

**JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH**

**PEDAGOGICKÁ FAKULTA**

**KATEDRA FYZIKY**

# **Diplomová práce**

**Autor:** Bc. Jan Čejka

**Vedoucí práce:** Ing. Michal Šerý

**2010**

**Návrh a realizace řízení  
výkonových prvků  
pomocí počítače**

**Prohlášení:**

Prohlašuji, že jsem tuto práci vypracoval samostatně a že jsem všechny použité zdroje uvedl v seznamu použité literatury na konci této práce. Zároveň v souladu s § 47b zákona č. 111/1998 Sb. v platném znění povoluji Katedře fyziky PF JU v Č. Budějovicích libovolné využití této práce.

V Českých Budějovicích dne 20. 4. 2010

Jan Čejka .....

## **Poděkování**

Především musím poděkovat panu Ing. M. Šerému, který mi umožnil vypracovat tuto práci a ochotně poradil pokaždé, když jsem potřeboval. Dále bych chtěl poděkovat všem vyučujícím, kteří mne alespoň nepřímo poradili a pomohli prohloubit mé znalosti.

## **Anotace**

Práce se zabývá návrhem, konstrukčním řešením a realizací prototypu a projektu. Prototyp přes PC pomocí obvodu ASIX FT232BM ovládá mikrokontrolér ATtiny2313, který umožňuje spínání výkonové části a PWM modulaci výstupů. Projekt popisuje řízení programovatelným terminálem AMIT APT3100(S) propojeným s PC, základní zapojení bezpečnostní smyčky a několika měřících zařízení.

## **Annotation**

My diploma thesis is concerned with the scheme, constructive solution and implementation of the prototype and project. Via PC with the help of a circuit ASIX FT232BM, the prototype operates a ATTiny2313 microcontroller, which enables switching of the power part and PWM modulation outputs. The project describes operating by a programmable terminal AMIT APT3100 (S) connected to a PC, and the wiring of a basic safety-loop and several measuring devices.

# Obsah

ÚVOD .....	9
<b>1 AUTOMATIZACE A ŘÍZENÍ .....</b>	<b>10</b>
1. 1 Historie.....	11
1. 2 Kybernetika.....	11
1. 3 Řízení .....	12
1. 3. 1 Dělení řízení podle zpětné kontroly .....	12
1. 3. 2 Rozdělení podle principu působení řídicího systému na řízený systém .....	13
<b>2 HISTORIE POČÍTAČŮ .....</b>	<b>15</b>
2. 1 Počátky vývoje a výroby počítačů v Československu .....	18
2. 1. 1 Počítače vyvíjené a vyráběné v rámci řady JSEP.....	22
2. 1. 2 Počítače vyvíjené a vyráběné v rámci řady SMEP .....	23
2. 1. 3 Minipočítače řady ADT 4000.....	24
2. 1. 4 Analogové počítače .....	25
2. 1. 5 Současnost .....	25
<b>3 PROGRAMOVATELNÝ LOGICKÝ AUTOMAT .....</b>	<b>26</b>
<b>4 PRVNÍ TESTOVACÍ PROTOTYP .....</b>	<b>29</b>
4. 1 Program v PC .....	29
4. 2 Převodník USB-485, FTDI .....	30
4. 2. 1 FTDI.....	31
4. 2. 2 Modulární řešení .....	31
4. 3 Výkonová část.....	34

<b>5 DRUHÝ PROTOTYP</b> .....	37
5. 1 Program v PC .....	37
5. 2 Výkonová část .....	39
5. 3 Programová část jednočipu .....	42
<b>6 TŘETÍ PROTOTYP</b> .....	45
6. 1 Proč ... .....	45
6. 2 Řídící automaty AMIT .....	46
6. 3 AMIT APT3100(S) .....	46
6. 4 Moduly AMIT .....	47
6. 4. 1 DM-AI12 .....	47
6. 4. 2 DM-DI24 .....	48
6. 4. 3 DM-DO18 .....	48
6. 5 DetStudio .....	49
6. 6 ViewDet .....	51
6. 7 Bezpečnost .....	54
6. 8 Popis zapojení a funkce .....	56
6. 8. 1 Propojení s PC .....	56
6. 8. 2 Čidla a měření .....	58
6. 8. 3 Programování .....	58
6. 8. 4 Ukázková aplikace .....	61
<b>ZÁVĚR</b> .....	62



# Úvod

Počítače jsou nedílnou součástí našeho každodenního života a pronikají do nejrůznějších oblastí lidské činnosti. Právě jednou z oblastí, kde výpočetní technika má stále větší uplatnění je řízení a automatizace. Tato řešení nám slouží k zjednodušení různých procesů a mělo by nám ulehčit práci.

Problematika automatizace není jen o řízení procesu, ale je třeba zabezpečit připojení různých technologických celků k PC. Tyto celky se skládají z řady komponent na různých napěťových úrovních, komunikujících na nejrůznějších rozhraních.

Cílem mé práce bylo vytvořit několik ukázek, jak připojit výkonové prvky k řídicím počítačům. Od jednočipového řízení přes řídicí automat.

# 1 Automatizace a řízení

Automatizace označuje použití řídicích systémů (např. regulátorů, počítačů, snímačů) k řízení průmyslových zařízení a procesů. Z pohledu industrializace jde o krok následující po mechanizaci. Zatímco mechanizace poskytuje lidem k práci zařízení, které jim usnadňuje práci, automatizace snižuje potřebu přítomnosti člověka při vykonávání určité činnosti. Za splnění ideálního předpokladu tzv. komplexní automatizace by teoreticky mohlo dojít až vyřazení člověka z příslušného výrobního procesu. V praxi se prozatím jeví tato možnost jako neuskutečnitelná. [1]

Ve smyslu industrializace je to krok za mechanizací. Zatímco v mechanizaci byl jediným operátorem člověk, automatizace velmi redukuje lidskou duševní a smyslovou přítomnost.

## 1. 1 Historie

K automatizaci vede snaha člověka osvobodit se nejen od fyzické činnosti, ale i od jednotvárné a unavující činnosti duševní. Nejprve byly stroje jednoduché přístroje, které nahradily jeden způsob činnosti, např. systém kladek pro ulehčení zvedání břemen. Pozdější stroje byly schopné nahradit přirozeně používané obnovitelné energie jako jsou vítr, příliv, tekoucí voda. [3]

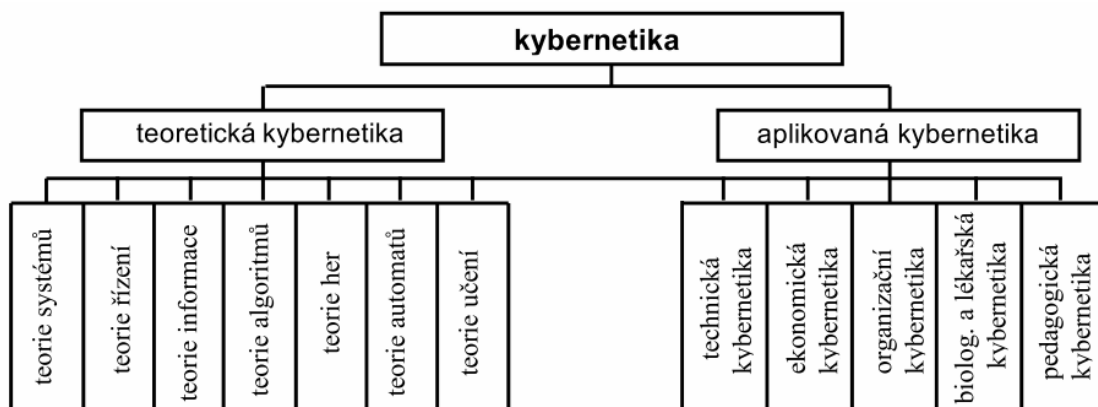
Později pak, s dalším rozvojem techniky a nárůstem nároků na řídicí činnost, přistoupil i k osvobození od často již i velmi náročné a rovněž namáhavé řídicí duševní práce (automatizace – např. přechod ze strojního obrábění s lidskou obsluhou na číslicově řízené obráběcí stroje). Postupně jsou tak vytvářeny řídicí systémy buď plně automatické (bez jakékoliv účasti člověka na řízení), nebo více či méně automatizované, kde člověk do jinak automaticky řízeného procesu zasahuje způsobem, který je spíše závislý na charakteru

řízeného procesu (např. volí nebo potvrzuje další uplatňovaný způsob řízení, modifikuje způsob řízení podle okamžitého průběhu řízeného procesu apod.). [2]

## 1. 2 Kybernetika

Řízení je tedy neoddělitelným základem automatizace. A teoretickou disciplínou, která se zabývá řízením, je vědní obor zvaný kybernetika. Za jejího zakladatele je považován americký matematik Norbert Wiener, který jako první zpracoval teorii zpětnovazebních systémů řízení pro účely protiletectvé obrany. Tuto teorii zobecnil pro všechny druhy technických a biologických systémů. Shrнул ji ve své proslulé knize Kybernetika neboli řízení a sdělování v živých organismech a strojích (Cybernetics or Control and Communication in the Animal and the Machines). Tato kniha vyšla v roce 1948 a autora proslavila jako zakladatele kybernetiky. Většina definic kybernetiky vychází z Wienerovy definice, ten ji definoval jako „vědu o řízení a sdělování v živých organismech a strojích“. Nedostatkem této definice je, že nedoceňuje systémový přístup při řízení a jako objekty zkoumání zahrnuje pouze živé organismy a stroje. Nezahrnuje tedy další poměrně velmi důležité objekty, z dnešního hlediska nabývající ještě na důležitosti, zkoumané dnešní kybernetikou, jako jsou objekty společenské, ekonomické a z technických systémů, dnes tak různorodých, se omezuje jen na stroje. Ani informační hledisko této definice není úplné, protože se omezuje jen na sdělování čili na přenos informací a neuvažuje dnes tak důležité procesy uchování a zpracování informace. [2]

Kybernetika je věda, která zkoumá obecné vlastnosti a zákonitosti řízení v biologických, technických a společenských systémech. Jako každá věda musí také kybernetika disponovat teoretickým základem a tento aplikovat na jednotlivé vědní oblasti. Tímto rozlišením dělíme kybernetiku na teoretickou a aplikovanou kybernetiku – obr. 1.1. [2]



Obr. 1.1 Rozdělení kybernetiky na základní oblasti [2]

### 1. 3 Řízení

Z dílčích částí teoretické kybernetiky nás bude v dalším zajímat především teorie řízení, která se zabývá zkoumáním obecných vlastností a zákonitostí řízení, které je základem automatizace. Řízení je cílené působení na řízený objekt tak, aby se dosáhlo určitého předepsaného cíle. [2]

Podle toho, jak řízení provádíme, rozlišujeme řízení ruční a automatické. U automatického řízení rozlišujeme přímé řízení, u kterého řídicí proces probíhá bez přívodu energie (regulace výšky hladiny odvozená od síly plováku) a nepřímé řízení s přívodem energie, což je dnes běžné.

Důležitým hlediskem pro dělení řízení je, zda výsledek řízení je anebo není zpětně kontrolován, zda je či není zpětná vazba při řízení. Podle toho rozlišujeme ovládání, regulaci a vyšší formy řízení. [2]

#### 1. 3. 1 Dělení řízení podle zpětné kontroly

**Ovládání** je řízení bez zpětné kontroly – bez zpětné vazby

**Regulace** je řízení se zpětnou vazbou. Regulace je udržování určité fyzikální veličiny na konstantní hodnotě nebo jinak podle nějakého pravidla se

měnící hodnotě. Během regulace se zjišťují hodnoty této veličiny a srovnávají se s hodnotou, kterou má mít. Podle zjištěných odchylek se zasahuje do regulačního procesu v tom smyslu, aby se odchylky odstranily. [2]

### Vyšší formy řízení

**optimální řízení** je takové, kdy systém dosáhne požadovaných vlastností např. při minimu vynaložené energie, tedy s maximální účinností, nebo naopak v nejkratším čase. Systém je přitom schopen vyhledat nejvýhodnější působení a dosáhnout tak co nejlepšího chování celého systému v daných omezujících podmínkách. [2]

**adaptivní řízení** je takové, kdy systém je schopen měnit svou strukturu tedy i své para-metry tak, aby proces řízení probíhal stále optimálně, a to i při změnách parametrů řízeného objektu. [2]

Řízení s umělou inteligencí je nejvyšším stupněm řízení. Umělá inteligence je vlastnost uměle vytvořeného systému, který má schopnost rozpoznávat předměty, jevy, analyzovat vztahy mezi nimi a tak si vytvářet modely okolí, dělat účelná rozhodnutí a předvídat jejich důsledky, řešit problémy včetně objevování nových zákonitostí a zdokonalování své činnosti. [2]

### **1. 3. 2 Rozdělení podle principu působení řídicího systému na řízený systém**

**Logické řízení** využívá k řízení dvouhodnotových veličin. Jejich působení je takové, že jsou vždy jen dvě možnosti – ventil je otevřen / zavřen, vypínač je sepnut / vypnut, atd. Podobně i informace o stavu objektu jsou dvouhodnotové veličiny – hladina je nad / pod minimální hodnotou, teplota je nad / pod 18°C, atd. Dvouhodnotové veličiny jsou formálně vyjadřovány hodnotami 0 a 1. Jsou analogické s proměnnými výrokové logiky, a proto jsou vztahy mezi proměnnými nazývány logické funkce a řídicí obvody pracující na tomto principu jsou logické řídicí obvody. [2]

**Spojité řízení** je tam, kde jak akční zásah je spojitě nastavován, tak i údaje o řízeném systému jsou měřeny jako veličiny spojitě proměnné v čase. Spojitý řídicí systém vytváří (na rozdíl od diskrétního systému) nepřetržitou vazbu mezi vstupy a výstupy. Všechny veličiny spojitého systému jsou spojitě proměnné v čase, žádná z nich není ani dvouhodnotová ani diskrétní. [2]

**Diskrétní řízení** je dnes důsledkem nasazení počítačů jako regulátorů i když jeho počátky byly při řízení spojitých systémů, diskrétně měřených (řízení polohy letadla, měřené radiolokátorem). U řídicích počítačů, které ani nedovedou zpracovávat spojitý signál, je nutný spojitý signál převádět na diskrétní. Diskrétní řídicí systém vytváří vztah mezi vstupy a výstupy jako vztah mezi posloupnostmi impulsů, snímaných v časovém sledu daném tzv. vzorkovací periodou. Mezi okamžiky vzorkování není regulovaná veličina měřena a ani akční veličina není upravována. Tato vzorkovací perioda je tím kratší, čím rychlejší je řízený proces. [2]

U **fuzzy řízení** není základem řízený systém a jeho model, ale pozornost je zaměřena na člověka (tzv.experta), který umí systém řídit, ale přitom nemá pojem o klasickém matematickém modelu řízeného systému. Takový člověk pak soustavu řídí na základě pravidel typu „jestliže hladina klesá, otevři trochu přívod vody“. [2]

Fuzzy regulátor musí nejprve přiřadit zvoleným vstupním veličinám jazykovou hodnotu. To se provede nejlépe pomocí tzv. funkce příslušnosti – bývají voleny obvykle ve tvaru lichoběžníka či trojúhelníka. Tato etapa je označována jako fuzzifikace. V dalším kroku určí fuzzy regulátor na základě znalostí experta slovní hodnoty akčních veličin (např. regulační odchylka je záporná malá). Nakonec převede slovní vyjádření na konkrétní číselné hodnoty veličin – tzv. defuzzifikaci.

