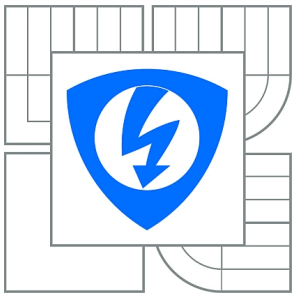
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF RADIO ELECTRONICS

VYHODNOCENÍ SPOTŘEBY OSOBNÍ LODI

PASSENGER VESSEL CONSUMPTION COMPUTATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

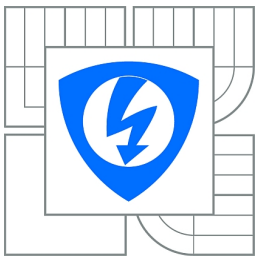
AUTOR PRÁCE — Bc. LUKÁŠ DRBOHLAV

AUTHOR

VEDOUCÍ PRÁCE — Ing. ZBYNĚK FEDRA, Ph.D.

SUPERVISOR

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student:	Bc. Lukáš Drbohlav	***ID:***	109646
Ročník:	2	***Akademický rok:***	2011/2012

NÁZEV TÉMATU:

Vyhodnocení spotřeby osobní lodi

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s potřebným programovým vybavením pro vývoj algoritmů pro řídící systémy SIMANTIC S7-200. Otestujte možnosti a navrhněte blokové schéma výpočtu pro odhad spotřeby lodi. Zvažte vlastnosti dostupných údajů, jejich přesnost, interval obnovování a případnou potřebu rozšíření o nové měřené parametry (měření rychlosti a směru větru).
Otestujte vámi navržený algoritmus a navrhněte možnost integrace výsledků do ovládacího panelu. Zaměřte se i na propojení odhadu s předpokládanou trasou.
Integrujte celý návrh do existujícího HW a otestujte jeho přesnost a další možnosti.

DOPORUČENÁ LITERATURA:

[1] SIMANTIC Programmable Controller System Manual. Siemens, 2005 - [cit. 31.1.2011]. Dostupné na: http://www1.siemens.cz/ad/current/file.php?fh=26930450fe&aid=729951

Termín zadání:	6.2.2012	***Termín odevzdání:***	18.5.2012

Vedoucí práce: Ing. Zbyněk Fedra, Ph.D.

prof. Dr. Ing. Zbyněk Raida
Předseda oborové rady

UPOZORNĚNÍ:

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení:	Bc. Lukáš Drbohlav
Bytem:	Gagarinova 610, Hradec Králové, 500 03
Narozen/a (datum a místo):	1. října 1987 v Chlumci nad Cidlinou

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Technická 3058/10, Brno, 616 00
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika
(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

   ☐ disertační práce
   ☒ diplomová práce
   ☐ bakalářská práce
   ☐ jiná práce, jejíž druh je specifikován jako .....
   (dále jen VŠKP nebo dílo)

Název VŠKP:	Vyhodnocení spotřeby osobní lodi
Vedoucí/ školitel VŠKP:	Ing. Zbyněk Fedra, Ph.D.
Ústav:	Ústav radioelektroniky
Datum obhajoby VŠKP:	_____

   VŠKP odevzdal autor nabyvateli*:

   ☒ v tištěné formě – počet exemplářů: 2
   ☒ v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.

3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.

4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

* hodící se zaškrtněte

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizovaní výpisů, opisů a rozmnoženin.

2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.

3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti

   - ☒ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)

4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.

2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.

3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.

4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 18. května 2012

……..	……
Nabyvatel	Autor

ABSTRAKT

Tato diplomová práce se zabývá návrhem a realizací programu pro vyhodnocení spotřeby osobní lodi pomocí řídícího systému Siemens Simatic S7-200 a dotykového ovládacího panelu Weintek Easy View MT8150X. Součástí je teoretické shrnutí vlastností a parametrů řídících automatů a způsobu jejich programování. Práce obsahuje popis návrhu výpočetního algoritmu, blokového schématu zapojení a testovacího vizualizačního softwaru. Větší část této publikace je věnována podrobnému popisu programu automatu a jeho vizualizaci, především měření spotřeby, vzdálenosti, průměrné rychlosti a rychlosti a směru větru. V závěru je uveden postup při realizaci a měření na osobních lodích na Brněnské přehradě včetně vyhodnocení stažených dat.

