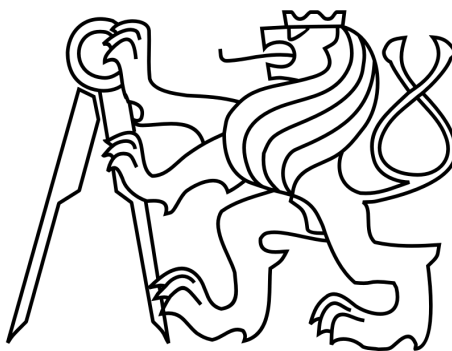


České vysoké učení technické v Praze

Fakulta strojní



Bakalářská práce

Sběr dat z pásových vah třídící linky kamene

Jaroslav Bušek

Vedoucí práce: Ing. Marie Martinásková PhD.

srpen 2009

Studijní program: Strojírenství, bakalářský

Obor: Informační a automatizační technika

Poděkování

Děkuji Ing. Marii Martináskové PhD. za vedení práce a za její pomoc při teoretickém rozboru a popisu úlohy. Dále bych chtěl poděkovat firmě Ing. Jaroslav Bušek – TEVAS za poskytnutí veškerého zkušebního zařízení a potřebného technického zázemí a firmě Hynek Pangrác – HYPEL, zejména jejímu řediteli Ing. Hynku Pangrácovi, za významnou pomoc při realizaci programového vybavení pro zařízení firmy HYPEL.

Prohlášení

Prohlašuji, že předkládanou diplomovou práci jsem zpracoval samostatně a souhlasím s tím, že její výsledky mohou být dále využity podle uvážení vedoucího diplomové práce Ing. Marií Martináskové PhD. jako jejího spoluautora. Souhlasím také s případnou publikací výsledků diplomové práce nebo její podstatné části, pokud budu uveden jako spoluautor.

Datum

Podpis

Abstrakt

Cílem této práce je navrhnout a realizovat sběr dat z pásových vah třídící linky kamene s ohledem na dané požadavky a technické možnosti stávajícího řídicího systému.

Na základě informací, získaných analýzou stávajícího řídicího systému, jsou s přihlédnutím k požadavkům navrženy možné varianty řešení přenosu dat. Z nich je po zvážení výhod a nevýhod vybrána nejvhodnější varianta založená na přenosu dat s využitím GSM sítě. Pro tuto variantu jsou vybrány technické prostředky a provedena praktická realizace, jež obnáší volbu formátu sbíraných dat, konfiguraci zařízení a vytvoření programového vybavení pro tato zařízení.

Klíčová slova: GSM modem, sběr dat, FTP, pásová váha, kontinuální vážení

Obsah

Úvod.....	1
1 Popis stávajícího systému.....	3
1.1 Řízená technologie.....	3
1.2 Řídicí systém.....	4
1.2.1 Vážicí stolice	4
1.2.2 Vyhodnocovací jednotka	5
1.3 Použité principy měření	6
1.3.1 Vážení pomocí tenzometrických snímačů.....	6
1.3.2 Měření rychlosti indukčním snímačem	8
1.3.3 Kontinuální vážení na pásové váze.....	10
2 Návrh a výběr řešení.....	14
2.1 Východiska návrhu	14
2.1.1 Požadavky provozovatele	14
2.1.2 Provozní podmínky.....	15
2.1.3 Technické možnosti stávajícího systému.....	15
2.2 Návrh variant řešení	17
2.2.1 Optická přenosová cesta	18
2.2.2 Metalická přenosová cesta.....	18
2.2.3 Rádiová přenosová cesta.....	20
2.3 Výběr varianty řešení.....	22
3 Výběr a popis zařízení.....	24
3.1 Výběr zařízení	24
3.2 Programovatelný převodník RoutPro FX	25
3.2.1 Popis převodníku RoutPro FX.....	25
3.2.2 Popis jazyka SIMPLE3.....	26
3.2.3 Program HypEd4	28
3.3 GSM modem MAESTRO 100.....	29
3.3.1 Základní informace o GSM/GPRS	29
3.3.2 Popis GSM modemu MAESTRO 100.....	30

3.3.3 Konfigurace a ovládání GSM modemu	32
4 Realizace vybrané varianty řešení	33
4.1 Návrh struktury dat	33
4.2 Konfigurace GSM modemu a určení příkazů odeslání.....	36
4.2.1 Formát AT příkazů	37
4.2.2 Konfigurace	37
4.2.3 Určení příkazů pro odeslání dat pomocí FTP	39
4.3 Implementace navržené datové struktury do T3A	41
4.4 Vytvoření obslužného programu pro RoutPro FX.....	42
4.4.1 Struktura programu	42
4.5 Ladění	46
4.5.1 Návrh PHP skriptu a databáze	48
Závěr	49
Použité zdroje.....	51
Přílohy	53
Příloha A: Obsah příloženého CD	54
Příloha B: RoutPro FX - zdrojový text IOHND.LIB.....	55
Příloha C: RoutPro FX - zdrojový text CMDS.LIB	60
Příloha D: RoutPro FX - zdrojový text PROG.STP.....	63
Příloha E: Hlavní stránka webové aplikace	67
Příloha F: Textový přehled sebraných dat za jeden den.....	68
Příloha G: Grafický přehled sebraných dat za jeden den.....	69

Seznam obrázků

Obr. 1.1: Schéma uspořádání stávajícího systému.....	4
Obr. 1.2: Měřicí úsek pásové váhy	5
Obr. 1.3: Kompaktní řídicí terminál T3A (zdroj [18]).....	6
Obr. 1.4: Funkční skupiny indukčního snímače	8
Obr. 1.5: Indukční snímač (zdroj [20])	9
Obr. 1.6: Schéma měření obvodové rychlosti válečku	9
Obr. 1.7: Schématické znázornění hmotnostního toku na pásovém dopravníku	11
Obr. 2.1: Obecný návrh varianty řešení – komunikační řetězec	17
Obr. 3.1: Konkrétní podoba komunikačního řetězce	25
Obr. 3.2: Programovatelný převodník RoutPro FX (zdroj [18]).....	26
Obr. 3.3: Hlavní okno programu HypEd4.....	28
Obr. 3.4: Servisní převodník CA1 (zdroj [18]).....	29
Obr. 3.5: GSM/GPRS modem Maestro 100 (zdroj [19]).....	31
Obr. 3.6: Komunikace s modemem v prostředí programu Hyperterminál	32
Obr. 4.1: Formát sbíraných dat	34
Obr. 4.2: Formát a velikost jednoho záznamu v datové paměti.....	36
Obr. 4.3: Konečný zvolený formát a velikost datového záznamu	36
Obr. 4.4: Struktura obslužného programu převodníku RoutPro FX	43
Obr. 4.5: Ukázka textového řetězce reprezentujícího jeden záznam	44
Obr. 4.6: Ukázka názvu textového souboru	44
Obr. 4.7: Stavový diagram programové sekvence zadávání AT příkazů.....	46

Seznam tabulek

Tab. 3.1: Základní technické parametry zařízení RoutPro FX.....	25
Tab. 3.2: Základní datové typy jazyka SIMPLE3.....	27
Tab. 3.3: Základní technické parametry GSM/GPRS modemu MAESTRO 100.....	31

Slovník použitých zkratek

APN	Access Point Name
AT	Attention
CD	Compact Disc
CPU	Central Processing Unit
CSD	Circuit Switched Data
DDE	Dynamic Data Exchange
DIN	Deutsches Institut für Normung
DVD	Digital Video Disc
EEPROM	Electrically Erasable Programmable Read-Only Memory
FIFO	First In, First Out
FTP	File Transfer Protocol
GD	Graphics Draw
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HTTP	Hypertext Transfer Protocol
iPC	Industrial PC
LAN	Local Area Network
LCD	Liquid Crystal Display
OS	Operating System
OSI	Open Systems Interconnection
OSPIH	Internet Hosted Octet Stream Protocol
PC	Personal Computer
PHP	Hypertext Preprocessor
PIN	Personal Identification Number
PLC	Programmable Logic Controller
PNG	Portable Network Graphics
POP	Post Office Protocol
PPP	Point-to-Point Protocol
RAM	Random-Access Memory

SCADA	Supervisory Control And Data Acquisition
SIM	Subscriber Identity Module
SMA	SubMiniature version A
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
TCP/IP	Transmission Control Protocol/Internet Protocol
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
WLAN	Wireless LAN

Úvod

Třídící linka kamene je spolu s drtiči a pásovými dopravníky součástí technologie recyklace sutí, která se nachází v areálu Poldi v Kladně. Jejím provozovatelem je firma Zbyněk Karas – DESTRO (dále jen DESTRO). Nakládací pásové dopravníky jsou vybaveny vážicími stolicemi a řídicím systémem, pomocí něhož je mimo jiné možné dávkovat předvolené množství materiálu a také sledovat tok materiálu během provozu. Cílem této práce je navrhnout a realizovat sběr dat z již hotového a funkčního řídicího systému pásové váhy, a to s ohledem na požadavky provozovatele, na možnosti rozšíření stávajícího řídicího systému a na provozní podmínky technologie. Výrazné omezení je dáno nemožností měnit hardwarovou konfiguraci stávajícího řídicího systému, jež dodávala firma Ing. Jaroslav Bušek (dále jen TEVAS). Další omezení se týká obslužného programu řídicí jednotky, v němž mám povoleno definovat pouze nové datové struktury. Případné nutné změny v řídicím algoritmu provede sama firma TEVAS.

Před samotným návrhem variant řešení je třeba se seznámit s řízenou technologií, kterou je v tomto případě pásový dopravník, a jejím řídicím systémem. Důležité je objasnit princip kontinuálního vážení na pásovém dopravníku a pomocí teoretického rozboru založeného na fyzikálních principech určit základní výpočtové vztahy a případně vyslovit předpoklady, které zjednoduší praktický výpočet. Dále je nutné popsat konfiguraci stávajícího řídicího systému, seznámit se s jeho komponentami a zjistit možnosti rozšíření.

Na základě předchozí analýzy je již možné provést návrh obecných variant řešení problému. Jejich princip bude značně ovlivněn rozšiřitelností stávajícího řídicího systému. Při výběru vhodné varianty je nutné přihlídnout k mnoha faktorům, mezi něž například patří preferenční požadavky provozovatele, dostupnost komunikačních sítí, provozní podmínky a jiné.

Realizace vybrané varianty obnáší výběr potřebných zařízení, jejich kompletaci, vytvoření softwarového vybavení a odladění. Volba konkrétního zařízení je řešena formou obecného seznámení s principem činnosti a výběrem typu dle

doporučení firmy TEVAS, která vybraná zařízení následně zakoupila a poskytla ke zkoušení.

1 Popis stávajícího systému

1.1 Řízená technologie

Řízená technologie je součástí drticí a třídící linky firmy DESTRO, která slouží k zpracování stavebních odpadů a strusky z oceláren Poldi a jiných ocelářenských podniků, ležících v jejím okolí. Linka se nachází na Dříní v areálu Poldi v Kladně a stojí přímo na navezené suti. Sídlo firmy není totožné s místem provozovny linky a je vzdálené přibližně 1,5 km vzdušnou čarou od uvedené provozovny.

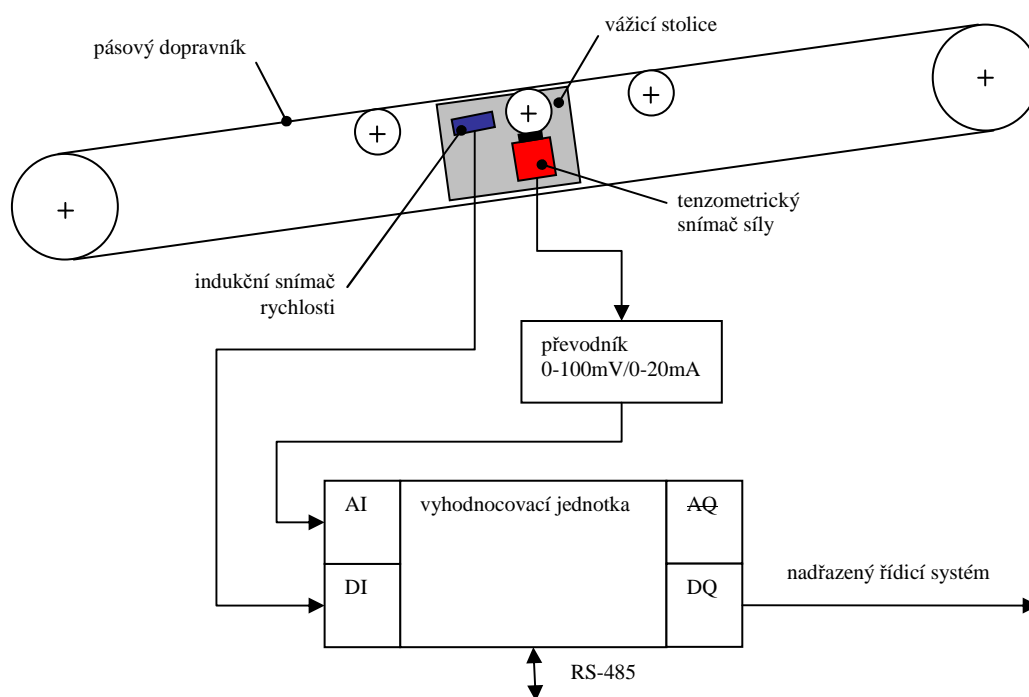
Drtecí a třídící linka se skládá z magnetického separátoru kovů, třídícího roštu, drtiče suti a pásových dopravníků, které slouží k přesunu materiálu mezi jednotlivými zařízeními linky a k naložení zpracované suti na nákladní vozidla. Ovládání technologie je umístěno v centrálním velínu na vyvýšeném místě přímo u linky. V jeho těsné blízkosti se též nachází hlavní rozváděcí skříň, ve které jsou sdruženy všechny hlavní řídicí prvky.

Řízenou technologií je výstupní pásový dopravník drticí a třídící linky, který slouží k nakládání zpracované suti na nákladní vozidla. Jelikož je prodejní cena suti vztahena na jednotku hmotnosti, je pro určení celkové ceny naložené suti nutné znát její hmotnost. Celková hmotnost se před zařazením řídicího systému určovala zvážením naloženého vozidla na mostní váze, která se nachází u brány areálu. Nakládání bylo řízeno ručně obsluhou. Aby nedošlo k přeplnění nákladního prostoru vozidla, byl nutný nepřetržitý dozor. Zařazením řídicího systému bylo určování hmotnosti přesunuto z mostní váhy přímo na pásový dopravník, kde probíhá vážení již během nakládání. Vyhodnocovací jednotka řídicího systému umožňuje mimo jiné měřit i hmotnost přepraveného materiálu a navážit zvolenou hmotnost - dávku, a tudíž nepřetržitý dozor již není nutný.

Linka běží v nepřetržitém provozu průběžně celý týden a v případě potřeby i mimo pracovní dobu – v nočních hodinách a o svátcích. Pravidelná odstávka je vždy mezi 11. hodinou večer a 5. hodinou ráno.

1.2 Řídicí systém

Řídicí systém výstupního pásového dopravníku se skládá z vyhodnocovací jednotky dodané firmou Hynek Pangrác – HYPEL (dále jen HYPEL) a jednopražcové vážicí stolice od firmy TEVAS. Uvedený řídicí systém pásového dopravníku je podřízen hlavnímu řídicímu systému, který řídí chod celé technologie včetně drtičů, třídících roštů a dalších zařízení. Výstupní signály z vyhodnocovací jednotky pásového dopravníku jsou přivedeny do nadřazeného řídicího systému, kde se z nich teprve vyvodí příslušné akční zásahy v technologii, jak je to znázorněno ve schématickém zobrazení (Obr. 1.1). Pokud by výstupní signály z vyhodnocovací jednotky pásového dopravníku přímo řídily například chod motoru pásového dopravníku, mohlo by dojít při zastavení pohonu k hromadění neodváděného materiálu a následnému poškození zařízení.



Obr. 1.1: Schéma uspořádání stávajícího systému

1.2.1 Vážicí stolice

Vážicí stolice je zařazena do původního pásového dopravníku, kde plní funkci měřicího úseku (Obr. 1.2). Součástí vážicí stolice je tenzometrický snímač síly

OT-11 s jmenovitým zatížením 1,5 kN a indukční snímač rychlosti pásu SELET B01E188N0C5.

Výstupní signál ze snímače síly OT-11 je přiveden přes převodník 0-100 mV/0-20 mA na analogový vstup vyhodnocovací jednotky. Indukční snímač rychlosti je umístěn různoběžně s osou válečku u jeho ukotvení, kde snímá impulzy ze tří clonek rovnoměrně rozmístěných po obvodu válečku. Výstupní signál snímače rychlosti je přiveden přímo na digitální vstup vyhodnocovací jednotky, ve které je čítání impulsů zajištěno programově kontrolou příslušného vstupu v každé smyčce uživatelského programu.



Obr. 1.2: Měřicí úsek pásové váhy

1.2.2 Vyhodnocovací jednotka

Vyhodnocovací jednotkou je kompaktní řídicí terminál T3A s LCD (Liquid Crystal Display) displejem o 4x20 znacích a průmyslovou membránovou klávesnicí (Obr. 1.3). Tento terminál je určen pro aplikace, ve kterých spolupracuje více PLC v rozsáhlém řídicím systému, a proto je vybaven malým počtem periférií. Jejich počet je však dostačující v případě řízení méně náročných technologií jako je tato. Varianta terminálu T3A, použitá v tomto případě, je navíc vybavena obvodem reálného času a větší zálohovanou datovou pamětí.

Dále je T3A vybaven sériovou sběrnici RS-485, která slouží jednak k zavádění uživatelského programu, jednak ke spojení s dalšími zařízeními v síti

automatizačních prostředků firmy HYPEL, která komunikují síťovým protokolem PESNET 3.32. Při použití automatizačních prostředků jiných firem, která využívají též sériovou sběrnici RS-485, není možné zaručit plnou kompatibilitu vzájemné komunikace všech zařízení v síti.



