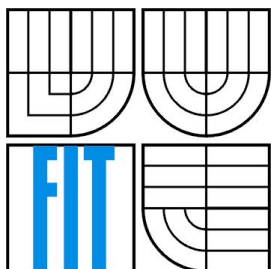




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

***PROGRAMOVÉ VYBAVENÍ PRO KOMUNIKACI A
NASTAVENÍ JEDNOTKY PRO SBĚR DAT JSD600***

SOFTWARE FOR COMMUNICATION WITH DATA COLLECTION UNIT JSD600

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

MICHAL PAJGRT

VEDOUCÍ PRÁCE

SUPERVISOR

Doc. Ing. ZDENĚK KOTÁSEK, CSc.

BRNO 2007

Licenční smlouva

Licenční smlouva je uložena v archivu Fakulty informačních technologií Vysokého učení technického v Brně.

Programové vybavení pro komunikaci a nastavení jednotky pro sběr dat JSD600

Vedoucí:

Kotásek Zdeněk, doc. Ing., CSc., UPSY FIT VUT

Přihlášen:

Pajgrt Michal

Zadání:

1. Seznamte se se základními termodynamickými vlastnostmi vody a vodní páry z hlediska principů měření energie.
2. Seznamte se s jednotkou pro sběr dat JSD600 firmy Smart s. r. o. a s jejím nasazením jako měřiče energie dodané vodní párou.
3. Seznamte se s komunikačním protokolem PC - JSD600 a ověřte jeho funkčnost na přenosu zkráceného stavu měřidla.
4. Vytvořte aplikaci pro MS Windows, která dovolí následující operace s jednotkou: načtení aktuálního stavu z jednotky a jeho zobrazení, úpravu měřících parametrů jednotky a jejich zpětné programování, vizualizaci měřených dat v reálném čase, uložení nastavení stanice a naměřených dat do souboru s možností jejich zpětného načtení a vyhledání jednotek zapojených v síti.
5. Vytvořte kontextový uživatelský "HELP".
6. Diskutujte možnosti použití jednotky a uveďte příklady použití jednotky v praxi.

Část požadovaná pro obhajobu SP:

Splnění prvních tří bodů zadání.

Kategorie:

Počítačová architektura

Implementační jazyk:

Borland Delphi

Operační systém:

Windows

Literatura:

- Leitner, J.: Termodynamika materiálů: stavové chování a termodynamické vlastnosti čistých látek, skripta ÚIPL VŠCHT
- Buchalcevoá, A.: Vývoj aplikací v DELPHI, Praha, VŠE, ISBN 80-245-0150-3
- Bejček, J.: Přehled metod a snímačů měření průtoku, In: Metrologie průtoku, Brno 2005, s. 105 - 138, ISBN 80-239-4312-X

Abstrakt

Práce se zabývá popisem jednotky JSD600 firmy Smart, spol. s r.o. Jedná se o jednotku pro matematické zpracování, měření a záznam dat využívané především v oblasti měření energie dodané vodní parou. Práce stručně představuje jednotku z pohledu základní koncepce, vysvětluje princip řešení uchování požadovaných dat jednotkou. Dále shrnuje základní poznatky z oblasti měření energie, měření průtoku a stavů vodní páry, nezbytné pro porozumění problematice měřičů tepla. Přináší ucelený přehled komunikačního protokolu jednotky od základní definice struktury paketu až po celkový přehled všech používaných paketů. Druhá část práce se věnuje především popisu vlastní implementace aplikace pro nastavení a konfiguraci jednotky, která tvoří hlavní část mé diplomové práce.

Klíčová slova

JSD600, Smart, spol. s r.o., matematická jednotka, jednotka pro měření a uchování dat, měření tepla dodaného vodní parou, měření energie, záznam dat, měření průtoku tekutin, stavy páry, přehřátá pára, sytá pára, přímá metoda, nepřímá metoda, komunikační protokol, registry stanice, zkrácený stav měřidla, aplikace pro nastavení parametrů jednotky

Abstract

This diploma thesis describes a JSD600 unit intended for mathematical processing, measuring and data recording. The JSD600 unit is highly intended for industrial measuring of water steam energy delivery.

First part of the work focuses on basics of the JSD600 unit detailing and describing main structures and principles for relevant data storing.

Next part summarizes some basic facts and knowledge from the area of industrial liquid flow, energy measuring and describes water steam states. All these points within this range are needed for dilemma understanding.

Last task deals with complete list of communication datagrams, which have been used for communication with JSD600 unit, including a main packet of structure descriptions.

The second part of this dissertation brings the unit setup application description including some implementation details of the most interesting parts.

Keywords

JSD600, Smart, spol. s r.o., mathematical unit, JSD600 unit for measuring mathematical processing and data recording, water steam heat measuring, energy measuring, data recording, liquid flow measuring, water steam states, superheated water steam, saturated water steam, direct method, indirect method, communication protocol, station registers, short gauge state, unit setup application

Citace

Michal Pajgrt: Programové vybavení pro komunikaci a nastavení jednotky pro sběr dat JSD600, diplomová práce, Brno, FIT VUT v Brne, 2007

Programové vybavení pro komunikaci a nastavení jednotky pro sběr dat JSD600

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením doc. Ing. Zdeňka Kotáska, CSc.

Další informace mi poskytl zejména p. Ing. Vlastimil Jaroš z firmy Smart, spol. s r.o.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Michal Pajgrt
10. května 2007

Poděkování

Oběma výše zmíněným bych rád poděkoval za neocenitelnou pomoc, vstřícný přístup a čas, který mi věnovali při řešení projektu a při sepisování technické zprávy.

© Michal Pajgrt, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	6
1 Úvod.....	7
2 Jednotka JSD600.....	8
2.1 Popis jednotky	8
2.2 Modularita zařízení.....	8
2.3 Komunikace s jednotkou.....	9
2.4 Paměť stanice.....	9
2.5 Jednotka JSD600 jako měřič tepla.....	11
2.6 Programování stanice.....	11
3 Základy teorie měření energie.....	13
3.1 Pára a její stavy.....	13
3.2 Hystereze a pásmo sytosti.....	14
3.3 Základy měření průtoku.....	15
4 Komunikační protokol.....	19
4.1 Základní použité datové typy	19
4.2 Základní schéma datového paketu.....	20
4.3 Průběh komunikace.....	22
4.4 Zabezpečení.....	22
4.5 Jednotlivé komunikační pakety.....	23
5 Implementace programu.....	29
5.1 Neformální specifikace projektu.....	29
5.2 První fáze projektu.....	31
5.3 Druhá fáze projektu.....	41
5.4 Kontextová nápověda k aplikaci.....	54
5.5 Parní výpočty.....	55
6 Závěr.....	56
Literatura.....	58
Seznam příloh.....	59

1 Úvod

Poslední rok spolupracuji jako externí zaměstnanec s firmou Smart, s.r.o, specializující se na vývoj průmyslových zařízení v oblasti měření a regulace. Jedním z mých prvních úkolů bylo vytvořit aplikaci pro platformu Windows, která by umožňovala komunikaci s jedním z prodávaných zařízení, jednotky pro měření a matematické zpracování dat JSD600.

Pro potřeby programování stanice již několik let existuje program, umožňující základní programovací úkony. Jedná se však o program pracující v prostředí MS-DOS, což se ukázalo jako velký problém při migraci zákazníků k systémům Windows 2000 a Windows XP. Tyto systémy přistupují k sériovým portům počítače v chráněném režimu a umožňují jejich využití pouze prostřednictvím WinAPI funkcí.

Dalším problémem se ukázalo být časování komunikace, takže ani po uvolnění portů specializovanými programy nebylo možné se stanicí uspokojivým způsobem komunikovat. Jelikož starý program pro MS-DOS měl řadu nedostatků, chybějících funkcí a bylo prakticky nemožné a neperspektivní do něj zasahovat, rozhodlo se o přepsání a znatelném rozšíření programu pro platformu MS Windows 2000 a vyšší. Vznikl tak nový program pro komunikaci s jednotkou.

Ve své práci popisuji postup prací, které vedly k vytvoření tohoto programu. Kapitoly jsou řazeny chronologicky ve stejném sledu, v jakém jsem se sám musel seznámit s problematikou.

Práce nejprve popisuje jednotku JSD600 jako zařízení. Zaměřuje se na seznámení čtenáře s obecnými principy její funkce v kontextu jednoho možného způsobu nasazení jako měřiče tepla dodaného vodní parou. Dále uvádí některé nezákladnější teoretické předpoklady z oblasti měření průtoku tekutin a stavů páry v takovém rozsahu, aby čtenář pochopil souvislosti s jednotlivými prvky nastavení jednotky. Další kapitola pojednává o komunikačním protokolu použitém v jednotce, včetně detailního popisu použitých paketů. Tyto první čtyři kapitoly byly zároveň náplní mého semestrálního projektu.

V páté kapitole jsem se soustředil především na vývoj samotné aplikace a na popis grafického uživatelského prostředí spolu s popisem některých implementačních detailů programu. Práce nemá návaznost na ročníkový projekt.

Vzhledem ke značné složitosti a rozsáhlosti aplikace i funkcí jednotky si práce neklade za cíl detailně popsat celý postup implementace aplikace, ale snaží se nastínit hlavní problémy, které při řešení projektu vznikly a které bylo třeba řešit.

2 Jednotka JSD600

Kapitola stručně představuje jednotku JSD600 jako zařízení. Snaží se v krátkosti popsat zaměření jednotky a její místo na trhu průmyslových měřicích zařízení. Dále pak vysvětluje základní terminologii možnosti jednotky.

2.1 Popis jednotky

Jednotka JSD600 je osmikanálové zařízení pro měření a záznam dat a fyzikálních veličin používaných zejména v energetice, průmyslu a laboratorní technice. Její primární určení je do oblasti měření a zpracování tepla dodaného vodní parou a měření tepla vratného kondenzátu, je však schopna měřit a zaznamenávat jakékoliv veličiny měřené standardizovanými čidly připojenými na vstupní kanály. Typickým příkladem nasazení tohoto zařízení jsou například teplárny, prodávající energii ve formě páry. Jednotka v takovémto případě plní podobnou funkci jako průtokoměr na vodním potrubí v každé domácnosti.

2.2 Modularita zařízení

Vzhledem k modulární koncepci zařízení jsou však možnosti jeho použití značně širší. V současné době existuje k zařízení několik modulů schopných sběru dat pomocí senzorů založených na různých principech a technologiích. Vstupy jednotky tedy mohou být veškerá čidla poskytující standardizovaný proudový, odporový, napěťový nebo frekvenční vstup případně kombinovaná průmyslová zařízení využívající rozhraní protokolu HART. Výstupní moduly pak mohou mít charakter analogového proudového nebo napěťového výstupu, reléových nebo tranzistorových modulů stavového nebo impulsního výstupu. Je tedy pokryta většina principů používaných v praxi pro sběr dat z externích čidel a komunikaci s inteligentními zařízeními.

Vstupní moduly	Výstupní moduly
Proudový 0-20mA aktivní	Dvojitý relé modul
Proudový 0-20mA pasivní	Dvojitý tranzistor. modul
Odporový	Dvojitý analog. Modul
Napěťový 0-50mV, 0-1V, 0-10V	
Frekvenční 5-2000Hz	
Impulsní 0-50 imp/s	
Protokol HART	

Tab. 2.1: Vstupní a výstupní moduly

2.3 Komunikace s jednotkou

Jednotka JSD600 je plně autonomní zařízení a ke své práci tedy nepotřebuje nadřazený počítač. Pro sběr základních naměřených dat je možné použít dvouřádkový displej na jednotce. Z důvodů omezeného prostoru je ale možné takto vyčíst pouze informace ze základních registrů jednotky. Pro ostatní komplexnější data a programování nastavení měřících parametrů jednotky je nutné použít nadřazené PC s příslušným programem.

Pro připojení jednotky k PC lze využít rozhraní RS232, RS485 nebo infračerveného čidla pro bezkontaktní přenos dat. V jednu chvíli může být na lince připojeno v lokální síti až 32 jednotek. Ovládací program pak pomocí adresace osloví jednotku požadovanou uživatelem.

2.4 Paměť stanice

Kapitola pojednává o struktuře interní paměti stanice, jejím rozdělení a využití pro různá zaznamenávaná data.

Paměť kanálů

Zařízení disponuje pamětí o velikosti 32 nebo 64kB pro ukládání naměřených veličin. Uživatel má možnost pro každý kanál definovat vzorkovací interval uložení aktuálně změřené veličiny a množství alokované paměti pro tuto veličinu. Z těchto parametrů se pak odvíjí doba, po kterou je jednotka schopna data udržet. Po zaplnění celé kapacity alokované paměti pro kanál se tato chová jako cirkulární buffer. Nejstarší data tak začínají být nenávratně přepisována novými. Mimo paměť kanálů disponuje zařízení ještě sdílenou pamětí hlášení stanice a pevně definovanou pamětí archivů.

Systemová hlášení stanice

Paměť hlášení stanice zaznamenává systémová hlášení jako například výpadky měření, výpadky proudu, zahájení měření nebo chybové stavy vstupních veličin. Podle těchto hlášení je tedy jednoznačně identifikovatelné, kdy stanice neměřila a proč se tak stalo. Mělo by se tak zabránit možnosti vypnutí napájení jednotky nebo odpojení vstupních čidel ve snaze ovlivnit naměřené hodnoty. Tato paměť je sdílená s pamětí kanálů a uživatel může tedy ovlivnit množství systémových hlášení, které je možné zaznamenat.

Paměť archivů

Jednotka JSD600 disponuje celkem 15-ti tzv. 'totalizéry měřiče tepla'. Jedná se o soubor registrů s celkovým množstvím několika měřených hodnot od počátku měření stanice. Strukturu všech totalizérů zobrazuje následující tabulka.

Q celk	celkové množství tepla	GJ
Q1 sucha	teplo 1. okruhu dodané v suché páře	GJ
Q1 mokra	teplo 1. okruhu dodané v mokré páře	GJ
Q2 sucha/kond	teplo 2. okruhu dodané v suché páře	GJ
	teplo 2. okruhu vratného kondenzátu	GJ
Q2 mokra	teplo 2. okruhu dodané v mokré páře	GJ
Q1 kond	teplo vratného kondenzátu 1. okruh	GJ
M1 sucha	množství dodané suché páry 1. okruhu	tun
M1 mokra	množství dodané mokré páry 1. okruhu	tun
M2 sucha/kond	množství vratného kondenzátu 2. okruhu	tun
	množství dodané suché páry 2. okruhu	tun
M2 mokra	množství dodané mokré páry 2. okruhu	tun
M1 kond	množství vratného kondenzátu 1. okruhu	tun
Cas1 mokra	čas dodávky mokré páry v 1. okruhu	hod
Cas1 m.p.	čas kdy pára v 1. okruhu nebyla měřena	hod
Cas2 mokra	čas dodávky mokré páry v 2. okruhu	hod
Cas2 m.p.	čas kdy pára v 2. okruhu nebyla měřena	hod
Cas Vyp.	čas kdy měřidlo bylo vypnuto	hod

Tab. 2.2: Totalizéry tepla

Pro uchování dat totalizérů slouží právě paměť archivů. Skládá se ze 24 měsíčních archivů, 8 denních archivů a jednoho uživatelského archivu. Každý den o půlnoci a na konci každého kalendářního měsíce jsou tedy aktuální totalizéry uloženy do paměti. Tato data nejsou (s výjimkou uživatelského registru) programově nulovatelná a pouze opětovné naprogramování stanice smaže archivy i totalizéry.

2.5 Jednotka JSD600 jako měřič tepla

Primární určení jednotky JSD600 je pro měření tepla dodaného vodní parou a tepla vratného kondenzátu ve třech nezávislých okruzích. Osm vstupních kanálů zařízení se rozdělí následovně:

- První okruh tvoří kanály 1, 2 a 3. Pro měření tepla přímou metodou jsou pak na jejich vstupu čidla teploty, tlaku a průtoku
- Druhý okruh tvoří kanály 4, 5 a 6. Může být použit shodně s prvním nebo třetím okruhem (záleží na nastavení)
- Třetí okruh tvoří kanály 7 a 8. Slouží pro měření tepla vratného kondenzátu na základě teploty kondenzátu a průtoku

Jednotka po každém měření kontroluje validitu vstupních veličin. Jejich rozsah musí splňovat uživatelem definovaná omezení. Protože pára může nabývat různých stavů ve kterých se její energetická kapacita liší (viz kapitola 3. Základy teorie měření energie), musí jednotka podle vstupních parametrů určit o jaký stav se jedná a zařadit aktuální hodnotu do správného registru mokré nebo suché páry, příp. do registru *mimo parametry*.

2.6 Programování stanice

Stanice je dodávána ve dvou možných variantách:

Stanovené měřidlo

Pokud zákazník požaduje, aby stanice sloužila jako stanovené a ověřené měřidlo, jsou její parametry přednastaveny výrobcem podle způsobu nasazení zařízení a jednotka je uzamčena proti opakovanému přeprogramování. Takto nastavená stanice je zkalibrována a putuje na ověření do laboratoří Českého metrologického institutu v Brně, kde se testuje, zda měření probíhají v souladu s platnými předpisy a normami a zda jsou dodrženy předepsané hodnoty přesnosti. Každá takto zkalibrovaná jednotka je následně zapečetěna.