KLÍČOVÁ SLOVA

PLC, SIMATIC, S7-200, STEP7, MicroWin, Easy View, EasyBuilder, MT8150X, spotřeba, vzdálenost, rychlost větru, směr větru

ABSTRACT

This thesis is focused on a design and an implementation of a power consumption computation for a passenger vessel using the industrial automation system Siemens Simatic S7-200 and the touch panel Weintek Easy View MT8150X. A part of the thesis represents a theoretical summary of features and parametres of the programmable logic controllers (PLC) and a way how they are programmed. The work includes a description of a computational algorithm, a block diagram of interconnections and a visualization software for testing. A greater part of this publication is devoted to a detailed decription of a program in the PLC and its visualization, especially the power consumption computation, a distance, an average speed and a wind speed and a direction. At the end, the thesis shows a procedure for an implementation and a measurement on the passenger ships at the Brno dam, including an evaluation of downloaded data.

KEYWORDS

PLC, SIMATIC, S7-200, STEP7, MicroWin, Easy View, EasyBuilder, MT8150X, consumption, distance, wind speed, wind direction

DRBOHLAV, Lukáš *VYHODNOCENÍ SPOTŘEBY OSOBNÍ LODI*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2012. 95 s. Diplomová práce. Vedoucí práce: Ing. Zbyněk Fedra, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma VYHODNOCENÍ SPOTŘEBY OSOBNÍ LODI jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 140/1961 Sb.

Brno ....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Zbyňku Fedrovi, Ph.D. za účinnou metodickou, pedagogickou pomoc a další cenné rady při zpracování mé diplomové práce.

Dále děkuji pracovníkům firmy RegulTech servis s.r.o., projektantu Janu Zahradníkovi a programátorovi Lukáši Mládkovi, DiS., za odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne .....

.....

(podpis autora)

Výzkum realizovaný v rámci této diplomové práce byl finančně podpořen projektem
CZ.1.07/2.3.00/20.0007 **Wireless Communication Teams**
operačního programu **Vzdělávání pro konkurenceschopnost**.

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Finanční podpora byla poskytnuta Evropským sociálním fondem
a státním rozpočtem České republiky.

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Zadání tohoto projektu bylo stanoveno firmou ReglуTech servis s.r.o., která se podílela elektronistalací a řídícím softwarem na konstrukci osobní lodě Lipsko. Jako doplněk zadání od Dopravního podniku města Brna, byl požadavek na vytvoření programu pro výpočet spotřeby lodi a jeho implementaci do stávajícího softwaru tak, aby bylo možné vyhodnocovat styl jízdy jednotlivých kapitánů.

Tato diplomová práce se zabývá návrhem a realizací programu pro vyhodnocení spotřeby osobní lodi nejprve na úrovni testovacího softwaru. Dále jeho implementací do stávajícího řídícího softwaru, vizualizací na ovládací panel lodi a v neposlední řadě také vyhodnocením naměřených a vypočítaných dat získaných z provozu lodi.

V úvodní části práce se nachází bližší seznámení s lodí Dopravního podniku města Brna, její konstrukcí, elektroinstalací, pohonem a zejména jejím řídícím systémem. Další část první kapitoly je věnována základním vlastnostem a funkci moderní automatizační sběrnice AS-Interface a také jejímu uplatnění na samotné lodi Lipsko. Pak už teoretická část této práce zachází přímo k popisu programovatelných automatů různých typů, jejich funkcí a rozdělení. Zejména je pak část věnována řídícímu systému firmy Siemens Simatic S7-200 s CPU 224XP. Dále pak následuje seznámení s programovacím prostředím automatů řady Simatic S7-200 STEP7/MicroWIN a s normou IEC EN 61131-3, která definuje programovací jazyky pro programovatelné automaty. V závěru první kapitoly se nachází popis použitého dotykového panelu Weintek Easy View MT8150X, který zajišťuje servisní a provozní informace pro posádku a přiblížení funkce vývojového nástroje pro vizualizaci EasyBuilder EB8000.