Obr. 1.3: Kompaktní řídicí terminál T3A (zdroj [18])

Zdrojový text uživatelského programu programovatelného terminálu se vytváří v programovacím jazyce SIMPLE3, který je specifický pro programovatelné automatizační prostředky firmy HYPEL. Vytvořený zdrojový text je následně nutné zkompilovat v programu HypEd4 a za použití servisního převodníku CA1 nebo převodníku USB485 zavést výsledný binární soubor do programové paměti terminálu. Oba převodníky jsou dodávány firmou HYPEL.

1.3 Použité principy měření

1.3.1 Vážení pomocí tenzometrických snímačů

Základním prvkem tenzometrického snímače je odporový tenzometr, který je pevně spojen s namáhaným tělesem snímače. Tenzometr je zařízení určené k měření mechanického napětí vyvolaného mechanickým namáháním vnějšími silami nebo vnitřním působením. Funkce tenzometru je založena na změně elektrického odporu v závislosti na deformaci.

Teoretickým východiskem pro odvození vztahu (1), pomocí kterého lze určit elektrický odpor v závislosti na deformaci, je Hookeův zákon a závislost elektrického odporu R obecného vodiče s měrným elektrickým odporem ρ na jeho délce l a průřezu S .

$$R = \rho \cdot \frac{l}{S} \quad (1)$$

Jestliže dojde k deformaci vodiče tahem, jeho délka se prodlouží o Δl , jeho průřez se zmenší o ΔS a také dojde ke změně měrného elektrického odporu $\Delta \rho$. Tuto změnu lze vyjádřit totálním diferenciálem (2).

$$dR = \frac{\partial R}{\partial \rho} d\rho + \frac{\partial R}{\partial l} dl + \frac{\partial R}{\partial S} dS \quad (2)$$

Úpravou vztahu (2) a vyjádřením veličin lze nakonec získat obecný vztah pro poměrnou změnu odporu $\Delta R / R_0$ v závislosti na deformaci (3), ve kterém je R_0 elektrický odpor ve výchozím stavu bez mechanického namáhání a ε relativní prodloužení. Konstanty k_i vyjadřují citlivost tenzometru.

$$\frac{\Delta R}{R_0} = k_1 \cdot \varepsilon + k_2 \cdot \varepsilon^2 + k_3 \cdot \varepsilon^3 + \dots \quad \varepsilon = \frac{\Delta l}{l} \quad (3)$$

Odporové tenzometry se dělí na kovové (drátkové, fóliové, vrstevné) a polovodičové (monokrystalické, polykrystalické). Kovové tenzometry jsou zpravidla vyrobeny z konstantanu a mají lineární charakteristiku. Na rozdíl od kovových je závislost polovodičových tenzometrů nelineární. Jsou vyráběny z monokrystalu polovodiče (např. křemík, germanium) s příměsemi. Vztahy pro výpočet relativní změny odporu se pro oba druhy liší (4).

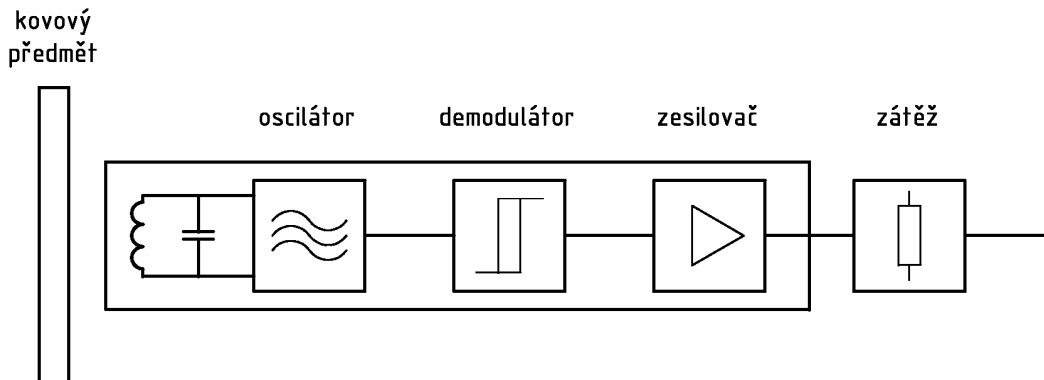
$$\begin{array}{l} \text{kovové} \\ \text{polovodičové} \end{array} \quad \frac{\Delta R}{R_0} = k_1 \cdot \varepsilon + k_2 \cdot \varepsilon^2 \quad (4)$$

Při měření pomocí odporových tenzometrů je nutné předejít rušivým vlivům, které výrazně ovlivňují vlastnosti materiálu použitého k jejich výrobě a tedy

i přesnost měření. Hlavním rušivým činitelem je teplota. S teplotou se kvůli délkové teplotní roztažnosti výrazně mění deformace a tedy i naměřená změna odporu. Vhodnou kompenzací vlivu teploty je použití můstkového zapojení několika tenzometrů (např. Wheatstoneův můstek). K dalším rušivým činitelům patří tečení (tzv. creep), vlhkost, elektromagnetické pole a další.

1.3.2 Měření rychlosti indukčním snímačem

Princip indukčního snímače vychází ze vzájemného působení vysokofrekvenčního elektromagnetického pole a elektricky vodivého materiálu. Aktivním prvkem indukčního snímače je cívka navinutá na feritovém (hrníčkovém) jádru, kterou protéká vysokofrekvenční střídavý proud generovaný oscilátorem, a vytváří tak v jejím okolí magnetické pole. Umístíme-li do tohoto pole elektricky vodivý materiál, jsou v něm indukovány vířivé proudy, které zpětně zeslabují magnetický tok cívky a snižují tak velikost oscilační amplitudy. Tato změna je detekována elektronikou snímače, vyhodnocena a převedena na příslušný výstupní signál.



Obr. 1.4: Funkční skupiny indukčního snímače

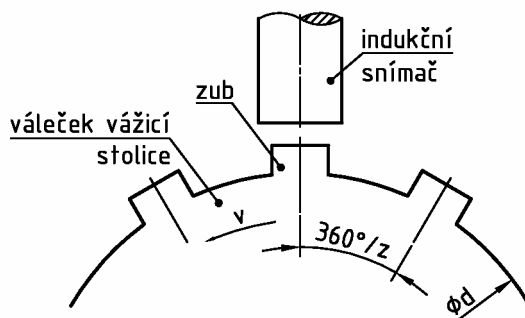
Mezi hlavní funkční charakteristiky indukčního snímače patří především velikost aktivní plochy, kterou vystupuje vysokofrekvenční pole do vzduchové mezery mezi snímačem a snímaným předmětem, dále dosah, což je vzdálenost, při které indukční snímač zaznamená přítomnost předmětu, a frekvence spínání, která udává maximální počet změn stavu snímače za sekundu. Dosah indukčního snímače je ovlivněn hlavně prostředím provozu a také materiálem, z něhož je snímaný

předmět vyroben. Pro úpravu hodnoty dosahu snímače v závislosti na materiálu snímaného objektu slouží korekční součinitel (např.: 1,0 pro ocel; 0,4 pro mosaz).



Obr. 1.5: Indukční snímač (zdroj [20])

Metoda měření rychlosti pásu pomocí indukčního snímače se řadí mezi elektrické impulsní snímání rychlosti. V případě tohoto řídicího systému je přímočará rychlost pásu dopravníku určena obvodovou rychlostí válečku vážicí stolice.



Obr. 1.6: Schéma měření obvodové rychlosti válečku

Váleček vážicí stolice o průměru d je rovnoměrně po svém obvodu osazen z zuby, které slouží jako měřicí clonky indukčního snímače. Při překrytí zubu s indukčním snímačem dojde k vyslání impulzu do vyhodnocovací jednotky, která čítá počet signálů n detekovaných za konstantní dobu Δt . Okamžitá rychlost v je obecně dána první derivací dráhy s podle času t (5).

$$v = \frac{ds}{dt} \quad (5)$$

Jelikož vstupní veličiny výpočtu jsou měřeny vždy za určitou dobu Δt , přechází vztah (5) na výpočet průměrné obvodové rychlosti dle rovnice (6).

$$v = \frac{s}{t} = \frac{n \cdot \pi \cdot d}{z \cdot \Delta t} \quad (6)$$

Za předpokladu, že skluz a pružnost pásu dopravníku je zanedbatelná, je možné považovat obvodovou rychlost válečku za rovnou přímočaré rychlosti pásu, a tedy vztah (6) je společný oběma rychlostem.

1.3.3 Kontinuální vážení na pásové váze

Pásová váha je tvořena vážicí stolicí, která je osazena snímačem síly a snímačem rychlosti pásu. Vážicí stolice se zařazuje do pásového dopravníku na místo zvolené s ohledem na nejvyšší přesnost měření silového zatížení, která ale závisí na mnoha faktorech

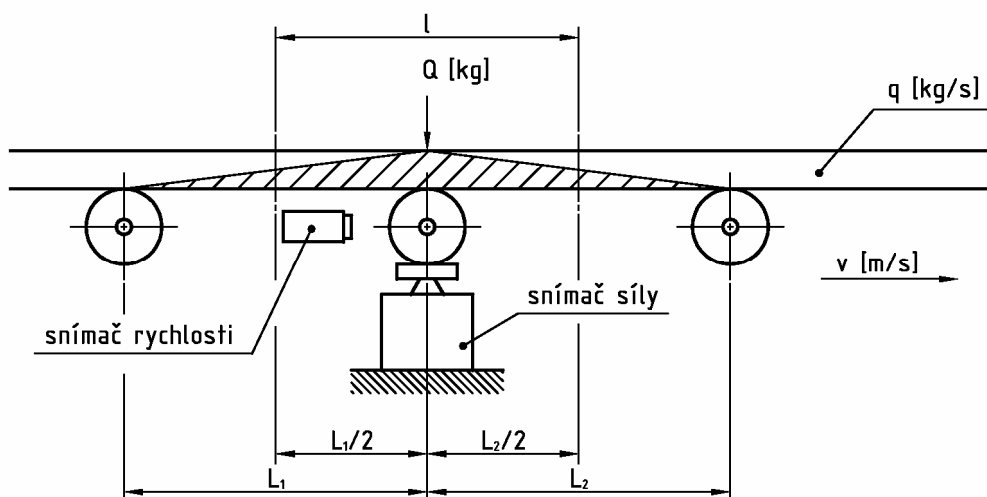
- technický stav dopravníku – špatný technický stav pásového dopravníku se zpravidla projevuje vibracemi, které se přenášejí i na vážicí úsek a zvyšují tak nejistotu měření zatížení snímače síly
- napnutí pásu – optimální napnutí pásu je důležité pro správné měření zatížení vážicí stolice
- úhel sklonu pásového dopravníku – větší úhly sklonu pásového dopravníku zapříčiňují zvýšené horizontální namáhání snímačů síly, které způsobuje nepřesnosti měření zatížení (v případě řízené technologie firmy DESTRO má pásový dopravník sklon přibližně 4° a tento sklon není kompenzován, jelikož chyba vážení při sklonu 4° je pouze 0,2%)
- velikost a kolísání rychlosti pásového dopravníku – jelikož je rychlost měřena v konečně malých časových intervalech, vysoká rychlost a její kolísání mezi okamžiky měření výrazně zvyšuje nejistotu měření
- rovnoměrnost rozložení materiálu na pásu dopravníku – je-li materiál nerovnoměrně rozložen na pásu, odchyluje se skutečnost od modelu

rovnoměrného hmotnostního toku a při vyhodnocování dochází k nárůstu nejistoty měření

- a jiné

Výběr umístění vážicí stolice vyžaduje tedy komplexní přístup a rozhodně i značnou míru zkušenosti.

Při odvozování vztahů pro výpočet veličin měřených na pásovém dopravníku je vhodné zavést zjednodušující předpoklady, díky nimž dojde ke snížení výpočtové náročnosti při nepatrném zhoršení přesnosti výsledků. V teoretickém rozboru vážení na pásovém dopravníku jsem vycházel z firemní dokumentace firmy TEVAS, která je výrobcem použité vážicí stolice, a jako oporu jsem použil učební text [7]. Teoretické závěry se proto přímo shodují s reálnou aplikací a implementací do výpočetních algoritmů vyhodnocovací jednotky.



Obr. 1.7: Schématické znázornění hmotnostního toku na pásovém dopravníku

Výchozí veličinou teoretického rozboru vážení na pásové váze je hmotnostní tok q , který obecně udává prošlou hmotnost látky za jednotku času.

$$q = \frac{dm}{dt} \quad (7)$$

Jelikož je ale hmotnost materiálu m na vážicí stolici pásového dopravníku dána zatížením snímače síly, které je snímáno přes napnutý pás dopravníku, je třeba ve vztahu (7) zohlednit i vliv rozložení materiálu mimo válečky vážicí stolice

(Obr. 1.7). Za předpokladu, že vážicí stolice je umístěna uprostřed mezi dvěma původními válečky pásového dopravníku, je možné teoretickou délkou l , na které je vážený materiál o hmotnosti m rovnoměrně rozložen (zatížení lineárně klesá se vzdáleností od válečku vážicí stolice), přibližně odhadnout jako vzdálenost mezi válečky vážicí stolice a válečky dopravníku.

$$l = \frac{L_1 + L_2}{2} = |L_1 = L_2 = L| = L \quad [m] \quad (8)$$

Na základě tohoto předpokladu lze upravit vztah (7) pro hmotnostní tok a zohlednit v něm zatížení rozložené na délce pásu l .

$$q = \frac{dm}{dl} \cdot \frac{dl}{dt} \quad (9)$$

První zlomek ve vztahu (9) je zmíněné zatížení vztažené na jednotku délky pásu a druhý zlomek je vztah pro výpočet okamžité rychlosti. Jelikož jsou měřené veličiny dány diskretními hodnotami a nikoliv spojitými, je možné vztah zjednodušit (10).

$$q = \frac{m}{l} \cdot v \quad [kg \cdot s^{-1}] \quad (10)$$

Ze známé hmotnosti materiálu m , dané zatížením snímače síly, a rychlosti pásu v je tedy možné určit hmotnostní tok q , který udává množství materiálu prošlého po pásu za 1 sekundu. Hmotnost materiálu m a rychlost pásu v jsou diskretní veličiny, jejichž hodnoty jsou odečítány vždy za určitou konstantní dobu vzorkování Δt , a tudíž vyčíslováním hmotnostního toku q vždy jednou za dobu Δt získáme posloupnost hodnot q_i .

Celková hmotnost prošlého materiálu Q za dobu τ je obecně dána, za předpokladu spojitých veličin, časovým integrálem hmotnostního toku q .

$$Q = \int_0^{\tau} q dt \quad [kg] \quad (11)$$

S přihlédnutím k zmíněné diskretní povaze měřených veličin ve vztahu (10) je třeba upravit vztah (11) a nahradit integrál sumou.

$$Q = \int_0^{\tau} q dt \rightarrow Q = \Delta t \cdot \sum_{i=1}^n q_i \quad n = \frac{\tau}{\Delta t} \quad (12)$$

Výsledný vztah určuje celkovou hmotnost materiálu Q , který byl přepraven pásovým dopravníkem za dobu τ . Doba τ je n -násobek periody vzorkování Δt .

Aby byla zajištěna správná funkce pásové váhy během provozu, je nutné splnit mnoho nutných předpokladů, mezi které patří

- správný převod signálu ze snímače síly
- pravidelná kalibrace pásové váhy
- časté tárování (kontrola nezatíženého stavu)
- kontrola mechanického seřízení a stavu vážicí stolice

2 Návrh a výběr řešení

Tato část práce je věnována návrhu variant řešení daného problému s přihlédnutím k známým faktům - možnostem rozšíření popsaného stávajícího řídicího systému, provozním podmínkám a požadavkům provozovatele, jejichž základní body jsou obsahem zadání této práce. Krátce je popsán princip jednotlivých variant, jejich výhody a nevýhody v kontextu požadavků, provozních podmínek a technických možností stávajícího systému.

2.1 Východiska návrhu

Samotnému procesu návrhu předchází důležitý krok, a to zvážení možností stávajícího systému. Na základě faktických parametrů lze navrhnout adekvátní varianty řešení.

2.1.1 Požadavky provozovatele

Hlavním východiskem návrhu variant jsou požadavky provozovatele popsané technologie a účel, který navrhované rozšíření stávajícího systému bude mít. Z těchto požadavků lze získat prvotní představu o navrhovaném systému ještě před zvážením technických možností a provozních podmínek. Mezi hlavní požadavky patří

- minimální provozní náklady
- snadný přístup k sebraným datům
- možnost vizualizace sebraných dat
- spolehlivost systému.

Účelem sběru dat z pásových vah je vytvoření dlouhodobé statistiky naváženého materiálu a také kontrola technologické kázně personálu, jenž zmíněnou technologii obsluhuje.

Zmíněné požadavky jsou natolik obecné, že je třeba jejich upřesnění na základě osobní konzultace s provozovatelem. Pod pojmem snadného přístupu je zamýšlena občasná kontrola (několikrát týdně) sebraných dat, tudíž není nutný kontinuální přenos dat. Postačí dávkové posílání souboru dat s určitým zpožděním.

Jako prostředek pro zobrazení přehledů je preferováno libovolné PC (Personal Computer), které je umístěno v sídle vedení firmy, vzdáleného od provozovny technologie přibližně 1,5 km. Zvolený prostředek vizualizace dat (PC) a jeho umístění je stěžejní informací pro návrh variant řešení. Důležitým upřesněním je také to, že provozovatel nepožaduje využití navrhovaného rozšíření stávajícího řídicího systému k řízení technologie, takže přenos dat bude pouze jednosměrný – od technologie k úložišti dat.