Pouze měřící jednotka

V případě, že zákazník nepožaduje metrologické ověření zařízení, je jednotka pouze zkalibrována (příp. nastavena podle přání zákazníka). Zákazník má pak možnost zařízení programovat podle vlastních pravidel a potřeb.

Z uvedeného tedy vyplývá, že programování stanice se týká pouze těch stanic, která neslouží jako stanovené měřidlo, neprošly metrologickým ověřením a nejsou tedy uzamčeny proti

programování.

Nastavení stanice uživatel provádí pomocí připojeného PC. Po nadefinování vlastností jednotlivých kanálů, vlastností měřidla a stanice je celý soubor nastavení najednou odeslán do stanice. Změna většiny parametrů tedy vyžaduje úplné přeprogramování stanice.

Obvyklý postup při změně parametrů je následující:

1. načtení aktuální konfigurace ze zařízení
2. modifikace požadovaných nastavení
3. opětovné naprogramování stanice pozměněnými daty

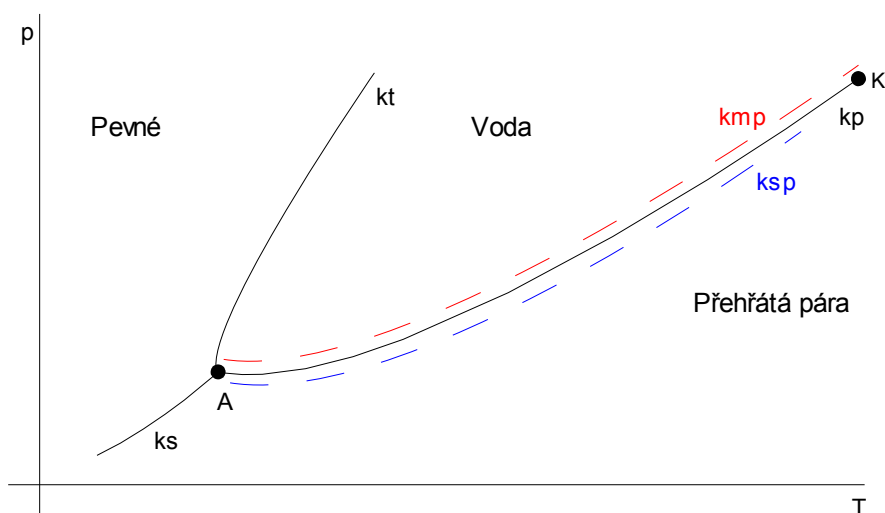
Informace o jednotce byly čerpány z [1].

3 Základy teorie měření energie

Tato kapitola si klade za cíl nastínit nejzákladnější principy používané v praxi měření energie. Z důvodu značné komplexnosti problému se jedná pouze o letmé seznámení s fakty potřebnými k pochopení tématu.

Ze základních pojmů je nutné definovat zejména termín *tekutina*, který zahrnuje kapaliny, páry i plyny. Pod pojmem *průtok* pak rozumíme množství tekutiny proteklé daným průřezem za časovou jednotku.

3.1 Pára a její stavy



- ks ... sublimační křivka
- kt ... křivka tání
- kp ... křivka sytých par
- kmp ... pásmo mokré páry
- ksp ... pásmo suché páry
- A ... trojný bod
- K ... kritický bod

Obr 3.1: Fázový diagram vody

Pára se může vyskytovat v několika různě energeticky bohatých stavech. Odebírat páru s nižším energetickým obsahem samozřejmě není z pohledu koncového zákazníka žádoucí. Zařízení musí tedy zaznamenat nejen množství proteklé páry, ale také vyhodnotit její stav a kvalitu v daném okamžiku [2].

Při měření energie dodané vodní parou rozlišujeme tři stavy páry

1. **syťá pára** – jedná se o páru, která má teplotu přesně odpovídající teplotě kapaliny a je s ní v termodynamické rovnováze. Ve stavovém diagramu syťou páru reprezentuje křivka syťých par
2. **mokrý pára** – Pára o nižší teplotě než odpovídá termodynamické rovnováze s vodou. Při tomto stavu jsou v páře přítomny jemné kapičky vody. Ve stavovém diagramu ji reprezentuje oblast mezi křivkou pásma mokré páry a mezi křivkou syťých par.
3. **suchý (přehřátý) pára** – pára o vyšší teplotě než odpovídá termodynamické rovnováze s vodou. Při tomto stavu pára již neobsahuje žádnou vodu. Ve stavovém diagramu ji reprezentuje oblast mezi křivkou syťých par a křivkou suché páry. [2], [3]

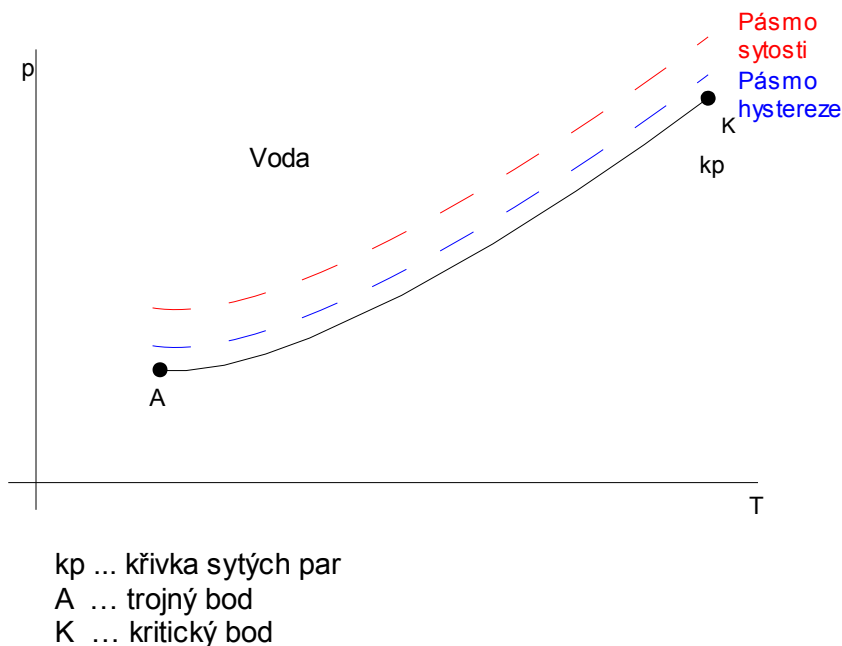
Vysoce nežádoucí stav je zejména pára mokrá. Kapičky vody, které obsahuje, nesou znatelně méně energie než samotná pára, přitom jejich hmotnost znatelně převyšuje hmotnost páry, což se negativně projeví na změřeném průtoku. Další nevýhodou tohoto stavu je potenciální nebezpečnost vodních kapiček v samotných rozvodech a měřicích zařízeních, protože při rychlostech, jakými jsou unášeny, mají nemalé abrasivní schopnosti. Díky těmto schopnostem cestou obrušují vnitřní stranu rozvodů a spolu s uvolněnými mikročástčkami z potrubí se z dlouhodobého hlediska stávají nezanedbatelným nebezpečím hlavně pro citlivá měřicí a koncová zařízení. Podrobnější přehled viz [3], pro doplnění [4].

3.2 Hystereze a pásmo syťosti

Ideální pára z hlediska distribuce je pára přesně na mezi syťosti (kopírující křivku syťých par ve fázovém diagramu). Této podmínky se však v distribučním systému zpravidla nedosahuje a je proto nutné definovat toleranční pásmo, ve kterém bude stav páry stále považován za dostačující. Takovéto pásmo by se dalo označit za jakési *rozšíření* křivky syťých par. [2], [3]

Pásmo hystereze vymezuje oblast, ve které je pára ještě stále považována za páru syťou.

Pásmo syťosti vymezuje oblast nad křivkou syťých par, ve kterém se ještě tekutina považuje za páru (suchou nebo mokrou) [5].



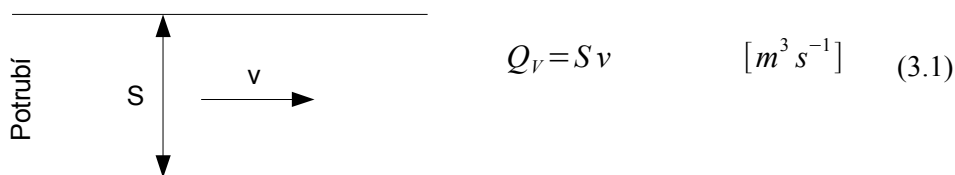
Obr 3.2: Hystereze a pásmo sytosti

3.3 Základy měření průtoku

Kapitola popisuje základní použité principy měření průtoku tak, jak jsou definovány v jednotce JSD600. V literatuře je však možné narazit na různé výklady stejně pojmenovaných metod.

Objemová metoda

Objemový princip měření průtoku se používá ve spojení s průtokoměry, pracujícími na principu měření rychlosti proteklé tekutiny a výpočtu průtoku pomocí známého průřezu potrubí. Průtokoměr měří střední rychlost většinou prostřednictvím lopatkového čidla, které je roztáčeno protékající tekutinou. Tyto otáčky jsou pak lineárně úměrné střední rychlosti proudění tekutiny a naměřené hodnoty se převádějí zpravidla na impulsní výstup.



S průřez potrubí
 v rychlost proudící tekutiny

Obr 3.3: Měření průtoku objemovou metodou

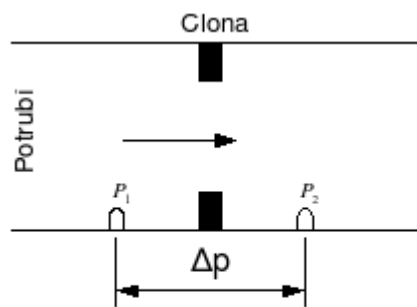
Pro potřeby měření energie je nutné naměřené hodnoty průtoku převést na tzv. *korigované množství*. Tato veličina zohledňuje aktuální hustotu tekutiny získanou aproximací z parních tabulek podle momentální teploty a tlaku tekutiny a zároveň zahrnuje i korekci nepřesnosti měřícího zařízení [6], [7], [8].

$$m_k = Q_v \frac{P_{sk}}{P_{vyp}} \rho \quad [kg \ s^{-1}] \quad (3.2)$$

m_k	<i>korigované množství</i>
Q_v	<i>objemový průtok</i>
P_{sk}	<i>skutečný tlak v potrubí</i>
P_{vyp}	<i>tlak odpovídající referenčnímu tlaku při kalibraci zařízení</i>
ρ	<i>aktuální hustota kapaliny v potrubí</i>

Hmotnostní metoda

Hmotnostního systému měření využívají převážně snímače na bázi difference tlaků v zúženém potrubí. Jedná se o rozsáhlou skupinu s různými typy normalizovaných škrťících členů. Nejčastěji se používá tzv. *clona*. Průtočný průřez je stálý a mění se rychlost proudící tekutiny, která se škrťícím členem převádí na rozdíl tlaku nebo teploty. Z Bernoulliho rovnice (zákonu o zachování energie v tekutinách) vyplývá, že výstup je rovný druhé odmocnině rozdílu tlaků.



P_1, P_2 tlaková čidla

Obr 3.4: Hmotnostní metoda

Pro průtok škrtícím členem platí průtokové rovnice

$$Q_m = \alpha \varepsilon_p \frac{\pi D^2}{4} \sqrt{2(P_1 - P_2)} \quad [kg s^{-1}] \quad (3.3)$$

α průtokový součinitel škrtícího členu (tabulky)
 ε expanzivní činitel škrtícího členu (tabulky)
 $P_1 - P_2$ tlakový rozdíl (ΔP) na škrtícím členu
 ρ hustota měřené tekutiny

Podobně jako v případě objemového principu je nutný převod na *korigované množství*. [6] , [7] , [9]

$$m_k = Q_m \frac{\sqrt{P_{sk}}}{\sqrt{P_{vst}}} \quad [kg s^{-1}] \quad (3.4)$$

m_k korigované množství
 Q_m hmotnostní teplo průtok
 P_{sk} skutečný tlak v potrubí
 P_{vyp} tlak odpovídající referenčnímu tlaku při kalibraci zařízení
 ρ aktuální hustota kapaliny v potrubí

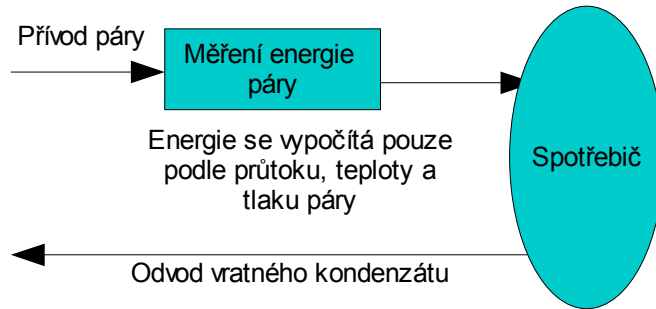
Přímá metoda

Přímá metoda počítá na místě průtokoměru s inteligentním zařízením, které předává přímo hodnoty *korigovaného množství*. Pro takto definovaný kanál již tedy není nutné měřit teplotu a tlak tekutiny ani provádět jakékoliv jiné výpočty [6] , [7] .

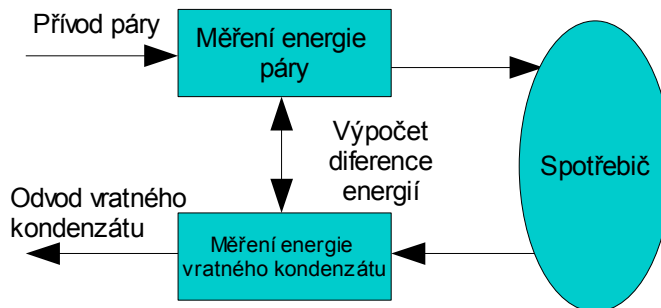
Vratný kondenzát

Při průmyslové distribuci energie ve formě páry je nutné pro její přípravu používat speciálním postupem upravenou vodu. Taková úprava však celý proces prodražuje. Ze strany distributora je proto snaha o vrácení alespoň části zkondenzované páry zpět do výrobního procesu a její opětovné použití. Tato vrácená pára (resp. voda) se nazývá *vratný kondenzát*.

Vratný kondenzát se dá ale stále považovat za nositele energie. V některých aplikacích je vyžadováno jeho měření a zahrnutí této energie do výpočtů [5], [6] .



Obr 3.5: Měření nezahrnuje vratný kondenzát



Obr 3.6: Měření zahrnuje vratný kondenzát

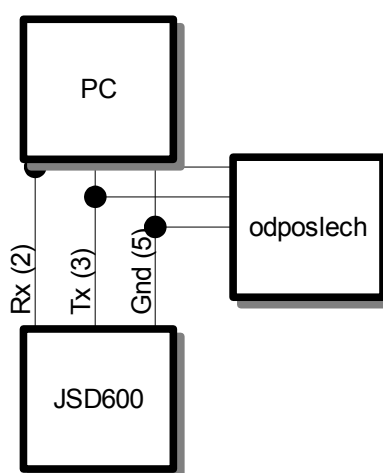
4 Komunikační protokol

Jednotka JSD600 používá vlastní jednoduchý komunikační protokol vyvinutý firmou Smart, s.r.o. Problémem při vývoji programu byla však absence jeho úplné dokumentace. Od počátku byl k dispozici pouze základní popis formátu paketů a významy většiny jednotlivých datových bytů bylo nutné dešifrovat z odposlechnuté komunikace mezi jednotkou a starší verzí softwaru.

V základní konfiguraci jednotka umožňuje komunikaci prostřednictvím protokolu RS-232. Jednoduchou úpravou standardního 9-ti pinového konektoru jsem připojil na komunikační linku třetí počítač, který byl schopen odposlechnout kompletní komunikaci mezi oběma stranami [10].

Pin	Význam
1	DCD – detekce nosného signálu
2	RX – příjem dat
3	TX – vysílání dat
4	DTR – připraven ke komunikaci
5	GND – zem
6	DSR – odpověď na DTR
7	RST – požadavek na data
8	CTS – odpověď na RTS
9	R1 – zvonek

Tab. 4.1: Zapojení konektoru RS232



Obr 4.1: Schéma zapojení odposlechu

Podle známého klíče základního tvaru paketů již nebylo složité rozlišit jednotlivé datové bloky. Konzultací se spoluvůdcem programového vybavení stanice a zkoušením reakcí jednotlivých slabik na různá nastavení jsem poté postupně vytvářel mapu komunikačních paketů a jejich významu.

