšku obsluhou. Vyjetí zmíněného válce (obvykle bývá jeden) v prvním kroku do popsané „Výchozí polohy“ se provádí na zmáčknutí tlačítka START1. Rychlost výsuvu je nastavena na měniči s ohledem na bezpečnost.

Sepnutím startovacích tlačítek lze spustit automatický běh cyklu. Stejně je tomu i v případě cyklu „Bez výchozí polohy“, kde je startovacími tlačítky nastartován přímo automatický běh cyklu. Automat ovšem musí odepnout kontakty relé K14.7, K14.8 pro překlenutí kontaktů relé K2.3, K2.4 světelné závory v obvodu CentralSTOP, to je zkontrolováno zpětným hlášením na vstupu I0.6. Až poté je fakticky spuštěn cyklus.

2.3.3 Běh cyklu

Motor hydraulického agregátu běží mimo výrobní cyklus na nastavenou základní rychlost, bypass je spojen a v obvodu není tlak. V cyklu je bypass rozepnut a automat pomocí čtyřech DO přeřazuje přednastavené rychlosti v měniči na adresách 0-15 dle závislosti na počtu válců Δ , které se v aktuálním kroku pohybují. Maximální počet válců je 31, přičemž 24 je aktuálně osazeno. Změna počtu válců Δ tedy nabývá hodnot z rozsahu $\langle 0; 24 \rangle$, přičemž závislost rychlosti r (rychlost v tomto pojetí představuje adresu v měniči) byla po sérii testů optimalizována do následující podoby (2.1). V rovnici figuruje proměnná *Offset* jakožto v PC volitelný parametr cyklu.

$$r(\Delta) = 3\Delta + \textit{Offset} \begin{cases} \exists r(\Delta) > 15 \implies r(\Delta) = 15 \\ \Delta \in \langle 0; 24 \rangle \\ \textit{Offset} \in \langle 0; 5 \rangle \end{cases} \quad (2.1)$$

Krok cyklu musí být v automatu nějakým vhodným způsobem ukončen. Trvání kroku je omezeno shora implicitně nastaveným časem 5 s pro případ, že nebylo dosaženo jiné podmínky ukončení, respektive jiná podmínka ukončení nebyla vůbec definována. Tuto komplikovanou definici lze osvětlit výčtem jednotlivých variant ukončení kroku i s prioritou v Tab., neboť je možné z PC nastavit pro jeden konkrétní krok více ukončovacích podmínek.

Nyní je nutné ještě vysvětlit principy ukončení kroku na koncový snímač a tlak.

Čtyři indukčnosti koncové snímače jsou připojeny ke vstupům I1.4-I1.7. Snímače mohou být volitelně připevněny na pracovní desku ke konkrétním hydraulickým válcům, jejichž vyjetí má být omezeno. Zasažení koncového snímače je vyhodnoceno automatem jako ukončení aktuálního kroku.

Pro ukončení kroku na základě hodnoty tlaku se provádí měření průběhu tlaku pomocí tlakového čidla S1.8 připojeného na svorku AI1 řídicího modulu automatu (příl. E, str. 128). Při opření hlavice na pístnici hydraulického válce proti matici (tj. předloze výrobku) vzroste prudce tlak v hydraulickém obvodu. Velikost překmitnutí tlaku v okamžiku nárazu, než zareaguje přetlakový ventil, je závislá na velikosti průtoku oleje a z praxe bylo zjištěno, že špičkové hodnoty pochází z intervalu 40-70 bar. Pro snížení opotřebování nářadí je žádoucí, aby nedocházelo k velkým nárazům, proto automat při přesahu určitého tlaku provede ukončení kroku. Prahová hodnota tlaku je zjištěna automatem na základě měření v předchozích cyklech a pokud není hodnota tlaku prahu dosažena, je prahová hodnota přenastavena.

Výhodou ukončení na tlak je taktéž rychlejší nastavení parametrů nového výrobního cyklu obsluhou, tj. není nutné testovat, jak dlouho daný krok trvá a nastavovat explicitně čas pomocí PC aplikace. Rovněž viskozita hydraulického oleje se mění během provozu a doba nutná pro trvání kroku není tudíž konstantní.

Nastavené varianty ukončení	Priorita ukončení
žádné	1. implicitní čas 5 s
koncový snímač	1. koncový snímač, 2. implicitní čas 5 s
explicitní čas	1. explicitní čas, 2. implicitní čas 5 s
koncový snímač, explicitní čas	1. koncový snímač, 2. explicitní čas, 3. implicitní čas 5 s
tlak	1. tlak, 2. implicitní čas 5 s
tlak, explicitní čas	1. tlak, 2. explicitní čas, 3. implicitní čas 5 s
koncový snímač, tlak, explicitní čas	1. koncový snímač, 2. explicitní čas, 3. implicitní čas 5 s

Tab. 2.3: Podmínky ukončení kroku

Poslední krok musí být definovaný v PC aplikaci tak, že všechny ventily jsou vypnuty, tj. válce se zasouvají. Jinak obsluha zvolí obvyklým způsobem, jako výše, ukončení tohoto kroku. Bypass se sepne, obvodem proudí olej bez tlaku. Světelná závora je pomocí relé ovládaných z automatu překlenuta, obsluha může výrobek vyjmout. Automat odešle do PC aplikace informaci o ukončení cyklu. Následuje opět procedura nového startu cyklu.

2.3.4 Události, Poruchy

Pro toto zařízení lze definovat několik typů událostí, při jejichž výskytu dochází k odlišným opatřením automatu. Informace o výskytu události je logována v au-

tomatu na SD kartu a rovněž je odeslána na PC, kde jsou podrobnosti zobrazeny. Rozdělení událostí je následující:

- Informace
 - při výskytu proběhne odeslání do PC
 - bez opatření
- Varování
 - při výskytu proběhne odeslání do PC
 - bez opatření
- Porucha nekritická
 - neovlivňuje bezpečný provoz zařízení
 - výskyt zachycen a řešen automatem
 - při výskytu proběhne odeslání do PC
- Porucha kritická
 - ovlivňuje bezpečný provoz zařízení
 - musí být rozpojen obvod CentralSTOP, tzn. odpojení obvodu řídicího napětí a odpojení silových částí
 - automat rozepne WatchDog (rozpojení obvodu CentralSTOP)
 - DO O3.0-O7.7 do log.0
 - signalizace pomocí kontrolky H13.5 připojené na výstup O2.4
 - při výskytu proběhne odeslání do PC

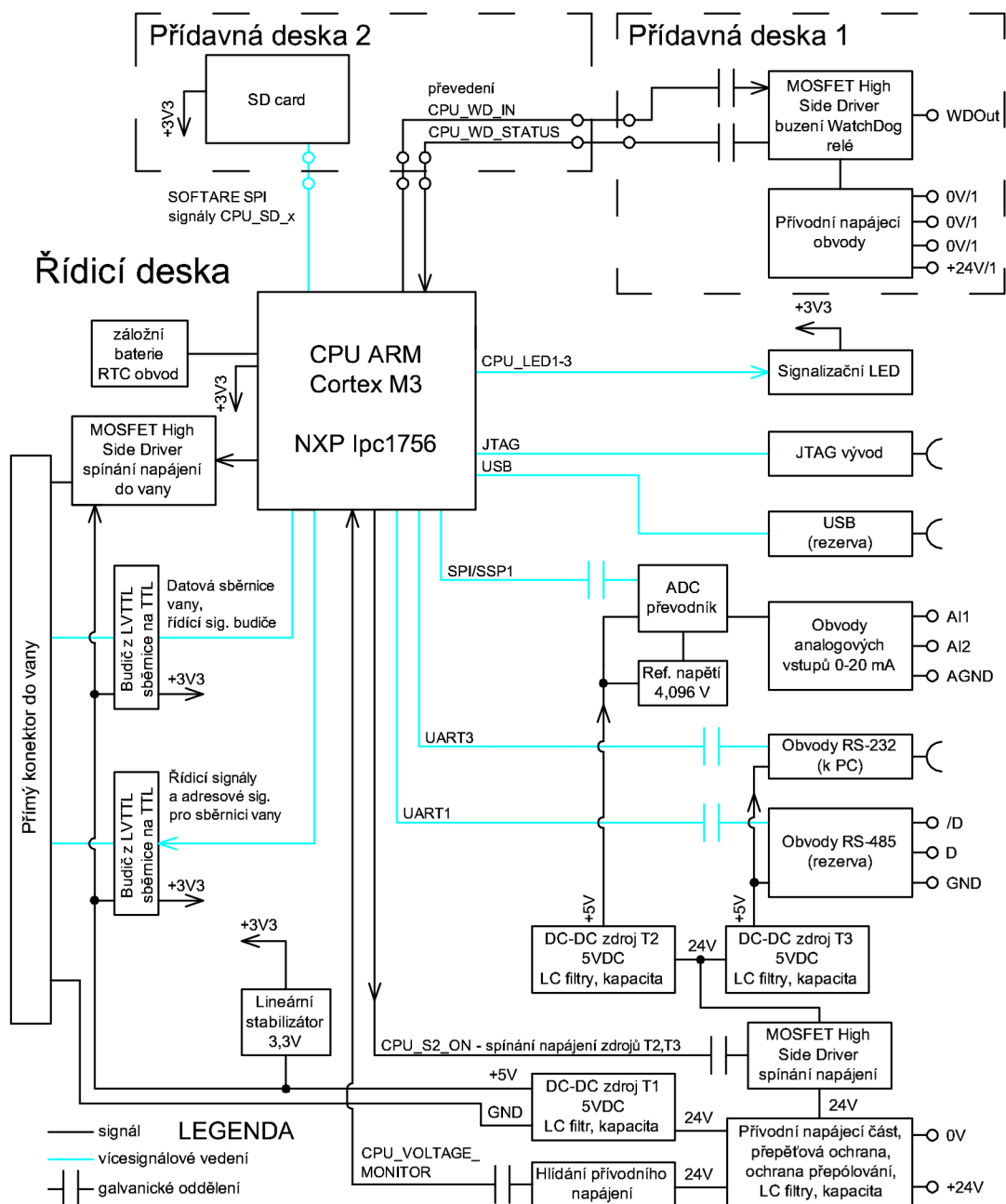
Přehled konkrétních událostí obsahuje Tab. 2.4. Kvitace poruchy se provádí při-
držením tlačítka S11.7 (I0.7), což vyvolá nejprve sepnutí WatchDog relé automatu
v obvodu CentralSTOP, aby bylo nahozeno řídicí napětí a silové části. Pokud ne-
vzniknou nové příčiny poruch, porucha je zkvitována.

Kód	Typ	Popis
0	Informace	Kvitace poruchy provedena pomocí tlačítka S11.7 (I0.7)
1	Porucha kr.	Výběr nedostupného režimu, přepínač S11.4 (I0.4,I0.5)
2	Porucha kr.	Zakrytí světelné závory během výrobního cyklu (I0.1)
3	Porucha kr.	Bezpečnostní tlačítko zamáčknuto (I0.0)
4	Porucha kr.	Měnič U3.1 porucha (I1.0)
5	Porucha kr.	Ventilátor chlazení oleje motorový spouštěč Q3.7 porucha (I1.3)
6	Porucha kr.	Kontakty relé K14.7 nebo K14.8 neodlepeny při vstupu do výrobního cyklu (I0.6)
7	Porucha kr.	Porucha budiče WatchDog relé (WDOOut)
8	Porucha nekr.	Porucha čtení tlaku
9	Varování	Výrobní cyklus nenahrán (reakce na mačkání startovacích tlačítek)
10	Varování	Manuální krok nenahrán (reakce na mačkání startovacích tlačítek)
11	Varování	Probíhá rekonfigurace zařízení (při přepnutí režimu nebo nahrání nového výrobního cyklu)
12	Varování	Hydraulický agregát nespouštěn (reakce na mačkání startovacích tlačítek)
13	Varování	Baterie CPU řídicího modulu automatu vybita

Tab. 2.4: Přehled událostí a poruch

3 NÁVRH HARDWARE ŘÍDÍČÍHO MODULU

3.1 Základní přehled



Obr. 3.1: Blokové schéma obvodů hardware řídicího modulu

Blokové schéma na Obr. 3.1 představuje propojení CPU s obvody externích periferií. Řídicí modul je tvořen sestavou tří DPS: „Řídicí deska“, „Přídavná deska 1“

a „Přídavná deska 2“. Kompletní schéma je uvedeno v příloze B.1, str. 93. Podrobnějším popisem jednotlivých funkčních částí se budou zabývat následující kapitoly. Schémata zapojení jsou umístěna v rámečku obsahujícím souřadnice, při odkazování na umístění součástek bude použito tvaru „(3.3, D2)“, kde odkaz „3.3“ označuje Obr. obsahující schéma a „D2“ značí souřadnice čtverce na schématu.

3.2 Napájení

Přívodní napájecí část

Jmenovité napájecí napětí řídicího modulu odpovídá hodnotě 24 VDC běžné pro napájení průmyslových zařízení. Bezprostředně za vstupními napájecími svorkami (3.3, D2) je zapojena pojistka (1 A) a unipolární transil (průrazné napětí 33 V), tvoří ochranu proti přepětí a přepólování.

Proudově kompenzovaná tlumivka L4 (3.3, C2) slouží pro potlačení nesymetrické složky indukovaného rušení v napájecích obvodech. Příčný i zpětný vodič jsou navinuty na společném jádru ve stejném smyslu, pro rušivé proudy opačné fáze mají příslušné magnetické toky opačný směr, vzájemně se vyruší. Pro rušivé proudy stejné fáze mají magnetické toky ve vinutích stejný směr, celková indukčnost je pro vysokofrekvenční složky velmi vysoká.

Zapojení kondenzátorů C73, C61 a C62, tlumivky L4 (3.3, D3) tvoří II filtr, který provádí odrušení na vstupech použitých DC-DC měničů. Odstup hraniční frekvence filtru je navržen o dva řády nižší než spínací frekvence měničů. Kondenzátory rovněž akumulují energii pro udržení napájení CPU po dobu pro bezpečné odstavení. Z důvodu nežádoucího přesunu akumulovaného náboje do obvodu hlídání napájení je zapojena dioda D8 (3.3, C3).

DC-DC měniče

Obvody CPU a veškeré I/O karty jsou napájeny z měniče T1 (3.3, D4), obvody analogových vstupů ze zdroje T3 a komunikační část z T3 (obojí 3.4, x3). Důvodem tohoto řešení je vzájemné galvanické oddělení a eliminace možných zemních smyček. Přívodní napájení obvodů T2, T3 je navíc spínáno pomocí High Side Driveru S1 (3.4, A2) přes optočlen OK2 ovládaný z CPU. Odpojení měničů nastane v okamžiku, kdy obvod hlídání napájení nahlásí pokles napětí, CPU si tak ušetří energii.

DC-DC měnič T1 (3.3, C4) typu JTC0624S05 (XpPower) konvertuje napětí na 5 VDC/1,2 A při spínací frekvenci cca 270 kHz, přičemž disponuje mj. vstupním napětím 9 - 36 VDC, izolací 1500 VDC, ochranou proti zkratu na výstupu. Na jeho výstupu je připojen LC filtr L2, C67, respektive EMI filtr pro deklarovanou frekvenci 2 MHz (3.3, C6). To vše pro stabilizaci napájení a odrušení možných

vysokofrekvenčních složek prošlých skrze parazitní kapacitu měniče. Pro ochranu připojených obvodů je zapojen transil (5 VDC) (3.3, D5). Veškeré použité elektrolytické kondenzátory mají nízkou hodnotu ESR a vysokou životnost. Vyfiltrovaný výstup měniče T1 slouží pro napájení vany a rovněž jako vstup lineárního stabilizátoru 3,3 VDC IC13 (3.3, C7), který poskytuje napájení pro CPU a další obvody. Kondenzátor C70 (3.3, D8) slouží ke stabilizaci zmíněného lineárního stabilizátoru, na jehož výstupu je připojen ochranný transil (3,3 VDC).

Další dva měniče T2, T3 stejného typu IW2405SA (XpPower) mají výstup 5 VDC/0,2 A při spínací frekvenci 100 - 650 kHz. Disponují izolačním napětím 1000 VDC a ochranou proti zkratu na výstupu. Na výstupu obou měničů jsou připojeny pro stabilizaci LC filtry druhého řádu. Jelikož měnič disponuje „SoftStartem“, může být spuštěn do kapacitní zátěže 3300 μ F. Doba náběhu na jmenovitou hodnotu napájení odpovídá 70 ms. Z důvodu této poměrně nízké strmosti bylo v případě měniče T2 možno nasadit větší hodnotu tlumivky LC filtru na výstupu.

Výrobce měniče IW2405SA (XpPower) doporučuje minimální zátěž o hodnotě 25%, čehož bylo dosaženo připojením dvou 20mA LED: LED3, LED4 (3.4, D6) na výstup T3. Nevýhodou je snížení impedance obvodu odběru napájení, čímž se snižuje filtrační schopnost kondenzátorů. V případě T2 je připojena pouze jedna 20mA LED: LED7 (3.4, D6), neboť analogové obvody samotné mají stálý odběr více než 20 mA (kap. 3.6, str. 35).

Jelikož zdroj T2 budí kromě analogových obvodů i digitální obvody, jsou ve vhodném místě rozděleny země, přičemž analogová zem je oddělena přes tlumivku L9 (3.4, B8).

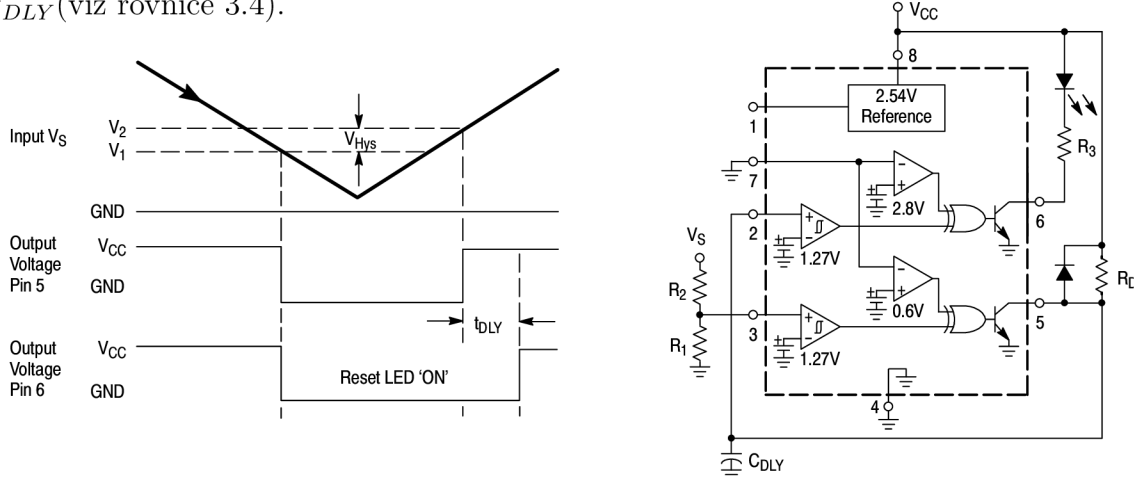
Na výstupech měničů si lze ve všech třech případech všimnout trojice kondenzátorů, lze je popsat na příkladu C36, C37 a C74 (3.3, D7). Kondenzátor má totiž parazitní vlastnosti a jelikož je vhodné blokovat napájení na všech frekvencích, není možné toho dosáhnout jedním kondenzátorem. Impedance kondenzátoru klesá pouze do bodu frekvence jeho vlastní rezonance, poté začne převládat parazitní indukčnost a impedance stoupá. Použitou kombinací 10 μ F (tantal), 100 nF (obecně keramika) a hlavně 1 n (NPO) lze dosáhnout lepších vlastností.

Obvod hlídání napájení

Součástí tohoto obvodu je vývod K7 k připojení indikující LED „DC OK“ do čelního panelu řídicího modulu. Pro bezprostřední zjištění poklesu napětí přívodního napájení je použit int. obvod IC14 MC34161 (3.3, A3) v poupraveném zapojení dle schématu z datasheetu (viz Obr. 3.2). Pomocí napěťového děliče tvořeného rezistory R25, R26 a potenciometru R23 (3.3, B3) se nastavuje hodnota napětí V_1 . Při poklesu napětí pod tuto hodnotu jsou oba výstupy na pinu 5 a 6 obvodu MC34161 přitlačeny

k zemi. To způsobí zkratování diody optočlenu OK1 (3.3, B5) a na pinu CPU se objeví log. 1 (signál CPU_VOLT_MONITOR).

Při případném obnovení napájení na úroveň vyšší než V_2 , je výstup na pinu 6 rozepnut se zpožděním t_{DLY} daným rezistorem R24 a kondenzátorem C105. Pro náběh na napájecí napětí 24 VDC lze spočítat konkrétní hodnotu doby zpoždění t_{DLY} (viz rovnice 3.4).



Obr. 3.2: Funkce a zapojení obvodu MC34161 v režimu zpožděného resetu (lit.[5])

Jak již bylo zmíněno, pomocí potenciometru R23 lze nastavit V_1 následovně, přičemž R''_{23} v obecném vyjádření (rovnice 3.1) je hodnota odporu části potenciometru proti napájení, R'_{23} je zbytek. Dolní hranice vypínacího napětí V_1 je spočítána v rovnici 3.2, horní V_1' poté v rovnici 3.3.

$$V_1 = (V_{th} - V_H) \left(\frac{R_{25} + R''_{23}}{R_{26} + R'_{23}} + 1 \right) [V]; \quad V_2 = V_{th} \left(\frac{R_{25} + R''_{23}}{R_{26} + R'_{23}} + 1 \right) [V] \quad (3.1)$$

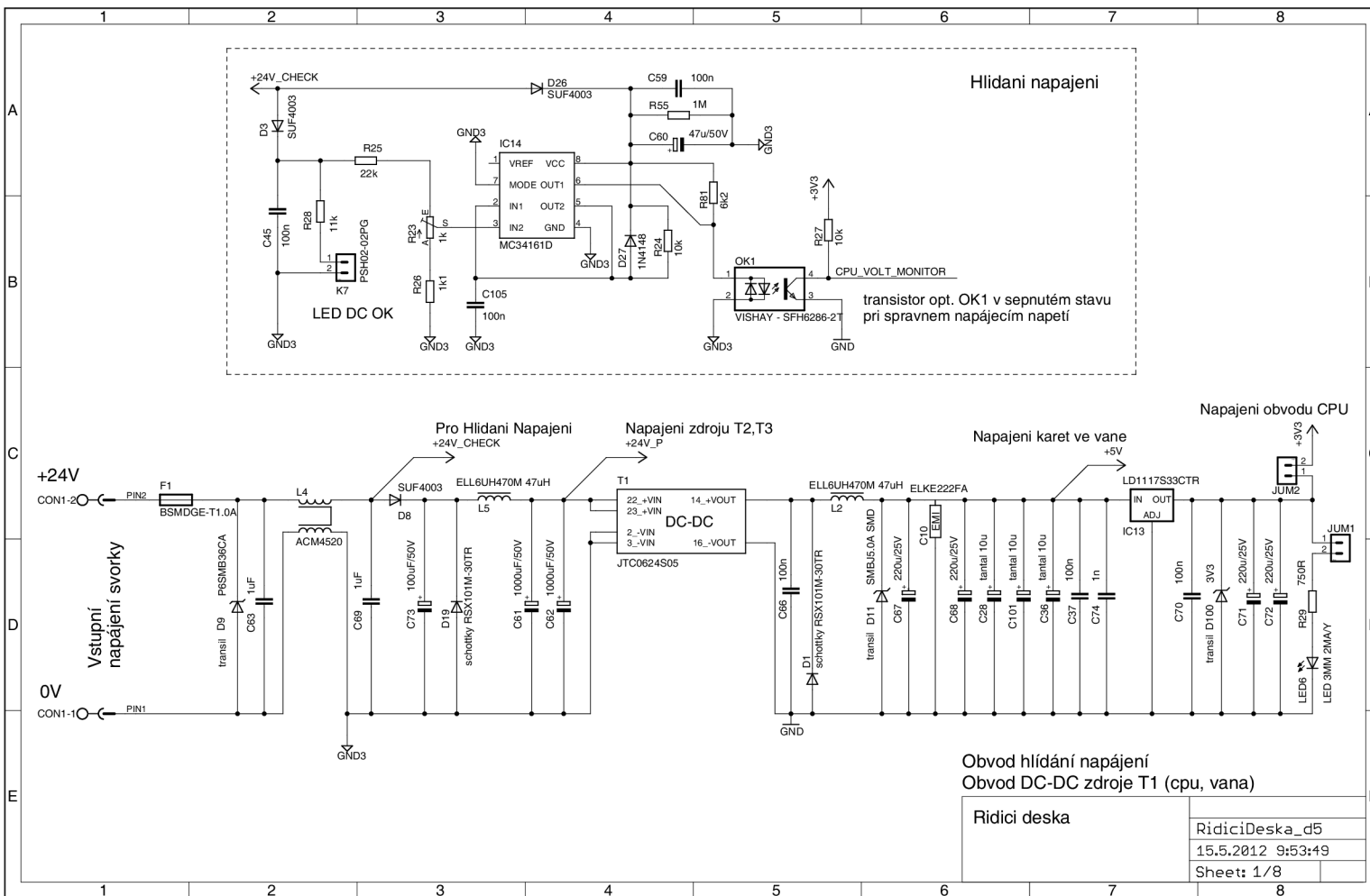
$$V_1 = (1,270 - 0,025) \left(\frac{22000 + 0}{1100 + 1000} + 1 \right) = 14,3 V \quad (3.2)$$

$$V_1' = (1,270 - 0,025) \left(\frac{22000 + 1000}{1100 + 0} + 1 \right) = 27,3 V \quad (3.3)$$

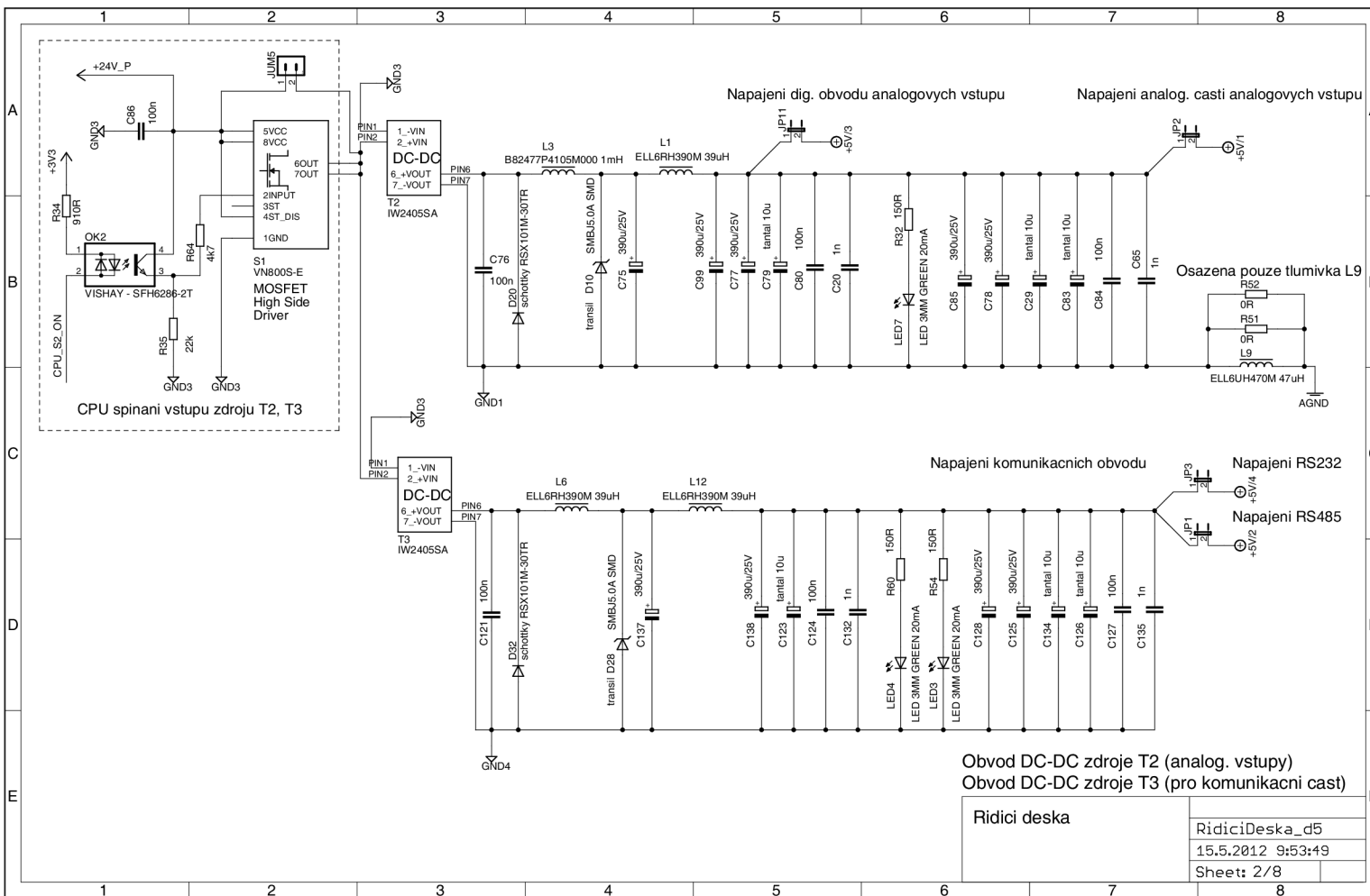
$$t_{DLY} = R_{24} C_{105} \ln \left(\frac{1}{1 - \frac{V_{th}}{V_{CC}}} \right) = 10^4 * 10^{-7} * \ln \left(\frac{1}{1 - \frac{1,27}{24}} \right) = 54 \mu s \quad (3.4)$$

V předchozích rovnicích vystupuje proměnná $V_{th} = 1,27 V$, což je srovnávací napětí interního komparátoru, jehož typická hystereze V_H odpovídá 24 mV.

Obr. 3.3: Řídící modul 1/8 - převodní napájecí část, napájení vany a CPU



Obr. 3.4: Řídící modul 2/8 - napájecí komunikačních obvodů a analog. vstupu



3.3 CPU

Připojení CPU

Schéma na Obr. 3.5 představuje zapojení stěžejních diskretních součástek k CPU. Jedná se o 100nF blokovací kondenzátory napájení. Obvod RTC CPU je zálohován z baterie BAT1 (3.5, D1). Procesor má osazeny dva krystalové oscilátory. Krystal Q2 (3.5, D5) slouží jako hlavní zdroj hodin, krystal Q3 je zdroj hodin pro obvod reálného času RTC.

Por rezervní účely je zde navržen obvod pro připojení FRAM paměti IC3 FM24CL16, která je připojitelná přes sběrnici I2C. Výstupy budiče I2C mají otevřený kolektor, proto je nutné připojit pull-up rezistory R53, R105 o hodnotě 1,1 k Ω (3.5, D7).

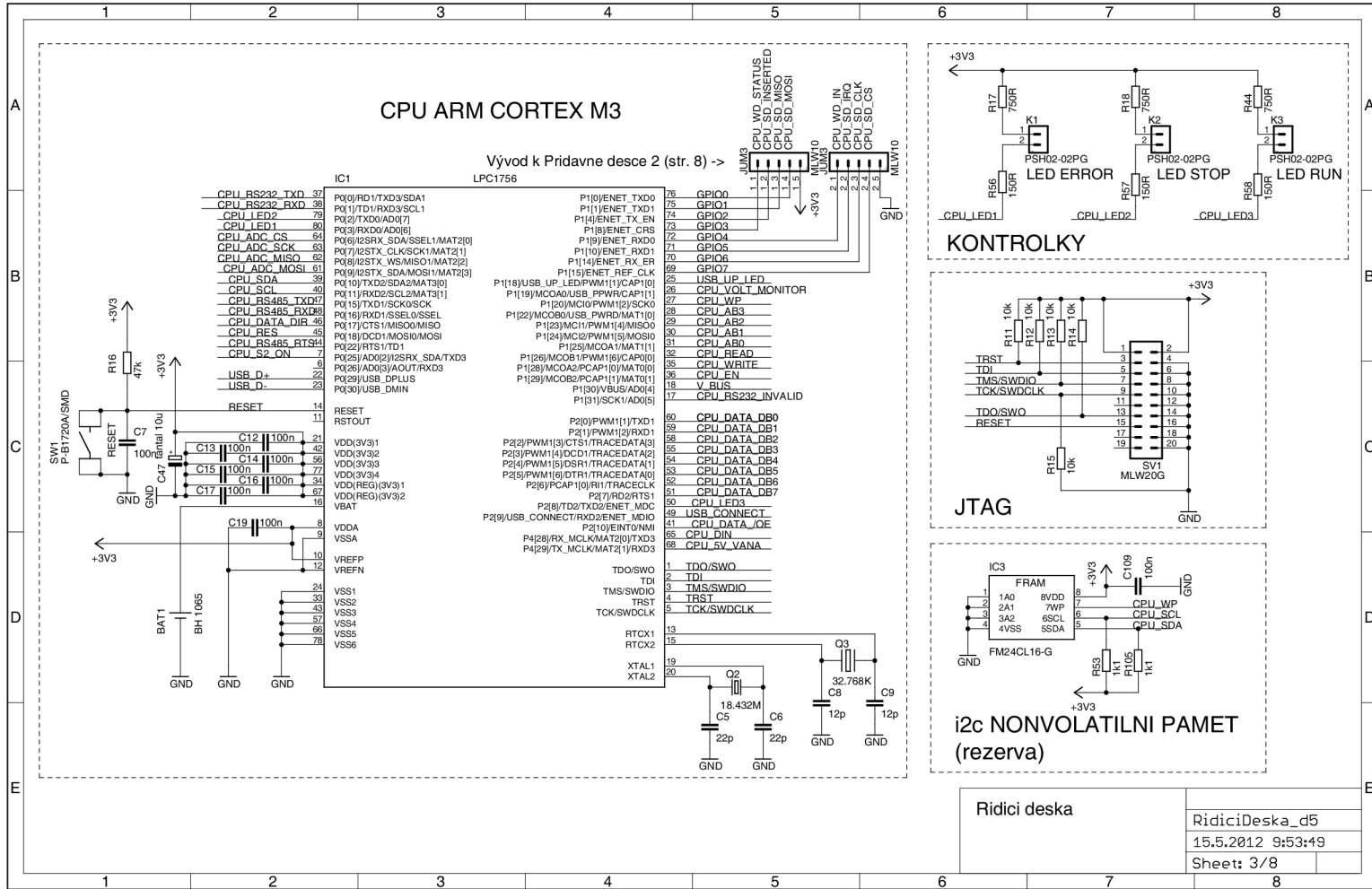
JTAG rozhraní pro připojení programátoru je vyvedeno skrze SV1 konektor (3.5, C7). Pro indikaci stavu firmware v CPU jsou osazeny do čelního panelu krabice řídicího modulu 2mA LED, které jsou připojeny do prostřednictvím konektorů K1-K3 (3.5, A6-A8). Konektor JUM3 (3.5, A5) slouží pro připojení „Přídavné desky 2“, která obsahuje SD kartu, vybrané signály konektoru JUM3 přes tuto desku putují do „Přídavné desky 1“, která obsahuje budič CPU WatchDog relé.

CPU NXP LPC1756 ARM Cortex-M3

Charakteristika jádra CPU Cortex-M3 je obsažena v kapitole 6.1. Na tomto místě budou shrnuty vybrané parametry tohoto konkrétního použitého typu CPU (lit.[11]):

- až 100 MHz, krystal 1 MHz to 25 MHz, 4 MHz interní RC oscilátor, PLL
- 256 kB flash, celkem 32kB SRAM
- JTAG debug, In-System Programming (ISP), In-Application Programming (IAP) přes bootloader
- 8kanálový General Purpose DMA controller
- USB2.0 device/Host/OTG controller
- 4UART kanály
- CAN controller se 2 kanály, 2 kanály I2C
- SPI controller, SSP controller
- 12bitový ADC s 6 vstupy, 10bitový DAC
- 52 konfigurovatelných I/O pinů
- 4 Timer/Counter, Real-Time Clock (RTC) obvod s 20 B zálohovaných registrů
- Watchdog Timer (WDT)
- ARM Cortex-M3 Systick timer
- detekce podpětí (Brownout detect), Power-On Reset (POR)

Obr. 3.5: Řídicí modul 3/8 - připojení CPU, připojení kontrolky



3.4 Připojení na sběrnici, USB

Signálové vodiče sběrnice

Sběrnice vany obsahuje napájecí a signálové vodiče, přičemž využívá logická úroveň TTL. Jelikož CPU používá 3,3V logickou úroveň LVTTL, pro převod úrovní jsou využity obvody IC9 a IC8 (3.7, x7). Jedná se o třístavové budiče sběrnice, kdy port na jedné straně je napájen ze 3,3 VDC a na druhé z 5 VDC. Vodiče, které obsahuje sběrnice (3.7, x8) jsou popsány níže, přičemž popis je rozšířen o signály, jimiž CPU řídí obvod buzení sběrnice IC9.

+5V, 0V Napájecí napětí

CPU_DIN Signál je v log. 0, pokud je adresována vstupní karta (nepoužíváno)

CPU_DATA_/OE Signál „Output Enable“, tedy pokud je signál v log. 1, výstupní port budiče IC9 je ve stavu vysoké impedance, převádí při log. 0.

CPU_DATA_DIR Při log. 0 platí, že port $A = B$, port B je vstup. Hodnota log. 1 přináší opačnou konfiguraci a je aktivní po resetu CPU.

CPU_DATA_DB0-7 Datová sběrnice o šíři jednoho bajtu.

CPU_RES Signál resetu (aktivní v log. 1) D-latch obvodů na výstupních kartách. Pro normální funkci je v log. 0.

CPU_EN Signál povolení (aktivní v log. 1) přístupu k I/O kartě na sběrnici.

CPU_WRITE Signál povolení (aktivní v log. 1) zápisu na výstupní kartu. Při současné CPU_EN=1 a adresaci skutečně existující karty je aktivní latch budiče výstupů, který zrcadlí datovou sběrnici na výstupy karty.

CPU_READ Signál povolení (aktivní v log. 1) čtení ze vstupní karty. Při současné CPU_EN=1 a adresaci skutečně existující karty je aktivní latch, který zobrazí stav vstupů na datovou sběrnici.

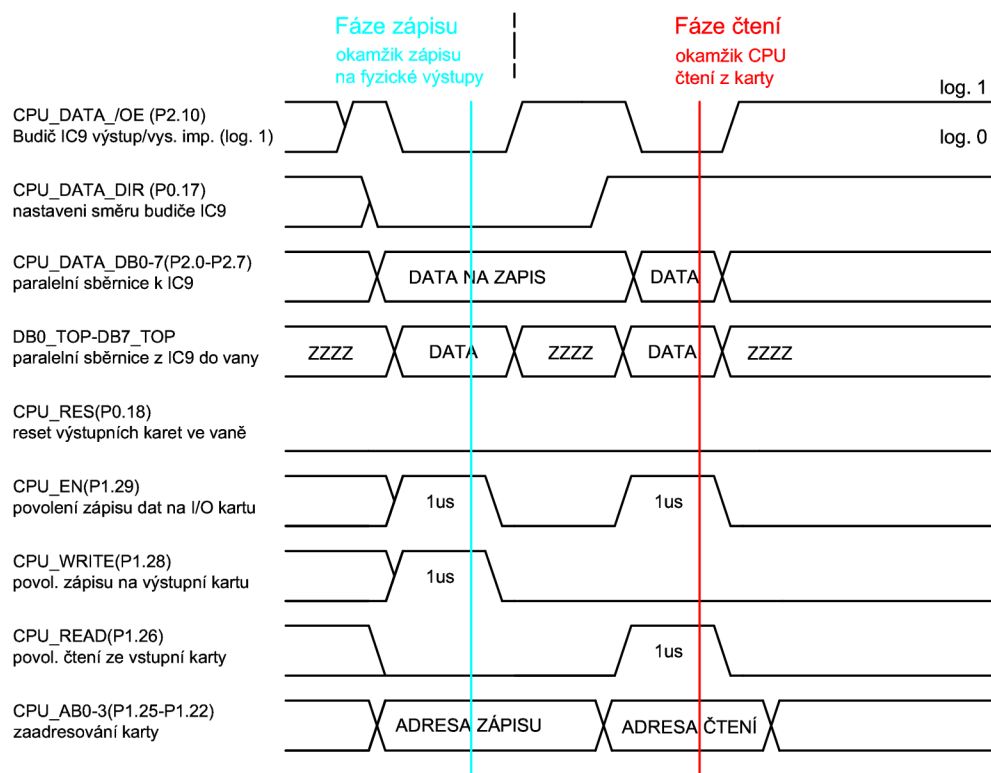
CPU_AB0-AB3 4bitová adresová sběrnice

Časová posloupnost signálů na sběrnici ovládaných z CPU pro přístup na I/O karty je zobrazena na Obr. 3.6.

Spínání napájení vany

Odepnutí napájení vany je současně s aktivním signálem Reset způsob bezpečné deaktivace sepnutých výstupů karet. Tento krok je prováděn z CPU při startu napájení nebo při odpojení napájení z důvodu, aby optočleny galvanického oddělení výstupů byly co nejdříve připojeny k log. 0. Pomocí samotného odepnutí dojde obecně k odpojení napájení všech karet.

CPU má možnost odepnout napájení pomocí int. obvodu S2 (3.7, A3). Jedná se o MOSFET High Side Driver, který má pracovní napětí od 4,5 VDC. Tento obvod chrání svůj tranzistor proti přetížení nad 5,4 A, tento proud by ovšem při aktivní



Obr. 3.6: Zápis na výstupní kartu, čtení ze vstupní karty

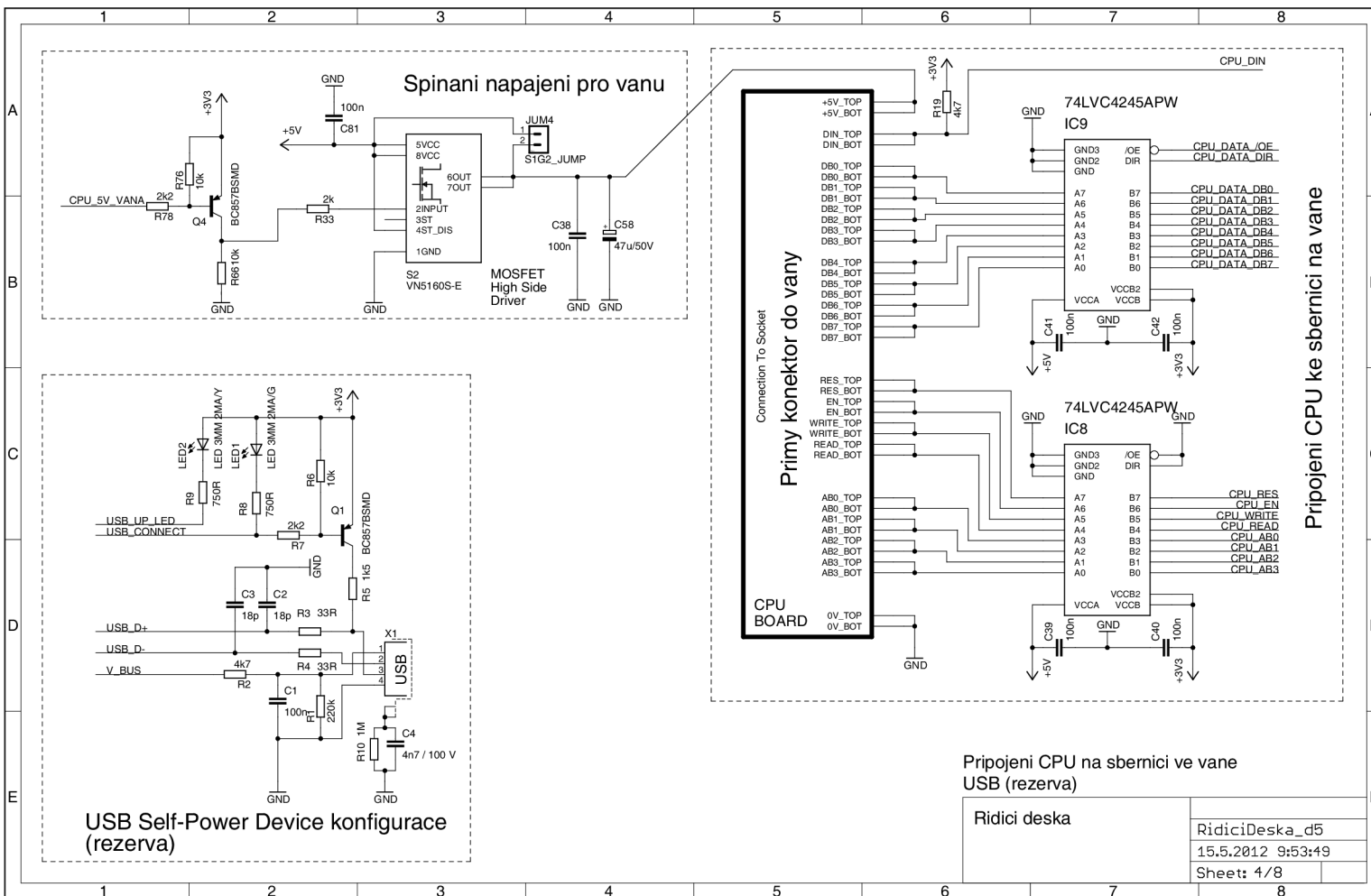
ochraně výstupu napájecího měniče proti zkratu neměl být přítomen. Obvod má vysokoimpedanční logický vstup na pinu 2, který může být buzen přímo z CPU pinu, což při použití N-channel MOSFET není možné. Při využití int. obvodu odpadají rovněž proudové pulsy, které by protékali delšími vzdálenostmi na plošném spoji. I/O piny CPU jsou při startu CPU nastaveny jako vstupy s připojeným pull-up rezistorem. Docházelo by tedy ke krátkodobému sepnutí S2, než by byly I/O piny nastaveny softwarově do výstupů s log. 0. Proto bylo použito obvodového řešení s PNP transistorem Q4 (3.7, B2), který budí vstup int. obvodu S2.