2.1.2 Provozní podmínky

Aby byla zajištěna dostatečná životnost zařízení, je třeba zvážit i provozní podmínky, v nichž bude zařízení pracovat.

Základní činnosti technologického postupu zpracování strusky a stavebního odpadu jsou drcení a třídění. Provádění obou těchto činností je spojeno se vznikem značných vibrací, které jsou přenášeny do celé konstrukce linky, a produkcí velkého množství jemného prachu, který neblaze působí na jakékoliv mechanické prvky. V celém areálu provozu se pohybuje těžká technika, která nepřetržitě přemísťuje materiál buď v rámci areálu, nebo k zákazníkům, takže zmíněný prach je neustále vířen. Pásové dopravníky nejsou pevně spojeny s technologií a dle potřeby jsou přemísťovány v rámci areálu.

Údržba zařízení se neprovádí pravidelně a personál dozírající na provoz technologie není stálý. Často se obměňuje, a tudíž nelze uvažovat, že by přenos dat měl být pravidelně a spolehlivě inicializován a řízen lidským faktorem.

2.1.3 Technické možnosti stávajícího systému

Z uvedených požadavků lze vyvodit velké množství návrhů variant řešení, které se více či méně budou principiálně lišit. Výčet všech variant by byl velice dlouhý. Požadavky nejsou ale jediným východiskem návrhu. Není možné realizovat ty varianty, jejichž požadavky na technické vybavení stávající systém nespĺňuje. Proto je potřeba zohlednit i technické možnosti stávajícího systému, čímž se sníží i počet možných variant řešení.

Areál provozu se nachází na těžko přístupném místě, jež není spojeno s žádnou vnější metalickou komunikační linkou. S ohledem na provozní podmínky by položení linky přinášelo velkou investici, protože by bylo nutné zajistit dostatečné zabezpečení proti poškození pohybující se těžkou technikou. Linka by dále musela dovolovat změnu umístění při přesunu pásových dopravníků.

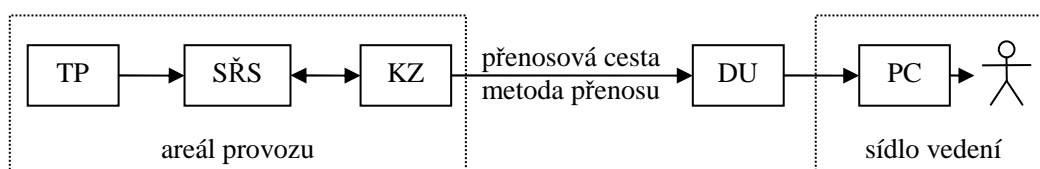
Jak již bylo uvedeno, vyhodnocovací jednotkou stávajícího systému je kompaktní řídicí terminál T3A od firmy HYPEL. Jeho základní charakteristika je uvedena v kapitole 1.2.2. Terminál disponuje pouze jedním komunikačním rozhraním, a to sériovou sběrnici RS-485, kterou lze využít pro nahrávání uživatelských programů, jejich ladění, konfiguraci terminálu pomocí PC nebo pro spojení v síti s dalšími automatizačními prvky firmy HYPEL, mezi něž patří také zařízení zprostředkovávající i jinou komunikaci než jen po sériové sběrnici RS-485.

Pro přímou komunikaci mezi PC a T3A se využívá buď servisní převodník CA1 nebo převodník USB485. Liší se v připojení k PC a vlastním použitím. Převodník CA1 se připojuje k sériovému portu PC a je určen pouze pro krátkodobé servisní zásahy, jelikož nemá galvanické oddělení a není vybaven náležitými ochrannými obvody. Naproti tomu převodník USB485 se připojuje k USB portu PC a disponuje galvanickým oddělením, proto ho lze použít pro dlouhodobé připojení zařízení bez nebezpečí poškození PC. Oba zmíněné převodníky jsou určeny pouze k programové obsluze zařízení firmy HYPEL a nikoliv k sledování systému, sběru dat a podobným účelům.

Komunikace v síti zařízení firmy HYPEL probíhá síťovým protokolem PESNET 3.32, kvůli němuž není možné do sítě zařízení firmy HYPEL zapojit zařízení jiných firem, aniž by byla zachována plná kompatibilita síťové komunikace. Firma HYPEL ale dodává i zařízení, které je schopno zprostředkovat komunikaci s jinými zařízeními. Tímto zařízením je programovatelný převodník RoutPro FX, díky němuž je možné komunikovat s automatizačními prostředky vybavenými sériovou linkou RS-232C.

2.2 Návrh variant řešení

Nyní je již možné přistoupit k samotnému návrhu variant řešení spolu s uvedením jejich výhod a nevýhod. Podstatnou informací v návrzích variant je princip jejich činnosti spolu s uvedením obecných prvků, užitých pro realizaci přenosu dat mezi stávajícím řídicím systémem a PC, které se nachází v sídle vedení společnosti. Samotný výběr konkrétního zařízení je vhodné ponechat až k řešení zvoleného návrhu, jelikož uvedení všech použitelných zařízení by bylo nesmírně obsáhlé a náročné, přestože to není hlavní náplní principiálního návrhu řešení.



Obr. 2.1: Obecný návrh varianty řešení – komunikační řetězec

Dle uvedených východisek z kapitoly 2.1 lze určit obecné schéma návrhu řešení (Obr. 2.1). Skládá se z technologického procesu (TP), stávajícího řídicího systému (SŘS), komunikačních zařízení (KZ), definice přenosové cesty spolu s metodou přenosu dat, datového úložiště (DU) a nakonec osobního počítače (PC), na kterém bude prováděna vizualizace sebraných dat. Technologický proces spolu se stávajícím řídicím systémem jsou neměnné části navrhovaného komunikačního řetězce a PC jako prostředek vizualizace byl nepřímým zadáním provozovatelem. K návrhu tedy zbývá pouze výběr komunikačního zařízení, přenosové cesty s metodou přenosu dat a nakonec úložiště sebraných dat.

Výběr přenosové cesty je hlavním a určujícím aspektem v návrzích variant řešení. S volbou přenosové cesty úzce souvisí volba komunikačních zařízení, a proto jsem je v návrhu řešení rozdělil dle přenosových cest – optická, metalická, rádiová. Docílil jsem tím vytvoření přehledného rozdělení s hierarchickou strukturou, jež umožňuje vyřadit již při návrhu komunikační zařízení využívající nevhodnou přenosovou cestu.

Datové úložiště je možné realizovat mnoha způsoby. Výběr vhodného datového úložiště je vázán na volbu přenosové cesty a konkrétní metody přenosu, která je dána možnostmi komunikačního zařízení. Jelikož je volba datového úložiště komplexní činností, ponechal jsem ji až na závěr výběru konkrétní varianty řešení, kdy již budou známy téměř všechny potřebné informace.

2.2.1 Optická přenosová cesta

Použití optické přenosové cesty je v případě této technologie naprosto zbytečné a přinášelo by závratně vysoké náklady, přestože by možnosti této komunikace nikdy nebyly plně využity. Varianty řešení s touto přenosovou cestou je tedy možné hned zpočátku zavrhnout a dále se jimi nezabývat.

2.2.2 Metalická přenosová cesta

Jak bylo zmíněno v kapitole 2.1.3, areál provozu není připojen k žádné metalické komunikační síti a její vybudování by bylo příliš nákladné, tudíž bude řešení s metalickou přenosovou cestou omezeno na návrhy, jejichž komunikační řetězec bude končit přímo v areálu provozu. To však nesplňuje požadavky provozovatele, a proto budou tyto návrhy popsány stručně bez podrobnějších informací.

Jelikož jedním z požadavků na rozšíření stávajícího systému je možnost vizualizace dat, bude třeba, aby navrhované varianty řešení byly vybaveny dostatečně velkým zobrazovacím zařízením. Dále jsou známy omezené komunikační možnosti stávajícího zařízení T3A, které může komunikovat pomocí sériové sběrnice RS-485 buď přímo s PC (pouze servisní zásahy), nebo automatizačními prostředky firmy HYPEL. Firma HYPEL ale nenabízí vlastní vizualizační prostředky, proto je nutné volit zobrazovací zařízení jiných firem. Tato zařízení je třeba kvůli kompatibilitě vzájemné komunikace připojit ke stávajícímu systému přes sériovou linku RS-232C pomocí programovatelného převodníku RoutPro FX, který je vybaven příslušným obslužným programem. RoutPro FX je tedy komunikačním zařízením (KZ), které bude společné pro všechny navrhované varianty s metalickou přenosovou cestou.

PC/iPC – Sběr a vizualizace dat s využitím PC/iPC, které by bylo připojené k stávajícímu systému přes sériovou linku RS-232C prostřednictvím programovatelného převodníku RoutPro FX, má mnoho možností. Spíše než PC je ale vhodnější použít iPC (Industrial PC), které je uzpůsobeno k provozu v náročných průmyslových podmínkách. PC/iPC v tomto návrhu má význam datového úložiště (DU) a také koncového PC pro vizualizaci v komunikačním řetězci (Obr. 2.1).

Jako vhodné, nikoliv však jediné možné, konkrétní použití PC/iPC je v kombinaci se SCADA systémem (například český software PROMOTIC pro OS Windows®), který je určen pro monitorování a ovládání procesů v nejrůznějších oblastech průmyslu. Firma HYPEL dodává k zařízení RoutPro FX obslužný program a programové vybavení pro OS (Operating System) Windows® založené na technologii DDE (Dynamic Data Exchange), což je metoda sdílení dat mezi aplikacemi běžícími pod operačním systémem Windows® či OS/2. Pomocí této technologie je možné vyčítat a zapisovat veškerá aktuální data - proměnné ze sítě HYPEL a dále je zpracovávat.

Tímto uspořádáním lze snadno naplnit většinu hlavních požadavků na sběr a zobrazení dat ze stávajícího řídicího systému. SCADA systémy jsou uzpůsobeny k jednoduchému vytváření uživatelských aplikací, nabízejí široké možnosti zpracování dat a způsobů jejich zobrazení. Provozní náklady se v tomto návrhu neuplatní, což je velice výhodné. Při přednostním použití iPC lze předpokládat i určitou odolnost v provozu se silnými vibracemi a velkým množstvím jemného prachu.

Přes všechny nesporné výhody je PC/iPC v panelovém či boxovém provedení se všemi svými periferiemi drahé a choulostivé zařízení, které se snadno poškodí - zvláště v takovém provozu, jako je drticí a třídící linka kamene. Aby byl splněn požadavek provozovatele, dle něhož by mělo být možné zobrazit sebraná data na PC v sídle vedení, bylo by potřeba data pravidelně přenášet do uvedeného PC na některém současně používaném datovém nosiči (USB flash disk, CD, DVD, paměťové karty a jiné). To vzhledem k nízké spolehlivosti obslužného personálu nelze předpokládat.

2.2.3 Rádiová přenosová cesta

Jelikož optická přenosová cesta je z hlediska výše nákladů nevhodná a metalická nespĺňuje všechny požadavky, rádiová přenosová cesta je nejvhodnějším řešením. Existuje mnoho způsobů bezdrátové komunikace, avšak jen několik z nich vyhovuje užití za podmínek této technologie.

Podobně, jako tomu bylo v případě metalické přenosové cesty, nenabízí firma HYPEL automatizační prostředky pro bezdrátovou komunikaci. Proto je opět nutné využít společně pro všechny varianty řešení programovatelného převodníku RoutPro FX pro zprostředkování komunikace mezi jednotkou T3A a zařízením, které bude zajišťovat vlastní bezdrátovou komunikaci. V tomto případě ale nebude RoutPro FX jediným komunikačním zařízením (KZ) v komunikačním řetězci, bude jednou z jeho částí spolu se zařízením realizujícím vlastní bezdrátovou komunikaci, které musí mít možnost řízení za pomoci sériové linky RS-232C.

GSM modem – Průmyslové GSM modemy jsou kompaktní a odolná zařízení, která nabízejí široké možnosti bezdrátové komunikace v síti GSM (Global System for Mobile Communications) s využitím placených služeb mobilních operátorů. Jsou využívány k přenosu dat v mnoha průmyslových aplikacích a to například v kombinaci s moduly GPS (např.: systémy monitorování pohybu nákladních vozidel). Kromě základních služeb pro přenos hlasu, SMS a faxů jsou dnes většinou dodatečně vybaveny dalšími možnostmi komunikace – FTP, SMTP a další.

V rámci sítě GSM jsou vhodné pro průmyslové využití dva standardy a to GPRS (General Packet Radio Service) a CSD (Circuit Switched Data), které se liší v samotném pojetí datové komunikace. V případě standardu CSD je datový přenos založen na přepojování okruhů a je podobný přenosu hovoru. Mezi příjemcem a odesílatelem je založena souvislá přenosová cesta s vyhrazenou přenosovou kapacitou a tarifkace je vztažena k trvání spojení bez ohledu na to, zda jsou či nejsou data přenášena. CSD je vhodný pro krátkodobé občasné přenosy. Přenos dat metodou CSD má stejnou prioritu jako hovor. Naproti tomu standard GPRS je založen na přepojování paketů. Poplatky nejsou vztaženy k délce přenosu, ale k objemu přenesených dat. Přenos dat pomocí GPRS má nižší prioritu než hovor, což

znamená, že při příchozím hovoru je přenos dat odsunut na pozadí a vyčkává na ukončení hovoru. Metoda přenosu dat pomocí GPRS je vhodná pro dlouhodobé častější přenosy.

GSM modemy jsou malá komunikační zařízení s velkými možnostmi, jejichž funkce se neustále rozšiřují. Nabízejí široké možnosti komunikace přes rozšířenou, snadno dostupnou a relativně spolehlivou bezdrátovou komunikační síť.

Nevýhodou datové komunikace s využitím GSM modemu je nutnost platit za využívání zmíněné komunikační sítě. Celkové náklady na provoz datového přenosu jsou v porovnání s výhodami GSM sítě zanedbatelné.

Převodník sériové komunikace na Wi-Fi – Wi-Fi je obchodní značka certifikovaných produktů založených na standardu IEEE 802.11. Dnes je označení Wi-Fi častěji používané jako synonymum pro označení bezdrátové LAN (Local Area Network) – tzv. WLAN (Wireless LAN). Wi-Fi je rozšířená bezdrátová technologie komunikující v bezlicenčním pásmu s frekvencí 2,4 GHz a založená na volně dostupném standardu, díky němuž jsou zařízení využívající tuto technologii velice rozšířená a levná. V případě použití zařízení RoutPro FX by bylo možné využít převodníku sériové komunikace na Wi-Fi. Komunikace se sériovou linkou je přenášena bezdrátovou sítí k PC vybavenému Wi-Fi technologií a připojenému do stejné WLAN. Na PC je nainstalován obslužný software, který vytvoří v OS virtuální sériový port. Ten je přístupný všem aplikacím, které jsou spuštěny pod tímto OS, a je tedy možné k němu přistupovat stejným způsobem, jako kdyby bylo vzdálené zařízení přímo připojeno k PC. Nakonec stačí vybavit toto PC vhodným vizualizačním software a stane se jak datovým úložištěm (DU), tak vizualizačním prostředkem.

Bezdrátová technologie Wi-Fi je velice rozšířená. Zařízení pro průmyslovou automatizaci využívající tuto technologii jsou relativně levná stejně jako provoz této bezdrátové komunikace.

Velkým problémem je dosah signálu této sítě. Jelikož v okolí neexistuje žádná WLAN, bylo by třeba vytvořit vlastní síť. Jak již bylo uvedeno, leží technologie na členitém terénu tvořeném hromadami sutí a ve značné vzdálenosti (přibližně 1,5 km)

od PC určeném k vizualizaci. Z tohoto důvodu by bylo třeba pro zajištění dostatečné síly signálu použít směrovou anténu, jež směřuje signál do úzkého prostoru ve zvoleném směru. Jelikož by anténa měla být v blízkosti stávajícího systému, který je součástí pásového dopravníku určeného k občasnému přemístění, bylo by třeba po každém přemístění anténu znovu nasměřovat a také by anténa byla silně ohrožena poškozením. Síla signálu a tedy i dosah sítě je také ovlivněn počasím a překážkami mezi oběma místy, což by v tomto případě bylo ještě znásobeno velkou vzdáleností.

2.3 Výběr varianty řešení

Zvážením výhod a nevýhod variant řešení, jež byly uvedeny v kontextu výchozích podmínek, lze určit tu nejvhodnější. Je zřejmé, že využití optické či metalické přenosové cesty není vhodné, jelikož tyto varianty nesplňují všechny uvedené požadavky a jsou finančně značně náročné, tudíž zbývají varianty založené na bezdrátové komunikaci – rádiové přenosové cestě.

Pro obě varianty bezdrátové komunikace je společné použití programovatelného převodníku RoutPro FX, který bude zprostředkovávat komunikaci mezi stávajícím řídicím systémem a vlastním komunikačním zařízením.

Nyní je třeba vybrat mezi standardem GSM a Wi-Fi. Ve prospěch Wi-Fi hovoří nižší provozní náklady při využití vlastní sítě. Dosah signálu na tak velkou vzdálenost není zaručen a slabý signál, rušený například provozem linky, by mohl způsobit značnou nespolehlivost navrhovaného přenosového systému. Oproti tomu standard GSM nabízí mnohem stabilnější síť, z čehož ale plynou i vyšší provozní náklady. Průmyslové GSM modemy jsou dnes široce rozšířeny a vybaveny velkým počtem komunikačních protokolů, které umožňují rozmanitá řešení přenosu dat.

Výběr mezi zmíněnými dvěma variantami byl obtížný, jelikož obě mají svá významná pozitiva. Na základě konzultace s technicky vzdělanými osobami (z firmy HYPEL a TEVAS), které pracují v oboru automatizace delší dobu a mají bohaté zkušenosti, jsem určil jako dodatečnou určující vlastnost spolehlivost navrhovaného přenosového systému. Přenos sebraných dat bude tedy realizován pomocí programovatelného převodníku RoutPro FX a GSM modemu.