4.1 Základní použité datové typy

byte – jednoslabiková proměnná (celé číslo)

int – dvouslabiková proměnná (celé číslo)

real – čtyřslabiková proměnná (reálné číslo)

sekundočas – čtyřslabiková proměnná (celé číslo) udávající počet sekund od 1.1.1980

4.2 Základní schéma datového paketu

Komunikace s jednotkou probíhá poloduplexním provozem na sériové lince rychlostí 9600 bps, 8 datových bitů, 1 stop bit, bez parity. Jako nadřazený systém (*master* slouží vždy PC, stanice pouze reaguje na povely nadřazeného zařízení. Vzniklé kolize musí řešit PC znovuvysláním povelů do stanice. V případě zapojení více stanic do sítě je možné pomocí adresace komunikovat se stanicemi selektivně.

Maximální délka zprávy v jednom paketu je 256 slabik. V případě delší zprávy je nutné ji rozdělit na několik kratších.

Protokol rozlišuje dva základní druhy zpráv

1. příkazy pro řízení přenosu dat – jedná se o příkazy navazující a rušící spojení a nastavující parametry přenosu
2. příkazy nadřazené úrovně – vlastní přenos dat stanice

Tvar paketu:



Tab. 4.2: Základní tvar paketu

SOH – startovací byte (pro JSD600 má vždy hodnotu 0x01)

identifikace zprávy – jednotlivé bity určují parametry zprávy podle Tab. 4.3: Význam jednotlivých bitů identifikace zprávy

délka zprávy – obsahuje celkovou délku paketu včetně úvodních a zabezpečovacích slabik (délka zprávy je limitována velikostí 255 bytů)

povel – určuje co vyjadřuje datová část, příp. co se má provést

obsah zprávy – pole obsahuje samotnou datovou část zprávy

[11]

bit		Význam
0	0	řídící povel protokolu
	1	příkazy nadřizené úrovni
1	0	zpráva neobsahuje délku za <Povel>
	1	zpráva obsahuje délku za <Povel>
2	0	první část příkazu
	1	pokračovací část příkazu neobsahuje <povel>
3	0	příkaz je ukončen
	1	příkaz není ukončen, následuje pokračování
4	0	bez žádosti o potvrzení
	1	žádost o potvrzení příkazu
5	0	zpráva není číslována
	1	zpráva je číslována
6	0	rezervováno
	1	rezervováno
7	0	rezervováno
	1	rezervováno

Tab. 4.3: Význam jednotlivých bitů identifikace zprávy

Komunikační protokol definuje jednoduché pakety odpovědí na přijatý paket:

Označení	Hodnota	Význam
ACK	0x06	zpráva byla přijata v pořádku
NACK	0x15	chybná zpráva (chybné zabezpečení...etc)
DLE	0x10	vyčkej, pracuji (používá se na pozdržení komunikace v případě, že jednotka nestihá zpracovat data)

Tab. 4.4: Odpovědi na přijatý paket

4.3 Průběh komunikace

Samotnou komunikaci inicializuje vždy tzv. master, což v případě JSD600 znamená nadřazené PC.

1. Master vyšle žádost o navázání spojení. Žádost nese adresu oslovené stanice pro případ komunikace v síti.
2. Stanice odpoví jedním z paketů odpovědí (viz Tab. 4.4: Odpovědi na přijatý paket).
3. Master vyšle paket s požadavkem nebo daty.
4. Podle povahy paketu stanice odpoví jedním z paketů odpovědí a

- čeká na další příkaz
- odešle požadovaný datagram dat, pro jehož 4. slabiku (povel) platí:

$$\text{povel} = \text{povel_dotazu} + 1$$

5. Master opakuje bod 3 (mezi opakováním nesmí být prodleva delší než 5s, jinak stanice ukončí spojení a je nutné jej znovu navázat).
6. Master ukončí spojení vysláním ukončovacího datagramu.

4.4 Zabezpečení

Jednoduché zabezpečení paketů je realizováno dvojicí zabezpečovacích slabik. Jedná se o poslední dvě slabiky (byty) každého paketu délky n , které mají následující význam:

$$\text{slabika } n \quad \text{slabika}(1) \text{ xor slabika}(2) \text{ xor } \dots \text{ xor slabika}(n)$$

$$\text{slabika } (n-1) \quad (\text{slabika}(1) + \text{slabika}(2) + \dots + \text{slabika}(n)) \text{ mod } 256$$

Pokud se nepodaří přenést příkaz, stanice odpoví paketem NACK (viz Tab. 4.4: Odpovědi na přijatý paket) a nastaví se zpět do režimu příjmu. Na chybné přenosy musí tedy reagovat nadřazená strana tím, že chybný paket odešle do stanice znovu.

Zabezpečovací slabiky jsou součástí každého paketu s výjimkou paketů odpovědí (viz Tab. 4.4: Odpovědi na přijatý paket). V tabulkách popisujících strukturu jednotlivých zpráv nejsou zabezpečovací slabiky z důvodu zjednodušení uváděny, ale při implementaci komunikace se s nimi musí vždy počítat.

4.5 Jednotlivé komunikační pakety

Kapitola přináší kompletní přehled všech paketů definovaných komunikačním protokolem stanice.

Paket navázání spojení

Paket navázání spojení slouží k otevření komunikačního kanálu ke konkrétní stanici. V případě zapojení stanic v síti odpoví pouze ta stanice, jejíž adresa je shodná se slabikou adresy v paketu.

Paket navázání spojení je nutné použít vždy před vlastní komunikací a vždy, pokud je v komunikaci pauza delší než 5 sekund.

Struktura paketu je zřejmá z následujících příloh:

Tabulka 1: Navázání spojení - dotaz

Tabulka 2: Navázání spojení - odpověď

Paket zjištění stavu stanice

Dotaz na stav stanice se používá zpravidla ihned po navázání spojení s jednotkou pro zjištění, zda je stanice schopná vykonat požadovanou operaci. Jeho použití však není nezbytně nutné.

Struktura paketu je zřejmá z následujících příloh:

Tabulka 3: Stav stanice - dotaz

Tabulka 4: Stav stanice - odpověď

Paket definice měřidla

Paket definice měřidla slouží k načtení aktuálního nastavení vlastností měřidla ze stanice. Jedná se převážně o obecné parametry měření, definice měřících intervalů a mezí a konfigurace použité měřící metody.

Struktura paketu je zřejmá z následujících příloh:

Tabulka 5: Definice měřidla - dotaz

Tabulka 6: Definice měřidla - odpověď

Význam jednotlivých bitově kódovaných dat je zřejmý z následujících příloh:

Tabulka 8: Význam 7. slabiky definice měřidla

Tabulka 7: Význam 9. slabiky definice měřidla

Tabulka 9: Význam 89. slabiky definice měřidla

Paket definic procesní veličiny

Paket definic procesní veličiny nese informace o nastavení parametrů jednoho z osmi kanálů a pro každý z nich má stejnou strukturu. Součástí každého kanálu je tzv. *linearizační tabulka* (2-16 řádků), od jejíž délky se odvíjí také délka paketu.

Struktura paketu je zřejmá z následujících příloh:

Tabulka 10: Definice kanálu (PV) - dotaz

Tabulka 11: Definice kanálu (PV) - odpověď

Význam jednotlivých bitově kódovaných dat je zřejmý z následujících příloh:

Tabulka 12: Význam 8. slabiky definice kanálu (PV)

Tabulka 13: Význam 21.-22. slabiky definice kanálu (PV)

Paket definice stanice

Paket definice stanice nese informace týkající se obecně celé stanice. Typicky se může jednat o její název, velikost osazené paměti, příp. její aktuální stav.

Struktura paketu je zřejmá z následujících příloh:

Tabulka 14: Definice stanice - dotaz

Tabulka 15: Definice stanice - odpověď

Sledování dat stanice v reálném čase

Sledování dat stanice v reálném čase umožňuje obsluze stanice okamžitě monitorovat naměřená data. Program v jednosekundovém intervalu žádá stanici o aktuální data, která následně zakreslí do přehledného grafu. Každý paket aktuálních dat je označen časovou značkou, která udává přesný čas změření konkrétního vzorku.

Konkrétní hodnota procesní veličiny kanálu je závislá na uživatelem definovaných maximálních a minimálních hodnotách daného kanálu. Stanice tedy neposílá konkrétní hodnotu procesní veličiny, ale dvouslabikovou proměnnou, normovanou na tento uživatelem definovaný rozsah kanálu. Výpočet konkrétní hodnoty kanálu pak probíhá podle následujícího vzorce:

$$\text{hodnota} = \left(\frac{\text{vstup}}{0xFFFFE} \right) \cdot (\text{Max} - \text{Min}) + \text{Min}$$

vstup ... dvoubytová proměnná načtená ze stanice

Max ... maximální definovaná hodnota procesní veličiny

Min ... minimální definovaná hodnota procesní veličiny

Struktura paketu je zřejmá z následujících příloh:

Tabulka 27: Sledování dat v reálném čase - dotaz

Monitorovaná data

Monitorovaná data stanice vyjadřují obsah paměti jednotlivých kanálů. Nesou v minulosti naměřené hodnoty, uložené v paměti stanice v uživatelem definovaných intervalech. Počet možných uložených vzorků stanovuje uživatel nastavením příslušného vzorkovacího intervalu a alokováním paměti o požadované velikosti pro jednotlivé kanály.

Načítání monitorovaných dat v praxi většinou znamená přenos značného množství dat. Jelikož délka jedné zprávy je omezena na 255 slabik (viz kapitola 4.2 Základní schéma datového paketu), je nutné zprávu rozdělit na několik částí. Komunikační protokol obsahuje podporu pro takovéto situace. Slabika *identifikace příkazu* (druhá slabika každé zprávy) prvního paketu z řady nese informaci o tom, že je zpráva dělena do více částí a v dalším paketu následuje její pokračování.

Struktura paketu je zřejmá z následujících příloh:

Tabulka 29: Monitorovaná data stanice - dotaz

Tabulka 30: Monitorovaná data stanice - první paket odpovědi

Tabulka 31: Monitorovaná data stanice - pokračovací paket odpovědi

Poslední paket zprávy nastaví slabiku *identifikace příkazu* zpět na hodnotu *datagram nemá pokračování* a přenos se tak ukončí.

Slabika *Identifikace zprávy* je kódována bitově a obsahuje tedy více informací o parametrech přenosu. Detailní přehled významu jednotlivých bitů vyjadřuje Tab. 4.3: Význam jednotlivých bitů identifikace zprávy v kapitole 4.2 Základní schéma datového paketu. Hodnota v tabulce tedy vyjadřuje nastavení používané při této konkrétní úloze.

Archiv stanice

Archivy stanice reprezentují souhrnné hodnoty naměřených dat v určitých časových obdobích. Jedná se o soubory několika registrů uspořádaných tak, jak je popisuje kapitola 2.4 Paměť archivů.

Vzhledem k tomu, že se význam jednotlivých řádků archivu může lišit v závislosti na úpravě zařízení pro konkrétního zákazníka, nese jednotka popisky významů jednotlivých řádků archivu v paměti. Před načtením archivů je tedy nutné tyto popisky ze stanice získat.

Struktura paketů načtení popisek je zřejmá z následujících příloh:

Tabulka 16: Popisek archivu - dotaz

Tabulka 17: Popisek archivu - odpověď

Struktura paketů načtení jednotlivých archivů je zřejmá z následujících příloh:

Tabulka 18: Totalizér tepla (archiv) - dotaz

Tabulka 19: Totalizér tepla (archiv) - odpověď

Systemová hlášení stanice

Hlášení stanice je funkce umožňující identifikovat systémové výpadky a poruchy včetně času, kdy k nim došlo. Velikost paměti alokované pro systémová hlášení stanice je uživatelsky ovlivnitelná. Je proto nutné počítat s velkým počtem možných hlášení, která bude nutné přenést do stanice. Při samotném přenosu se tedy využije možnosti dělení zprávy na více paketů tak, jak je tomu u přenosu vzorků monitorovacích dat (viz kapitola 4.5 Monitorovaná data).

Struktura paketu hlášení stanice je zřejmá z následujících příloh:

Tabulka 20: Hlášení stanice - dotaz

Tabulka 21: Hlášení stanice - odpověď

Patnáctiminutové maximum

Patnáctiminutové maximum reprezentuje průměrně odebrané teplo v suché páře za posledních 15 minut ve všech definovaných okruzích. Do paměti se pak uloží zároveň s časem, kdy byl tento odběr dosažen.

Struktura paketu patnáctiminutového maxima je zřejmá z následujících příloh:

Tabulka 22: Patnáctiminutové maximum - dotaz

Tabulka 23: Patnáctiminutové maximum - odpověď

Letmé měření

Letmé měření umožní uživateli odstartovat měření na definovaném časovém intervalu. Z hlediska komunikace je nutné odeslat stanici žádost o zahájení letmého měření spolu s jeho parametry (čas, okruh...). Po definovaném časovém intervalu se do stanice odešle další paket, kterým se žádá o přenos výsledků letmého měření.

Struktura paketu pro letmé měření je zřejmá z následujících příloh:

Tabulka 24: Letmé měření - zahájení měření

Tabulka 25: Letmé měření - dotaz na výsledek

Tabulka 26: Letmé měření - výsledek měření

Programování stanice

Programování stanice je z hlediska struktury paketů triviální záležitostí. Programovací pakety zpravidla kopírují strukturu paketů korespondující funkce načítaných ze stanice, jediným rozdílem je pozměněná hlavička zprávy. Paket programování vlastností měřidla bude mít tedy například podobný tvar jako jemu odpovídající paket pro přenos vlastností měřidla ze stanice.

Struktura paketu programování stanice je zřejmá z následujících příloh:

Tabulka 32: Programování vlastností stanice

Tabulka 33: Programování vlastností kanálu (PV)

Tabulka 34: Programování vlastností měřidla

Zkrácený stav měřidla (registry stanice)

Paket zkráceného stavu měřidla má mezi ostatními specifické postavení. Jedná se totiž o jediný paket, který je veřejně přístupný. Zákazník tak má možnost získat ze stanice některé základní informace například pro použití ve vlastním systému. Z toho důvodu taky pro paket neplatí stejná pravidla navazování (ukončování) spojení jako je tomu u ostatních paketů.

Struktura paketu je zřejmá z následujících příloh:

Tabulka 35: Zkrácený stav měřidla (registry) - dotaz

Tabulka 36: Zkrácený stav měřidla (registry) - odpověď

Tabulka 37: Význam bitů stavů páry

Tabulka 38: Význam bitů chybového statutu

5 Implementace programu

Vzhledem k rozsáhlosti programu (cca 43000 řádek kódu) se ve své práci zaměřím pouze na popis nejzajímavějších částí. Zájemce o implementační detaily budu u důležitějších pasáží odkazovat na příslušné části zdrojového kódu, který je přehledně komentován.

Celkový přehled konstrukce programových formulářů zobrazuje Obr 1: Diagram organizační struktury aplikace příloh práce.

5.1 Neformální specifikace projektu

Prvotní požadavek na zadání projektu byl vytvořit aplikaci pro platformu Windows 2000/XP jako náhradu za stávající SW pro MS DOS, který již dále nesplňoval především tyto požadavky:

- možnost použití v nových operačních systémech firmy Microsoft
- udržovatelnost programu – opravování chyb programu se vzhledem k jeho celkové koncepci stalo náročným a neperspektivním
- modifikovatelnost – program ve své původní podobě již nevyhovoval požadavkům některých zákazníků a přidávání dalších funkcí se opět jevílo jako neperspektivní řešení

Jako vývojová platforma bylo zvoleno prostředí Borland Delphi ve verzi 7. Přípravná část projektu počítala s fází detailního seznamování se s jednotkou a jejími principy a funkcemi. K tomu bylo možno využít konzultace u spoluvůdce jednotky JSD600 p. Ing. Vlastimila Jaroše a p. Petra Bláhy z ČMI Brno.

V první fázi vývoje bylo cílem vytvořit aplikaci, která by umožnila základní komunikaci s jednotkou, což znamenalo:

- vytvořit nezávislé komunikační rozhraní pro komunikaci se zařízeními firmy Smart, spol. s r.o. prostřednictvím sériového rozhraní (RS232, RS485)
- nastudovat a implementovat komunikační protokol stanice
- implementovat funkce pro načtení a dekodování základních parametrů z jednotky (vlastnosti měřidla, vlastnosti procesních veličin a obecné vlastnosti stanice)
- zobrazit načtené a dekodované parametry jednotky v uživatelsky přívětivé formě
- umožnit změnit načtené parametry a znovu naprogramovat do jednotky (včetně kontroly validity programovaných parametrů)

Po úspěšném vytvoření této základní části programu následovala druhá fáze vývoje, která se nesla ve znamení doplňování dalších funkcí programu. Jednalo se především o následující funkce:

- export/import nastavení měřících parametrů do/ze souboru kompatibilního s původním programem
- načtení z jednotky, export a další operace s archivy stanice
- sledování dat měřených jednotkou v reálném čase, jejich vizualizace v grafu a vyhodnocení jejich stavu
- vizualizace, export, import a vyhodnocení naměřených dat v archivech stanice
- načtení hlášení výpadků stanice z jednotky a jeho vyhodnocení
- doplňkové funkce programu (změna mezi alarmu, čtení registrů stanice, změna mezi měřidla, letmé měření)
- lokalizace grafického uživatelského prostředí do angličtiny s možností přepínání jazyků z prostředí programu
- automatická alokace paměti pro jednotlivé kanály
- zjednodušení práce v síti implementací automatického vyhledávání připojených jednotek a jejich snadného přepínání

Jako poslední část projektu bylo nutné poskytnout zákazníkům technickou podporu a co nejvíc jim tak usnadnit přechod na nový komunikační program. Tato fáze také zahrnuje aktivní spolupráci se zákazníky ochotnými se podílet na vývoji programu, převážně při opravě chyb a při snaze vytvořit uživatelsky přívětivé a přehledné prostředí. Součástí této snahy byla také tvorba kontextové nápovědy k aplikaci, která uživatelům se základní znalostí problematiky umožní se v programu lépe zorientovat.