USB (rezerva)

Pro připojení USB sběrnice k CPU jako zařízení s vlastním napájením (Self-powered device) disponuje CPU kromě základních signálů USB sběrnice i dalšími signály (3.7, D2). USB Host lokalizuje nově připojené zařízení na základě připnutí linky USB_D+ pull-up rezistorem na úroveň 3,3 V, poté následuje proces konfigurace USB zařízení. Nicméně CPU po restartu potřebuje nějaký čas na inicializaci firmware. Proto CPU připne výstupem USB_CONNECT pull-up R5 skrze PNP transistor Q1 ve vhodný okamžik (SoftConnect) a tím se přihlásí na sběrnici.

LED2 buzená signálem USB_UP_LED informuje o správné konfiguraci a enumeraci zařízení na sběrnici. Připojení rezistorů R3, R4 provedeno dle výrobce CPU.

Obr. 3.7: Řídící modul 4/8 - připojení CPU na sběrnici ve vaně, USB (rezerva)



3.5 Komunikační obvody

Připojení k rozhraní RS-232 (lit.[6])

Jak již bylo zmíněno v souvislosti s napájecími měniči, komunikační obvody jsou galvanicky odděleny od CPU a napájeny nezávisle z měniče T3. Obvod IC6 MAX3221 (3.8, B4) obsahuje pouze jeden budič ($T1OUT$) a přijímač ($R1IN$) úrovně RS-232. Hodnoty kapacity podpůrných kondenzátorů C30 - C33 byly určeny dle výrobce pro napájecí napětí 5 VDC. Vstup \overline{EN} (1) povoluje v log.0 činnost přijímače.

Obvod dále disponuje automatickou funkcí vypnutí budiče, která je aktivována v případě, že přijímač „necítí“ RS-232 úrovně na svém vstupu. Nastavením vstupů $FORCEON$ a $\overline{FORCEOFF}$ je tato funkce vypnuta a budič je stále aktivní. Výstup pro monitorování činnosti obvodu $\overline{INVALID}$ je v log. 1, jestliže pro úroveň napětí V_i na vstupu přijímače platí $|V_i| > 2,7 V$. Výstup $\overline{INVALID}$ je v log. 0 pokud se linka zdrží v pásmu $|V_i| < 0,3 V$ více než $50 \mu s$. Obvod má kromě blokovacího kondenzátoru na napájení tantal C56 10μ co nejbližší, kvůli snížené ploše proudové smyčky a tím i vyzařování.

Signály UART ($R1OUT$, $T1IN$), společně s diagnostickým výstupem $\overline{INVALID}$ jsou galvanicky odděleny obvodem ADUM1301ARWZ IC10 (3.8, B3) od UART3 kanálu CPU, respektive pinu P1.31 CPU. Přenos signálů funguje na principu transformátoru, přičemž v této verzi obvod umožňuje přenos až 1 Mbps. Výstupy tvoří třístavové budiče, které mohou být uvedeny do stavu vysoké impedance pomocí příslušného vstupu $VE1$ na straně komunikační části nebo $VE2$ na straně CPU. Budiče jsou trvale povoleny připojením log. 1 na zmíněné vstupy (3.8, B3).

Preventivní opatření vzhledem poměrně dlouhým vedením UART k CPU je využití impedančního přizpůsobení podélnou impedancí. Trojice rezistorů R45-R47 (3.8, B2) je umístěna tak, aby impedance byla těsně u výstupu na vedení. Řešení je použito i v případě RS845.

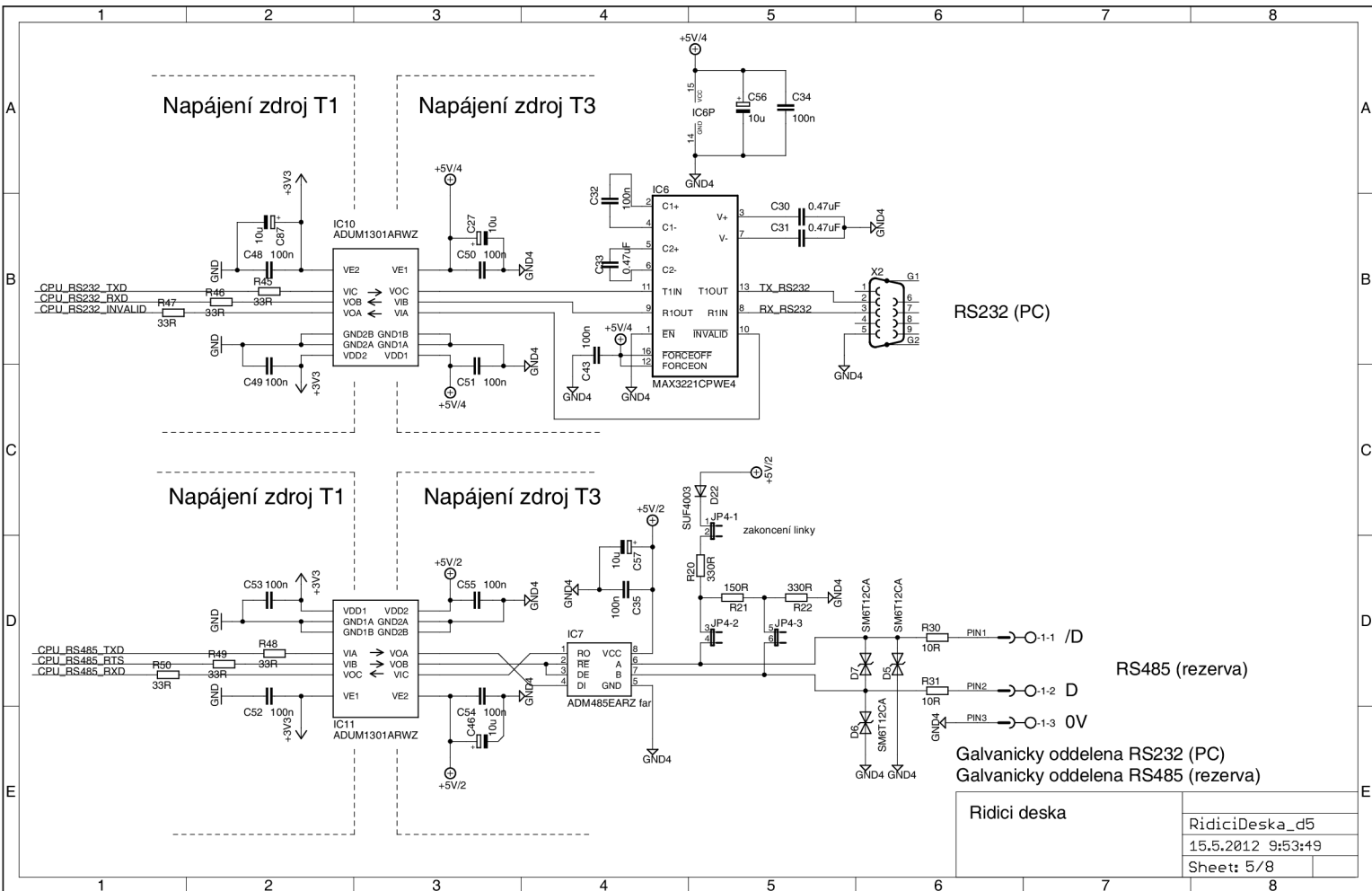
Připojení na sběrnici RS-485

Připojení budiče sběrnice RS-485 ADM485EARTZ IC7 (3.8, D4) k UART1 kanálu CPU je opět realizováno pomocí obvodu galvanického oddělení ADUM1301ARWZ IC11 (3.8, D3). Ovšem UART signály jsou doplněny signálem pro přepínání směru komunikace na sběrnici RS-485.

Symetrické vedení sběrnice RS-485 má realizovanou přepětovou ochranu bezprostředně za vstupními svorkami. Jedná se o zapojení trojice bipolárních transilů D5 - D7 s průrazným napětím 12 V (3.8, D6). Podélně kolem sběrnice RS485 je tažen zemní vodič, který brání proti příliš velkému potenciálovému rozdílu mezi stanicemi na lince, hlavně v případě velkých vzdáleností. Transily D5 a D6 chrání proti

přepětí vůči tomuto vodiči spojeného se zemí komunikační části. Transil D7 omezuje napětí mezi vodiči sběrnice. Toto zapojení je dle doporučení doplněno vložením dvou rezistorů R30 a R31 o hodnotě 10Ω (3.8, D6) jako podélné impedance mezi přívodní svorky a ochranné zapojení s transily.

Zapojení sběrnice RS-485 je rozšířeno o sérii rezistorů pro impedanční přizpůsobení sběrnice a udržení definovatelného stavu sběrnice R20 - R22 (3.8, D5) připojitelnou přes jumper JP4. Dioda D22 (3.8, C5) slouží k ochraně zdroje.



Obr. 3.8: Řidič modul 5/8 - galvanicky oddělená komunikace RS-232, RS-485

3.6 Analogové vstupy

Vstupní obvody

Pro připojení tlakového snímače byly navrženy dva analogové vstupy pro proudovou smyčku 0-20 mA, přičemž jeden vstup jako rezervní. Princip zapojení bude popsán na vstupu „AI1“ ((3.10, B5 - B8)). Bezprostředně za vstupní svorkou je zapojena rychlá pojistka F2 a unipolární transil D14 (průrazné napětí 6 V) jako vstupní ochrana proti přepětí i přepólování. NPO keramický kondenzátor C98 slouží k odfiltrování vysokofrekvenčního rušení.

Proudová smyčka 0-20 mA vytváří úbytek 0-4 V na přesném rezistoru R38 o hodnotě 200 Ω. Je použit „Thin Film“ rezistor PAT1206E2000BST5 s minimální parazitní indukčností, přesností hodnoty ±0.1% a poměrně nízkou teplotní závislostí ±25 ppm/°C. Za tímto odporem je filtr typu dolní propust R37 10 kΩ, C96 6,8 nF.

Kvůli krátké době nabíjení vstupní kapacity kanálu ADC převodníku by měl mít zdroj měřeného signálu nízkou impedanci, proto je zajištěno impedanční oddělení mezi popsanou vstupní obvodovou částí a vstupním kanálem ADC převodníku ((3.10, B4)) pomocí OZ v zapojení napěťového sledovače IC2B (3.10, B5). Napěťový sledovač na svém výstupu budí RC článek R41, C23, který je osazen dle doporučení výrobce.

Pro ochranu vstupu OZ před přepětím slouží zapojení rezistoru R36, Zenerovy diody D15 jako stabilizátoru napětí 3,9 V a diody D12. Vstupní napětí je tedy omezeno na úroveň cca 4,5 V, aniž by ovlivňovalo měřený signál. Řešení doplňuje funkci vstupní ochrany transilem.

ADC převodník

Použitý 12bitový ADC převodník ADC1 LTC1861 (3.10, B4) disponuje dvěma vstupními kanály CH0 a CH1, které jsou používány jako „Single Ended“. Převodník pracuje na principu postupné aproximace a dosahuje převodní rychlosti 250 ksp/s, přitom charakterizován 72dB poměrem signálu vůči šumu. ADC převodník obsahuje vstup VREF pro zdroj externího referenčního napětí, přičemž jako zdroj byl využit int. obvod REF1 LT6656 (3.10, D3) s jmenovitou výstupní hodnotou napětí 4,096 V.

Při komunikaci je využito SPI v konfiguraci pro obousměrné spojení (viz Obr. 3.9), ADC převodník je schopen komunikovat při SCK signálu až 20 MHz, použitá rychlost je 9 MHz. Prostřednictvím obvodu galv. oddělení U1 ADUM1401BRWZ, který pracuje do 10 Mbps, je ADC převodník připojen k CPU na kanál SSP1. Komunikace je popsána v datasheetu (viz Obr. 3.9), přičemž níže je popsáno připojení signálů (dle označení pinů ADC převodníku) k CPU kanálu SSP1. Jelikož ADC převodník disponuje i možností diferenciálního měření mezi oběma kanály, je nutné

pomocí MOSI linky nastavit výběr „Single ended“ hodnoty a provést výběr požadovaného kanálu CH0/CH1.

- SDO - MISO
- SDI - MOSI
- CONV - SSEL
- SCK - SCK

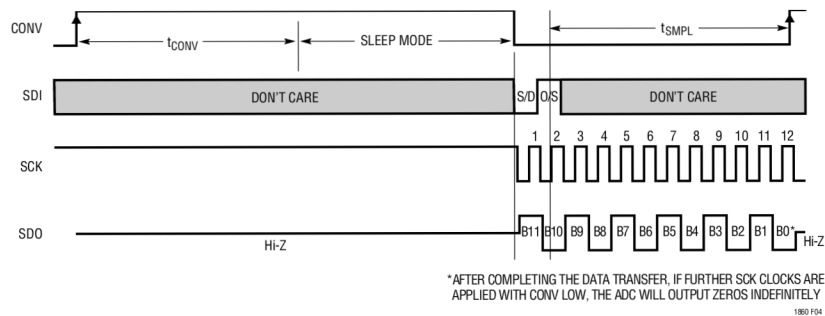


Figure 4. LTC1861 Operating Sequence

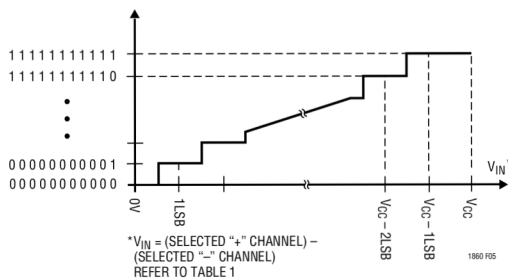


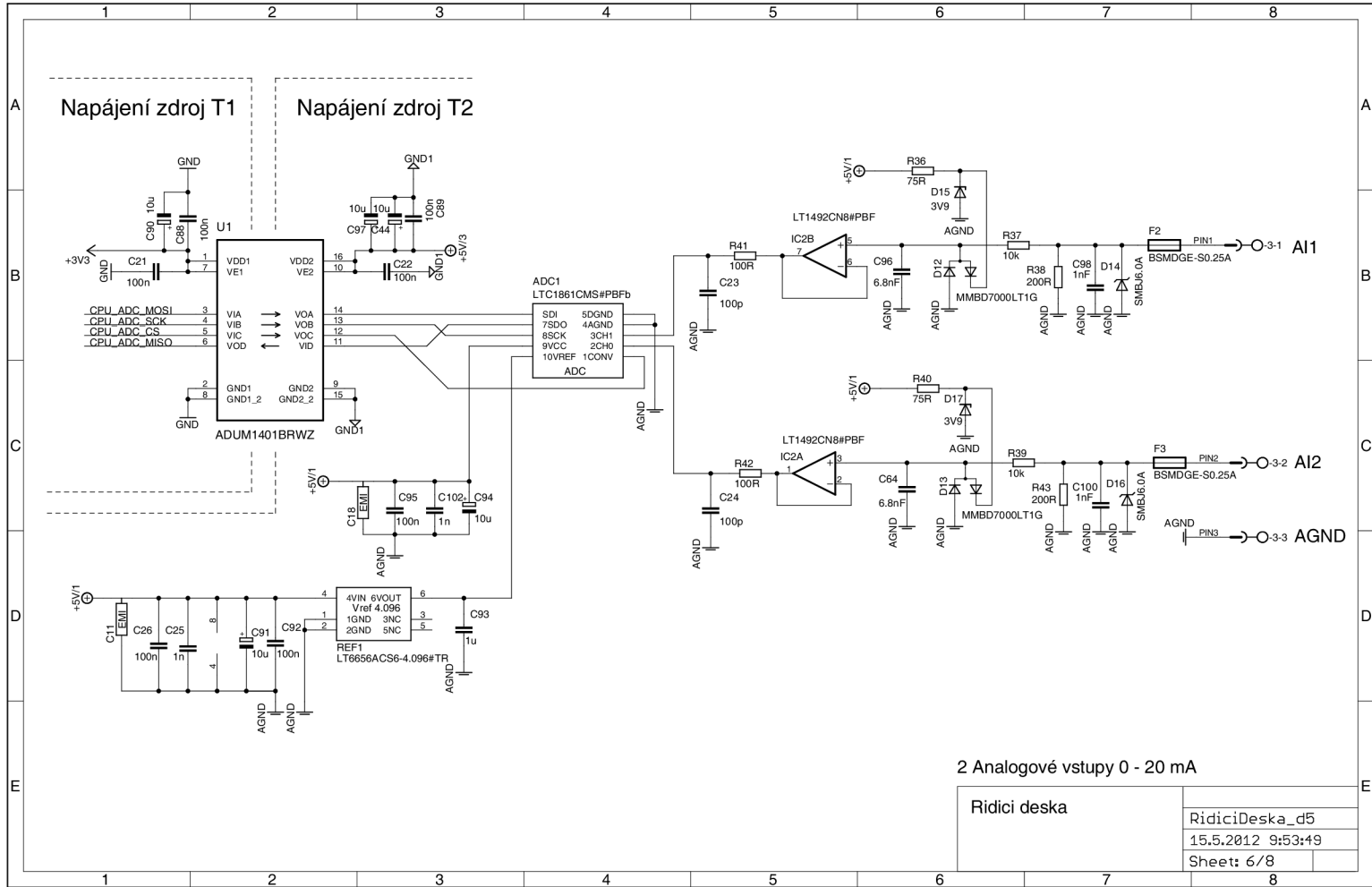
Figure 5. LTC1861 Transfer Curve

Table 1. Multiplexer Channel Selection

	MUX ADDRESS		CHANNEL #		
	SGL/DIFF	ODD/SIGN	0	1	GND
SINGLE-ENDED MUX MODE	1	0	+		-
	1	1		+	-
DIFFERENTIAL MUX MODE	0	0	+	-	
	0	1	-	+	

Obr. 3.9: SPI pro komunikaci s ADC převodníkem (lit.[7])

Obř. 3.10: Řídící modul 6/8 - galvanicky oddělené analogové vřtupy 0 - 20 mA



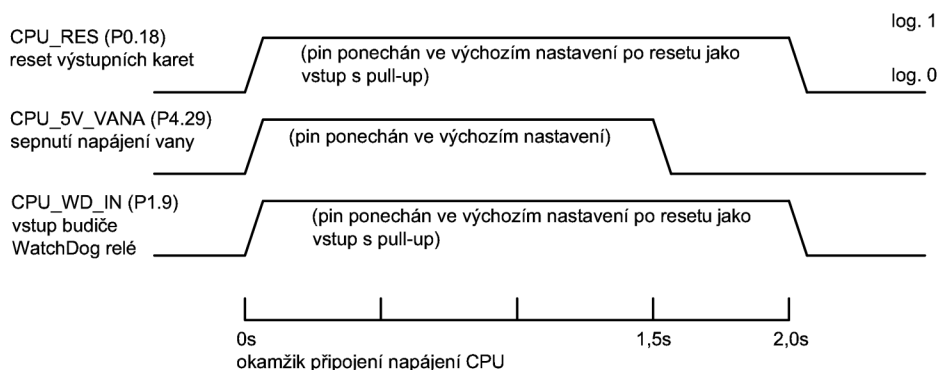
3.7 CPU WatchDog obvod

Toto obvodové řešení je umístěno na samostatné desce plošného spoje, označené jako „Přídavná deska 1“, tudíž obsahuje vlastní vstupní napěťovou část (3.12, B2). Přepětíovou ochranu a ochranu proti přepólování tvoří pojistka F4 (250mA) a transil (průrazné napětí 33 V). Dioda D2 slouží k oddělení energie akumulované na kondenzátoru C111 od popsaných vstupních obvodů.

Budící prvek S3 (MOSFET High Side Driver VN800S-E) (3.12, B4) slouží k napájení cívek relé zapojených obvodu Central STOP řídicího rozvaděče (signál OUTPUT na svorku „WDOut“). Je tedy důležité, aby byla kontrolována činnost tohoto budiče. k tomu účelu slouží pin 3 „STATUS“, na němž je připojen výstup s otevřeným kolektorem. Při správné funkci budiče není transistor sepnut. Pro realizaci hlášení do CPU je k tomuto výstupu připojen pull-up R77 vůči napájení a zároveň LED optočlenu OK4 proti zemi. V tomto zapojení (3.12, B6) pozná totiž CPU poruchu budiče i ztrátu napájení celé „Přídavné desky 1“. (lit.[8])

Optočlen OK3 slouží pro galvanické oddělení ovládání budiče S3 z CPU (CPU_WD_IN) . LED tohoto optočlenu je sepnuta přes odpor R56 (3.12, C7) z log. 0. Jak ale bylo uvedeno, I/O piny jsou po resetu CPU nastaveny jako vstupy s připojeným pull-up rezistorem, proto je pin ponechán jako vstup do doby než má být sepnut. Softwarové nastavení pinu CPU jako výstup způsobí log. 0 na tomto výstupu.

Pozice řídicího modulu je v rozvaděči začleněna tak, že v kombinaci s nastaveným resetem (CPU_RES) na sběrnici vany a odepnutím napájení vany (CPU_5V_VANA) slouží vypnutí (CPU_WD_IN) Watchdog budiče S3 jako třetí bezpečnostní prvek. V případě resetu vyvolaného interním WatchDog obvodem CPU dojde hardwarově k nastavení vstupu WatchDog budiče S3 do log. 0 (tranzistor optočlenu OK3 ne-sepnut). To má za následek rozpojení obvodu Central STOP. Časový průběh výše uvedených signálů bezprostředně po resetu CPU je implementován dle Obr. 3.11.



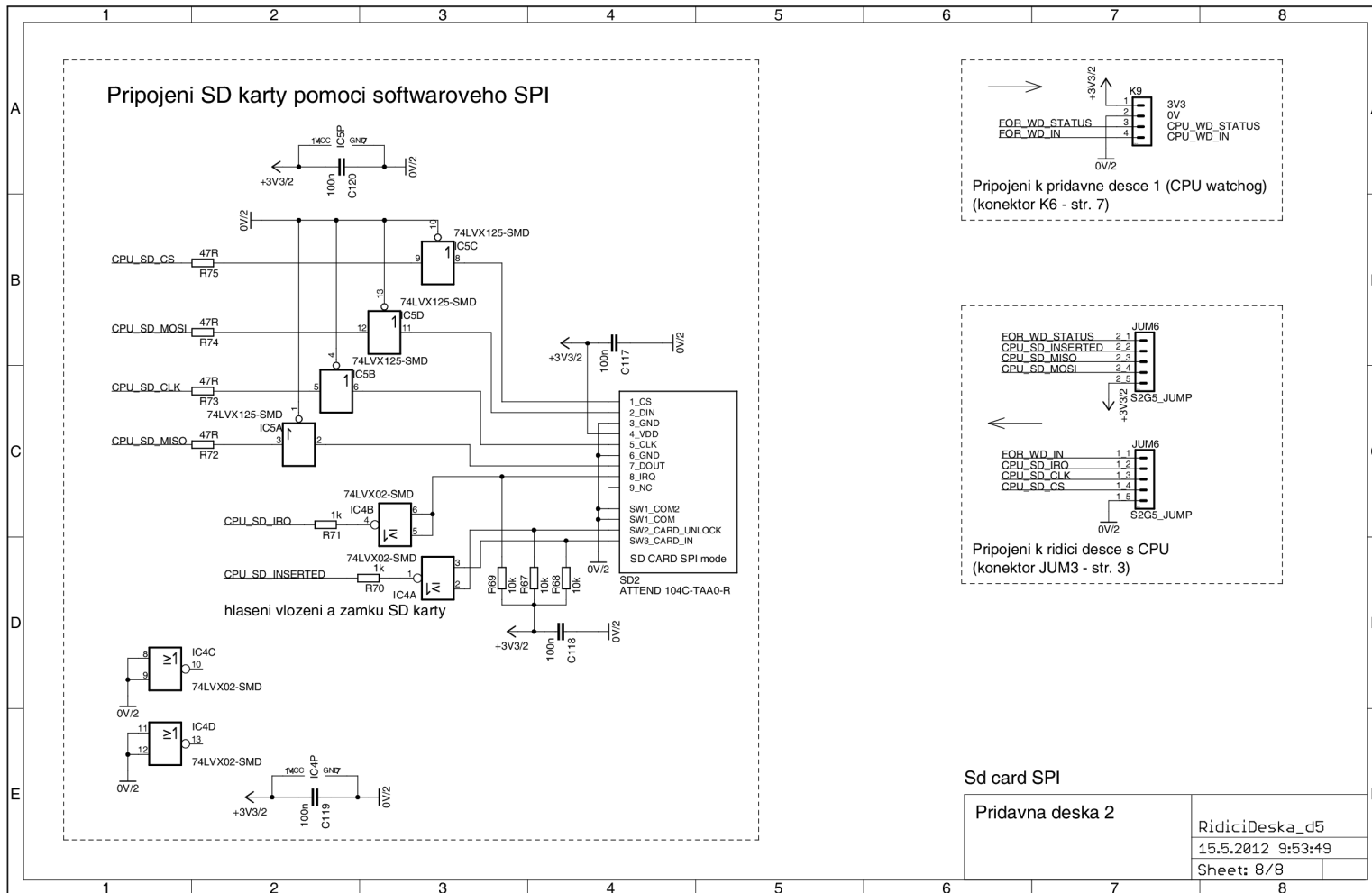
Obr. 3.11: Opatření při resetu CPU

3.8 Připojení SD karty

Obvod připojení SD karty se nachází na samostatné desce s označením „Přídavná deska 2“ (viz schéma Obr. 3.13). Připojení k CPU je realizováno prostřednictvím softwarového SPI, neboť byla SD karta pro účely logu dodělána dodatečně a pro připojení musí být využity obecné I/O piny CPU, které byly z řídicí desky vytaženy jako rezerva. (lit.[12])

Ke slotu SD karty jsou připojeny signály SPI přes budič IC5 (3.13, B3), a to z důvodu ochrany pinů CPU proti ESD a případnému zkratu. Slot SD karty obsahuje dva mechanické kontakty pro kontrolu zasunutí a odemčení vložené SD karty. k těmto kontaktům jsou připojeny pull-up rezistory R67 a R68, přičemž je vyhodnocována funkce NOR hradlem IC4 (3.13, D3), jehož výstup je přiveden do CPU.

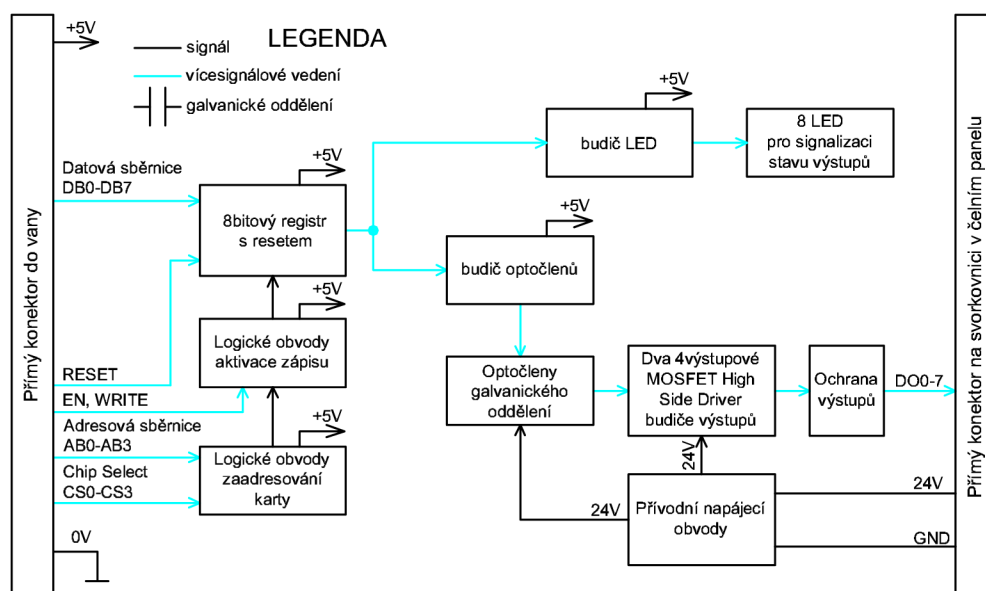
Obr. 3.13: Řídící modul 8/8 - připojení SD karty pomocí softwarového SPI



4 NÁVRH HARDWARE 8BITOVÉ 24VDC DO KARTY

4.1 Základní přehled

Blokové schéma na Obr. 4.1 prezentuje hlavní funkční části navržené 8bitové výstupní karty s galvanicky oddělenými výstupy. Logické obvody jsou napájeny 5 V z vany a zajišťují příjem dat ze sběrnice. Výkonové budiče výstupů jsou napájeny z 24 V a jejich logické vstupy jsou optočleny galvanicky odděleny od logických obvodů.



Obr. 4.1: Blokové schéma výstupní karty

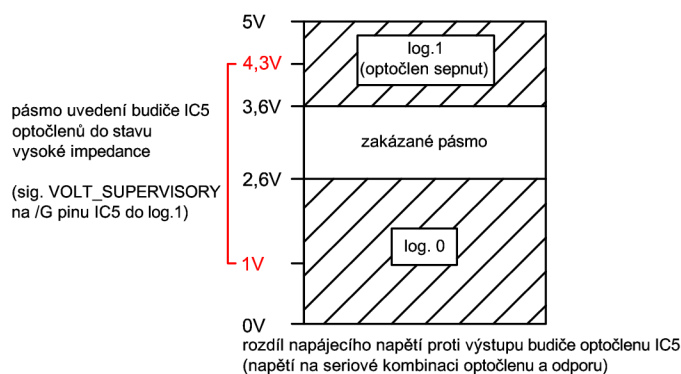
4.2 Logické obvody

Přímý konektor na jedné straně karty slouží k připojení na sběrnici prostřednictvím slotu ve vaně. Na schématu (viz Obr. 4.3, B2) jsou zobrazeny veškeré signály sběrnice, kterých výstupní karta využívá. Tyto signály mají připojeny pull-up rezistory, řešení stejné jako v případě originálních karet, které byly aplikovány v dobách budičů s otevřeným kolektorem. Vzhledem k přehlednosti bylo toto zapojení zachováno.

Navíc v případě datové sběrnice (viz Obr. 4.3, A3) mají $100k\Omega$ rezistory význam hlavně v situaci, kdy je budič sběrnice ve stavu vysoké impedance.

Obdobná situace je i v případě řídicí sběrnice *RES*, *ENABLE*, *WRITE* a adresové sběrnice *AB0-AB3*, i když u těchto případů není směr budiče na řídicí desce měněn a při správné činnosti by ke stavu vysoké impedance nemělo dojít. Nicméně pokud se stav vysoké impedance na řídicí sběrnici objeví, signál *RES* bude přes pull-up nastaven do log.1 a výstupy budou zresetovány do log. 0. Signály Chip Select *CS0-CS3* představují fyzickou adresu pozice modulu ve vaně, zde jsou pull-up rezistory povinné, neboť jedničkové bity adresy nejsou ve vaně zadrátovány na 5 V. Zmíněná fyzická adresa je drátována ve vaně jako bitová negace čísla pozice ve vaně.

Adresace karty (4.3, C5) je z výše popsaných důvodů logickou funkcí XOR mezi jednotlivými příslušnými signály adresové sběrnice a Chip Select. Signálům sběrnice byl věnován prostor zde (kap. 3.4, str. 29). Nicméně pokud je na výstupu hradla AND IC2B (4.3, D6) log. 1, která je přivedena na D-latch IC4 (4.3, C8), potom IC4 překllopí stav datové sběrnice na svoje výstupy.



Obr. 4.2: Zakázané pásmo budících LED optočlenů

Na výstupu D-latch obvodu IC4 jsou připojeny dva budiče. Konkrétně budič IC3 74HCT254N (4.5, B6) slouží k buzení osmice 2mA LED pro indikaci stavu výstupů. Druhý třístavový budič s výstupem typu otevřený kolektor IC5 74LS642 přizemňuje optočleny OK1 (4.4, B3) a OK2 (4.4, B2), které galvanicky oddělují logické vstupy silových budičů výstupů.

Každý optočlen je buzen proudem 5 mA , nicméně aby nedošlo k situaci, že optočlen opustí pásmo saturace vlivem poklesu napájecího napětí 5 V, bylo provedeno připojení obvodu IC7 MCP810T (4.3, A5). Jedná se o hlídač napájení, který je schopen v pásmu napájecího napětí od 4,3 V po 1 V udržet svůj výstup *RST* čili signál *VOLT_SUPERVISORY* v log. 1. Tento signál je připojen na vstup \overline{G} budiče

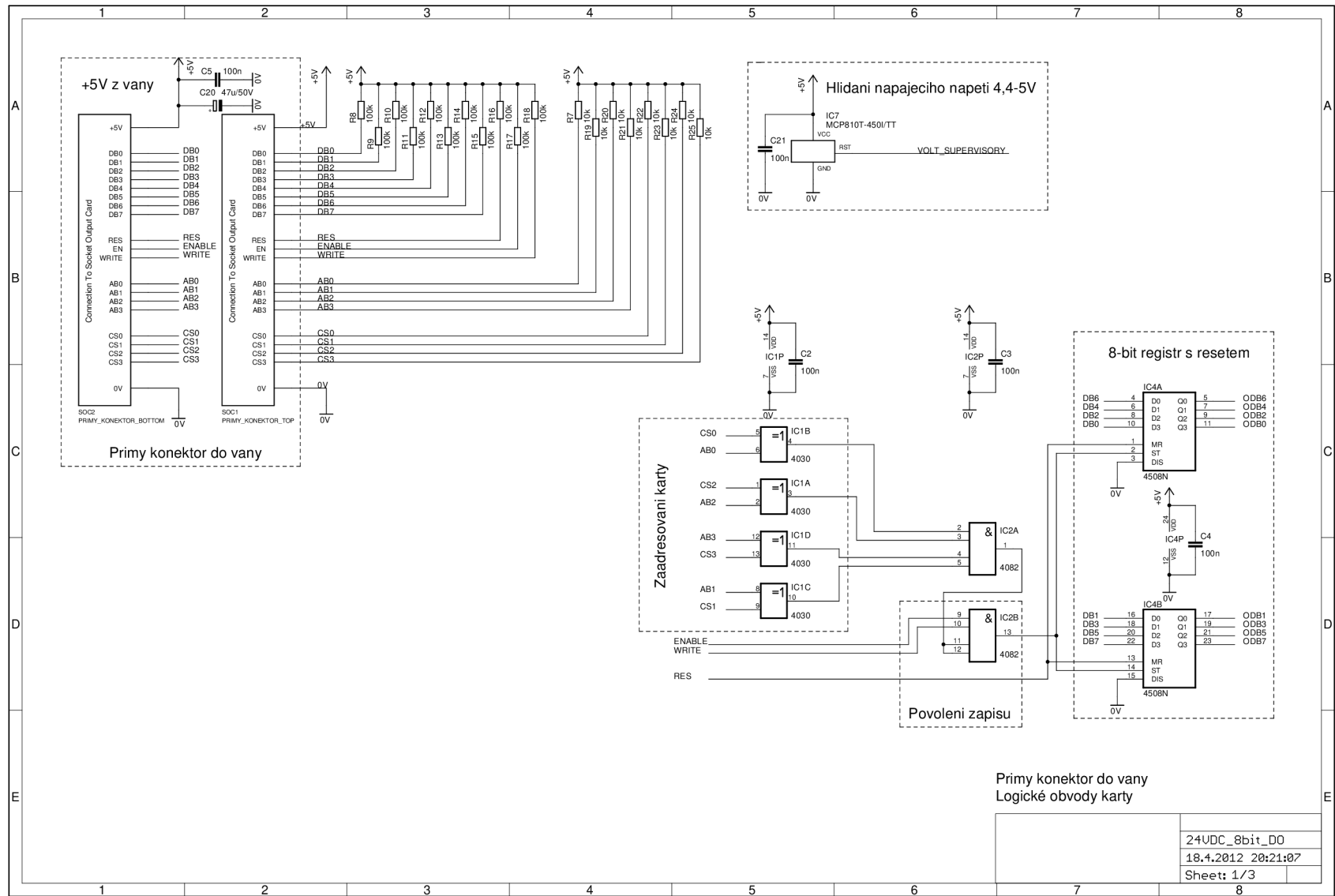
optočlenů IC5 a v pásmu zmíněného napájecího napětí uvede výstupy budiče IC5 do stavu vysoké impedance, tudíž bezpečně odpojí optočleny. Nedojde tedy k situaci, že se optočlen dostane ze stavu saturace do náhodného stavu (zakázané pásmo). Hodnota vypínacího napětí hlídače byla zvolena na základě testů (viz Obr. 4.2) a napájecího napětí budičů. Výše zmíněné řešení zajistí, že při vypnutí napájení vany budou výstupy v klidu deaktivovány i bez přítomnosti signálu *RES*, který přímo deaktivuje výstupy D-latch obvodu IC4 do log. 0.

4.3 Výstupy

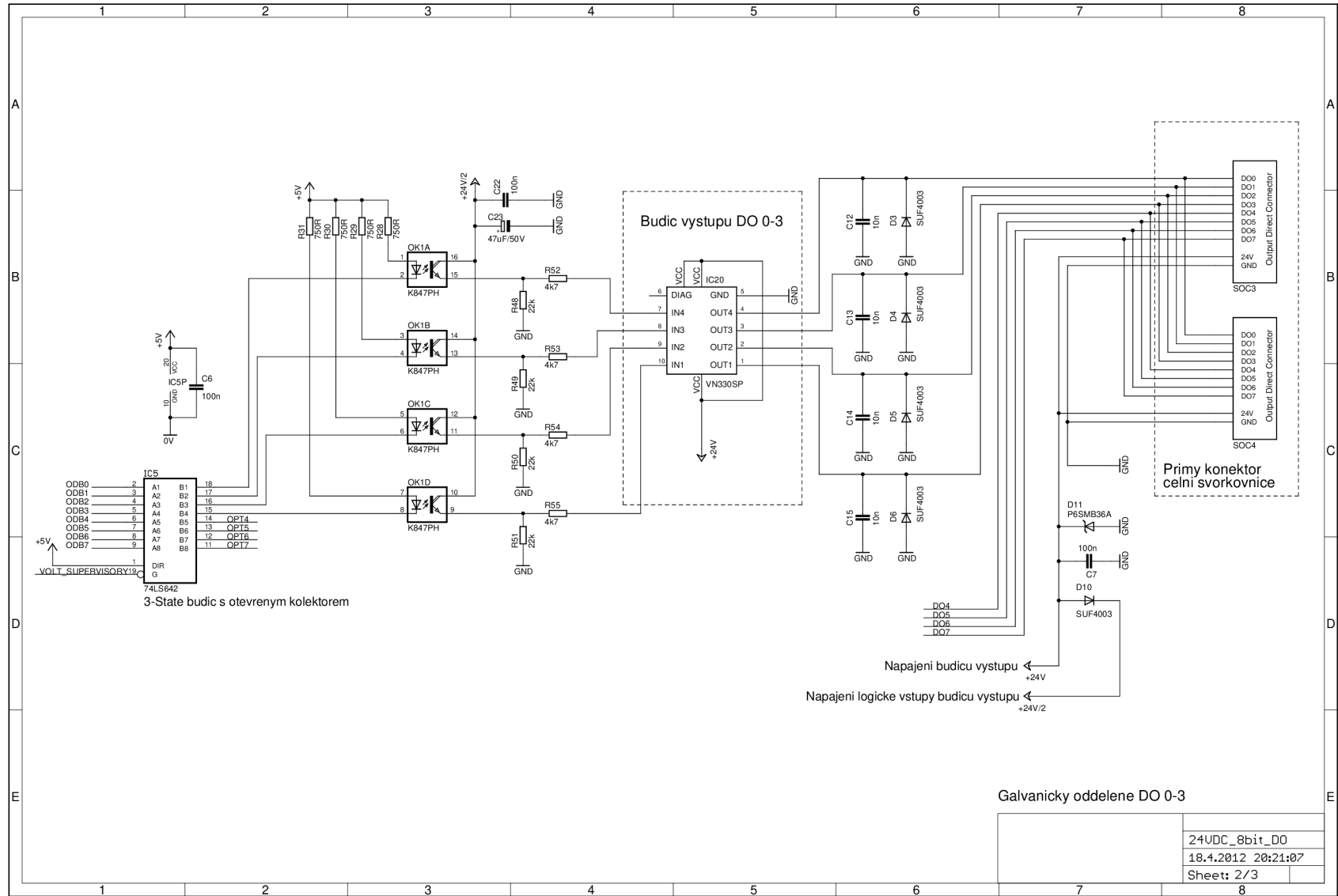
Napájecí napětí 24 VDC je na vstupu chráněno transilem D11 (4.4, D7), napájení pro logické vstupy budičů výstupů je odděleno přes diodu D10 (4.4, D7).