Druhá kapitola obsahuje podrobný popis jak softwaru pro vyhodnocení spotřeby, tak jeho vizualizace na ovládací panel. Začíná podrobným popisem blokového schématu zapojení jednotlivých periférií, které jsou k vyhodnocení spotřeby použity. Pokračuje návrhem algoritmu v podobě vývojového diagramu a úvodním představením testovacího vizualizačního softwaru. V podkapitole detekce lodi v kotvišti se nachází podrobný popis algoritmu pro správné určení zda se loď po zastavení nachází v kotvišti. Dále jsou zde umístěny části zabývající se popisem výpočtu spotřeby, vzdálenosti a rychlosti a směru větru. Závěr této kapitoly je věnován ukládání naměřených a vypočítaných dat na paměťové médium.

Poslední kapitola je vyčleněna pro vlastní realizaci a implementaci programu do hardwaru a vyhodnocení naměřených a vypočítaných údajů. První podkapitola ukazuje úpravy skutečně nahraného softwaru na loď oproti programu vytvořenému pro testování. Dále je představena vizualizace na ovládací panel a její funkce. A v neposlední řadě grafické znázornění naměřených hodnot.

V přílohách jsou pak uvedena makra vykonávající v displeji výpočet veličin použitých při vyhodnocení spotřeby, která svojí délkou přesahují jednu stranu A4. Dále

využitá paměťová místa v programovatelném automatu umístěném na lodi a aplikační příručka, která byla vytvořena na základě požadavku firmy Regultech servis s.r.o. při předávání softwaru Dopravnímu podniku města Brna.

1 OSOBNÍ LOĎ - LIPSKO

Jedná se o motorovou loď, která je určena pro lodní dopravu na Brněnské přehradě. Byla vyrobena podle zadání Dopravního podniku města Brna (DPmB) firmou JESKO CZ s.r.o. Pardubice a elektroinstalaci celé lodi zajistil RegulTech servis s.r.o. Hradec Králové.

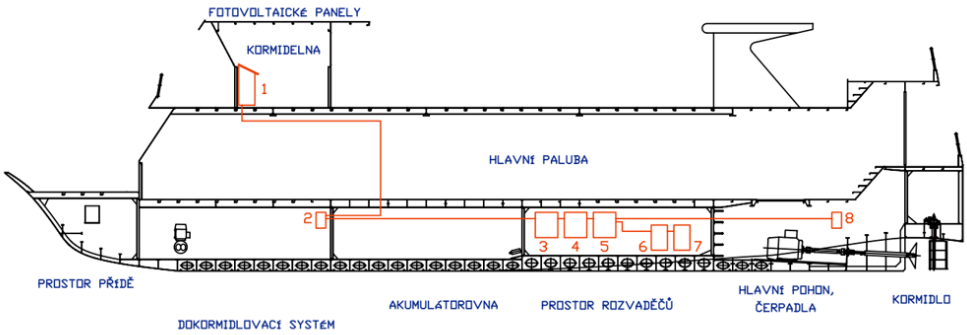
Loď má vyvýšenou příď a rovný tvar zádě. Spodní paluba je určena pro přepravu cestujících, kterých může být až 200. Je uzavřená a nacházejí se zde prostory pro stolky a v zadní části potom schody na horní palubu. Tato paluba je tzv. sluneční a slouží jako vyhlídková. V přední části horní paluby je umístěna kormidelna a v zadní části maketa komínu, která je užívána, jako úložný prostor. Podpalubí je rozděleno pěti přepážkami na šest vodotěsných komor, které jsou přístupné vodotěsně uzavíratelnými poklopy. Loď je dlouhá 25 m, široká 6,2 m a vysoká 6,65 m. Její ponor může být až 1,15 m. Pohyb lodi je zajištěn bronzovou čtyřlistou vrtulí o průměru 686 mm, díky které může loď dosahovat rychlosti až 13,5 km/h. Loď je zobrazena na fotografii na obr. 1.1. Obecný popis lodi dále naleznete v [1].