Toto řízení je vhodné pro řízení systémů, které nedovedeme popsat, ale které dovedeme řídit. Je možné určit hodnotu výstupu, aniž známe vzorce mezi vstupem a výstupem. [2]

## 2 Historie počítačů

První předchůdce dnešních počítačů byl založen na mechanických principech. Před přibližně 5000 lety vznikl stroj Abakus, který usnadňoval počítání s čísly. Původně šlo o zaprášený kámen (starohebrejsky abaq = prach). Nejstarším dochovaným exemplářem je salaminská tabulka (cca 300 př.n.l.), ale o více než století dříve popsal příklady tabulky tohoto typu Hérodotos. Ve starověkém Řecku a Římě se používaly dřevěné nebo hliněné destičky, do které se vkládaly kamínky (tzv. calculi). [1]

Roku 1614 vymyslel John Naper novou matematickou metodu, která umožňovala pomocí logaritmů sčítat, odčítat, násobit a dělit. Poté byly v Anglii sestaveny první logaritmické tabulky a následovalo logaritmické pravítko, které se k provádění výpočtů používalo další dvě století. [1]

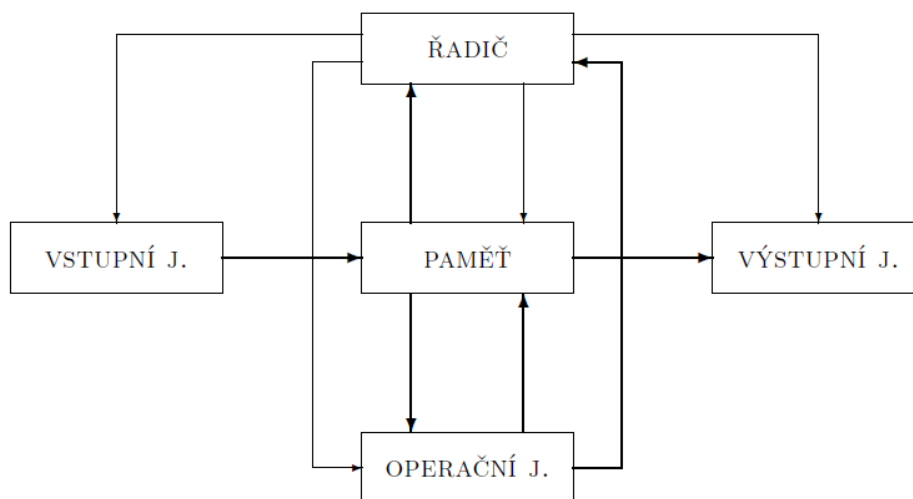
Mechanický kalkulátor (obr. 2.1) sestavil Wilhelm Schickard roku 1623 z ozubených koleček z hodinových strojků, uměl sčítat a odčítat šestimístná čísla. Údajně je používal Johann Keller při astronomických výpočtech. Kolem roku 1820 vytvořil Charles Xaver Thomas první sériově vyráběný kalkulátor, který byl schopen, sčítat, odčítat, násobit i dělit. Mechanické kalkulátory se hojně používali až do 70. let 20. století, většina byla založena na desítkové soustavě. [1]



Obr. 2.1: Replika prvního mechanického počítačového stroje [1]

Za další vývojový stupeň výpočetní techniky lze považovat vynález děrných štítků. V roce 1725 použil Basile Bouchon děrovaný papír pro řízení tkacího stroje. V roce 1801 Joseph Marie Jacquard použil v tkacím stroji děrné štítky, které bylo možné vyměnit beze změny v mechanice samotného stavu. Tento okamžik je považován za počátek v programování strojů. Herman Hollerith použil děrné štítky k uchovávání dat, pomocí nich se v roce 1890 uskutečnilo v USA sčítání lidu. Jeho firma se později stala základem pro IBM.

Skutečný rozvoj výpočetní techniky začal v první polovině dvacátého století. Začaly se formulovat dnes již notoricky známé pojmy (bit, byte, výpočetní soustavy, logické operátory atd.), především však vznikla Von-Neumannova koncepce počítače (obr. 2.2), podle které byly počítače konstruovány od počátku až do nedávné doby. [5]



Obr. 2.2: Von-Neumannovská koncepce počítače. Tok dat je znázorněn silnými čarami, řídicí příkazy čarami slabými. [5]



Generace	Rok	Konstrukční prvky	Výkon [op/s]	RAM [b]	SW
0	1941	relé	$10^1$	$10^3$	stroj. kód
1	1946	elektronky	$10^4$	$10^4$	assembler
2	1960	polovodiče	$10^5$	$10^5$	FORTTRAN
2,5	1964	mikromoduly	$10^6$	$10^6$	op. systémy
3	1966	int. obvody	$10^7$	$10^7$	sdílení času
3,5	1971	int. obvody	$10^8$	$10^8$	virt. paměti

Tab. 2.1: Hlavní charakteristiky výpočetní techniky [5]

## 2. 1 Počátky vývoje a výroby počítačů v Československu [6]

Prvním počítačem vyvinutým a vyrobeným v Československu byl počítač SAPO (Samočinný Počítač), který byl uveden do provozu v roce 1957. Vývoj počítače začal v roce 1950 ve Výzkumném ústavu matematických strojů (VÚMS) pod vedením profesora A. Svobody a trval celých sedm let než se jej podařilo uvést do zkušebního provozu. Dlouhá doba vývoje byla hlavně způsobena vysokou poruchovostí, nedostatkem zkušených pracovníků a neplněním dodávek potřebných komponent a součástí, což bylo v té době celostátním problémem.

Počítač SAPO v oblasti technologické základny obsahoval 7 000 relé a 400 elektronek a byl vybaven magnetickou bubnovou pamětí o kapacitě 1024 dvaatřiceti bitových slov. Pracoval v dvojkové soustavě s pohyblivou řádovou čárkou.

Tento počítač měl dvě podstatné zvláštnosti. Za prvé byl pěti adresový, neboli součástí každé instrukce bylo pět adres (2 operandy, výsledek a adresy skoků v případě kladného a záporného výsledku). Druhou zvláštností bylo to, že se vlastně jednalo o tři shodné počítače, které pracovaly paralelně. Výsledek každé operace z jednotlivých počítačů se mezi sebou porovnával a o konečném výsledku se rozhodlo hlasováním. Pokud byl výsledek shodný alespoň u dvou počítačů, byl považován za správný. Pokud se ve všech třech případech lišil operace se opakovala.

Počítač byl instalován v objektu VÚMS na Loretánském náměstí. Tři roky po jeho instalaci v roce 1960 vyhořel. Požár vznikl od jiskřících reléových kontaktů v důsledku vznětí oleje, kterým se relé promazávala.

Vývoj a výroba počítače SAPO přinesla mnoho zkušeností, které při nedostupnosti zahraničních informací byly hlavním přínosem pro vývoj dalších počítačů.

Souběžně s počítačem SAPO byly ve VÚMS vyvíjeny další reléové počítače. Na základě žádosti Fyzikálního ústavu ČSAV byl vyvinut jednoúčelový počítač s názvem M1 pro výpočet molekul.

Vývoj dalšího počítače pod označením E1 byl dokončen v roce 1961. Tento počítač byl řízen děrnou páskou, měl bubnovou paměť s kapacitou 1000 slov a desítkové zobrazení čísel. Do provozu byl uveden v roce 1962.

Dalším krokem v naší počítačové historii byl experimentální počítač s označením MP10. Tento počítač měl ještě řídicí logiku na reléové platformě, ale v sériové aritmetické jednotce již byly použity první polovodiče, a to diody a feritová jádra.

Všechny tyto počítače byly elektromechanické. Jejich základním prvkem byly elektromagnetické přepínače (relé), které pracovaly jako přepínače v klopném obvodu. Každý obvod s tímto přepínačem představoval jeden bit. Bylo-li relé sepnuto protékal elektrický proud a obvod představoval bit s hodnotou „1“. Nesepnutý pak představoval bit s hodnotou „0“. Každý z těchto počítačů měl svůj vlastní program ve strojovém kódu uloženém na děrné pásce nebo otáčejícím se magnetickém bubnu. Jejich programování bylo velmi obtížné a pohybovalo se v rozmezí několika hodin až týdnů. V činnosti je musel udržovat rozsáhlý tým odborníků a používat jej mohl vždy pouze jeden uživatel. Vzhledem ke svým rozměrům a elektrickému příkonu bylo možné provozovat jej pouze ve velkých sálech s výkonnou klimatizací.

Jako operační nebo vnější paměť používaly tyto počítače bubnovou magnetickou paměť. Tato paměť byla tvořena rotujícím bubnem z nemagnetického materiálu a řadou čtecích a zapisovacích elektromagnetických hlav umístěných v malé vzdálenosti od povrchu bubnu. Počet hlav odpovídal délce slova procesoru, zároveň se zapisovalo a četlo celé slovo. Na ose rotujícího bubnu byla připevněna kruhová clona s otvory, která sloužila k adresaci (každá kombinace otvorů odpovídala jedné adrese paměti). Kapacita této paměti dosahovala několika kilo bite a průměrná přístupová doba k adrese byla rovna době poloviny otáčky bubnu.

Bohužel všechny výše uvedené počítače trpěly podobnými neduhy jako SAPO – vysokou poruchovostí a technickou zaostalostí, jejíž hlavní příčinou byla nedostupnost kvalitních součástek v důsledku stávajícího rozdělení světa.

Dalším úspěšně dokončeným počítačem byl EPOS1 – Elektronkový počítač střední. Vlastní počítač měl celou řadu unikátních prvků. Například byl čistě dekadický (12 dekadických čísel na slovo). Multiprogramování bylo řešeno výhradně hardwarovými prostředky a měl i důkladné zajištění proti chybovosti.

Po ukončení vývoje počítače EPOS1 zahájil v roce 1963 VÚMS vývoj počítače EPOS2. Jeho vývoj byl však nepříznivě ovlivněn emigrací profesora A.Svobody s řadou spolupracovníků do USA. EPOS2 navázal na svého předchůdce EPOS1, byl však již plně osazen polovodičovými součástkami (diodami a tranzistory). Kromě výměny součástkové základny byly provedeny rozsáhlé změny v jeho logice a zcela přepracováno programové vybavení. Mimo dalších unikátních řešení byly u EPOSu2 nejzajímavější operace maskování, které umožňovaly programovat jednu instrukci i velmi neobvyklé operace. Rychlost počítače byla 40 000 operací za sekundu. Počítač EPOS2 byl po řadě nezdarů jedním z poměrně úspěšných počítačů.

Po dalších modifikacích se tento počítač dočkal sériové výroby pod označením ZPA 600. Jeho výroba byla zahájena v roce 1969 v Závodech průmyslové automatizace (ZPA) Čakovice. Díky orientaci na národní součástkovou základnu našel uplatnění i v armádě.

Vzhledem k tomu, že se prakticky jedná o první sériově vyráběný počítač u nás, uvádím alespoň některé jeho hlavní parametry:

- Dekadický počítač bez možnosti ukládat binární data
- Pevná a pohyblivá řádová čárka
- Aritmetická jednotka jako přídavné zařízení (sčítání a odečítání v pevné řádové čárce aritmetickou jednotku nepotřebuje)
- Feritová paměť až 40 000 slov (po 12 dekadických číslicích), 9 registrů a střídač

- Jedna úplná adresa a až pět polovičních (registr operandu, registr uložení výsledku, maskovací registr, index registr a pokračovací registr)
- Hardwarově řešení mikroprogramování, až pět souběžných programů
- Část paměti zamykatelná nebo možnost přidání pevné paměti
- Osmistopé magnetické pásky, později i disková paměť
- Speciální alfanumerický kód, úplná česká a slovenská abeceda
- Periferie: snímače a děrovače pásky, snímače děrovače štítků (80 nebo 90 sloupcových), psací stroje, řádkové tiskárny, souřadnicový zapisovač (DIGIGRAF)
- Software: základní operační systém (orientovaný na magnetické pásky, později i na disk), programovací jazyk Assembler, Fortran, Cobol, RPG a uživatelské autokódy.

Souběžně s počítačem EPOS byl ve VÚMS v součinnosti s podnikem Aritma vyvíjen univerzální malý počítač DP 100. Konceptně byl zaměřen jako řídicí prvek děroštitkových výpočetních soustav, použitých hlavně ke zpracování hromadných dat. Poučení z předchozích neúspěšných projektů a také díky zkušenostem s výrobou děroštitkových strojů v Aritmě se vývojáři především zaměřili na jednoduchost a spolehlivost počítače.

Pro vstup sloužil elektromagnetický snímač děrných štítků. Štítek procházel lamelami. Při průchodu dírkou lamela propadla otvorem a způsobila elektrický kontakt. Jako mezipaměť sloužil děrovač děrných štítků. Po načtení vstupních dat a programu se po zpracování vyděrovaly mezivýsledky, které se znovu načetly na vstup a tak to pokračovalo až do konečného výsledku, který se vytiskl na tiskárně. Jednoduchost a spolehlivost počítače se ukázaly jako správné předpoklady a od roku 1967 po zavedení sériové výroby bylo vyrobeno a prodáno v průběhu deseti let přibližně 200 souprav počítačů DP 100.

Počítač EPOS2 (ZPA 600) a DP 100 je dovršením vývoje výpočetní techniky v ČSSR, který sledoval vlastní cestu. V další etapě využívání výpočetní techniky se již uplatňovala koncepce masového dovozu výpočetní techniky ze

zemí východního bloku (např. MINSK 1, MINSK 22 ze SSSR, ODRA z Polska a Z-23 z tehdejší NDR) a ve výjimečných případech pro specifická pracoviště byla dovážena výpočetní technika i ze zemí západního bloku (např. IBM 701 a IBM 1410 z USA, ICT 1901 a SIRIUS z Velké Británie a velký univerzální počítač D 21 ze Švédska).

V 70 letech pak následoval v rámci socialistických zemí sdružených v Radě vzájemné hospodářské pomoci (RVHP) společný vývoj počítačů řady JSEP a později i SMEP zaměřený na kompatibilitu, unifikaci a jednotnost řešení. Ukončení vlastního vývoje byla nutná daň pro zrychlení dalšího rozvoje počítačů i za cenu, že se upustilo od mnoha směrů slibného rozvoje započatého ve VÚMS.

## **2. 1. 1 Počítače vyvíjené a vyráběné v rámci řady JSEP**

V roce 1969 založily státy sdružené v RVHP společný projekt jednotného systému elektronických počítačů (JSEP). Ve svých začátcích byl zaměřen na počítače třetí generace a nesl pracovní označení JSEP 1. Později přišly i systémy tříapůlté generace (JSEP 2) a byl v rozpracované fázi i projekt počítačů čtvrté generace (JSEP 3).

V rámci řady JSEP 1 byl v Československu zkonstruován a vyráběn počítač EC 1021 určený zejména pro hromadné zpracování dat. I přesto, že se řadil do kategorie malých počítačů, měl poměrně velký výkon, což bylo dosaženo použitím zápisníkové a řídicí paměti. Počítač měl 65 instrukcí, délku adresy paměti 16 bitů, nepřímou čtyř úrovnovou adresaci a zabezpečení lichou paritou. Řídicí paměť měla kapacitu 3072 slov o velikosti 72 bitů s vybavovacím cyklem 300ns. Hlavní paměť měla kapacitu 16 – 64 K slabik s vybavovací dobou 800ns. Počítač měl 1 multiplexní kanál s 16 podkanály a dva selektorové

kanály. Byl plně kompatibilní s ostatními počítači řady JSEP a z hlediska kódu a formátu dat na mediích byl kompatibilní i s počítači IBM 360 a Siemens 4004.

V rámci řady JSEP 2 byl v Československu zkonstruován a vyráběn počítač EC 1025. Počítač měl virtuální paměť do 16 M slabik, dynamické překládání adres, nepřímou adresaci dat v kanálech a ovládání přes konzolový display. Záznam o chodu systému a poruchových stavech se ukládal na pružný disk. Velikost operační paměti 256 K slabik a rychlost procesoru 30 až 40 tisíc operací za sekundu. Později byl inovován a distribuován pod označením EC 1026. Rychlost procesoru stoupla na 80 tisíc operací za sekundu a operační paměť se zvětšila na dvojnásobek stejně jako vnější diskové paměti. V letech 1985 až 1986 byla provedena další inovace pod označením EC 1027. Operační rychlost stoupla na 200 tisíc operací za sekundu a kapacita operační paměti vzrostla až na 1 M slabiky. Dále se již vývoj počítačů řady JSEP nedostal a byl ukončen v roce 1989.

## **2. 1. 2 Počítače vyvíjené a vyráběné v rámci řady SMEP**

Rozhodnutí o společném vývoji a budování Systému malých elektronických počítačů (SMEP) bylo přijato v roce 1974.

V rámci projektu SMEP 1 byly v Československu ve VÚVT Žilina zkonstruovány a v ZVT Bánská Bystrica vyráběny systémy SM 3/20 a SM 3/40. Jednalo se o počítače se společnou sběrnici, asynchronním přenosem dat a s paralelním připojením přídavných zařízení. Základem byl šestnácti bitový paralelní procesor se zásobníkovou pamětí, jeho instrukční kód měl 76 instrukcí.

V rámci projektu SMEP 2 byly v Československu zkonstruovány a vyráběny počítače SM 50/40, SM50/50 a SM 52/12. Počítač SM 50/50 byl šestnácti bitový mikropočítač řešený stavebnicovým způsobem, byl plně kompatibilní z předcházející řadou a umožňoval použití všech periferních zařízení projektu SMEP. Počítač SM 52/12 byl pak naším prvním

mikropočítačem patřícím do třídy označované jako superminipočítače, byl ekvivalentní k systému firmy DEC označovaného VAX-11. Byl vyvinut rovněž ve VÚVT Žilina a od roku 1986 vyráběn v ZVT Banská Bystrica.

Vývoj a výroba počítačů řady SMEP byla postupně tlumena a v roce 1989 z ekonomických a technologických důvodů zcela zastavena.