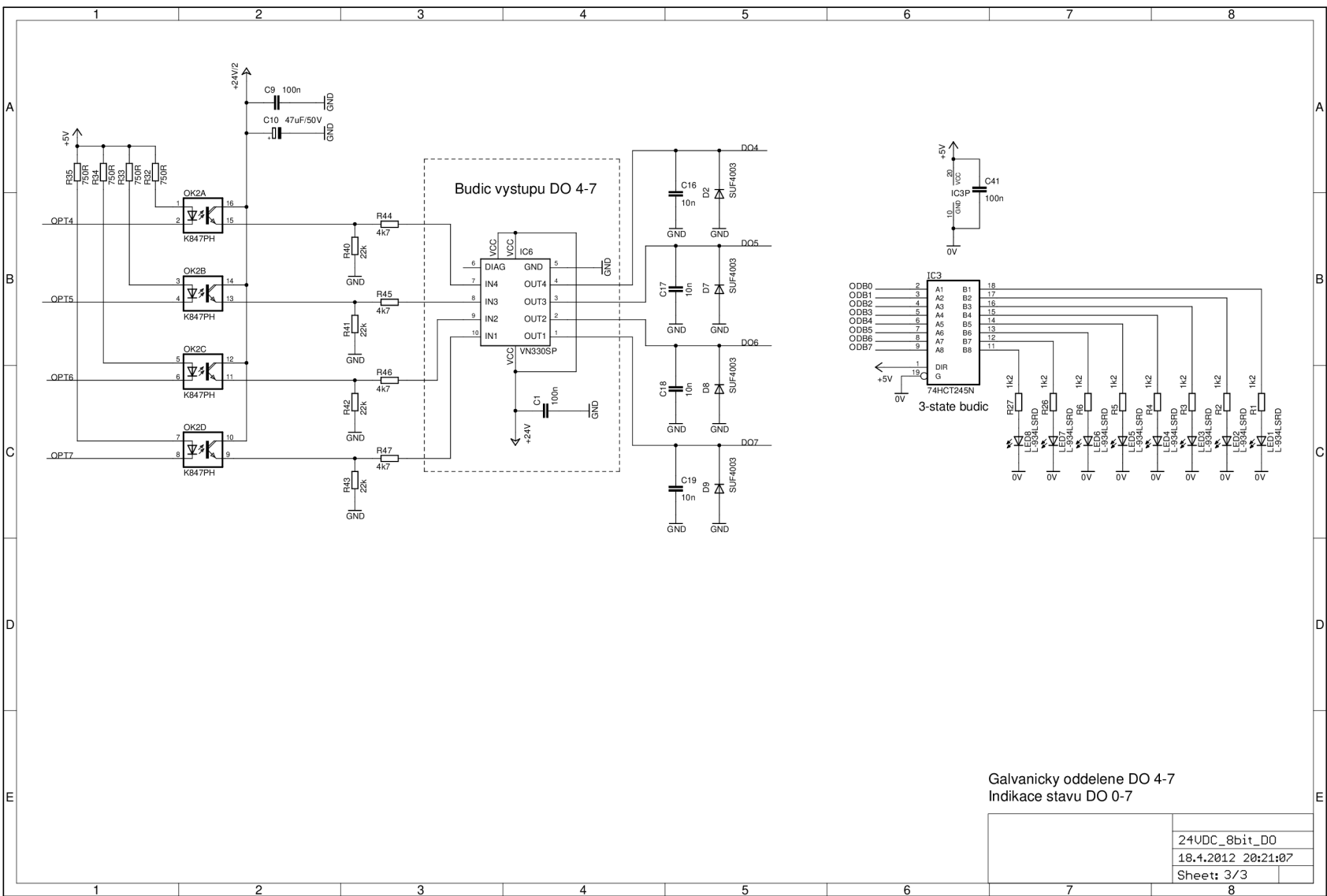
Pro buzení 8 výstupů jsou použity dva 4výstupové budiče MOSFET High Side Driver VN330SP IC20 (4.4, B5) a IC6 (4.5, B4), které mají zatížení 0,7 A na jeden výstupní kanál. Obvod má ochranu proti přetížení a zkratu i ztrátě zemního potenciálu. Speciální pouzdro PowerSO-10 umožňuje odvést teplo do měděné plochy na potenciálu napájení. Výkresy DPS obsahuje Příloha C.2, str. 119. Na výstupech budičů jsou diody v závěrném směru a kondenzátory 10 nF, obojí je opatření při odpínání indukční zátěže. Výstupy jsou zavedeny na přímý konektor, na který je v čelním panelu krytu karty zasunuta svorkovnice. (lit.[8])

Obr. 4.3: Výstupní karta 1/3 - ovládací logika karty



Obr. 4.4: Výstupní karta 2/3 - zapojení výstupů DO0 - 3





Obr. 4.5: Výstupní karta 3/3- zapojení výstupů DO4 - 7, LED pro indikaci stavu

Galvanicky oddelene DO 4-7
Indikace stavu DO 0-7

24VDC_8bit_DO	
18.4.2012 20:21:07	
Sheet: 3/3	

5 KOMUNIKAČNÍ PROTOKOL

Popis a možnosti protokolu byly navrženy pro přímé potřeby zařízení a lze je abstraktně přiřadit fyzické, linkové, síťové a aplikační vrstvě ISO/OSI modelu.

5.1 Fyzická vrstva - RS-232

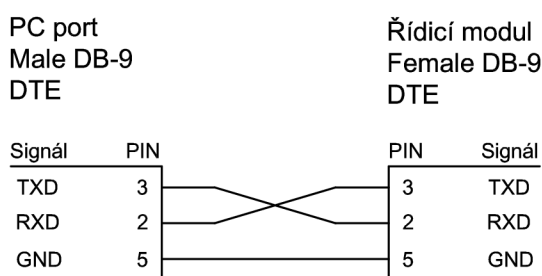
Obecná charakteristika

Jak již bylo zmíněno, pro komunikaci mezi PC a řídicím modulem je využito sériové linky RS-232 vycházející ze standardu TIA-232. Jedná se spojení dvou bodů obousměrnou bezkolizní metodou. Primárně byla RS-232 určen pro komunikaci s modemem dálkového přenosu přes telefonní linky, disponuje tedy větším množstvím vodičů. Z toho vychází označení počítače jako *Data Terminal Device (DTE)*, čili koncové zařízení komunikace. Modem odpovídá typu *Data Circuit-terminating Device (DTE)*, zprostředkovávající komunikaci po jiném médiu.

Pro užití v automatizaci je však dostačující použití následujících signálů: jeden směr dat *TXD*, druhý směr dat *RXD* a společný vodič *GND*.

V této konfiguraci signálů není k dispozici hardwarové řízení toku dat, používáno pro komunikaci s modemem, v automatizaci je zajištěno vyššími vrstvami ISO/OSI.

Při přímém propojení PC a řídicího modulu je uvažováno propojení dvou DTE zařízení, k čemuž je nutné použít překříženého vodiče (anglicky *nullmodem cable*). Standard specifikuje použití 9pinových nebo 25pinových D-Sub konektorů v definovaném zapojení. PC a řídicí modul disponuje 9pinovým konektorem, spojení mezi PC a řídicím modulem pomocí kabelu znázorňuje Obr. 5.1.



Obr. 5.1: Propojení mezi PC a řídicím modulem pomocí 9pinových D-Sub konektorů

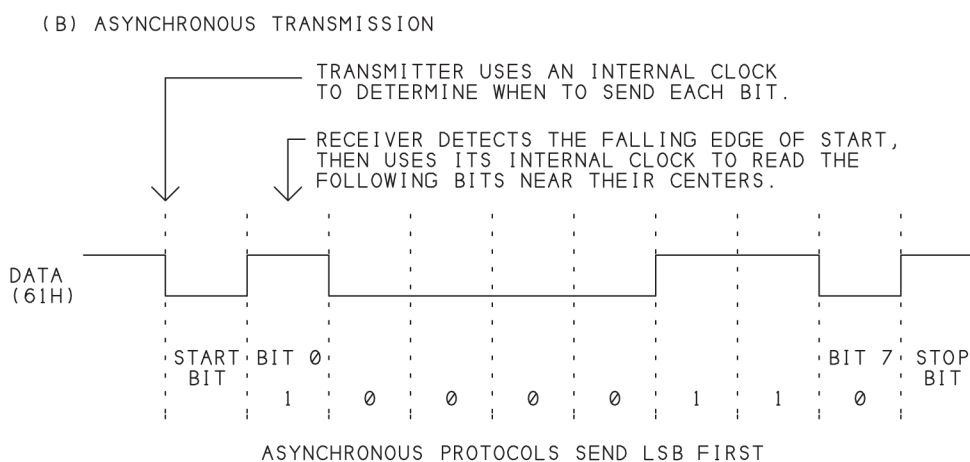
Pro přenos bitů platí stejné principy jako v případě asynchronního sériového rozhraní UART, liší se pouze napěťové úrovně logických hodnot (viz Tab. 5.1).

Logická úroveň	Směr komunikace	Napětí linky
log. 0	Vysílání	+5 až +15 VDC
log. 1	Vysílání	-5 až -15 VDC
log. 0	Příjem	+3 až +13 VDC
log. 1	Příjem	-3 až -13 VDC

Tab. 5.1: Napěťové úrovně linky RS-232 (lit.[9])

Základní jednotka přenosu dat se nazývá slovo (word). Slovo začíná vysláním tzv. start bitu (uvedení do log. 0), pomocí jehož sestupné hrany se synchronizuje přijímač dle vysílače, neboť obě strany mají vlastní zdroj hodin. Další bity přijímač vzorkuje blízko středu časového intervalu bitu třemi vzorky, výsledek je dán minimálně dvěma stejnými vzorky. Pro případ falešného start bitu v důsledku rušení kontrolují některé budiče i logickou úroveň v polovině start bitu.

Po start bitu následují, v pořadí od 0. bitu (LSb), datové bity v počtu 5 - 9 a za nimi je připojen paritní bit. Slovo je zakončeno jedním tzv. stop bitem (popřípadě 1,5 nebo 2 stop bity). Velice používaný formát slova je označován 8N1, obsahuje start bit, 8 datových bitů, bez paritního bitu a s jedním stop bitem (viz Obr. 5.2).



Obr. 5.2: Odeslání bajtu o hodnotě 61h pomocí UART při nastavení 8N1 (lit.[9])

Ve standardu TIA-232 není definována přenosová rychlost, pouze je uvedeno, že doporučení ohledně kabelu odpovídají rychlosti do 20 kb/s. Pro eliminaci vlivu vnějšího rušení, vyzařování a přeslechů lze vytvořit dva kroucené páry vodičů *TXD-GND* a *RXD-GND*. Ty mohou být umístěny ve stíněném kabelu, ovšem stínění zkracuje celkovou délku kabelu kvůli vyšší kapacitě vodiče. Dle dodatku standardu TIA-232 existuje vzorec pro výpočet délky popsaného stíněného vodiče (viz rovnice

5.1), maximální dosah pak při kapacitě vodiče 39 pF/m a kapacitě přijímače 100 pF odpovídá cca 20 m. Při konkrétní aplikaci je nutné dbát na uzemnění stínění pouze na jednom konci vedení (lit.[9]).

$$délka\ kabelu[m] = \frac{(2500 - kapacita\ přijímače[pF])}{(0.914 * kapacita\ kabelu[pF/m])} \quad (5.1)$$

Přenosové rychlosti jsou rovněž odvozeny od UART, např. 9600, 19200, 57600, 115200 b/s, přičemž vyšší přenosová rychlost klade požadavky na nižší celkovou kapacitu vodiče. To znamená kratší komunikační vzdálenost nebo nižší kapacitu kabelu.

Jelikož jsou logické hodnoty na datových vodičích *TXD* a *RXD* vyhodnocovány přijímačem vůči společnému vodiči *GND*, není žádoucí, aby společný vodič byl součástí vnějších obvodů, neboť procházející proud může vytvořit napěťovou špicí. To může způsobit nesprávné vyhodnocení logické úrovně přijímačem. Společný vodič *GND* musí být spojen s nulovým potenciálem zdrojů obou komunikujících jednotek. Rovněž je nutné vycházet z předpokladu, že dané jednotky jsou uzemněny na zemní svorku napájecí sítě. Oba konce společného vodiče *GND* jsou tedy uzemněny, společný vodič pak může tvořit nejnižší impedanční překážku (než samotná zem) a může ním protékat proud, tato situace je známá jako *zemní smyčka*. Na vzdálenosti kratší např. v rámci jedné místnosti bude vliv menší, obecně lze však doporučit jednu komunikační jednotku napájet z neuzemněného zdroje.

Tohoto řešení je využito na straně řídicího modulu, jehož komunikační obvod pro RS-232 je napájen z vlastního galvanicky odděleného zdroje a na straně komunikace UART galvanicky oddělen od CPU.

Použitá konfigurace

Použité nastavení komunikace mezi PC a řídicím modulem je následující:

- Formát: 8N2 (start bit, 8 datových bitů, žádná parita, 2 stop bity)
- Řízení toku dat: žádné
- Přenosová rychlost: 57600 b/s
- Kabel: stíněný (stínění uzemněno na straně PC portu), délka 2m

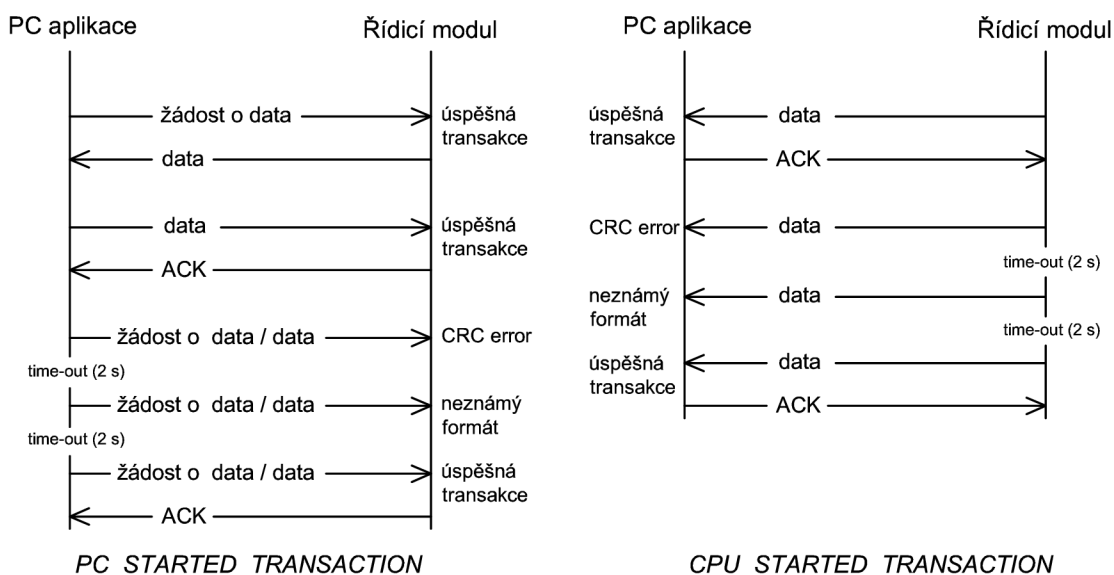
5.2 Linková vrstva

Obecná charakteristika

Druhá vrstva ISO/OSI modelu označovaná jako linková (Data Link Layer) propojuje fyzickou vrstvu s vyššími vrstvami. Jejím hlavním úkolem je zajištění mechanismu dorozumívání mezi komunikujícími jednotkami.

Pomocí RS-232 lze komunikovat bezkonfliktní formou pomocí dvou signálových vodičů. Je tedy možné, aby komunikaci zahajovaly obě strany. Transakce vyvolaná PC aplikací může být označena jako *PC_STARTED_TRANSACTION*, prostřednictvím PC aplikace jsou odesílána data do CPU řídicího modulu, respektive je z něho čteno. Pokud řídicí modul generuje svoji transakci typu *CPU_STARTED_TRANSACTION*, jedná se o odeslání dat ohledně vzniklé události, např. o poruše zařízení, dokončení cyklu, apod.

Vysílající jednotka očekává v případě úspěšného přijetí nepoškozeného rámce potvrzující ACK odpověď nebo datový paket. Pokud jej neobdrží v intervalu 2 s (pro přenosovou rychlost 57600 b/s), odešle ji znovu. Celkem realizuje tři pokusy, v případě neúspěchu transakce informuje vyšší vrstvy o selhání přenosu a přechází na další transakci. Druhá strana neodešle ACK v těchto případech: neodpovídá kontrolní součet, formát rámce je pro ni neznámý nebo jsou žádána data, která neexistují (viz Obr. 5.3).



Obr. 5.3: Transakce iniciovaná PC aplikací a CPU řídicího modulu

Rámec je uvozen bajtem „SZ“ o hodnotě B5h, která se nikde jinde uvnitř rámce nesmí vyskytovat (platí i pro kontrolní součet). Případný bajt o této hodnotě je překódován na dva bajty C500h, hodnota C5h na C501h. Po SZ bajtu je vložen paket síťové vrstvy.

Třetí bajt od konce rámce označuje číslo transakce „TN“, které je přiděleno vysílačem, přičemž ACK zpráva nebo datový paket odeslaná přijímačem musí bezpodmínečně obsahovat toto číslo transakce. Toto opatření je implementováno, aby vysílač obdržel odpověď na aktuální rámec a nikoli na nějaký předchozí, s jehož

zpracováním měl přijímač zpoždění. TN nabývá hodnoty 0 - 255.

Linková vrstva dále provádí výpočet kontrolního součtu CRC-16-CCITT, který je dosazen na pozici posledních dvou bajtů rámce. Výpočet je prováděn z celého rámce (viz Tab. 5.2).

SZ (1B)	paket síťové vrstvy	TN (1B)	CRC16-CCITT (2B)
---------	---------------------	---------	------------------

Tab. 5.2: Rámec linkové vrstvy

Popis algoritmu CRC-16-CCITT

Data tvořená jednotlivými bity lze uvést ve tvaru polynomu, např. 0110b odpovídá $x^2 + x$. Metoda kontrolního součtu používá klíč dle polynomu $x^{16} + x^{12} + x^5 + 1$, čemuž odpovídá binární ekvivalent 1 0001 0000 0010 0001b. Obvykle se při uvádění číselných hodnot nezmiňuje nejvyšší bit s hodnotou 1, jeho přítomnost je pouze demonstrativní, aby klíč měl specifikovanou délku.

Číslo „16“ udává aktivní délku klíče 16 bitů. Kódování je užíváno pro maximální délku 2048 bajtů. Odhalení poškození rámce je statisticky téměř jisté. Uvádí se, že metoda není odolná proti hackerům, kteří mohou změnit rámec i jeho kontrolní součet. Data jsou kódována dle polynomiálních operací (viz rovnice 5.2) (lit.[13]).

$$\begin{aligned}
 G(x) &= x^{16} + x^{12} + x^5 + 1 \dots \textit{polynom klíče} \\
 M(x) &= (0.\textit{bit})x^{n-1} + (1.\textit{bit})x^{n-2} + \dots + ((n-1).\textit{bit})x^0 \dots \textit{polynom dat} \\
 CRC &= \textit{zbytek}[(M(x) * x^{16})/G(x)] \dots \textit{kódování} \\
 n &\dots \textit{délka dat v bitech}
 \end{aligned}
 \tag{5.2}$$

5.3 Síťová vrstva

Obecná charakteristika

Tato vrstva provádí označení zprávy adresou jednotky, pro kterou je zpráva určena. Na této úrovni je posloupnost bajtů označována jako paket. Přestože se v tomto případě jedná o komunikaci mezi dvěma stanicemi prostřednictvím RS-232, protokol uvažuje o možnosti komunikace prostřednictvím např. RS-485 s využitím Master-Slave řízení sběrnice a připojením většího množství stanic. Je-li uvažován řídicí modul jako Slave stanice na sběrnici RS-485, nemohl by samozřejmě spontánně vysílat informace na linku jako bylo popsáno výše v souvislosti s linkovou vrstvou nad RS-232. Musí být zřejmé na jakou adresu stanice je zpráva odesílána.

Formáty paketů

Byly vytvořeny tři typy paketů, které mají odlišný formát hlavičky (viz podkapitola níže zde 5.3). Adresa Slave zařízení může nabývat hodnot od 0 - 255, PC (Master) nemusí mít adresu přidělenou. Potvrzující paket ACK odesílají univerzálně obě strany Master i Slave a neobsahuje žádnou datovou část.

Paket od PC (Master) pro CPU řídicího modulu (Slave):

00h (1B)	adresa (1B)	data aplikační vrstvy
----------	-------------	-----------------------

Zpráva nebo odpověď od CPU řídicího modulu pro PC:

01h (1B)	data aplikační vrstvy
----------	-----------------------

Potvrzení ACK:

02h (1B)	00h (1B)
----------	----------

5.4 Aplikační vrstva

Aplikační vrstva specifikuje formát a obsah dat posílaných prostřednictvím nižších vrstev na základě určité funkce. Vícebajtové hodnoty jsou odesílány tak, že nejméně důležitý bajt (LSB) je odeslán jako první. Obdobný princip jako v případě *Little Endian* realizace paměťového prostoru. V této kapitole budou pro přehlednost zobrazovány celé rámce linkové vrstvy.

Funkce 00h - Echo

(PC_STARTED_TRANSACTION)

Odesláno do CPU řídicího modulu při startu PC aplikace pro ověření komunikace. Rovněž je odeslána informace o aktuálním datu a času pro případ, že má CPU vybitou záložní baterii.

PC:

SZ	Adr	Fun 00h	Rok	Měsíc	Den roku	Den týdne
(1B)	(1B)	(1B)	(2B)	(1B)	(2B)	(1B)
Den	Hodina	Minuta	Sekunda	TN	CRC16	
(1B)	(1B)	(1B)	(1B)	(1B)	(2B)	

Řídicí Modul: ACK (nebo bez odpovědi)

SZ	02h	00h	TN	CRC16
(1B)	(1B)	(1B)	(1B)	(2B)

Funkce 01h - Nastavení výrobního cyklu (PC_STARTED_TRANSACTION)

Odesíláno na úvod při odesílání výrobního cyklu do automatu, následují pak jednotlivé kroky odesílané v rámci funkce 02h.

PC:

SZ	Adr	Fun 01h	PK	VP	OR	TPM
(1B)	(1B)	(1B)	(1B)	(1B)	(1B)	(2B)
TN	CRC16					
(1B)	(2B)					

Řídicí Modul: ACK (nebo bez odpovědi)

- PK počet kroků výrobního cyklu (0 - 99)
 VP informace o výchozí poloze (0. bit = 1, poloha aktivní)
 OR offset rychlosti (0 - 10)
 TMP hodnota, o kterou je snížena hodnota maxima tlaku kroku, při překonání výsledného tlaku je krok ukončen (0 - 4095)

Funkce 02h - Odeslání kroku výrobního cyklu (PC_STARTED_TRANSACTION)

Odeslání individuálního kroku do řídicího modulu. v případě že jeden krok je poškozen, je ignorován celý přesun výrobního cyklu, přičemž aktuální cyklus v CPU řídicího modulu je neúplně přepsán, tedy nefunkční a nelze spustit.

PC:

SZ	00h	Adr	Fun 02h	IK	SV	KTP
(1B)	(1B)	(1B)	(1B)	(1B)	(4B)	(1B)
ČUK	TUK	Delta	TN	CRC16		
(2B)	(2B)	(1B)	(1B)	(2B)		

Řídicí Modul: ACK (nebo bez odpovědi)

- IK index nahrávaného kroku výrobního cyklu (0 - 99)
 SV 4bajtová hodnota digitálních výstupů pro buzení ventilů (31 bitů)
 ČUK explicitní doba maximálního trvání kroku [ms]
 TUK tlak pro ukončení kroku v hodnotě 12 bitů (0-4095)
 Delta počet pohybujících se válců pro výpočet rychlosti měniče (0 - 31)
 KTP forma ukončení kroku (K - koncový snímač, T - čas, P - tlak)

0	0	ČK 1. bit	ČK 0. bit	0	K	T	P
7	6	5	4	3	2	1	0

ČK 2bitova hodnota adresy koncového snímače (0-3)

Funkce 03h - Odeslání manuálního kroku

(PC_STARTED_TRANSACTION)

Transakce je vyjma hodnoty bajtu Fun 03h zcela totožná z předchozí. Pro účel manuálního vyjetí v režimu stroje M3 jsou využita pouze stavy výstupů „SV“.

Funkce 04h - Čtení statistiky

(PC_STARTED_TRANSACTION)

O statistická data uživatel žádá prostřednictvím nabídky v PC aplikaci.

PC:

SZ	00h	Adr	Fun 04h	TN	CRC16
(1B)	(1B)	(1B)	(1B)	(1B)	(2B)

Řídicí Modul:

SZ	01h	Fun 04h	C	ČB	AHT	TN	CRC16
(1B)	(1B)	(1B)	(4B)	(4B)	(4B)	(1B)	(2B)

C počet uplynulých výrobních cyklů

ČB ... uplynulá doba běhu hydraulického agregátu [s]

AHT . aktuální hodnota tlaku v obvodu (0 - 4095)

Funkce 04h - Odeslání statistiky

(CPU_STARTED_TRANSACTION)

CPU řídicího modulu odesílá spontánně stav statistických hodnot při ukončení cyklu, aby výpočtem rozdílu s minulou hodnotou počtu výrobních cyklů bylo aktualizováno počítadlo v PC aplikaci.

Řídicí Modul:

SZ	01h	Fun 04h	C	ČB	AHT	TN	CRC16
(1B)	(1B)	(1B)	(4B)	(4B)	(4B)	(1B)	(2B)

PC: ACK (nebo bez odpovědi)

Funkce 05h - Odeslání události

(CPU_STARTED_TRANSACTION)

Při výskytu události (informace, porucha, varování) CPU řídicího automatu spontánně odešle kód události a přesný čas jejího výskytu. V PC aplikace „vyskočí“ informační okno.

Řídicí Modul:

SZ	01h	Fun 05h	Rok	Měsíc	Den	Hodina	Minuta
(1B)	(1B)	(1B)	(2B)	(1B)	(1B)	(1B)	(1B)
Sekunda	Kód	TN	CRC16				
(1B)	(1B)	(1B)	(2B)				

PC: ACK (nebo bez odpovědi)

Funkce 06h - Odeslání vyhledaných maxim tlaku

(CPU_STARTED_TRANSACTION)

V okamžiku, kdy CPU řídicího automatu vyhledá pomocí algoritmu maxima tlaku při dojetí kroku, odešle spontánně data do PC.

Řídicí Modul:

SZ	01h	Fun 06h	PT	NHT ...	TN	CRC16
(1B)	(1B)	(1B)	(1B)	(PT*(2B))	(1B)	(2B)

PC: ACK (nebo bez odpovědi)

PT ... počet kroků (maxim tlaku) (0 - 99)

NHT . . . nalezené 2bajtové hodnoty maxim tlaku řazené za sebou (0 - 4095)

Funkce 07h - Odeslání aktuálního kroku při krokování

(CPU_STARTED_TRANSACTION)

Implementováno v režimu běhu M3.

Řídicí Modul:

SZ	01h	Fun 07h	IAK	TN	CRC16
(1B)	(1B)	(1B)	(1B)	(1B)	(2B)

PC: ACK (nebo bez odpovědi)

IAK . . . index aktuálního kroku (0 - 99)

Funkce 08h - Čtení reálného času

(PC_STARTED_TRANSACTION)

Načtení hodnoty času z registrů RTC obvodu CPU řídicího modulu na žádost uživatele v PC aplikaci.

PC:

SZ	00h	Adr	Fun 08h	TN	CRC16
(1B)	(1B)	(1B)	(1B)	(1B)	(2B)

Řídící modul:

SZ	01h	Fun 08h	Rok	Měsíc	Den roku	Den týdne
(1B)	(1B)	(1B)	(2B)	(1B)	(2B)	(1B)
Den	Hodina	Minuta	Sekunda	TN	CRC16	
(1B)	(1B)	(1B)	(1B)	(1B)	(2B)	

Funkce 09h - Zápis reálného času (PC_STARTED_TRANSACTION)

Nastavení reálného času RTC obvodu v CPU řídicího modulu na žádost uživatele v PC aplikaci.

PC:

SZ	00h	Adr	Fun 09h	Rok	Měsíc	Den roku
(1B)	(1B)	(1B)	(1B)	(2B)	(1B)	(2B)
Den týdne	Den	Hodina	Minuta	Sekunda	TN	CRC16
(1B)	(1B)	(1B)	(1B)	(1B)	(1B)	(2B)

Řídící modul: ACK (nebo bez odpovědi)

Funkce 0Ah - Čtení CPU dat (PC_STARTED_TRANSACTION)

Ve speciálním formuláři PC aplikace lze přečíst speciální CPU data, které nejsou standardnímu uživateli přístupné a slouží k oživení a nastavení stroje. Pomocí následné funkce 0Bh lze vybrané parametry i přenastavit. Jedná se o RTC čas CPU aj.

PC:

SZ	00h	Adr	Fun 0Ah	TN	CRC16
(1B)	(1B)	(1B)	(1B)	(1B)	(2B)

Řídící modul:

SZ	01h	Fun 0Ah	PD	D...	TN	CRC16
(1B)	(1B)	(1B)	(1B)	(PD*(4B))	(1B)	(2B)

PD ... počet 4bajtových procesních dat (0 - 255)

D procesní data řazena za sebou

Funkce 0Bh - Zápis CPU dat
(PC_STARTED_TRANSACTION)

(viz předchozí)

PC:

SZ	00h	Adr	Fun 0Bh	PD	D...	TN	CRC16
(1B)	(1B)	(1B)	(1B)	(1B)	(PD*(4B))	(1B)	(2B)

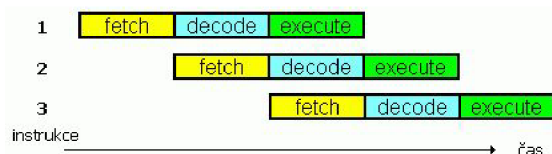
Řídicí modul: ACK (nebo bez odpovědi)

6 FIRMWARE CPU ŘÍDICÍ DESKY

6.1 Architektura CORTEX-M3

Použité CPU NXP LPC1756 je založeno na 32bitovém jádru ARM Cortex-M3, které bylo od počátku navrženo jako zástupce architektury ARM7-M pro vysoký výkon a efektivitu aplikace v embedded zařízeních. Tvoří jej pouze 33 tisíc transistorů, jedná se o nejmenší jádro ARM i s optimalizací velikosti okolních periférií. Změny v základní architektuře přinesly hlavně zjednodušení tvorby strojového kódu. Efektivnější zpracování strojového kódu přineslo zvýšení výkonu. V testech s kvantem operací pro jeden strojový cyklus může být dosaženo poměru až 1,25 MIPS/MHz.

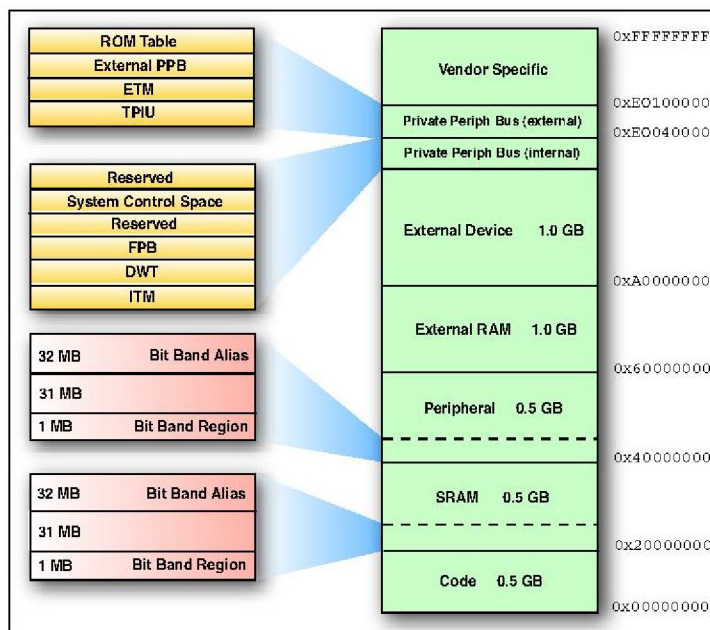
Cortex-M3 procesory jsou založené na Harvardské architektuře se třemi fázemi zpracování dat (zřetězení neboli pipeline), přičemž během jednoho strojového cyklu proběhne zpracování jedné instrukce, dekodování druhé instrukce a vyzvednutí třetí instrukce z paměti (viz Obr. 6.1). Tato schopnost společně s predikcí vede ke zrychlení zpracování instrukcí a je-li při načítání instrukce známo pokračování kódu, není nutné vykonat instrukci skoku a na stack jsou načítány data z cílové oblasti (lit.[3]).



Obr. 6.1: Pipelining (lit.[3])

Jádro Cortex-M3 obsahuje 32bitovou datovou sběrnici, 13 registrů pro všeobecné použití, dva ukazatele na zásobník, programový čítač, registr vazby, stav rejstříku a sadu speciálních registrů. Obsahuje dekodér pro instrukce ARM, Thumb i nové Thumb-2. Disponuje 4GB adresovým prostorem, kam patří paměť Flash, RAM, externí paměť a I/O prostor periférií. Jistá část paměti je využita dle konkrétního výrobce CPU (viz Obr. 6.2).

CPU Cortex-M3 disponuje novou instrukční sadou Thumb-2, která umožňuje dosáhnout až o 70% vyššího výkonu než klasické ARM na jádře ARM7TDMI a s instrukční sadou Thumb. Důvodem je efektivnější využití prostředků, hlavně paměti, řídicí logiky a rozšířené ALU s hardwarovými operacemi násobení a dělení. Navíc přináší možnost rychlejšího vytvoření efektivního strojového kódu bez vložek assembleru, což nevyžaduje na programátorovi hluboké znalosti architektury ani při vytváření středně náročných aplikací. Typickým příkladem je řízení přerušování, která vytvoří jednoduchý kód a bez manipulace s registry, především protože tyto



Obr. 6.2: Paměťový prostor Cortex-M3 (lit.[10])

CPU využívají zjednodušený programovací model zásobníku, který vychází z ARM a 8bitových nebo 16bitových procesorů.

Stěžejním přínosem instrukční sady Thumb-2 je účinnější C kompilátor, který umožňuje operace s bity a bitovými poli, hardwarovým tříděním a účinnější provádění podmínek typu „if/then“. Snížila se také frekvence přístupu do paměti v souvislosti s nasazením zmíněné sady a taktéž změnou práce bitových operací. Tradiční ARM CPU podporují manipulaci s celými datovými slovy, respektive zarovnávají operace na 32 bitů. Cortex-M3 podporuje práci s nezarovnanými daty, což zrychluje přístup k těmto datům. Krom toho podporuje 32bitové násobení v jediném cyklu a obecně práce se znaménkovými a neznaménkovými typy, které zaujímají 2 až 12 taktů v závislosti na velikosti operandů.

Konfigurovatelný blok přerušení NVIC generuje ve výchozím nastavení nemaskované přerušení a 32 fyzických utilit s 8 hladinami priority, může být ale rekonfigurován na generování 1-240 fyzických přerušení s 256 úrovněmi priority. NVIC obsahuje 24bitový časovač používaný pro získání přerušení. Řízení NVIC je nově implementováno hardwarově a nasazen je nově i systém tzv. dokování přerušení, což spočívá v cyklovém procesu přímého přesunu z obsluhy jednoho přerušení do druhého přerušení. Toto řešení je o mnoho rychlejší než softwarová implementace obsluhy přerušení jako v případě klasických architektur, kdy jsou data stavových registrů obnovována a zálohována (lit.[3]).

6.2 Vytvořené zdrojové soubory

Firmware CPU byl vytvářen ve vývojovém nástroji CooCox CoIDE 1.2.4, který je volně ke stažení a k dispozici na přiloženém CD v kořenovém adresáři. Prostřednictvím rozhraní JTAG bylo provedeno ladění přímo na čipu programátorem „OpenJTAG ARM JTAG USB“ ze zmíněného vývojového prostředí.

my_com.h, my_com.c	implementace komunikačního protokolu
my_CoOS.h, my_CoOS.c	nastavení CoOS, úloh a stacku
my_crc.h, my_crc.c	výpočet kontrolního součtu
my_data.h, my_data.c	inicializace a správa dat výrobního cyklu, flagů pro komunikaci
my_events.h, my_events.c	obsluha procesních událostí
my_fat32.h, my_fat32.c	převzatá knihovna funkcí pro manipulaci se soubory FAT32
my_gpio.h, my_gpio.c	nastavení a přístup k I/O pinům, inicializace a ovládání sběrnice vany
my_logging.h, my_logging.c	obsluha zápisu do logu
my_pinsel.h, my_pinsel.c	nastavení funkce jednotlivých pinů CPU
my_pomocne.h, my_pomocne.c	pomocné funkce pro FAT32
my_pressure.h, my_pressure.c	příjem hodnoty z ADC, filtrace měření, hledání maxima tlaku
my_program.h, my_program.c	řídící algoritmus procesu
my_quickselect.h, my_quickselect.c	převzatá knihovna pro výpočet mediánu
my_rtc.h, my_rtc.c	nastavení a čtení reálnému času, čtení a zápis do zálohované paměti
my_sdcard_call.h, my_sdcard_call.c	nastavení instance softwarového spi
my_sdcard.h, my_sdcard.c	převzatá knihovna funkce pro přístup na SD kartu jako médium
my_softspi_call.h, my_softspi_call.c	nastavení instance softwarového spi
my_softspi.h, my_softspi.c	reentrantní funkce softwarového spi
my_spi.h, my_spi.c	inicializace SSP1 kanálu pro SPI komunikaci s ADC převodníkem analogových vstupů
my_system.h, my_system.c	inicializace CPU, testování příčin resetu
my_systick.h, my_systick.c	obsluha Systick časovače (nepoužíváno),
my_timer.h, my_timer.c	inicializace a obslužná rutina pro časovač Timer0
my_uart.h, my_uart.c	inicializace a funkce pro zápis a čtení na UART3
my_wdt.h, my_wdt.c	obsluha WatchDog časovače CPU, ovládání budiče WatchDog relé

Tab. 6.1: Seznam vytvořených zdrojových a hlavičkových souborů

Vytvořené funkce volají pro přístup k perifériím CPU API funkce poskytnuté

výrobcem NXP. Tyto soubory jsou označeny prefixem `lpc17xx_` nebo `lpc_` a nacházejí se ve složce `\CoxOS\proj01\lpc17xx_lib` společně s vytvořenými zdrojovými soubory. Zdrojové soubory pro manipulaci s jádrem CORTEX-M3 na nejnižší úrovni (přímo od společnosti ARM) obsahují složky `\CoxOS\proj01\cmsis`, respektive `\CoxOS\proj01\cmsis_boot`. Zdrojové soubory operačního systému CoOS obsahuje složka `\CoxOS\proj01\CoOS`

Poskytnuté zdrojové kódy umožňují programování na vyšší úrovni, zpřehledňují kód a zrychlují implementaci firmware. Vytvořené zdrojové soubory v rámci této práce jsou přehledně pojmenovány podle periférie CPU nebo funkce, kterou vykonávají (viz Tab. 6.1).

6.3 RTOS CoCoX CoOS

Základní přehled

Tento operační systém byl autory, organizací CoCoX, vytvořen speciálně pro sérii CPU Cortex-M, přičemž disponuje následujícími možnostmi (lit.[4]):

- Free, open source RTOS
- Velikost jádra 974 B
- Adaptivní plánovač úloh
- Podpora preemptivního a round-robin multitaskingu
- Interrupt latency 0
- Synchronizační prostředky: semafor, mutex, flag, mailbox, fronta (queue)
- Možnost detekce přetečení zásobníku

Úlohy (Tasks)

Každá úloha (task) je tvořena funkcí, která obsahuje nekonečnou smyčku a definuje se ve formátu: `void myTask (void* pdata) { for(;;){ } ,` úloha se ukončí vlastním voláním funkce `CoExitTask ()`, popř. může ukončit jinou úlohu `taskID` pomocí funkce `CoDelTask (taskID)`.

Každá úloha má vlastní instanci struktury TCB (Task Control Block), ve které jsou uloženy informace o stavu úlohy, prioritě, o synchronizačních prostředcích, na které čeká, a ukazatele na strukturu TCB předchozí a následující úlohy ve frontě.

Plánování úloh (Task Scheduling)

CoCoX CoOS podporuje dvě varianty plánovače: preemptivní pro řazení úloh různé priority a round-robin pro řazení úloh stejné priority. K plánování dojde zaprvé

v situaci, kdy úloha s vyšší prioritou přejde do stavu READY. V druhé situaci dojde k tomu, že běžící úloha RUNNING je zablokována čekáním např. na událost a je převedena do stavu WAITING. V třetí situaci běží úloha stejné priority jako READY úloha, RUNNING úloha vyčerpala svůj díl a CPU je přiděleno READY úloze.

Kritická sekce (Critical Section)

Na rozdíl od jiných RTOS, CoOS neřeší zajištění kritické sekce zakázáním přerušování, ale zamknutím plánovače. To má za následek sníženou latenci a zvýšení real-time odezvy. Samozřejmě má to svá omezení, neboť uživatelská úloha nesmí volat funkce, které zmrazí nebo zruší běh úlohy v kritické sekci, např. ukončení úlohy `CoExitTask ()`, čekání `CoTickDelay ()`, `CoTimeDelay ()`, apod.

Přerušování (Interrupts)

V případě, že ISR volá v sobě nějaké API funkce CoOS musí být definována následovně: `void myISR (void) { CoEnterISR(); ... CoExitISR(); }`

Příčemž pokud nastane situace, že je přerušování vyvoláno v okamžiku uzamčení plánovače, systém ISR odešle do fronty pro zpracování a obsluží ji po odemčení plánovače.

Systémové hodiny (System Ticks)

Jako zdroj systémových hodin je využit hardwarový časovač `Systick`. CoOS má konfigurační soubor `config.h`, ve kterém lze změnou hodnoty makra `CFG_SYSTICK_FREQ` nastavit systémové hodiny v rozsahu 1-1000 Hz. V ISR časovače je inkrementována proměnná s časem systému. Mimoto pokud čekají některá vlákna voláním `CoTickDelay ()`, `CoTimeDelay ()`, je v ISR dekrementována doba čekání úloh v jejich struktuře TCB. Po opuštění ISR je volán plánovač úloh.

CoOS má k dispozici až 32 softwarových časovačů, které mají expirace v násobcích systémových hodin. Časovač je vytvořen voláním funkce `CoCreateTmr ()`, což má za následek přidání nové instance struktury `TimCB` (Timer Control Block) do fronty časovačů (obdobné jako úlohy). Expirace časovače proběhne v ISR `Systick`, a musí zde být i spuštěna obslužná rutina softwarového časovače, tzv. *Callback function*, která musí být co nejkratší.

Ověřování přetečení zásobníku (Stack Overflow Check)

Každá úloha má při vytváření alokovaný zásobník, jeho spodní adresa je uložena do struktury TCB. Systém kontroluje stav zásobníků při každém plánování úlohy. Pokud

se prokáže přetečení zásobníku, je zavolána funkce:

```
void CoStkOverflowHook(OS_TID taskID) { for(;;){...} },
```

kteřá implicitně obsahuje nekonečnou smyčku, nicméně její kód lze předefinovat v souboru hook.c (lit.[4]).

6.4 Implementace firmware

Struktura

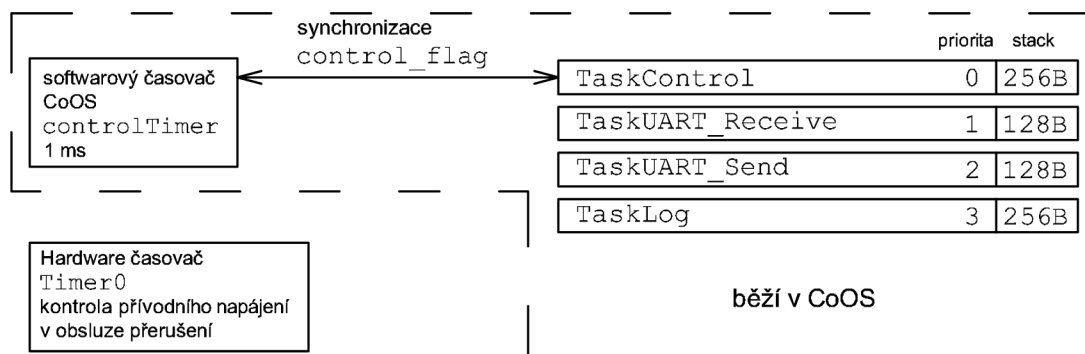
Firmware CPU je postaven na použití výše popsaného operačního systému CoCox CoOS. Přičemž jak je patrné z vývojového diagramu funkce `Main` (viz Obr. 6.4), služby operačního systému jsou spuštěny až po inicializaci všech periférií (lit.[11]).

Úlohy běžící v systému včetně parametrů přehledně zobrazuje Obr. 6.3, přičemž nejvyšší prioritu má úloha `taskControl`, která zabezpečuje běh řídicího algoritmu procesu. Úloha `taskControl` čeká na flag od 1ms softwarového časovače. Časovač `Timer0` běží mimo operační systém a úkol jeho ISR `TIMER0_Handler()` je periodicky skenovat hlášení od hlídání vnějšího napájení (viz Obr. 6.5).