Při sběru dat není vyžadován přehled okamžitých hodnot. Sebraná data budou po skupinách jednorázově posílána v konstantních časových intervalech. Pro takovýto přenos je vhodnější datová služba GPRS. Standardní komunikační služby, jako například SMS či fax, nejsou kvůli vysoké ceně a malé datové kapacitě příliš vhodné. Výběr konkrétního GSM modemu bude tedy omezen podmínkou implementovaného TCP/IP stacku*, díky němuž lze využít protokoly aplikační vrstvy – například FTP, SMTP, POP, HTTP. Tyto protokoly umožní efektivnější datový přenos při nízké ceně.

Posledním článkem komunikačního řetězce, jenž doposud nebyl určen, je datové úložiště. Jeho konkrétní výběr je vázán na zvolené komunikační zařízení a metodu přenosu. Pro přenos jsou preferovány protokoly aplikační vrstvy modelu OSI. Vhodnou kombinací datového úložiště a protokolu lze vytvořit pevný základ pro zpracování a vizualizaci dat. Firma DESTRO má pronajatý webový server, na němž je umístěna internetová prezentace firmy. Využitím tohoto serveru lze ušetřit náklady na datové úložiště a umožnit přístup k datům z jakéhokoliv PC připojeného k internetu. Pro přenos dat na tyto servery je standardně využíván FTP, který je též podporován TCP/IP stackem implementovaným do GSM modemů.

Vybrané řešení reprezentované komunikačním řetězcem (Obr. 2.1) bude mít následující konfiguraci

- KZ – **RoutPro FX + GSM modem** (GPRS, TCP/IP stack)
- přenosová cesta – **rádiová**
- metoda přenosu – **FTP**
- DU – **webový server**.

* Pojem stack je označení, jež v českém jazyce nemá příhodný ekvivalent. Stack označuje implementaci určité skupiny protokolů, které pokrývají všechny vrstvy referenčního modelu ISO/OSI.

3 Výběr a popis zařízení

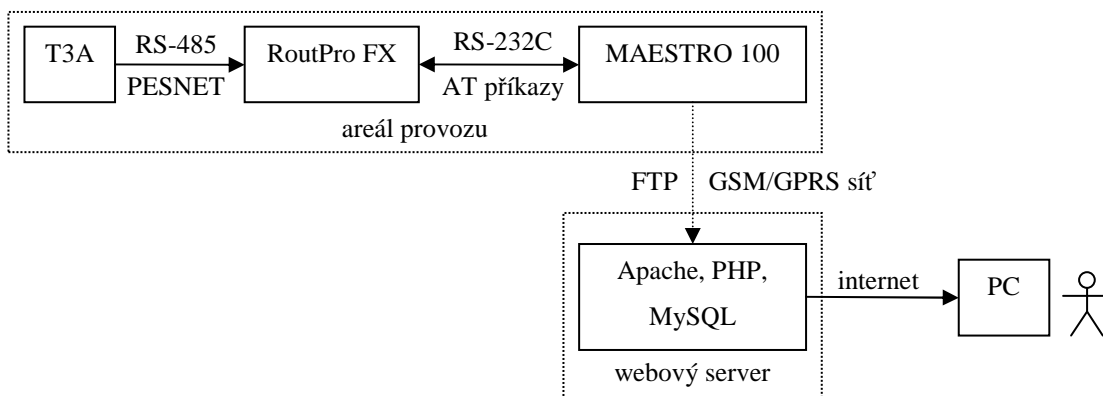
3.1 Výběr zařízení

Pro zvolenou variantu bylo třeba vybrat konkrétní zařízení, s nimiž jsem následně realizoval zadanou úlohu sběru dat. Pořízení a zapůjčení všech potřebných zařízení přislíbila firma TEVAS.

Prvním konkrétním zařízením v komunikačním řetězci byl programovatelný převodník RoutPro FX firmy HYPEL. Použití tohoto zařízení nebylo podmíněno výběrem, ale nutností zprostředkovat komunikaci mezi stávajícím řídicím systémem a GSM modemem při zachování maximální kompatibility komunikace v řetězci.

Druhá volba se týkala GSM modemu. Pro modem bylo v návrhu stanoveno, že musí mít implementovaný TCP/IP stack, jenž podporuje FTP. Další podmínkou byla odolnost v těžkých provozních podmínkách – otřesy, prach, nízké teploty. Jelikož český trh nabízí velké množství GSM modemů, které splňují tato obecná kritéria, byl by výběr konkrétního zařízení obtížný a obsahově velice rozsáhlý. Proto jsem na základě konzultace s Ing. Jaroslavem Buškem z firmy TEVAS zvolil GSM modem MAESTRO 100, s jehož použitím má tato firma zkušenosti. Tento GSM modem je založený na osvědčeném modulu firmy WAVECOM, která se vývojem telekomunikačních zařízení dlouhodobě zabývá, a splňuje všechny stanovené požadavky. Volbu operátora mobilní sítě jsem ponechal na firmě DESTRO, které jsem doporučil zaměřit se na nejnižší ceny datových služeb. Obdržel jsem SIM kartu mobilního operátora Vodafone, jehož služby využívá celá firma DESTRO.

Poslední navrhovanou částí komunikačního řetězce je webový server. Ten jsem dle návrhu zvolil totožný s webovým serverem, na kterém je umístěna internetová prezentace firmy DESTRO. Webový server je založen na OS Linux, HTTP požadavky obsluhuje Apache HTTP Server 2.0. Server podporuje skriptovací jazyk PHP 4.4.2 a pro uložení a zpracování dat nabízí databázový systém MySQL 5.0.51a.



Obr. 3.1: Konkrétní podoba komunikačního řetězce

3.2 Programovatelný převodník RoutPro FX

3.2.1 Popis převodníku RoutPro FX

Programovatelný převodník RoutPro FX je zařízení, které slouží k zprostředkování komunikace mezi sítí automatizačních prostředků firmy HYPEL (sériová sběrnice RS-485, protokol sítě PESNET 3.32) a zařízením komunikujícím libovolným znakovým protokolem po sériové lince RS-232C. Přebodník může být volitelně vybaven obvyklými prvky, které jsou charakteristické pro PLC (obvod reálného času, vnitřní zálohovaná paměť aj.). Nedisponuje však žádnými digitálními či analogovými vstupy. Místo nich je vybaven dvěma konektory DE-9M, z nichž jeden slouží pro připojení linky RS-232C a druhý k připojení sériové sběrnice RS-485.

Tab. 3.1: Základní technické parametry zařízení RoutPro FX

Napájecí napětí	12 až 30V
Odběr ze zdroje	50 až 110mA
Izolační napětí RS-485	1500VAC
Rozsah pracovních teplot	-5°C ~75°C
Rozměry	65 x 125 x 52 mm
Hmotnost	165 g

Princip programování převodníku je shodný se všemi automatizačními prostředky firmy HYPEL. Zdrojový text uživatelského programu se zapisuje v programovacím jazyce SIMPLE3. K rozhraní RS-232C se v programu přistupuje pomocí proměnných, které u ostatních PLC firmy HYPEL reprezentují analogové vstupy a výstupy.



Obr. 3.2: Programovatelný převodník RoutPro FX (zdroj [18])

3.2.2 Popis jazyka SIMPLE3

Programovací jazyk SIMPLE3 je specifický pro výrobky firmy HYPEL. Pomocí něj je možné vyjádřit řídicí algoritmy programovatelných automatů, terminálů a kompaktních řídicích stanic (dále jen programovatelné jednotky), vyráběných touto firmou.

Programové vybavení uvedených programovatelných jednotek je rozděleno do dvou stěžejních částí. První část, systémový firmware, je základním programovým vybavením, které slouží k zprostředkování vazeb mezi uživatelským programem a komunikačními obvody jednotky. Systémový firmware je nahrán výrobcem. Pro zajištění správného chodu řídicí jednotky není zásah do systémového firmware při vytváření vlastních aplikací možný.

Druhou částí je zmiňovaný uživatelský program, který zachycuje vlastní řídicí algoritmus. Uživatelský program přistupuje k fyzickým vstupním či výstupním rozhraním programovatelné jednotky zprostředkovaně přes předdefinované

proměnné, které jsou v paměti RAM umístěny na pevně stanoveném místě. Tyto proměnné se vždy aktualizují na začátku vykonávané programové smyčky a zůstávají tedy po celou dobu vykonávání programových instrukcí probíhající smyčky nezměněny. Pro snadný přístup jsou těmto proměnným přiřazeny specifické identifikátory, jejichž použití v uživatelském programu nevyžaduje předchozí deklaraci.

Vykonávání uživatelského programu je sekvenční a není tedy možné použít vlastního programového skoku. Výjimku tvoří implementace podmíněných skoků v konstrukcích IF a CASE. Tímto opatřením se předchází nebezpečné situaci, kdy se program zacyklí, nebo dojde k jeho zablokování.

Syntaxe, neboli formální zápis, programovacího jazyka SIMPLE3 je velice podobná procedurálnímu programovacímu jazyku PASCAL. Na začátku programu se zapisují řídicí direktivy pro překladač. Dále jsou uvedeny definice konstant a deklarace proměnných a za nimi následuje výkonná část programu zakončená klíčovým slovem END. Bližší specifikace programovacího jazyka SIMPLE3 je uvedena ve firemní dokumentaci [13]. Pro použití při realizaci návrhu řešení zadané úlohy je vhodné zmínit, že jazyk SIMPLE3 rozlišuje 4 základní datové typy (Tab. 3.2).

Tab. 3.2: Základní datové typy jazyka SIMPLE3

Označení	Rozsah	Velikost v paměti
BIT	0 až 1	1 bit
WORD	0 až 65 535	2 bajty
INT	-32 768 až 32 767	2 bajty
REAL	$-3,4 \times 10^{38}$ až $3,4 \times 10^{38}$	4 bajty

Vytvoření zdrojového textu uživatelského programu není na rozdíl od překladu a nahrání omezeno použitím vývojového prostředí HypEd4 a lze ho vytvořit v libovolném textovém editoru. Pro zápis zdrojového textu v jazyce SIMPLE3 jsou určeny textové soubory s příponami *.STP (zdrojový soubor) a *.LIB (knihovna). Při překladu zdrojového textu v prostředí HypEd4 je zkontrolována syntaxe a pokud nebyly nalezeny syntaktické chyby, je vytvořen binární soubor

s příponou *.DNL. Tento soubor je již určen k přímému nahrání do programovatelné jednotky, což lze provést opět v prostředí HypEd4 spuštěním programu LOADER3.

3.2.3 Program HypEd4

Program HypEd4 je integrované vývojové prostředí určené pro vývoj uživatelských programů pro programovatelné logické automaty firmy HYPEL. Součástí prostředí je sada podpůrných nástrojů usnadňujících vytváření a ladění uživatelských aplikací. Prostor programu umožňuje volbu mezi dvěma základními režimy - režim editace zdrojového textu a režim simulace programovatelných logických automatů firmy HYPEL.

```

#Configuration "ROUTPRO FX"
var
  I0 # RxBuffer : array[31] of word;
  O0 # TxBuffer : array[15] of word;
  O15 # TxPointer : word; //nulovy pri ODESLANI vseh znaku
  O17 # RxPointer : word;
  O19 # BaudRate : word;
  Y9 # StopBits : bit;
  Y10 # Parity : bit;
  Y11 # ParityType : bit;
  Y13 # ParityError : bit;
  Y7 # GsmMode : bit;
  Y0 # RTS : bit;
  X0 # CTS : bit;
end
17
18 symbol
19 1 # Bd2400;
20 2 # Bd4800;
21 3 # Bd9600;
22 4 # Bd19200;
23 5 # Bd38400;
24 6 # Bd57600;
25 7 # Bd115200;
26 end
27
28 //nastaveni vlastnosti prenosu RS232C
29 BaudRate = Bd9600;
30
31 if RESET then
32   T0 = 0; //nulovani casovace
33   TEN0; //povoleni citani casovace 0 - log. 1 -> casovac cita
34   TPA0; //zarazeni preddelice 1/100 - log. 1 -> cita po 1 s
35
36 //definice vystupni fronty
37 O2 = 'A';
38 O1 = 'T';
39 FORMAT = 220;
40 O0 = 'I'; //CR
41
42 RESET';
43 endif
44 if T0 >= 5 then
45   T0 = 0; //nulovani casovace
46   TxPointer = 3; //nastaveni ukazatele na vystupni frontu
47   RxPointer = 0;
48 endif
49
50 end
  
```

Obr. 3.3: Hlavní okno programu HypEd4

V režimu editace zdrojového textu je přístupné okno úpravy, ve kterém je zobrazen syntakticky zvýrazněný zdrojový text. Po dokončení úprav zdrojového textu se spuštěním překladače provede kontrola syntaxe a přeloží se zapsaný uživatelský program do binárního souboru *.DNL. Vytvořený binární soubor se pomocí nástroje LOADER3 nahraje do programové paměti programovatelné jednotky a spustí se jeho vykonávání. Nahrání souboru se provádí pomocí CA1

servisního převodníku firmy HYPEL (Obr. 3.4), který převádí úrovně sériové sběrnice RS-232C ze sériového portu osobního počítače na úrovně sběrnice RS-485, nebo za použití převodníku USB485 také od firmy HYPEL, který umožňuje nahrání programu prostřednictvím USB rozhraní standardně dostupného u všech dnešních PC. Aktuální hodnoty všech programových a systémových proměnných lze sledovat a měnit pro účely ladění v dialogovém okně po spuštění nástroje VIEWER3.



Obr. 3.4: Servisní převodník CA1 (zdroj [18])

Režim simulace slouží k ladění uživatelských programů, aniž by bylo nutné pracovat s fyzickým automatem. Po spuštění simulace programu je editace zdrojového textu zakázána a v textu jsou zvýrazňovány právě prováděné kroky. Lze tak snadno odstranit závažné chyby v programu a předejít nebezpečí poškození řízeného systému ještě před samotným nasazením řídicí jednotky.

3.3 GSM modem MAESTRO 100

3.3.1 Základní informace o GSM/GPRS

GSM je otevřený standard pro funkční jednotky mobilního rádiového systému. Komunikace v celulárním (buňkovém) radiotelefonním systému GSM je digitální a patří tedy mezi systémy druhé generace, označované 2G. Na rozdíl od analogových systémů nabízí kvalitnější spojení, silnější šifrování a efektivněji využívá přidělená kmitočtová pásma. Díky digitálnímu přenosu informace je také možné zajistit kompatibilitu i s jinými digitálními systémy a případně implementovat nové technologie, jako je například GPRS.

Základním zařízením pro využívání služeb GSM systému je mobilní stanice, jejíž důležitou součástí je modul identity účastníka SIM, jenž obsahuje důležité individuální účastnické informace nutné pro využívání služeb GSM systému – mají zabezpečující charakter. Podrobnější popis struktury a organizace GSM systému je velice rozsáhlý a není možné se s ním krátce seznámit – podrobné informace lze nalézt např. v [1].

GPRS patří mezi vyšší generaci mobilních rádiových sítí, označovaných 2.5G. Jedná se o datovou službu založenou na přepojování paketů. Síť GPRS existuje v rámci GSM sítě. GPRS zahrnuje podporu protokolů IP, PPP, OSPIH a X.25, z nichž v praxi je využíván protokol IP a v některých sítích i PPP. V rámci GPRS jsou rozeznávány 3 třídy zařízení rozlišené dle možnosti simultánního přenosu dat a hovoru.

- Class A - simultánní využití datových i hlasových služeb
- Class B - datový nebo hlasový přenos zvlášť s možností držení spojení a přepínání (většina dnešních zařízení)
- Class C - pouze datový nebo hlasový přenos zvlášť, nutné manuálně přepínat

GPRS v rámci GSM sítě vytvoří podmínky pro vyšší přenosové rychlosti, než jsou používány v síti GSM. Provádí tak pomocí 4 různých kódových schémat a obsazením více kanálů najednou (až 8 kanálů). Tímto postupem je možné teoreticky dosáhnout přenosové rychlosti až $171,1 \text{ kbit.s}^{-1}$, v reálném provozu je však dosahováno přenosové rychlosti 40 až 60 kbit.s^{-1} .

3.3.2 Popis GSM modemu MAESTRO 100

GSM modem MAESTRO 100 je kompaktní komunikační zařízení, které slouží pro bezdrátový přenos dat v síti GSM. Podporuje dualband GSM 900/1800 MHz, GPRS Class B Multislot Class 10. Modem je napájen napětím širokého rozsahu (Tab. 3.3) a je uzpůsoben i k provozu s napájením z elektrických článků (režim spánku při dlouhodobé nečinnosti ze strany linky RS-232C).

Tab. 3.3: Základní technické parametry GSM/GPRS modemu MAESTRO 100

Vstupní napětí	5V až 32 V DC
Maximální odběr při 5V	650mA
Rozměry	88 mm x 60 mm x 26 mm
Hmotnost	100 g
Teploty	-15°C ~ +50°C

Je založen na osvědčeném modulu Q24PL001 firmy WAVECOM, který je umístěn v odolném kovovém pouzdře (možnost montáže na DIN lištu) s vyvedeným konektorem linky RS-232C, napájecím konektorem, držákem pro SIM kartu a SMA anténním konektorem.