### **2. 1. 3 Minipočítače řady ADT 4000**

Souběžně s vývojem počítačů řady SMEP byl ve VÚMS otevřen národní projekt vývoje minipočítačů řady ADT 4000. Koncepce minipočítačů řady ADT 4000 prakticky kopírovala minipočítače Hewlett Packard řady 2100 včetně jejich programového vybavení. V rámci uvedeného projektu byly ve VÚMS vyvinuty a v ZPA Čakovice postupně vyráběny minipočítače s označením ADT 4100, ADT 4300, ADT 4500 a ADT 4700.

Vzhledem k vysoké provozní spolehlivosti a dobrému programovému vybavení byly hojně využívány zejména k řízení výrobních linek, obráběcích center, měřicích, diagnostických a grafických systémů a v neposlední řadě typ ADT 4500 sloužil jako číslicová část hybridního počítače ADT 7000.

Hojně byly využívány i v armádě a to jak ve stacionárním tak mobilním provedení. V 80 letech byla do výzbroje armády postupně zavedena řada malých mobilních počítačů pod označením MOMI (konkrétně MOMI 1, MOMI 2 a MOMI 3), postavených na bázi minipočítačů ADT 4300, ADT 4500 a ADT 4700. Soupravy MOMI 2 a MOMI 3 již umožňovaly výstavbu terminálových sítí v polních podmínkách.

Vývoj a výroba minipočítačů řady ADT byla v důsledku nástupu nové generace výpočetní techniky prezentované osobními (personálními) počítači na začátku 90 let ukončena.



#### **2. 1. 4 Analogové počítače**

Spolu s vývojem číslicových počítačů se vyvíjela a vyráběla celá řada analogových počítačů. Pro úplnost považuji za vhodné uvést v krátkosti alespoň ty nejznámější.

Univerzální analogový počítač MEDA, který se v různých obměnách vyráběl od roku 1956 dosáhl i světové úrovně, vyrobilo se jich asi 1200 a většina z nich skončila v zahraničí. Z řady dalších stojí za to vzpomenout počítač SARO, který získal roku 1958 Grand Prix na světové výstavě v Bruselu.

Za zmínku stojí i počítač AP 3M, který byl jedním z největších našich analogových počítačů. Nelze dále nevzpomenout počítač APŠ, kterého se vyrobilo v 60 letech přes 400 kusů a pro svou cenovou dostupnost byl instalován v mnoha středních školách.

V dalším období byla výroba analogových počítačů, z nichž mnohé dosáhly světové úrovně, utlumována a nahrazována výrobou číslicové techniky.

#### **2. 1. 5 Současnost**

V druhé polovině 80 let se začíná psát nová etapa historie počítačů. Dochází k zastavení vývoje a postupnému utlumení výroby sálových počítačů a objevuje se nový fenomén osobní (personální) počítač a na něj navazující výstavba rozsáhlých (celosvětových) počítačových sítí.

Výpočetní technika se stěhuje z klimatizovaných počítačových sálů na pracoviště jejich koncových uživatelů. Využití výpočetní techniky přestává být doménou úzkého okruhu počítačových specialistů a inženýrů a postupně nabývá masových rozměrů a proniká do všech oblastí života společnosti.

### 3 Programovatelný logický automat

Programovatelný logický automat nebo-li PLC (z anglického Programmable Logic Controller) je relativně malý průmyslový počítač používaný pro automatizaci procesů v reálném čase - řízení strojů nebo výrobních linek v továrně. Pro PLC je charakteristické, že program se vykonává v tzv. cyklech. V moderním pojetí je výraz PLC nahrazován tzv. PAC i když označení PLC je celosvětově hojně rozšířené a udrží se i nadále. [1]

PLC automaty jsou odlišné od běžných počítačů nejen tím, že zpracovávají program cyklicky ale i tím, že jejich periferie jsou přímo uzpůsobeny pro napojení na technologické procesy. Převážnou část periferií v tomto případě tvoří digitální vstupy (DI) a digitální výstupy (DO). Pro další zpracování signálů a napojení na technologii jsou určeny analogové vstupy (AI) a analogové výstupy (AO) pro zpracování spojitých signálů. S rozvojem automatizace v průmyslu jsou používány i další moduly periferních jednotek připojitelných k PLC, které jsou nazývány funkčními moduly (FM) např. pro polohování, komunikačními procesory (CP) pro sběr a přenos dat a další specifické moduly podle výrobce konkrétního systému. [1]

Z hlediska konstrukce PLC se tyto dělí do skupiny „kompaktních“ a „modulárních“ systémů. [1]

Kompaktní systém je takový systém, který v jednom modulu obsahuje CPU (Central Processor Unit), digitální a analogové vstupy/výstupy a základní podporu komunikace, v některých případech i zdroj. Rozšiřitelnost kompaktních systémů je omezena. [1]

Modulární systém je takový systém, kde jsou jednotlivé komponenty celku rozděleny do modulů. Celý systém PLC se potom skládá z modulů: zdroje, CPU, vstupů/výstupů, funkčních modulů. Modulární systém je možno dále rozšiřovat

(s ohledem na limity výstavby systému) a to v nepoměrně větším rozsahu než u kompaktních systémů. [1]

První používané PLC byly převážně schopny zpracovávat binární logiku řízení, jejich prvotním cílem byla náhrada reléových automatů. Postupně se s rozvojem polovodičových součástek rozšiřovalo spektrum použitelnosti těchto systémů na zpracování analogových signálů, matematických funkcí (zprvu v pevné řádové čárce, postupně v plovoucí řádové čárce) až po možnost realizace složitých systémů řízení obsahující zpracování binárních signálů, analogových hodnot, komunikaci s jinými systémy, přenos dat, archivaci naměřených hodnot, vlastní diagnostiku, tiskové výstupy atd.. [1]

Původně malé počítače pro automatizaci již dorostly do výkonných řídicích systémů, kdy jádro řídicího systému (modul CPU) obsahuje i několik procesorů, z nichž má každý svoji specifickou funkci. Malé a levné jednotky CPU samozřejmě neobsahují veškeré vymoženosti a komfort jako výkonově velké CPU. U velkých CPU je architektura (více procesorů) použita z důvodu zajištění potřebné odezvy a rychlosti zpracování dat v reálném čase. Každé CPU obsahuje jeden „hlavní“ procesor, který zpracovává programový algoritmus řízení (vytvořený programátorem jako uživatelská aplikace) a další procesory, které jsou tomuto podřízeny. Tyto podřízené procesory zajišťují komunikaci po interní sběrnici s jednotkami vstupu/výstupu, komunikaci s dalšími procesory (např. na síti), sběr dat z decentralních periférií a další funkce. V dnešní době není výjimkou ani případ, kdy CPU obsahuje WWW server (HTML generátor), tzn. že toto CPU může být připojeno do sítě (zpravidla neveřejné) a být sledováno a řízeno použitím běžného prohlížeče WWW. Řídící jednotky některých modulárních systémů jsou ve skutečnosti klony osobních počítačů v provedení se zvýšenou odolností vůči vnějším vlivům a s upraveným standardním desktopovým operačním systémem. [1]

Orientačně se cena malých kompaktních systémů pohybuje v cenách od 2 000-10 000 Kč, cena velkých a výkonově vyšších systémů v rozsáhlé konfiguraci může dosahovat částek 500 000 Kč i vyšších. Vzhledem k efektivnosti těchto systémů v průmyslu nejsou tyto položky nijak závratné. Tato cena je ale hlavně závislá na systému, který má řídit, protože velkou část pořizovacích nákladů tvoří ne samotné PLC, ale právě snímače a napsání funkčního programu. V případě modulárních systémů samozřejmě i počet samotných modulů. [1]

## 4 První testovací prototyp

První prototyp jsem navrhl pro ovládání výkonových částí, především světel. Toto uvažované řešení mělo zjednodušit dosavadní mechanické ovládání, které již sloužilo řadu let bez větší chyby a bylo poměrně spolehlivé a které jsem chtěl nahradit ovládáním pomocí počítače.

Počítač měl být vzdálen několik málo metrů od výkonové části. Na PC by běžel jednoduchý program, přes který by se ovládala výkonová část kliknutím na jednotlivá tlačítka, která ovládají výstupy na jednočipu umístěného ve výkonové části přes převodník USB-RS485.

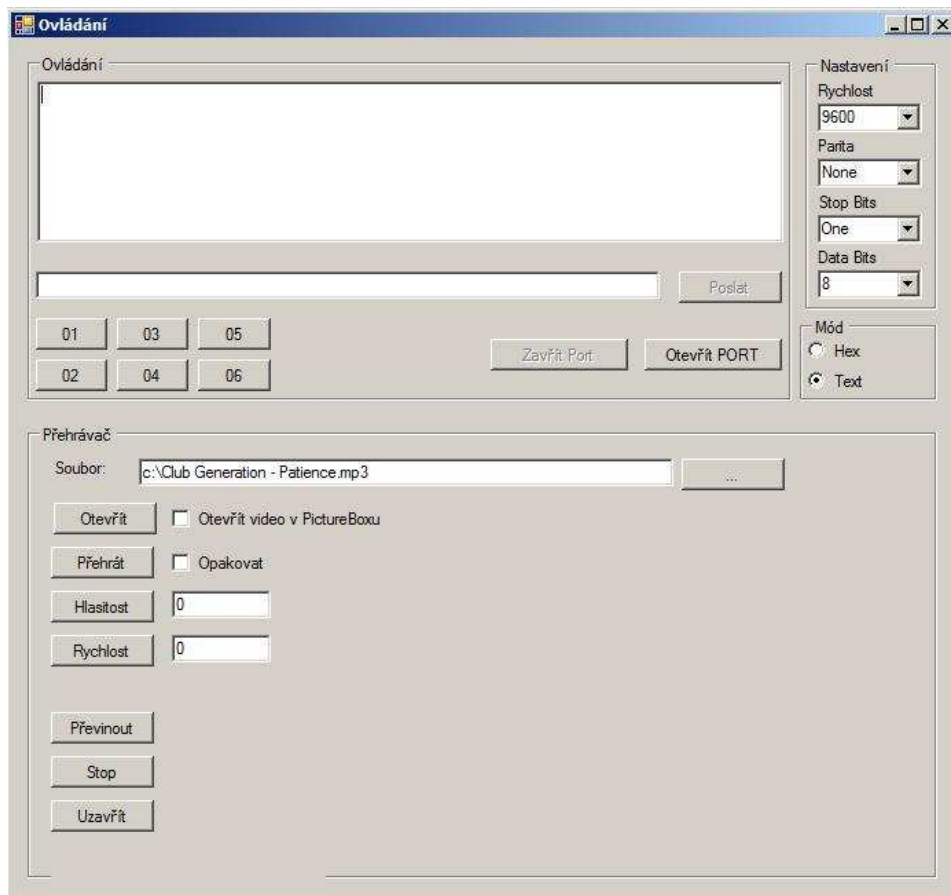
### 4. 1 Program v PC

Vlastní program je vytvořen ve studentské verzi Microsoft Visual C# 2005.

Po otevření programu je třeba nastavit parametry přenosu a otevřít PORT převodníku USB-RS485. Po stisknutí tlačítka „Otevřít PORT“ se najdou všechny dostupné sériové porty pomocí „GetPortNames()“ a posléze je výběr potvrzen.

Pro odesílání přednastavených znaků slouží tlačítka 01 až 06, kterými se spínají jednotlivé výstupy. Odesílání se provádí přes „comPort.Write“ . Další možností odesílání znaků je možné přes TextBox, kam se napíše znak a odešle po stisknutí tlačítka „Poslat“.

Program byl původně navržen pro snadné ovládání světel a různých spínaných efektů, proto je implementován i přehrávač multimédií. Soubor multimedia se otevře pomocí tlačítka „Otevřít“ a spustí pomocí tlačítka „Spustit“. Je možné zvyšovat a snižovat hlasitost i rychlost přehrávání, převinout multimediální soubor nebo jej stopnout a zavřít.



Obr. 4.1: Náhled okna ovládacího programu [autor]

## 4. 2 Převodník USB-485, FTDI

Vyzkoušel jsem několik převodníků. Některé jsem si vyrobil sám pomocí schémat na internetu, jiné jsem koupil jako modulární stavebnice. Zatím nejlepšími převodníky, podle mých zkušeností, jsou převodníky s čipem FTDI. Jejich nespornou výhodou je spolehlivost přenosu, kompatibilita s hardwarem a podpora s velkým množstvím operačních systémů.

#### **4. 2. 1 FTDI**

Firma FTDI vyrábí obvody FT232BM a FT245BM pracující jako konvertory USB – UART a USB – FIFO. Spojují tedy možnosti sběrnice USB spolu s jednoduchým připojením vnějších zařízení, protože komunikace sériovým asynchronním kanálem nebo po paralelní sběrnici je jistě snazší než po sběrnici USB. [7]

Obvody FT232BM a FT245BM jsou již druhou generací populárních USB konvertorů. Tyto součástky však pouze nepřidávají nové funkce do svých předchůdců (FT8U232AM a FT8U245AM), ale navíc zachovávají částečnou vývodovou kompatibilitu a redukuje počet vnějších součástek. Tím se snižují náklady na vývoj a výrobu zařízení a otevírají se nové možnosti v dalších aplikačních oblastech. [7]

#### **4. 2. 2 Modulární řešení:**

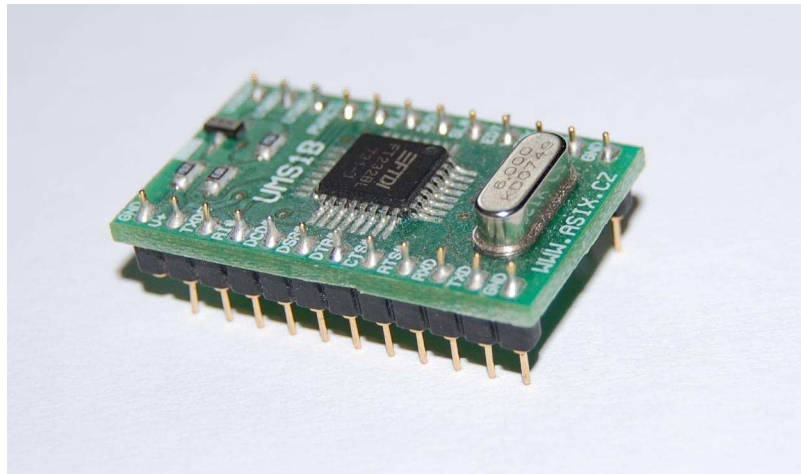
Nejjednodušší, spolehlivé a funkční řešení bylo pořízení modulu převodníku USB-485. Jde o dva moduly UMS1B a UDS1, které jsou běžně k dostání na internetu a je na ně možné připojit všechny výstupy i přes TTL.

##### UMS1B: [7]

UMS1B je modul, foto viz. obr. 4.2, založený na obvodu FT232BM firmy FTDI, který umožňuje velmi snadné připojení uživatelské aplikace k PC pomocí sběrnice USB, aniž by byly nutné jakékoliv předchozí znalosti o problematice USB.

Vstupem do modulu jsou 2 signály z konektoru USB, výstupem jsou signály sběrnice UART s možností plně hardwarového řízení toku dat. Modul umožňuje přenos dat rychlostí až 920 kbaud po sběrnici RS232 a až

3000 kbaud po sběrnici RS422/485. Navíc modul podporuje připojení externí paměti EEPROM s VID, PID sériovým číslem pro OEM aplikace.