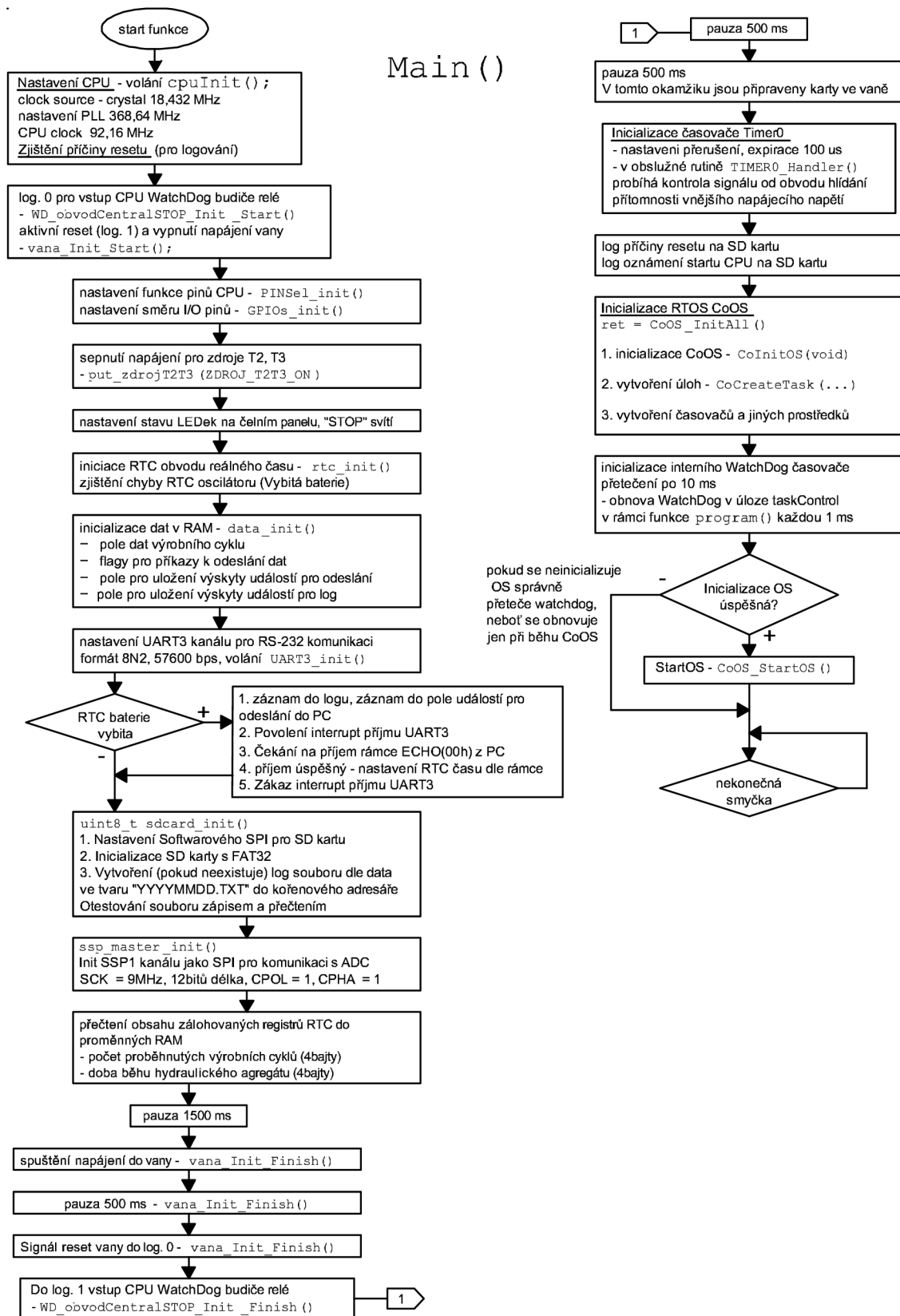
Z důvod bezpečnosti a rychlosti byla zvolena varianta předávání dat úlohám s nižší prioritou nikoli přes buffer s exkluzivním přístupem, nýbrž skrze struktury v paměti. Příkladem je záznam procesní události v úloze `taskControl` do položky dle typu události pole `EventData EventDataArray[EVENTS_NUMBER]`.

Programové řešení

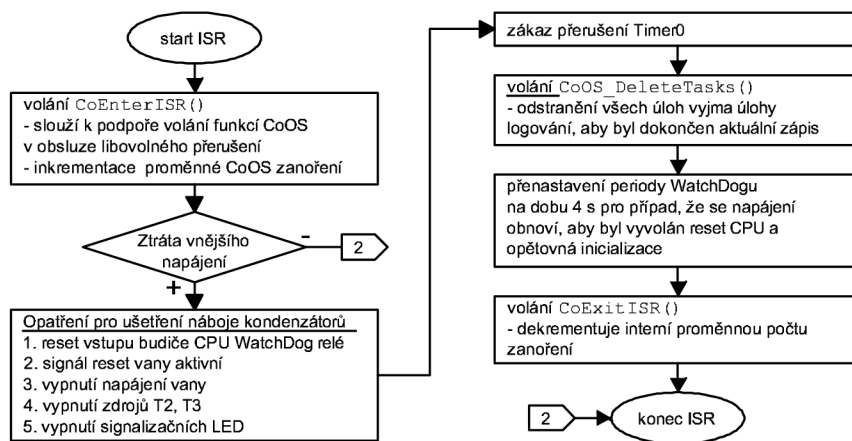
Nyní navazuje vyobrazení vývojových diagramy mj. jednotlivých úloh s kompletním vysvětlením kroků a použitých prostředků.



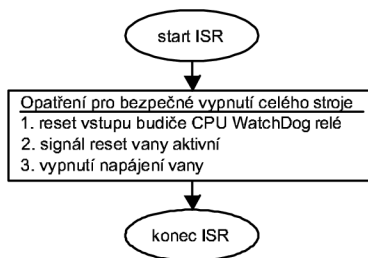
Obr. 6.3: Procesy běžící v systému



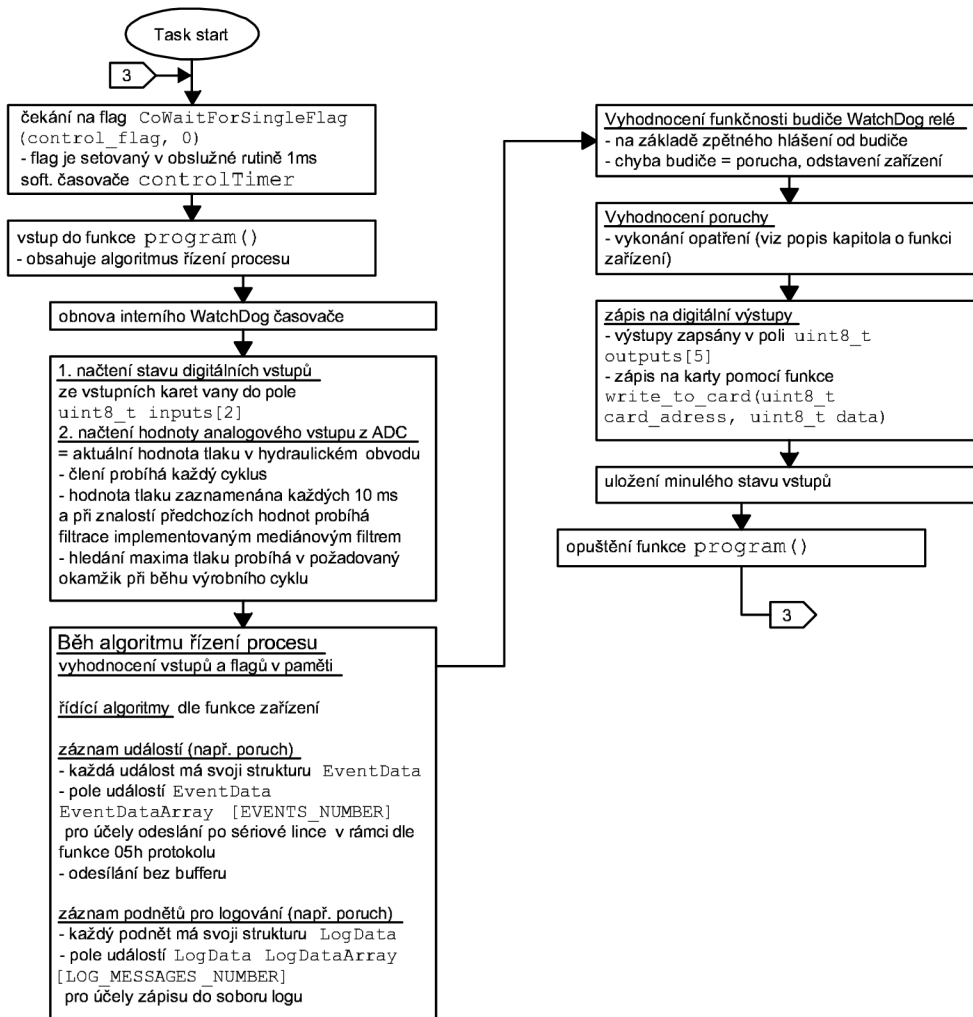
Obr. 6.4: Inicializace prostředků a RTOS CoOS Main ()



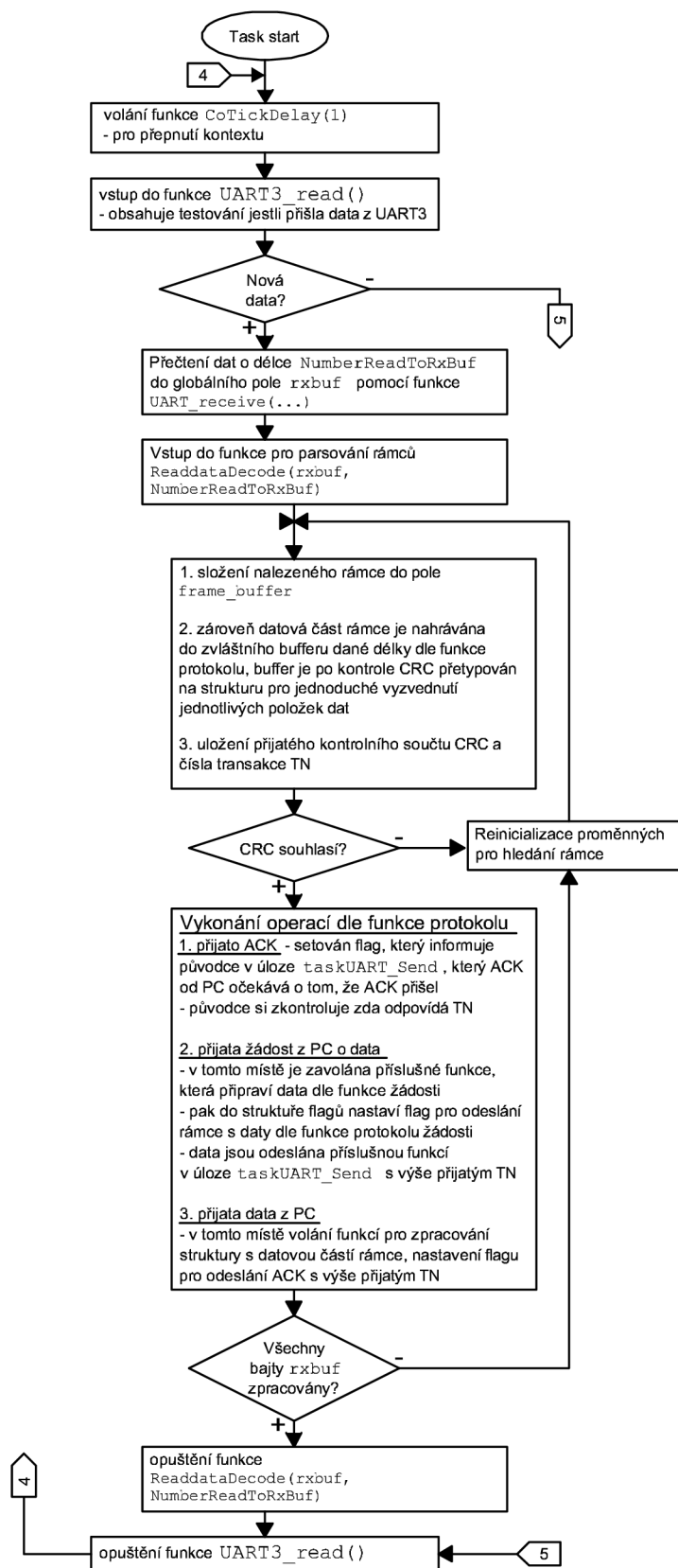
Obr. 6.5: Obslužná rutina `TIMER0_Handler()` přerušení časovače `Timer0`, hlídání přívodního napájení



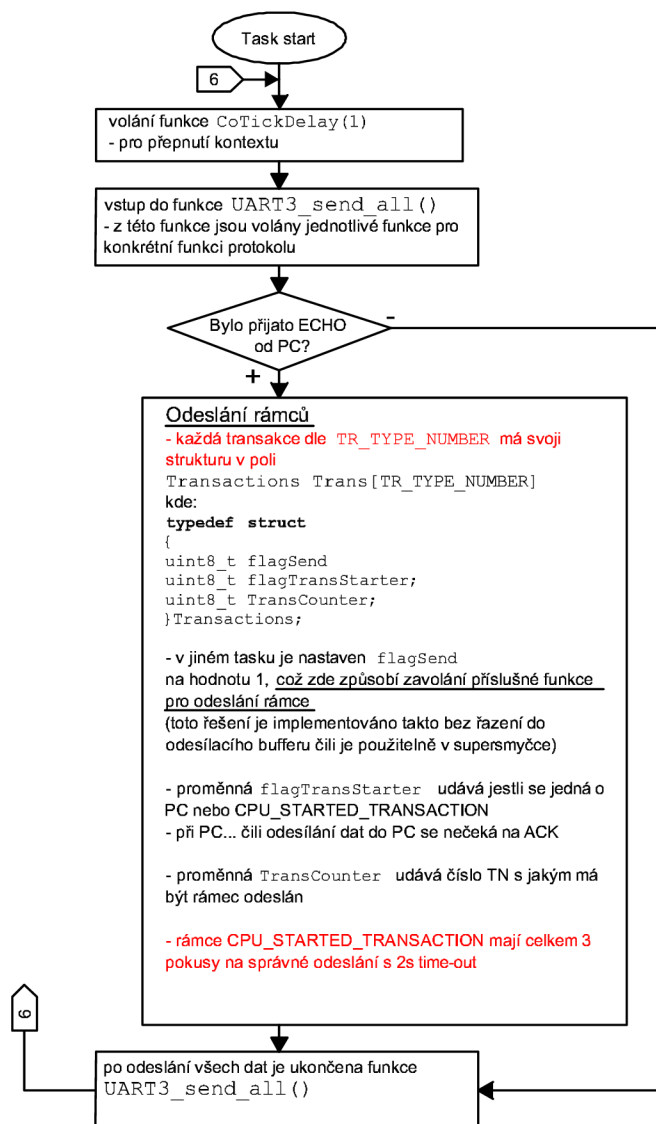
Obr. 6.6: Obslužná rutina `WDT_IRQHandler()` při přetečení `WatchDog`



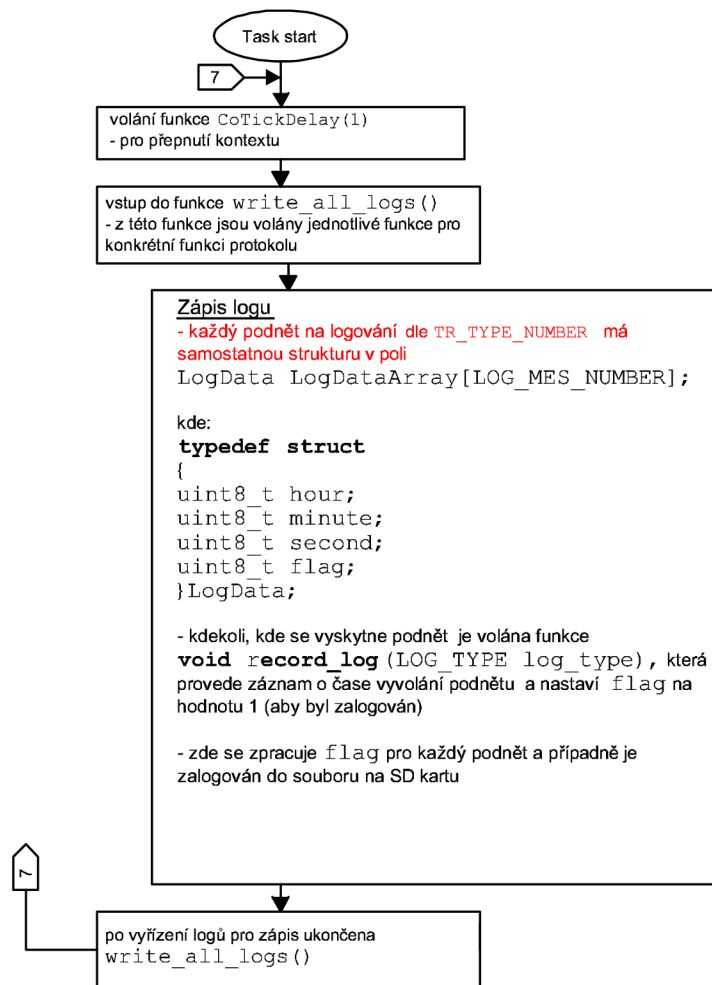
Obr. 6.7: Úloha taskControl obsluhující řízení procesu



Obr. 6.8: Úloha taskUART_receive pro příjem dat ze sériové linky (UART3)



Obr. 6.9: Úloha `taskUART_send` odeslání rámců po sériové lince (UART3)

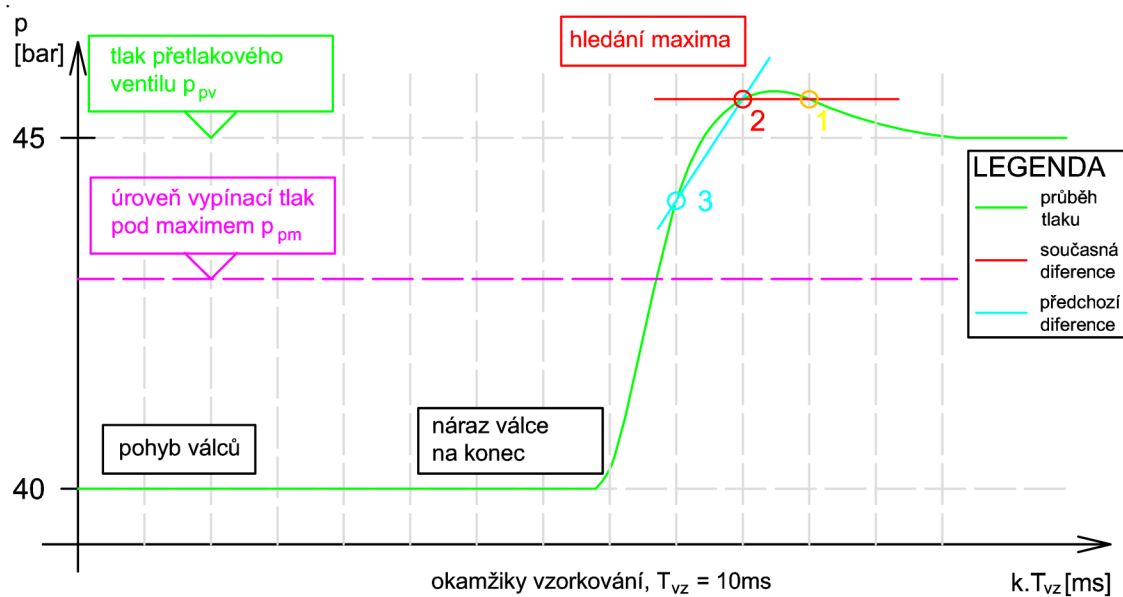


Obr. 6.10: Úloha taskLog pro zápis logu na SD kartu

Hledání maxima tlaku

Průběh tlaku v hydraulickém obvodu během kroku výrobního cyklu obsahuje Obr. 6.11, přičemž vzorkování je prováděno s periodou 10 ms. Signál je filtrován mediánovým filtrem o šíři okna pět dílků. Algoritmus vyhodnocuje podmínku nalezení maxima tlaku dle rovnice 6.1, ve které vystupují tlaky na označených bodech křivky na zmíněném Obr. 6.11.

$$(p_2 - p_3) > 0 \cap (p_1 - p_2) \leq 0 \Rightarrow p_2 = p_{max} \quad (6.1)$$



Obr. 6.11: Průběh tlaku v hydraulickém obvodu ke konci kroku výrobního cyklu

Hodnotu rozdílu mezi nalezeným maximem p_{max} a vypínacím tlakem pod maximumem p_{pm} zadává uživatel z PC. Po nalezení maxima automat vypočte pomocí zadaného rozdílu hodnotu p_{pm} a ukončuje krok v okamžiku, kdy tlak v obvodu překoná hodnotu p_{pm} . Na Obr. 6.11 by byl krok ukončen vzorkem v bodě 3.

7 PC APLIKACE

7.1 Základní charakteristika

Aplikace pro komunikaci s CPU řídicího automatu byla vytvořena ve vývojovém nástroji Microsoft Visual Studio 2010 s použitím objektového programování v programovacím jazyku *C#*. Aplikace vyžaduje pro běh nainstalovanou verzi platformy .NET4.0. Kompletní sestava („Solution“) okomentovaných strojových souborů je na přiloženého CD (viz Elektronická příloha A, str. 92) pod názvem „Rizeni_Automat_Plisky“.

Aplikace byla navržena pro potřeby obsluhy tvarovacího stroje. Strukturu aplikace dle tříd s danou funkcí je shrnuta níže, přičemž daná třída je definována v souboru stejnojmenného názvu s příponou *.cs. Třídy oken aplikace (Form) mají obvykle konstruktor se dvěma základními parametry (jako reference): třída datového zdroje (nejčastěji proměnná ve třídě v MDIMain) a třída nadřazeného okna tedy MDIMain. Okno má tedy přístup k proměnným nadřazeného okna.

- grafické rozhraní (GUI)
 - hlavní okno - MDIMain (objekt Windows Form MDIParent)
 - okno programu výrobního cyklu zobrazované v hlavním okně - Form1 (objekt Windows Form)
 - ostatní okna jsou zobrazovány mimo hlavní okno v popředí
 - okno nastavení COM portu - FormConnectSettings (objekt Windows Form)
 - okno nastavení CPU dat - FormControlCPUDataSettings (objekt Windows Form)
 - okno pro chráněný přístup k nastavení CPU dat - FormLogin (objekt Windows Form)
 - okno pro zobrazení procesní události z řídicího modulu - FormProcessEventInfoWindow (objekt Windows Form)
 - okno, zobrazení historie procesních událostí - FormProcessEvent (objekt Windows Form)
- komunikace s CPU řídicího modulu
 - sériový COM port - serialPort1 (instance ve třídě MDIMain)
 - nastavení COM portu - třída ConnectSetting (instance ve třídě MDIMain)
 - příjem a dekodování rámců -
metoda void serialPort1_DataReceived(...) (třída MDIMain)
 - metody pro sestavování a dekodování rámců

- (statická třída `CommunicationFrames`)
 - odeslání rámců implementováno ve form jednotlivých oken
 - kontrolní součet rámců - CRC16-CCITT (statická třída `CRC16_CCITT`)
- vytváření profilu výrobního cyklu
 - programové řešení a okno ve třídě `Form1`
 - možnost uložení a načtení z XML souboru (třídy `XMLworker` a `ProductionCycleSaveData`, interface `IWorker`)
- vizualizace událostí
 - errorry aplikace - dialogová okna (systémová třída `MessageBox`)
 - procesní události - každé okno (třída `FormProcessEventInfoWindow`) běží v samostatném vlákne
 - záznam jednoho vlákna s oknem - instance třídy `ProcessEventInfoWindowOne`
 - správa oken - třída `ProcessEventInfoWindows` (instance ve třídě `MDIMain`), obsahuje list položek `ProcessEventInfoWindowOne`
- správa historie procesních událostí
 - jedna konkrétní událost - instance třídy `ProcessEvent`
 - správa událostí, obsahuje list událostí - třída `ProcessEvents` (instance ve třídě `MDIMain`)
 - zobrazení - třída `FormProcessEvents`
- statistické informace procesu
 - jedna konkrétní statistika - instance třídy `OneStatistics`
 - správa statistik, obsahuje list jednotlivých statistik - třída `Statistics` (instance ve třídě `MDIMain`)
 - zobrazení - třída `FormStatistics`
- nastavení CPU dat
 - konkrétní položka - instance třídy `OneCPUData`
 - správa CPU dat - třída `CPUData` (instance ve třídě `MDIMain`)
 - zobrazení - třída `FormControlCPUDataSettings`
- logování procesních událostí, statistiky a dějů aplikace
 - do souboru - metoda `static void LogMessageToFile(...)` (statická třída `LogMessages`)

7.2 Programové řešení

Spouštění aplikace

Při spouštění aplikace je volána metoda `Main` ve statické třídě `ApplicationRun`, kde nejprve proběhne prohledání běžících procesů systému. Zjistí se, jestli tato aplikace již neběží v některém procesu. Pokud ano, je tomuto procesu předáno řízení a GUI je vloženo do popředí a maximalizováno. V případě, že daný proces neexistuje, je inicializována nová aplikace a v tomto procesu spuštěna nová instance hlavního okna `MDIMain`.

Toto řešení je implementováno z důvodu, aby obsluha nespouštěla aplikaci nesmyslně vícekrát, neboť na jedno PC je vázáno jedno CPU řídicího modulu. Spuštění by mohlo mít za následek např. znehodnocení informací v log souboru, který je vytvářen pouze jednou za kalendářní den do konkrétního adresáře a děje aplikace spojené s jeho startem jsou do něho zaznamenávány. Z hlediska bezpečnosti to nemá vliv.

Ve třídě `MDIMain` jsou mimo vlastních grafických objektů okna definovány následující vybrané proměnné a metody:

- třídy datových zdrojů (alokovány v konstruktoru) pro okna (viz výše)
- události typu `AutoResetEvent` pro synchronizaci mezi vlákny při obsluze transakcí komunikačního protokolu (alokovány v konstruktoru)
- události typu vlastního `delegate` pro dálkový refresh oken
- proměnné a buffery pro obsluhu komunikace
- okno typu `Form1` pro zobrazení programu výrobního cyklu
- sériový port
- metoda příjmu ze sériového portu `void serialPort1_DataReceived(...)`

Formát logu

Soubor logu je vytvářen do cílového adresáře se jménem ve tvaru nesoucím informaci od datu vzniku: `\\logy\log_9_4_2012.txt`. Prováděn je záznam hlavně chyb aplikace a komunikace, dále procesní chyby. Před ukončením aplikace je rovněž zapsána statistika.

Na příkladu obsahu log souboru (viz níže) je patrné, že někdy v čase 15:04:27 až 15:05:03 byla aplikace úspěšně připojena k CPU řídicího modulu, aby mohla být následně obdržena zpráva o procesní události typu porucha.

Příklad logu:

```
logged: 9.4.2012 14:59:25: Aplikace spuštěna!.
logged: 9.4.2012 14:59:30: Chyba PC: connect_to_automat() - K automatu se nepodařilo připojit!.
```

```

logged: 9.4.2012 14:59:37: Chyba PC: set_counting(Form1) - Program není připojen k zařízení!.
logged: 9.4.2012 14:59:38: Otevřen program: C:\Documents and Settings\Administrator\Plocha\data555.xml.
logged: 9.4.2012 15:00:05: Chyba PC: butSendprogram_Click(Form1) - Program není připojen k automatu!.
logged: 9.4.2012 15:04:11: Chyba PC: butSendprogram_Click(Form1) - Program není připojen k automatu!.
logged: 9.4.2012 15:04:13: Chyba PC: butSendprogram_Click(Form1) - Program není připojen k automatu!.
logged: 9.4.2012 15:04:15: Chyba PC: butSendprogram_Click(Form1) - Program není připojen k automatu!.
logged: 9.4.2012 15:04:25: Chyba PC: butSendStep_Click(Form1) - Program není připojen k automatu!.
logged: 9.4.2012 15:04:27: Chyba PC: butSendprogram_Click(Form1) - Program není připojen k automatu!.
logged: 9.4.2012 15:05:03: Proces: Porucha Automat_cas: 29.3.2012 22:50:06
                          Kód:7 Kritická porucha WatchDog relé.
logged: 9.4.2012 15:08:32: Statistika: Počet cyklů = 16843009 cyklů.
logged: 9.4.2012 15:08:32: Statistika: Doba provozu = 9357,23 h.
logged: 9.4.2012 15:08:32: Statistika: Tlak = 254,6 bar.
logged: 9.4.2012 15:08:32: Uzavřen program: C:\Documents and Settings\Administrator\Plocha\data555.xml.
logged: 9.4.2012 15:08:32: Aplikace ukončena!.

```

Odeslání programu výrobního cyklu

Blokové schéma implementace (viz Obr. 7.1) popisuje, že transakce odesílání programu do CPU řídicího automatu vyžaduje použití dvou metod. V tomto případě je transakce iniciována na stisknutí tlačítka „Odeslat program“ v rámci třídy (okna) `Form1` metodou `void butSendprogram_Click(...)` (viz Obr. 7.1 vpravo nahoře). Druhou metodou je obsluha přijetí dat ze sériového portu `void serialPort1_DataReceived(...)` (viz Obr. 7.1 dole vlevo).

Důležitou roli v první zmíněné metodě hraje mutex `ACKMutex`, který zajišťuje exkluzivní přístup k odesílání po sériové lince, tudíž transakce nemůže být nikde jinde v programu vyvolána, např. z jiného vlákna. Chrání kritickou sekci, ve které se čeká na obdržení události `AutoEventACK`. Druhý mutex `TNMutex` je určen pro exkluzivní přístup k hodnotě čísla transakce (TN) odesílaného rámce a k události `AutoEventACK`.

V kritické sekci `TNMutex` je proveden reset události `AutoEventACK` a uložení hodnoty TN odesílaného rámce. V metodě obsluhy příjmu dat je v případě úspěšného přijetí `ACK` rámce nutné také vstoupit do kritické sekce `TNMutex`, kde se vyhodnotí rovnost odeslaného TN a TN přijatého v `ACK` rámci. V případě souhlasu se nasetuje událost `AutoEventACK`. Následuje uvolnění `TNMutex` kritické sekce.

Implementací této kritické sekce je tak zabráněno situaci, kdy by byla sice resetována `AutoEventACK` a uloženo odeslané TN , ale mezitím by se nahodila `AutoEventACK` nějakým pozdě přijatým `ACK` rámcem, neboť ověření správnosti přijatého TN v `ACK` rámci proběhlo před uložením nového odeslaného TN . Takto bude TN pozdě přijatého rámce různé od aktuálně očekávaného TN a událost nebude nasetována.

Jiná transakce protokolu, kde dochází k odesílání dat do CPU, má v jakékoli jiné metodě obdobnou implementaci. Dokonce jiné metody sdílejí stejné pomocné pro-

měnné a buffery pro skládání rámců ve třídě `CommunicationFrames`) a parsování přijatého rámce ve třídě `MDIMain`).

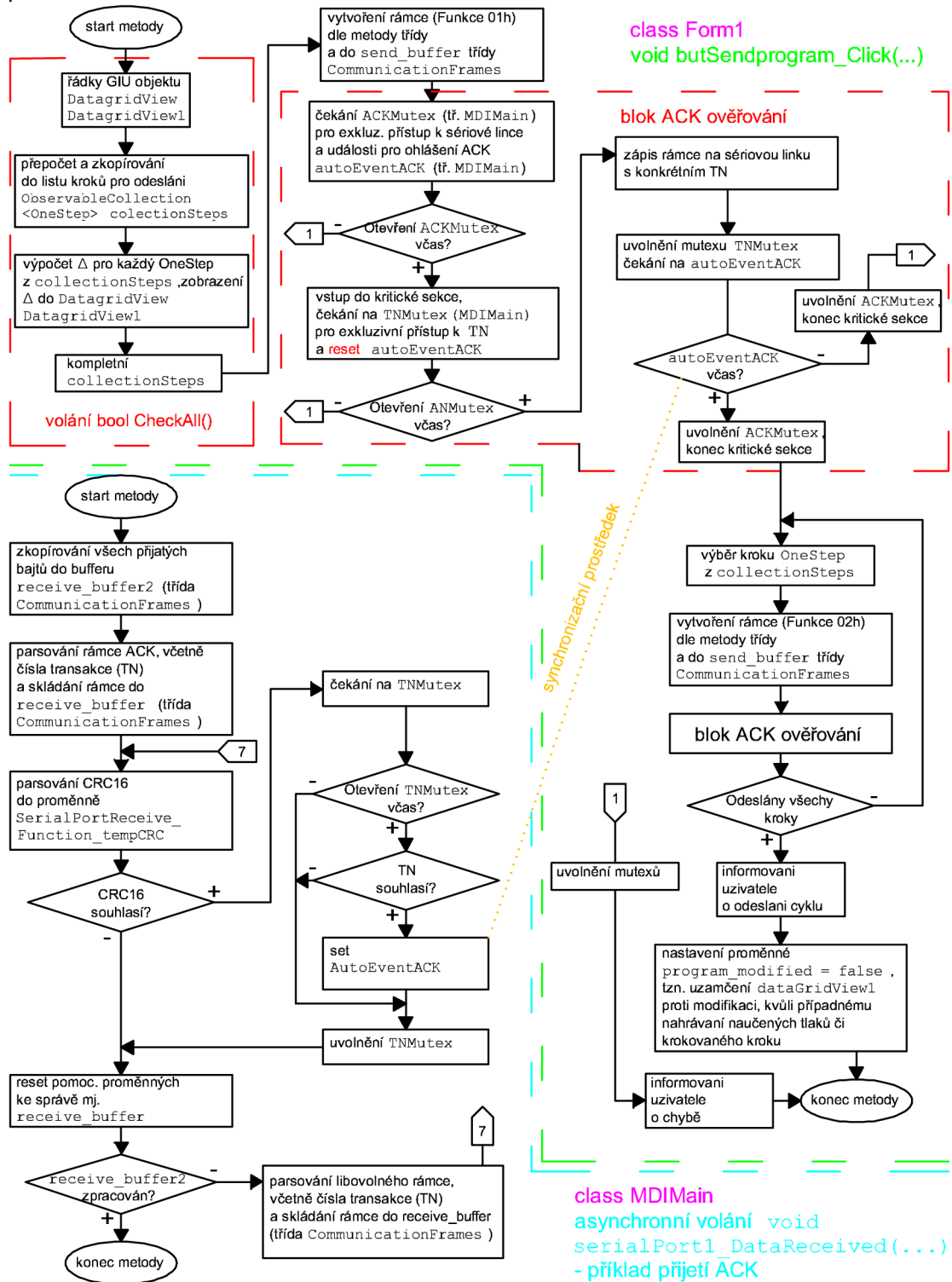
V asynchronně volané obslužné metodě příjmu dat je libovolný počet přijatých bajtů nakopírován do bufferu `receive_buffer2`, z něhož je v cyklu parsován rámeček a kompletován do bufferu `receive_buffer`. Je proveden kontrolní součet rámce a v tomto okamžiku poté navazují další kroky dle funkce rámce, např. výše popsaný ACK rámeček vyvolá událost `AutoEventACK`.

Pokud metoda, která vyvolala transakci, neobdrží událost `AutoEventACK` do doby `time-out`, je uživatel informován o neúspěchu transakce.

Příjem dat

Při transakci typu `CPU_STARTED_TRANSACTION` obdrží PC aplikace asynchronně data od CPU řídicího automatu, přičemž následuje odeslání ACK rámce (obsahuje přijatou hodnotu TN). Následné kroky prováděné aplikací dle funkce protokolu přijatých dat jsou uvedeny v přehledu u blokového schématu příjetí dat (viz Obr. 7.2).

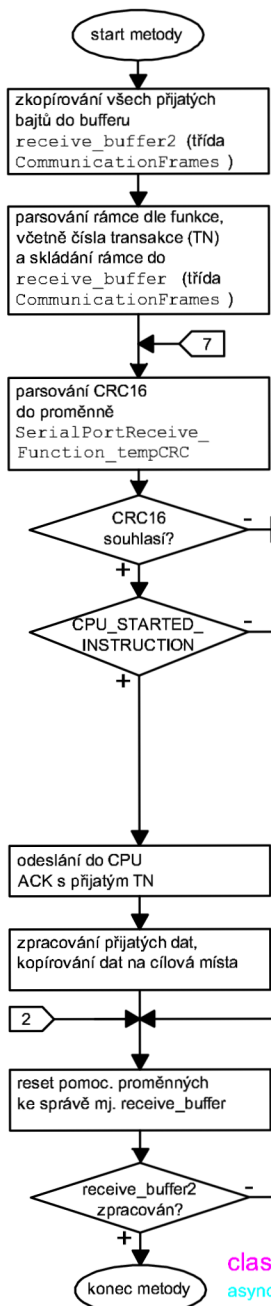
Pokud PC vysílá žádost o data, jedná se o `PC_STARTED_TRANSACTION`, po obdržení dat aplikace ACK rámeček neodesílá. Zdroj žádosti o data (metoda) má ve třídě `MDIMain` definovanou konkrétní událost typu `AutoResetEvent`, která jej informuje, že byly data přijaty. Uživatel je informován, pokud data nedorazí včas.



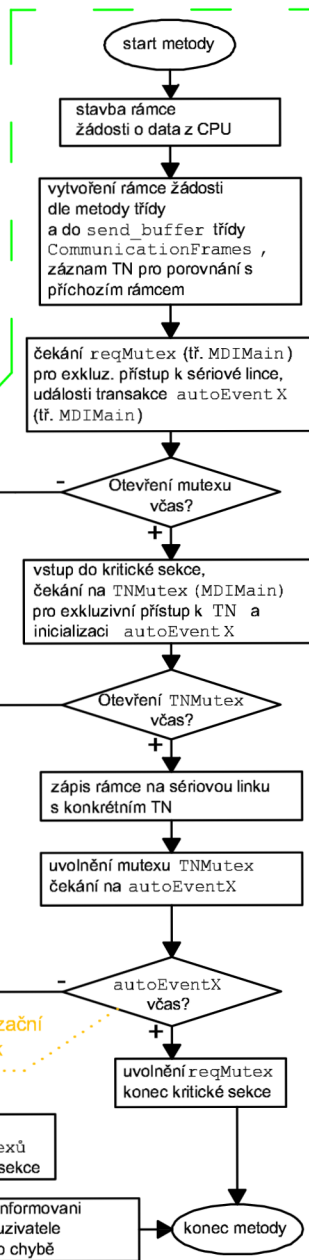
Obr. 7.1: Blokové schéma odeslání dat (výrobního cyklu)

Zpracování příchozích dat a s tím spojené procedury, použito ve funkcích protokolu ve tř. aplikace

- 04h (PC_STARTED) - autoEventStatistics
- volání ve tř. FormStatistics
- refresh okna FormStatistics
- refresh dat tř. Statistics (instance v MDIMain)
- 04h (CPU_STARTED) (jako předchozí)
- 05h (CPU_STARTED)
- refresh okna FormProcessEvents
- refresh dat tř. ProcessEvents (instance v MDIMain)
- refresh dat tř. ProcessEventInfoWindows (instance v MDIMain)
- vyskočení nového okna typu FormProcessEventInfoWindow
- 06h (CPU_STARTED)
- zápis dat do objektu a refresh objektu dataGridView1 (Form1)
- 07h (CPU_STARTED)
- označení řádku v objektu dataGridView1 (Form1)
- 08h,0Ah (PC_STARTED) - autoEventCPUdata
- volání z tř. FormControlCPUDataSettings
- obnova dat tř. CPUdata (instance v MDIMain)
- refresh okna FormControlCPUDataSettings



class MDIMain
asynchronní volání void serialPort1_DataReceived(...)
- příklad: přijetí dat z CPU řídicího modulu



class neurčena (viz vysvětlivky výše)
metoda iniciující PC_STARTED_TRANSACTION

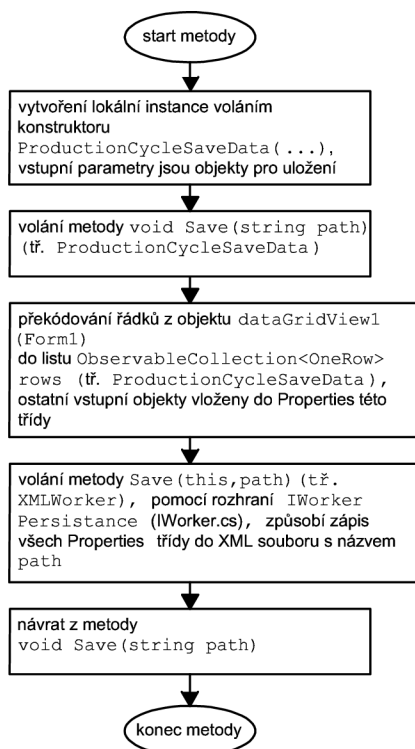
Obr. 7.2: Blokové schéma příjmu dat

Ukládání programu výrobního cyklu do souboru

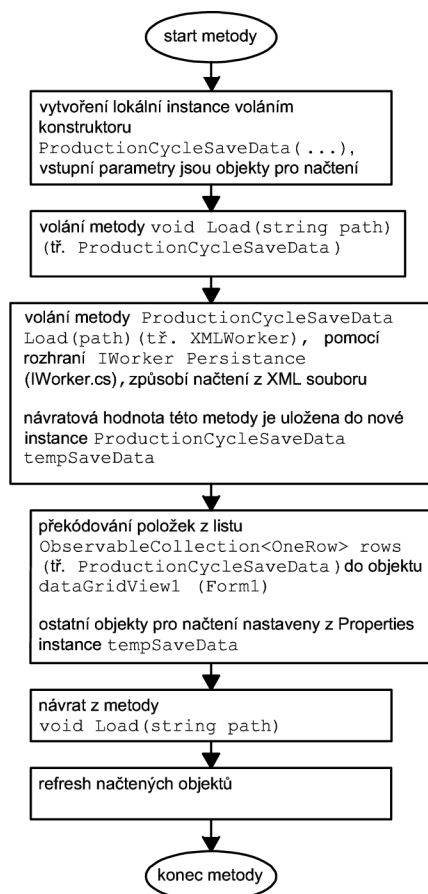
Postup ukládání programu do XML souboru, respektive načítání ze souboru představuje blokové schéma (viz Obr. 7.3). Operace zajišťují metody třídy Form1 prostřednictvím instance třídy ProductionCycleSaveData. Data načítaná nebo pro uložení jsou předávána přes konstruktor ProductionCycleSaveData a musí být uložena do Properties této třídy ProductionCycleSaveData.

Takže např. zobrazovací objekt dataGridView1 (Form1) je vevnitř třídy ProductionCycleSaveData překódován na Properties ObservableCollection<OneRow> Rows.

```
class Form1
void save_to_file(string path)
```



```
class Form1
bool load_from_file(string path)
```



Obr. 7.3: Blokové schéma uložení programu do XML souboru a načtení programu

Uvnitř třídy ProductionCycleSaveData je proměnná typu rozhraní IWorker Persistence = new XMLWorker(), prostřednictvím kterého jsou volány metody

pro uložení a načtení ze třídy `XMLWorker`. Rozhraní je použito z důvodu abstrakce, třídu v rozhraní (a tím i implementaci metod) lze nahradit jinou a data ukládat např. na úplně jiné médium.

Metody třídy `XMLWorker` obsahují volání metod systémové třídy `XMLSerializer`, ty provádějí provádí samotnou manipulaci s XML souborem tak, že zpracovávají `Properties` třídy `ProductionCycleSaveData`.

7.3 Grafické rozhraní (GUI)

Hlavní okno (MDIMain)

Aplikace je spuštěna do hlavního maximalizovaného okna „Automat Control“ (`MDIMain`), který disponuje základním menu o čtyřech rozšiřovacích nabídkách. V hlavním okně jsou zobrazovány okna programu výrobního cyklu (`Form1`), jak je patrné na Obr. 7.5. Jednotlivé položky menu (viz Obr. 7.4) mají následující význam:

Nový Vytvoří a zobrazí nové okno programu výrobního cyklu čili novou instanci formuláře (`Form1`)

Otevřít Dialogové okno „Otevřít“ pro výběr souboru *.xml, po návratu z něj vytvoří a zobrazí nové okno programu výrobního cyklu s daty načtenými z *.xml

Uložit Dialogové okno „Uložit jako“ pro výběr souboru *.xml (pokud již program nebyl uložen), uloží data programu do *.xml

Uložit jako Dialogové okno „Uložit jako“ pro výběr souboru *.xml , uloží data programu do specifikovaného *.xml

Zavřít Dialogové okno s dotazem o uložení, zavření okna programu

Exit Dialogové okno s dotazem o uložení programu (pokud je otevřen), uzavření celé aplikace

Připojit Explicitní provedení transakce „echo“ dle funkce 00h protokolu, děje se totiž implicitně při startu aplikace

Nastavení Vyvolání okna (`FormConnectSetting`) pro nastavení COM portu (viz 7.6)

Statistika Vyvolání okna (`FormStatistics`) pro zobrazení vybraných statistických hodnot (viz 7.7)

Procesní události Vyvolání okna (`FormProcessEvents`) obsahující historii přijatých procesních událostí(viz 7.8)

Nastavení automatu Vyvolání okna (`FormProcessEvents`) obsahující historii přijatých procesních událostí(viz 7.8)

Dokumentace Otevření souboru dokumentace (rezerva)

O aplikaci ... Vyvolání okna (`AboutBox1`) s informací o verzi aplikace(viz 7.8).