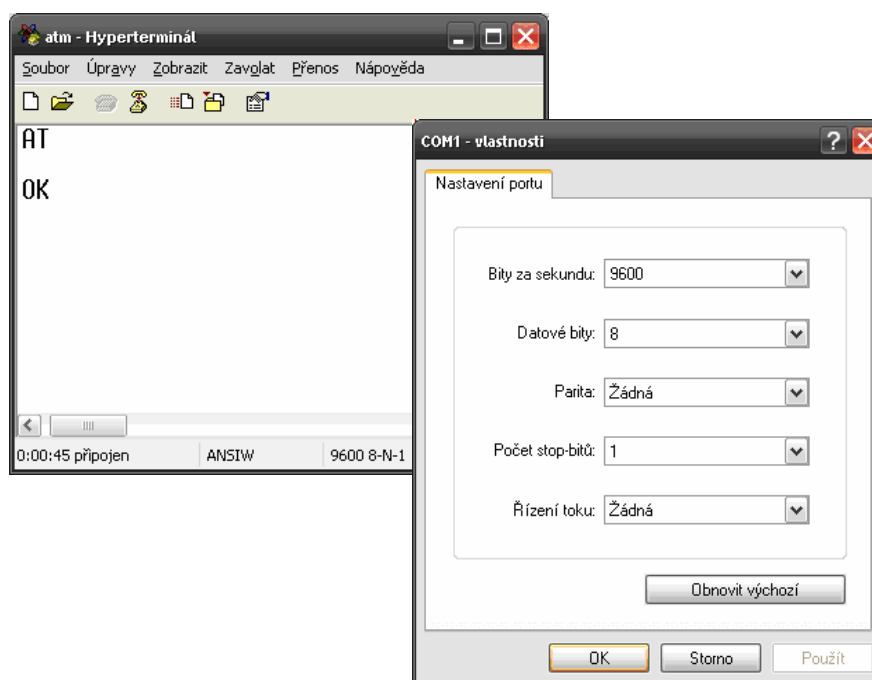
Obr. 3.5: GSM/GPRS modem Maestro 100 (zdroj [19])

Kromě základních služeb pro přenos hlasu, SMS, faxů umožňuje datovou komunikaci protokoly FTP, SMTP, HTTP nebo POP3 s využitím TCP/IP stacku, který je implementován v operačním systému modulu firmy WAVECOM. Dále jsou uživatelům v případě potřeby zpřístupněny možnosti modulu za použití softwaru Open AT[®] firmy WAVECOM, ve kterém lze vytvářet vlastní uživatelské aplikace a definovat svou vlastní rozšiřující sadu AT příkazů.

Možnosti modulu firmy WAVECOM jsou natolik rozsáhlé, že by jejich výčet zastínil obsah této práce, aniž by byly při řešení využity. Podrobnější informace lze nalézt např. v [14] a [15].

3.3.3 Konfigurace a ovládání GSM modemu

Komunikace mezi modemem a připojeným zařízením probíhá přes sériovou linku RS-232C pomocí definované posloupnosti znaků, tzv. AT příkazů. Jedná se o soubor příkazů poprvé použitý firmou Hayes, jenž slouží k nastavení modemu, navázání, udržování, ukončení spojení a k přístupu k dalším základním funkcím modemu. Modem je vybaven částí základní sady AT příkazů uvedenou v [8] a doplněn speciálními AT příkazy firmy WAVECOM ([14] a [15]), které zpřístupňují rozšiřující funkce operačního systému modulu.



Obr. 3.6: Komunikace s modemem v prostředí programu Hyperterminál

Počáteční konfiguraci modemu před jeho nasazením je možné provést například v programu Microsoft[®] Hyperterminál, který je součástí operačního systému Microsoft[®] Windows[®]. Po výběru sériového portu PC a nastavení příslušných hodnot komunikačních konstant sériové linky (modulační rychlost, počet datových bitů, parita, počet stop bitů, řízení toku) je možné začít komunikovat s modemem. Výchozí hodnoty jsou dány továrním nastavením modemu a jsou uvedeny v manuálu.

4 Realizace vybrané varianty řešení

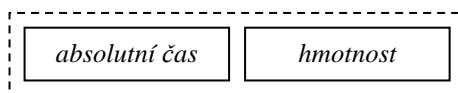
Z předchozího textu je již známa varianta řešení a také konkrétní prostředky, které budou použity. Následující text se věnuje praktické realizaci zvolené varianty. Sestává se z několika dílčích kroků

- návrh struktury dat
- konfigurace GSM modemu MAESTRO 100 a určení sekvence AT příkazů pro posílání dat pomocí FTP
- vytvoření programového vybavení obsluhy modemu pro programovatelný převodník RoutPro FX a doplnění potřebných proměnných do obslužného programu programovatelného terminálu T3A
- ladění.

4.1 Návrh struktury dat

Sebraná data slouží k vytvoření dlouhodobé statistiky naváženého materiálu a také pro kontrolu technologické kázně, z čehož jsem při určení formátu sbíraných dat vycházel.

Důležitými veličinami jsou čas a hmotnost. Společně udávají, jaká hmotnost materiálu prošla pásovou vahou za určitou dobu. Z těchto dvou veličin lze snadno určit celkové navážené množství a kontrolovat, zda linka byla v danou dobu v provozu. Časový údaj lze zapsat buď jako dobu, za kterou prošla naměřená hmotnost, nebo jako absolutní čas uložení záznamu. Absolutní čas je, i přes značný nárůst objemu dat, výhodnější z hlediska kontroly konzistence naměřených hodnot, jelikož ani při náhlém výpadku napájení linky a řídicího systému není narušena kontinuita dat. Rozdíl mezi absolutním časem aktuálního a předchozího záznamu je roven době měření hmotnosti prošlého materiálu. Záznam dat jsem tedy zvolil v uspořádání absolutní čas a hmotnost prošlého materiálu (Obr. 4.1).



Obr. 4.1: Formát sbíraných dat

Pro kontrolu technologické kázně bylo třeba zvolit dostatečně krátké časové úseky měření hmotnosti prošlého materiálu. Aby bylo možné rozlišit pracovní přestávky trvající nejméně 30 minut, omezil jsem nejdelší časový interval na 10 minut. Nejmenší dílek hmotnosti jsem zvolil 1 kg, jelikož pro účel sběru dat je to dostačující přesnost. Denní produkce je přibližně 1 000 t materiálu na jednom pásovém dopravníku (maximálně 7 t za 1 minutu – experimentálně určeno).

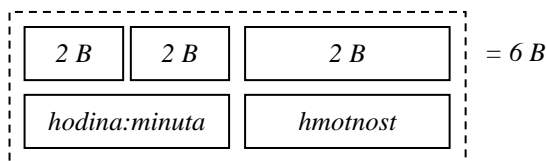
Při návrhu struktury dat bylo třeba zohlednit požadavky na navrhovaný systém sběru dat a technické možnosti použitých komponent jak stávajícího systému, tak jeho navrhovaného rozšíření. Požadavky nebylo třeba striktně dodržet, kdežto technické možnosti zařízení nelze reálně překročit, proto jsem jako hlavní východisko návrhu struktury dat nejdříve uvažoval technické možnosti jednotlivých částí komunikačního řetězce (Obr. 3.1). Jedinou informací, která vyplývá z požadavků a je podstatná pro návrh struktury dat, je možnost zasílat data dávkově. Znamená to, že data mohou být ukládána za určité časové období v řídicí jednotce stávajícího systému a následně až jednorázově zaslána do datového úložiště (DU).

První částí komunikačního řetězce je technologický proces (TP). Tímto technologickým procesem je popsána drtící a třídící linka, jež je v přetržitém provozu po celý rok. Pravidelné odstávky jsou prováděny v nočních hodinách, kdy je ale celá linka včetně řídicího systému odpojena od elektrické sítě. Aby bylo zajištěno, že všechna naměřená data budou vždy uložena a odeslána, zvolil jsem jeden den (24 hodin) jako nejmenší časový interval ukládání souboru dat. Vyhnul jsem se tím nebezpečí, že odesílání bude přerušeno například náhlým odstavením technologie během dne. Strukturu dat jsem rozdělil na dvě totožné části (soubory dat) – jednu určenou pro zápis a jednu pro odeslání. Tyto dva soubory dat se vždy ve své funkci vystřídají před začátkem posílání a bude tím zajištěno, že při předávání souboru dat k odeslání nebudou přepisovány jeho první záznamy. Odesílání souboru dat bude

automaticky zahájeno vždy po spuštění technologie v ranních hodinách, kdy řídicí systém není ještě příliš zatížen.

Za technologickým procesem následuje v komunikačním řetězci stávající řídicí systém (SŘS), jehož podstatnou částí pro návrh datové struktury je programovatelný terminál T3A. Datová paměť tohoto zařízení má velikost 32 kB (32 768 bajtů), z čehož je 4096 bajtů obsazeno systémovými proměnnými a přibližně 100 bajtů je využíváno pro proměnné stávajícího obslužného programu. Volně k užití tedy zbývá zhruba 28 000 bajtů, je-li uvažována rezerva pro případná další programová rozšíření. Obslužný program je napsán v programovacím jazyce SIMPLE3, který rozlišuje čtyři základní datové typy (Tab. 3.2). Z těchto čtyř datových typů přicházejí v úvahu pro reprezentaci sbíraných dat pouze dva - datový typ WORD a REAL. Ze zbývajících dvou datových typů se typ BIT využívá jako logická proměnná a INTEGER zahrnuje záporné hodnoty, které jsou v případě této aplikace zbytečné, a proto nejsou tyto dva datové typy vhodné. Jelikož by rozsah hodnot datového typu REAL nebyl efektivně využit ani při maximálním okamžitém vytížení pásového dopravníku 7 000 kg za 1 minutu, vybral jsem pro uchování času a naměřené hmotnosti datový typ WORD (2 bajty). Rozsah hodnot tohoto datového typu ale nedostačuje při měření hmotnosti materiálu prošlého za dobu 10 minut ($70\,000 > 65\,535$), proto jsem tuto dobu zkrátil na 1 minutu.

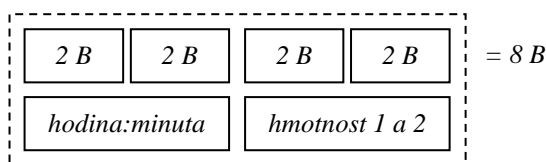
V datové paměti je k dispozici 28 000 bajtů pro dva identické datové soubory, což znamená 14 000 bajtů na datový soubor. Za předpokladu, že datový soubor bude obsahovat data naměřená v intervalech 1 minuty za 1 den (1 440 minut), postačuje pro určení absolutního času jednoho záznamu příslušná minuta a hodina měření – obě hodnoty v proměnné typu WORD. Ke každému souboru dat je třeba ještě doplnit pro jednoznačné určení dne měření tři proměnné typu WORD, které reprezentují datum - rok, měsíc a den. Tyto hodnoty budou společné všem záznamům v příslušném datovém souboru a jejich vyčleněním jsem ušetřil místo v datové paměti.



Obr. 4.2: Formát a velikost jednoho záznamu v datové paměti

Jeden datový záznam potom bude mít velikost 6 bajtů (Obr. 4.2), což znamená, že soubor dat naměřených za 1 den zabere v datové paměti 8 640 bajtů (6 bajtů x 1 440 minut). Pro období delší než 1 den by již datová paměť nevystačila.

Na základě doporučení firmy TEVAS jsem pro případ rozšíření stávajícího systému o vážení na dalším pásovém dopravníku přidal do formátu datového záznamu místo pro další ukládanou hmotnost. Počet záznamů v jednom souboru dat jsem zaokrouhlil na 1 500. Datový záznam tím pádem narostl o další 2 bajty (Obr. 4.3) a celková velikost jednoho souboru dat vzrostla na 12 000 bajtů.



Obr. 4.3: Konečný zvolený formát a velikost datového záznamu

Programovatelný převodník RoutPro FX je vybaven stejně velkou datovou pamětí jako programovatelný terminál T3A. Lze tedy přepokládat, že pro datovou strukturu o celkové velikosti 24 000 bajtů, bude dostatek volné datové paměti.

Další zařízení v komunikačním řetězci již výraznější omezení nemají.

4.2 Konfigurace GSM modemu a určení příkazů odeslání

Než-li jsem mohl začít vytvářet programové vybavení převodníku RoutPro FX pro ovládání funkcí GSM modemu MAESTRO 100, bylo třeba modem nakonfigurovat a zjistit AT příkazy určené k odeslání dat na webový server. Jelikož je modem založen na modulu firmy WAVECOM, čerpal jsem při práci s modemem z podkladů této firmy (zvláště z [14] a [15]). V následujícím textu uvádím soupis použitých AT příkazů s jejich krátkým popisem. Podrobný popis všech

podporovaných AT příkazů včetně výčtu jejich možných parametrů a odpovědí lze nalézt právě v dokumentech [14] a [15].

Konfiguraci a testování modemu jsem prováděl pomocí programu Microsoft® Windows® Hyperterminál. Parametry komunikace jsem nastavil dle manuálu GSM modemu, který k němu byl přiložen.

4.2.1 Formát AT příkazů

AT příkazy jsou řetězce znaků určené pro řízení činnosti modemů a vycházejí z tzv. Hayesovy skupiny příkazů. Základní AT příkazy GSM modemu MAESTRO 100 začínají znaky „AT“, za kterými následuje posloupnost dalších znaků, které určují konkrétní příkaz. AT příkaz je vždy ukončen znakem <CR> a následně zpracován. V následujícím textu jsem tento znak v konkrétních AT příkazech vynechal, jelikož jeho význam je zřejmý.

4.2.2 Konfigurace

V modemu je předinstalován firmware Open AT® (v6.57f) od firmy WAVECOM s implementovaným TCP/IP stackem, díky němuž lze provádět datové přenosy pomocí FTP. Aplikace, která implementuje TCP/IP stack a umožňuje jeho ovládání AT příkazy, se nazývá WIPSoft (v3.21). Tato aplikace se spouští níže uvedeným AT příkazem a informace o jejím spuštění je uchována i po vypnutí modemu.

→ *AT+WOPEN=1*

spustí OpenAT® aplikaci WIPSoft

Konfigurace rozhraní UART

UART (Universal Asynchronous Receiver/Transmitter) obvody slouží k převodu paralelních dat na sériová. V případě tohoto modemu je rozhraní využito pro komunikaci pomocí sériové linky RS-232C. Parametry komunikace jsem nastavil tak, aby její programová obsluha v převodníku RoutPro FX byla co nejjednodušší.

- $AT+IFC=0,0$
nastaví řízení toku – žádné
- $AT+IPR=9600$
nastaví modulační rychlost - 9600 bps
- $AT+ICF=3,4$
nastaví přenosový rámeček – 8 bitů data, 1 stop bit, bez parity

Konfigurace GPRS připojení

Abych se mohl připojit k službě GPRS, musel jsem nakonfigurovat parametry připojení. Mezi tyto parametry patří APN, uživatelské jméno a heslo. Lze je získat od příslušného mobilního operátora – v případě této aplikace od mobilního operátora Vodafone. Po zadání parametrů jsem aktuální konfiguraci uložil do nonvolatilní paměti, která je určena pro dlouhodobé uchování parametrů nastavení.

- $AT+WIPBR=2,6,0,““$
nastaví přihlašovací jméno GPRS nositele
- $AT+WIPBR =2,6,1,““$
nastaví heslo GPRS nositele
- $AT+WIPBR =2,6,11,“internet“$
nastaví APN GPRS nositele
- $AT+WIPBR =6,6,1$
uloží aktuální konfiguraci zvoleného nositele – v tomto případě GPRS

Ostatní konfigurace

- $AT+CLCK=“SC“,0,1234$
zruší autorizace SIM karty pomocí PIN (předpokládám, že karta nebude zneužita)
- $AT+CMEE=0$
nastaví formát chybných hlášení mobilního zařízení na jednoduchou odpověď ERROR (pro realizaci obslužného programu RoutPro FX)

není třeba znát čísla chyb, jelikož nebude řešena programová diagnostika)

→ *AT+WIND=0*

vypne hlavní indikace (rozlišení různých stavů - vyjmutí SIM karty, ztráta sítě, zapnutí zvuku a jiné nejsou třeba, jelikož postačí hlášení obecné chyby)

→ *ATE0*

opakování zaslaných znaků (opakování by zbytečně zatěžovalo sériovou linku a převodník RoutPro FX)

→ *ATVI*

nastaví formát odezvy na jednotný formát pro informační odpovědi i výsledkové kódy <CR><LF>...<CR><LF> (společný formát je pro programovou kontrolu odpovědí vhodnější)

Na závěr konfigurace modemu jsem uložil provedené změny do nonvolatilní paměti EEPROM, aby byly zachovány i po vypnutí modemu a nebylo je třeba po opětovném spuštění znovu zadávat.

→ *AT&W*

uloží aktuální konfiguraci do paměti EEPROM modemu

4.2.3 Určení příkazů pro odeslání dat pomocí FTP

Odeslání dat pomocí FTP je dáno přesnou sekvencí AT příkazů, jejíž podoba je uvedena v [15]. Při zadávání je nutné dodržet pořadí příkazů a vždy před zadáním dalšího příkazu vyčkat na potvrzení provedení aktuálně zadaného.

Níže uvádím upravenou sekvenci z [15] s příslušnými odpověďmi úspěšně vykonaných příkazů. Tato sekvence bude implementována do programového vybavení převodníku RoutPro FX.

→ *AT+WIPCFG=1*

↘ *OK*

spustí TCP/IP stack

- *AT+WIPBR=1,6*
 - ↘ *OK*
 - otevře GPRS nositele
- *AT+WIPBR=4,6,0*
 - ↘ *OK*
 - spustí GPRS nositele
- *AT+WIPCREATE=4,1,"FTP server",21,"jmeno","heslo"*
 - ↘ *OK*
 - vytvoří FTP relaci k specifikovanému FTP serveru (FTP server, jméno, heslo)
- *AT+WIPFILE=4,1,2,"./soubor.txt"*
 - ↘ *CONNECT*
 - vytvoří/přepíše na FTP serveru specifikovaný soubor a přepne UART do datového režimu
- *<data>[ETX]*
 - ↘ *OK*
 - zapiše <data>, ukončí (po zadání znaku <ETX>) datový režim, přepne zpět do režimu AT a odešle <data>
- *AT+WIPCFG=0*
 - ↘ *OK*
 - ukončí TCP/IP stack

Posledním AT příkazem, který se přímo netýká FTP přenosu, ale je užitečný při jeho vykonávání, je softwarový restart. Jestliže dojde k neočekávané chybě a GSM modem přestane správně pracovat, lze tímto příkazem obnovit jeho funkčnost s původními parametry, jestliže jejich nové hodnoty nebyly uloženy do paměti EEPROM.