Obr. 4.2: Modul UMS1B [autor]

### 3. VLASTNOSTI MODULU UMS1B

- Datové vstupy/výstupy 5 V - CMOS kompatibilní
- Napájení modulu ze sběrnice USB nebo externí
- Přenosová rychlost až 1 MBd po sběrnici RS232 a až 3 MBd po sběrnici RS485
- Možnost plně hardwarového řízení toku dat
- Vyrovňovací paměť pro příjem 384 byte
- Vyrovňovací paměť pro vysílání 128 byte
- Podpora řízení převodníku úrovní RS485
- Integrovaný frekvenční násobič 6 MHz - 48 MHz
- Protokol USB 1.1
- Podpora OEM aplikací (možnost uložení VID a PID produktu v externí paměti EEPROM)
- Možnost hardwarové optimalizace toku dat pomocí změny signálu CTS, DSR, DCD nebo RI (více informací poskytuje internet v aplikační poznámce č.4)
- Možnost využití volné kapacity v nevyužité oblasti EEPROM

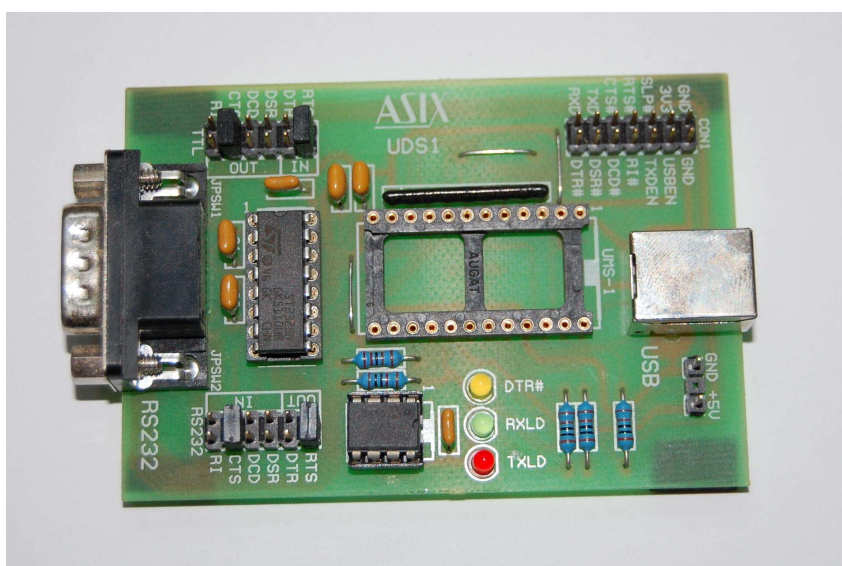


- Ovladače pro WIN 98/2K/ME/XP, Mac OS8/OS9/OS X a Linux zdarma  
Modul je určen pro zasunutí do patice DIL24 nebo pro montáž do PCB.

### UDS1 [8]

Vývojová deska UDS1, foto viz. obr. 4.3, umožňuje snadné připojení a testování uživatelské aplikace komunikující po sběrnici V.24 (např. mikrokontrolér s rozhraním UART) nebo RS232 (např. myš, COM) k PC po sběrnici USB. Jádrem komunikace je modul UMS1B obsahující obvod FT232BM firmy FTDI, který zajišťuje veškerou potřebnou komunikaci. Pro tento modul je na desce UDS1 připravena patice DIL24. Připojení aplikace RS232 je možné pomocí konektoru CANON 9M s rozmístěním pinů dle standardu RS232. Navíc si uživatel může zvolit vlastní umístění signálů RTS, CTS, DTR, DSR, DCD a RI v konektoru CANON 9M. Připojení aplikace V.24 se provádí konektorem CON1, který obsahuje navíc ještě signály USBEN, TXDEN, SLP#, 3V3OUT. Stav komunikace je indikován třemi signalizačními diodami.

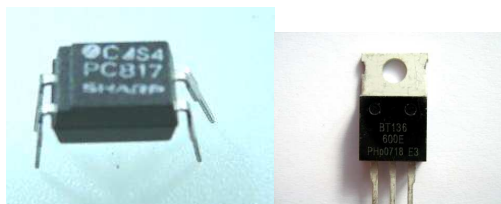
UDS1 umožňuje testování aplikací s vlastním zákaznickým číslem (PID, VID) uloženým v paměti EEPROM 93LC46B, která je osazena na desce.



Obr.: 4.3: Vývojová deska UDS1 [autor]

### 4.3 Výkonová část:

Srdcem výkonové části je programovatelný jednočip ATtiny2313, který je napájen přímo z USB portu (více o ATtiny2313 viz příloha). Vstupy jsou chráněny přes rezistory dvěma transily. Samotné zapojení je prakticky katalogové. Napájení je detekováno svítivou led diodou.



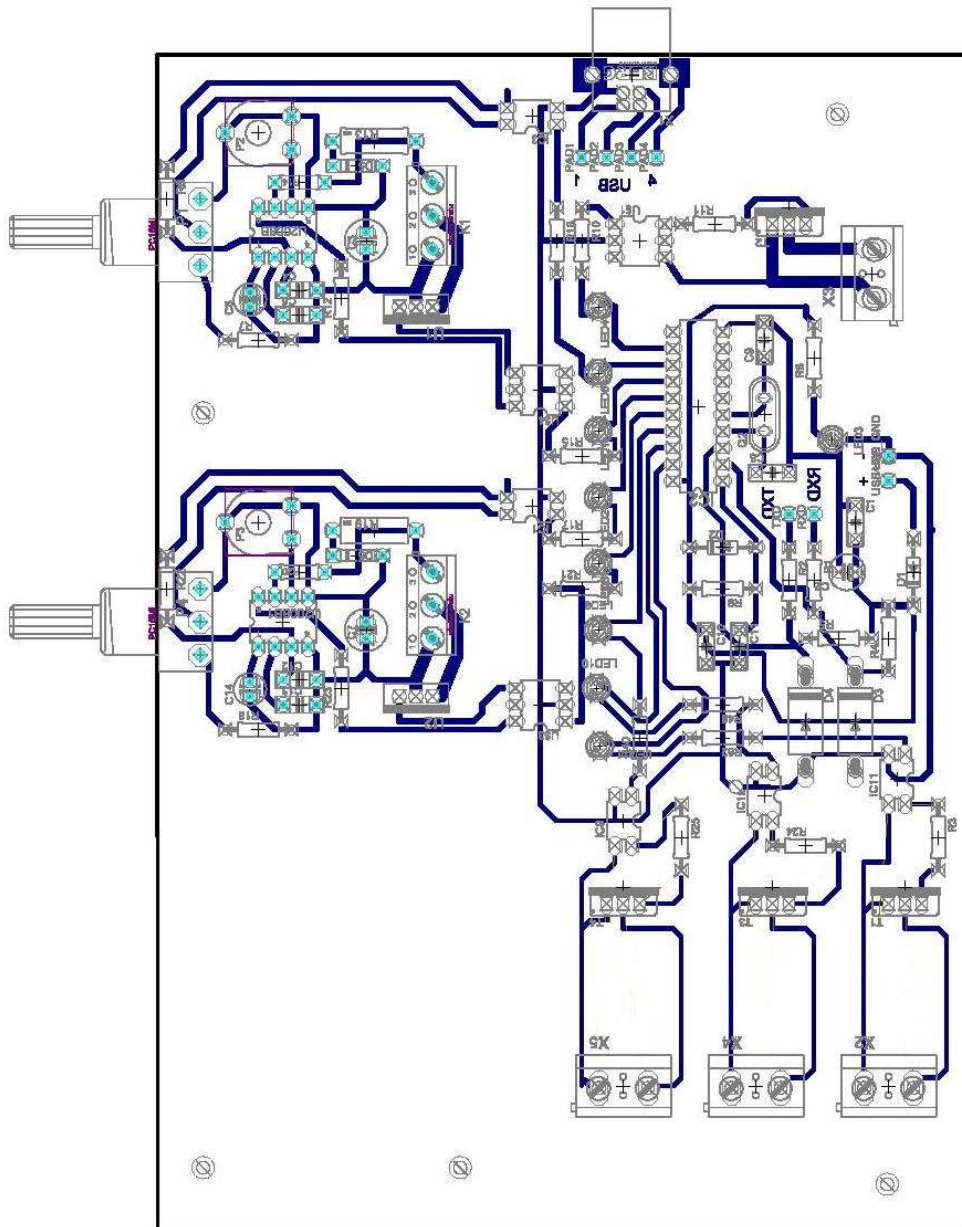
Obr. 4.4: vlevo optočlen pc 817, vpravo triak [autor]

Signál výstupů je zapojen přes optočleny, které jsou zobrazeny na obr. 4.4, aby bylo zachováno galvanické oddělení výkonové části. Čtyři výstupy jsou poté přivedeny na triaky bt139, které obstarávají samotné výkonové spínání. Připojení je přes klasické PA svorkovnice.

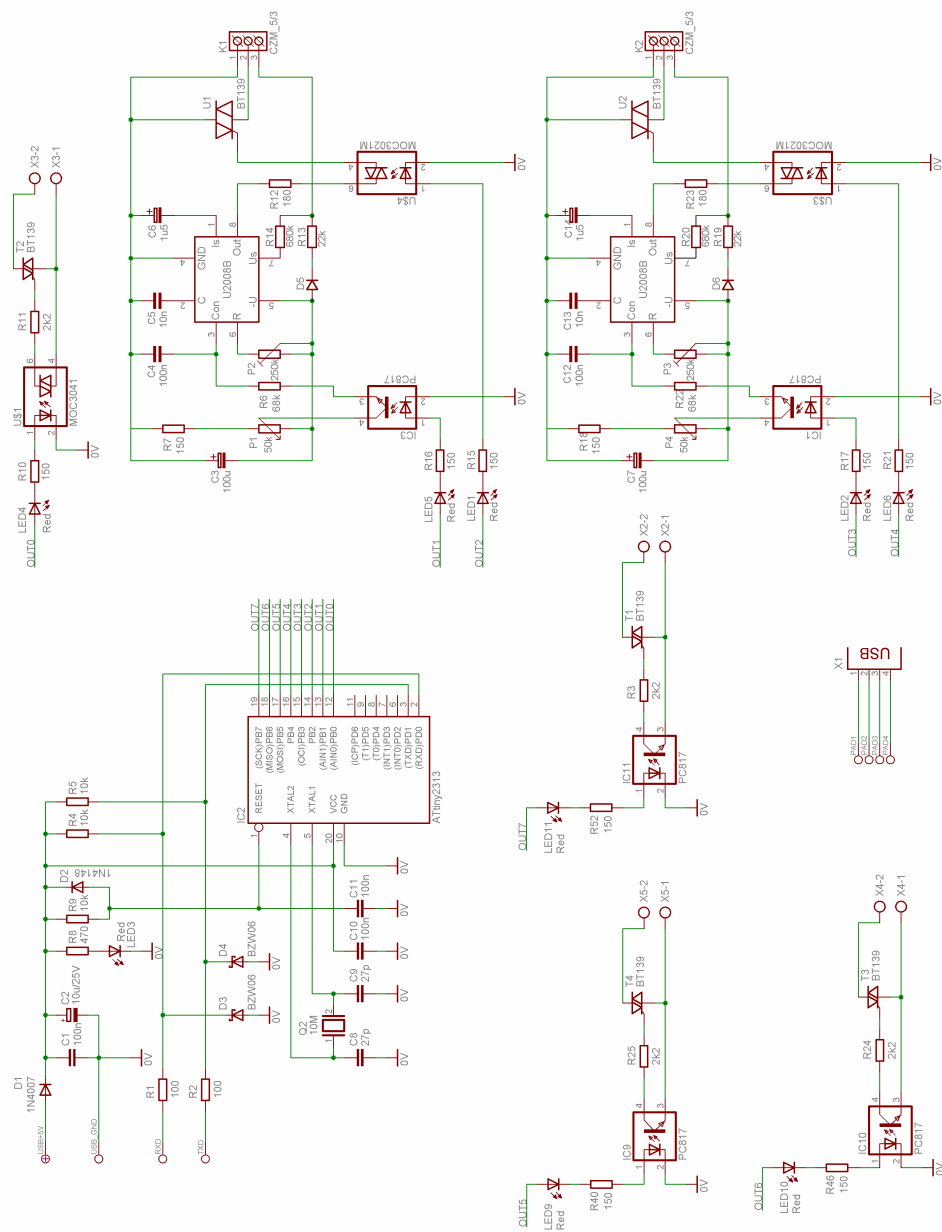
Dvakrát dva výstupy jsou přes optočleny zapojeny k výkonovému regulátoru. Napětí se připojuje na svorky 1 a 3, které je přes ochranný rezistor přiváděno na synchronizační vstup řídicího obvodu IO1. Tento obvod je určen pro spolehlivé fázové řízení s proudovou zpětnou vazbou a pomalým spouštěním. Kvůli velmi malé vlastní spotřebě jej lze přímo napájet ze sítě přes jednocestné usměrnění diodou a omezovací rezistor, jenž sníží napětí na cca 15V. Rezistor i přes malou spotřebu IO je dimenzován na výkonovou ztrátu 2W, hlavně kvůli proudovým impulsům jdoucím do řídicí elektrody triaku a úbytek na rezistoru, který může dosahovat až 230V. Řídicí napětí určující úhel sepnutí triaku je získáváno odporovým děličem a je přiváděno na vstup Noc. Odporovým trimrem se nastavuje minimální úhel sepnutí v plném rozsahu 180 °.

Pomocí optočlenu PC817 je spínán výkonový regulátor. Buď je sepnut a výstup je výkonově omezen podle mechanického nastavení ovládacího potenciometru. Přes optočlen MOC3021M je spínán výkonový triak, při sepnutí je triak sepnut, při rozepnutém optočlenu je triak uzavřený. Celé schéma zapojení

je zobrazeno na obr.4.6 a návrh plošného spoje s osazenými součástkami je vidět na obr. 4.5.



Obr. 4.5: Návrh plošného spoje s osazenými součástkami prvního prototypu [autor]



Obr. 4.6: Schéma zapojení prvního prototypu [autor]

Toto zapojení bylo sestavené pouze na nepájivém poli, kde bylo i zdárně odzkoušeno. Bohužel se neuskutečnila výroba plošného spoje a osazení. Původní myšlenka, ovládat světla a další pomocné efekty, padla, když jsem si chtěl vyzkoušet i PW modulaci. Proto jsem od tohoto projektu upustil a začal jsem se ubírat trochu jiným směrem. Myšlenka je ale pořád velmi reálná a snad si někdo, možná i já osobně, tento výrobek opravdu vyrobí a bude používat.

## 5 Druhý prototyp

Druhým prototypem bylo ovládání diodových světel pro železniční model. Původně mělo být zařízení použito na jednom zahraničním modelovém kolejišti. Po dokončení se několikrát přesunulo a jediný prototyp je na cestách po střední Evropě společně s dopravními nadšenci. Proto jsem byl nucen vyrobit druhý výrobek, který je prakticky totožný s prvním prototypem.

Prvotní myšlenka měla být pouze o zapínání a vypínání světel, ale pro takzvané „noční jízdy“ bylo třeba vytvořit efektnější projekci. Po několika diskuzích a výměně názorů jsme dospěli ke konečné realizaci a vytvořili nástroj pro ovládání a stmívání světel podle naprogramovaného schématu.

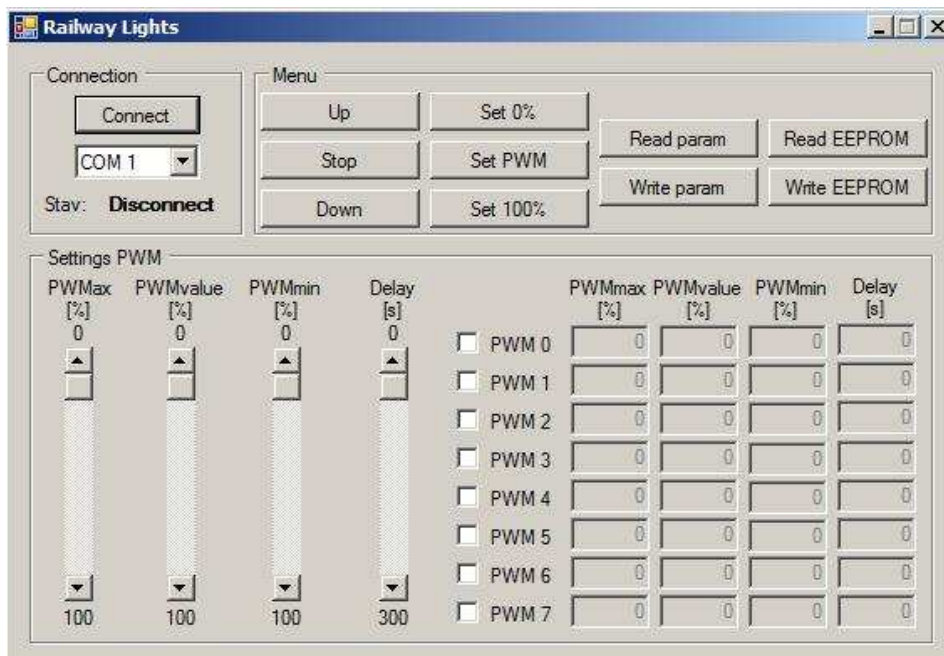
### 5. 1 Program v PC

Vlastní program byl vytvořen ve studentské verzi Microsoft Visual C# 2005. Jde podobně jako v prvním projektu o základní ovládání. Po otevření programu je třeba vybrat port, přes který se budeme připojovat. Pomocí FTDI čipu se simuluje sériový port (viz. kapitola 3).

V horní části uprostřed je šestice ovládacích tlačítek, pomocí kterých lze ovládat již nastavené výstupy. Tlačítka „UP, STOP, DOWN“ jsou i na vlastním hardwarovém zařízení a proto je možné i „ruční ovládání“. Pomocí tlačítka „UP“ zvyšujeme výstupní výkon až do nastavené hodnoty, naopak tlačítkem „DOWN“ snižujeme výstupní výkon až na minimální nastavenou hodnotu pomocí PWM modulace. Tlačítkem „Stop“ vypneme výstupy. Pomocí „Set 0%“ a „Set 100%“ ihned nastavíme minimální a maximální hodnotu a stisknutím „PWM“ se výstupy nastaví na nastavenou hodnotu „PWMvalue“.