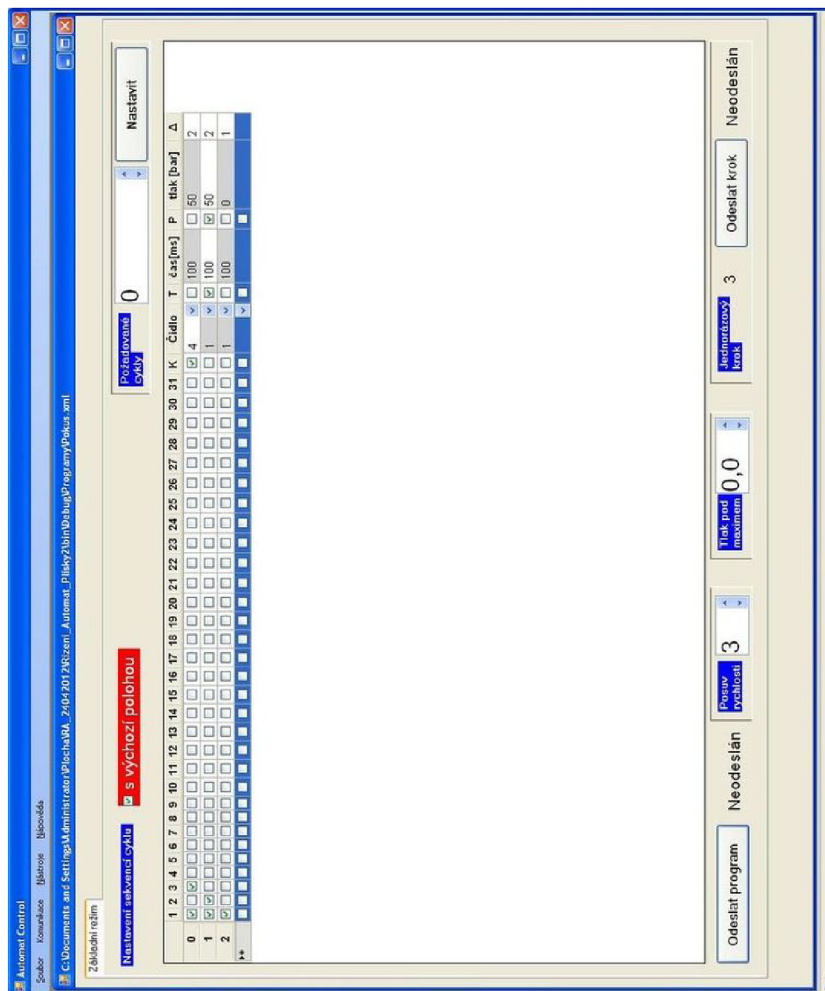
Obr. 7.4: Jednotlivé položky menu aplikace (MDIMain)

Okno programu výrobního cyklu (Form1)

Po startu okna je implicitně provedena transakce čtení statistiky (funkce 04h protokolu) z důvodu inicializace počítačdra „Požadované cykly“ na hodnotě 0. Je tady zjištěn výchozí absolutní počet provedených výrobních cyklů stroje ($APVC$). Po realizování jednoho výrobního cyklu CPU řídicího modulu provede transakci odeslání statistiky (funkce 04h) a aplikace tak získá nový absolutní počet výrobních cyklů ($NAPVC$). Na počítadlo je přičten rozdíl $NAPVC - APVC$, přičemž hodnota $APVC = NAPVC$ je výchozí pro výpočet po příštím ukončení cyklu. Tato metoda zajistí, že pokud se po ukončení jednoho cyklu nepodaří odeslat data, po ukončení dalšího cyklu se přičte dvojka.

Nejdůležitějším prvkem tohoto okna je objekt `dataGridView1`, do kterého jsou zadávány jednotlivé kroky výrobního cyklu. Dále je možno měnit pomocí kursoru pořadí řádků přemístěním a mazat řádky klávesou Delete. Každá taková operace je ukončena přečíslováním řádků. Uvedené operace nejsou standardní pro tento objekt a byly implementovány s použitím dostupných metod třídy objektu. Stav digitálního výstupu karty automatu je nastaven pomocí zatržítka ve sloupcích „1“ - „31“. Pomocí zatržítok „K“, „T“ nebo „P“ se aktivuje způsob ukončení kroku a zároveň jsou povolena příslušná políčka, kde se zadává odpovídající číselná hodnota. Uživatel je upozorněn v případě zadávání nevhodné číselné hodnoty.

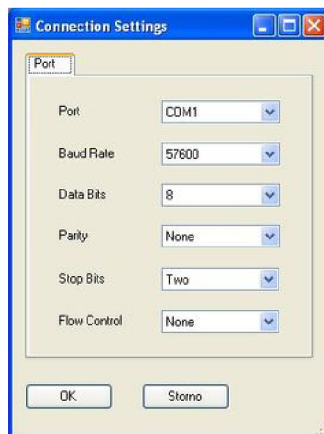
Tlačítkem „Odeslat program“ je provedena sekvence příkazů (viz Obr. 7.1 výše), která provede překontrolování výrobního cyklu, výpočet a zobrazení hodnoty změn výstupů Δ do určeného sloupce objektu `dataGridView1` a samotné odeslání. Úspěšné odeslání programu výrobního cyklu je potvrzeno oknem se zprávou, přičemž text vedle tlačítka „Neodeslán“ je obměněn za „Odeslán“. Odeslání manuálního neboli jednorázového kroku je provedeno tlačítkem „Odeslat krok“, přičemž je odeslán kursorem označený krok v `dataGridView1`, jehož index je zobrazen bezprostředně vedle tlačítka. Úspěšné odeslání je indikováno stejně jako v případě odeslání výrobního cyklu.



Obr. 7.5: Hlavní okno aplikace (MDIMain)

Po odeslání programu se uzamkne objekt `dataGridView1` pro přístup pomocí kursoru, při pokusu o vstup do nějaké buňky je zobrazeno dialogové okno „Modifikace programu způsobí rozdílnost mezi PC a automatem. Opravdu chcete pokračovat? Ano/Ne“. Zvolením možnosti „Ne“ je zaručena totožnost podoby výrobního cyklu v aplikaci i v CPU řídicího modulu a data jako index kroku při krokování či hodnoty zjištěným maxim tlaku mohou být promítnuty do `dataGridView1`. Naučené hodnoty tlaku jsou zapsány do sloupce „Tlak“. Aktuální krok v krokování je presentován jako označený řádek v `dataGridView1`.

Naopak odsouhlasení vstupu způsobí odemknutí objektu `dataGridView1` pro volný přístup. Výše popsaná data od CPU řídicího automatu jsou přijata, ale nijak zpracována. Tím jsou v podstatě vyčerpány funkce tohoto okna, při jeho zavírání je uživatel tázán na uložení programu.



Obr. 7.6: Okno nastavení COM portu (FormConnectSetting)

Okno nastavení COM portu (FormConnectSetting)

Při startu aplikace je uložené nastavení COM portu načteno do instance třídy `ConnectSetting` (v `MDIMain`) ze souboru `\\set\\serial_port.b`. Datová instance `ConnectSetting` je předána při volání tohoto okna, přičemž objekty okna jsou dle ní nastaveny. Obr. 7.6 ukazuje přesné nastavení pro komunikaci s CPU řídicího automatu. Stisknutím tlačítka „OK“ je provedeno uložení změn do souboru a obnoveno připojení k CPU řídicího automatu (echo). Tlačítko „Storno“ eliminuje změny a provede připojení se stávající konfigurací. V obou případech následuje uzavření okna.

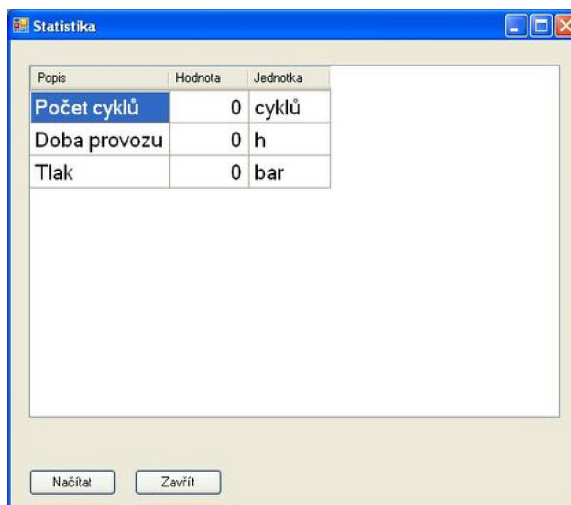
Okno statistiky (FormStatistics)

Jednotlivé kategorie statistiky jsou uloženy v listu `ObservableCollection<OneStatistics>` v rámci instance třídy `Statistics` (v `MDIMain`), která je předávána při volání okna konstruktorem. Zobrazovací objekt `dataGridViewStat` je pomocí prostředku „binding“ (.NET4.0) vázán na zmíněný list. Jedná se o velice rychlou formu implementace propojení objektu na datový zdroj.

Stiskem tlačítka „Načítat“ je aktivován časovač, jehož obslužná rutina v periodě 3 s provádí čtení statistiky z CPU řídicího modulu (funkce 04h protokolu). Přijatá data jsou v asynchronně volané metodě pro příjem ze sériové linky (v `MDIMain`) uložena do listu v rámci `Statistics`. Jelikož obsluha příjmu běží na jiném vlákne než objekty okna, je implementováno ošetření pro manipulaci s `dataGridViewStat` z jiného vlákna. Tlačítko „Zavřít“ zastaví expirace časovače a zavře okno.

Číselná hodnota kategorie statistiky má svůj fyzikální rozměr, který ve většině případů neodpovídá reprezentaci v CPU řídicího modulu. Proto mohou být přijatá

„hrubá“ data vynásobena koeficientem, popř. posunuta nějakým offsetem. Oba parametry jsou uloženy v polích o délce počtu kategorií v instanci třídy `Statistics`.



Obr. 7.7: Okno zobrazení převážně statistických hodnot (FormStatistics)

Okno historie procesních událostí (FormProcessEvents)

Zobrazovací objekt `dataGridViewProcessEvents` je vázán obdobně jako v případě předchozím pomocí „binding“ k listu `ObservableCollection<ProcessEvent>` `processEventsList` v instanci `ProcessEvents` (v `MDIMain`).

Zprávy o procesních událostech přichází asynchronně z CPU řídicího modulu (funkce 05h protokolu) a příjem dat a refresh objektů je implementován obdobně jako u předchozího okna. Data z CPU obsahují pouze kód, který odpovídá indexu v poli s vysvětlujícím textem ve třídě `ProcessEvents`, přičemž přiřazení realizuje metoda této třídy. Ve sloupci „Automat_cas“ je zobrazen čas v CPU, aby byla při analýze zřetelná posloupnost vzniku událostí.

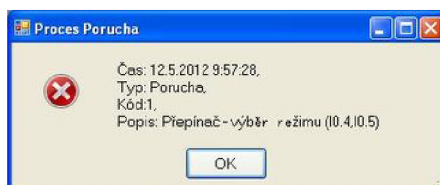
PC_cas	Automat_cas	Typ	Kod	Popis
12.5.2012 9:57:25	10.5.2012 9:55:29	Oznámení	0	Kvitace provedena
12.5.2012 9:57:28	10.5.2012 9:55:29	Porucha	1	Přepínač - výběr režimu (10.4,10.5)

Obr. 7.8: Historie procesních událostí (FormProcessEvents)

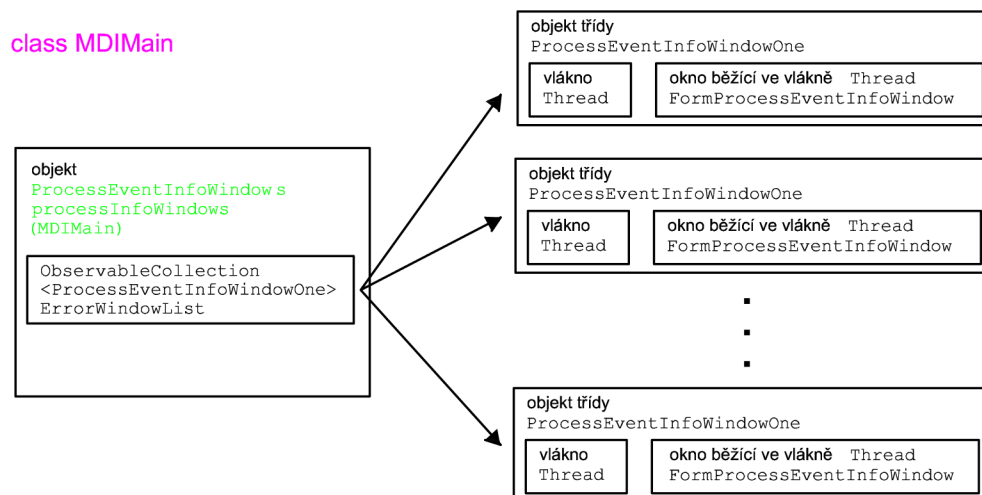
Okno procesní události (FormProcessEventInfoWindow)

V okamžiku přijetí zprávy o procesní události je vytvořeno toto speciální „vyskakovací“ okno (Obr. 7.9), přičemž jak popisuje Obr. 7.10, každé okno běží v svém samostatném vlákne, aby nedošlo k zablokování činnosti aplikace. Okno je ve vlákne zobrazeno jako „modal“, čili přerušuje běh vlákna. Obr. 7.10 dále ukazuje jak jsou vlákna a okna organizována.

Při zavírání celé aplikace musí být vlákna ukončena, tudíž je na okna zavolána metoda `close` třídy `FormProcessEventInfoWindow`, která uvolní běh vlákna. Poté probíhá čekání na ukončení vlákna a teprve v tomto okamžiku je aplikace ukončena (pozn. při vypínání se vykonávají i jiné procedury).



Obr. 7.9: „Vyskakovací“ okno procesní události (FormProcessEventInfoWindow)



Obr. 7.10: Vizualizace procesních událostí

Okno nastavení CPU dat (FormControlCPUDDataSettings)

Toto okno není obsluhuje přístupné, lze ho vyvolat pouze zadáním správného hesla do předchozího okna FormLogin. Slouží pro nastavení a testování CPU řídicího modulu. Okno obsahuje celkem dva zobrazovací objekty typu DataGridView:

- dataGridViewData (Nastavení CPU času)
- dataGridViewTime (Nastavení vybraných dat)

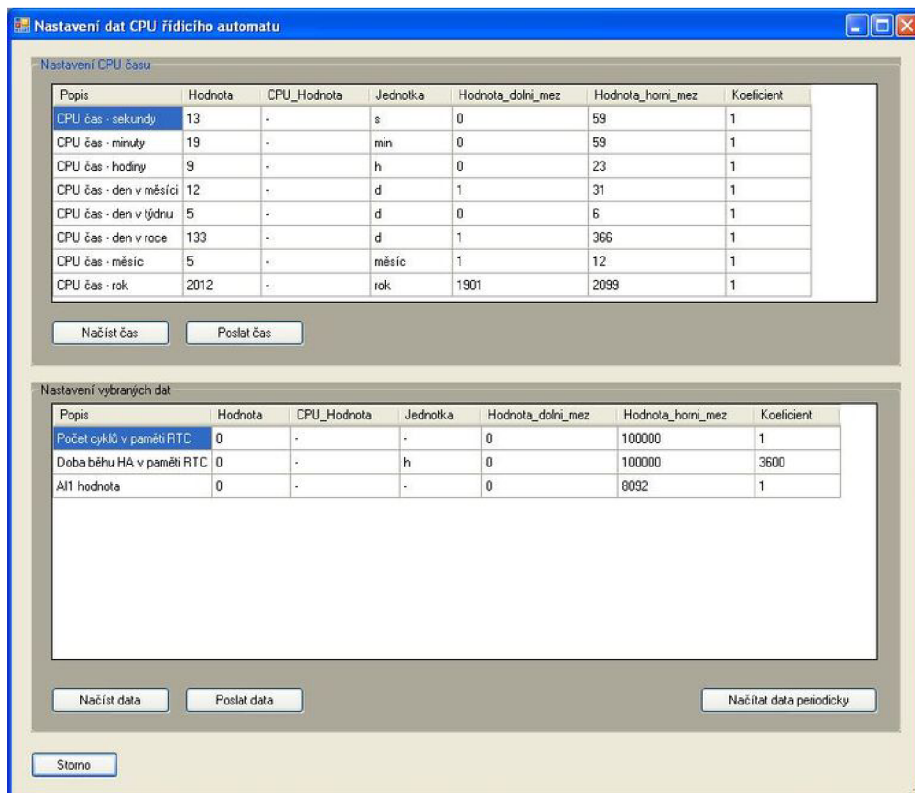
kteřé jsou vázány na příslušné listy v rámci instance třídy CPUdata (v MDIMain):

- ObservableCollection<OneCPUData> CPUtimeList
- ObservableCollection<OneCPUData> otherCPUdataList

Položka dat specifikovaná třídou OneCPUData obsahuje vlastnosti (Properties) odpovídající sloupcům objektů typu DataGridView na Obr. 7.11. Listy jsou vytvořeny metodou třídy CPUdata z nadefinovaných polí.

Řešení příjmu dat (funkce 08h, 0Ah protokolu) a refresh objektů je implementován stejně jako v případě statistiky (viz 7.3, str. 83). Odesílání dat (funkce 09h, 0Bh protokolu) je obdobné funkci 01h protokolu v sekvenci odesílání výrobního cyklu (Obr. 7.1, str. 77).

Na Obr. 7.11 je patrné, kromě zřetelného použití pro nastavení času RTC obvodu CPU, také využití pro přenastavení hodnot proměnných v zálohované paměti CPU. Dále je možnost číst z CPU řídicího modulu číselnou hodnotu kanálu ADC převodníku, na který je připojen analogový vstup „AI1.“



Obr. 7.11: Okno nastavení CPU dat (FormControlCPUDataSettings)

Okno „O aplikaci...“ (AboutBox1)



Obr. 7.12: Informace o aplikaci (AboutBox1)

8 ZÁVĚR

V rámci vypracované diplomové práce byl pro tvářecí stroj realizován řídicí automat, pro který bylo předem dané začlenění v rámci řídicího rozvaděče. Rovněž byly předloženy I/O moduly a vana, které měly být použity. Byl vytvořen řídicí modul pro zasunutí do vany, který ovládá skrze paralelní sběrnici I/O moduly.

Řídicí modul byl koncipován pro dlouhodobý provoz, čemuž byla přizpůsobena kvalita použitých součástí. Řídicí modul byl osazen WatchDog budičem relé zapojených v bezpečnostním obvodu CentralSTOP, tento budič je resetován v případě interní chyby CPU, která vede na přetečení interního WatchDog časovače.

Rovněž byla navržena galvanicky oddělená 8bitová výstupní karta napájená 24 VDC, ve které byly provedeny opatření proti hazardním stavům, např. při podpětí napájecího napětí karty.

Firmware CPU řídicího modulu je tvořen operačním systémem CoCoX CoOS, na kterém běží na nejvyšší prioritě řídicí algoritmus stroje. Procesní událost (např. porucha) hlásí CPU pomocí vlastního protokolu na RS-232 do PC. Uživatel je informován o ději v procesu. Ze záložních důvodů je řídicí modul vybaven SD kartou, kam probíhá zápis logu.

Navržený protokol je koncipován tak, aby byla data bezpečně doručena. Každá transakce komunikace je číslována tak, aby komunikující stanice zpracovávaly odpovídající si transakci.

Navržená aplikace obsahuje GUI pro pohodlné nastavování výrobního cyklu, jeho uložení, dále správu příchozích procesních událostí, správu vybraných statistických informací o výrobě aj. Především je zde implementován protokol komunikace. Veškeré děje v aplikaci a příchozí procesní události jsou zapisovány neprodleně do souboru logu na disk.

Samotná podoba výrobního cyklu byla zadána pouze rámcově a v rámci této práce byla optimalizována. Hlavně možnost ukončení kroku po přesahu určité hodnoty tlaku zrychlila výrobní cyklus jako takový a snížila náklady na oživení výrobního cyklu pro nově nasazený výrobek. Nutnou podmínkou je schopnost zařízení nalézt maximum tlaku a zmíněný vypínací práh spočítat.

LITERATURA

- [1] ZÁHLAVA, CSc., Ing. Vít. *Návrh a konstrukce desek plošných spojů*. Vydání první. Praha : Česká technika, prosinec 2005. 75 s. ISBN 80-01-03351-1
- [2] Hardware Manual for the PCD3 Series [online]. E8. Murten / Switzerland : Saia-Burgess Controls AG, 1.11.2007 [cit. 2011-11-20]. Dostupné z WWW: <http://saia.tecnosistemas.es/pdf/productos/26-789_E8_PCD3.pdf>.
- [3] *Pandatron.cz - Elektrotechnický magazín: Úvod do architektury Cortex-M3 - díl. 1* [online]. [cit. 2012-04-11]. ISSN 1803-6007. Dostupné z: <http://pandatron.cz/?1252&uvod_do_architektury_cortex-m3_-_dil__1>.
- [4] *CooCox CoOS User's Guide* [online]. Revision 1.1: www.coocox.org, August, 2009, www.coocox.org [cit. 2012-04-11]. Dostupné z: <http://www.coocox.org/downloadfile/CoOS/PDF/CooCox_CoOS_User_Guide.pdf>.
- [5] ON SEMICONDUCTOR. *Universal Voltage Monitors: MC34161, MC33161, NCV33161* [online]. Rev. 10. July, 2007 [cit. 2012-04-11]. Dostupné z: <<http://www.farnell.com/datasheets/89953.pdf>>.
- [6] TEXAS INSTRUMENTS. *MAX3221* [online]. Rev. . March, 2004 [cit. 2012-04-11]. Dostupné z: <<http://www.farnell.com/datasheets/89890.pdf>>.
- [7] LINEAR TECHNOLOGY. *LTC1860/LTC1861* [online]. [cit. 2012-04-11]. Dostupné z: <<http://www.farnell.com/datasheets/62263.pdf>>.
- [8] STMICROELECTRONICS. *AN2208 APPLICATION NOTE: Designing Industrial Applications with VN808/VN340SP High-side Drivers* [online]. Rev. 1. September, 2005 [cit. 2012-04-11]. Dostupné z: <http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/APPLICATION_NOTE/CD00063806.pdf>.
- [9] AXELSON, Jan. *Serial Port Complete: COM Ports, USB Virtual COM Ports, and Ports for Embedded Systems*. Second Edition. Madison: Lakeview Research LLC, 2007. ISBN 978-1931448-07-9
- [10] NAGEL, Brian. *Advantages of the Cortex-M3* [online]. 2008 [cit. 2012-04-11]. Dostupné z: <http://www.iqmagazineonline.com/IQ/IQ25/pdfs/IQ25_pgs27-32.pdf>.

- [11] NXP. *UM10360: LPC17xx User manual* [online]. Rev. 2. August, 2010 [cit. 2012-04-11]. Dostupné z:
<http://www.nxp.com/documents/user_manual/UM10360.pdf>.
- [12] SD Group. *SD Specifications Part 1 Physical Layer Simplified Specification* [online]. Rev. 2. May, 2010 [cit. 2012-04-11]. Dostupné z:
<https://www.sdcard.org/downloads/pls/simplified_specs/Part_1_Physical_Layer_Simplified_Specification_Ver_3.01_Final_100518.pdf>.
- [13] SHARP, Robin. *Principles of protocol design* [online]. Berlin : Springer-Verlag Berlin Heidelberg, 2008 [cit. 2012-04-11]. Dostupné z:
<http://books.google.com/books/about/Principles_of_Protocol_Design.html?id=-wbHkGf5pwC>.

SEZNAM PŘÍLOH

A Elektronická příloha	92
B Řídicí modul	93
B.1 Kompletní schéma zapojení	93
B.2 DPS	102
B.3 Seznam součástek	107
B.4 Fotodokumentace osazených desek	112
C 8bitová 24VDC DO karta	115
C.1 Schéma zapojení	115
C.2 DPS	119
C.3 Seznam součástek	123
D Sestava automatu	125
E Řídicí rozvaděč	127

A ELEKTRONICKÁ PŘÍLOHA

Přiložený disk obsahuje na kořenovém adresáři následující položky:

xvasko06_dp.pdf - elektronická verze diplomové práce

CoIDE-1.2.4.exe - nástroj pro vývoj firmware CPU řídicího modulu

\\Cox0S\ - projekt CoIDE-1.2.4 - firmware CPU řídicího modulu

\\Rizeni_Automat_Plisky\ - MS Visual Studio 2010 Solution - PC aplikace

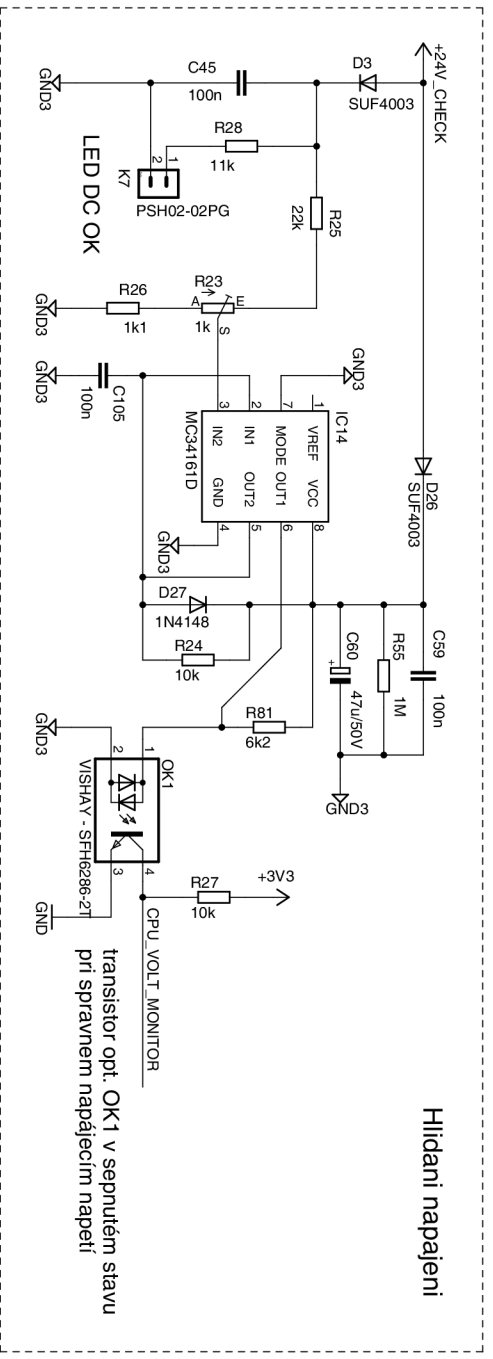
B ŘÍDICÍ MODUL

B.1 Kompletní schéma zapojení

Schéma je přehledně rozčleněno do několika funkčních celků na jednotlivých stranách, přičemž obvody jsou fyzicky realizovány na třech DPS, převážná část v rámci „Řídicí desky“.

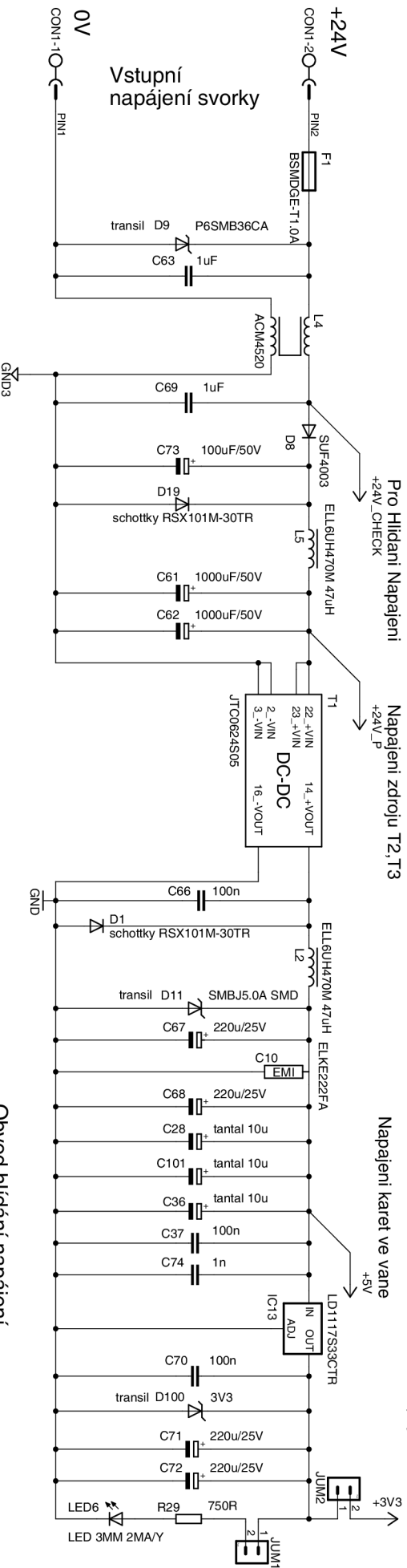
1. obvod hlídání napájení, obvod DC-DC zdroje T1 pro obvody CPU a napájení sběrnice vany
2. obvod DC-DC zdroje T2 pro obvody proudových analogových vstupů, obvod DC-DC zdroje T3 pro obvody komunikace RS232, RS485
3. zapojení pinů CPU, připojení paměti, rozhraní JTAG
4. obvod připojení CPU ke sběrnici vany, spínání napájení sběrnice vany, rezervní zapojení USB „Self-Power Device Configuration“
5. obvody galvanicky oddělených komunikací RS232, RS485
6. obvody dvou galvanicky oddělených analogových proudových vstupů (0-20 mA)
7. schéma přídatné DPS „Přídatná deska 1“, která obsahuje obvod transistorového výstupu pro buzení externího relé (obvod řídicího napětí)
8. schéma přídatné DPS „Přídatná deska 2“, obsahující obvod připojení SD karty pomocí softwarového SPI k CPU

Deska „Přídatná deska 2“ je zasunuta přes svůj konektor JUM6 do portu JUM3 desky „Řídicí deska“. Druhá doplňková deska „Přídatná deska 1“ je připojena pomocí vodičů z konektoru K6 na konektor K9 desky „Přídatná deska 2“.



Hlídání napájení

transistor opt. OK1 v sepnutém stavu při správnem napájecím napetí



Pro Hlídání Napájení

Napájení zdroju T2,T3

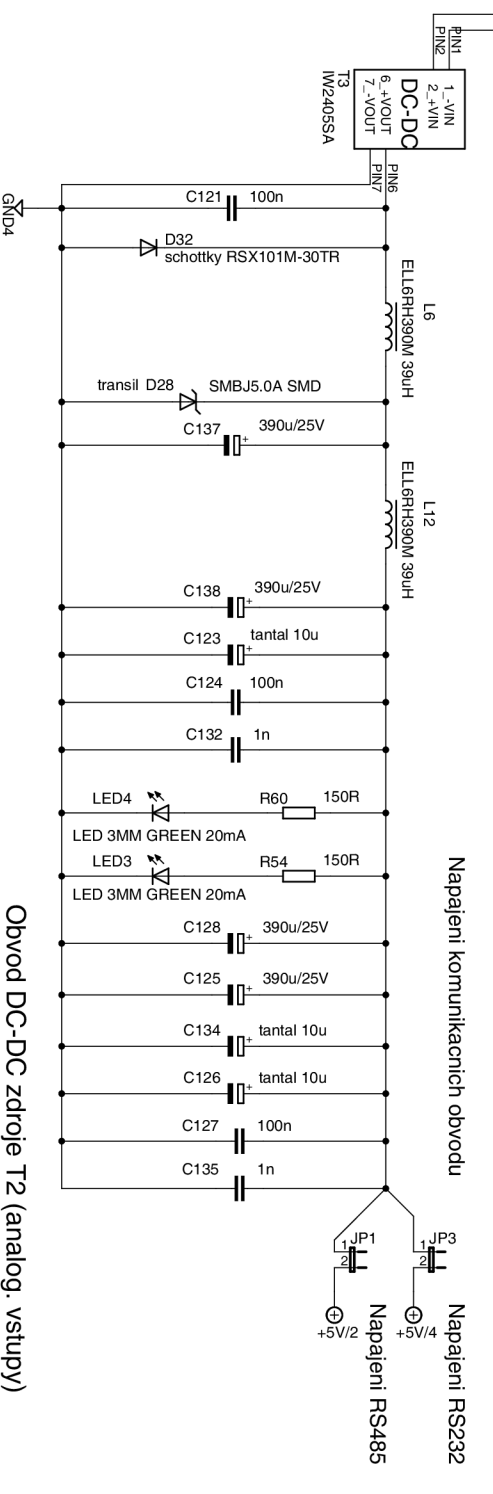
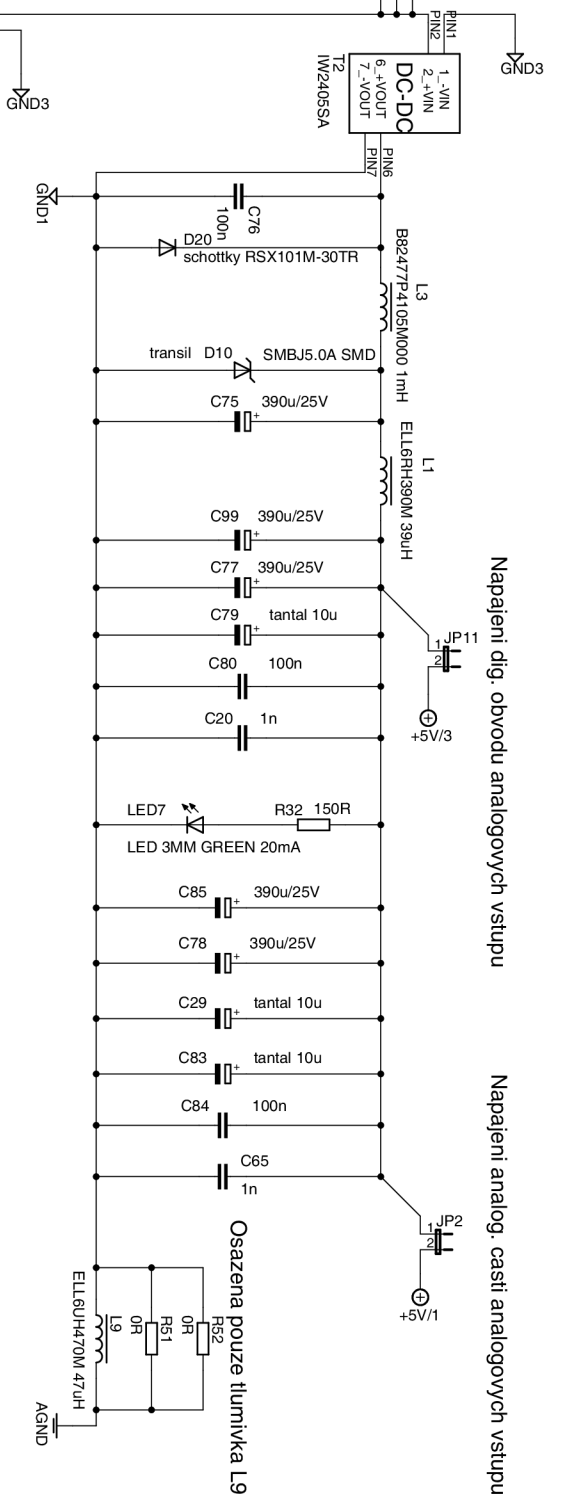
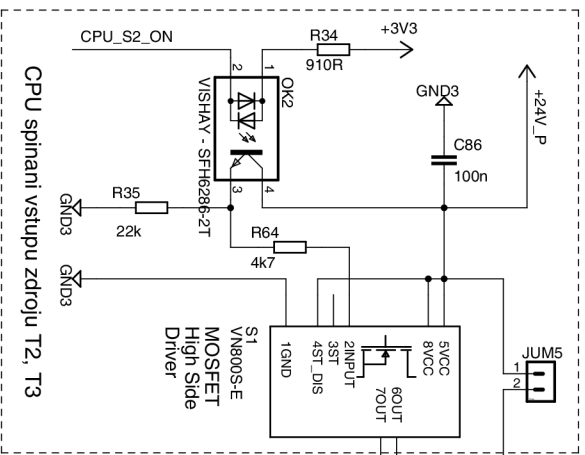
Napájení karet ve vane

Napájení obvodu CPU

Obvod hlídání napájení
Obvod DC-DC zdroje T1 (cpu, vana)

Ridici deska

RidiciDeska_d5
15.5.2012 9:53:49
Sheet: 1/8



Obvod DC-DC zdroje T2 (analog. vstupy)
Obvod DC-DC zdroje T3 (pro komunikacni cast)

Ridici deska

RidiciDeska_d5
15.5.2012 9:53:49
Sheet: 2/8

CPU ARM CORTEX M3

Vývod k Pridavne desce 2 (str. 8) ->

IC1 LPC1756

- CPU_RS232_TXD 37
- CPU_RS232_RXD 38
- CPU_LED2 79
- CPU_LED1 80
- CPU_ADC_CS 64
- CPU_ADC_SCK 63
- CPU_ADC_MISO 62
- CPU_ADC_MOSI 61
- CPU_SDA 39
- CPU_SCL 40
- CPU_RS485_TXD#7 48
- CPU_RS485_RXD#8 49
- CPU_DATA_DIR 46
- CPU_RES 45
- CPU_RS485_RTS#4 44
- CPU_S2_ON 7
- USB_D+ 22
- USB_D- 23

- P0[0]/RD1/TXD3/SDA1
- P0[1]/TD1/RXD3/SCL1
- P0[2]/TXD0/AD0[7]
- P0[3]/RXD0/AD0[6]
- P0[6]/I2SRX_SDA/SSEL1/MAT2[0]
- P0[7]/I2STX_CLK/SCK1/MAT2[1]
- P0[8]/I2STX_WS/MISO1/MAT2[2]
- P0[9]/I2STX_SDA/MOSI1/MAT2[3]
- P0[10]/TXD2/SDA2/MAT3[0]
- P0[11]/RXD2/SCL2/MAT3[1]
- P0[15]/TXD1/SCK0/SCK
- P0[16]/RXD1/SEL0/SSEL
- P0[17]/CTS1/MISO0/MISO
- P0[18]/DCD1/MOSI0/MOSI
- P0[22]/RTS1/TD1
- P0[25]/AD0[2]/I2SRX_SDA/TXD3
- P0[26]/AD0[3]/AOUT/RXD3
- P0[29]/USB_DPLUS
- P0[30]/USB_DMIN

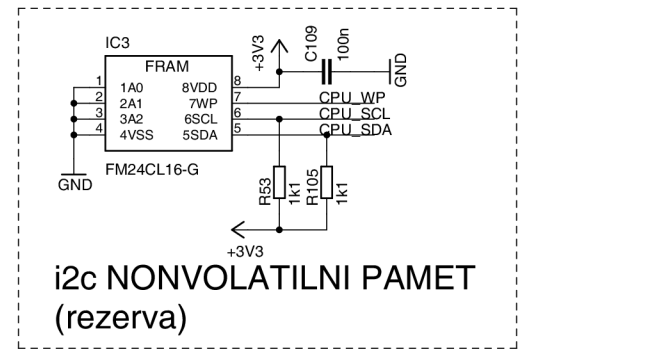
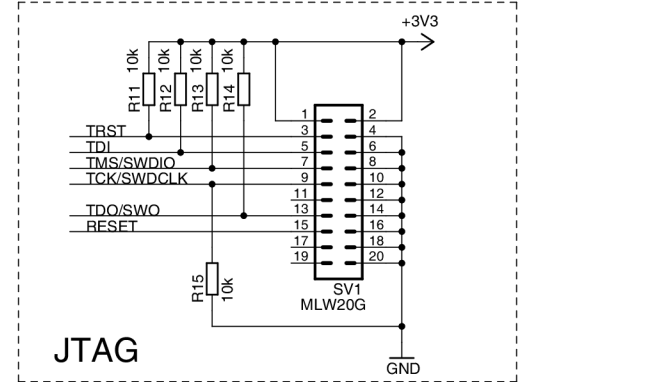
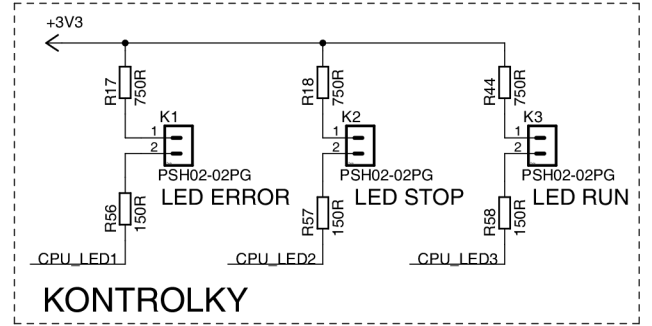
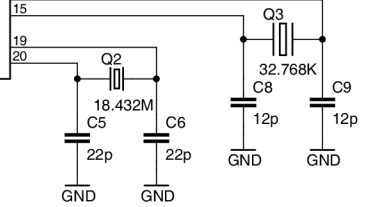
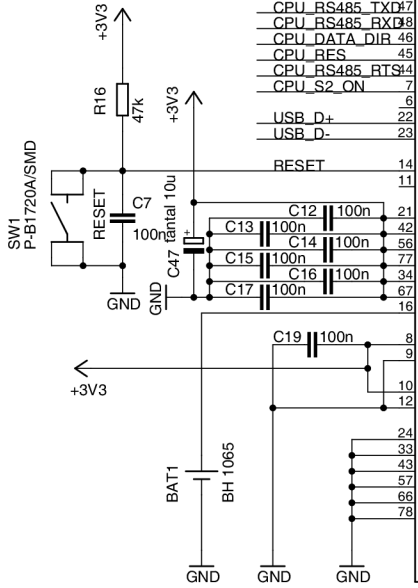
- P1[0]/ENET_TXD0
- P1[1]/ENET_TXD1
- P1[4]/ENET_TX_EN
- P1[8]/ENET_CRS
- P1[9]/ENET_RXD0
- P1[10]/ENET_RXD1
- P1[14]/ENET_RX_ER
- P1[15]/ENET_REF_CLK
- P1[18]/USB_UP_LED/PWM1[1]/CAP1[0]
- P1[19]/MCOA0/USB_PPWR/CAP1[1]
- P1[20]/MCI0/PWM1[2]/SCK0
- P1[22]/MCOB0/USB_PWRD/MAT[0]
- P1[23]/MCI1/PWM1[4]/MISO0
- P1[24]/MCI2/PWM1[5]/MOSI0
- P1[25]/MCOA1/MAT1[1]
- P1[26]/MCOB1/PWM1[6]/CAP0[0]
- P1[28]/MCOA2/PCAP1[0]/MAT0[0]
- P1[29]/MCOB2/PCAP1[1]/MAT0[1]
- P1[30]/VBUS/AD0[4]
- P1[31]/SCK1/AD0[5]

- GPIO0 76
- GPIO1 75
- GPIO2 74
- GPIO3 73
- GPIO4 72
- GPIO5 71
- GPIO6 70
- GPIO7 69
- USB_UP_LED 25
- CPU_VOLT_MONITOR 26
- CPU_WP 27
- CPU_AB3 28
- CPU_AB2 29
- CPU_AB1 30
- CPU_AB0 31
- CPU_READ 32
- CPU_WRITE 35
- CPU_EN 36
- V_BUS 18
- CPU_RS232_INVALID 17