- *AT+CFUN=1*
 - provede softwarový restart

4.3 Implementace navržené datové struktury do T3A

Navrženou strukturu dat bylo třeba začlenit do stávajícího obslužného programu programovatelného terminálu T3A. Jelikož ale firma TEVAS nepovolila přímou úpravu zdrojového textu tohoto programu, předal jsem jí své požadavky a ona potřebné úpravy provedla sama.

Datová struktura tvořená dvěma soubory dat, z nichž jeden je použit pro zápis a druhý pro odesílání, je realizována dvěma totožnými dvourozměrnými poli o rozměrech 4 x 1 500 proměnných datového typu WORD. Obě pole tedy pojmu 1 500 záznamů, v nichž je obsažena informace o času měření (hodina a minuta) a dvě naměřené hmotnosti.

Ke každému poli jsou asociovány další proměnné, které charakterizují naměřený soubor dat. Pro jednoznačné určení dne ukládání patří mezi tyto proměnné rok, měsíc a den měření. Jelikož není zaručeno, že pole bude vždy naplněno celé, je nutné posílat jen tu část pole, jejíž prvky byly opravdu přepsány. Naplněnou část pole charakterizuje proměnná, která udává počet zapsaných prvků. Poslední z asociovaných proměnných je index právě posílaného záznamu. Ten slouží k obsluze přenosu dat mezi T3A a RoutPro FX.

Obsluha zápisu a střídání funkce polí byla navržena firmou TEVAS. Není třeba se zabývat vlastním principem této obsluhy, jelikož to není pro další realizaci podstatné. Jedinou důležitou informací je to, že zápis do polí je dle návrhu prováděn vždy po 1 minutě.

Inicializace přenosu příslušného pole a asociovaných proměnných po síti HYPEL je automaticky inicializována při změně data. To znamená, že po spuštění technologie a řídicího systému je provedena kontrola změny data a v případě, že se datum neshoduje s uloženým, vzájemně se zamění v obslužném programu T3A funkce polí (odesílání - zápis) a pole určené k odeslání je předáno do sítě, tvořené programovatelným terminálem T3A a programovatelným převodníkem RoutPro FX.

Přenos proměnných mezi T3A a RoutPro FX lze realizovat dvěma způsoby. Buď pomocí síťových proměnných, nebo za použití adresných zpráv (dle [10]). První způsob je založen na sdílení vyhrazené části datové paměti PLC v síti. Jeho implementace do zdrojového textu programu je jednoduchá, ale přenos proměnných

je pomalý a není zaručeno, že síťové proměnné opravdu dorazí do všech zařízení v síti, jelikož jejich úspěšný příjem není potvrzován. Oproti tomu je sdílení dat pomocí adresných zpráv programově komplikovanější, ale zaručuje spolehlivý přenos a možnost zápisu do celé datové paměti libovolného zařízení v síti. Aby nebylo nutné provádět zpětnou kontrolu přijatých dat, zvolil jsem metodu sdílení proměnných pomocí adresných zpráv a předal tento požadavek firmě TEVAS, která provedla patřičné úpravy v programu.

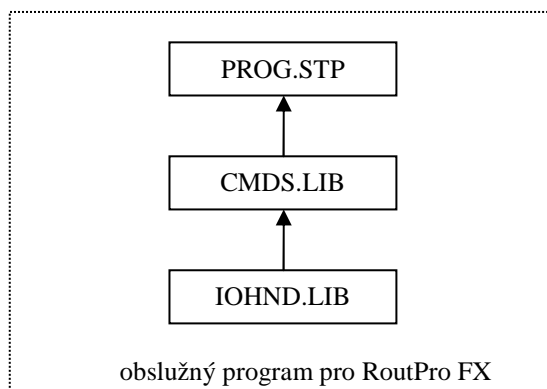
4.4 Vytvoření obslužného programu pro RoutPro FX

Programovatelný převodník RoutPro FX slouží k obsluze GSM modemu pomocí AT příkazů. Obsluhou se rozumí odeslání dat na FTP server zadáním přesné sekvence AT příkazů uvedených v kapitole 4.2.3.

V následujícím textu, ve kterém je popsán vytvořený obslužný program, se věnuji hlavním bodům řešení s důrazem na obecný popis principu komunikace mezi převodníkem a GSM modemem. Podrobnější informace o obsluze sériové linky převodníku, systémových proměnných a další detaily je možné nalézt v literatuře [10], [12] a [13].

4.4.1 Struktura programu

Obslužný program se skládá ze tří částí (Obr. 4.4) – základního programu a dvou knihoven pro obsluhu sériové linky a pro definici zadávaných AT příkazů. Každá část tvoří samostatnou vrstvu programu. Docílil jsem tím oddělení principiálních částí a zpřehlednil zdrojový text pro případné následné úpravy při ladění.



Obr. 4.4: Struktura obslužného programu převodníku RoutPro FX

Soubor dat, určených k přenosu, je uložen v poli, jež má totožnou strukturu jako pole použité pro uchování dat v terminálu T3A včetně asociovaných proměnných. Konkrétní hodnoty prvků pole jsou plněny z terminálu T3A zasíláním adresných zpráv.

IOHND.LIB (Příloha B:)

Tato knihovna slouží pro základní obsluhu vstupní a výstupní vyrovnávací paměti sériové linky. Aby nedošlo k případné ztrátě přijímaných či vysílaných znaků při vyšším vytížení CPU převodníku, použil jsem pro jejich ukládání FIFO fronty. Nové přijaté znaky jsou zapsány na konec fronty, kdežto čtení probíhá od začátku.

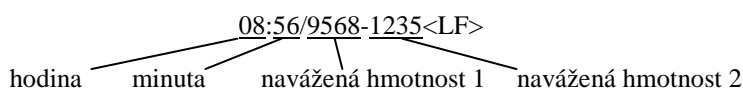
Dále jsou v této knihovně definovány formáty odpovědí modemu, které indikují úspěšné provedení příslušného AT příkazu, a nastaveny parametry komunikace sériové linky. Parametry komunikace sériové linky se shodují s parametry nastavenými v GSM modemu.

Hlavními procedurami obsluhy vyrovnávacích pamětí jsou *TxProcess*, *RxProcess* a *Test*. Procedura *TxProcess* obsluhuje výstupní vyrovnávací paměť sériové linky a procedura *RxProcess* vstupní vyrovnávací paměť. Testování přijatých znaků je prováděno procedurou *Test*, která je porovnává s definovanými formáty odpovědí.

CMDS.LIB (Příloha C:)

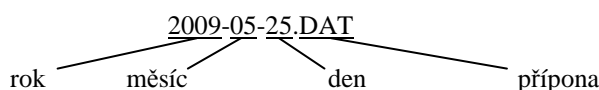
V této knihovně jsou definovány procedury, které předávají k odeslání příslušné AT příkazy. Zadané příkazy jsou následně procedurou *RxProcess* z knihovny **IOHND.LIB** znak po znaku odeslány přes sériovou linku do GSM modemu. Skupina definovaných AT příkazů se shoduje s příkazy užitými v sekvenci pro odeslání dat pomocí FTP. Jejich popis byl již uveden v kapitole 4.2.3, proto není třeba tyto příkazy podrobněji rozebírat.

Za zmínku stojí navržený princip ukládání dat do textového souboru. K formátování záznamů pole slouží procedura *WriteData*, která převede prvky záznamu do textového řetězce (Obr. 4.5). Oddělovací znaky mezi jednotlivými prvky záznamu jsem zvolil tak, aby přenesená data byla srozumitelná i bez předchozího strojového zpracování. Znak <LF> značí konec řádky v unixových systémech. Na rozdíl od odřádkování pomocí dvou znaků <CR><LF>, jež je charakteristické hlavně pro systémy DOS a Microsoft® Windows®, umožňuje ušetřit až 1 440 znaků v jednom textovém souboru.



Obr. 4.5: Ukázka textového řetězce reprezentujícího jeden záznam

Všechny záznamy jsou na FTP serveru uloženy do textového souboru s příponou *.DAT a názvem, který je tvořen datem měření – rok, měsíc a den (Obr. 4.6).



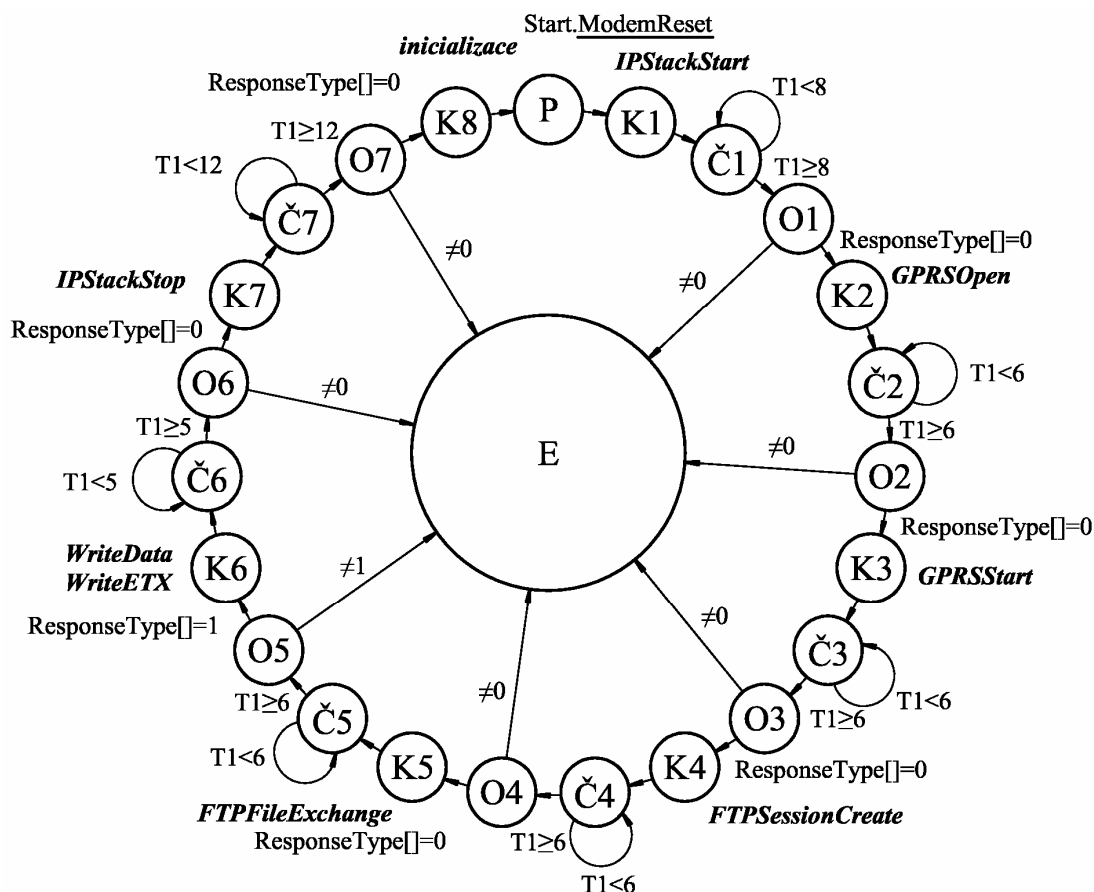
Obr. 4.6: Ukázka názvu textového souboru

Zadání příkazu pro vytvoření textového souboru s příslušným názvem je provedeno procedurou *FTPFileExchange*.

PROG.STP (Příloha D:)

Tento program za pomoci uvedených knihoven **IOHND.LIB** a **CMDS.LIB** vykonává vlastní odesílání dat uchovaných v poli, které je naplněno z terminálu T3A. Odesílání je prováděno přesnou sekvencí AT příkazů, jež zadávají příslušné procedury z knihovny **CMDS.LIB**. Sekvence odesílání je zahájena automaticky po naplnění pole a provádí se opakovaně, dokud není celá úspěšně ukončena. Úspěšné ukončení znamená, že všechny zadané AT příkazy byly bezchybně vykonány a potvrzeny odpovědí příslušného formátu z GSM modemu.

Programová sekvence (Obr. 4.7) je tvořena osmi základními kroky (K) s kontrolami odpovědí (O) a čekacími smyčkami (Č). Výchozím stavem je čekání na spuštění – pohotovost (P). Spuštění sekvence je inicializováno naplněním pole dat z T3A a připraveností modemu (tzn. modem je připraven přijímat a vykonávat AT příkazy). Následuje sedm kroků, v nichž jsou zadávány jednotlivé AT příkazy. Každý z těchto kroků je následován čekací smyčkou (použit časovač T1), ve které se čeká na přijetí odpovědi z modemu. Po uplynutí stanovené doby je provedena kontrola odpovědi. Jestliže se odpověď shoduje s definovanou odpovědí, která indikuje úspěšné provedení AT příkazu, přejde se k provádění dalšího kroku. Osmým a posledním krokem sekvence je inicializace, během které se program navrací do stavu pohotovosti.



Obr. 4.7: Stavový diagram programové sekvence zadávání AT příkazů

V případě, že není detekována očekávaná odpověď, přejde program do chybného stavu (E). Ošetření všech možných chyb a příslušných odpovědí, které mohou být vráceny při vykonávání AT příkazů, je řešeno softwarovým restartem GSM modemu, po němž následuje další pokus o odeslání dat. Diagnostiku přijatých chybových odpovědí jsem neimplementoval, jelikož nebyla nezbytná a počet všech možných odpovědí je příliš vysoký.

4.5 Ladění

Navržený systém sběru dat byl odzkoušen v podmínkách technické laboratoře. Testovány byly jak funkční části, tak systém jako celek. Bylo vyzkoušeno sdílení proměnných mezi T3A a RoutPro FX. Za pomoci programu Hyperterminál byla ověřena správnost formátování dat v RoutPro FX před předáním k odeslání. Ručním zadáním AT příkazů bylo v programu Hyperterminál přes sériový port PC simulováno provádění sekvence odesílání a současně byly kontrolovány odpovědi na

jednotlivé AT příkazy. Funkčnost celého systému byla ověřena několikadenním zkušebním provozem, během něhož byla veškerá sebraná data odeslána na FTP server. Následně byl tento systém firmou TEVAS namontován do reálného provozu a po dobu 14 dnů testován. Během této doby byly také všechny soubory dat odeslány na FTP server firmy DESTRO.

Při nastavování GSM modemu byla odhalena chyba firmwaru. Zadáním níže uvedeného příkazu měly být deaktivovány všechny hlavní indikace.

→ *AT+WIND=0*
↘ *+WIND: 13*

Přestože po každé konfiguraci bylo důsledně prováděno uložení nových parametrů do paměti EEPROM, nedošlo k vypnutí všech indikací. Při zapnutí modemu byla vždy za určitou dobu indikována událost s číselným kódem 13 („rack byl detekován jako uzavřený“). Nejednalo se o závažný nedostatek, ale aby tato odpověď nekolidovala s odpověďmi na AT příkazy sekvence odeslání, zařadil jsem před vykonávání sekvence čekací smyčku. Závada byla nahlášena jak českému dodavateli GSM modemu, tak výrobcí modulu. Ani jeden z nich zatím neodpověděl.

Navržený systém sběru dat byl také testován s řídicím systémem, který byl umístěn na mobilní drticí lince. Mobilní drticí linka je poháněna výkonným spalovacím motorem, v jehož těsné blízkosti se nachází rozváděcí skříň s veškerým zařízením řídicího systému. Jelikož spalovací motor je zdrojem silného rušení a zařízení nebylo možné umístit dále od motoru, bylo nutné provést opatření pro zachování spolehlivosti systému. Přenosová rychlost jak síťové komunikace mezi zařízeními firmy HYPEL, tak komunikace mezi RoutPro FX a GSM modemem byla nastavena na hodnotu 9 600 bps, čímž byl omezen vliv rušení. Aby nebyla snížena intenzita signálu GSM sítě, byla anténa GSM modemu vyvedena ven z rozváděcí skříně.

4.5.1 Návrh PHP skriptu a databáze

V rámci ladění a testování navrženého systému byla ve skriptovacím programovacím jazyce PHP vytvořena jednoduchá webová aplikace pro zpracování a vizualizaci sebraných dat.

Sebraná data jsou touto aplikací na základě manuální inicializace (potvrzení stiskem tlačítka) extrahována z textových souborů, které byly zaslány GSM modemem na webový server, a uložena do databáze MySQL. Z této databáze jsou pomocí PHP skriptu vybrány požadované soubory dat a předány ke zpracování - zobrazení v uživatelském rozhraní aplikace.

Uživatelské rozhraní webové aplikace se skládá z hlavní stránky (Příloha E), textových přehledů (Příloha F) a grafů (Příloha G). Na hlavní stránce se nachází kalendář, pomocí kterého se provádí výběr sebraných dat za zvolené období (dny, týdny a měsíce). Po výběru sebraných dat jsou zobrazeny základní informace o těchto datech (počet záznamů, celkové hmotnosti a jejich součty) a nabídnuta volba jejich zobrazení – textové nebo grafické.

Textové zobrazení je realizováno jednoduchým PHP skriptem (založený na cyklu), který vybraná data zapíše do přehledné tabulky. Pro grafické zobrazení byla vytvořena PHP třída, která pomocí grafické GD knihovny vytvoří obrázek v grafickém formátu PNG s bezztrátovou kompresí. Na tomto obrázku jsou zobrazena vybraná data do grafu s hmotností na vertikální ose a časem na ose horizontální.