Na přání zadavatele byl celý program (s výjimkou komunikačního rozhraní) vyvíjen neobjektovým způsobem.

5.2 První fáze projektu

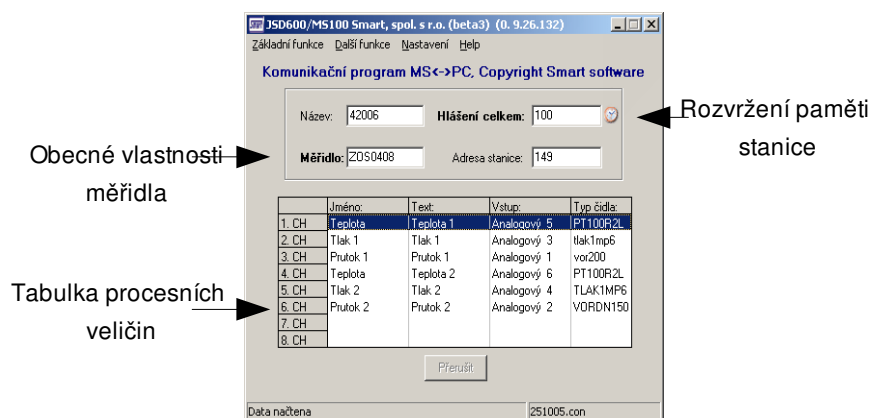
V první fázi projektu bylo nutné implementovat základní a zároveň nejdůležitější funkce programu (viz. 5.1 Neformální specifikace projektu). Jedná se o operace s nastavením procesních veličin (kanálů) stanice, obecných vlastností měřidla a obecných vlastností stanice. Z důvodů snadného přístupu jsou umístěny přímo do hlavního formuláře aplikace (viz Obr 5.1: Zobrazení stanice (hlavní formulář)).

Před samotnou implementací těchto funkcí však bylo nutné vytvořit sadu základních programových konstrukcí opakovaně používaných v celém programu.

Jednalo se zejména o následující konstrukce:

- implementace třídy zajišťující komunikaci s jednotkou přes rozhraní RS232/RS485
- implementace převodních funkcí pro dekódování informací z paketů komunikace
- implementace funkcí pro dekódování a formátování data a času
- implementace sdílených proměnných pro parametry jednotky, které vstupují do parních výpočtů
- a další opakovaně používané elementární funkce

Všechny tyto základní a do jisté míry aplikačně univerzální operace jsou centralizovány do souboru *CommonFunctions.pas*.



Obr 5.1: Zobrazení stanice (hlavní formulář)

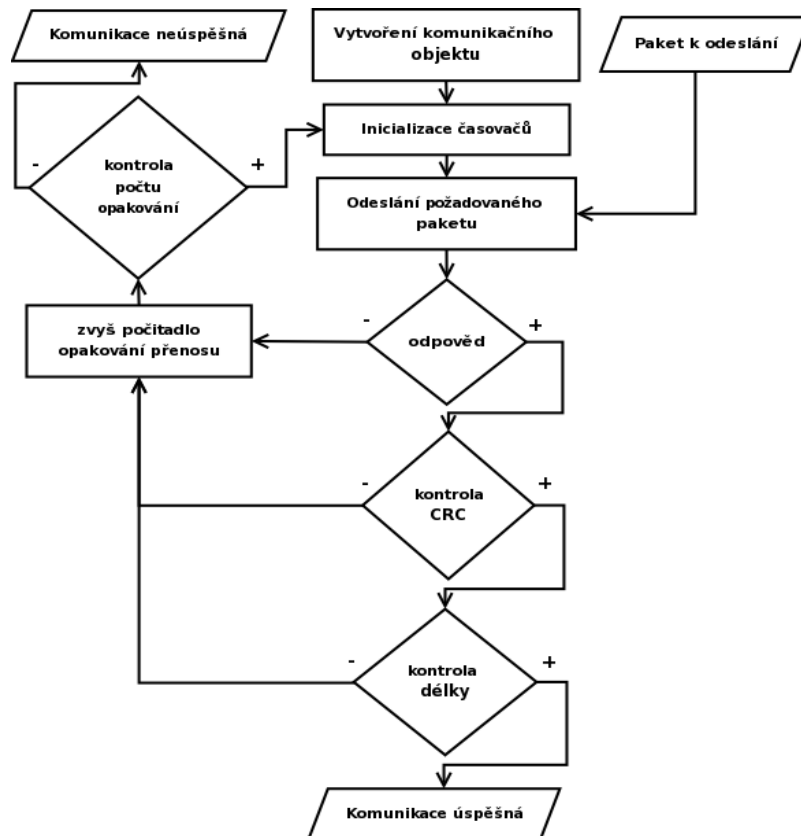
Implementace komunikace s jednotkou

Komunikační část programu je jako jediná implementována objektově. Jedná se totiž o objekt, používaný v několika dalších menších programech vyvíjených pro firmu Smart, spol. s r. o. Jednoduchá struktura objektu obsahuje pouze jednu třídu zapouzdřující veškeré komunikační operace. Metody třídy přehledně znázorňuje Tab. 5.1: Přehled metod komunikační třídy.

ClearBuffers	vymaže přijímací i vysílací buffery
SendPacket	odešle patřičný paket stanici
SendPacketAndWaitForAnsw	odešle patřičný paket a vyčká na odpověď ze stanice
CheckPresence	otestuje přítomnost jednotky na aktuální adrese
SendACK	odešle potvrzovací paket (ACK)
Stop	ukončí probíhající komunikaci
Data_CRC_Check	zkontroluje CRC přijatých dat

Tab. 5.1: Přehled metod komunikační třídy

Funkce komunikační třídy je založena převážně na metodě *SendPacketAndWaitForAnsw()*. Jak už napovídá její samotný název, stará se o vyslání požadovaného datagramu do stanice a vyčkání na potvrzení stanicí (viz Tab. 4.4: Odpovědi na přijatý paket). Pokud odpověď ze stanice nepřijde v určitém časovém limitu, pokusí se metoda přenos ještě dvakrát zopakovat. Poté zaregistruje a vyšle uživatelskou zprávu o úspěchu nebo neúspěchu komunikace do formuláře, který komunikaci inicioval. Ten se pak na základě této zprávy rozhodne, zda komunikaci přeruší nebo bude pokračovat vysláním dalšího datagramu v pořadí. Funkci komunikační třídy pro jeden paket demonstruje Obr 5.2: Diagram funkce komunikační třídy.



Obr 5.2: Diagram funkce komunikační třídy

Pro inicializaci použitých komunikačních paketů třída využívá jejich definice v souboru *Datagrams.pas*. Zde jsou definovány všechny pakety, které jednotka pro komunikaci používá (viz 4.5 Jednotlivé komunikační pakety) včetně některých obslužných funkcí (například výpočtu kontrolního součtu zprávy).

Implementační detaily jsou zřejmé z následujících zdrojových kódů:

CommunicationEngine2.pas (komunikační třída aplikace)

Datagrams.pas (komunikační datagramy)

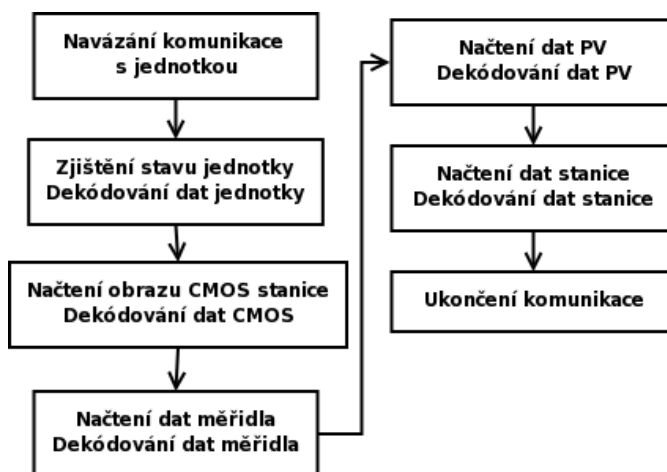
Sdílené proměnné

Aplikace používá velké množství informací, které jsou sdílené mezi několika formuláři. Většinou se jedná o uživatelem nastavené parametry měření načtené ze stanice, které tvoří vstupy matematickým výpočtům nad daty stanice v různých funkcích programu. Jako nejvhodnější řešení se tedy jeví použití globálních proměnných, centralizovaných v jednom souboru. Tuto funkci plní formulář *MainUnit1*. Jedná se o formulář uživatelsky nepřístupný, který mimo definice proměnných nese některé podpurné prostředky pro ladění programových konstrukcí.

Struktura dat stanice v globálních proměnných kopíruje strukturu komunikačních paketů. Usnadní se tím orientace v kódu a pro nalezení konkrétní proměnné lze tak s výhodou použít tabulky popisující strukturu komunikačních paketů (viz kapitola 4.5 Jednotlivé komunikační pakety).

Čtení základních dat stanice

Základní data stanice (tak jak je specifikuje kapitola 5.1 Neformální specifikace projektu) se načítají ze stanice v pořadí, které demonstruje Obr 5.3: Čtení základních parametrů stanice.



Obr 5.3: Čtení základních parametrů stanice

Každý rámeček diagramu představuje výměnu komunikačního paketu mezi stanicí a programem. Každý rámeček tedy zároveň reprezentuje paket dotazu a paket odpovědi, nesoucí vyžádaná data ze stanice. Výjimku tvoří pouze okrajové rámečky diagramu, které představují servisní pakety pro sestavení komunikace s jednotkou a z hlediska dat jsou bezvýznamné. Po přijetí paketu ze stanice je nutné data rozdělit podle pravidel definovaných strukturou paketů (viz kapitola 4.5 Jednotlivé komunikační pakety) a uložit je do příslušných sdílených proměnných. K tomuto účelu slouží fáze *dekódování dat*. Pro každý paket je definována funkce, která se stará o jeho dekódování do sdílených proměnných.

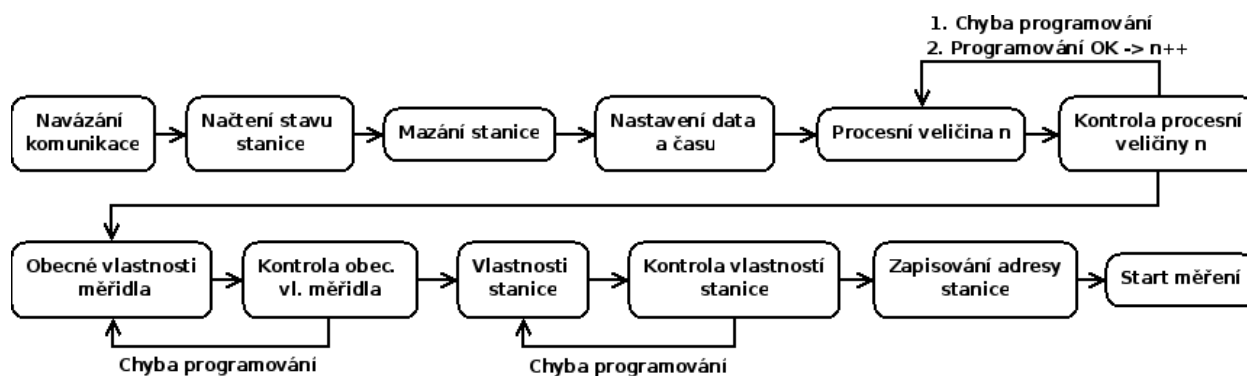
Programování stanice

Programování stanice je proces, při kterém dochází ke změně měřicích parametrů stanice. Z toho důvodu se při každém programování mažou archivy stanice a ta začíná měřit od začátku. Změna jednotlivých parametrů z bezpečnostních důvodů ve většině případů vyžaduje kompletní přeprogramování stanice. Jinak by totiž snadno mohlo dojít k nekonzistenci vzájemných vztahů mezi parametry, protože jednotka nemá dostatečnou kapacitu pro kontrolu jejich korektního zadání. Dalším důvodem je míchání hodnot v archivech naměřených před a po změně parametrů, které by takto ztrácely svoji vypovídací hodnotu.

Samotné programování podléhá všem zákonitostem komunikačního protokolu (viz kapitola 4 Komunikační protokol):

- Program nejprve naváže spojení s jednotkou na definované adrese, zjistí její stav a ověří tak, že je na programování připravena (není v chybě, nevykonává výpočet nebo není jinak zaneprázdněna).
- Následuje vymazání současných parametrů stanice a po tomto kroku již nelze programování přerušit. Pokud se tak přeci jen stane, není stanice schopna měřit a je třeba ji znovu naprogramovat.
- Stanici je nastaven čas podle systémového času PC, ze kterého je programována.
- Přenos programovaných dat podléhá zpětné kontrole. Po přenesení parametrů jednoho kanálu je tento opět načten ze stanice a porovnán s programovanými daty. Stejně pravidlo platí pro programování vlastností měřidla a stanice.
- Pokud všechny stupně programování proběhnou bez problémů, запиše se nová síťová adresa stanice a vyšle se příkaz pro zahájení měření.

Celou posloupnost programovacího procesu znázorňuje Obr 5.4: Diagram programování stanice



Obr 5.4: Diagram programování stanice

Obecné vlastnosti měřidla

Obecné vlastnosti měřidla reprezentují globální nastavení jednotky, které je nezávislé na nastavení vlastností procesních veličin nebo stanice. Program nabízí jednoduchý formulář (*Definice měřidla*), který všechny tyto parametry přehledně zobrazí, případně je umožní změnit.

Konstanty měřicích mezí	
TP Max [°C]	400,000000
TP min [°C]	1,000000
TK max [°C]	100,000000
TK min [°C]	0,000000
PP max [MPa]	2,000000
PP min [MPa]	0,000000
Průtok č.1 max [kg/h]	70000,000000
Průtok č.1 min [kg/h]	0,000000
Průtok č.2 max [kg/h]	40000,000000
Průtok č.2 min [kg/h]	0,000000
Maximum č.1 [GJ]	1000,000000
Maximum č.2 [GJ]	1000,000000
Pásmo sytosti [°C]	10,000000
Hystereze [°C]	0,000000

Konstanty par. měření	
Korekce MP [%/°C]	1,000000
Interval P1 [sec]	5
Interval P2 [sec]	5
Interval kond [sec]	60
Impulsní koef.1: [1/GJ]	0,000000
Impulsní koef.2: [1/GJ]	0,000000

Obr 5.5: Definice měřidla

Parametr měřidla *Způsob měření* slouží pro nastavení principu použitého průtokoměru. Stanice v současné době podporuje tři metody – objemová, hmotnostní a přímá (viz kapitola 3.3 Základy měření průtoku).

Dále vlastnosti měřidla definují převážně dovolené rozsahy jednotlivých veličin (teploty, tlaky, průtoky - při jejichž překročení je vyhlášen chybový stav), *pásma sytosti* a *pásma hystereze* (viz 3.2 Hystereze a pásmo sytosti).

Pro detailní význam ostatních parametrů měřidla je k dispozici kontextový help aplikace nebo manuál stanice.

Implementační detaily jsou zřejmé z následujících zdrojových kódů:

DefMeridlaUnit.pas

(formulář obecných vlastností měřidla)

DefMeridlaRoutines.pas

(pomocné funkce)

Vlastnosti procesních veličin

Formulář procesních veličin (*Definice veličiny*) detailně definuje parametry každého kanálu stanice.

The screenshot shows a window titled "Definice veličiny" for channel "CH. 1". The fields are as follows:

Jméno:	Teplota	Interval:	0 h 10 m 0 s
Text:	Teplota 1	Pamět:	6295
Jednotky:	°C	Měření:	Průměrná
Typ čidla:	PT100R2L	Display:	Okamžitá
Minimální hodnota:	-30,0000	Výstup:	Žádná
Max. hodnota:	350,0000	Minimální meze:	0,002
Vstup:	Analogový 5	Maximální meze:	7,998
Teplota kalibrační vody:	0,0000	Teplotní koeficient:	0,0000

At the bottom, there are two checkboxes: Inverzní hodnoty výstupu and Automatická alokace paměti. Buttons include Zrušit, Smazat, Do stanice..., and Uložit.

Obr 5.6: Definice veličiny

Některé základní parametry kanálu zobrazuje Tab. 5.2: Význam některých parametrů procesní veličiny.