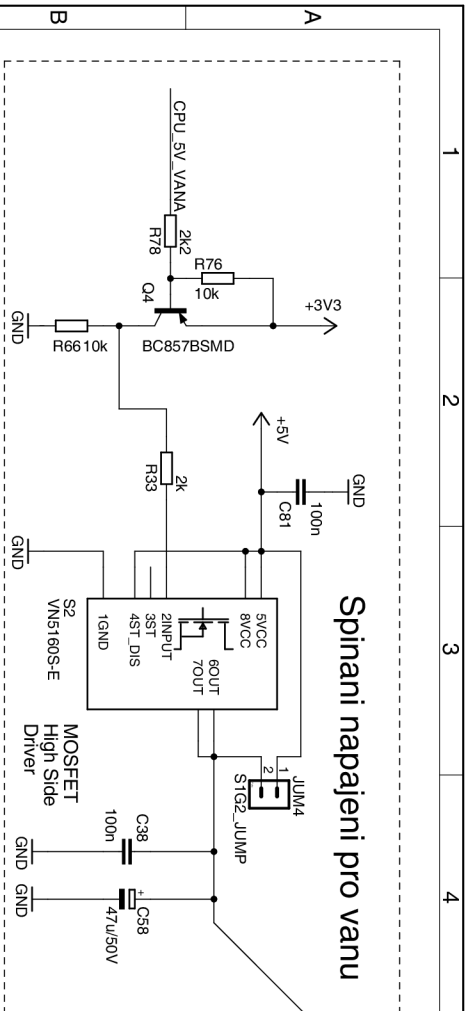
- P2[0]/PWM1[1]/TXD1
- P2[1]/PWM1[2]/RXD1
- P2[2]/PWM1[3]/CTS1/TRACEDATA[3]
- P2[3]/PWM1[4]/DCD1/TRACEDATA[2]
- P2[4]/PWM1[5]/DSR1/TRACEDATA[1]
- P2[5]/PWM1[6]/DTR1/TRACEDATA[0]
- P2[6]/PCAP1[0]/R11/TRACECLK
- P2[7]/RD2/RTS1
- P2[8]/TD2/TXD2/ENET_MDC
- P2[9]/USB_CONNECT/RXD2/ENET_MDIO
- P2[10]/EINT0/NMI
- P4[28]/RX_MCLK/MAT2[0]/TXD3
- P4[29]/TX_MCLK/MAT2[1]/RXD3

- 1 TDO/SWO
- 2 TDI
- 3 TMS/SWDIO
- 4 TRST
- 5 TCK/SWDCLK

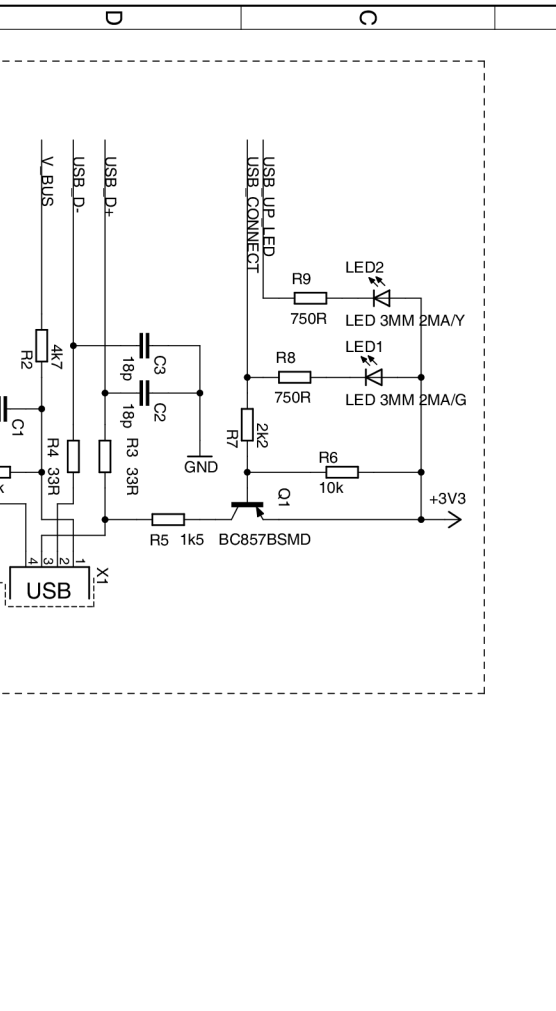
- 13 RTCX1
- 15 RTCX2
- 19 XTAL1
- 20 XTAL2



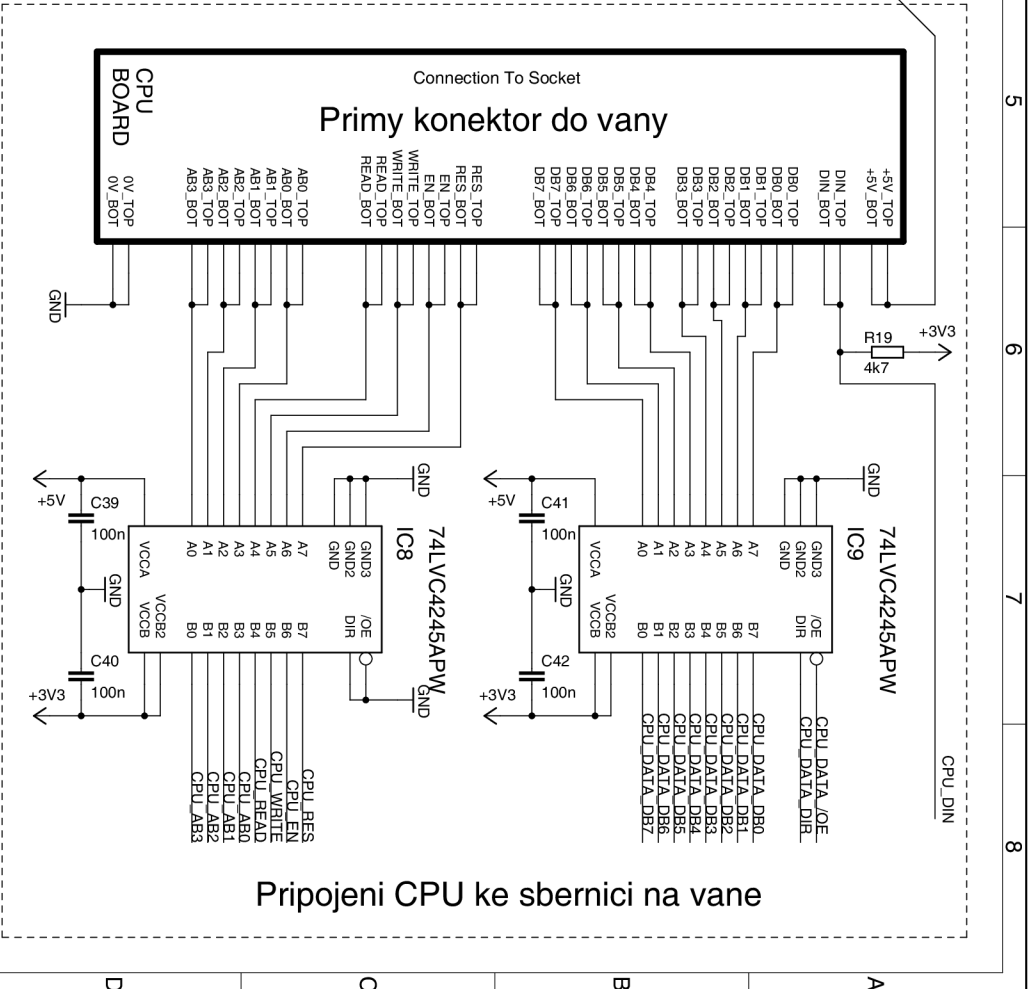
Ridici deska	RidiciDeska_d5
	15.5.2012 9:53:49
	Sheet: 3/8



Spinani napajeni pro vanu



USB Self-Power Device konfigurace (rezerva)

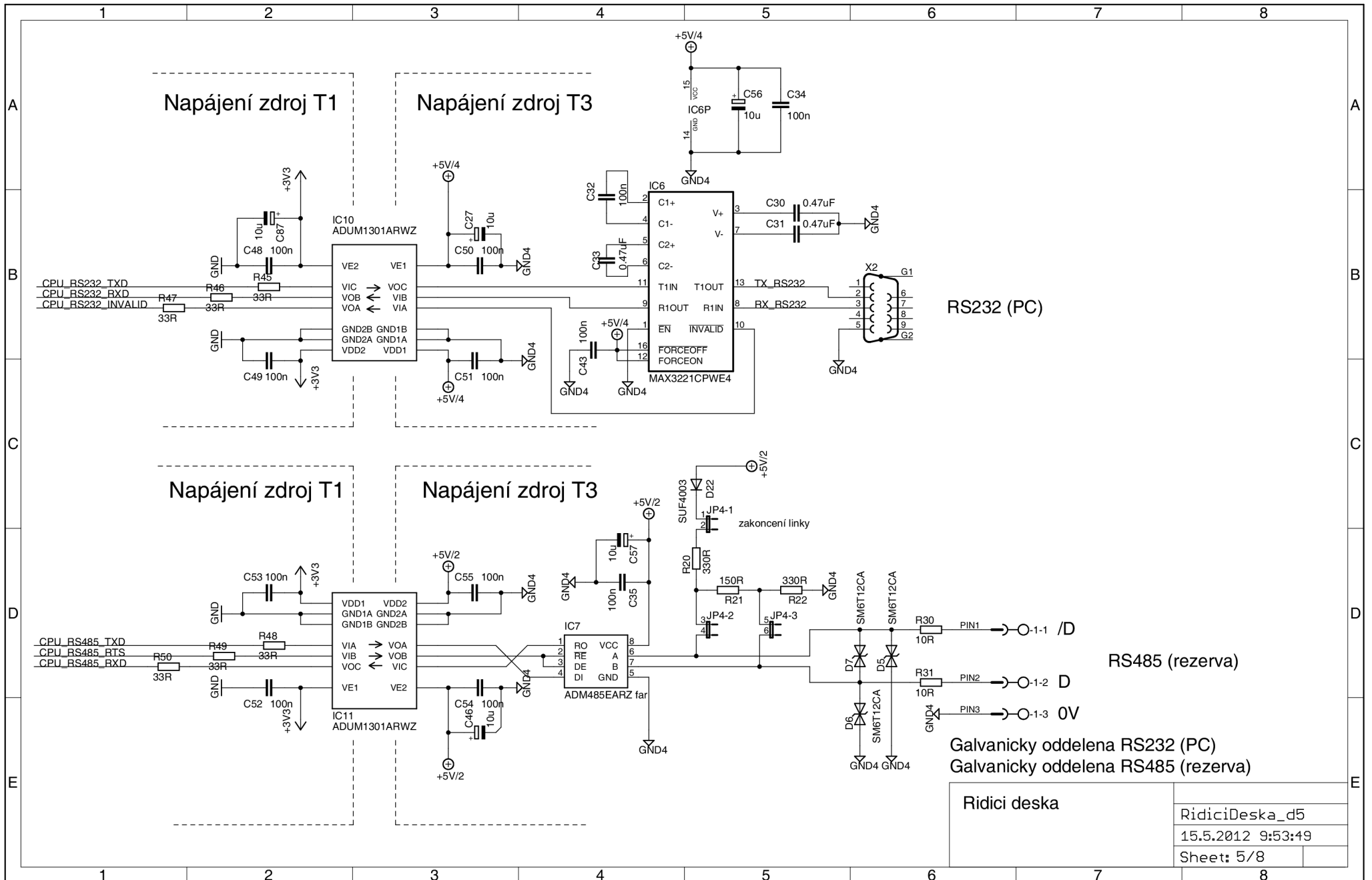


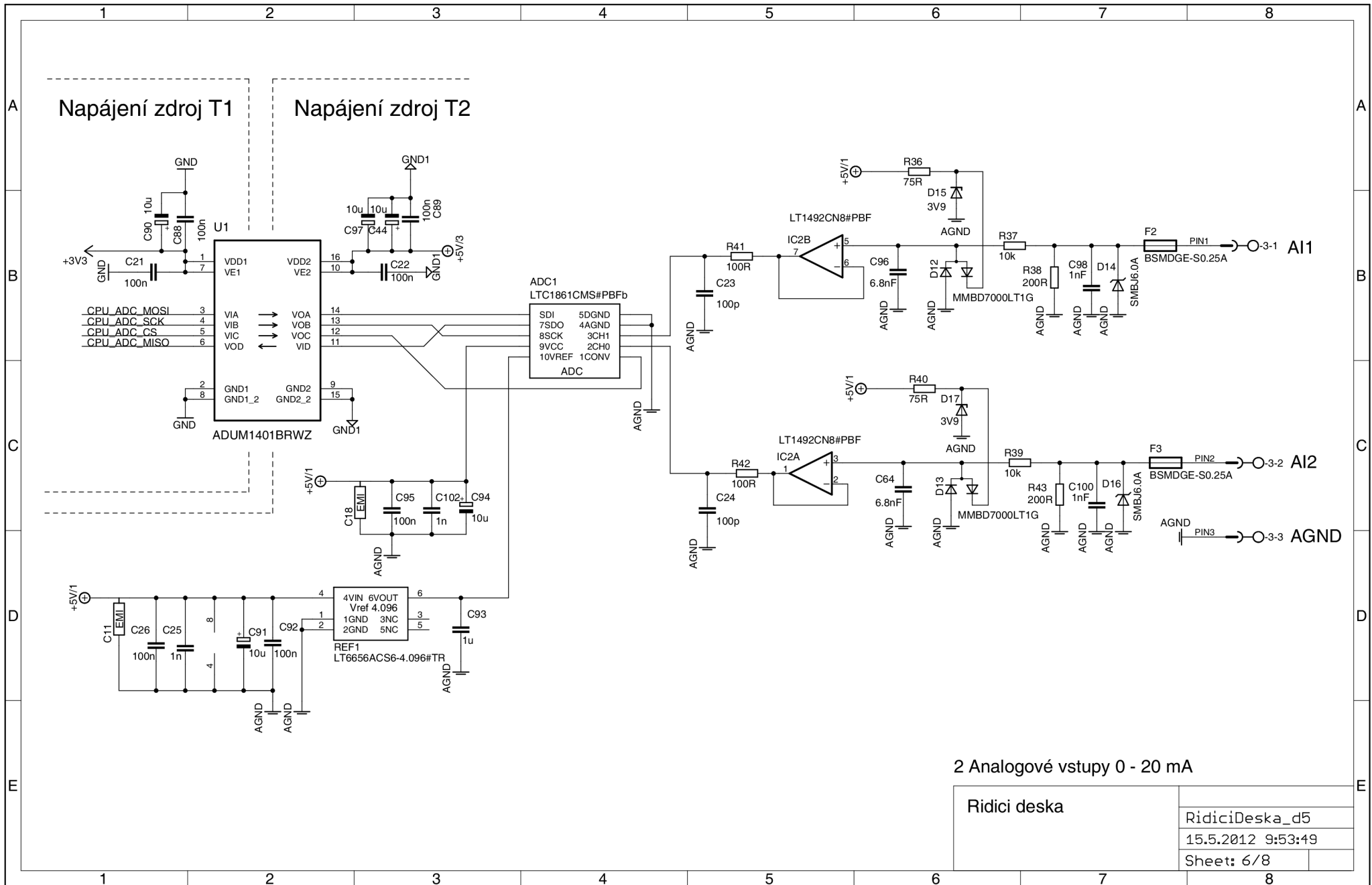
Primry konektor do vany

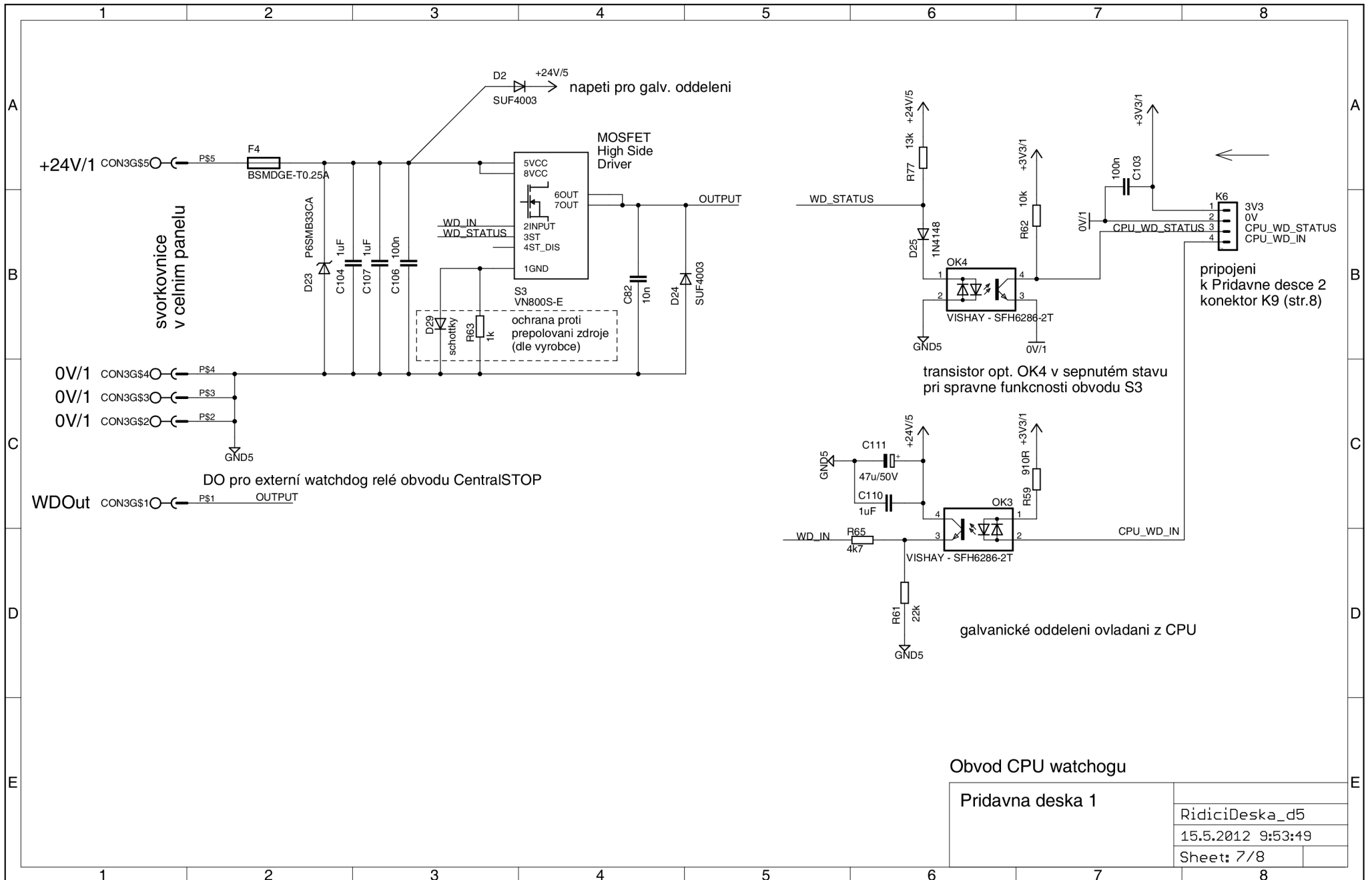
Pripojeni CPU ke sbernici na vane

Pripojeni CPU na sbernici ve vane
USB (rezerva)

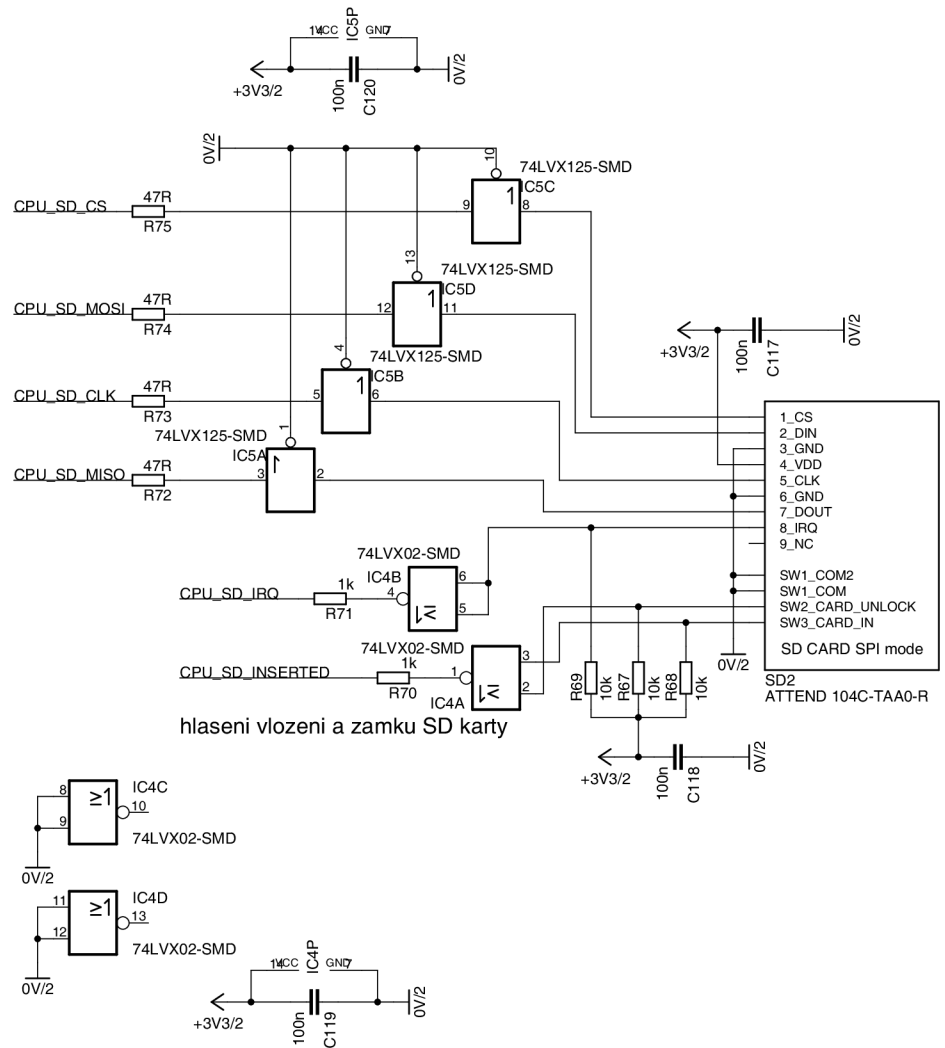
Ridici deska



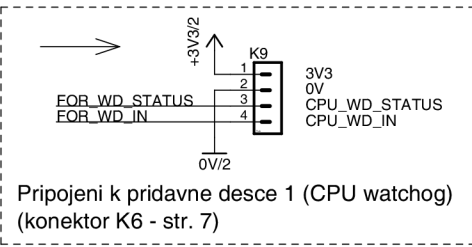




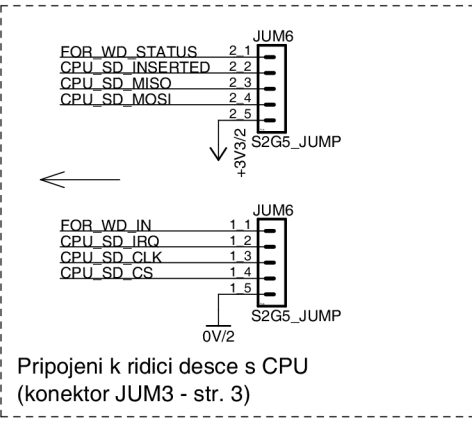
Priponjeni SD karty pomoci softwaroveho SPI



hlaseni vlozeni a zamku SD karty



Priponjeni k pridavne desce 1 (CPU watchdog) (konektor K6 - str. 7)



Priponjeni k ridici desce s CPU (konektor JUM3 - str. 3)

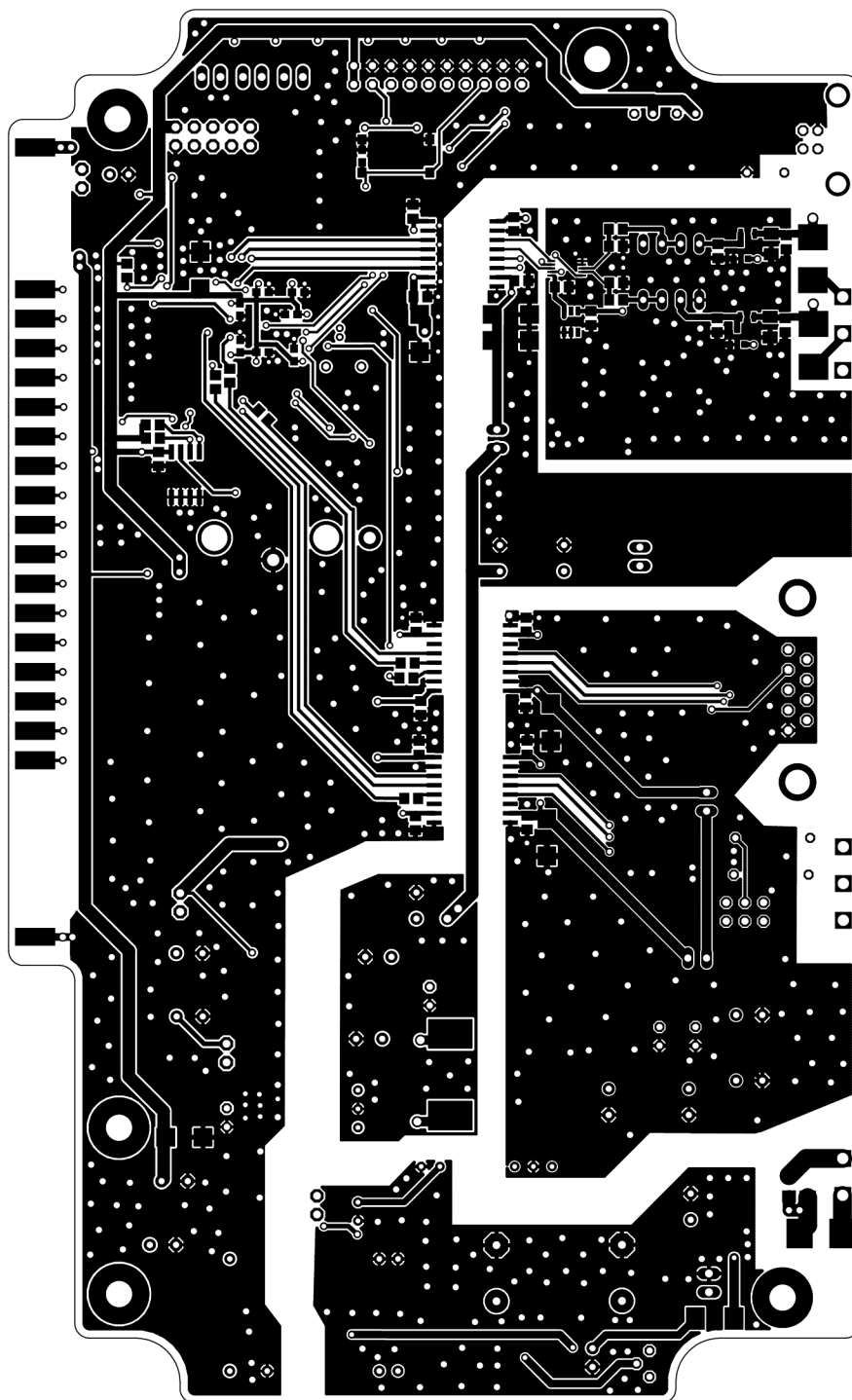
Sd card SPI

Pridavna deska 2	RidiciDeska_d5
	15.5.2012 9:53:49
	Sheet: 8/8

B.2 DPS

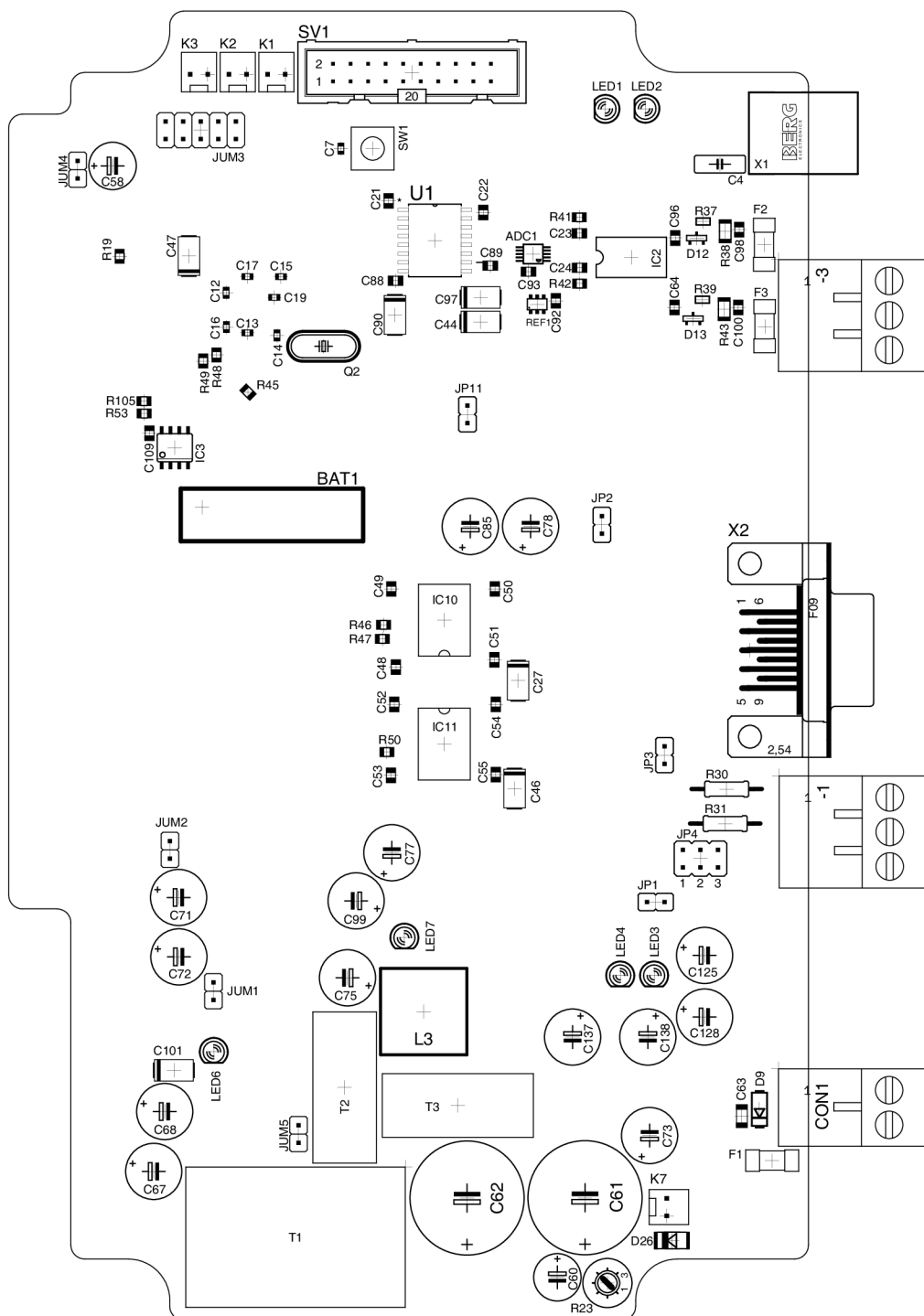
Výkres plošného spoje - hladina TOP

Ridici deska (CPU)



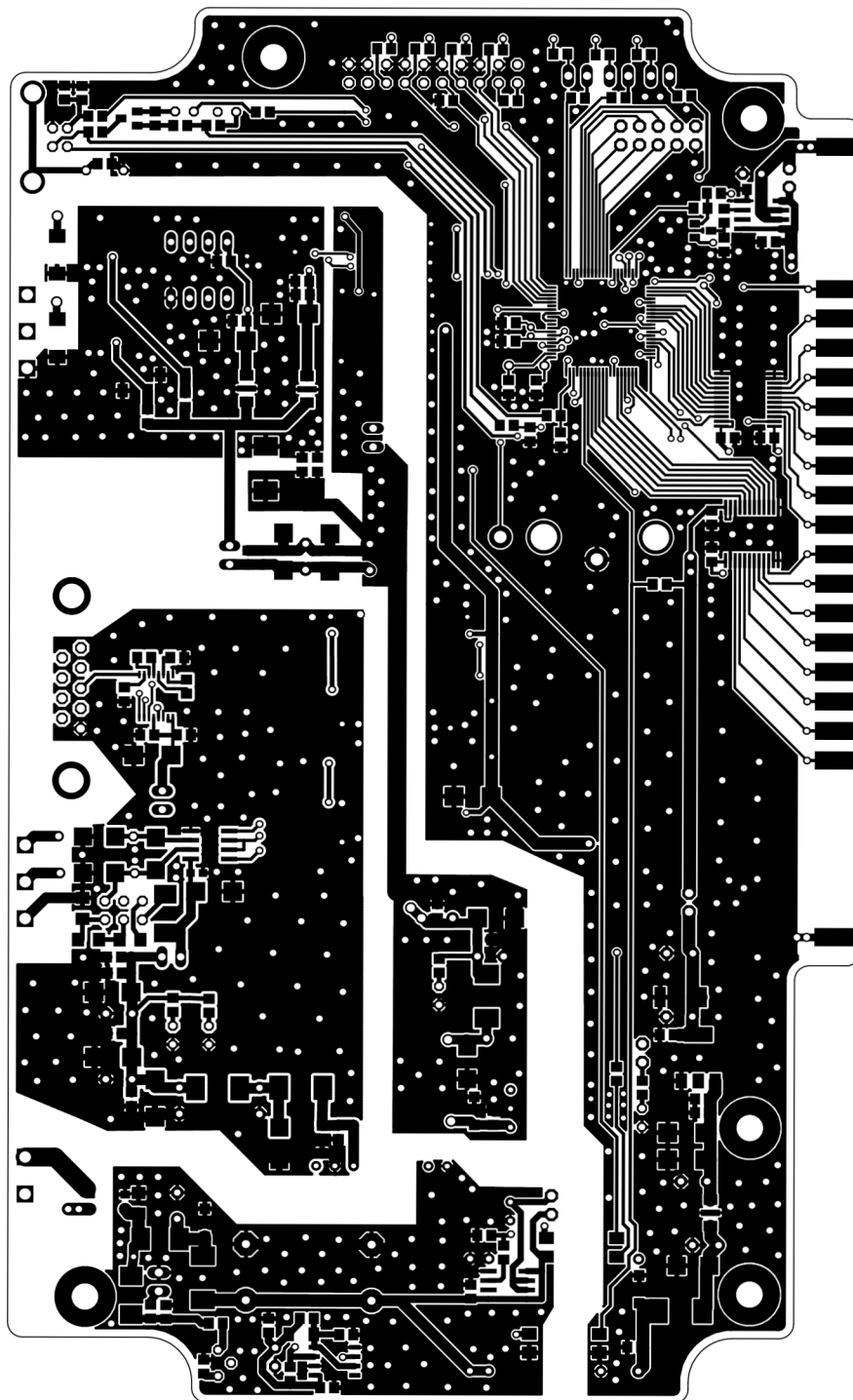
Osazovací výkres - hladina TOP

Ridici deska (CPU)



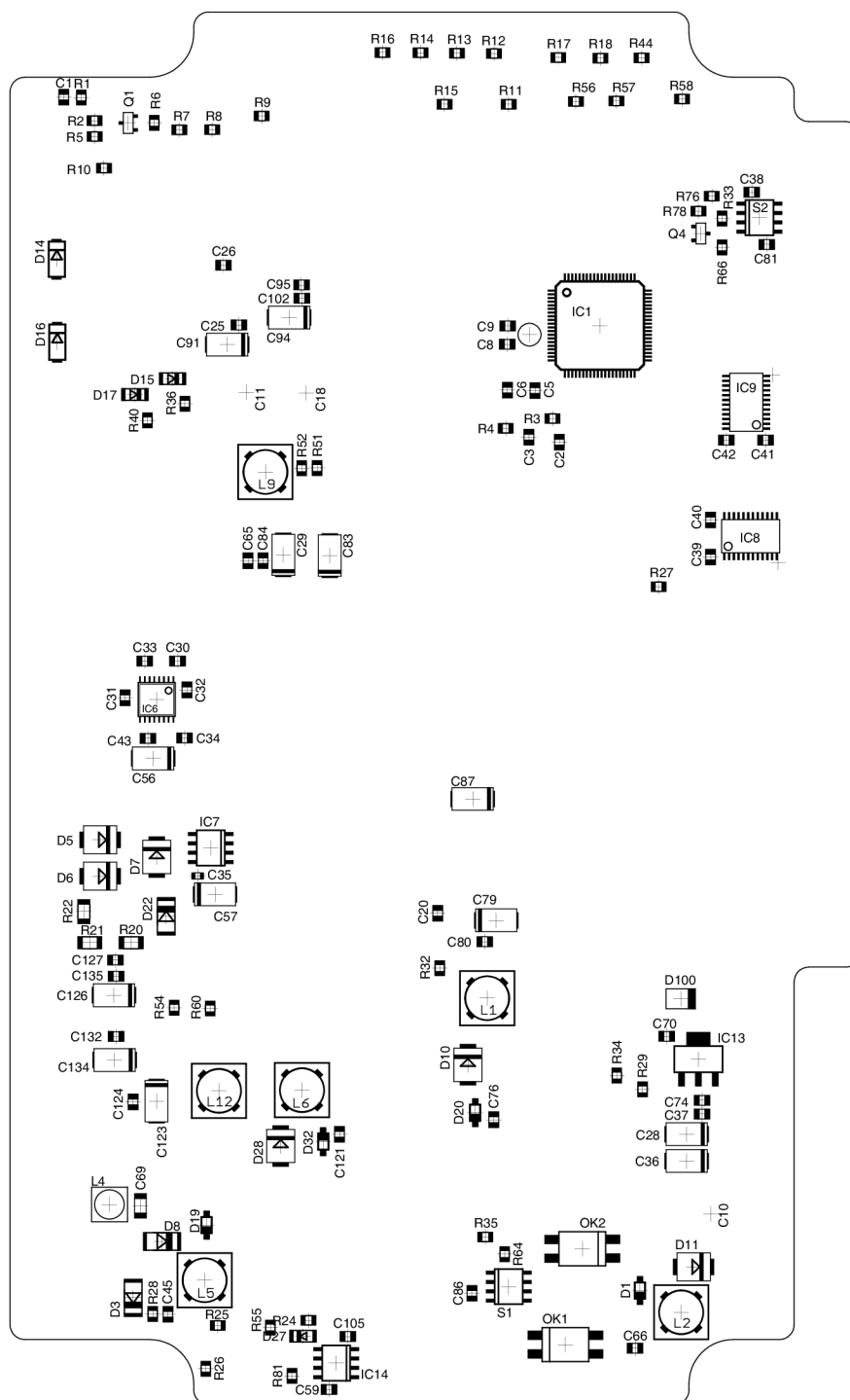
Výkres plošného spoje - hladina BOTTOM

Ridici deska (CPU)



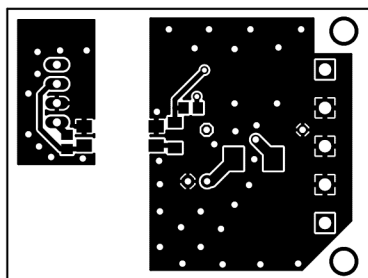
Osazovací výkres - hladina BOTTOM

Ridici deska (CPU)

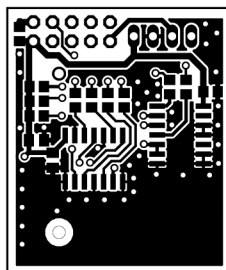


Výkres plošného spoje - hladina TOP

Pridavna deska 1 (WD)

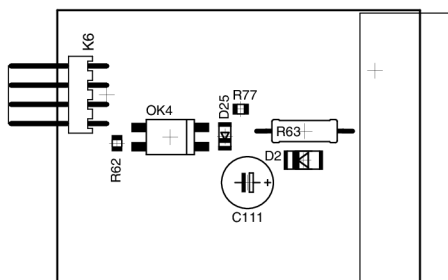


Pridavna deska 2 (SPI)

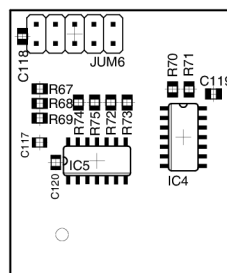


Osazovací výkres - hladina TOP

Pridavna deska 1 (WD)

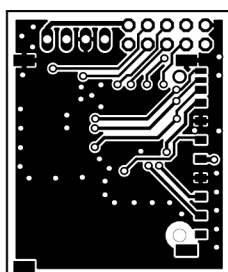


Pridavna deska 2 (SPI)

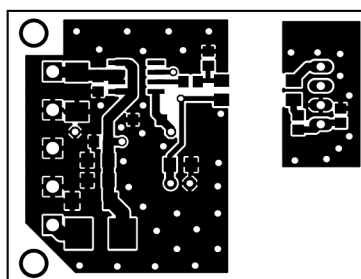


Výkres plošného spoje - hladina BOTTOM

Pridavna deska 2 (SPI)

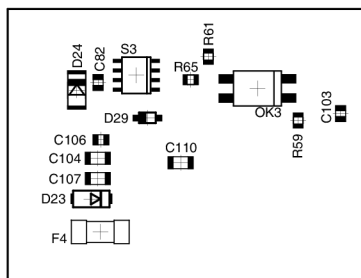
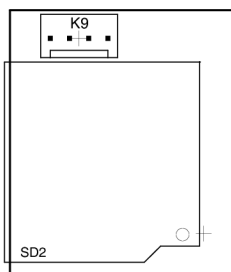


Pridavna deska 1 (WD)



Osazovací výkres - hladina BOTTOM

Přidavná deska 2 (SPI) Přidavná deska 1 (WD)



B.3 Seznam součástek

Ozn.	Hodnota	Pouzdro	Str.
-1	CAMDEN CTB9350/3A		5
-3	CAMDEN CTB9350/3A		6
ADC1	LTC1861CMS#PBFb	MSOP10	6
BAT1	BH 1065		3
C1	100n	C0805	4
C2	18p	C0805	4
C3	18p	C0805	4
C4	4n7 / 100 V	C050-025X075	4
C5	22p	C0805	3
C6	22p	C0805	3
C7	100n	C0603	3
C8	12p	C0805	3
C9	12p	C0805	3
C10	ELKE222FA	EMI	1
C11	ELKE222FA	EMI	6
C12	100n	C0603	3
C13	100n	C0603	3
C14	100n	C0603	3
C15	100n	C0603	3
C16	100n	C0603	3
C17	100n	C0603	3
C18	ELKE222FA	EMI	6
C19	100n	C0603	3
C20	1n	C0805	2
C21	100n	C0805	6
C22	100n	C0805	6
C23	100p	C0805	6
C24	100p	C0805	6
C25	1n	C0805	6
C26	100n	C0805	6
C27	10u	C/6032-28R	5
C28	tantal 10u	C/6032-28R	1
C29	tantal 10u	C/6032-28R	2
C30	0.47uF	C0805	5
C31	0.47uF	C0805	5
C32	100n	C0805	5

Ozn.	Hodnota	Pouzdro	Str.
C33	0.47uF	C0805	5
C34	100n	C0805	5
C35	100n	C0603	5
C36	tantal 10u	C/6032-28R	1
C37	100n	C0805	1
C38	100n	C0805	4
C39	100n	C0805	4
C40	100n	C0805	4
C41	100n	C0805	4
C42	100n	C0805	4
C43	100n	C0805	5
C44	10u	C/6032-28R	6
C45	100n	C0805	1
C46	10u	C/6032-28R	5
C47	tantal 10u	C/6032-28R	3
C48	100n	C0805	5
C49	100n	C0805	5
C50	100n	C0805	5
C51	100n	C0805	5
C52	100n	C0805	5
C53	100n	C0805	5
C54	100n	C0805	5
C55	100n	C0805	5
C56	10u	C/6032-28R	5
C57	10u	C/6032-28R	5
C58	47u/50V	E2,5-7	4
C59	100n	C0805	1
C60	GF47/50	E2,5-7	1
C61	1000uF/50V	E7,5-16	1
C62	1000uF/50V	E7,5-16	1
C63	1uF	C1206	1
C64	6.8nF	C0805	6
C65	1n	C0805	2
C66	100n	C0805	1
C67	220u/25V	E3,5-8	1
C68	220u/25V	E3,5-8	1
C69	1uF	C1206	1
C70	100n	C0805	1
C71	220u/25V	E3,5-8	1
C72	220u/25V	E3,5-8	1
C73	100uF/50V	E3,5-8	1
C74	1n	C0805	1
C75	390u/25V	E3,5-8	2
C76	100n	C0805	2
C77	390u/25V	E3,5-8	2
C78	390u/25V	E3,5-8	2
C79	tantal 10u	C/6032-28R	2
C80	100n	C0805	2
C81	100n	C0805	4
C82	10n	C0805	7
C83	tantal 10u	C/6032-28R	2
C84	100n	C0805	2
C85	390u/25V	E3,5-8	2
C86	100n	C0805	2
C87	10u	C/6032-28R	5
C88	100n	C0805	6