Tlačítka „Read param“ a „Write param“ můžeme jednotlivé parametry načíst ze souboru. Pomocí „Read EEPROM“ a „Write EEPROM“ načteme a zapíšeme hodnoty do EEPROM paměti jednočipu.

V dolní části okna se nastavují jednotlivé parametry pro jednotlivé výstupy. Zaškrtnutím políčka u výstupu aktivujeme nastavování, můžeme zaškrtnout i více výstupů najednou a parametry se pak nastavují stejně. Nastavování hodnot se provádí pomocí posuvného. Můžeme nastavit maximální a minimální hodnotu od 0 až do 100% výkonu, nastavíme i PWM hodnotu, která se okamžitě objeví na výstupech po stisknutí tlačítka „Set PWM“. Posuvníkem „Delay“ nastavíme zpoždění PWM modulace. Maximální zpoždění je 300s.

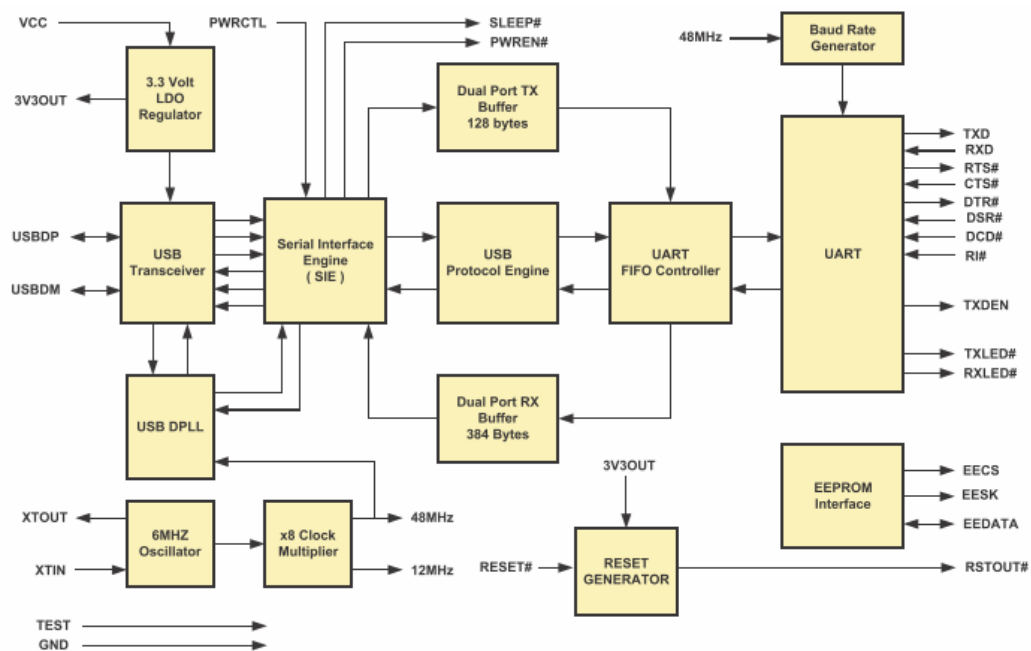


Obr. 5.1: Ovládání RailWay Lights pomocí PC [autor]

## 5. 2 Výkonová část:

V druhém prototypu jsem již nepoužil modulární řešení pro převodník z USB. Chtěl jsem si vyzkoušet mé dovednosti a docílit maximální seberealizace. Nakonec to asi nebylo zrovna moudré rozhodnutí, protože samotný IO převodník má několik málo milimetrů (7x7mm) a vývody opravdu nejsou z největších (0,8mm). Maximálně jsem si vyzkoušel trpělivost i zručnost při boji s páječkou, u které jsem hrot co nejvíce zbrousil do špičky.

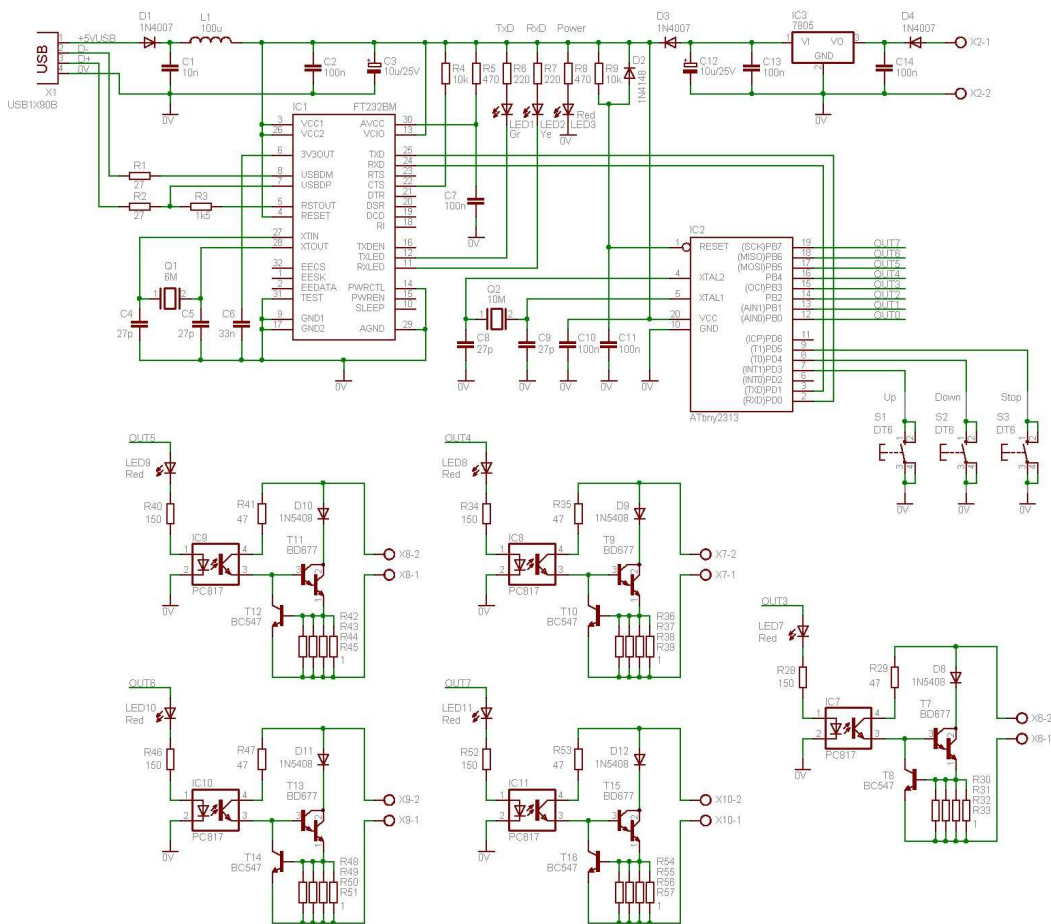
Převod USB-UART zprostředkovává IO FT232BM, který pracuje jako konvertor USB-UART. Umožňuje tedy spojení USB sběrnice s jednoduchým vnějším zařízením. (viz. kapitola 3).



Obr. 5.2: Blokové schéma FT232BM [10]

Datový přenos a napájecí napětí je indikováno pomocí LED diod.

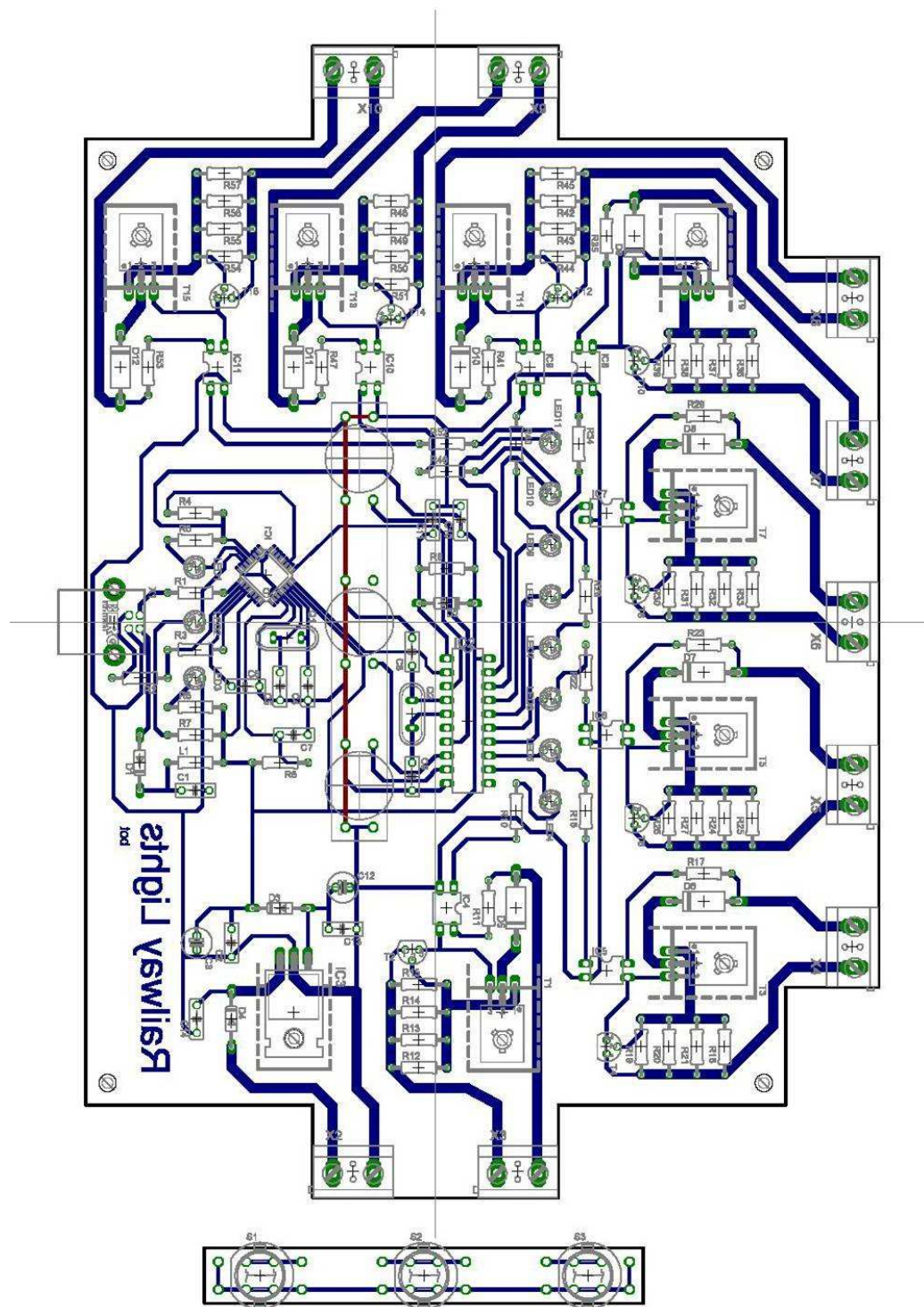
Srdcem zapojení je jednočip ATtiny2313 (viz. schéma zapojení obr. 5.3). Na IO jsou připojena ovládací tlačítka, jejich funkce je vysvětlena v předchozí kapitole. Tlačítka je možné stisknout přes díry ve vrchním víku, které jsou přelepeny potisknutou nálepkou. Sepnutí výstupů je indikováno LED diodami a výstupy jsou galvanicky odděleny od výkonové části optočleny, které spínají Darlingtonovy tranzistory (max. 400VA).



Obr. 5.3: Schéma zapojení druhého uvažovaného projektu [autor]

Výkonový výstup je zakončen dvojitými PA svorkovnicemi, které jsou navrženy tak, aby vodiče mohly být vyvedeny ven z krycí krabice. Plošný spoj (viz. obr. 5.4) je ke krycí krabice (viz. obr. 5.5) upevněn pomocí distančních sloupků.





Obr. 5.4: Návrh plošného spoje s osazenými součástkami druhého prototypu [autor]



Obr. 5.5: Druhý prototyp [autor]

### 5. 3 Programová část jednočipu:

Vlastní program pro jednočip byl napsán za pomoci jazyka C. Po odeslání určitého znaku z PC přes vlastní program (nebo třeba i přes hyperterminál) dojde k přečtení znaku a přes funkci CASE je přiřazena nastavená funkce, která vyvolá danou odezvu (použité funkce viz zdrojový kód na přiloženém CD).

Základní přehled vlastností symbolů:

**"A" : test spojení s HW**

- odpověď: "1"

**"B" : simuluje stisk tlačítka UP**

- odpověď: "1"

**"C" : simuluje stisk tlačítka STOP**

- odpověď: "1"

**"D" : simuluje stisk tlačítka DOWN**

- odpověď: "1"

**"E" : nastavení všech výstupů na 100%**

- odpověď: "1"

**"F" : nastavení všech výstupů na 0%**

- odpověď: "1"

**"G" + CH + PWMmin + PWMmax + Time: nastavení parametru  
jednoho výstupu**

- parametry:

CH -> požadovaný kanál (0 .. 7)

PWMmin -> minimální výkon

PWMmax -> maximální výkon; PWMmin <= PWMmax!!!

Time -> doba přejezdu

- odpověď:

"1" -> příkaz přijat úspěšně

"0" -> chyba při příjmu (time-out, out of range,...)

**"H" + CH: žádost o zaslání parametru požadovaného kanálu**

- parametry:

CH -> požadovaný kanál (0 .. 7)

- odpověď:

"1" + PWMmin + PWMmax + Time

"0" -> nesprávné číslo kanálu

**"I" : uložení parametru do EEPROMky**

- odpověď: "1"

**"J" : načtení parametru z EEPROMky**

- odpověď: "1"

**"K" : žádost o zaslání aktuálního stavu PWM všech kanálů**

- odpověď:

"1" + PWM0 + PWM1 + PWM2 + PWM3 + PWM4 + PWM5 +  
PWM6 + PWM7

PWM0 -> aktuální stav PWM výstupu 0

PWM1 -> aktuální stav PWM výstupu 1

PWM2 -> aktuální stav PWM výstupu 2

PWM3 -> aktuální stav PWM výstupu 3

PWM4 -> aktuální stav PWM výstupu 4

PWM5 -> aktuální stav PWM výstupu 5

PWM6 -> aktuální stav PWM výstupu 6

PWM7 -> aktuální stav PWM výstupu 7

**-"L" + PWM0 + PWM1 + PWM2 + PWM3 + PWM4 + PWM5 +  
PWM6 + PWM7 : okamžité nastavení PWM všech kanálů**

- odpověď: "1"

## 6 Třetí projekt

Když je člověk náctiletý a ještě školou povinný, má velkou chuť si něco vyzkoušet a zkoumat. Já jsem měl vždycky rád elektrotechniku a jsem moc rád, že se jí mohu zabývat i v pracovním procesu. Předchozí zde uvedené projekty jsou spíše amatérského ražení. Poslední třetí projekt se přímo týká profesionálního řízení a je plně využit v praxi.

### 6. 1 Proč...

Předchozí řešení jsou postavena na jednočipu, který má své omezení a hodí se spíše pro aplikace menších rozměrů a pro speciální aplikace. V životě jsem četl i viděl řadu možností nejrůznějšího ovládání a automatizování. Ať šlo o mechanické ovládání strojů, po jednoduchá releová ovládání, přes jednočipy, až po velmi složité automatizace v průmyslu.

Než jsem nastoupil do pracovního procesu, neměl jsem možnost se více seznámit s automaty a s projekty, které se právě řídí pomocí těchto automatů. Byl jsem velmi překvapen funkcí a možnostmi PLC, proto jako třetí projekt uvádím toto řešení.

Chtěl bych nastínit projekt a realizaci zautomatizování destilačního zařízení, které již řadu let funguje a snahou je ušetřit čas i energii.

Řešení, které zde chci uvést, je založeno na automatu Amit APT3100(S). Tento automat je modulový a jeho rozšiřitelnost je velmi jednoduchá. Řešení není nejlevnější, ale praktičnost je vysoká. Hlavní výhodou je snadná rozšiřitelnost, kterou např. u jednočipů těžko dosáhneme. Další velkou devizou je již profesionálně navržené řízení, které můžeme použít v průmyslových aplikacích.

## 6. 2 Řídící automaty AMIT

AMIT není světově známá značka automatu, jde o českou firmu, ale velmi rychle dohnala světové trendy a velmi se snaží držet krok a maximálně vyjít vstříc svým zákazníkům. Jako jedna z mála firem je schopna vyvinout zařízení přímo na míru již od několika málo desítek kusů, za velmi přiměřené finanční prostředky. Jejich návrhové prostředí „DetStudio“ a zobrazovací prostředí „ViewDet“ je zdarma, což velmi přispívá ke konkurenceschopnosti. Určitě se najde řada nedostatků, ale pro zde řešenou aplikaci je více než dostačující.