Jméno	název kanálu
Text	informativní text kanálu (poznámka)
Jednotky	jednotky kanálu
Min/Max. hodnota	meze měřené procesní veličiny
Vstup	fyzický vstup veličiny

Tab. 5.2: Význam některých parametrů procesní veličiny

Dvojklikem na oblast *Typ čidla* se zpřístupní převodní (linearizační) tabulka kanálu (viz 5.2 Linearizační tabulka).

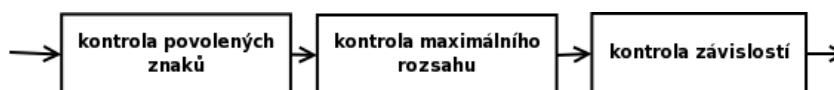
Za pozornost stojí také parametry *Paměť* a *Interval*. První udává počet slabik (resp. maximální možný počet naměřených vzorků) paměti alokovaných pro danou procesní veličinu (kanál), druhý pak vzorkovací interval kanálu (tj. doba mezi uložením dvou vzorků do paměti). Dvojklikem na oblast alokované paměti je pak možné zobrazit distribuci paměti mezi jednotlivé kanály (viz 5.3 Rozvržení paměti).

S alokováním paměti souvisí i příznak *Automatická alokace paměti*. Při jeho zvolení je uživateli znemožněno ručně upravovat velikost paměti kanálu a program se pokusí dostupnou paměť rozdělit mezi kanály označenými tímto příznakem tak, aby všechny byly schopny uchovat naměřená

data po stejně dlouhý časový úsek (viz 5.3 Automatická alokace paměti).

Některé z vlastností kanálu jsou navzájem provázány vztahy, které vyplývají z principů měření energie a před uložením kanálu je proto nutné ověřit jejich konzistenci. Zodpovědnost za korektní nastavení stanice je plně na aplikaci, která stanici programuje. V případě chybného nastavení nebude stanice vykazovat správné výsledky měření nebo vyhlásí chybu.

Další nutnou podmínkou je provedení testů, zda uživatel nezadá do vstupních polí hodnoty nepodporovaného typu (například znaky do celočíselné proměnné nebo prázdnou hodnotu do veličiny, která ji nepřipouští). V krajním případě by pokus o uložení takových hodnot mohl způsobit až generování výjimky a nekorektní ukončení aplikace. Ve funkci stisku tlačítka pro uložení dat formuláře je pro každé vstupní pole definována funkce, která kontroluje zadání přípustných hodnot a v případě chyby uživateli oznámí její přesné umístění včetně pozice prvního chybného znaku v řetězci. Zároveň znemožní provést uložení formuláře. Problematika testování vstupních dat se netýká pouze formuláře procesních veličin, ale naopak je nezbytnou součástí každého formuláře programu, ve kterém je požadováno zadání vstupních dat. Obr 5.7: Diagram posloupnosti testů vstupů formuláře vyjadřuje posloupnost aplikace testů.



Obr 5.7: Diagram posloupnosti testů vstupů formuláře

Ostatní parametry kanálu se týkají nastavení výstupních modulů a některých dalších specifických vlastností stanice. Detailní informace poskytují manuály k jednotce a stanici.

Implementační detaily jsou zřejmé z následujících zdrojových kódů:

DefVelicinyUnit.pas (formulář vlastností procesních veličin)

DefVeličinyRoutines.pas (pomocné funkce procesních veličin)

CommonFunctions.pas (*AsciiCheck()* (801) – kontrola vstupů formuláře)

Linearizační tabulka

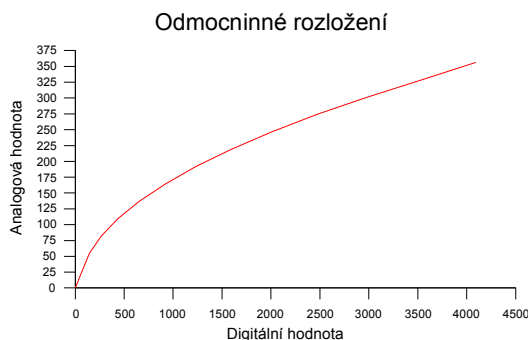
Linearizační tabulka je nástroj ovlivňující převod vstupního analogového signálu na digitální. Umožňuje přiřadit stupni analogově-digitálního převodníku konkrétní analogovou hodnotu. Tato modifikace vstupu nachází využití převážně v aplikacích, u kterých je třeba soustředit větší rozlišení převodníku do určité oblasti. Aplikaci na příkladu vstupních analogových hodnot 0-350 znázorňuje Obr 5.8: Kvadratické rozložení.

Z grafu je jasné, že převodník má v tomto případě větší rozlišení (menší velikost analogové hodnoty na jeden dílek převodníku) pro nižší hodnoty měřeného rozsahu analogového vstupu. Naopak pro vyšší hodnoty rozlišení pomalu klesá.

Naproti tomu opačný stav demonstruje Obr 5.9: Odmocninné rozložení.



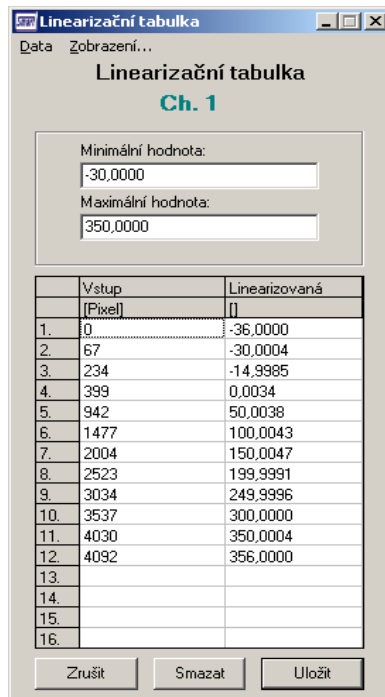
Obr 5.8: Kvadratické rozložení



Obr 5.9: Odmocninné rozložení

Kvadratické a odmocninné rozložení představuje v praxi nejčastěji používané nastavení vstupního převodníku a program tedy umožňuje automatické generování koeficientů linearizační tabulky pro dané rozložení. Základním požadavkem pro generování je zadání pouze dvou řádků tabulky, které definují rozsah výstupní digitální a vstupní analogové veličiny.

Pro každou linearizační tabulku je nutné definovat také očekávaný rozsah analogové vstupní veličiny. Nastavení těchto parametrů je vhodné věnovat zvýšenou pozornost, protože ovlivňují rozlišení AD převodníku a mají tedy nezanedbatelný vliv na přesnost měření stanice. Pro dosažení co největší přesnosti je proto vhodné nastavit rozsahy tak, aby se co nejvíce blížily reálným rozsahům měřené soustavy.



Obr 5.10: Linearizační tabulka

Při každém pokusu o uložení tabulky je kontrolována její konzistence, řádky jsou opět seřazeny vzestupně podle dílků převodníku (viz zdrojový kód *LinTabUnitRoutines.pas* (67) – *BubbleSort()*) a až po kontrole validity všech parametrů je tabulka přijata a uložena. Pro řazení byl záměrně vybrán jeden z nejjednodušších algoritmů – Bubblesort. Vzhledem k tomu, že se jedná o seřazení pouze šestnácti položek, jeho nízká efektivita se v podstatě neprojevuje.

Program nabízí další automatický výpočet pro třetí a šestý kanál (kanál rezervovaný pro měření průtoku). Jedná se o pomocnou tabulku pro převody jednotek z kg/h na m^3/h . Uživateli umožní ověřit validitu nastavení výpočtem mezních tlaků a teplot, při kterých je pára v požadovaném stavu (viz kapitola 3 Základy teorie měření energie). Výpočet lze pro třetí a šestý kanál spustit opět z kontextového menu prostřednictvím nabídky *Převod jednotek*.

Implementační detaily jsou zřejmé z následujících zdrojových kódů:

LinTabUnit.pas (formulář linearizační tabulky)

LinTabUnitRoutines.pas (pomocně funkce linearizační tabulky)

LinTabUnitRoutines.pas (*BubbleSort*, řádek 67)

5.3 Druhá fáze projektu

První fáze projektu se soustředila na vývoj základního rozhraní pro implementaci souboru funkcí nad daty jednotky. Po odzkoušení těchto funkcí na základních operacích bylo možné pokračovat v implementaci pokročilých funkcí programu.

Export do souboru

Nastavení základních vlastností jednotky (obecné vlastnosti měřidla, vlastnosti stanice, vlastnosti jednotlivých procesních veličin) lze exportovat do souboru. Tento soubor má textovou podobu a v případě potřeby je možné data opět načíst do programu a opětovně s nimi stanici naprogramovat. Program tak nabízí možnost snadné archivace a zálohy nastavení jednotek pro případ poruchy nebo havárie stanice.

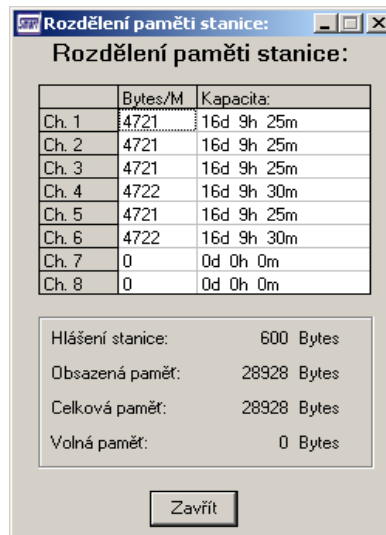
V požadavcích na projekt byla explicitně uvedena podmínka zpětné kompatibility souborů exportovaných dat se soubory původního programu. Zatímco v prostředí operačního systému MS DOS se používalo kódování *Latin2*, operační systémy z rodiny Windows používají v české mutaci kódování *CP-1250*. Vzhledem k tomu, že jsou soubory z důvodů lepší čitelnosti uloženy v textovém a nikoliv binárním formátu, vzniká problém nejednoznačnosti namapování některých českých znaků mezi kódovými stránkami. Z tohoto důvodu je do exportních funkcí zařazen filtr problémových znaků spolu s převodní tabulkou, která by měla zajistit případnou korekci.

Implementační detaily jsou zřejmé z následujících zdrojových kódů:

<i>ZobrStaniceUnit.pas</i>	<i>(formulář hlavního okna aplikace)</i>
<i>FileOperunit.pas (672)</i>	<i>(funkce pro práci se soubory – export)</i>
<i>FileOperunit.pas (112)</i>	<i>(filtr a převodní tabulka znaků)</i>

Rozvržení paměti

Formulář rozvržení paměti přehledně ukazuje množství paměti volné a paměti přidělené jednotlivým kanálům.



	Bytes/M	Kapacita:
Ch. 1	4721	16d 9h 25m
Ch. 2	4721	16d 9h 25m
Ch. 3	4721	16d 9h 25m
Ch. 4	4722	16d 9h 30m
Ch. 5	4721	16d 9h 25m
Ch. 6	4722	16d 9h 30m
Ch. 7	0	0d 0h 0m
Ch. 8	0	0d 0h 0m

Hlášení stanice:	600 Bytes
Obsazená paměť:	28928 Bytes
Celková paměť:	28928 Bytes
Volná paměť:	0 Bytes

Zavřít

Obr 5.11: Rozdělení paměti stanice 1

Sloupec *kapacita* zobrazuje velikost alokované paměti v souvislosti s vzorkovacím intervalem kanálu (viz 5.2 Vlastnosti procesních veličin). Jde o dobu, za kterou se přidělená paměť zaplní a (vzhledem k tomu, že paměť stanice je implementována jako kruhový buffer) nejstarší data začnou být přepisována novými. Obsluha tak má jasný přehled o tom, kdy musí naměřená data z jednotky zálohovat než budou znehodnocena novými.

Jelikož koncepce stanice dovoluje alokovat každému kanálu libovolný prostor v paměti a libovolný vzorkovací interval, uživatel může nadefinovat větší rozlišení tomu kanálu, který je pro něj důležitější a u kterého vyžaduje častější zápis vzorků. Nerovnoměrné rozložení paměti demonstruje Obr 5.12: Rozdělení paměti stanice 2. V tomto případě bude na stejném časovém intervalu pro třetí a šestý kanál zaznamenáno přibližně o 2500 vzorků víc, protože byl nastaven poloviční vzorkovací interval.

Formulář zároveň ukazuje počet slabik alokovaných pro systémová hlášení stanice (viz 2.4 Systémová hlášení stanice). Každému takovému hlášení přísluší 6 slabik sdílené paměti. V nastavení z Obr 5.11 je tedy prostor pro 100 hlášení.

	Bytes/M	Kapacita:
Ch. 1	3862	13d 9h 50m
Ch. 2	3862	13d 9h 50m
Ch. 3	6438	13d 9h 54m
Ch. 4	3862	13d 9h 50m
Ch. 5	3862	13d 9h 50m
Ch. 6	6438	13d 9h 54m
Ch. 7	0	0d 0h 0m
Ch. 8	0	0d 0h 0m

Hlášení stanice:	600 Bytes
Obsazená paměť:	28924 Bytes
Celková paměť:	28928 Bytes
Volná paměť:	4 Bytes

Zavřít

Obr 5.12: Rozdělení paměti stanice 2

Implementační detaily jsou zřejmé z následujících zdrojových kódů:

MemInfoUnit.pas

(formulář rozvržení paměti)

MemInfoRoutines.pas

(pomocné funkce rozvržení paměti)

Automatická alokace paměti

V případě, že uživatel nemá specifické požadavky na rozvržení paměti mezi jednotlivé kanály, může použít funkci automatické alokace paměti. Každý kanál s tímto příznakem pak bude zahrnut do výpočtu distribuce paměti.

Z uživatelského hlediska nestačí pouhé rozdělení dostupné paměti rovnoměrně mezi kanály, ale je nutné brát v úvahu také vzorkovací interval jednotlivých kanálů a paměť rozdělit tak, aby při daných parametrech bylo možné uchovat aktuální data ve všech kanálech po stejnou dobu.

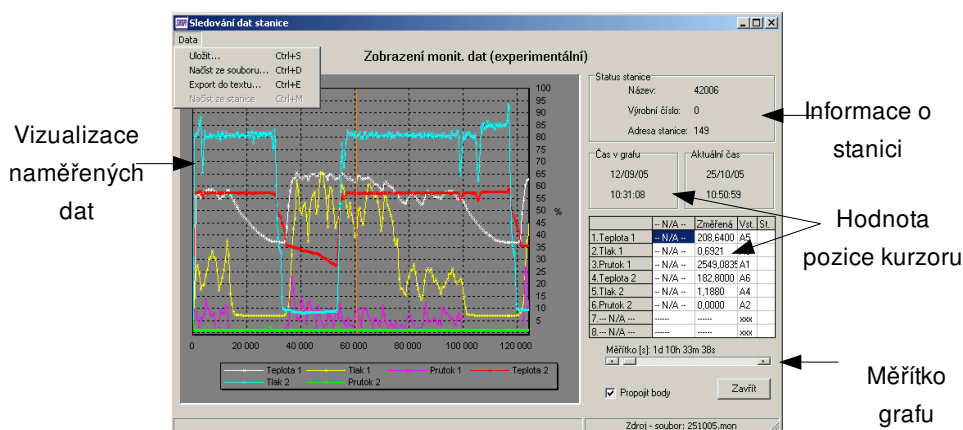
Problematiku výpočtu v programu řeší funkce *mem_AutoAlloc* (viz zdrojové kódy *MemInfoRoutines.pas*, řádek 25), jejímž vstupem jsou parametry všech kanálů (procesních veličin) a výstupem pak požadovaná distribuce paměti mezi kanály. Samotný výpočet probíhá následujícím způsobem:

- kontrola intervalu kanálu (musí spadat do 12-ti hodinového intervalu)
- výpočet celkové paměti, kterou lze rozdělit mezi kanály se zapnutou funkcí *automatické alokace paměti*
- výpočet podílu na zaplnění paměti jednotlivých kanálů se zapnutou funkcí *automatické*

Monitorovaná data stanice

Formulář monitorovaných dat stanice vizualizuje v grafu naměřené hodnoty každého kanálu uložené v definovaných vzorkovacích intervalech (viz 2.4 Paměť kanálů). Počet vzorků každého kanálu je uživatelsky definovatelný ve vlastnostech jednotlivých procesních veličin (viz 5.2 Vlastnosti procesních veličin).

Obsluhu stanice jsou monitorovaná data k dispozici z kontextového menu -> *Další funkce* -> *Zobrazení monit. dat.*



Obr 5.15: Monitorovaná data stanice

Počet vzorků kanálů závisí na aktuálním nastavení a zaplnění paměti. Může však dosahovat značné velikosti (v závislosti na kapacitě paměti, kterou je stanice osazena). Tento případ se jeví jako ideální pro implementaci některou z dynamických datových struktur. V programu je použit objekt standardní knihovny Delphi *TList* dynamické velikosti, do něhož je načten kompletní soubor dat jednotlivých kanálů.

Po načtení monitorovaných dat ze stanice (nebo ze souboru) se uživateli zobrazí přehledný graf, zobrazující průběh vstupních veličin. Pohybem kurzoru je možné graf procházet a vyčítat hodnoty veličin na pozici kurzoru, které jsou v reálném čase zpracovávány a z dat načtených ze stanice jsou pak aplikací parních výpočtů vypočítány naměřené hodnoty jednotlivých kanálů v korespondujícím čase. Pro implementaci grafu je využita komponenta ze standardní knihovny Delphi *TChart*.