Ozn.	Hodnota	Pouzdro	Str.
C89	100n	C0805	6
C90	10u	C/6032-28R	6
C91	10u	C/6032-28R	6
C92	100n	C0805	6
C93	1u	C0805	6
C94	10u	C/6032-28R	6
C95	100n	C0805	6
C96	6.8nF	C0805	6
C97	10u	C/6032-28R	6
C98	1nF	C0805	6
C99	390u/25V	E3,5-8	2
C100	1nF	C0805	6
C101	tantal 10u	C/6032-28R	1
C102	1n	C0805	6
C103	100n	C0805	7
C104	1uF	C1206	7
C105	100n	C0805	1
C106	100n	C0805	7
C107	1uF	C1206	7
C109	100n	C0805	3
C110	1uF	C1206	7
C111	47u/50V	E2,5-7	7
C117	100n	C0805K	8
C118	100n	C0805	8
C119	100n	C0805	8
C120	100n	C0805K	8
C121	100n	C0805	2
C123	tantal 10u	C/6032-28R	2
C124	100n	C0805	2
C125	390u/25V	E3,5-8	2
C126	tantal 10u	C/6032-28R	2
C127	100n	C0805	2
C128	390u/25V	E3,5-8	2
C132	1n	C0805	2
C134	tantal 10u	C/6032-28R	2
C135	1n	C0805	2
C137	390u/25V	E3,5-8	2
C138	390u/25V	E3,5-8	2
CON1	CAMDEN CTB9350/2A		1
CON3			7
D1	schottky RSX101M-30TR	SOD123	1
D2	SUF4003	MELF-MLL41	7
D3	SUF4003	MELF-MLL41	1
D5	SM6T12CA	SMB	5
D6	SM6T12CA	SMB	5
D7	SM6T12CA	SMB	5
D8	SUF4003	MELF-MLL41	1
D9	P6SMB36A	DO214AA	1
D10	SMBJ5.0A SMD	SMB	2
D11	SMBJ5.0A SMD	SMB	1
D12	MMBD7000LT1G	SOT-23	6
D13	MMBD7000LT1G	SOT-23	6
D14	SMBJ6.0A	DO214AA	6
D15	3V9	SOD80C	6
D16	SMBJ6.0A	DO214AA	6
D17	3V9	SOD80C	6

Ozn.	Hodnota	Pouzdro	Str.
D19	schottky RSX101M-30TR	SOD123	1
D20	schottky RSX101M-30TR	SOD123	2
D22	SUF4003	MELF-MLL41	5
D23	P6SMB33A	DO214AA	7
D24	SUF4003	MELF-MLL41	7
D25	1N4148	MINIMELF	7
D26	SUF4003	MELF-MLL41	1
D27	1N4148	MINIMELF	1
D28	SMBJ5.0A SMD	SMB	2
D29	schottky	SOD123	7
D32	schottky RSX101M-30TR	SOD123	2
D100	3V3	SOD128	1
F1	BSMDGE-T1.0A	SSQC	1
F2	BSMDGE-S0.25A	SSQC	6
F3	BSMDGE-S0.25A	SSQC	6
F4	BSMDGE-T0.25A	SSQC	7
IC1	LPC1756	LQFP80-12X12	3
IC2	LT1492CN8#PBF	DIL08	6
IC3	FM24CL16-G	SOIC8	3
IC4	74LVX02-SMD	SO14	8
IC5	74LVX125-SMD	SO14	8
IC6	MAX3221CPWE4	TSSOP16	5
IC7	ADM485EARZ far	SOIC8	5
IC8	74LVC4245APW	TSSOP24	4
IC9	74LVC4245APW	TSSOP24	4
IC10	ADUM1301ARWZ	RW-16	5
IC11	ADUM1301ARWZ	RW-16	5
IC13	LD1117S33CTR	SOT223	1
IC14	MC34161D	SOIC8	1
JP1		JP1	2
JP2		JP1	2
JP3		JP1	2
JP4		JP3Q	5
JP11		JP1	2
JUM1		S1G2_JUM	1
JUM2		S1G2_JUM	1
JUM3	MLW10	S2G5_JUM	3
JUM4	S1G2_JUMP	S1G2_JUM	4
JUM5		S1G2_JUM	2
JUM6	S2G5_JUMP	S2G5_JUM	8
K1	PSH02-02PG	PSH02-02	3
K2	PSH02-02PG	PSH02-02	3
K3	PSH02-02PG	PSH02-02	3
K6	PSH02-04W	PSH02-04W	7
K7	PSH02-02PG	PSH02-02	1
K9	PSH02-04	PSH02-04	8
L1	ELL6RH390M 39uH	DR73	2
L2	ELL6UH470M 47uH	DR73	1
L3	B82477P4105M000 1mH	B82477P4105M000	2
L4	ACM4520	ACM	1
L5	ELL6UH470M 47uH	DR73	1
L6	ELL6RH390M 39uH	DR73	2
L9	ELL6UH470M 47uH	DR73	2
L12	ELL6RH390M 39uH	DR73	2
I.01	LED 3MM 2MA/G	LED3MM	4
I.02	LED 3MM 2MA/Y	LED3MM	4

Ozn.	Hodnota	Pouzdro	Str.
I.03	LED 3MM GREEN 20mA	LED3MM	2
I.04	LED 3MM GREEN 20mA	LED3MM	2
I.06	LED 3MM 2MA/Y	LED3MM	1
I.07	LED 3MM GREEN 20mA	LED3MM	2
OK1	VISHAY - SFH6286-2T	DIL4-SMD	1
OK2	VISHAY - SFH6286-2T	DIL4-SMD	2
OK3	VISHAY - SFH6286-2T	DIL4-SMD	7
OK4	VISHAY - SFH6286-2T	DIL4-SMD	7
Q1	BC857BSMD	SOT23-BEC	4
Q2	18.432M	HC49/S	3
Q3	32.768K	CFS308	3
Q4	BC857BSMD	SOT23-BEC	4
R1	220k	R0805	4
R2	4k7	R0805	4
R3	33R	R0805	4
R4	33R	R0805	4
R5	1k5	R0805	4
R6	10k	R0805	4
R7	2k2	R0805	4
R8	750R	R0805	4
R9	750R	R0805	4
R10	1M	R0805	4
R11	10k	R0805	3
R12	10k	R0805	3
R13	10k	R0805	3
R14	10k	R0805	3
R15	10k	R0805	3
R16	47k	R0805	3
R17	750R	R0805	3
R18	750R	R0805	3
R19	4k7	R0805	4
R20	330R	R1206	5
R21	150R	R1206	5
R22	330R	R1206	5
R23	1k	RJ6	1
R24	10k	R0805	1
R25	22k	R0805	1
R26	1k1	R0805	1
R27	10k	R0805	1
R28	11k	R0805	1
R29	750R	R0805	1
R30	10R	0207/10	5
R31	10R	0207/10	5
R32	150R	R0805	2
R33	2k	R0805	4
R34	910R	R0805	2
R35	22k	R0805	2
R36	75R	R0805	6
R37	10k	MINI_MELF-0102W	6
R38	200R	M1206	6
R39	10k	MINI_MELF-0102W	6
R40	75R	R0805	6
R41	100R	R0805	6
R42	100R	R0805	6
R43	200R	M1206	6
R44	750R	R0805	3

Ozn.	Hodnota	Pouzdro	Str.
R45	33R	R0805	5
R46	33R	R0805	5
R47	33R	R0805	5
R48	33R	R0805	5
R49	33R	R0805	5
R50	33R	R0805	5
R51	0R	R0805	2
R52	0R	R0805	2
R53	1k1	R0805	3
R54	150R	R0805	2
R56	150R	R0805	3
R57	150R	R0805	3
R58	150R	R0805	3
R59	910R	R0805	7
R60	150R	R0805	2
R61	22k	R0805	7
R62	10k	R0805	7
R63	1k	0309/12	7
R64	4k7	R0805	2
R65	4k7	R0805	7
R66	10k	R0805	4
R67	10k	R0805	8
R68	10k	R0805	8
R69	10k	R0805	8
R70	1k	R0805	8
R71	1k	R0805	8
R72	47R	R0805	8
R73	47R	R0805	8
R74	47R	R0805	8
R75	47R	R0805	8
R76	10k	R0805	4
R77	13k	R0805	7
R78	2k2	R0805	4
R81	6k2	R0805	1
R105	1k1	R0805	3
REF1	LT6656ACS6-4.096#TR	SOT23-6	6
S1	VN800S-E	SO8	2
S2	VN5160S-E	SO8	4
S3	VN800S-E	SO8	7
SD2	ATTEND 104C-TAA0-R	SD	8
SV1	MLW20G	ML20	3
SW1	P-B1720A/SMD	P-B1720XXSMD	3
T1	JTC0624S05	XPPOWER_JCJ_PACKAGE	1
T2	IW2405SA	IW2405SA	2
T3	IW2405SA	IW2405SA	2
U1	ADUM1401BRWZ	SOIC127P1032X265-16N	6
X1	USB socket	PN61729-S	4
X2	CAN 9 Z 90	F09HP	5

B.4 Fotodokumentace osazených desek

Provedení DPS je jako oboustranná deska s prokoy, ošetřena nepájivou maskou. Přímé konektory pro zasunutí do soketu na vaně jsou galvanicky pokovené zlatem.

Řídicí deska a přídatné desky (dále jen řídicí modul) jsou namontovány do krabice (obrázky B.1, respektive B.2) namísto originálního rozšiřovacího modulu, který sloužil pro připojení vany k centrální jednotce automatu a napájení karet. Do čelního panelu jsou vyvedeny 3mm LED indikující stav automatu. Vyvedena je rovněž i LED „DC OK“ signalizující připojení napájení pro řídicí desku. 25pinový CAN konektor slouží pro programování CPU pomocí rozhraní JTAG. Prostřednictvím svorkovnice „WD“ je k čelnímu panelu přimontována „Přídavná deska 2“ s obvodem WatchDog, která má vlastní vnější 24VDC napájení. Svorka „WDOut“ slouží pro připojení vnějšího relé se spínacím kontaktem, který je zařazen do obvodu CentralSTOP. Ostatní svorkovnice jsou přímo osazeny na DPS.



Obr. B.1: Řídicí modul - sestava řídicí desky a přídatných desek



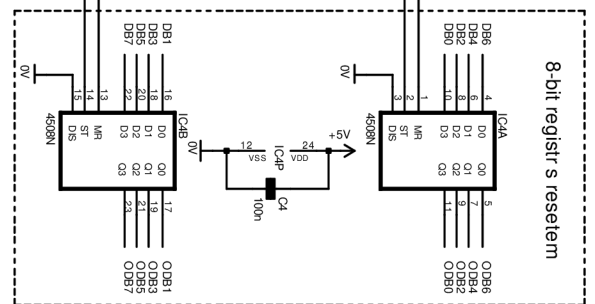
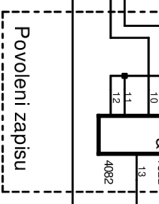
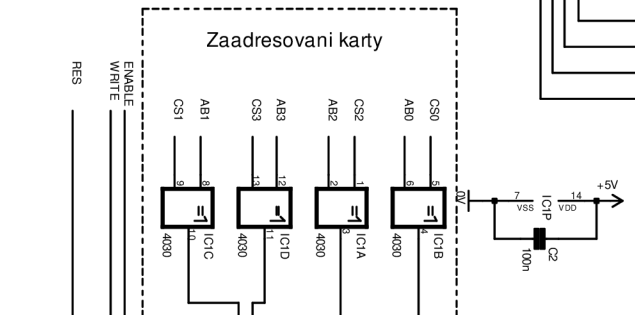
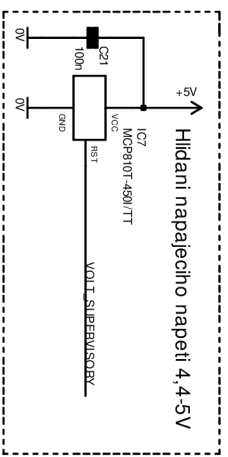
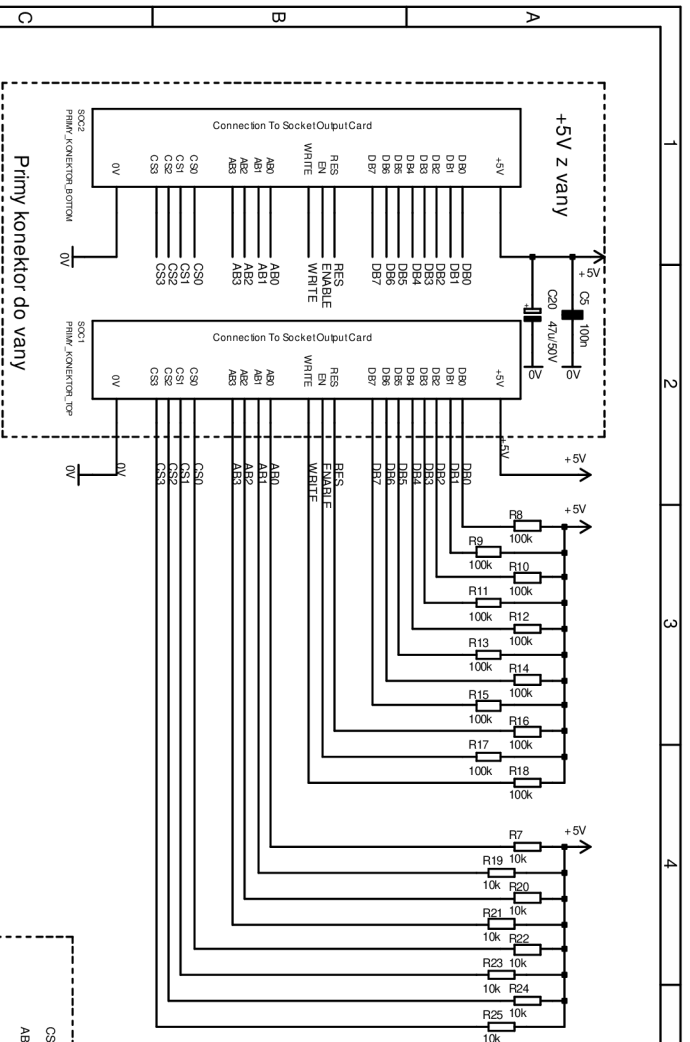
Obr. B.2: Řídicí modul - čelní panel

C 8BITOVÁ 24VDC DO KARTA

C.1 Schéma zapojení

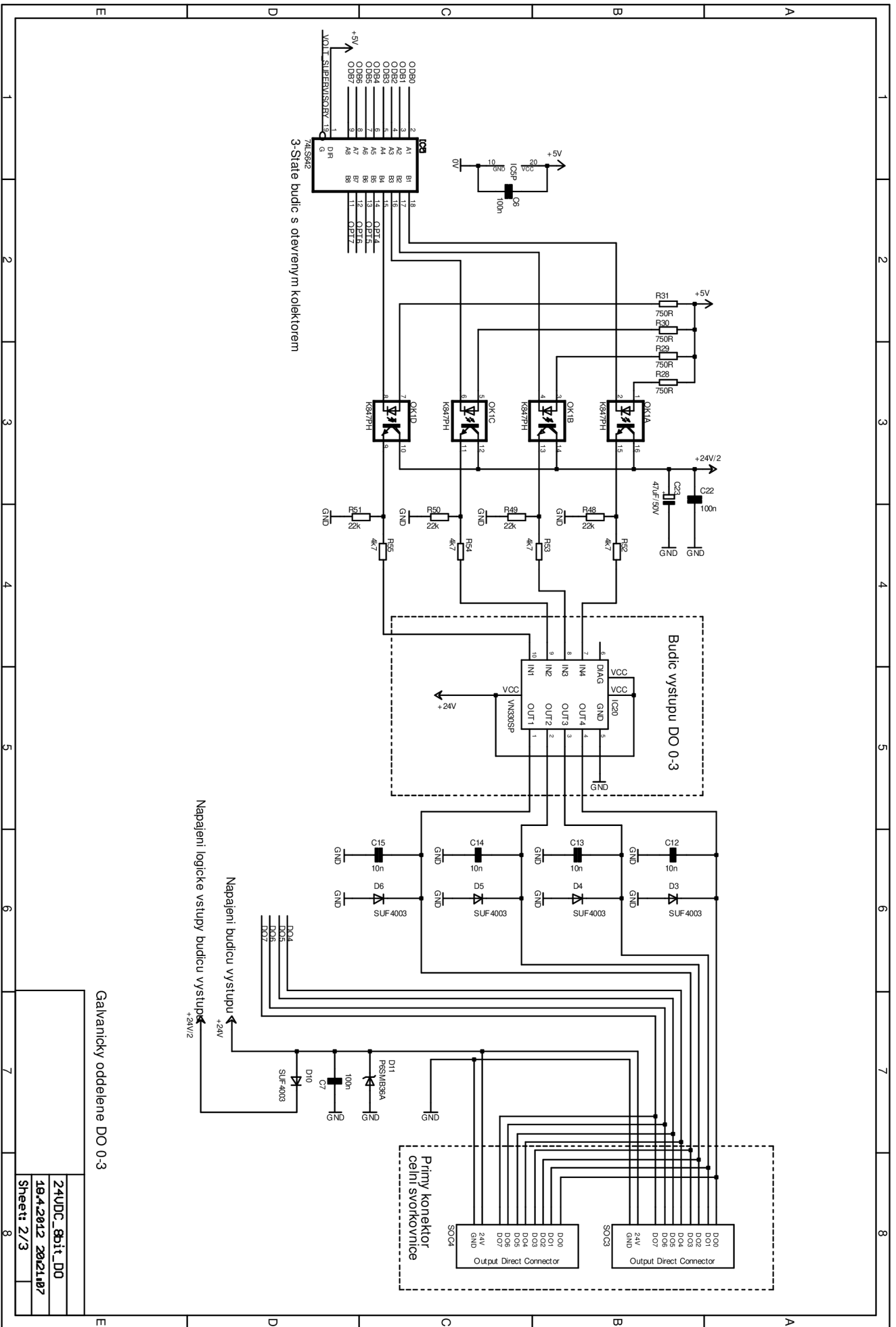
Schéma je přehledně rozčleněno do několika funkčních celků na jednotlivých stránkách:

1. připojení na sběrnici vany, logické obvody pro adresaci karty, 8bitový paměťový registr
2. obvody galvanicky oddělených výstupů DO 0-3
3. obvody galvanicky oddělených výstupů DO 4-7, LED kontrolky pro zobrazení stavu výstupů



První konektor do vany
Logické obvody karty

24UDC_8bit_DO
19.4.2012 20:21:07
Sheet 1/3



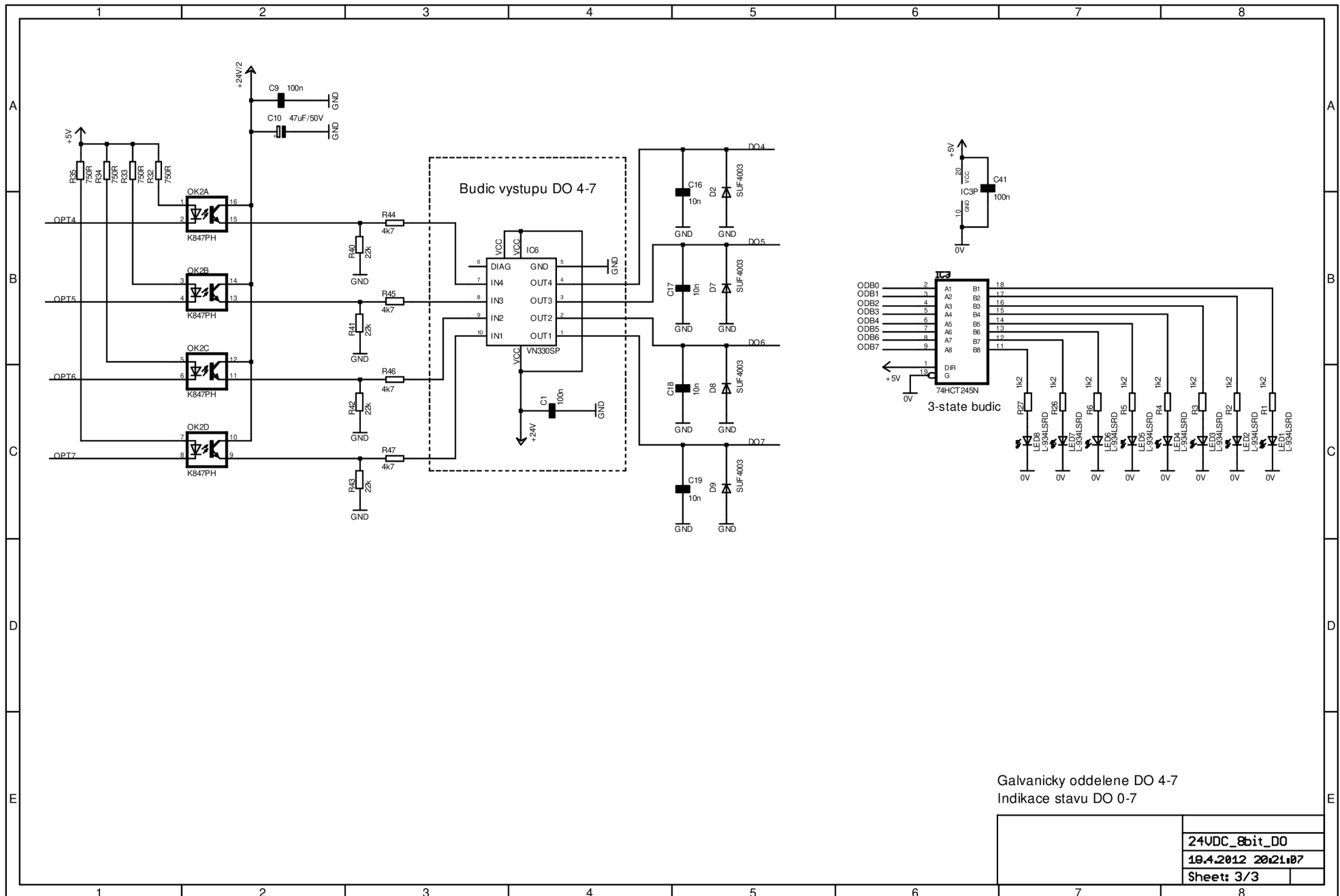
3-State buďic s otevrenym kolektorem

Buďic vstupu DO 0-3

Primny konektor
caini svorkovnice

Galvanicky oddelene DO 0-3

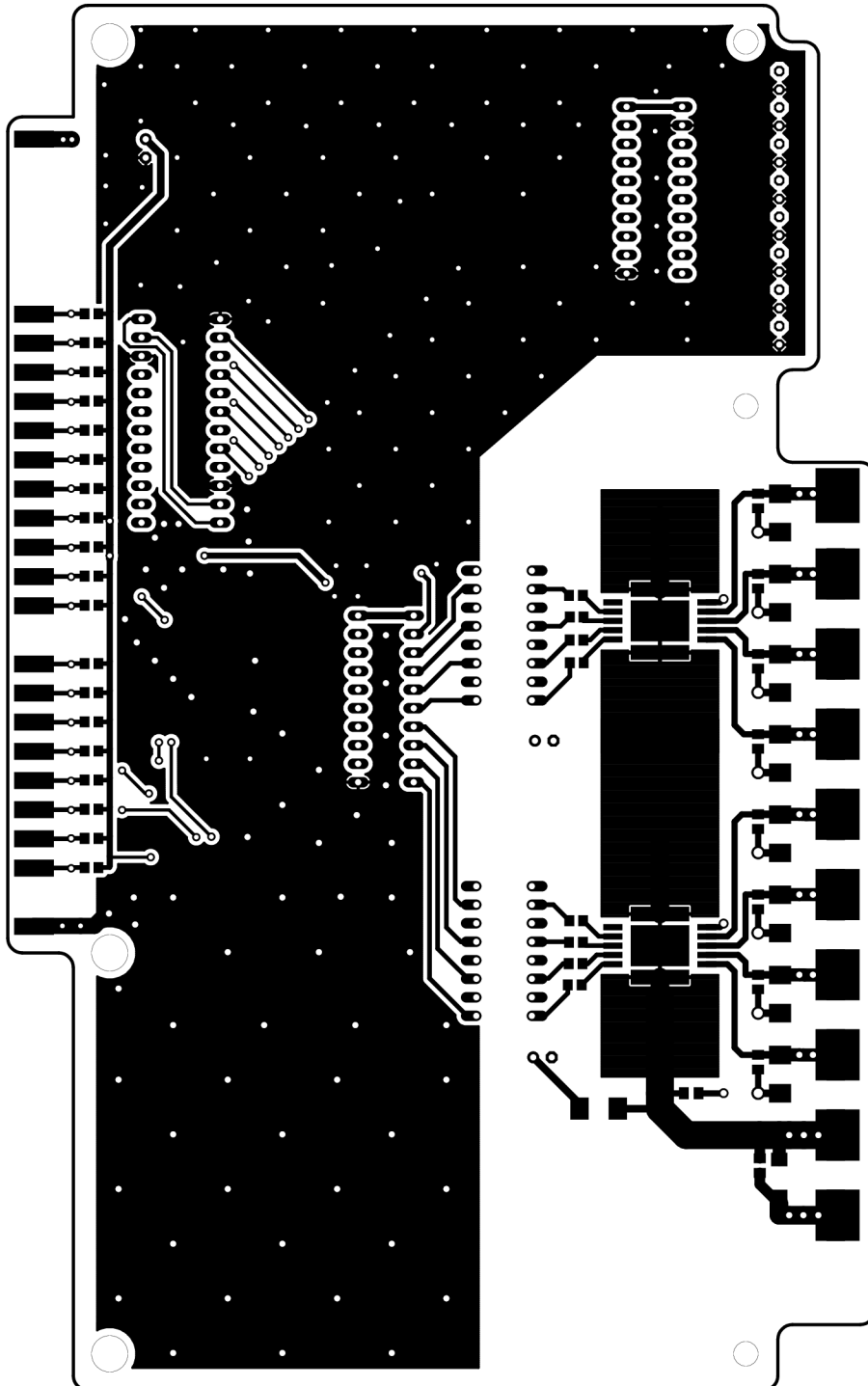
24VDC_Buť_DO
19.4.2012 20:21:07
Sheet: 2/3



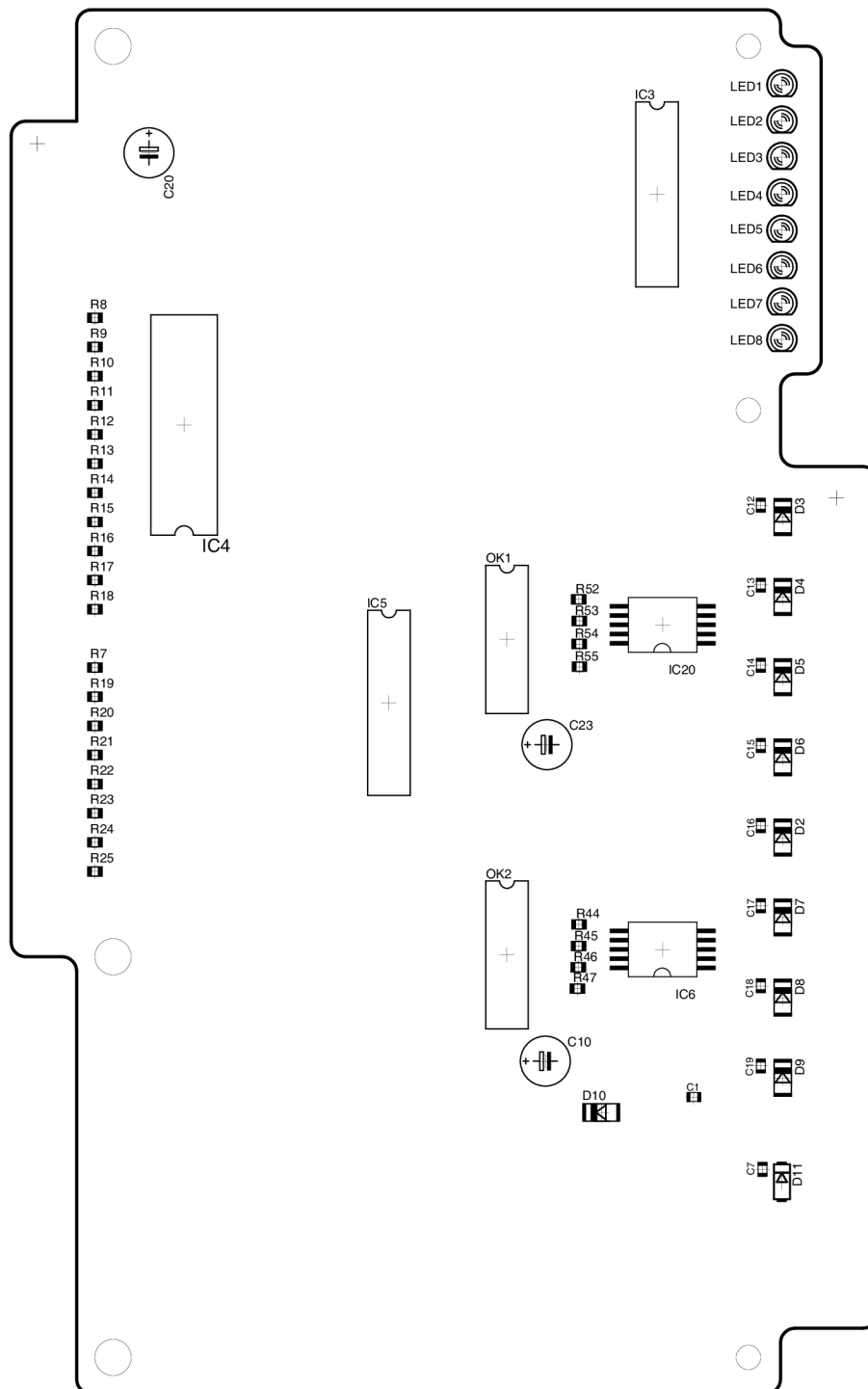
Galvanicky oddelene DO 4-7
Indikace stavu DO 0-7

C.2 DPS

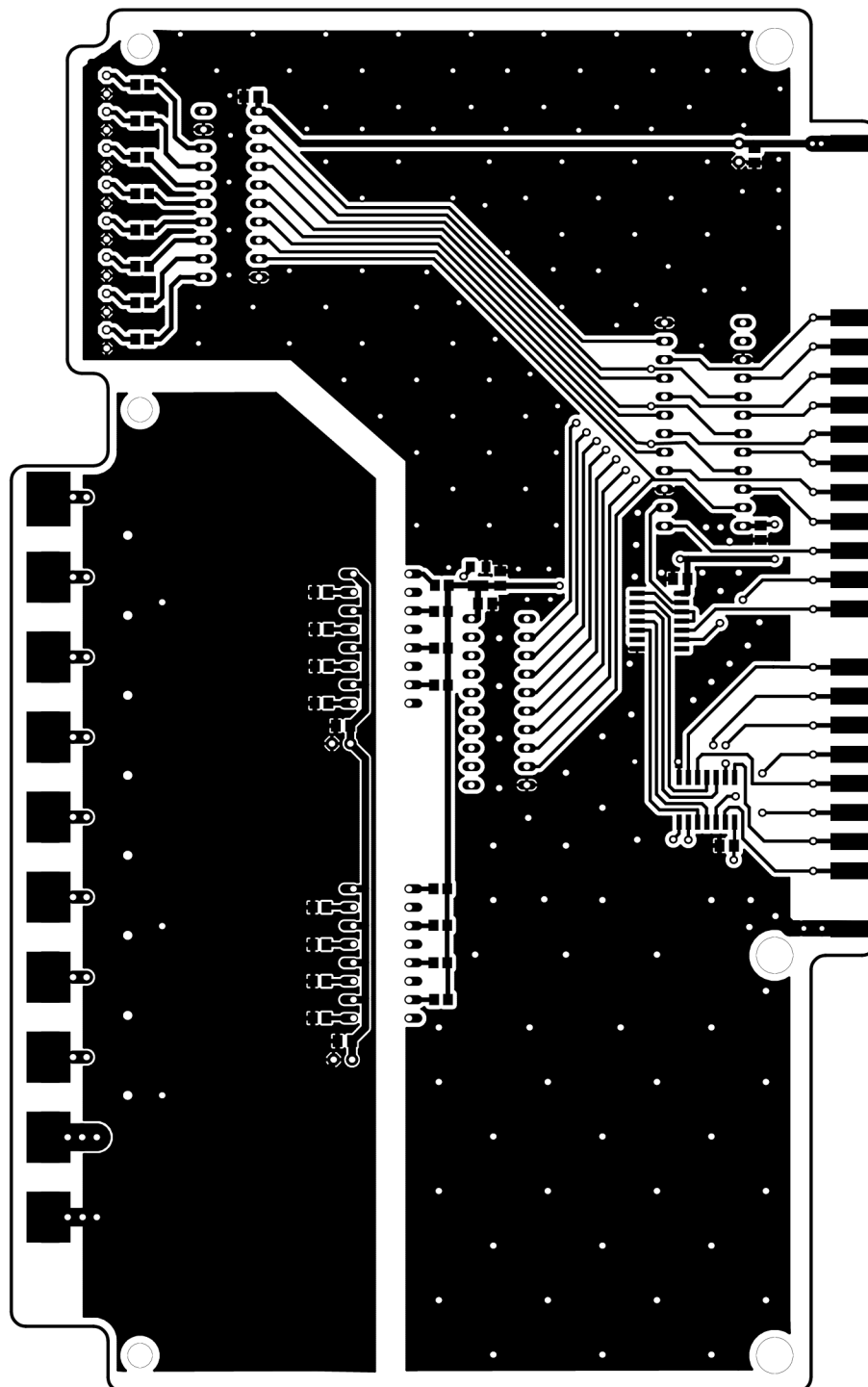
Výkres plošného spoje - hladina TOP



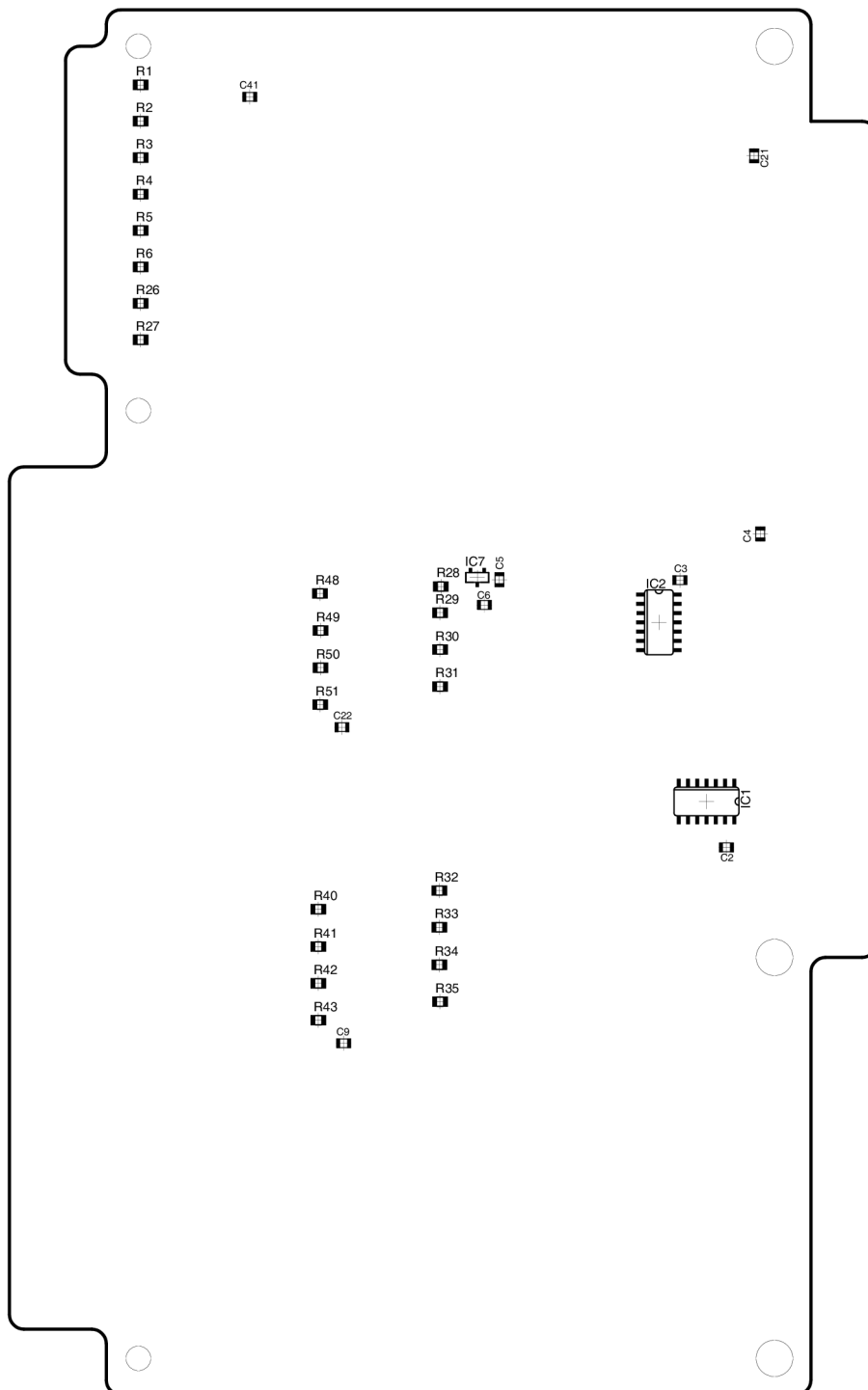
Osazovací výkres - hladina TOP



Výkres plošného spoje - hladina BOTTOM



Osazovací výkres - hladina BOTTOM



C.3 Seznam součástek

Ozn.	Hodnota	Pouzdro	Str.
C1	100n	C0805K	3
C2	100n	C0805K	1
C3	100n	C0805K	1
C4	100n	C0805K	1
C5	100n	C0805K	1
C6	100n	C0805K	2
C7	100n	C0805K	2
C9	100n	C0805K	3
C10	47uF/50V	E2,5-7	3
C12	10n	C0805K	2
C13	10n	C0805K	2
C14	10n	C0805K	2
C15	10n	C0805K	2
C16	10n	C0805K	3
C17	10n	C0805K	3
C18	10n	C0805K	3
C19	10n	C0805K	3
C20	47u/50V	E2,5-7	1
C21	100n	C0805K	1
C22	100n	C0805K	2
C23	47uF/50V	E2,5-7	2
C41	100n	C0805K	3
D2	SUF4003	MELF-MLL41	3
D3	SUF4003	MELF-MLL41	2
D4	SUF4003	MELF-MLL41	2
D5	SUF4003	MELF-MLL41	2
D6	SUF4003	MELF-MLL41	2
D7	SUF4003	MELF-MLL41	3
D8	SUF4003	MELF-MLL41	3
D9	SUF4003	MELF-MLL41	3
D10	SUF4003	MELF-MLL41	2
D11	P6SMB36A	DO214AA	2
IC1	4030	SO14	1
IC2	4082	SO14	1
IC3	74HCT245N	DIL20	3
IC4	4508N	DIL24-6	1
IC5	74LS642	DIL20	2
IC6	VN330SP	SO-08	3
IC7	MCP810T-450I/TT	SOT23-W	1
IC20	VN330SP	SO-08	2
I.01	L-934LSRD	LED3MM	3
I.02	L-934LSRD	LED3MM	3
I.03	L-934LSRD	LED3MM	3
I.04	L-934LSRD	LED3MM	3
I.05	L-934LSRD	LED3MM	3
I.06	L-934LSRD	LED3MM	3
I.07	L-934LSRD	LED3MM	3
I.08	L-934LSRD	LED3MM	3
OK1	K847PH	DIL16	2
OK2	K847PH	DIL16	3
R1	1k2	R0805	3
R2	1k2	R0805	3
R3	1k2	R0805	3
R4	1k2	R0805	3

Ozn.	Hodnota	Pouzdro	Str.
R5	1k2	R0805	3
R6	1k2	R0805	3
R7	10k	R0805	1
R8	100k	R0805	1
R9	100k	R0805	1
R10	100k	R0805	1
R11	100k	R0805	1
R12	100k	R0805	1
R13	100k	R0805	1
R14	100k	R0805	1
R15	100k	R0805	1
R16	100k	R0805	1
R17	100k	R0805	1
R18	100k	R0805	1
R19	10k	R0805	1
R20	10k	R0805	1
R21	10k	R0805	1
R22	10k	R0805	1
R23	10k	R0805	1
R24	10k	R0805	1
R25	10k	R0805	1
R26	1k2	R0805	3
R27	1k2	R0805	3
R28	750R	R0805	2
R29	750R	R0805	2
R30	750R	R0805	2
R31	750R	R0805	2
R32	750R	R0805	3
R33	750R	R0805	3
R34	750R	R0805	3
R35	750R	R0805	3
R40	22k	R0805	3
R41	22k	R0805	3
R42	22k	R0805	3
R43	22k	R0805	3
R44	4k7	R0805	3
R45	4k7	R0805	3
R46	4k7	R0805	3
R47	4k7	R0805	3
R48	22k	R0805	2
R49	22k	R0805	2
R50	22k	R0805	2
R51	22k	R0805	2
R52	4k7	R0805	2
R53	4k7	R0805	2
R54	4k7	R0805	2
R55	4k7	R0805	2
SOC1	PRIMY_KONEKTOR_TOP	VANA	1
SOC2	PRIMY_KONEKTOR_BOTTOM	VANA_BOTTOM	1
SOC3		KON_PRIMY_SVORKA	2
SOC4		KON_PRIMY_SVORKA	2

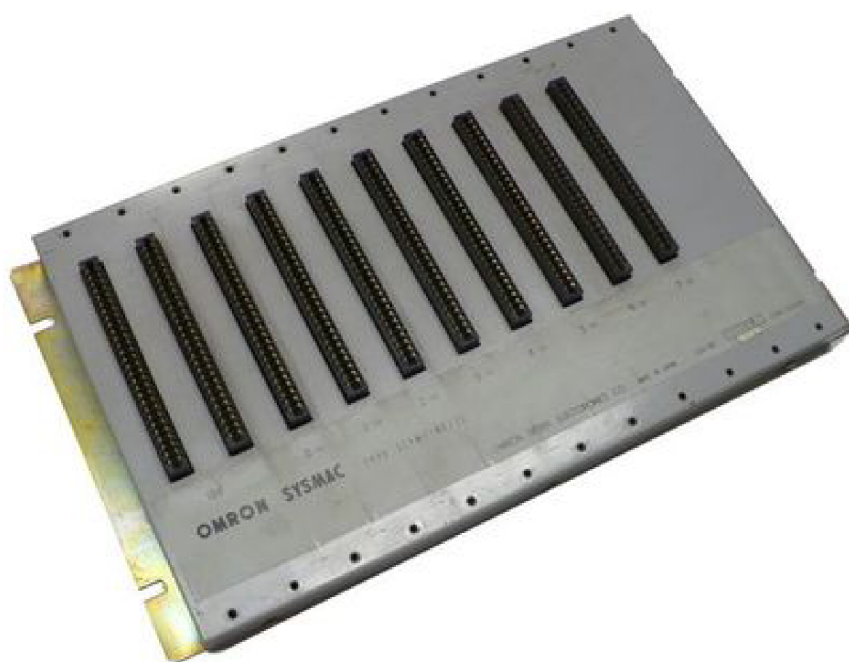
D SESTAVA AUTOMATU

Kompletní sestava automatu je zobrazena namontovaná v rozvaděči na obrázku D.1, přičemž jednotlivé moduly (I/O karty) jsou připevněny na vaně (viz obrázek D.2).

První zleva je nainstalován vytvořený řídicí modul (zde ve starší verzi). Následují dvě originální 8bitové 24VDC DI karty. Třetí a čtvrtý modul obsahuje navrženou 8bitovou 24VDC DO kartu (C), která slouží pro buzení kontrolky a stykačů. Poslední čtyři moduly obsahují originální 110VAC triakovou DO kartu SCYM1-OA201 pro spínání cívek hydraulických ventilů.