Obr. 1.1: Osobní loď - Lipsko [2]

Zdrojem elektrické energie pro celou loď jsou dva typy akumulátorových baterií, trakční a spotřebitelská, které jsou umístěny v podpalubí. Akumulátorové trakční baterie, s parametry 1120 Ah/300 V DC, jsou zdrojem pro hlavní pohon lodi, tedy motor o výkonu 55 kW, dvě požární čerpadla a drenážní čerpadlo. Akumulátorová spotřebitelská baterie, 1120 Ah/24 V DC, napájí ostatní systémy, kterými jsou systém dokormidlování, osvětlení, větrání, klimatizace, řídící systém a systém měření a regulace. Dále jsou na střeše kormidelny instalovány fotovoltaické panely s výkonem 1,1 kWe sloužící, jako doplňkový zdroj elektrické energie. Hlavním pohonem lodi je asynchronní motor od firmy Siemens 3×230 V, 55 kW. Ten je přes spojku

propojen s hřídelí bronzového lodního šroubu. Řízení asynchronního motoru je zajištěno pomocí měniče frekvence, který je napájen z trakční baterie a vytváří třífázové střídavé napětí. Právě použitím takto řízeného asynchronního motoru je odebíraný proud rovnoměrně rozložen na celou baterii a tím značně stoupá její životnost.

Řízení pomocných okruhů a monitorování provozu lodě je zabezpečeno systémem Simatic S7-200 od firmy Siemens. Je zde použit decentralizovaný sběr dat, který využívá průmyslovou sběrnici AS-i. Periférie pro tuto sběrnici dodala firma IFM electronic. Na obr. 1.2 je naznačeno, jak jsou jednotlivé periférie připojeny ke sběrnici. Řídící systém Simatic S7-200 včetně 15" dotykového displeje firmy Weintek MT8150X je součástí řídícího panelu přímo v kormidelně (1). Sběrnice vede z kormidelny do podpalubí stromovou topologií a je na ni připojena skříň sběru dat a signalizace (2), rozváděče 300 V DC (3), 24 V DC (4) a 230 V AC (5). Následuje rozváděč měniče frekvence hlavního pohonu (6) a náhradního měniče frekvence (7). Poslední je skříň pro sběr dat a signalizaci od bronzového lodního šroubu.



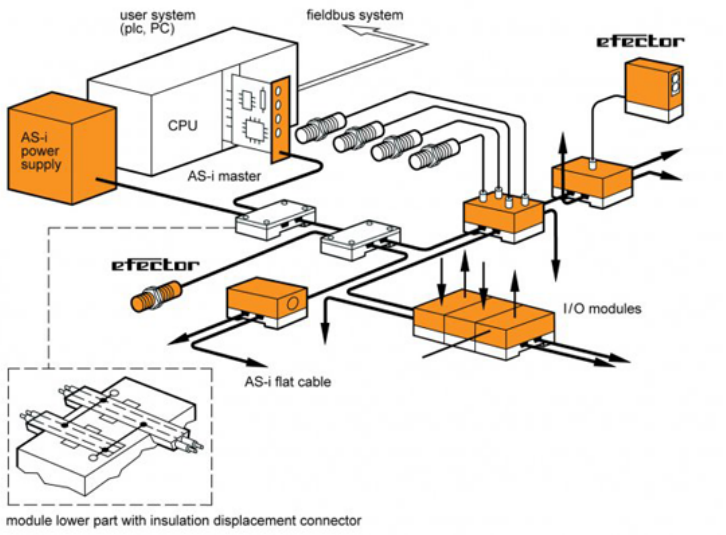
Obr. 1.2: Řez lodí Lipsko se schématem vedení sběrnice AS-interface [2]

Více informací lze nalézt v [2], ze kterého bylo čerpáno pro tuto podkapitolu.