Při nastavení jednotlivých procesních veličin může snadno vzniknout situace, kdy je pro každý kanál nastaven jiný vzorkovací interval.

Představme si následující příklad nastavení vzorkovacích intervalů:

1. kanál – 2s
2. kanál – 6s
3. kanálů – 10s

V tomto případě je nutné, aby si graf automaticky přizpůsobil svoje rozlišení na nejmenší společný dělitel intervalů, což jsou v našem případě 2 sekundy. Pohyb kurzorovými klávesami v grafu tedy způsobí časový posun o 2 sekundy. Z uvedeného je jasné, že zatímco vzorky pro první kanál budou definovány pro každý krok v grafu, pro druhý bude definován každý třetí vzorek a pro třetí pak každý pátý vzorek. Sloupec tabulky označený jako *Změřeno* tedy ukazuje poslední hodnotu, kterou kurzor při svém pohybu překročil. Aktualizace se provede ve chvíli, kdy kurzor narazí na další platnou hodnotu kanálu.

Další dva sloupce tabulky pak patří vstupu definovanému pro daný kanál (*Inp*) a příznaku kvality páry (suchá, mimo meze, mokrá).

Funkce monitorování dat je v programu zatím v experimentálním režimu a stále se na ní pracuje.

Implementační detaily jsou zřejmé z následujících zdrojových kódů:

SledDatRoutines.pas(378) – prepare_Monitoring()

SledDatRoutines.pas(591) – CountAndApprox()

SledDatRoutines.pas(663) – MonitoringValues()

MonSplashUnit.pas (načítání vzorků ze stanice)

Možnosti funkce monitorování dat nekončí pouze s vizualizací dat. Ty je nutné exportovat pro archivaci nebo pozdější zpracování. Program nabízí dvě možnosti exportu dat načtených ze stanice:

Binární formát souboru nese data načtená přímo ze stanice v nezměněné (binární) podobě, na která nebyly aplikovány žádné výpočty. Tento typ souboru je možné programem znovu načíst a

zobrazit do grafu. Vzhledem k tomu, že výpočty nad naměřenými daty stanice vychází z jejího aktuálního nastavení, je nutné tyto parametry k binárnímu souboru připojit, aby bylo možné po opětovném načtení naměřená data vyhodnotit. Ve skutečnosti se tedy ukládají soubory dva. První má příponu *.con a v textové podobě definuje parametry stanice. Jedná se o stejný soubor, který je vytvořen při exportu dat nastavení stanice do souboru z hlavního formuláře programu (viz kapitola 5.3 Export do souboru). Druhý soubor (tentokrát s příponou *.mon) pak má již zmíněnou binární podobu a nese naměřená data načtená z jednotky.

Textový formát souboru obsahuje podrobný výpis celé doby měření a hodnot naměřených veličin v každém vzorkovacím intervalu, na které jsou již aplikovány parní výpočty. Výhoda tohoto formátu je snadná čitelnost a možnost importu do dalších aplikací (například tabulkové procesory). Naopak nevýhodou může být nemožnost soubor znovu načíst a zobrazit do grafu aplikace a nemožnost zpětně zjistit nastavení stanice v době měření, protože k tomuto souboru není ukládán druhý soubor se specifikací parametrů stanice.

Stanice1	TEMPERAT	Tlak	Prutok	---	---	---	---	---
46049	Tep. par	Tlak par	Prut. pa	---	---	---	---	---
17/01/07	řC	MPa	kg/h	---	---	---	---	---
20:05:00	180.0000	0.904799	204.0176	xxx	xxx	xxx	xxx	xxx
20:10:00	180.0000	0.898400	112.8439	xxx	xxx	xxx	xxx	xxx
20:15:00	180.0000	0.904799	204.0176	xxx	xxx	xxx	xxx	xxx
20:20:00	180.0000	0.898400	90.27518	xxx	xxx	xxx	xxx	xxx
20:25:00	180.0000	0.898400	169.2659	xxx	xxx	xxx	xxx	xxx
20:30:00	180.0000	0.898400	135.4127	xxx	xxx	xxx	xxx	xxx
20:35:00	180.0000	0.904799	170.0147	xxx	xxx	xxx	xxx	xxx
20:40:00	180.0000	0.898400	146.6971	xxx	xxx	xxx	xxx	xxx
20:45:00	180.0000	0.898400	146.6971	xxx	xxx	xxx	xxx	xxx
20:50:00	180.0000	0.898400	101.5595	xxx	xxx	xxx	xxx	xxx

Obr 5.16: Příklad textového formátu souboru monitorovaných dat

Při exportu monitorovaných dat do textu vzniká problém volby časového kroku jednotlivých řádků podobně, jako u grafu monitorovaných dat. Jak bylo dříve zmíněno, data jednotlivých kanálů jsou měřena v uživatelsky definovatelných intervalech. Mezi zvoleným intervalem je tedy hodnota monitorovaných dat nedefinovaná. Stanice obchází toto omezení tak, že v každém časovém okamžiku považuje za platnou hodnotu veličiny právě poslední změřenou.

Validitu dat v určitých časových intervalech demonstruje Obr 5.16: Příklad textového formátu souboru monitorovaných dat. V tomto případě byly vzorkovací intervaly kanálů nastaveny následovně:

1. kanál – 15 minut (900 sekund)
2. kanál – 10 minut (600 sekund)
3. kanál – 5 minut (300 sekund)

Červeně zvýrazněné hodnoty tedy odpovídají hodnotám skutečně změřeným, zatímco u ostatních se jedná pouze o hodnotu předpokládanou.

Implementační detaily jsou zřejmé z následujících zdrojových kódů:

SledDatUnit.pas (formulář sledování dat)

SledDatRoutines.pas (pomocné funkce sledování dat)

Lokalizace uživatelského prostředí

Jednotka JSD600 a její konfigurační program jsou v praxi využívány i v zahraničí. Z toho důvodu je aplikace navržena pro snadnou lokalizaci až do deseti světových jazyků. V současné době je k dispozici anglický překlad uživatelského rozhraní a ve stadiu příprav se nachází překlad polský.

Z implementačního hlediska je lokalizace řešena pomocí *resource* souboru, ve kterém jsou definovány všechny lokalizační řetězce. Vzhledem k faktu, že prostředí Delphi neumožňuje za běhu měnit resource soubory jednotlivých jazyků, je použit pouze jeden společný soubor pro všechny jazyky. Při vytváření každého formuláře jsou z tohoto souboru načteny příslušné lokalizační řetězce, které se následně aplikují na celý formulář. Každému formuláři tedy přísluší jedna lokalizační funkce ze souboru *LanguageUnit.pas*.

```
1006, "Nová stanice"  
2006, "New station"  
  
1007, "Programování stanice"  
2007, "Send setup to system"  
  
1008, "Uzamčení stanice"  
2008, "Lock out"  
  
1009, "Uložení dat"  
2009, "Store system setup"  
  
1010, "Čtení archivu stanice"  
2010, "Show archive values"
```

Obr 5.18: Struktura lokalizačního souboru

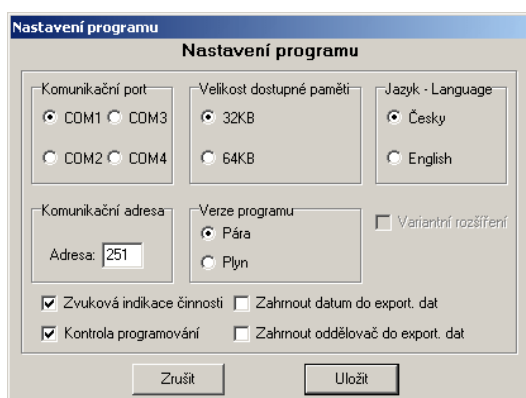
Strukturu lokalizačního (resource) souboru popisuje Obr 5.18: Struktura lokalizačního souboru. Jednotlivým řetězcům přísluší unikátní číslo, jehož první číslice označuje příslušnost řetězce k jednotlivým jazykům (1 – CZ, 2 – EN), zatímco zbylé tři číslice tvoří identifikátor samotného řetězce. Z uvedeného příkladu tedy vyplývá omezení pro 1000 lokalizačních řetězců a 10 jazyků, což je ovšem prostor pro potřeby aplikace zcela dostačující (v současné době je použito přibližně 400 lokalizačních řetězců a dva jazyky).

Implementační detaily jsou zřejmé z následujících zdrojových kódů:

LanguageUnit.pas (mapování řetězců na formuláře)

Nastavení programu

Formulář nastavení programu obsahuje globální volby pro chování programu.



Obr 5.19: Formulář nastavení programu

Mezi hlavní patří volba komunikačního portu, na kterém je stanice připojena. Před uložením nově zvoleného komunikačního portu je nutné provést kontrolu, zda je požadovaný port dostupný nebo není-li blokován jinou aplikací. Test spočívá v pokusu o otevření daného portu a odchyzení případné výjimky. Komunikační třída vyžaduje řádně otevřený port a formulář proto nesmí dovolit zvolení portu, který není dostupný.

Volba *Jazyk-Language* zajišťuje přepínání lokalizace programu. Přepnutí spočívá v podstatě pouze ve změně globální proměnné použitého jazyka, podle které se řídí lokalizační funkce každého formuláře.

Komunikační adresa značí adresu jednotky v síti, kterou využívá komunikační třída pro navázání spojení. Z pochopitelných důvodů není možné umístit na síť více stanic se stejnou adresou nebo použít adresu 251 (všesměrové vysílání – broadcast) pro topologii s více než jednou jednotkou. V takovém případě by komunikace nebyla úspěšná.

Další volby jsou specifické pro nestandardní hardware stanice, příp. ošetřují konkrétní požadavky jednotlivých zákazníků. Detaily těchto parametrů dostatečně vysvětluje programový manuál.

Pro zachování voleb uživatele i po novém startu aplikace je využito jednoduchého INI souboru (*settings.ini*) umístěného v pracovním adresáři. Pro správnou funkci je tedy nutné umožnit zápis do pracovního adresáře programu. Operace s INI soubory pohodlně poskytuje objekt `TIniFile` ze standardní knihovny Delphi. Stiskem klávesy *Uložit* je vytvořena instance objektu `TIniFile`. Následuje otevření (případně vytvoření) souboru a zápis nových nastavení. Analogický postup je využit při startu aplikace.

Strukturu jednoduchého *INI* souboru demonstruje Obr 5.3: Čtení základních parametrů stanice.

```
[Settings]
FTM=0
Port=1
Mem=32
sound=1
lang=1
addr=251
version=0
variant=0
ExportDate=0
Delimiter=0
[Debug]
Checking=1
```

Tab. 5.3: INI soubor aplikace

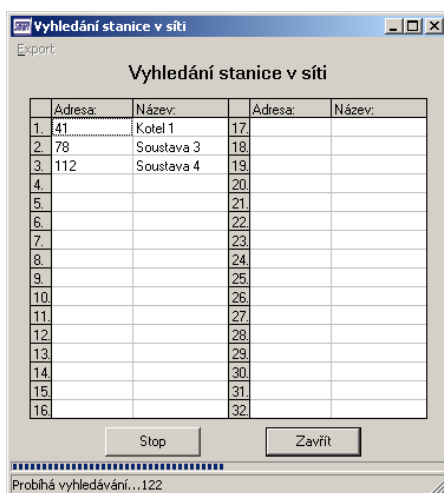
Implementační detaily jsou zřejmé z následujících zdrojových kódů:

NastaveniUnit.pas (*formulář nastavení*)

MainUnit1.pas (441) (*načtení nastavení při startu programu*)

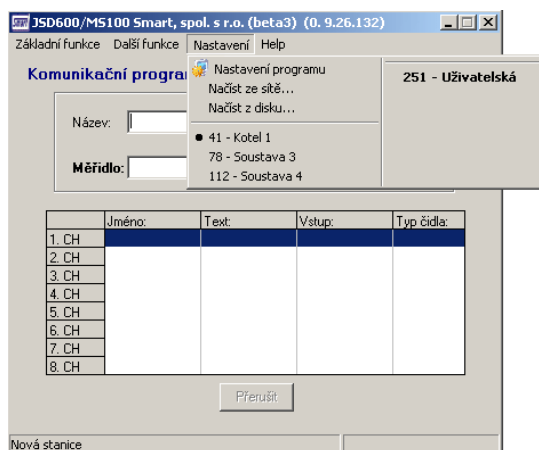
Vyhledání stanic v síti

Jak již bylo výše zmíněno, komunikace se stanicí JSD600 je adresovatelná a je proto možné provozovat soustavu zapojenou v síti. Typická aplikace takovéto sítě je průmyslová hala, ve které jsou na různých úkolech zapojeny stanice JSD600 a operátor má v kanceláři přehled o nastavení a o naměřených hodnotách jednotlivých stanic bez nutnosti je obcházet a stahovat data z každé zvlášť. Pro zjednodušení práce operátora program umožňuje prohledat celý rozsah adres a najít všechny připojené a funkční stanice (viz Obr 5.20: Vyhledávání stanic v síti).



Obr 5.20: Vyhledávání stanic v síti

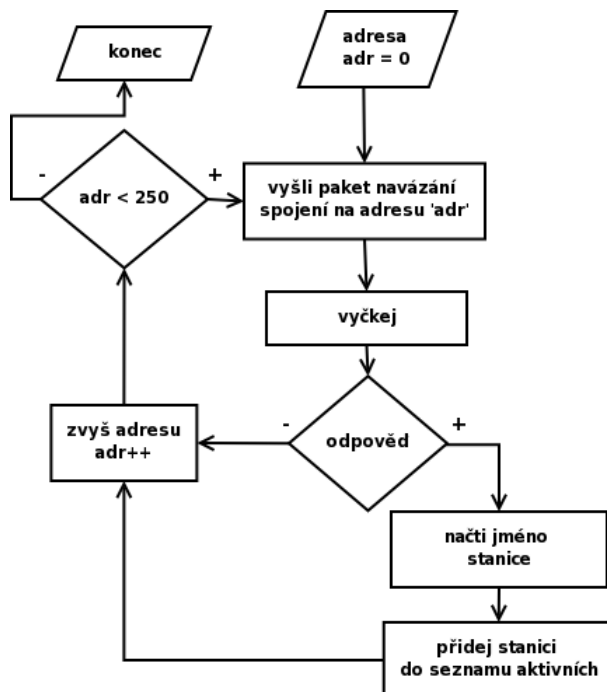
Operátor se pak mezi nimi může rychle přepínat pomocí kontextového menu hlavního formuláře (viz Obr 5.21: Přepínání nalezených stanic v síti).



Obr 5.21: Přepínání nalezených stanic v síti

Z implementačního hlediska je prohledávání sítě realizováno velmi jednoduše. Program postupně vysílá na adresy paket pro navázání spojení. V okamžiku, kdy obdrží z dané adresy odpověď, přeruší procházení sítě a přenechá základní informace o stanici, která se na adrese vyskytuje.

Zároveň přidá stanici do seznamu aktivních zařízení, čímž umožní obsluhu mezi nimi přepínat. Jedna z metod komunikační třídy obsahuje metodu specializovanou na tento typ úlohy s názvem *CheckPresence()*. Celý algoritmus vyhledávání pak demonstruje Obr 5.22: Algoritmus vyhledávání stanic v síti.



Obr 5.22: Algoritmus vyhledávání stanic v síti

Implementační detaily jsou zřejmé z následujících zdrojových kódů:

VyhledáníForm.pas

(formulář vyhledávání stanic v síti)

5.4 Kontextová nápověda k aplikaci

Pro řešení problémů a nejasností uživatele byl vytvořen kontextový návod k aplikaci. Ten je spolu s originálním návodem k jednotce (vytvořeným firmou Smart, spol. s r.o.) připojen k jednotce jako externí soubor *jsd600_jednotka (*.cnt, *.hlp)* a *jsd600program (*.cnt, *.hlp)*.

Oba manuály jsou dostupné z nabídky *Nastavení* hlavního okna aplikace. Nápovědu k aplikaci je také možné vyvolat stiskem klávesy F1 z jakéhokoliv okna, čímž se zobrazí stránka s tématem relevantním k aktivnímu oknu. Pro správnou funkci musí být oba soubory umístěny v pracovním adresáři aplikace.

5.5 Parní výpočty

Stanici ovládá procesor Philips PCB80C552-5 taktovaný na 18,432 MHz , takže v porovnání s PC má k dispozici jen velmi nízký výpočetní výkon zvláště v některých matematických operacích (například odmocnění nebo dělení). Z tohoto důvodu jsou některé výpočty ve stanici pouze aproximovány podle předpočítaných tabulek, čímž vzniká určitá chyba, která je však vzhledem k nepřesnosti měřících senzorů zcela zanedbatelná. Ve snaze co nejvíce se přiblížit interpretaci výpočtů stanice je tedy tato zjednodušená matematika použita v nezměněné formě i v aplikaci pro nastavení jednotky. Z hlediska implementace je tedy použito části kódu z programu stanice, který je pouze lehce upraven pro spolupráci se zbytkem systému. Jedná se převážně o parní tabulky vycházející z normy IAPWS-IF97, které definují stavy páry za různých podmínek a výpočty vycházející z kapitoly 3 Základy teorie měření energie. Autorem kódu je p. Ing. Vlastimil Jaroš.