## 6. 3 Amit APT3100(S)

Volně programovatelný terminál / řídicí systém s dotykovou obrazovkou

- Podsvětlený grafický 5,7“ TFT displej, 320 × 240 bodů
- Kombinace grafického a textového režimu
- Dotyková plocha na celém displeji
- RS232, Ethernet 10 Mbps
- Volitelně RS485 / RS232 / CAN / M-Bus
- Konektor PS2 pro externí klávesnici
- Procesor C167CR-LM / ST10F269 [12]



Obr. 6.1: AMIT APT3100(S) [17]

## 6. 4 Moduly AMIT

Jsou uvedeny jen zde použité moduly

Moduly komunikují po lince RS485 přes protokol ARION

Komunikační rychlosti: 9600 .. 57600 Bd

Počet modulů na síti RS485: 63

Počet modulů na segmentu RS485: 31

Napájení: 24 V ss.  $\pm 20\%$  [13]

### 6. 4. 1 DM-AI12

Jedná se o modul analogových vstupů s možnými vstupními rozsahy: Ni1000 / 5 V / 10 V / 20 mA (dle propojení propojkami na modulu). Vstupy jsou bez galvanického oddělení. Modul je komunikuje po lince RS485. [14]



Obr. 6.2: Modul analogových vstupů DM-AI12 [11]

#### **6. 4. 2 DM-DI24**

Modul 24 číslicových vstupů 24 V

Logická 0: Min. -30 V ss./stř., max. 5 V ss./stř.

Logická 1: Min. 16 V ss./stř., max. 30 V ss./stř.

Vstupní proud : 6 mA při 24 V ss./stř.

Špičkový vstupní proud: Max. 10 mA při 30 V ss./stř. [16]

#### **6. 4. 3 DM-DO18**

Modul 18 číslicových výstupů 24 V

Typ spínače: Spíná Ex+24V

Spínací prvek: MOS

Max. proud společnou svorkou: 1 A ss.

Doba sepnutí: 40  $\mu$ s

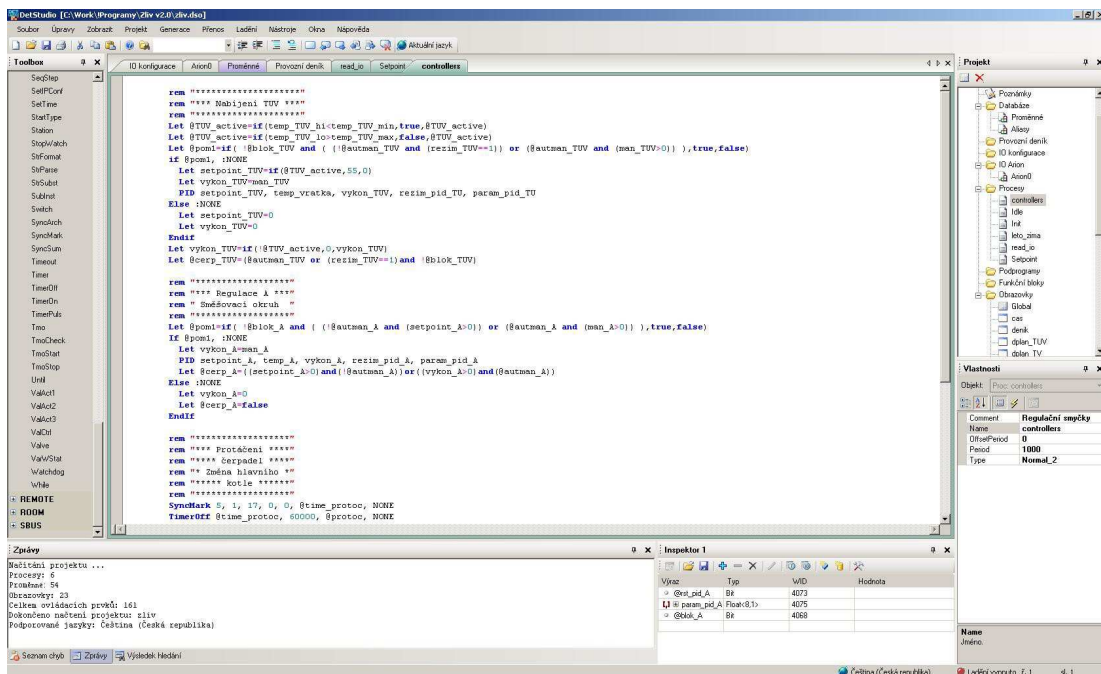
Doba rozepnutí: 100  $\mu$ s [15]



## 6.5 DetStudio

Návrhové prostředí DetStudio (náhledové okno viz obr. 6.3) je určeno pro tvorbu uživatelských aplikací pro všechny standardní řídicí systémy firmy AMiT. V jediném vývojovém prostředí lze vytvořit vlastní aplikaci, navrhnout a odsimulovat vzhled obrazovek, zobrazovačů řídicích systémů, definovat chybová hlášení, on-line ladit běžící aplikaci, vytvořit dokumentaci vytvořeného programu. Způsob programování a algoritmizace vychází ze staršího osvědčeného parametrizačního prostředí PSP3 a na úrovni vstupních zdrojových kódů je s ním DetStudio kompatibilní. [9]

Regulace pomocí automatu probíhá podle nastavených algoritmů. Dnes již každý víceúčelový automat má řadu přednastavených funkcí. Přehledné programování a zápis proměnných (viz. náhledové okno obr. 6.4) je dnes také samozřejmostí. Už není problém nastavit potřebnou hysterézi i s případným zpožděním. Můžeme si vybrat z velké řady čítačů, předdefinovaných algoritmů pro ovládání serv, regulaci přímotopných těles, atd.



Obr. 6.3: Náhled Controllers okna DetStudia [autor]

The screenshot displays the DetStudio interface with the 'Proměnné' (Variables) window open. The main window shows a list of variables with the following columns: Jméno (Name), Typ (Type), Právků (Flags), Sloupců (Columns), w/d, Vřam, Inš hodnota (Initial Value), Stavce (Status), and Komentář (Comment). The 'Vlastnosti' (Properties) window on the right shows details for the selected variable, including its name, type, and address.

Jméno	Typ	Právků	Sloupců	w/d	Vřam	Inš hodnota	Stavce	Komentář
AcoIndex	I			4017			4	Index pro SyncACh
AcoTime	ML	1	2016	4018			4	Synchronizace - Malice časů
ArchTime	MF	5	2016	4020			4	Synchronizace - Malice hodnot
Bk4	I			4000			4	Priznaky
Bk3	I			4069		8	4	Priznaky
bks4	I			4105			4	Priznaky
bks5	I			4106			4	Priznaky
Buffer	MI	1	408	4301			4	Buffer provozního demiku
DI	I			4010			4	Digitální výstupy AMNI
DO	I			4011			4	Digitální výstupy AMNI
DO_LED	I			4012			4	LED diody na panelu
dplen_cas_A	ML	8	2	4025		1.1.1980.4.00...	4	Malice časů denního plánu okruhu A
dplen_cas_TU	ML	8	2	4089		14400.028001...	4	Malice časů denního plánu TUV
dplen_hod_A	MI	8	2	4028		21.21.21.21...	4	Malice hodnot denního plánu okruhu A
dplen_hod_TU	MI	8	2	4089		1.01.01.01.0...	4	Malice hodnot denního plánu TUV
eq_dien_A	F			4016		25	4	Ekvivalentní reference A
eq_konst_A	F			4013		2.9	4	Ekvivalentní konstanta okruhu A
eq_noc_A	F			4002		19	4	Teplota v místnosti v režimu komfort
err	I			4081			4	Seznam odvozených chybových hlášek
err_ignore	I			4082			4	Seznam provozových chybových hlášení pro kláson
Index	I			4300			4	Index provozního demiku
kvilace	I			4035			4	Kvítace jednotlivých poutch
lc_dblaz	MI	8	1	4104			4	Automatický posun hodin lehn/zámi čas
lc_justakt1	L			4103			4	Automatický posun hodin lehn/zámi čas
lc_time	L			4101			4	Automatický posun hodin lehn/zámi čas
lc_uct1	L			4102			4	Automatický posun hodin lehn/zámi čas
man_A	F			4070			4	Manuální režim B
man_TUV	F			4021			4	Výkon TUV v manuálu
moto_cerp_A	ML	1	2	4059			4	Motobidny čerpadla A
moto_cerp_TUV	ML	1	2	4064			4	Motobidny čerpadla TUV
param_pid_A	MF	8	1	4075		0.3.200.0(2).100	4	Parametry PID regulátoru okruhu B
param_pid_TU	MF	8	1	4076		2.100.0.10.100	4	Parametry PID regulátoru okruhu C
period_arch	I			4024		20	4	Perioda archivace dat v sekundách
porif	F			4085			4	Pomocná proměnná

Obr. 6.4: Náhled zápisu proměnných v DetStudiu [autor]

## 6. 6 ViewDet:

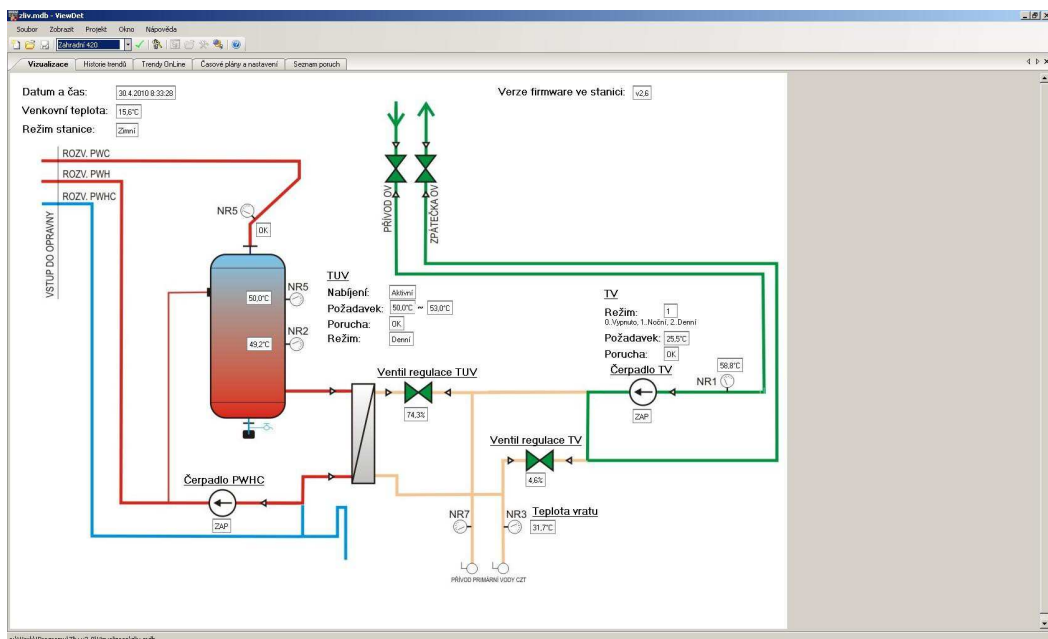
ViewDet je samostatný servisní nástroj pro Windows. Doplnuje a rozšiřuje možnosti návrhového prostředí DetStudio v oblasti sledování, ladění a nastavení aplikace v řídicím systému. V omezené míře lze program použít jako jednoduchou vizualizaci (viz. obr. 6.5). [4]

Základní vlastností je možnost čtení a zápisu jednotlivých hodnot databázových proměnných a aliasů (v zásadě stejné možnosti jako "watch" v DetStudios). Navíc ViewDet umí číst, zobrazovat, tisknout a exportovat archivy a provozní deník z řídicího systému (viz. obr. 6.6) včetně pamatování si jejich historie. Archivy lze zobrazit buď v tabulce nebo formou grafu. Další možností je tvorba archivů a zobrazování archivů vznikajících přímo na PC periodickým čtením určitých hodnot z procesní stanice. Zobrazované proměnné lze umístit libovolně v tzv. "scéně" s nadefinovaným obrázkem na pozadí - takto lze vytvořit velmi jednoduchou vizualizaci procesů. Pomocí zámku se zajistí, aby uživatel nechtěně nezměnil proměnné a parametry zobrazování v ViewDet. ViewDet umožňuje kompletní editaci IP konfigurace stanic připojených k průmyslovému Ethernetu. [17]

### Mezi hlavní benefity programu ViewDet patří:

- možnost uskupení zobrazovacích objektů ve scénách, rychlé přepínání mezi scénami
- pozadí ve scénách
- nové prvky - statický text a proměnná
- nový prvek pro snadnou editaci časových plánů grafickou metodou
- nastavitelná perioda komunikace, již lze nastavit individuálně ke každému prvku
- zavedení režimu user a admin - ochrana proti nechtěným změnám
- podpora aliasů
- tisky scén a prvků

- exporty zaarchivovaných dat
- download uživatelské aplikace do řídicího systému včetně možné zálohy a obnovení obsahu proměnných
- přehledné okno návrhu projektu
- vícenásobné komunikační profily se snadným přepínáním
- vyžaduje .NET (podpora Windows XP a vyšší) [17]



Obr. 6.5: Vizualizace – ViewDet [autor]

Aktuální poruchy:  
 Zapiavení prostoru  OK  
 Přehřátí prostoru  OK  
 Čerpadlo TV  OK  
 Čerpadlo TUV:  OK  
 Přehřátí TUV:  OK  
 Senzor teploty vratu:  OK  
 Senzor teploty TUV horní:  OK  
 Senzor teploty TUV dolní:  OK  
 Senzor venkovní teploty:  OK  
 Senzor teploty TV:  OK

Blokace okruhů od poruchy:  
 Blokace okruhu TV:  OK  
 Blokace okruhu TUV:  OK

Kvřitace poruchy:  
 Zapiavení TUV:  Zapiavení 1

**Provozní deník:**  
 Provozní deník, seznamník poruch ve stanicí

Datum	St.	Pře	Text
4.11.2008 15:06:08	4	lcd	Přihlášení uživatelé 65535.
4.11.2008 11:20:19	4	nos	Zavedení operčního systému.
4.11.2008 11:20:19	4	nos	Studny start sířového zavazšče (Flag:0000)
4.11.2008 11:28:19	4	nos	Teplý start sířového zavazšče (výpadek: 4.11.2008 11:28:19, Flag:0000).
4.11.2008 11:28:12	4	nos	Teplý start sířového zavazšče (výpadek: 4.11.2008 11:28:12, Flag:0000).
4.11.2008 11:29:31	4	nos	Studny start aplikace (Flag:0000)
4.11.2008 11:29:32	4	lcd	Přihlášení uživatelé 65535.
4.11.2008 11:30:49	4	nos	Teplý start aplikace (výpadek: 4.11.2008 11:30:45, Flag:0000).
4.11.2008 11:30:49	4	nos	Přihlášení uživatelé 65535.
4.11.2008 13:25:35	4	nos	Teplý start aplikace (výpadek: 4.11.2008 13:25:30, Flag:0000).
4.11.2008 13:25:35	4	lcd	Přihlášení uživatelé 65535.
4.11.2008 14:00:29	4	nos	Teplý start aplikace (výpadek: 4.11.2008 13:56:01, Flag:0000).
4.11.2008 14:00:29	4	lcd	Přihlášení uživatelé 65535.
4.11.2008 15:28:54	4	nos	Teplý start sířového zavazšče (výpadek: 4.11.2008 15:28:53, Flag:0000).
4.11.2008 15:29:47	4	nos	Studny start aplikace (Flag:0000)
4.11.2008 15:29:48	4	lcd	Přihlášení uživatelé 65535.
4.11.2008 15:32:08	4	nos	Teplý start aplikace (výpadek: 4.11.2008 15:32:07, Flag:0000).
4.11.2008 15:32:08	4	lcd	Přihlášení uživatelé 65535.
4.11.2008 15:33:26	4	nos	Teplý start aplikace (výpadek: 4.11.2008 15:33:21, Flag:0000).
4.11.2008 15:33:26	4	lcd	Přihlášení uživatelé 65535.
4.11.2008 15:34:43	4	nos	Teplý start aplikace (výpadek: 4.11.2008 15:34:38, Flag:0000).
4.11.2008 15:34:43	4	lcd	Přihlášení uživatelé 65535.
4.11.2008 15:35:19	4	nos	Teplý start aplikace (výpadek: 4.11.2008 15:35:14, Flag:0000).
4.11.2008 15:35:19	4	lcd	Přihlášení uživatelé 65535.
4.11.2008 15:35:48	4	nos	Teplý start aplikace (výpadek: 4.11.2008 15:35:37, Flag:0000).
4.11.2008 15:35:48	4	lcd	Přihlášení uživatelé 65535.
4.11.2008 11:20:19	4	nos	Zavedení operčního systému.
4.11.2008 11:20:19	4	nos	Studny start sířového zavazšče (Flag:0000)
4.11.2008 11:28:19	4	nos	Teplý start sířového zavazšče (výpadek: 4.11.2008 11:28:19, Flag:0000).
4.11.2008 11:29:12	4	nos	Teplý start sířového zavazšče (výpadek: 4.11.2008 11:29:12, Flag:0000).
4.11.2008 11:29:31	4	nos	Studny start aplikace (Flag:0000)
4.11.2008 11:29:32	4	lcd	Přihlášení uživatelé 65535.
4.11.2008 11:30:49	4	nos	Teplý start aplikace (výpadek: 4.11.2008 11:30:45, Flag:0000).
4.11.2008 11:30:49	4	lcd	Přihlášení uživatelé 65535.
4.11.2008 13:25:35	4	nos	Teplý start aplikace (výpadek: 4.11.2008 13:25:30, Flag:0000).
4.11.2008 13:25:35	4	lcd	Přihlášení uživatelé 65535.
4.11.2008 14:00:29	4	nos	Teplý start aplikace (výpadek: 4.11.2008 13:56:01, Flag:0000).
4.11.2008 14:00:29	4	lcd	Přihlášení uživatelé 65535.