1.1 AS-Interface

Tato sběrnicová technologie je v praxi užívaná od roku 1994 a tvoří inteligentní řešení pro nejnižší úroveň v provozu automatizace. Původní paralelní spojení každého senzoru a akčního členu s řídící jednotkou bylo nahrazeno jedním nekrouceným nestíněným dvouvodičovým kabelem žluté barvy, který je mimo jiné i obchodní značkou AS-i. Po něm jsou zařízení napájena, ale dostávají i datový signál. AS-i je koncipována, jako sériová komunikace Master-Slave s cyklickým přenosem dat. K připojení dalších modulů do AS-i sítě se používá prožezávací technika "Click & Go". Ta spočívá ve vložení kabelu mezi horní a spodní část zařízení. Po přitlačení obou částí

dojde k prořezání kabelu a tím ke galvanickému spojení. Díky této metodě připojení jednotlivých modulů do sítě je možné využít několik topologií. Jsou to sběrnicová, hvězdicová, kruhová a stromová topologie, která je naznačena na obr. 1.3. Pro sítě AS-i je možné vybírat z obrovského sortimentu od více než 260 výrobců na světě, lze tedy řešit široké spektrum automatizačních úloh. Další informace je možné nalézt na internetových stránkách Sdružení AS-interface Česká republika [3].



Obr. 1.3: Schéma AS-i systému [4]

Master automaticky rozpozná připojená zařízení slave na sběrnici a přidělí jim adresu, která je uložená v non-volatile paměti. Není tedy nutná konfigurace sítě. Cyklicky volá zařízení slave a v přesně daných okamžicích se dotazuje na jejich data. Dává také k dispozici data senzorů a akčních členů na rozhraní nadřazeného systému. Zařízení typu slave přijímá rozpoznané datové bity od mastera a zpět posílá vlastní data. Přenos mezi master-slave a naopak je zabezpečen kódem Manchester, ale také opakováním telegramu v případě chyby při přenosu do té doby, než je telegram v pořádku přijat. Master tak může detekovat chybu periférie, vlivem zkratů, přetížení apod., a určit přesně její polohu.

Napájecí napětí celé sítě je 29,5 až 31,6 V, to zaručuje připojený spínaný DC zdroj symetrického napětí. Odtud je odebíráno napětí pro modulaci přenášených dat i napájecí napětí pro zařízení slave, které je 24 V. Proudový odběr celé sítě pak může být maximálně 8 A, tedy 100 nebo 200 mA na jedno zařízení slave (tato hodnota je dána standardem AS-i). Maximální počet připojených slave zařízení je také přesně vymezen standardem. Pro standard 1.0 je to 31 zařízení slave a pro standard 2.1 až 62 zařízení slave.

Mezi hlavní výhody patří možnosti přímého připojení modulů pro chod asynchronních motorů a detekce poloh ventilů. Další významné výhody spočívají v úspoře 20 až 40% nákladů a času při instalaci sítě. Výrobky pro standard AS-i jsou certifikovány, z toho také plyne vyšší spolehlivost a životnost těchto součástí. Jistě nezanedbatelná je i možnost kombinace s jinými systémy. Mezi omezení sítě AS-i patří maximální délka vedení 100 m bez použití repetarů. AS-i nemá prostředky pro přenos dat řízený událostmi ani komunikaci mezi slave zařízeními navzájem. Také množství dat přenášených pomocí AS-i je omezeno. Lze komunikovat 4 bity se slavem v jednom cyklu, delší zprávy jsou rozděleny do cyklů následujích. Toto vyhovuje pomalu se měnícím veličinám, ale síť je nepoužitelná pro vysoce dynamické regulační obvody. Další informace lze nalézt v článku [4], nebo podrobněji v literatuře [5].