Implementační detaily jsou zřejmé z následujících zdrojových kódů:

ParaVypoctyUnit.pas

(převzatý soubor funkcí pro parní výpočty)

6 Závěr

Ve této zprávě jsem se pokusil stručně nastínit problémy, se kterými jsem se musel seznámit a které bylo třeba vyřešit při implementaci ovládací aplikace pro průmyslovou matematickou jednotku JSD600. Práce chronologicky sleduje postup vývoje programu od úplných počátků seznámení se se zařízením až po samotnou implementaci systému. Teoretické znalosti o funkcích jednotky a o její celkové koncepci byly převážně předmětem mého semestrálního projektu. V diplomové práci jsem se soustředil na vývoj a ladění aplikace, popis jejího uživatelského rozhraní. Zároveň jsem se pokusil nastínit postup řešení některých problémů, které bylo potřeba během vývoje aplikace řešit. Vzhledem k rozsahu celého projektu jsem se záměrně snažil příliš nezacházet do implementačních detailů, ale spíše jsem se soustředil na popis základních funkcí, které již jsou do programu zahrnuty.

V současné době se aplikace nachází ve stadiu testovací verze, která je distribuována převážně stálým zákazníkům, kteří se podílí na doladění implementačních chyb a doplnění požadovaných funkcí. Mezi jedny z větších uživatelů jednotky patří například firma Škoda Auto a.s., která s pomocí jednotky JSD600, zabudované přímo do testovacího vozu, měří tepelnou účinnost nových typů automobilových chladičů.

Někteří další uživatelé programu (jednotky):

- Spirax Sarco (CZ, SK, PL)
- Teplárna Bruntál
- Teplárna Náchod
- I&CS Třebíč
- JSP Nová Paka
- Aliachem Pardubice
- Varmuza Hodonín

Aplikace je nedílnou součástí jednotky JSD600, kterou funkčně doplňuje. Z toho důvodu je všem uživatelům jednotky k dispozici bezplatně.

Samotná jednotka JSD600 se neustále dál vyvíjí a upravuje podle přání zákazníků, čímž stále přibývají další funkce, které je nutno do programu implementovat nebo modifikovat. Především se jedná o přidání podpory pro další čidla pro měření veličin, podpory pro moduly schopné komunikace s profesionálními zařízeními pro měření průtoku, která komunikují na základě protokolu HART a

přidání funkce bezdrátového přenosu dat z jednotky prostřednictvím IrDa. Pro tyto nové funkce jednotky je zároveň nutné patřičně upravit i obslužný program tak, aby nové funkce podporoval. Zároveň se chystá revize a optimalizace realizace jednotky, s čímž bude souviset zřejmě i částečná změna komunikačního protokolu. Do budoucna se tedy dále počítá s rozvojem a optimalizací programu i jednotky.

Zároveň s dalším rozvojem je nutné stále poskytovat zákazníkům technickou podporu při řešení problémů s nasazením jednotky v praxi a co nejvíce jim usnadnit přechod na nový program. Pomocnou rukou je kontextová nápověda aplikace, která se snaží uživateli vysvětlit význam funkcí jednotky.

Při vývoji jsem využil znalostí (především z oblasti periferních zařízení, architektury počítačů, binární interpretace čísel, algoritmizace a programování) získaných během studia. Jedná se však o první aplikaci tohoto zaměření a rozsahu, kterou jsem vlastními silami realizoval, proto jsou v programu použity některé neoptimální konstrukce, které bych dnes již nezvolil.

Literatura

- [1] SMART, spol s r.o.: JSD600 popis a návod k obsluze, firemní dokumentace, verze 202
- [2] JANDORA, R: Skupenské přeměny [online] 18. února 2001, [cit. 7. listopadu 2006],
Dostupné z <<http://www.sweb.cz/radek.jandora/f09.htm>>
- [3] KAŠÍK, Z: TZB-Info: Má pára budoucnost? [online] 10. května 2002, [cit. 28. října 2006],
Dostupné z: <<http://voda.tzb-info.cz/t.py?t=2&i=979>>
- [4] LEITNER, J. Termodynamika materiálů: Stavové chování a termodynamické vlastnosti čistých látek [online] březen 2005, [cit. 7. listopadu 2006], Dostupné z
<<http://www.vscht.cz/ipl/termodyn/termmatpdf/kapitola2.pdf>>
- [5] EDR, V.: TLAKinfo: Měření tepla obsaženého v kondenzátu [online] 18. června 2004, [cit. 2. listopadu 2006], Dostupné z <<http://www.tlakinfor.cz/t.py?t=2&i=716&h=119>>
- [6] ČMI Brno: Metodické pokyny pro metrologii (MPM 18-95), 1995, 12
- [7] BEJČEK, L. Přehled metod a snímačů měření průtoku. In ČMI Brno. *Metrologie průtoku 2005*. Brno, 2005. s. 105-138. ISBN 80-239-4312-X
- [8] SOCHOR, M. Objemové průtokoměry. In ČMI Brno. *Metrologie průtoku 2005*. Brno, 2005. s. 149-156.
ISBN 80-239-4312-X
- [9] TESAŘ, J, PRAŽÁK, D. Δp průtokoměry. In ČMI Brno. *Metrologie průtoku 2005*. Brno, 2005. s. 173-190.
ISBN 80-239-4312-X
- [10] KLAPETEK, P. Komunikační protokoly. In ČMI Brno. *Metrologie průtoku 2005*. Brno, 2005. s. 229-232. ISBN 80-239-4312-X.
- [11] SMART, spol s r.o.: Firemní dokumentace ke komunikačnímu protokolu JS600/MS100
- [12] BUCHALCEVOVÁ, A. *Vývoj aplikací v delphi*. Praha : VŠE, 2001. 209 s. ISBN 80-245-0150-3.
- [13] MOFFATT, Neil. *Delphi basics* [online]. 2002-2006 [cit. 2007]. Dostupný z WWW:
<<http://www.delphibasics.co.uk/>>
- [14] CodeProject. *CodeProject* [online]. 1999-2006 [cit. 2007]. Dostupný z WWW:
<<http://www.codeproject.com/>>

Seznam příloh

- Tabulka 1: Navázání spojení - dotaz
- Tabulka 2: Navázání spojení - odpověď
- Tabulka 3: Stav stanice - dotaz
- Tabulka 4: Stav stanice -odpověď
- Tabulka 5: Definice měřidla - dotaz
- Tabulka 6: Definice měřidla - odpověď
- Tabulka 7: Význam 9. slabiky definice měřidla
- Tabulka 8: Význam 7. slabiky definice měřidla
- Tabulka 9: Význam 89. slabiky definice měřidla
- Tabulka 10: Definice kanálu (PV) - dotaz
- Tabulka 11: Definice kanálu (PV) - odpověď
- Tabulka 12: Význam 8. slabiky definice kanálu (PV)
- Tabulka 13: Význam 21.-22. slabiky definice kanálu (PV)
- Tabulka 14: Definice stanice - dotaz
- Tabulka 15: Definice stanice - odpověď
- Tabulka 16: Popisek archivu - dotaz
- Tabulka 17: Popisek archivu - odpověď
- Tabulka 18: Totalizér tepla (archiv) - dotaz
- Tabulka 19: Totalizér tepla (archiv) - odpověď
- Tabulka 20: Hlášení stanice - dotaz
- Tabulka 21: Hlášení stanice - odpověď
- Tabulka 22: Patnáctiminutové maximum - dotaz
- Tabulka 23: Patnáctiminutové maximum - odpověď
- Tabulka 24: Letmé měření - zahájení měření
- Tabulka 25: Letmé měření - dotaz na výsledek
- Tabulka 26: Letmé měření - výsledek měření
- Tabulka 27: Sledování dat v reálném čase - dotaz
- Tabulka 28: Sledování dat v reálném čase - odpověď
- Tabulka 29: Monitorovaná data stanice - dotaz
- Tabulka 30: Monitorovaná data stanice - první paket odpovědi
- Tabulka 31: Monitorovaná data stanice - pokračovací paket odpovědi
- Tabulka 32: Programování vlastností stanice
- Tabulka 33: Programování vlastností kanálu (PV)
- Tabulka 34: Programování vlastností měřidla
- Tabulka 35: Zkrácený stav měřidla (registry) - dotaz

Tabulka 36: Zkrácený stav měřidla (registry) - odpověď

Tabulka 37: Význam bitů stavů páry

Tabulka 38: Význam bitů chybového statutu

Obr 1: Diagram organizační struktury aplikace

Programové vybavení pro komunikaci s měřičem tepla JSD600
Přílohy

Byte	Hodnota	Bytů	Popis	Typ
1	0x01	1	SOH	byte
2	0x81	1	Identifikace zprávy	byte
3	0x0A	1	Povel – navázání spojení	byte
4		1	Rezervováno	byte
5		1		byte
6		1		byte
7	adr	1	Adresa stanice	byte
8		1	Rezervováno	byte

Tabulka 1: Navázání spojení - dotaz

Byte	Hodnota	Bytů	Popis	Typ
1	0x01	1	SOH	byte
2	0x81	1	Identifikace zprávy	byte
3	0x0B	1	Povel dotazu + 1	byte
4		1	Rezervováno	byte
5		1	Rezervováno	byte
6		1	Rezervováno	byte
7		1	Rezervováno	byte
8		1	Rezervováno	byte

Tabulka 2: Navázání spojení - odpověď

Programové vybavení pro komunikaci s měřičem tepla JSD600
Přílohy

Byte	Hodnota	Bytů	Popis	Typ
1	0x01	1	SOH	byte
2	0x40	1	Identifikace příkazu	byte
3	0x10	1	Povel	byte

Tabulka 3: Stav stanice - dotaz

Byte	Hodnota	Bytů	Popis	Typ
1	0x01	1	SOH	byte
2	0x02	1	Identifikace příkazu	byte
3	0x11	1	Povel dotazu + 1	byte
4	délka	1	Délka packetu	byte
5	Zpráva	1	Číslo zprávy PC	byte
6	Povel	1	Poslední vyslaný povel	byte
7	Povel	1	Poslední přijatý povel	byte
8	Stav stanice	1	0 – čekání na zprávu	byte
			1 – čekání a potvrzení zprávy	
			2 – tato zpráva zrušila vysílání zprávy do stanice	
9-27	Rezerva	19		

Tabulka 4: Stav stanice -odpověď

Programové vybavení pro komunikaci s měřičem tepla JSD600

Přílohy

Byte	Hodnota	Bytů	Popis	Typ
1	0x01	1	SOH	byte
2	0x40	1	Identifikace příkazu	byte
3	0x26	1	Povel	byte

Tabulka 5: Definice měřidla - dotaz

Byte	Hodnota	Bytů	Popis	Typ
1	0x01	1	SOH	byte
2	0x02	1	Identifikace příkazu	byte
3	0x27	1	Povel dotazu + 1	byte
4	Délka	1	Délka packetu	byte
5-6	ID	2	Identifikace měřidla	byte
7	Typ	1	Varianta, způsob měření (hmotnostní, objemové)	byte
8	Rezervováno	1		byte
9	Format	1	Způsob měření (přímé)	byte
10	0x00	1	Pevná hodnota	byte
11-14	RK1	4	Maximální teplota páry	real
15-18	RK2	4	Minimální teplota páry	real
19-22	RK3	4	TK min	real
23-26	RK4	4	TK max	real
27-30	RK5	4	Korekce mokré páry	real
31-34	RK6	4	Max. Průtok 2. okruhu	real
35-38	RK7	4	Maximální tlak	real
39-42	RK8	4	Minimální tlak	real
43-46	RK9	4	Sytost	real
47-50	RK10	4	Hystereze	real
51-54	RK11	4	Min. Průtok 2. okruhu	real
55-58	RK12	4	Max. Průtok 1. okruhu	real
59-62	RK13	4	Min. Průtok 1. okruhu	real
63-66	RK14	4	Maximální energie 1	real
67-70	RK15	4	Maximální energie 2	real
71-72	IK1	2	Způsob měření	int
73-74	IK2	2	Interval P1	int
75-76	IK3	2	Interval P2	int
77-78	IK4	2	Interval Kond.	int
79-80	0x16	2	Pevná hodnota	int
81-84	Impkon rev	4	Převrácená hodnota imp. Koeficientu	real
85-88	Impkon	4	Impulsní koeficient	real
89-90	Ikon9	2	Analogový výstup	int
91	Rezervováno	1		byte

Tabulka 6: Definice měřidla - odpověď

Programové vybavení pro komunikaci s měřičem tepla JSD600

Přílohy

Bit	Význam
0	Varianta měření
1	Varianta měření
2	Varianta měření
3	Varianta měření
4	0 – 1.okruh hmotnostní metoda 1 – 1.okruh objemová metoda
5	0 – 2. okruh hmotnostní metoda 1 – 2. okruh objemová metoda
6	0 – vodní měřidlo 1 – parní měřidlo
7	Rezervováno

Tabulka 8: Význam 7. slabiky definice měřidla

Bit	Význam
0	Varianta měření
1	Varianta měření
2	Varianta měření
3	Varianta měření
4	Rezervováno
5	Rezervováno
6	1 – 1. okruh přímá metoda
7	1 – 2. okruh přímá metoda

Tabulka 7: Význam 9. slabiky definice měřidla

Bit	Význam
0	0 – 1.okruh podle PV 1-8 1 – 1. okruh vypočtený
1	0 – 2. okruh podle PV 1-8 1 – 2. okruh vypočtený
2	Rezervováno
3	Rezervováno
4	0 – jednotky 1. okruhu – tuny 1 – jednotky 1. okruhu – GJ
5	0 – jednotky 2. okruhu – tuny 1 – jednotky 2. okruhu – GJ
6	Rezervováno
7	Rezervováno

Tabulka 9: Význam 89. slabiky definice měřidla

Programové vybavení pro komunikaci s měřičem tepla JSD600

Přílohy

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x02	1	Identifikace zprávy	byte
3	0x24	1	Povel	byte
4	0x01	1	délka	byte
5	Kanál	1	Požadovaný kanál	byte

Tabulka 10: Definice kanálu (PV) - dotaz

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x02	1	Identifikace zprávy	byte
3	0x25	1	Povel dotazu + 1	byte
4	delka	1	Délka packetu	byte
5	Typ PV	1	typ měření+příznak lin. tabulky (bitově)	byte
6	Typ PV1	1	příznak funkce pro display stanice	byte
7	0x00	1	rezervováno	byte
8	vstupy	1	Definice vstupů stanice	byte
9-10	INT1	2	Vzorkovací interval ukládání dat do paměti	int
11-12	0x1000	2	pevná hodnota	int
13-14	Blocksize	2	velikost paměti alokované pro kanál	int
15-16	0x00	2	rezervováno	int
17-18	DMH	2	Minimální mez výstupů	int
19-20	HMH	2	Maximální mez výstupů	int
21	výstup	1	Definice výstupů stanice (bitově)	byte
22	0x00	1	rezervováno	
23-24	Dolní Mez	2	Minimální hodnota vstupu (celočíslně)	int
25-26	Horní mez	2	Maximální hodnota vstupu (celočíslně)	int
27-28	0x00	2	pevná hodnota	
30-37	NamePV	2	Název kanálu	string
38-55	TextPV	18	Text kanálu (poznámka)	string
58-61	Jednotka PV	4	Jednotka veličiny měřené kanálem	string
62-67+ 56-57	Typ čidla	8	Název typu čidla	string
68-75	MulPar	8	Multiplikační koeficient pro Lin. tabulku	real
76-79	AddPar	4	Aditivní koeficient pro Lin. tabulku	real
80-83	FromV	4	Minimální hodnota vstupu (reálně)	real
84-87	ToV	4	Maximální hodnota vstupu (reálně)	real
88-89	Form	2	rezervováno	
90-93	TKV	4	Teplota kalibrační vody	real
94	Status	1	Příznak existence Lin. tabulky	byte
95-98	Tkoef	4	teplotní koeficient	real
99-100	Ntab	2	Počet řádků Lin. tabulky	int
Následující část se opakuje Ntab-krát – jednotlivé řádky Lin.tabulky				
101-102	převodník	2	hodnota převodníku (pixely)	int
103-104	Hodnota	2	přiřazená hodnota	int

Tabulka 11: Definice kanálu (PV) - odpověď

Programové vybavení pro komunikaci s měřičem tepla JSD600

Přílohy

Hodnota	Význam
0x00 – 0x07	Analogový vstup 1-8
0x09 – 0x0E	Digitální vstup 1-6
0x20 – 0x23	Frekvenční vstupy 1-4
0x30 – 0x33	Impulsní čítače 1-4

Tabulka 12: Význam 8. slabiky definice kanálu (PV)