Nenáročná jednoduchá webová aplikace umožňuje přehledně zobrazit sebraná data jak v textovém, tak v grafickém formátu. Díky použití databáze lze dle rozmanitých hledisek jednoduše vybírat libovolné množiny uložených dat a dále je zpracovávat statistickými funkcemi. Aplikace je přístupná z kteréhokoliv PC připojenému k internetu a vybavenému internetovým prohlížečem (nezávisí na platformě). Lze ji jednoduše a rychle dodatečně modifikovat, díky čemuž se z ní stává variabilní vizualizační prostředek s mnoha možnostmi.

Závěr

V rámci této bakalářské práce bylo navrženo a realizováno funkční rozšíření stávajícího řídicího systému, určené pro sběr naměřených dat z reálné technologie recyklace sutí.

Byl proveden popis a analýza řízené technologie a stávajícího řídicího systému. Na základě zvážení možností rozšíření tohoto systému a zadaných požadavků na přenos dat byly navrženy varianty řešení. Výhodné vlastnosti bezdrátového přenosu dat v GSM síti podměnily výběr konkrétního návrhu řešení, jež bylo založeno na použití GSM modemu. Dle tohoto návrhu mají být sebraná data uložena na webovém serveru, na který budou přenášena pomocí FTP. Pro zvolenou variantu byla zakoupena potřebná zařízení, jejichž stručný popis předchází samotné realizaci. Při realizaci řešení byla definována struktura přenášených dat a její implementace do systému. S ohledem na automatické programové řešení obsluhy byl nakonfigurován GSM modem. Pro programovatelný převodník RoutPro FX byl vytvořen obslužný program, jenž za použití AT příkazů provádí odesílání dat pomocí GSM modemu. Celý systém byl navíc pokusně doplněn PHP aplikací pro webový server. Sebraná data uložená v textových souborech aplikace vkládá do MySQL databáze a zobrazuje v textových či grafických přehledech.

Navržené rozšíření bylo odzkoušeno jak v podmínkách technické laboratoře, tak v reálném provozu, kde sbíralo po dobu 14 dní naměřená data. Během provozu se nevyskytly žádné závažné závady.

Při řešení této bakalářské práce jsem získal cenné zkušenosti pro analýzu a návrh řídicích systémů. Seznámil jsem se s principem vážení na pásových vahách a možnostmi GSM komunikace s využitím průmyslových GSM modemů. Rozšířil jsem své zkušenosti s automatizační technikou.

Dalším rozšířením této bakalářské práce by mohlo být kompletní webové rozhraní pro vizualizaci dat. Systém by měl být opatřen heslem, aby nebyl veřejně

přístupný. Měl by umožňovat grafické i textové zobrazení sebraných dat za libovolné období a měl by být vybaven většinou statistických funkcí.

Modul WAVECOM, který je použit v GSM modemu MAESTRO 100, umožňuje díky firmwaru Open AT[®] definovat vlastní uživatelské aplikace. Naprogramováním odesílání dat do jedné aplikace spouštěné specifickým AT příkazem by bylo možné zjednodušit programové vybavení převodníku RoutPro FX (nebylo by třeba řešit sekvenci zadávání AT příkazů) a zvýšit tak spolehlivost systému jako celku.

Použité zdroje

- [1] BLUNÁR, Karol; DIVIŠ, Zdeněk. *Telekomunikační sítě – díl 2*. 1. vyd. Ostrava: Vysoká škola báňská – Technická univerzita Ostrava, 2006. 631 s. ISBN 80-248-1077-8.
- [2] HANUS, Stanislav. *Bezdrátové a mobilní komunikace*. 1. vyd. Brno: VUT, 2002. 134 s. Fakulta elektrotechniky a komunikačních technologií. ISBN 80-214-1833-8.
- [3] JENČÍK, Josef; VOLF, Jaromír. *Technická měření*. 1. vyd. Praha: ČVUT, 2000. 212 s. Fakulta strojní. ISBN 80-01-02138-6.
- [4] KOKEŠ, Josef. *Algoritmy pro inženýrskou informatiku*. 1. vyd. Praha: ČVUT, 2006. 249 s. Fakulta strojní. ISBN 80-01-03515-8.
- [5] PUŽMANOVÁ, Rita. *TCP/IP v kostce*. 1. vyd. České Budějovice: KOPP, 2004. 607 s. ISBN 80-7232-236-2.
- [6] VODRÁŽKA, Jiří. *Přenosové systémy v přístupové síti*. 2. vyd. Praha: ČVUT, 2006. 196 s. Fakulta elektrotechnická. ISBN 80-01-03386-4.
- [7] TYBOR, Josef. *Váhy automatické a speciální*. 2. vyd. Praha: SNTL, 1974. 191 s. 04-224-74.
- [8] ETSI TS 100 916 V7.4.0 (1999-11). *Digital Cellular Telecommunications Systems (Phase 2+); AT Command Set for GSM Mobile Equipment (ME)*. 1999.
- [9] FCC průmyslové systémy. Praha. *Instalační manuál pro GSM modemy Maestro-100 a Maestro-100 lite*. 2006. 12 s.
- [10] Hynek Pangrác – HYPEL. Kladno. *Automatizační prostředky HYPEL*. 2004. 52 s.
- [11] Hynek Pangrác – HYPEL. Kladno. *DDE server – komunikační server pro automaty firmy HYPEL*. 2008. 16 s.
- [12] Hynek Pangrác – HYPEL. Kladno. *ROUTPRO FX – katalogový list*. 2007. 10 s.
- [13] Hynek Pangrác – HYPEL. Kladno. *SIMPLE3 – popis jazyka*. 2001. 56 s.
- [14] WAVECOM. Issy-les-Moulineaux Ceres, Francie. *AT Commands Interface Guide for Open AT® Firmware v6.57f*. 2008.

- [15] WAVECOM. Issy-les-Moulineaux Ceres, Francie. *WIP AT Commands User Guide (WIPSoft v3.21)*. 2008.
- [16] *GSM/CSD versus GSM/GPRS*. URL:
<<http://hw.cz/Firemni-clanky/Insys/ART1636-GSM-CSD-versus-GSM-GPRS.html>> [cit. 2009-05-11]
- [17] *Když se řekne Wavecom*. URL:
<<http://hw.cz/firemni-clanky/glyn-czech-republic/art2195-kdyz-se-rekne-wavecom.html>> [cit. 2008-12-06]
- [18] *Hynek Pangrác – HYPEL*. URL: <<http://www.hypel.cz>>
- [19] *Maestro Wireless Solutions Ltd*. URL:
<<http://www.maestro-wireless.com/modules/index.php>>
- [20] *SELET SENSOR S.r.l.*. URL: <<http://www.selet.it>>

Přílohy

Příloha A:	Obsah přiloženého CD
Příloha B:	RoutPro FX - zdrojový text IOHND.LIB
Příloha C:	RoutPro FX - zdrojový text CMDS.LIB
Příloha D:	RoutPro FX - zdrojový text PROG.STP
Příloha E:.....	Hlavní stránka webové aplikace
Příloha F:.....	Textový přehled sebraných dat za jeden den
Příloha G:	Grafický přehled sebraných dat za jeden den

Příloha A: Obsah příloženého CD

Dokumentace k GSM modemu

/doc-gsm-modem/

AT Command Interface Guide Open AT Firmware v6.57f.pdf

AT Commands User Guide for Wavecom IP v3.21.pdf

Manual pro GSM modem Maestro 100.pdf

Dokumentace k automatizačním prostředkům firmy HYPEL

/doc-hypel/

Automatizacni prostredky HYPEL.pdf

RoutPRO FX - Katalogovy list.pdf

SIMPLE3 - Popis jazyka.pdf

Implementace programového vybavení pro RoutPro FX

/imp-routpro-fx/

prog.stp

iohnd.lib

cmds.lib

Text bakalářské práce v elektronické podobě

/bd.pdf

/bd.doc

Příloha B: RoutPro FX - zdrojový text IOHND.LIB

```

1: // *****
2: // * IOHND.LIB *
3: // * knihovna procedur pro obsluhu vstupu/vystupu *
4: // *****
5:
6: // definice konstant
7: // -----
8: const
9: 1 # Bd2400;
10: 2 # Bd4800;
11: 3 # Bd9600;
12: 4 # Bd19200;
13: 5 # Bd38400;
14: 6 # Bd57600;
15: 7 # Bd115200;
16: 2 # NumberOfResponseTypes; // urcuje index posledni preddefinovane odpovedi
17: end
18:
19: // definice promennych
20: // -----
21: var
22: I0 # TextBuffer : array[1119] of word;
23: I0 # RxBuffer : array[31] of word;
24: O0 # TxBuffer : array[15] of word;
25: O16 # TxPointer : word;
26: O17 # RxPointer : word;
27: O18 # BaudRate : word;
28: Y9 # StopBits : bit;
29: Y10 # Parity : bit;
30: Y11 # ParityType : bit;
31: Y13 # ParityError : bit;
32: Y7 # GabMode : bit;
33: Y0 # RTS : bit;
34: X0 # CTS : bit;
35:
36: // 'First In, First Out' fronta Rx
37: RxFifo : array[15, 31] of word; // Rx fronta
38: RxRdPointer : word; // ukazatel cteni Rx bufferu
39: RxFifoWrPointer : word; // ukazatel zapisu do Rx FIFO
40: CRLF : bit; // promenna detekce prvnioho a druheho odradkovani
41:
42: // 'First In, First Out' fronta Tx
43: TxFifo : array[300] of word; // Tx fronta
44: TxFifoWrPointer : word; // ukazatel zapisu do Tx FIFO
45: TxFifoRdPointer : word; // ukazatel cteni z Tx FIFO
46:
47: // promenne pro kontrolu odpovedi
48: Response : array[15, 31] of word; // pole prijatych odpovedi
49: ResponseType : array[31] of word; // pole prirazeni typu odpovedi k jednotlivym odpovedim
50: ResponseWrPointer : word; // pomocna promenna pro zapis do pole prijatych odpovedi
51: ResponseRdPointer : word; // pomocna promenna pro prochazeni pole prijatych odpovedi
52: TestedResponse : word; // index testovaneho typu odpovedi
53: TestedChar : word; // index testovaneho znaku
54: Compare : bit; // povoleni testovani odpovedi
55:
56: DisplayRdPointer : word; // ukazatel pro posuv pri cteni z virtualniho displeje
57: end
58:
59: // definice obsluznych procedur
60: // -----
61:

```

```
62: // Init - inicializace po restartu programovatelného převodníku
63: Procedure Init
64:   BaudRate = Bd9600; // nastavení baudové rychlosti komunikace RS-232C na 9600 kbit/s
65:
66:   RxPointer = 0;
67:   RxRdPointer = 0;
68:   RxFifoWrPointer = 0;
69:   CRLF';
70:
71:   TxPointer = 0;
72:   TxFifoWrPointer = 0;
73:   TxFifoRdPointer = 0;
74:
75:   ResponseWrPointer = 0;
76:   ResponseRdPointer = 0;
77:   TestedResponse = 0;
78:   TestedChar = 2;
79:   Compare';
80:
81:   DisplayRdPointer = 0;
82:   POSITION = 0; // ukazatel aktuální pozice posledního prvku na virtuálním displeji
83:
84:   // definice odpovědi [CR][LF]OK[CR][LF]
85:   Response[0, 0] = 13;
86:   Response[0, 1] = 10;
87:   Response[0, 2] = 79;
88:   Response[0, 3] = 75;
89:   Response[0, 4] = 13;
90:   Response[0, 5] = 10;
91:
92:   // definice odpovědi [CR][LF]CONNECT[CR][LF]
93:   Response[1, 0] = 13;
94:   Response[1, 1] = 10;
95:   Response[1, 2] = 67;
96:   Response[1, 3] = 79;
97:   Response[1, 4] = 78;
98:   Response[1, 5] = 78;
99:   Response[1, 6] = 69;
100:   Response[1, 7] = 67;
101:   Response[1, 8] = 84;
102:   Response[1, 9] = 13;
103:   Response[1, 10] = 10;
104:
105:   // definice odpovědi [CR][LF]CONNECT[CR][LF]
106:   Response[2, 0] = 13;
107:   Response[2, 1] = 10;
108:   Response[2, 2] = 43;
109:   Response[2, 3] = 87;
110:   Response[2, 4] = 73;
111:   Response[2, 5] = 78;
112:   Response[2, 6] = 68;
113:   Response[2, 7] = 58;
114:   Response[2, 8] = 32;
115:   Response[2, 9] = 49;
116:   Response[2, 10] = 51;
117:   Response[2, 11] = 13;
118:   Response[2, 12] = 10;
119:
120:   // počáteční vyplnění pole typu odpovědi
121:   ResponseType[0] = NumberOfResponseTypes + 1;
122:   ResponseType[1] = NumberOfResponseTypes + 1;
123:   ResponseType[2] = NumberOfResponseTypes + 1;
124:   ResponseType[3] = NumberOfResponseTypes + 1;
125:   ResponseType[4] = NumberOfResponseTypes + 1;
126:   ResponseType[5] = NumberOfResponseTypes + 1;
```

```

127:   ResponseType[6] = NumberOfResponseTypes + 1;
128:   ResponseType[7] = NumberOfResponseTypes + 1;
129:   ResponseType[8] = NumberOfResponseTypes + 1;
130:   ResponseType[9] = NumberOfResponseTypes + 1;
131:   ResponseType[10] = NumberOfResponseTypes + 1;
132:   ResponseType[11] = NumberOfResponseTypes + 1;
133:   ResponseType[12] = NumberOfResponseTypes + 1;
134:   ResponseType[13] = NumberOfResponseTypes + 1;
135:   ResponseType[14] = NumberOfResponseTypes + 1;
136:   ResponseType[15] = NumberOfResponseTypes + 1;
137:   ResponseType[16] = NumberOfResponseTypes + 1;
138:   ResponseType[17] = NumberOfResponseTypes + 1;
139:   ResponseType[18] = NumberOfResponseTypes + 1;
140:   ResponseType[19] = NumberOfResponseTypes + 1;
141:   ResponseType[20] = NumberOfResponseTypes + 1;
142:   ResponseType[21] = NumberOfResponseTypes + 1;
143:   ResponseType[22] = NumberOfResponseTypes + 1;
144:   ResponseType[23] = NumberOfResponseTypes + 1;
145:   ResponseType[24] = NumberOfResponseTypes + 1;
146:   ResponseType[25] = NumberOfResponseTypes + 1;
147:   ResponseType[26] = NumberOfResponseTypes + 1;
148:   ResponseType[27] = NumberOfResponseTypes + 1;
149:   ResponseType[28] = NumberOfResponseTypes + 1;
150:   ResponseType[29] = NumberOfResponseTypes + 1;
151:   ResponseType[30] = NumberOfResponseTypes + 1;
152:   ResponseType[31] = NumberOfResponseTypes + 1;
153:   return
154:
155:   // DispInit - inicializace virtualniho displeje
156:   Procedure DisplayInit
157:     POSITION = 0;
158:     DisplayRdPointer = 0;
159:   return
160:
161:   // Inc - inkrementace cisla typu word
162:   Procedure Inc(var Num : word)
163:     Num = Num + 1;
164:   return
165:
166:   // ToTxFifo - zapis cisla typu WORD do TxFifo
167:   Procedure ToTxFifo(Num : word)
168:     TxFifo[TxFifoWrPointer] = Num;
169:     if TxFifoWrPointer < 300 then Inc(TxFifoWrPointer) else TxFifoWrPointer = 0 endif
170:   return
171:
172:   // TxProcess - vyprazdnovani a posilani jednoho znaku z TxFifo
173:   Procedure TxProcess
174:     if TxFifoWrPointer <> TxFifoRdPointer and TxPointer = 0 then
175:       TxBuffer[0] = TxFifo[TxFifoRdPointer];
176:       TxPointer = 1;
177:       if TxFifoRdPointer < 300 then Inc(TxFifoRdPointer) else TxFifoRdPointer = 0 endif
178:     endif
179:   return
180:
181:   // RxProcess - vyprazdnuje postupne RxBuffer a zapisuje pri kazdem zavolani jeden znak do RxFifa
182:   Procedure RxProcess
183:     if RxPointer <> RxRdPointer then
184:       if RxRdPointer < 32 then Inc(RxRdPointer) else RxRdPointer = 1 endif
185:       RxFifo[ResponseWrPointer, RxFifoWrPointer] = RxBuffer[RxRdPointer - 1];
186:       if RxBuffer[RxRdPointer - 1] = 10 then
187:         if CRLF' then
188:           CRLF;
189:           if RxFifoWrPointer < 15 then Inc(RxFifoWrPointer) else RxFifoWrPointer = 0 endif
190:         else
191:           RxFifoWrPointer = 0;