Obr. 6.6: Provozní deník ViewDet [autor]

## 6. 7 Bezpečnost

Pro správný a bezpečný chod všech zařízení je důležité dodržovat alespoň některá základní pravidla. Jde-li o komerční zařízení, musí projektant dodržovat řadu zákonů, vyhlášek, musí chodit na školení a seznamovat se s novými technologiemi a postupy.

Každý, kdo chce pracovat s elektrickým zařízením, by si měl uvědomit, jak nebezpečná může být elektrická energie, není vidět, není cítit, ale dokáže velmi snadno ublížit. Základním bezpečnostním prvkem by proto měla být ochrana neživých částí a odpojení zařízení od zdroje energie při poruše. K tomu slouží nejrůznější bezpečnostní čidla zapojená do tzv. bezpečnostní smyčky: Ta nám umožňuje odpojení zařízení, která by mohla ohrozit zdraví. V případě, že použijeme automat, je vhodné neodpojovat napájení celého zařízení, ale zkontrolovat stav bezpečnostní smyčky a v případě poruchy vypnout zařízení případně ohrožující zdraví a s rozvahou postupně odstavit celé zařízení tak, aby se minimalizovali škody. V některých případech ani není možné celkové odpojení od el. zdroje, ale je třeba optimálně zajistit bezpečnost lidí i zařízení. (samočinné odpojení od zdroje ČSN 33 2000-4-41).

Na obrázku 6.7 je zapojení bezpečnostní smyčky, která se skládá z jednotlivých bezpečnostních prvků. Jednotlivé prvky jsou zapojeny do série přes spínací kontakty. Je-li všechno pořádku, smyčka je uzavřena a proud se uzavírá přes cívku stykače. Pro větší bezpečnost se zapojují dva stykače za sebou. Kontakty stykače spínají napájení sdruženého napětí 400VAC. Při detekování bezpečnostní člen rozezne kontakt, bezpečnostní smyčkou se rozezne, přestane jí téci proud a kontakty stykače se rozeznou.



## 6. 8 Popis zapojení a funkce:

Automatizace již běžícího procesu má jednu velkou výhodu, z již získaných zkušeností můžete předpokládat chování systému a podle něj navrhnout určité řešení.

### 6. 8. 1 Propojení s PC

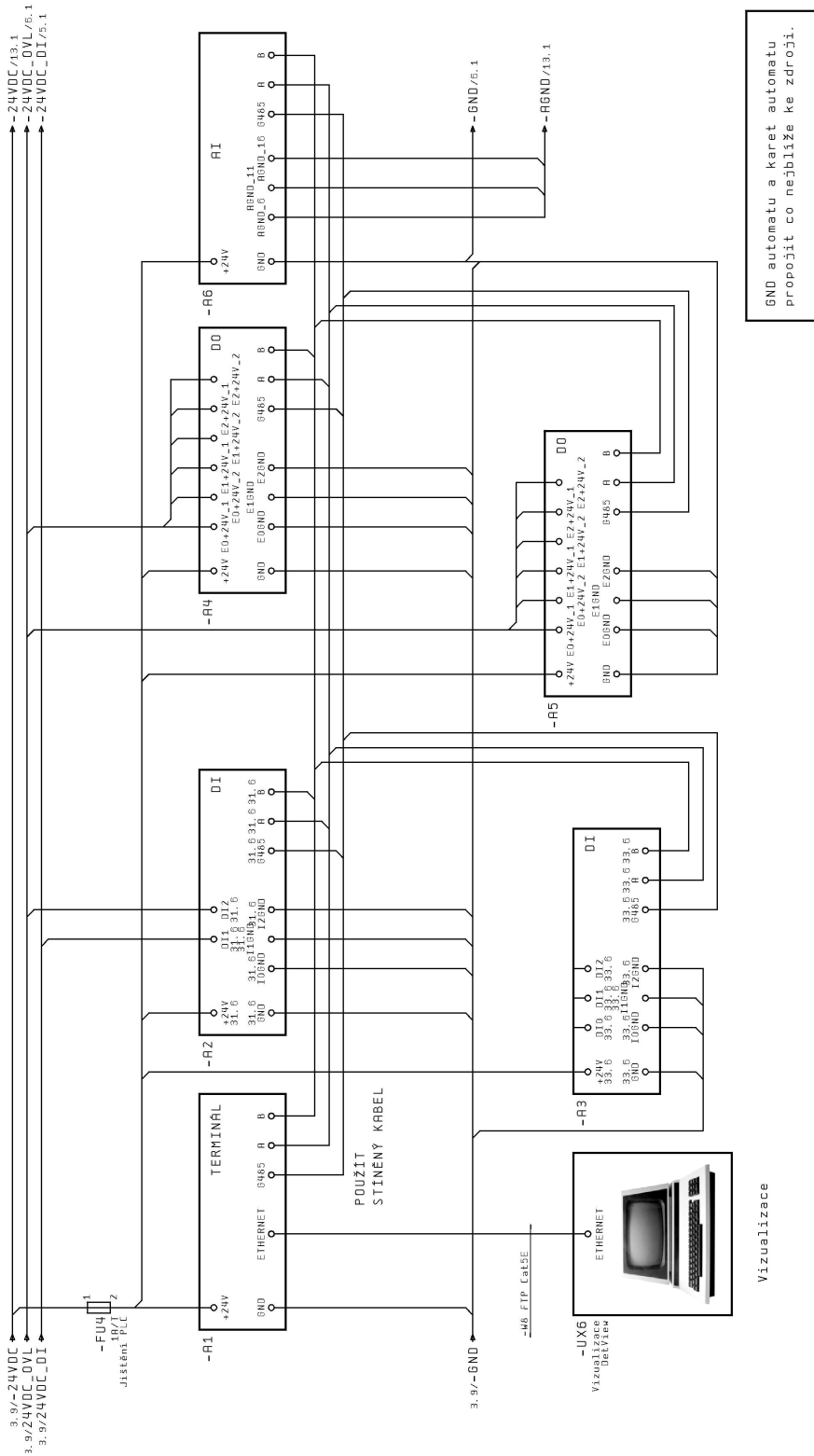
Základem je automat, který je napojen na PC pomocí rozhraní ETHERNET (viz obr. 6.8) přes které se dostaneme do automatu přes vizualizaci, pomocí které můžeme měnit nastavitelné hodnoty apod. Vizualizační stránky jsou nahané přímo v automatu a náhled nám může také zprostředkovávat již zmiňovaný ViewDet.

Vizualizace je primárně zobrazena na dotykovém displeji, přes který můžeme sledovat a měnit nastavené parametry. Zmiňovaný automat AMIT APT3100S má displej s rozlišením 320x240px, 256 barev, což umožňuje přiměřeně přehlednou vizualizaci.

Přehlednější může být ovládání přes PC. V DetStudiosu nastavíme IP adresu automatu, přes kterou nahlížíme do automatu např. pomocí View Det nebo jiného vizualizačního programu.

Jiné automaty, např. SAIA, mají přímo dotykové obrazovky s webovým rozhraním, které fungují na principu „webových obrazovek“. Stačí zadat IP adresu a můžete se „proklikávat“ obrazovkami bez jakéhokoli omezení.





GND automatu a karet automatu propojit co nejbliže ke zdroji.

Obr. 6.8: Zapojení PLC, PC a DI, DO, AI karet [autor]

## 6. 8. 2 Čidla a měření

Přes analogovou měřicí kartu sledujeme důležité parametry celého zařízení. Zejména nás zajímají tlaky a teploty v jednotlivých částech zařízení, hladiny stejnorodých i různorodých kapalin.

Pro měření teploty a tlaku je důležité vhodně zvolit měřicí rozsah, způsob měření, způsob uchycení a připojení čidla atd. Musíme si také uvědomit jak přesně, rychle, s jakou hysterezí danou veličinu můžeme a chceme měřit. V našem případě se budou teploty pohybovat od cca 20°C do 100°C a tlak nepřesáhne 0,4 bar.

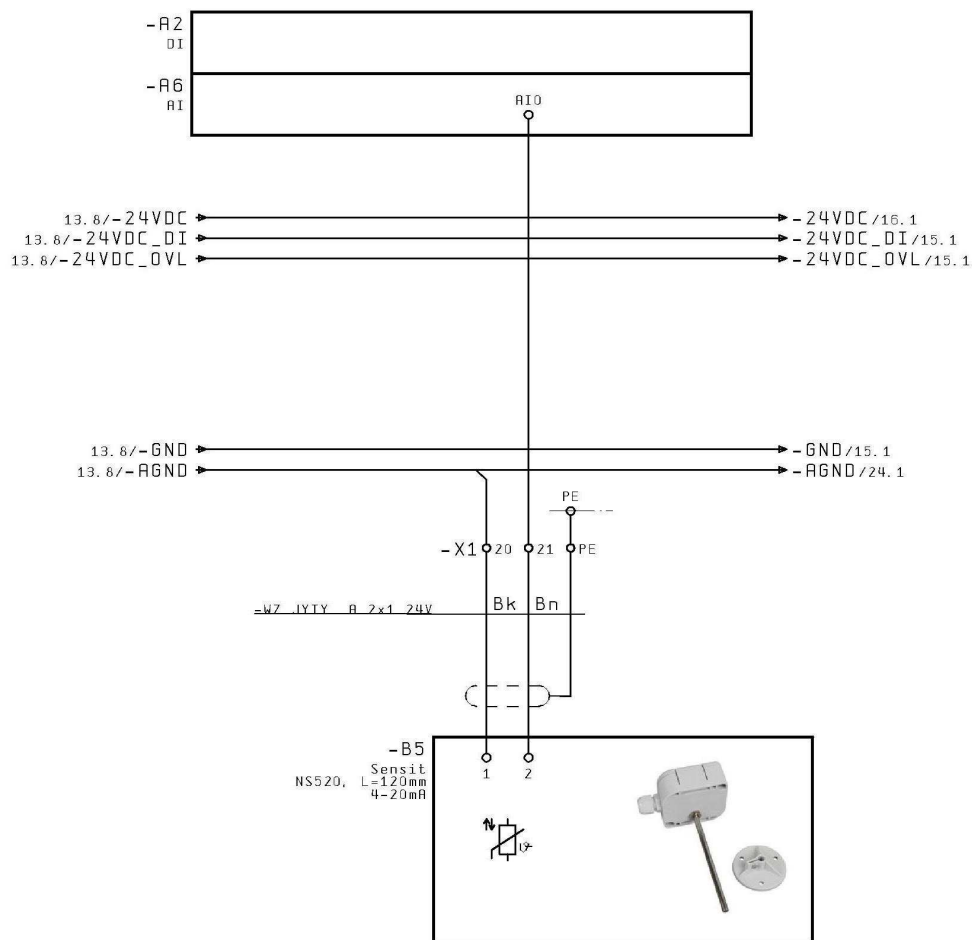
Na měření hladiny se musíme podívat z jiného úhlu. Nejprve si určíme jakou kapalinu v našem případě chceme měřit. V první řadě to bude různorodý kvas, který obsahuje velmi odlišné složky a v dalším případě měříme hladinu destilátu.

Pro analogová čidla je dobré vědět, že by se měli připojovat k AGND na analogové kartě. Protože každé sepnutí relátka ovlivňuje měřenou hodnotu. Bohužel toto se v manuálech neuvádí a člověk dlouho přemýšlí, proč někdy čidla nefungují tak, jak mají.

## 6. 8. 3 Programování

Všem vstupům a výstupům přiřadím byte karty a byte vstupu (výstupu), které pak přiřadíme k proměnné. S touto proměnnou pak dále můžeme pracovat. Pozor při zpracovávání provozního deníku, kde musíme opět přiřadit proměnnou správnému vstupu (výstupu).

Přes analogovou vstupní kartu získáváme hodnoty z měřících prvků, které mohou mít vstupní rozsahy 0 až 5 V<sub>ss</sub>, 0 až 10 V<sub>ss</sub>, 0 až 20,08 mA<sub>ss</sub>, Ni1000. Tyto jednotlivé rozsahy převedeme na skutečné hodnoty přepočtem. Schéma zapojení čidla viz. Obr. 6.9.



Senzor teploty nad kotlem

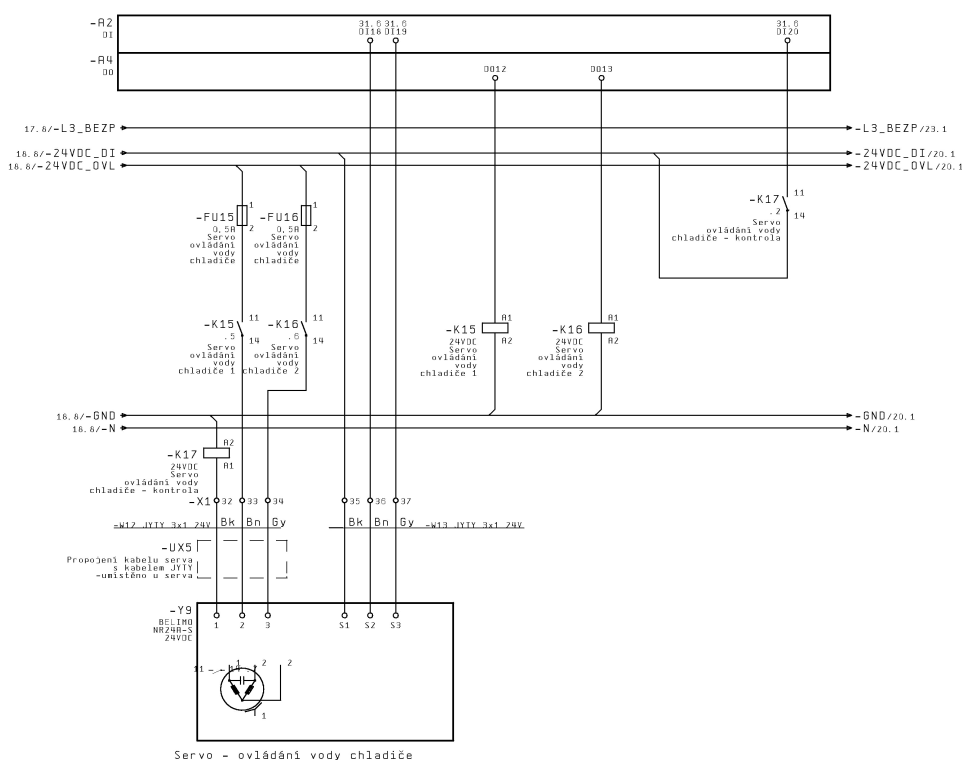
Obr. 6.9: Zapojení teplotního čidla Ni1000 [autor]

V katalogovém listu každého čidla najdeme měřicí rozsah a rozsah výstupních hodnot. Někdy se v katalogovém listu uvádí i srovnávací tabulka, podle které můžeme korigovat převod.