Hodnota	Význam
0x00	Žádný výstup
0x80 – 0x81	Int. analogový výstup 1-2
0x88 – 0x8F	Ext. analogový výstup 1-8
0xA0 – 0xA5	Int. digitální výstupy 1-6
0xA8 – 0xAF	Ext. digitální výstupy 1-8
Použití inverzní signalizuje 2. bit	

Tabulka 13: Význam 21.-22. slabiky definice kanálu (PV)

Programové vybavení pro komunikaci s měřičem tepla JSD600

Přílohy

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x40	1	Identifikace příkazu	byte
3	0x22	1	Povel	byte

Tabulka 14: Definice stanice - dotaz

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x02	1	Identifikace příkazu	byte
3	0x23	1	Poveldotazu + 1	byte
4	delka	1	Délka packetu	byte
5	0x01	1		byte
6	0x10	1		byte
7-8	addr	1	Adresa stanice	byte
9-10	status stanice	1	0 – stanice v poruše	int
			1 – stanice není inicializovaná	
			2 – stanice je inicializovaná	
11-12	pamet	1	Velikost paměti stanice	int
13-35		23	Rezervováno	
36-55	meridlo	20	Název použitého měřidla	string
56-60	nazev	5	Název stanice	string

Tabulka 15: Definice stanice - odpověď

Programové vybavení pro komunikaci s měřičem tepla JSD600
Přílohy

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x02	1	Identifikace příkazu	byte
3	0x2B	1	Povel	byte
4	0x01	1	Délka packetu	byte
5	cislo	1	Číslo řádku požadovaného popisku	byte

Tabulka 16: Popisek archivu - dotaz

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x02	1	Identifikace příkazu	byte
3	0x2C	1	Povel příkazu + 1	byte
4	delka	1	Délka packetu	byte
5		1	rezervováno	byte
6		1	rezervováno	byte
7-23	Název		Název položky archivu	string
24-26	jednotka		Jednotka položky archivu	string
27-34			rezervováno	

Tabulka 17: Popisek archivu - odpověď

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x02	1	Identifikace packetu	byte
3	0x5E	1	Povel	byte
4	0x02	1	Délka packetu	byte
5	číslo	1	Identifikační číslo požadovaného archivu	byte
6	0x61	1	Rezervováno	byte

Tabulka 18: Totalizér tepla (archív) - dotaz

Programové vybavení pro komunikaci s měřičem tepla JSD600

Přílohy

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x02	1	Identifikace packetu	byte
3	0x5F	1	Povel dotazu + 1	byte
4	delka	1	Délka packetu	byte
5-8	Q1 suchá	4	Teplo 1. okruhu dodané v suché páře	real
9-12	Q1 mokrá	4	Teplo 1. okruhu dodané v mokré páře	real
13-16	Q2 suchá	4	Teplo 2. okruhu dodané v suché páře	real
17-20	Q2 mokrá	4	Teplo 2. okruhu dodané v mokré páře	real
21-24	Q1 kond.	4	Teplo vratného kondenzátu 1. okruhu	real
25-28	M1 suchá	4	Množství suché páry 1. okruhu	real
29-32	M1 mokrá	4	Množství mokré páry v 1. okruhu	real
33-36	M2 suchá	4	Množství suché páry 2. okruhu	real
37-40	M2 mokrá	4	Množství mokré páry v 2. okruhu	real
41-44	M1 kond.	4	Množství vratného kondenzátu 1. okruhu	real
45-48	Čas1 mokrá	4	Čas dodávky mokré páry v 1. okruhu	real
49-52	Čas1 m.p.	4	Čas po který byla pára v 1. okruhu mimo parametry (neměřená)	real
53-56	Čas2 mokrá	4	Čas dodávky mokré páry v 2. okruhu	real
57-60	Čas2 m.p.	4	Čas po který byla pára v 2. okruhu mimo parametry (neměřená)	real
61-64	Čas vyp.	4	Čas po který bylo měřidlo vypnuto	real
65-68	Čas zahájení	4	Čas zahájení měření	sekundočas

Tabulka 19: Totalizér tepla (archiv) - odpověď

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x40	1	Identifikace packetu	byte
3	0x47	1	Povel	byte

Tabulka 20: Hlášení stanice - dotaz

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x02	1	Identifikace packetu	byte
3	0x48	1	Povel dotazu + 1	byte
4	delka	1	Délka datagramu	byte
5	počet	1	Celkový počet hlášení stanice	byte
6	0x00	1	Rezervováno	
7-10	sekundočas	4	Čas hlášení	sekundocas
11	0x00	1	Rezervováno	
12	příznak hlášení	1	Příznak hlášení	byte
			0x03 – začátek výpadku napětí	
			0x04 – konec výpadku napětí	
			0x0E – zahájení měření	

Tabulka 21: Hlášení stanice - odpověď

Programové vybavení pro komunikaci s měřičem tepla JSD600
Přílohy

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x40	1	Identifikace příkazu	byte
3	0x4D	1	Povel	byte

Tabulka 22: Patnáctiminutové maximum - dotaz

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x02	1	Identifikace packetu	byte
3	0x4E	1	Povel dotazu + 1	byte
4	delka	1	Délka packetu	byte
5-8	akt1	4	Aktuální stav 1. okruhu	real
9-12	akt2	4	Aktuální stav 2. okruhu	real
13-16	max1	4	Maximální stav 1. okruhu	real
17-20	cas1	4	Čas maximálního stavu 1. okruhu	sekundocas
21-24	max2	4	Maximální stav 2. okruhu	real
25-28	cas2	4	Čas maximálního stavu 2. okruhu	sekundocas

Tabulka 23: Patnáctiminutové maximum - odpověď

Programové vybavení pro komunikaci s měřičem tepla JSD600

Přílohy

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x02	1	Identifikace příkazu	byte
3	0x49	1	Povel	byte
4	delka	1	Délka packetu	byte
5	pasmo	1	Měřené pásmo	byte
6	interval	1	Interval měření	byte
7	buffer	1	Nulování bufferu	byte
			0 – nulovat buffer	
			1 – nenulovat buffer	

Tabulka 24: Letmé měření - zahájení měření

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x40	1	Identifikace příkazu	byte
3	0x4B	1	Povel	byte

Tabulka 25: Letmé měření - dotaz na výsledek

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x02	1	Identifikace příkazu	byte
3	0x4C	1	Povel dotazu + 1	byte
4	delka	1	Délka packetu	byte
5-8	teplo S	4	Změřené teplo suché páry	real
9-12	teplo M	4	Změřené teplo mokré páry	real
13-16	hmot S	4	Změřená hmotnost suché páry	real
17-20	hmot M	4	Změřená hmotnost mokré páry	real
21-24	cas S	4	Doba po kterou byla pára v suchém stavu	real
25-28	cas M	4	Doba po kterou byla pára v mokrému stavu	real
29-32	tepl. max S	4	Maximální teplota mokré páry	real
33-36	tepl. max M	4	maximální teplota suché páry	real
37-40	tlak max S	4	Tlak suché páry	real
41-44	tlak max M	4	Tlak mokré páry	real

Tabulka 26: Letmé měření - výsledek měření

Programové vybavení pro komunikaci s měřičem tepla JSD600

Přílohy

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x40	1	Identifikace příkazu	byte
3	0x43	1	Povel	byte

Tabulka 27: Sledování dat v reálném čase - dotaz

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x02	1	Identifikace příkazu	byte
3	0x44	1	Povel dotazu + 1	byte
4	Delka		Délka datagramu	byte
5-8	Sekundocas	4	Čas aktuálních dat	Sekundočas
9	0x11	1	Oddělovač	byte
10-11	Kanál1	2	Hodnota 1.kanálu	int
12-13	0x00	1	Rezervováno	byte
14	0x11	1	Oddělovač	byte
15-16	Kanál2	2	Hodnota 1.kanálu	int
17-18	0x00	1	Rezervováno	byte
19	0x11	1	Oddělovač	byte
...
45-46	Kanál8	2	Hodnota 8.kanálu	int
47-48	0x00	1	Rezervováno	byte
49	0x11	1	Oddělovač	byte

Tabulka 28: Sledování dat v reálném čase - odpověď

Programové vybavení pro komunikaci s měřičem tepla JSD600

Přílohy

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x02	1	Identifikace příkazu	byte
3	0x45	1	Povel	byte
4	Delka	1	Délka datagramu	byte
5	Kanal	1	Číslo požadovaného kanálu	byte

Tabulka 29: Monitorovaná data stanice - dotaz

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	Id	1	Identifikace příkazu	byte
			0x02 – datagram nemá pokračování	
			0x2a – datagram má pokračování	
3	0x46	1	Povel dotazu + 1	byte
4	Delka	1	Délka zprávy	byte
5	Kanál	1	Číslo posílaného kanálu	byte
6-9	Sekundočas	4	Čas načtení dat	Sekundočas
10-13	Sekundočas	4	Čas posledního programování stanice	Sekundočas
14	Hodnota 1	1	Hodnota jednoho vzorku kanálu	byte
...
	Hodnota n	1		byte

Tabulka 30: Monitorovaná data stanice - první paket odpovědi

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	Id	1	Identifikace příkazu	byte
			0x02 – datagram nemá pokračování	
			0x2a – datagram má pokračování	
3	Delka		Délka zprávy	byte
4	Hodnota 1	1	Hodnota jednoho vzorku kanálu	byte

n	Hodnota n	1		byte

Tabulka 31: Monitorovaná data stanice - pokračovací paket odpovědi

Programové vybavení pro komunikaci s měřičem tepla JSD600

Přílohy

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x40	1	Identifikace příkazu	byte
3	0x20	1	Povel	byte
4	delka	1	Délka packetu	byte
Packet pokračuje 5. bytem packetu pro načtení definice stanice ve stejném tvaru				

Tabulka 32: Programování vlastností stanice

1	0x01	1	SOH	byte
2	0x02	1	Identifikace příkazu	byte
3	0x21	1	Povel	byte
4	delka	1	Délka packetu	byte
5	kanal	1	Programovaná veličina (PV)	byte
Packet pokračuje 5. bytem packetu pro načtení definice veličiny (PV) ve stejném tvaru.				

Tabulka 33: Programování vlastností kanálu (PV)

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x02	1	Identifikace příkazu	byte
3	0x2D	1	Povel	byte
4	délka	1	Délka packetu	byte
Packet pokračuje 4. bytem packetu pro načtení definic měřidla ve stejném tvaru				

Tabulka 34: Programování vlastností měřidla

Programové vybavení pro komunikaci s měřičem tepla JSD600

Přílohy

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	0x81	1	Identifikace zprávy	byte
3	0x07	1	Povel	byte
4	delka	1	Délka zprávy	byte
5	adresa	1	Adresa stanice	byte

Tabulka 35: Zkrácený stav měřidla (registry) - dotaz

Byte	Hodnota	Bytů	Popis	typ
1	0x01	1	SOH	byte
2	xx	1	Identifikace zprávy	byte
3	adresa	1	Adresa stanice	byte
4	0x44	1	Délka zprávy	byte
5	status1	1	Status páry v 1.okruhu	byte
6	chyba1	1	Kód chyby měření 1.okruhu	byte
7	status2	1	Status páry v 2.okruhu	byte
8	chyba2	1	Kód chyby měření 2.okruhu	byte
9-12	teplota1	4	Teplota 1.okruhu	real
13-16	tlak1	4	Tlak 1.okruhu	real
17-20	entalpie1	4	Entalpie 1.okruhu	real
21-24	vykon1	4	Tepelný výkon 1.okruhu	real
25-28	TSP1	4	Stav registru tepla v suché páře 1.okruhu	real
29-32	TSM1	4	Stav registru tepla v mokré páře 1.okruhu	real
33-36	MSP1	4	Stav registru množství suché páry 1.okruhu	real
37-40	MMP1	4	Stav registru množství mokré páry 1.okruhu	real
41-44	teplota2	4	Teplota 2.okruhu	real
45-48	tlak2	4	Tlak 2.okruhu	real
49-52	entalpie2		Entalpie 2.okruhu	real
53-56	vykon2	4	Tepelný výkon 2.okruhu	real
57-60	TSP2	4	Stav registru tepla v suché páře 2.okruhu	real
61-64	TSM2	4	Stav registru tepla v mokré páře 2.okruhu	real
65-68	MSP2	4	Stav registru množství suché páry 2.okruhu	real
69-72	MMP2	4	Stav registru množství mokré páry 2.okruhu	real

Tabulka 36: Zkrácený stav měřidla (registry) - odpověď

Programové vybavení pro komunikaci s měřičem tepla JSD600

Přílohy

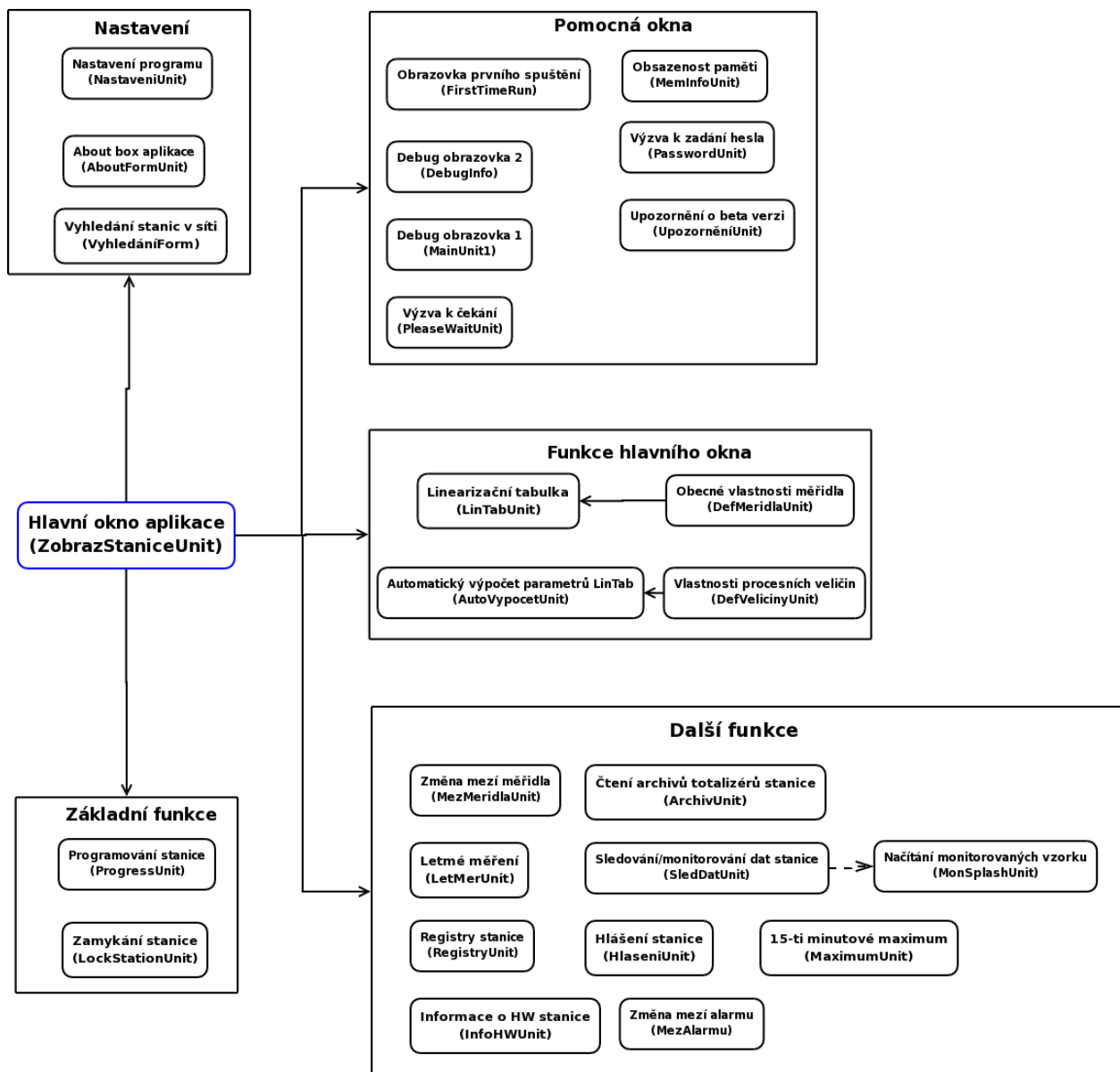
bit	význam
0	neměřeno
1	přehřátá pára
2	sytá pára
3	mokrá pára
4	pára mimo parametry
5	stav nebyl od posledního čtení aktualizován
6	stav byl od posledního čtení aktualizován

Tabulka 37: Význam bitů stavů páry

bit	význam
0	nepoužitý
1	průtok neměřen nebo mimo nastavený rozsah
2	tlak neměřen nebo mimo nastavený rozsah
3	teplota neměřena nebo mimo nastavený rozsah
4	mokrá pára
5	mokrá pára
6	pára mimo parametry

Tabulka 38: Význam bitů chybového statutu

Programové vybavení pro komunikaci s měřičem tepla JSD600
Přílohy



Obr 1: Diagram organizační struktury aplikace