```

```
192:     CRLF';
193:     if ResponseWrPointer < 15 then Inc(ResponseWrPointer) else ResponseWrPointer = 0 endif
194:     endif
195:     else
196:     if RxFifoWrPointer < 31 then Inc(RxFifoWrPointer) else RxFifoWrPointer = 0 endif
197:     endif
198:     endif
199:     return
200:
201: // PutChar - vrací kód ASCII znaku z virtuálního displeje
202: Function PutChar(Pos: word) : word;
203: var
204:     HalfPos : word;
205: end
206: HalfPos = Pos / 2;
207: if Pos - HalfPos * 2 = 0 then
208:     PutChar = TextBuffer[HalfPos + 1040] / 256
209: else
210:     PutChar = TextBuffer[HalfPos + 1040] - (TextBuffer[HalfPos + 1040] / 256) * 256
211:     endif
212: return
213:
214: // SendText - posílá znaky z virtuálního displeje na modem
215: Procedure SendText
216: if POSITION > DisplayRdPointer then
217:     ToTxFifo(PutChar(DisplayRdPointer));
218:     Inc(DisplayRdPointer);
219:     endif
220: return
221:
222: // Test - testuje přijaté odpovědi a prirazuje jim typy
223: Procedure Test
224: var
225:     AChar, BChar : word;
226:     CurrentResponse : word;
227:     Test : bit;
228: end
229: if ResponseRdPointer <> ResponseWrPointer and Compare' then
230: if RxFifo[ResponseRdPointer, 0] = 13 and RxFifo[ResponseRdPointer, 1] = 10 then
231:     Compare;
232:     Test;
233:     CurrentResponse = ResponseRdPointer;
234:     endif
235: if ResponseRdPointer < 15 then
236:     Inc(ResponseRdPointer);
237:     else
238:     ResponseRdPointer = 0;
239:     endif
240:     TestedChar = 2;
241:     TestedResponse = 0;
242:     endif
243: if Compare then
244: if TestedChar = 31 or RxFifo[CurrentResponse, TestedChar] = 10 then
245:     if Test then
246:         ResponseType[CurrentResponse] = TestedResponse;
247:         Compare';
248:     else
249:         if TestedResponse = NumberOfResponseTypes then
250:             ResponseType[CurrentResponse] = NumberOfResponseTypes + 1;
251:         endif
252:     endif
253:     TestedChar = 31;
254:     endif
255:     AChar = RxFifo[CurrentResponse, TestedChar];
256:     BChar = Response[TestedResponse, TestedChar];
```

```
257:     if AChar <> BChar and AChar - BChar <> 'a' - 'A' and BChar - AChar <> 'a' - 'A' then Test'; endif
258:     if TestedChar < 31 then
259:         Inc(TestedChar);
260:     else
261:         TestedChar = 2;
262:         Test;
263:         if TestedResponse < NumberOfResponseTypes then
264:             Inc(TestedResponse);
265:         else
266:             Compare';
267:             TestedResponse = 0;
268:         endif
269:     endif
270:     endif
271:     return
272:
273: // Process - základni obsluha prijmu a odesilani
274: Procedure Process
275:     SendText;
276:     TxProcess;
277:     RxProcess;
278:     Test;
279:     return
```

Příloha C: RoutPro FX - zdrojový text CMDS.LIB

```

1: // *****
2: // * CMDS.LIB *
3: // * knihovna definic AT prikazu *
4: // *****
5:
6: // pomocne funkce/procedury
7: // -----
8: Procedure Tens(n : word)
9:   if n < 10 then DISPLAY = 0; endif
10:  DISPLAY = n;
11:  return
12:
13: // AT prikazy
14: // -----
15: procedure AT
16:  DisplayInit;
17:  FORMAT = 0;
18:  DISPLAY = "AT";
19:  FORMAT = 220;
20:  DISPLAY = 13;
21:  return
22:
23: procedure WriteETX // [ETX] = End Of Text - ukonceni datoveho prenosu pri poruse
24:  FORMAT = 220;
25:  DISPLAY = 3; // ZNAK - End of Text [ETX]
26:  return
27:
28: procedure SoftwareReset
29:  DisplayInit;
30:  WriteETX;
31:  FORMAT = 0;
32:  DISPLAY = "AT+CFUN=1";
33:  FORMAT = 220;
34:  DISPLAY = 13;
35:  return
36:
37: procedure IPStackStart
38:  DisplayInit;
39:  FORMAT = 0;
40:  DISPLAY = "AT+WIPCFG=1";
41:  FORMAT = 220;
42:  DISPLAY = 13;
43:  return
44:
45: procedure IPStackStop
46:  DisplayInit;
47:  FORMAT = 0;
48:  DISPLAY = "AT+WIPCFG=0";
49:  FORMAT = 220;
50:  DISPLAY = 13;
51:  return
52:
53: procedure GPRSOpen
54:  DisplayInit;
55:  FORMAT = 0;
56:  DISPLAY = "AT+WIPBR=1,6";
57:  FORMAT = 220;
58:  DISPLAY = 13;
59:  return
60:
61: procedure GPRSStart

```

```

62:  DisplayInit;
63:  FORMAT = 0;
64:  DISPLAY = "AT+WIPBR=4,6,0";
65:  FORMAT = 220;
66:  DISPLAY = 13;
67:  return
68:
69:  procedure FTPSessionCreate
70:  DisplayInit;
71:  FORMAT = 0;
72:  DISPLAY = "AT+WIPCREATE=4,1,";
73:  FORMAT = 220;
74:  DISPLAY = 34;
75:  FORMAT = 0;
76:  DISPLAY = "****";           // FTP - nazev serveru
77:  FORMAT = 220;
78:  DISPLAY = 34;
79:  FORMAT = 0;
80:  DISPLAY = ",21,";
81:  FORMAT = 220;
82:  DISPLAY = 34;
83:  FORMAT = 0;
84:  DISPLAY = "****";           // FTP - prihlasovací jmeno
85:  FORMAT = 220;
86:  DISPLAY = 34;
87:  FORMAT = 0;
88:  DISPLAY = ",";
89:  FORMAT = 220;
90:  DISPLAY = 34;
91:  FORMAT = 0;
92:  DISPLAY = "****";           // FTP - prihlasovací heslo
93:  FORMAT = 220;
94:  DISPLAY = 34;
95:  FORMAT = 220;
96:  DISPLAY = 13;
97:  return
98:
99:  procedure FIPFileExchange(y, m, d : word)
100: DisplayInit;
101: FORMAT = 0;
102: DISPLAY = "AT+WIPFILE=4,1,2,";
103: FORMAT = 220;
104: DISPLAY = 34;           // ZNAK - apostrof [']
105: FORMAT = 0;
106: DISPLAY = "****";           // FTP - slozka
107:
108: FORMAT = 0;
109: DISPLAY = y;           // rok
110: FORMAT = 220;
111: DISPLAY = 45;           // ZNAK - pomlcka [-]
112: FORMAT = 0;
113: Tens(m);           // mesic
114: FORMAT = 220;
115: DISPLAY = 45;           // ZNAK - pomlcka [-]
116: FORMAT = 0;
117: Tens(d);           // den
118: DISPLAY = ".dat";           // pripona souboru .DAT
119: {
120: FORMAT = 0;
121: DISPLAY = "****";           // FTP - slozka
122: }
123: FORMAT = 220;
124: DISPLAY = 34;           // apostrof '
125: DISPLAY = 13;
126: return

```

```
127:
128: procedure WriteData(h, m, value1, value2 : word)
129:   DisplayInit;
130:   FORMAT = 0;
131:   Tens(h);
132:   FORMAT = 220;
133:   DISPLAY = 58;
134:   FORMAT = 0;
135:   Tens(m);
136:   FORMAT = 220;
137:   DISPLAY = 47;
138:   FORMAT = 0;
139:   DISPLAY = value1;
140:   FORMAT = 220;
141:   DISPLAY = 45;
142:   FORMAT = 0;
143:   DISPLAY = value2;
144:   //FORMAT = 220;
145:   //DISPLAY = 13;
146:   FORMAT = 220;
147:   DISPLAY = 10;
148:   return
```


Příloha D: RoutPro FX - zdrojový text PROG.STP

```

1: // *****
2: // * PROG.STP *
3: // * zakladni program *
4: // *****
5:
6: #Configuration "ROUTPRO FX" // direktiva definuje pametovou konfiguraci a typ automatu
7: #Pragma $LJ // prepnuti prekladace do rezimu dlouhych skoku
8: #Include "iohnd.lib" // knihovna obsluhy vstupu/vystupu
9: #Include "cmds.lib" // knihovna AT prikazu
10:
11: netaddr(0)
12:
13: // definice promennych a konstant
14: // -----
15: const
16: 300 # CycleWait;
17: end
18:
19: var
20: NetYear : word;
21: NetMonth : word;
22: NetDay : word;
23: CurrentRecord : word; // index soucasne zapisovaneho zaznamu
24: CurrentHour : word;
25: CurrentMinute : word;
26: Value1 : word;
27: Value2 : word;
28: Data : array [3, 1499] of word;
29:
30: D40 # NumberOfRecords : word; // celkovy pocet zaznamu
31:
32: Step : word; // promenna pro rizeni kroku sekvence
33: Start : bit; // NEPOUZITO - zalezi na implementaci!!!
34: ModemReset : bit;
35: ExpectedResponse : word; // promenna zaznamenavajici index ocekavane odpovedi
36: i, j : word;
37: end
38:
39: // pomocne procedury
40: // -----
41:
42: // ModemSWReset - SW reset modemu
43: procedure ModemSWReset
44: TEN1';
45: T1 = 0;
46: Step = 0;
47: SoftwareReset;
48: ModemReset;
49: TEN0;
50: i = 0;
51: j = CycleWait;
52: return
53:
54: // GetValues - obsluha predavani pole dat
55: procedure GetValues
56: Data[0, CurrentRecord] = CurrentHour;
57: Data[1, CurrentRecord] = CurrentMinute;
58: Data[2, CurrentRecord] = Value1;
59: Data[3, CurrentRecord] = Value2;
60: return
61:

```

```
62: // počatecni inicializace po resetu převodníku
63: // -----
64: if RESET then
65:   SENDCODE = 0;
66:
67:   Init;           // počatecni inicializace promenných
68:
69:   Step = 0;      // nulování kroku
70:   Start';
71:   ModemReset;
72:
73:   T1 = 0;       // nulování časovace T1
74:   TEN1';       // povolení citání časovace 1 - log. 1 -> časovac číta
75:   TPA1;        // zarazení předelice 1/100 - log. 1 -> číta po 1 s
76:
77:   T0 = 0;       // nulování časovace T0
78:   TEN0';       // povolení citání časovace 1 - log. 1 -> časovac číta
79:   TPA0;        // zarazení předelice 1/100 - log. 1 -> číta po 1 s
80:
81:   i = 0;
82:   j = CycleWait;
83:
84:   RESET';
85: endif
86:
87: // vlastní tělo programu
88: // -----
89:
90: // základní obslužný proces
91: Process;
92:
93: // obsluha posílání záznamů přes síť
94: if CurrentRecord < NumberOfRecords - 1 then
95:   GetValues;    // je-li aktuální hodnota záznamu menší než jejich počet, ukládá
96:   Start';
97: else
98:   Start;       // po naplnění pole hodnot se spustí proces odesílání
99: endif
100:
101: // čekání po výpadku napájení nebo po zaměření resetu na přihlášení modemu
102: if ModemReset then
103:   if T0 >= 35 then
104:     ModemReset';
105:     TEN0';
106:     T0 = 0;
107:   endif
108: endif
109:
110: // základní obsluha posílání dat
111: if Step = 0 and Start and ModemReset' then
112:   T1 = 0;
113:   TEN1;
114:   ExpectedResponse = ResponseWrPointer;
115:   IPStackStart;
116:   Start';
117:   Inc(Step);
118: endif
119:
120: if T1 >= 8 and Step = 1 then
121:   if ResponseType[ExpectedResponse] = 0 then
122:     ExpectedResponse = ResponseWrPointer;
123:     T1 = 0;
124:     GPRSOpen;
125:     Inc(Step);
126:   else
```

```
127:     ModemSWReset
128:     endif
129: endif
130:
131: if T1 >= 6 and Step = 2 then
132:     if ResponseType[ExpectedResponse] = 0 then
133:         ExpectedResponse = ResponseWrPointer;
134:         T1 = 0;
135:         GPRSStart;
136:         Inc(Step);
137:     else
138:         ModemSWReset
139:     endif
140: endif
141:
142: if T1 >= 6 and Step = 3 then
143:     if ResponseType[ExpectedResponse] = 0 then
144:         ExpectedResponse = ResponseWrPointer;
145:         T1 = 0;
146:         FTPSessionCreate;
147:         Inc(Step);
148:     else
149:         ModemSWReset
150:     endif
151: endif
152:
153: if T1 >= 6 and Step = 4 then
154:     if ResponseType[ExpectedResponse] = 0 then
155:         ExpectedResponse = ResponseWrPointer;
156:         T1 = 0;
157:         FTPFileExchange(NetYear, NetMonth, NetDay);
158:         Inc(Step);
159:     else
160:         ModemSWReset
161:     endif
162: endif
163:
164: if T1 >= 6 and Step = 5 then
165:     if ResponseType[ExpectedResponse] = 1 then
166:         // BEGIN DATA
167:         if j = CycleWait then
168:             j = 0;
169:             WriteData(Data[0, i], Data[1, i], Data[2, i], Data[3, i]);
170:             Inc(i);
171:         endif
172:         Inc(j);
173:         if i = NumberOfRecords or i = 1500 then
174:             i = 0;
175:             j = CycleWait;
176:             ExpectedResponse = ResponseWrPointer;
177:             T1 = 0;
178:             WriteEIX;
179:             Inc(step);
180:         endif
181:         // END DATA
182:     else
183:         ModemSWReset
184:     endif
185: endif
186:
187: if T1 >= 5 and Step = 6 then
188:     if ResponseType[ExpectedResponse] = 0 then
189:         ExpectedResponse = ResponseWrPointer;
190:         T1 = 0;
191:         IPStackStop;
```

```
192:     Inc(Step);
193:     else
194:         ModemSWReset
195:     endif
196: endif
197:
198: if T1 >= 12 and Step = 7 then
199:     if (ResponseType[ExpectedResponse] = 0 or ResponseType[ExpectedResponse] = 2) then
200:         TEN1';
201:         T1 = 0;
202:         Step = 0;
203:         CurrentRecord = 0;
204:         NumberOfRecords = 1500;
205:     else
206:         ModemSWReset
207:     endif
208: endif
209:
210: end
```

Příloha E: Hlavní stránka webové aplikace

Ukázka hlavní stránky webové aplikace s vybranými daty, naměřenými za jeden den.

DESTRO - Statistika výroby

Předchozí			4	2009	Další		
Týden	Po	Út	St	Čt	Pá	So	Ne
14			1	2	3	4	5
15	6	7	8	9	10	11	12
16	13	14	15	16	17	18	19
17	20	21	22	23	24	25	26
18	27	28	29	30			

Rychlý přehled: středa 29. 4. 2009

Základní informace:	
Váhy	Drcení 436.13 tun
	Třídění 6.36 tun
	Celkem 442.48 tun
Záznamů	1140

Nástroje

:: Správce databáze - kontrola zápisu dat do databáze ::
Upozornění: Data, která nejsou uložena v databázi, nebudou zahrnuta do týdenních/měsíčních statistik! Denních přehledů se to netýká.
 Z důvodu zajištění bezpečného provozu databáze a jejího nezahlcení je nutné provádět import dat MANUÁLNĚ.

:: Hlášení závady - e-mailový formulář pro hlášení závady ::
Upozornění: Ve formuláři je použita ochrana proti internetovým robotům, kteří spamují pomocí nazabezpečených formulářů na zadaný e-mail.
 Do daného okna, zapíšte číslo číslicí.

Systémové zprávy

Zdroj dat: Počet nahraných souborů 527 ve složce "./data/".

Příloha F: Textový přehled sebraných dat za jeden den

Ukázka části textového přehledu sebraných dat zobrazená ve formě tabulky.

Přehled naměřených hodnot ze dne 21. 5. 2009

Pořadí hodnoty	Čas [hod:min]	Drcení [tun]	Třídění [tun]	Součet [tun]	Pořadí hodnoty	Čas [hod:min]	Drcení [tun]	Třídění [tun]	Součet [tun]	Pořadí hodnoty	Čas [hod:min]	Drcení [tun]	Třídění [tun]	Součet [tun]
1	05:00	0,00	0,00	0,00	286	09:45	2,24	0,00	2,24	571	14:31	0,00	0,00	0,00
2	05:01	0,00	0,00	0,00	287	09:46	1,94	0,00	1,94	572	14:32	0,00	0,00	0,00
3	05:02	0,00	0,00	0,00	288	09:47	1,95	0,00	1,95	573	14:33	0,00	0,00	0,00
4	05:03	0,00	0,00	0,00	289	09:48	2,19	0,00	2,19	574	14:34	0,00	0,00	0,00
5	05:04	0,00	0,00	0,00	290	09:49	2,27	0,00	2,27	575	14:35	0,00	0,00	0,00
6	05:05	0,00	0,00	0,00	291	09:50	1,76	0,00	1,76	576	14:36	0,00	0,00	0,00
7	05:06	0,00	0,00	0,00	292	09:51	1,39	0,00	1,39	577	14:37	0,00	0,00	0,00
8	05:07	0,00	0,00	0,00	293	09:52	0,66	0,00	0,66	578	14:38	0,00	0,00	0,00
9	05:08	0,00	0,00	0,00	294	09:53	0,00	0,00	0,00	579	14:39	0,00	0,00	0,00
10	05:09	0,00	0,00	0,00	295	09:54	0,00	0,00	0,00	580	14:40	0,00	0,00	0,00
11	05:10	0,00	0,00	0,00	296	09:55	0,00	0,00	0,00	581	14:41	0,00	0,00	0,00
12	05:11	0,00	0,00	0,00	297	09:56	0,00	0,00	0,00	582	14:42	0,00	0,00	0,00
13	05:12	0,00	0,00	0,00	298	09:57	0,00	0,00	0,00	583	14:43	0,00	0,00	0,00
14	05:13	0,00	0,00	0,00	299	09:58	0,00	0,00	0,00	584	14:44	0,00	0,00	0,00
15	05:14	0,00	0,00	0,00	300	10:00	0,00	0,02	0,02	585	14:45	0,00	0,00	0,00
16	05:15	0,00	0,00	0,00	301	10:01	0,00	0,00	0,00	586	14:46	0,00	0,00	0,00
17	05:16	0,00	0,00	0,00	302	10:02	0,00	0,00	0,00	587	14:47	0,00	0,00	0,00
18	05:17	0,00	0,00	0,00	303	10:03	0,00	0,00	0,00	588	14:48	0,00	0,00	0,00

Příloha G: Grafický přehled sebraných dat za jeden den

Ukázka grafického přehledu sebraných dat v grafickém formátu PNG (graf není zobrazen celý).