1.1.1 AS-i na osobní lodi Lipsko

Pro sběr provozních veličin a ovládání lodě byla vybrána z několika důvodů. Díky podstatnému zmenšení požadavků na kabeláž, mohlo dojít ke zmenšení prostupů mezi jednotlivými vodotěsnými komorami v podpalubí, tento fakt má ale příznivý vliv i na celkovou cenu kabeláže. Zjednodušené propojení přístrojů je patrné z rozváděče na obr. 1.4. Dále pak byla využita vysoká odolnost proti rušení a velmi vysoká provozní spolehlivost.

Obr. 1.4: Zjednodušení zapojení přístrojů lodi s použitím AS-i [2]

Na osobní lodi Lipsko zajišťuje sběrnice AS-i signalizaci provozních a poruchových stavů hlavního i záložního frekvenčního měniče, signalizaci frekvenčních měničů

požárních čerpadel a drenážního čerpadla, signalizaci izolačních stavů a indikaci poloh spínacích a jistících prvků v silnoproudých rozváděčích. Sběr dat o proudech a napětích v jednotlivých soustavách, o teplotách hlavního pohonu a prostorech v podpalubí, signalizaci zaplavení vodotěsných komor, přenos informace o hladinách pitné, užitkové vody a fekálií. A v neposlední řadě ovládání klimatizace, houkaček, ventilátorů a čerpadel pitné a užitkové vody. Další podrobnosti lze nalézt v odborném článku [2].

1.2 Programovatelné automaty

Programmable Logic Controller – programovatelný logický automat (PLC) je uživatelsky programovatelný řídící systém přizpůsobený pro řízení průmyslových a technologických procesů, mnohdy specializovaných na úlohy převážně logického typu. Použití v tvrdých průmyslových podmínkách je dáno i robusní konstrukcí a velkou odolností proti rušení. Tato zařízení nahrazují již více než 40 let bezkontaktní a reléovou logiku. Realizují tedy program, který vykonává logické rovnice, na místo fyzického propojení logických členů tak, jak tomu bylo u reléové logiky. Mezi nesporné výhody programovatelných automatů patří rychlé přeprogramování úlohy, modularita, vestavěná diagnostika vlastního PLC, jednoduché programovací jazyky a jednoduchý a tím i spolehlivý OS reálného času. U nových programovatelných automatů je možnost použít vyšší programovací jazyky. Jejich možné využití je téměř ve všech oborech lidské činnosti, například ve strojírenství, energetice, ekologii, ale i zemědělství, potravinářství, chemii, farmacii a mnohých dalších.

Programovatelné automaty se skládají z CPU, operační paměti, která je společná pro vstupní data, výstupní data, vnitřní proměnné a paměťový prostor pro vlastní program. K těmto částem je připojeno rozhraní se vstupy a výstupy. Součástí základního programového vybavení každého PLC jsou běžně funkce časovače a funkce čítání impulzů.

Základní rozdíl oproti sekvenčně vykonávaným programům je u PLC v tom, že zde je hlavní program, tzv. OB1, vykonáván cyklicky, tedy neustále ve smyčce, viz. obr. 1.5. PLC po skončení výkonu poslední instrukce uživatelského programu spustí systémový program, který zajišťuje otočku cyklu. Během této otočky dojde k aktualizaci vstupů a výstupů a vykonání režijních úkonů, což může být například aktualizace časových údajů, ošetření komunikace apod. Po skončení této části smyčky se načte opět první instrukce uživatelského programu. Vedle cyklického režimu mají současné PLC i režim přerušení, který může být parametrován, takže kritické akce mohou být obslouženy i mimo cyklus PLC. Doba cyklu programovatelného automatu je zpravidla definována jako doba, kterou PLC potřebuje k načtení

dat, vysílání dat na výstupy a zpracování 1000 instrukcí. Tato doba se pohybuje řádově v jednotkách ms, pro PLC s velmi rychlými CPU pak hluboko pod 1 ms.