Důležitým regulačním členem je servo. V mém případě jsem použil serva na 24 VDC jak je vidět na obr. 6.10. Použitím nízkého napětí se zvyšuje bezpečnost, nemusíme prvky pospojovat a při rozpojení bezpečnostní smyčky stále můžeme serva ovládat a případně ventil uzavřít. Servo se většinou zapojuje přes dva ovládací kontakty (standardně) nebo přes jeden ovládací kontakt (tzv.

nucené nebo mrazové spínání). Při nuceném spínání, připojením napětí na ovládací kontakt dojde servo do jedné krajní polohy, při odpojení napětí servo přejede do druhé polohy, jeden směr je nadřazen. Při standardním ovládní, přivedením napětí na jeden kontakt otáčíme servem na jednu stranu a naopak, tak můžeme plynule ovládat otáčení serva a tím plynule ovládáme otevírání (zavírání) ventilu. V programu je plynulé ovládní řešeno přes časové spínání digitálního výstupu. Každé servo má určitý čas, který potřebuje k „přejetí“ z jedné krajní polohy do druhé. Chceme-li zajistit přesné ovládní, vycházíme z jedné krajní polohy. U speciálních případů je možné sledovat polohu serva pomocí snímače. Pro kontrolu krajních poloh se k servu přikupují spínací kontakty.

Je důležité sledovat i napájení jednotlivých prvků. To se nejčastěji dělá přes pomocné kontakty u jističe a přes ně se přivede požadované napětí na vstup DI. U pojistek se tento kontakt nahradí pomocí relé. V případě ztráty napětí (shození jističe), detekujeme chybu automaticky, který zahlásí chybové hlášení na obrazovce.



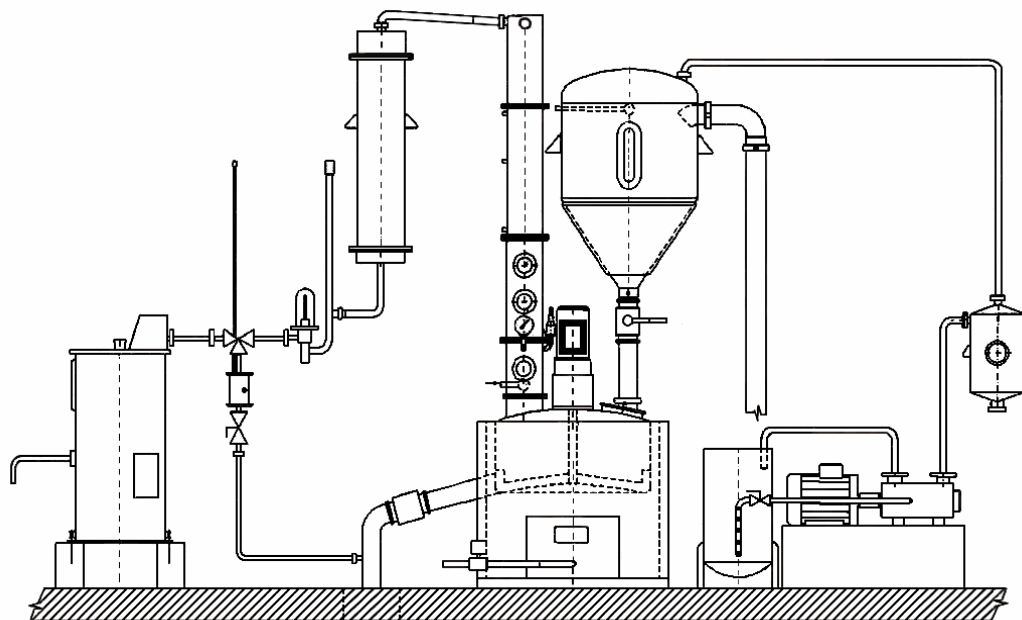
Obr. 6.10: Zapojení serva a pomocných kontaktů [autor]

#### 6. 8. 4 Ukázková aplikace

Ukázkovou aplikací třetího projektu je plně zautomatizovaná palírna. Technologické schéma je zobrazeno na Obr. 6.11. Dříve byla ovládaná ručně, za několik málo týdnů však proběhla rekonstrukce a dnes již je vše odzkoušeno a funguje bez problémů 12h denně.

Zautomatizován nebyl jen proces destilace, ale i napouštění a příprava kvasu. K automatu jsme připojili i stávající rekuperaci vody, plynový hořák, vývěvu pro napouštění zásobníku kvasu a na konci i měření hladiny destilátu.

Pro samotnou aplikaci je velmi důležité sledovat teploty a tlaky jak v deflegmátoru, tak i na chladiči. Teplota a tlak jsou nejdůležitějšími ukazateli výrobního procesu, petit se na základě jejich měření reguluje jak sprcha nad kotlem, tak i sprchy nad deflegmátorem a chladičem. Není jednoduché regulovat takovýto proces, který je velmi závislý na zkušenostech lihovárníka. Je mnoho nuancí, které ovlivňují subjektivní chutě a tyto chutě jsou závislé na přesné regulaci.



Obr. 6.11: schéma technologické části palírny [autor]

## Závěr

Výsledkem praktické části je funkční prototyp, v práci nazván jako druhý prototyp, na ovládání světel s PWM modulací a projekt ovládaný automatem, připojeným na PC. Využití druhého prototypu je na fantazii každého a možnost v nejrůznějších zapojení je obrovská. Stačí jen zapojit na svorky spotřebič a pomocí PWM modulace a jednoduchého programu můžeme spotřebič ovládat.

Projekt je jednoúčelový, řeší automatizaci výrobního procesu, pálení lihovin, ale kombinací měření a řízení jednotlivých součástí můžeme regulovat jakýkoli proces.

V práci jsem se snažil shrnout problematiku ovládání a nastítnit řízení výkonových prvků. Vyzdvihl jsem nejdůležitější a nejzákladnější části ovládání, řízení a zapojení zařízení používaných v automatizaci atd. S ohledem na délku textu, jsem nemohl uvést všechna zapojení a všechny problémy se kterými jsem se při návrhu setkal.

## Seznam použité literatury

- [1] Automatizace. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, [cit. 2010-01-24]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Automatizace>>.
- [2] ŠVARC, Ivan. skripta *Základy automatizace*, VUT v Brně fakulta strojního inženýrství, 2002
- [3] Automatizace. *Navajo : otevřená encyklopedie* [online]. : Navajo Foundation, [cit. 2010-02-24]. Dostupné z WWW: <<http://automatizace.navajo.cz/>>.
- [4] AMIT : Přední český výrobce řídicích systémů, počítačů a elektroniky pro průmyslovou automatizaci [online]. 2009 [cit. 2009-12-19]. Servisní programy. Dostupné z WWW: <[http://www.amit.cz/cz/products/servis\\_sw.htm](http://www.amit.cz/cz/products/servis_sw.htm)>.
- [5] HRACH, Rudolf. *Počítačová fyzika I*. Ústí nad Labem: PF UJEP, 2003. s. 137. ISBN 80-7044-521-1
- [6] ZIMA, Michal *Historie počítačů v Československu*. In *Historie počítačů v Československu* [online]. [s.l.] : [s.n.], 2008 [cit. 2009-09-05]. Dostupné z WWW: <<http://www.skola-radotin.cz/filemanager/files/file.php?file=8958>>.
- [7] MATOUŠEK, David. *USB prakticky s obvody FTDI 1.díl*. 1. vydání. Praha : BEN, 2003. Popis obvodu FT232BM, s. 272+CD. ISBN 80-7300-103-9.
- [8] ASIX : *To nejlepší z elektroniky...* [online]. c1991-2009 [cit. 2009-16-12]. Vývojová deska UDS1. Dostupné z WWW: <<http://www.asix.cz/a6uds1.htm>>.

- [9] AMIT : *Přední český výrobce řídicích systémů, počítačů a elektroniky pro průmyslovou automatizaci* [online]. 2009 [cit. 2010-02-12]. DetStudio - Návrhové prostředí. Dostupné z WWW: <[http://www.amit.cz/cz/products/param\\_sw.htm](http://www.amit.cz/cz/products/param_sw.htm)>.
- [10] ASIX : *To nejlepší z elektroniky...* [online]. c1991-2007 [cit. 2010-01-23]. Součástky FTDI pro USB. Dostupné z WWW: <<http://www.asix.cz/download/ftdi/ds232b12.pdf>>.
- [11] AMIT : *Přední český výrobce řídicích systémů, počítačů a elektroniky pro průmyslovou automatizaci.* [online]. 2008 [cit. 2009-10-12]. Fotogalerie. Dostupné z WWW: <<http://www.amit.cz/cz/gallery/dm-ai12.htm>>.
- [12] AMIT : *Přední český výrobce řídicích systémů, počítačů a elektroniky pro průmyslovou automatizaci* [online]. 2009 [cit. 2010-02-12]. Grafické terminály. Dostupné z WWW: <<http://www.amit.cz/cz/products/grterminals.htm>>.
- [13] AMIT : *Přední český výrobce řídicích systémů, počítačů a elektroniky pro průmyslovou automatizaci* [online]. 2009 [cit. 2010-02-14]. Vzdálené V/V s protokolem ARION. Dostupné z WWW: <[http://www.amit.cz/cz/products/io\\_arion.htm](http://www.amit.cz/cz/products/io_arion.htm)>.
- [14] AMIT : *Přední český výrobce řídicích systémů, počítačů a elektroniky pro průmyslovou automatizaci* [online]. 2009 [cit. 2009-12-14]. Vzdálené V/V - dokumentace. Dostupné z WWW: <[http://www.amit.cz/docs/cz/dm-xx/dm-ai12\\_d\\_cz\\_102.pdf](http://www.amit.cz/docs/cz/dm-xx/dm-ai12_d_cz_102.pdf)>.

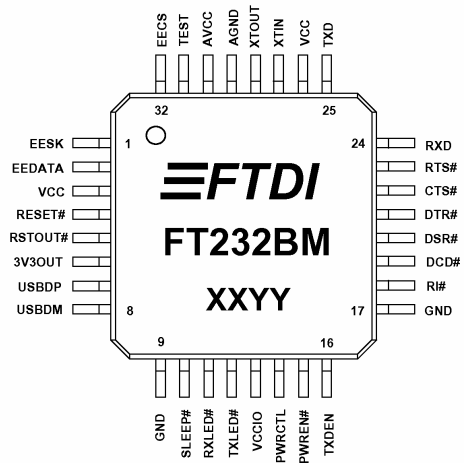


- [15] AMIT : *Přední český výrobce řídicích systémů, počítačů a elektroniky pro průmyslovou automatizaci* [online]. 2009 [cit. 2009-12-14]. Vzdálené V/V - dokumentace. Dostupné z WWW: <[http://www.amit.cz/docs/cz/dm-xx/dm-do18\\_d\\_cz\\_102.pdf](http://www.amit.cz/docs/cz/dm-xx/dm-do18_d_cz_102.pdf)>.
- [16] AMIT : *Přední český výrobce řídicích systémů, počítačů a elektroniky pro průmyslovou automatizaci* [online]. 2009 [cit. 2009-12-14]. Vzdálené V/V - dokumentace. Dostupné z WWW: <[http://www.amit.cz/docs/cz/dm-xx/dm-di24\\_d\\_cz\\_102.pdf](http://www.amit.cz/docs/cz/dm-xx/dm-di24_d_cz_102.pdf)>.
- [17] AMIT : *Přední český výrobce řídicích systémů, počítačů a elektroniky pro průmyslovou automatizaci*. [online]. 2008 [cit. 2009-10-12]. Fotogalerie. Dostupné z WWW: <<http://www.amit.cz/cz/gallery/apt3100.htm>>.

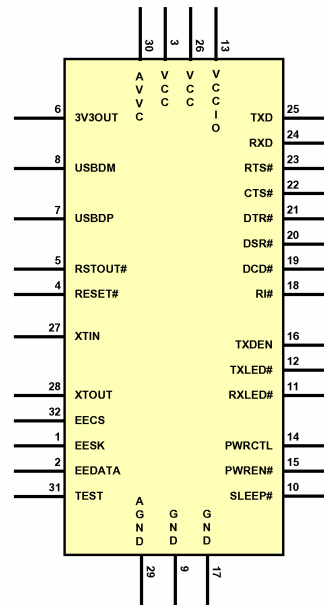
# Obrázkové přílohy:

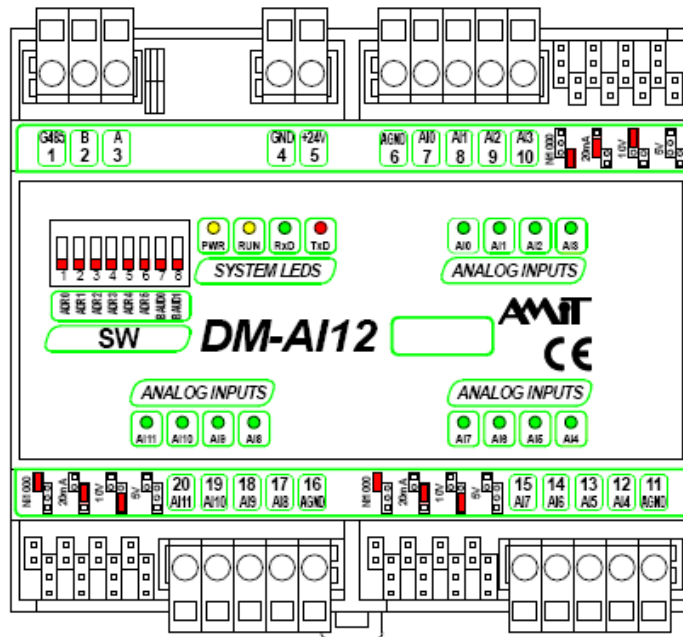
## 4.0 Device Pin-Out

**Figure 1  
Pin-Out  
(LQFP-32 Package )**

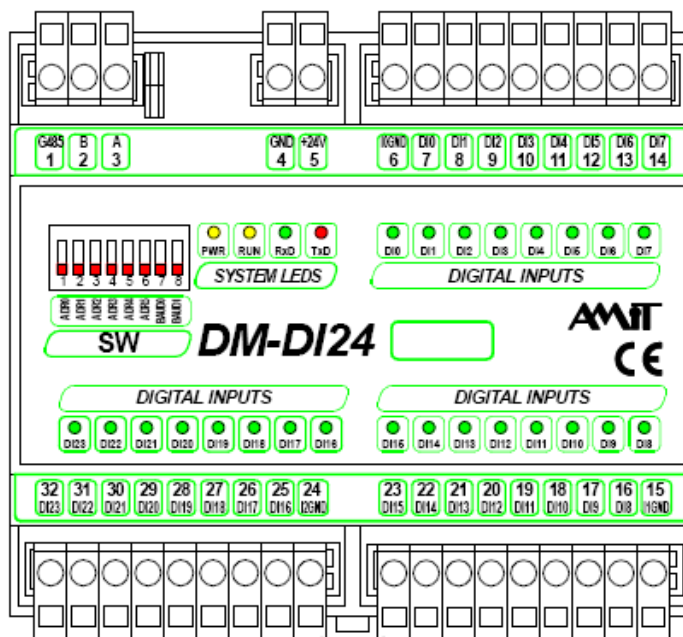


**Figure 2  
Pin-Out  
(Schematic Symbol)**

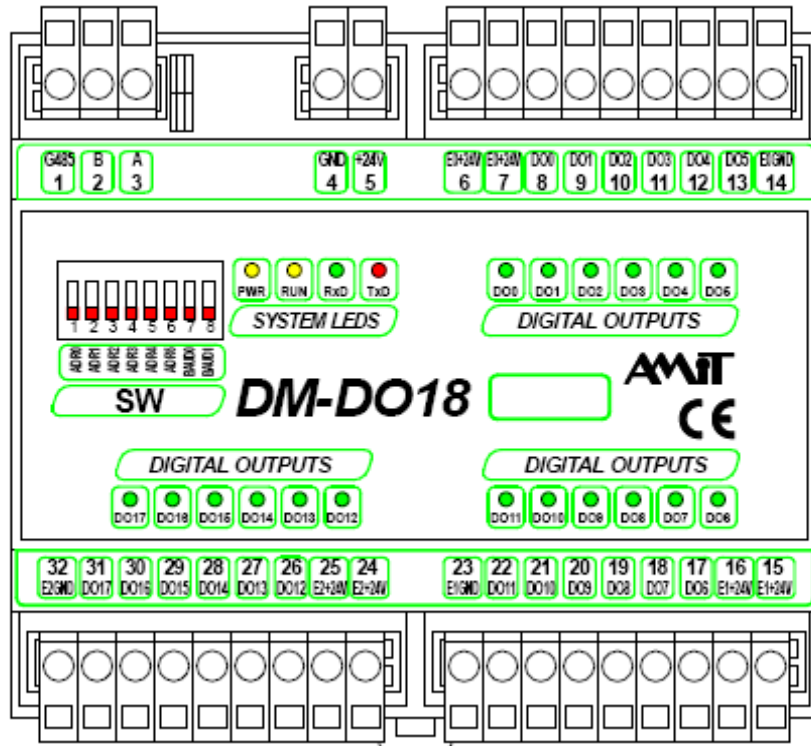




Příloha 2.: AMIT DM-AI12 [14]



Příloha 3.: AMIT DM-DI24 [16]



Příloha 4.: AMIT DM-DO18 [15]