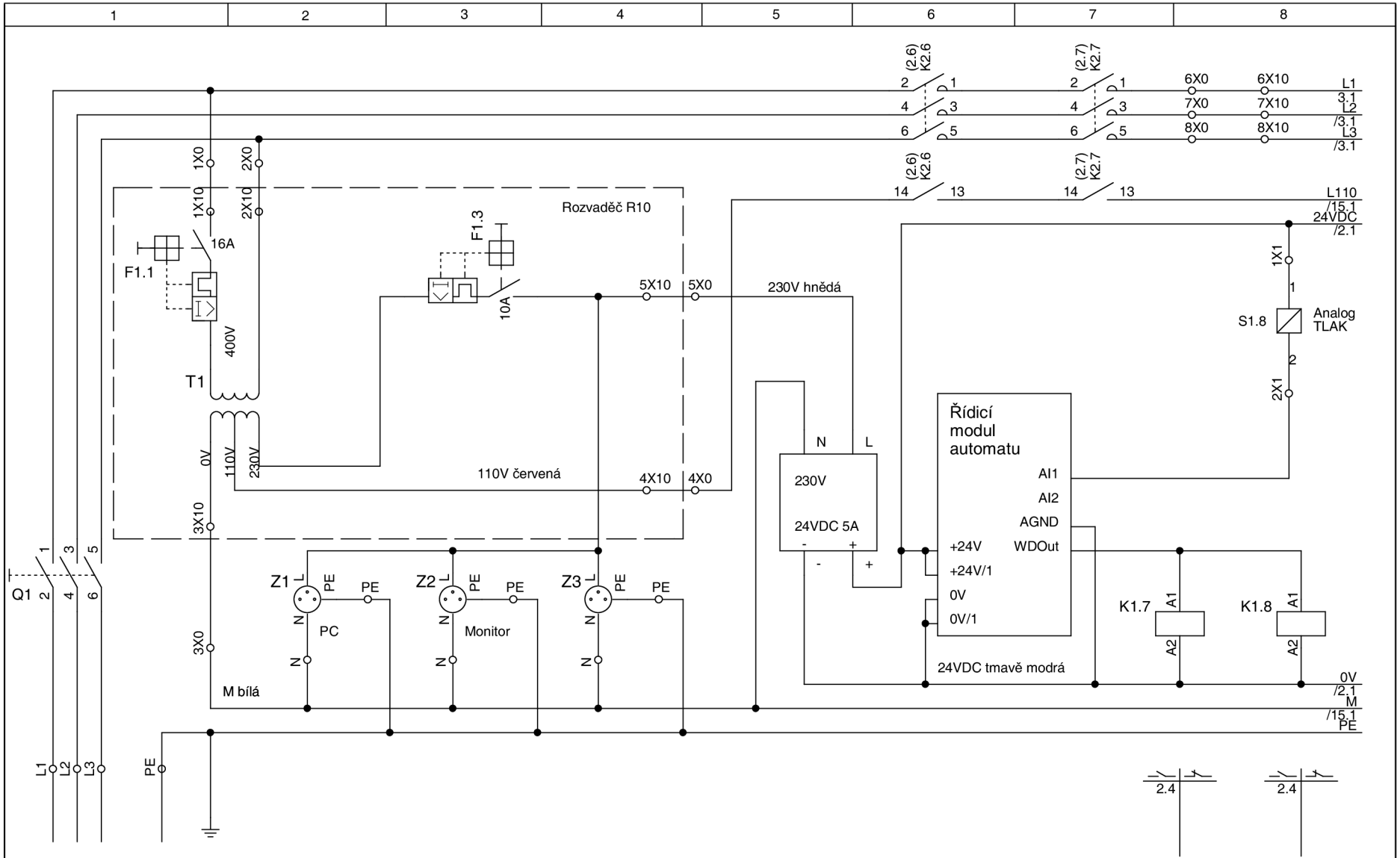
Obr. D.1: Sestava automatu



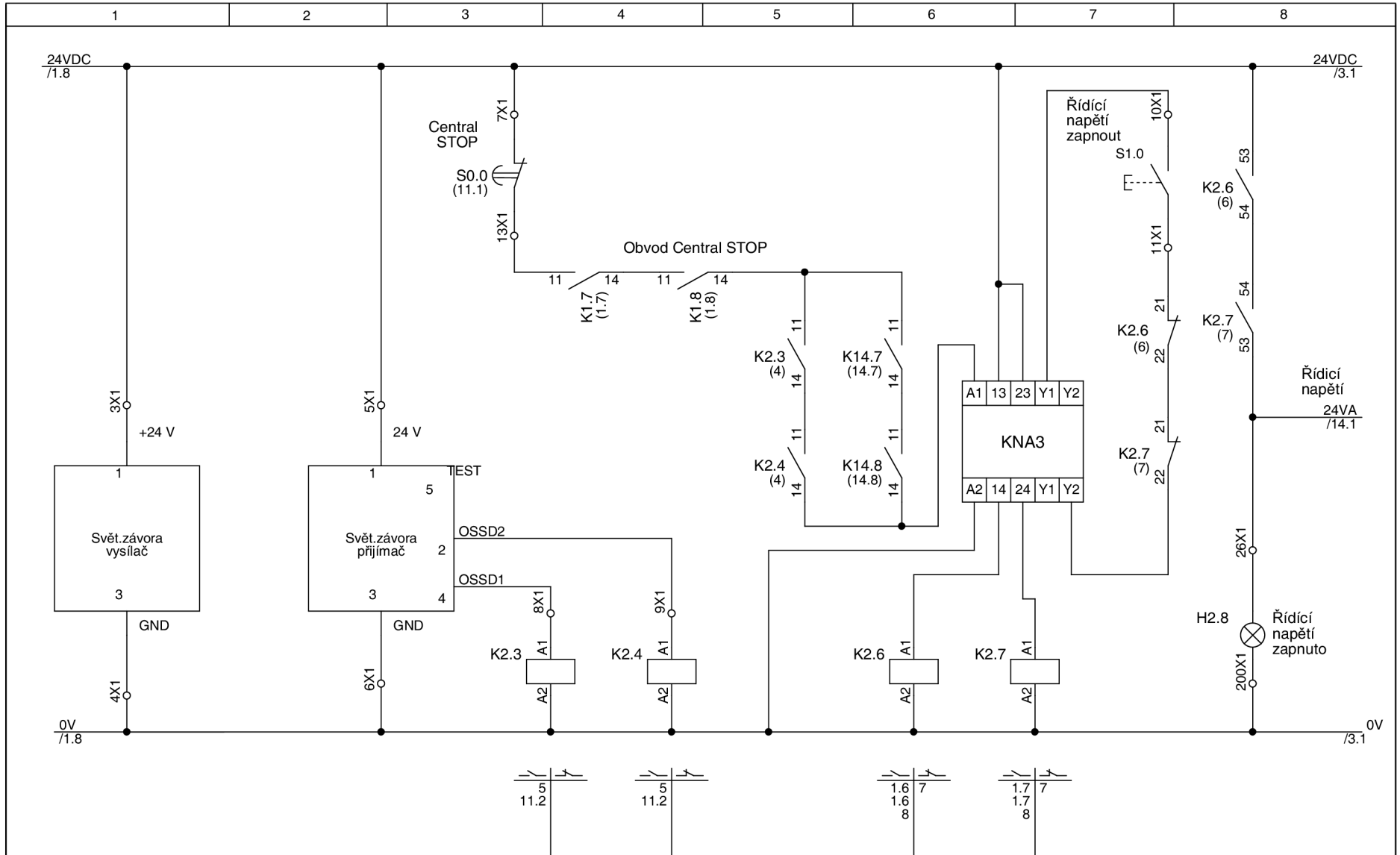
Obr. D.2: Vana pro modulární automat

E ŘÍDICÍ ROZVADĚČ

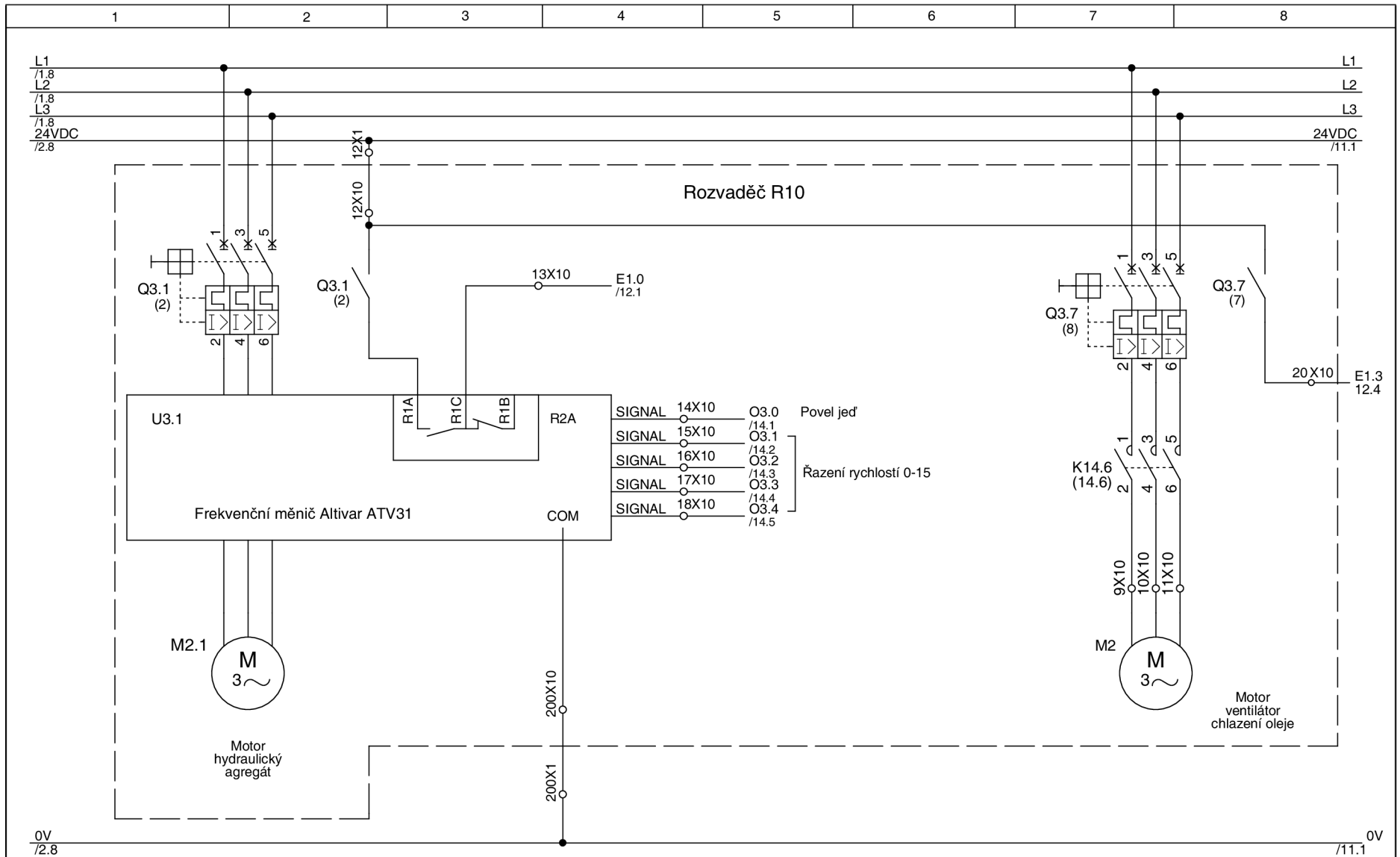
Následující strany obsahují kompletní schéma zapojení řídicího rozvaděče. Fyzicky je zapojení rozděleno do dvou skříní R1 a R10, přičemž prvky umístěné v druhé zmíněné jsou ohraničeny čárkovanou čarou.



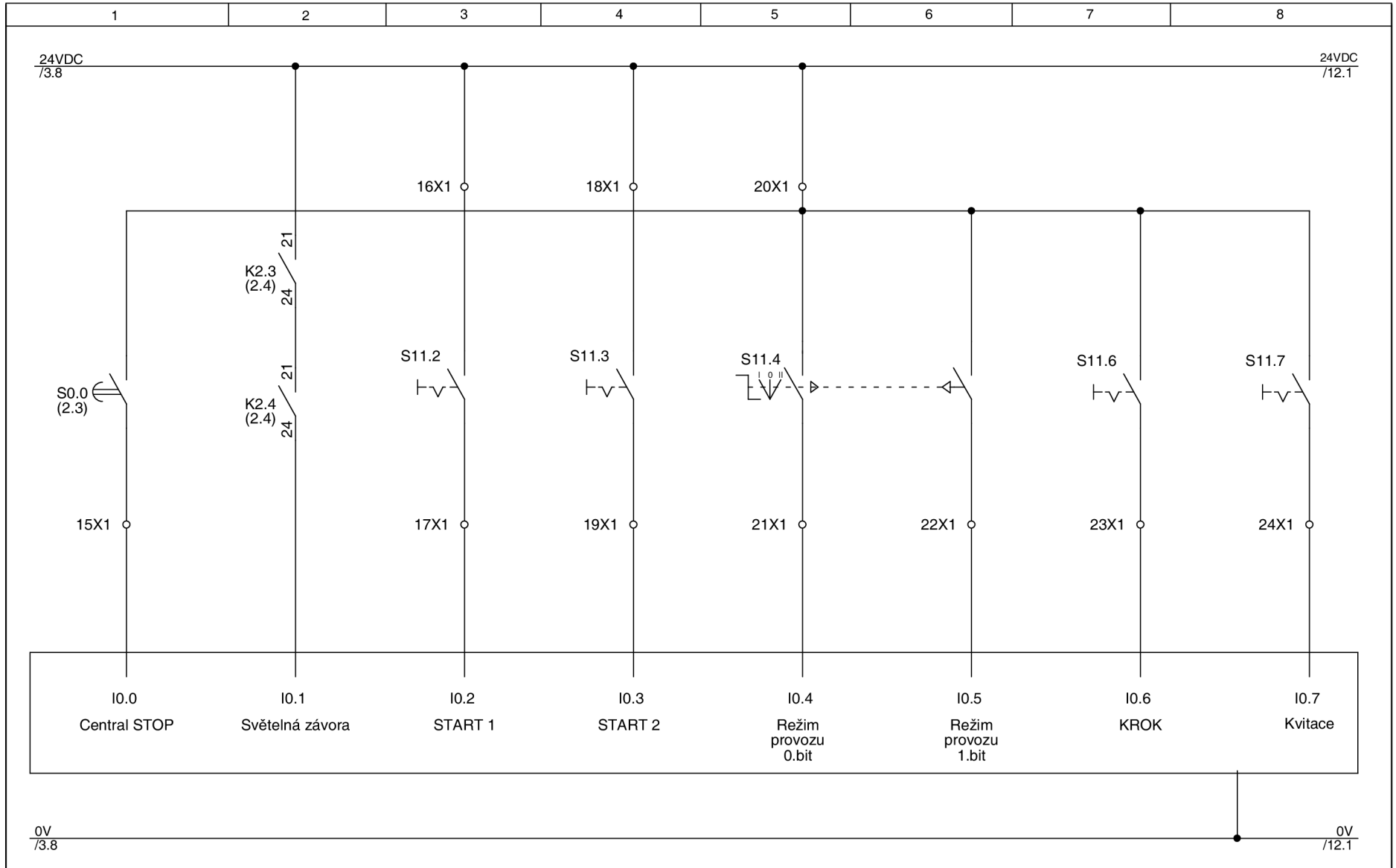
	Investor:	Vypracoval:	Datum: 4.5.2012	◀
	Zakázka:	Schválil:	Změna:	▶ 2
	Projekt: Tvarovací stroj TS-V31	Výkres: Napájení		Celkem 11
				List 1



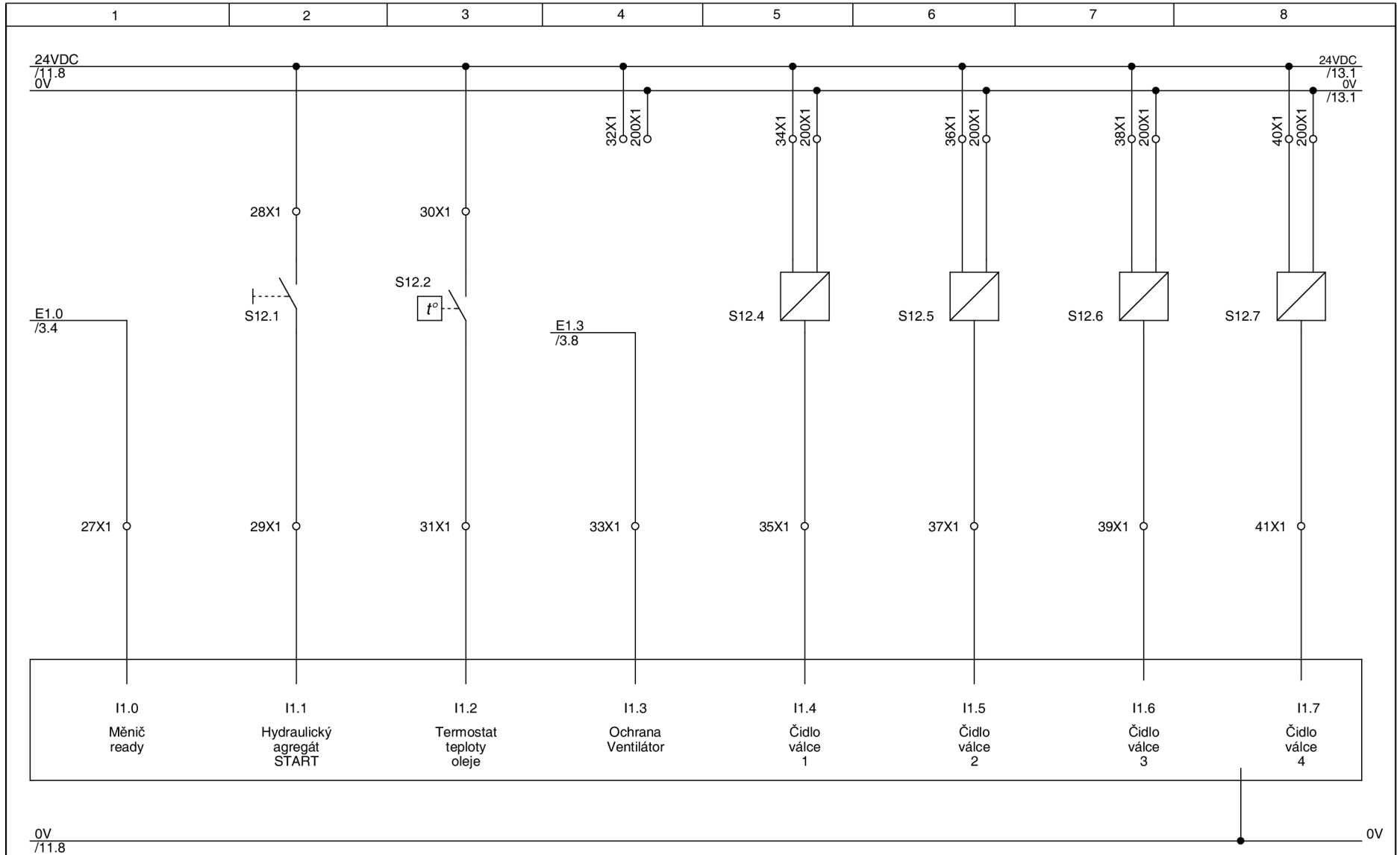
	Investor:	Vypracoval:	Datum: 4.5.2012	◀ 1
	Zakázka:	Schválil:	Změna:	▶ 3
	Projekt:	Výkres:		Celkem 11
	Tvarovací stroj TS-V31			List 2
			Obvod Central STOP	



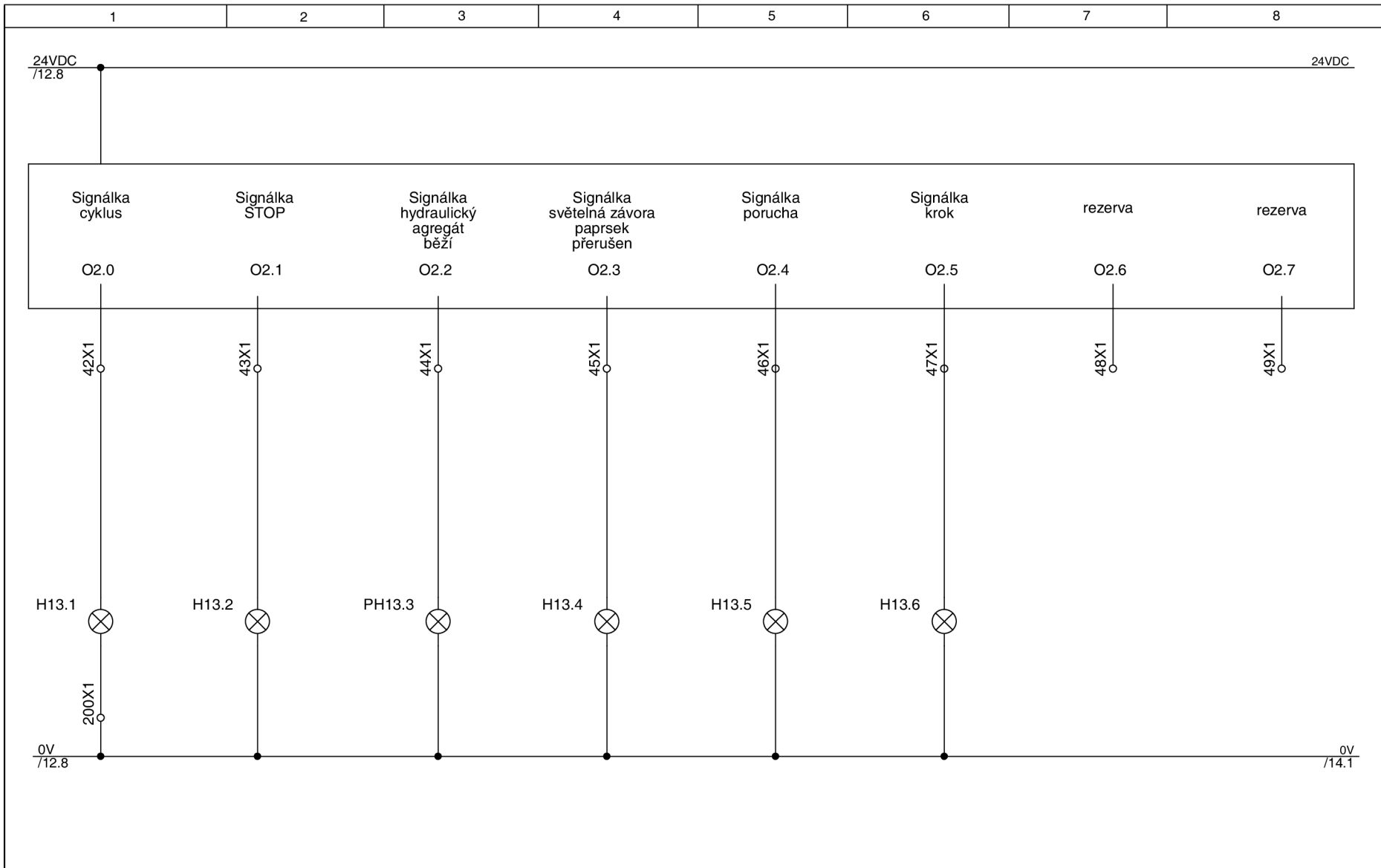
Investor:	Vypracoval:	Datum: 4.5.2012	◀ 2
Zakázka:	Schválil:	Změna:	▶ 11
Projekt:	Výkres:		Celkem 18
Tvarovací stroj TS-V31			List 3
Hydraulický agregát			



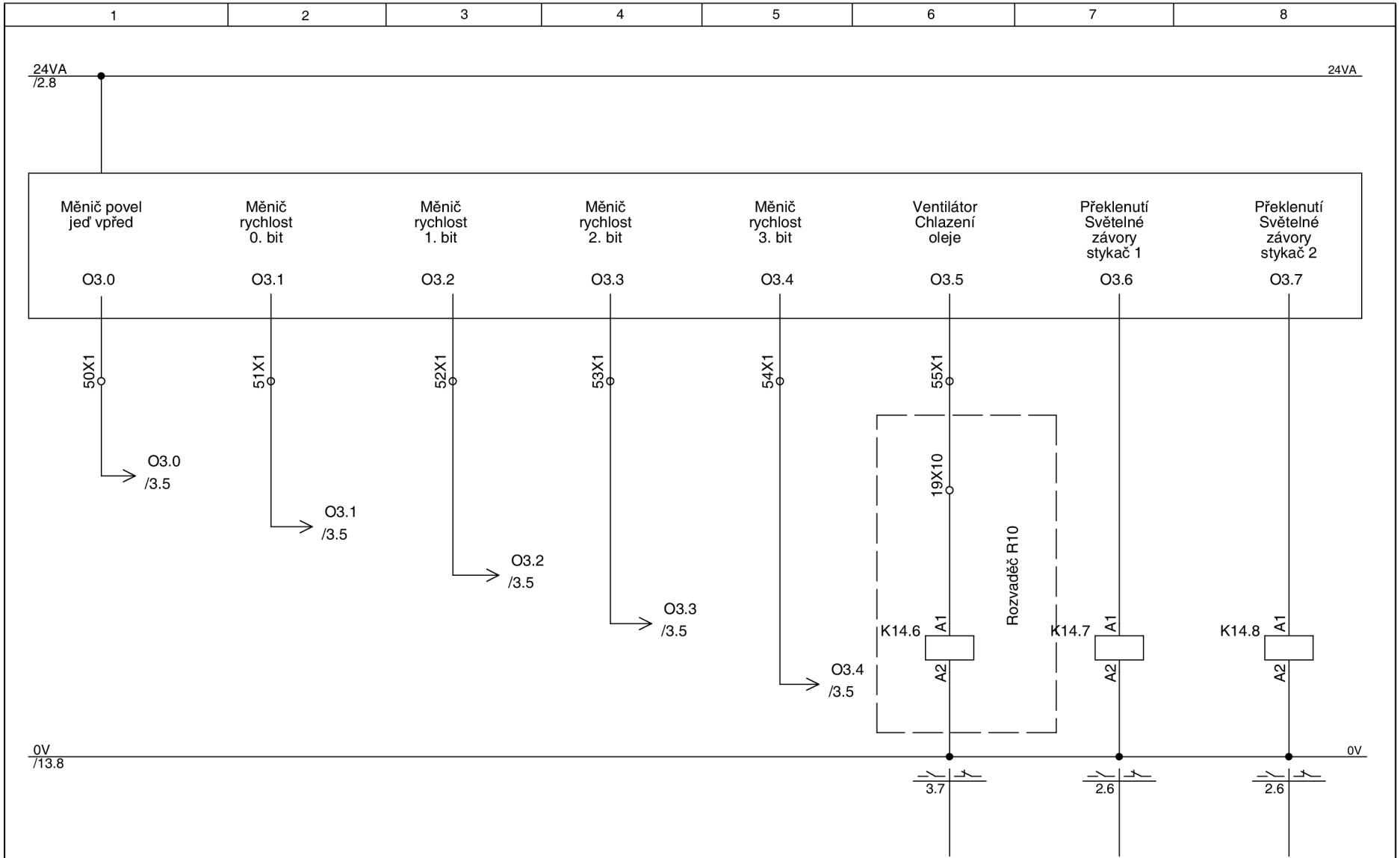
	Investor:	Vypracoval:	Datum: 4.5.2012	◀ 3
	Zakázka:	Schválil:	Změna:	▶ 12
	Projekt:	Výkres:		Celkem 18
	Tvarovací stroj TS-V31			List 11
			Vstupy I0.0-I0.7	



	Investor:	Vypracoval:	Datum: 4.5.2012	◀ 11
	Zakázka:	Schválil:	Změna:	▶ 13
	Projekt:	Výkres:		Celkem 18
	Tvarovací stroj TS-V31			List 12
			Vstupy I1.0-I1.7	

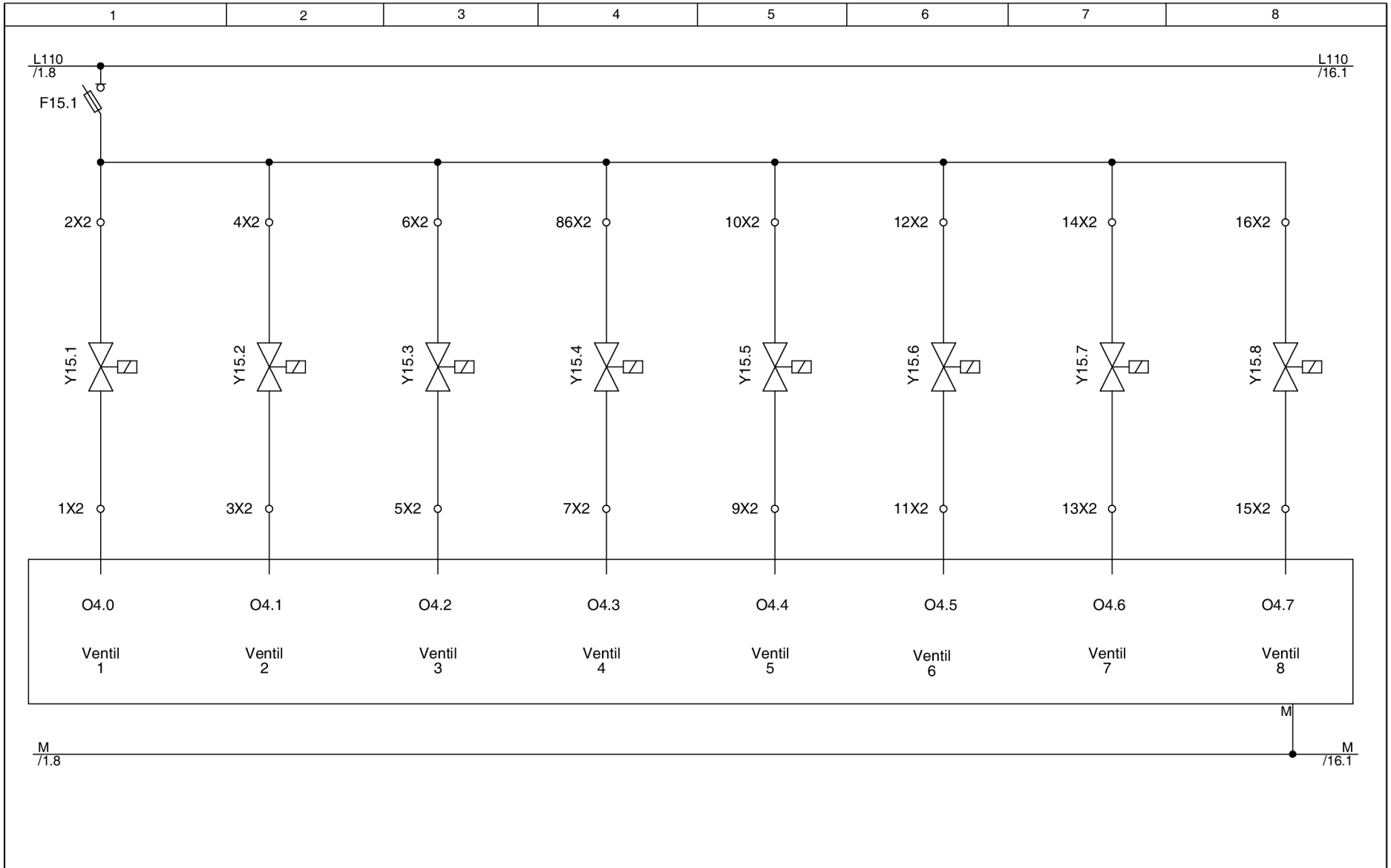


	Investor:	Vypracoval:	Datum: 4.5.2012	◀ 12
	Zakázka:	Schválil:	Změna:	▶ 14
	Projekt:	Výkres:	Celkem 18	
Tvarovací stroj TS-V31		Výstupy O2.0-O2.7		List 13

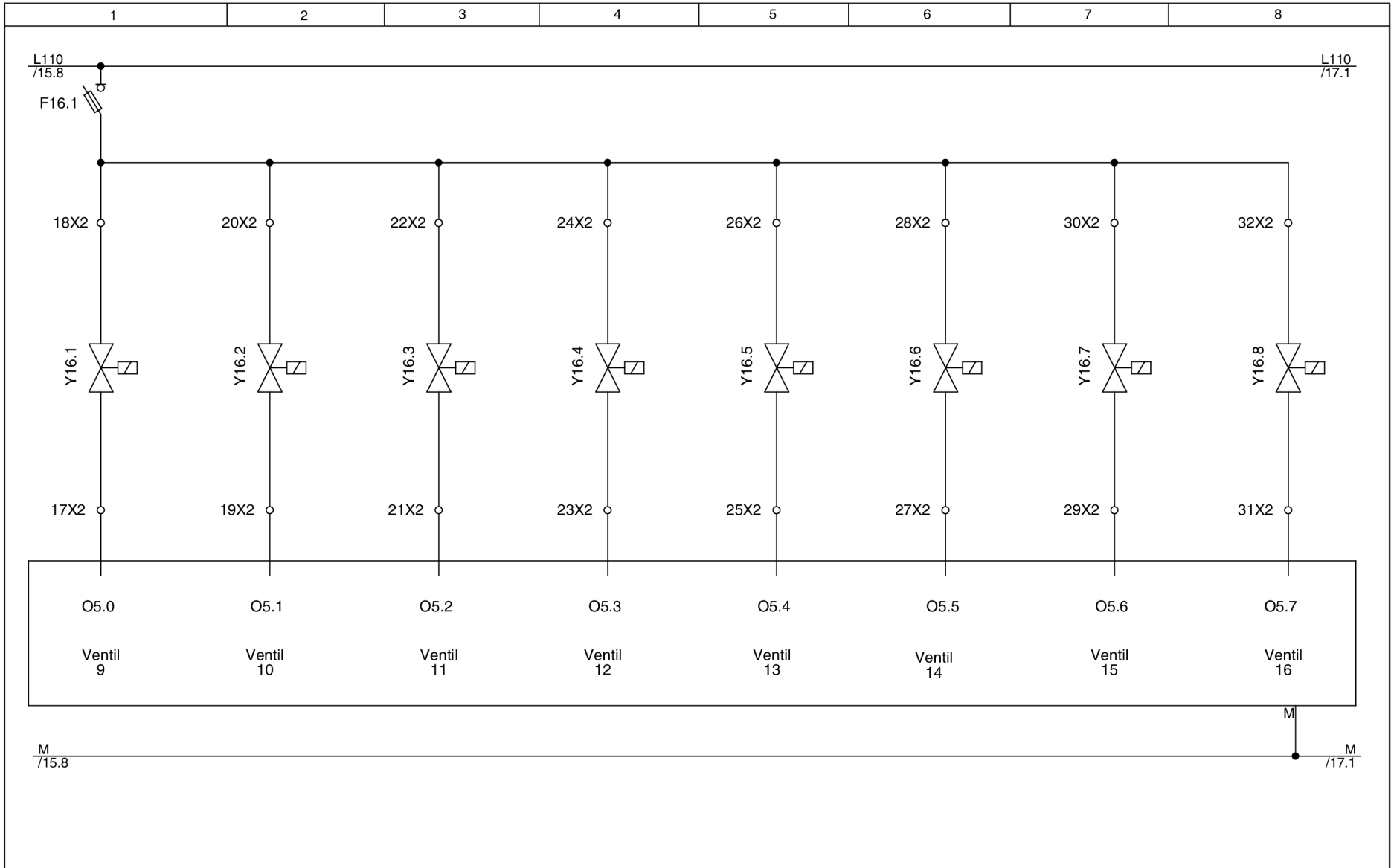


	Investor:	Vypracoval:	Datum: 4.5.2012	◀ 13
	Zakázka:	Schválil:	Změna:	▶ 15
	Projekt:	Výkres:		
	Tvarovací stroj TS-V31	Výstupy O3.0-O3.7		
				Celkem 18
				List 14

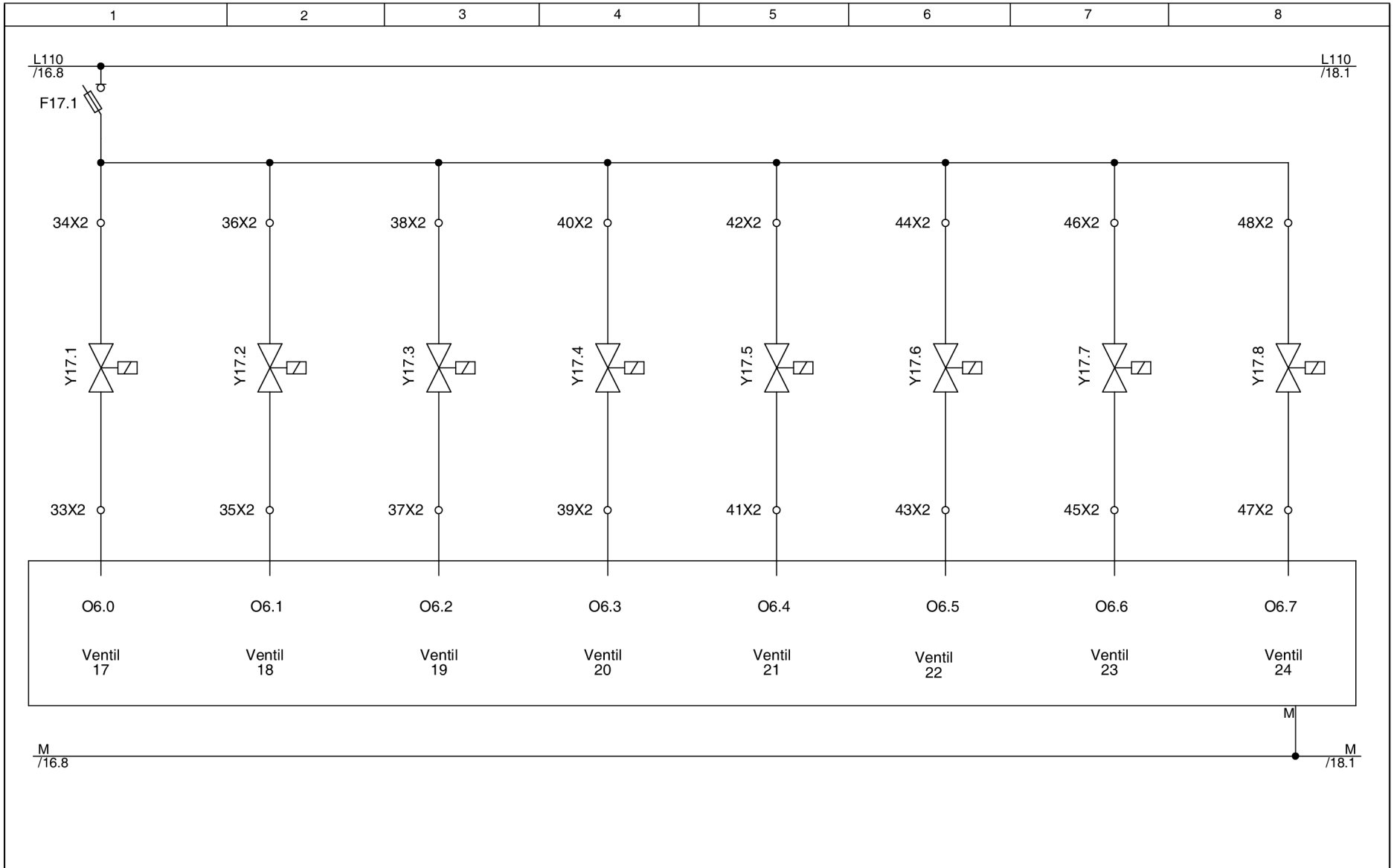
135



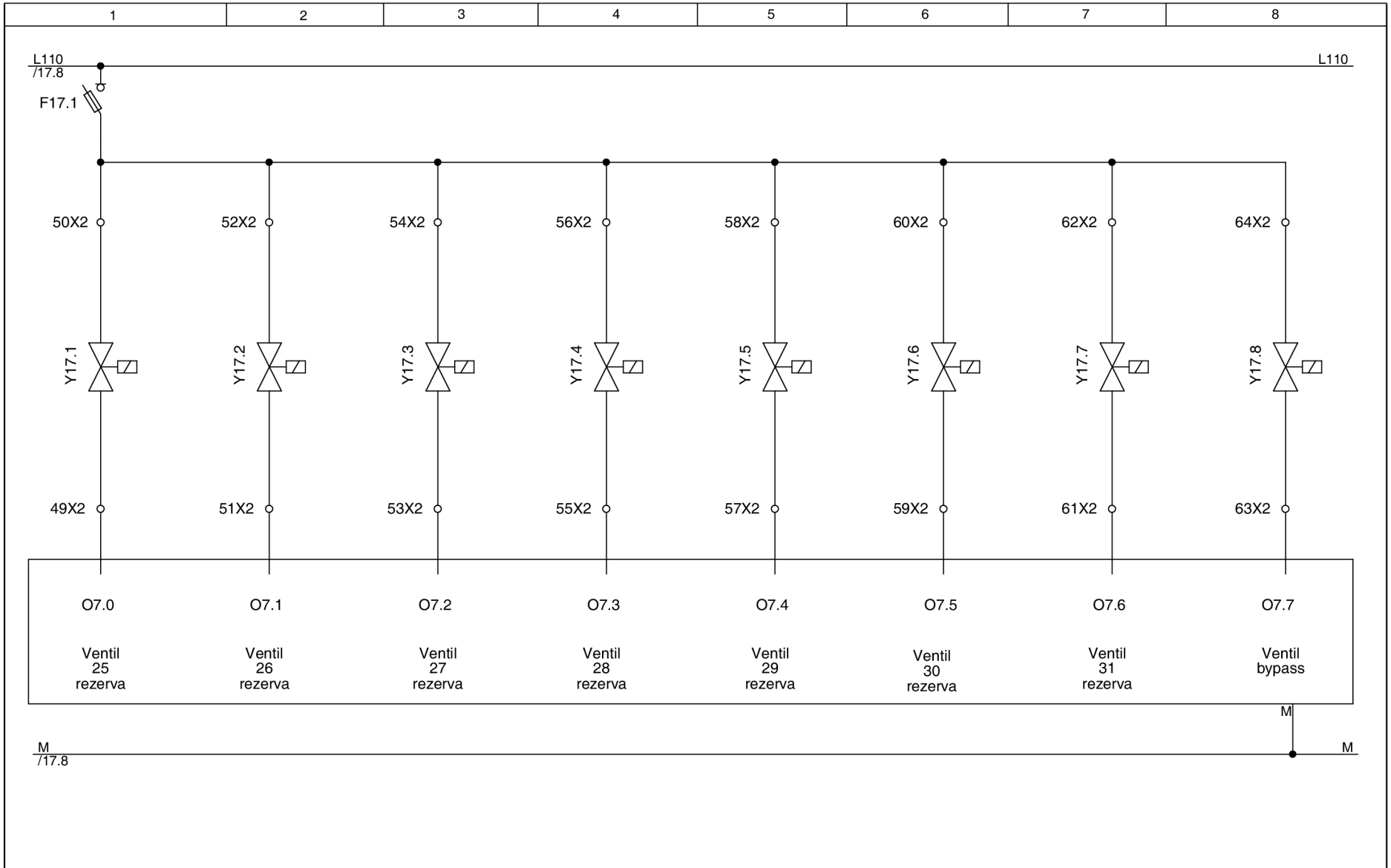
	Investor:	Vypracoval:	Datum: 4.5.2012	◀ 14
	Zakázka:	Schválil:	Změna:	▶ 16
	Projekt:	Výkres:		Celkem 18
	Tvarovací stroj TS-V31		Výstupy O4.0-O4.7	



	Investor:	Vypracoval:	Datum: 4.5.2012	◀ 15
	Zakázka:	Schválil:	Změna:	▶ 17
	Projekt:	Výkres:		Celkem 18
	Tvarovací stroj TS-V31		Výstupy O5.0-O5.7	



	Investor:	Vypracoval:	Datum: 4.5.2012	◀ 16
	Zakázka:	Schválil:	Změna:	▶ 18
	Projekt:	Výkres: Výstupy O6.0-O6.7		Celkem 18
	Tvarovací stroj TS-V31			List 17



	Investor:	Vypracoval:	Datum: 4.5.2012	◀ 17
	Zakázka:	Schválil:	Změna:	▶
	Projekt: Tvarovací stroj TS-V31	Výkres: Výstupy O7.0-O7.7		Celkem 18
				List 18