Obr. 1.5: Cyklus programu programovatelného automatu [6]

Další poměrně významný rozdíl je ten, že PLC pracuje pouze s obrazy vstupů a výstupu, nikoli s aktuálními hodnotami. Tyto obrazy jsou přepisovány vždy v otočce programu, jak bylo popsáno výše. Hlavní důvod tohoto způsobu přístupu ke vstupním a výstupním hodnotám je synchronizace změn vstupních a výstupních hodnot, tedy omezení časových kolizí a tím výskytu hazardů. Podrobné informace lze nalézt v literatuře [6] a [7].

Základní dělení automatů:

- **Mikro PLC** – Jsou to nejmenší a nejlevnější kompaktní PLC systémy. Mají pevný počet vstupů a výstupů a nelze je dále rozšiřovat. Vyznačují se omezeným programováním a komunikačními schopnostmi.
- **Kompaktní PLC** – Mají určitou, ale omezenou variabilitu konfigurace, lze k nim tedy připojit jen omezený počet rozšiřujících modulů. Do této skupiny například patří systém řady Simatic S7-200, ačkoli ho sama firma Siemens řadí do kategorie mikro PLC, ale podle zde patrného rozdělení automatů se tento PLC řadí mezi kompaktní.
- **Modulární PLC** – U automatů tohoto typu je již velká volnost v konfiguraci vstupů a výstupů, ale i velký sortiment přídavných modulů. Tyto moduly rozšiřují působnost PLC až za hranice jeho původního určení. Mezi rozšiřující moduly patří moduly diagnostiky, vizualizační moduly pro sledování procesu na připojených obrazovkách, moduly pro měření teploty, pro regulaci teploty. V neposlední řadě jsou to moduly komunikační, pomocí těchto modulů jsou automaty propojovány mezi sebou a s vyššími úrovněmi řízení po průmyslových sběrnicích. K takovýmto automatům se řadí Simatic S7-400.

1.2.1 Simatic S7-200

Je to řada malých PLC od firmy Siemens, které pokrývají široké spektrum automatizačních úloh. Stávají se jakýmsi mezikrokem mezi logickými moduly pro jednoduché aplikace LOGO! a výkonnými programovatelnými automaty S7-300 nebo S7-400. Z rodiny S7 je S7-200 nejmenší a nejméně výkonný programovatelný automat. Disponuje ovšem jednoduchou strukturou, snadným programováním a hlavně mezi obdobnými systémy i nízkou cenou. Vysokou rychlost zpracování také dokumentuje fakt, že jednu booleovskou instrukci zpracovávají automaty této řady za 0,22 µs. Jeho použití se pohybuje okolo méně náročných aplikací, jako jsou řízení výtahů, dopravníků apod. Pro použití tohoto automatu, jako řídícího systému osobní lodi Lipsko mluvil i fakt, že tato řada automatů je certifikována i pro použití v lodní dopravě. Obsahuje mikroprocesor, integrovaný napájecí zdroj, vstupní a výstupní obvody. Celý automat je uložen v kompaktním mechanicky odolném robustním pouzdru, což je pochopitelné s ohledem na použití v průmyslových aplikacích.

Samotné PLC této řady jsou vybavena výkonnou instrukční sadou, která umožňuje vykonávat logické operace, matematické operace s celými i reálnými čísly a systémem časových přerušení a přerušení od událostí. Mezi další integrované možnosti patří čítače impulzů, časovače, vysokorychlostní časovače pro použití v aplikacích velmi rychlého časování a v neposlední řadě impulzní vstup.

Všechny automaty této řady lze doplnit množstvím rozšiřujících modulů. Ke každému PLC jich může být najednou připojeno až sedm. Tato hodnota však závisí na energetické bilanci dané aplikace a v případě výkonově náročnějších aplikací může klesnout. Návod na výpočet energetické bilance je uveden v příloze datasheetu [8].

Rozšiřující moduly:

- **I/O modul** – Dělí se na analogové a digitální. Při jejich výběru se specifikuje počet vstupů a výstupů, případně zda mají být reléové, či normální, AC nebo DC.
- **Ethernetový modul** – Tento modul se využívá k připojení HMI. Další může být využití internetových funkcí, jako je WWW, FTP, e-mail.
- **Polohovací modul** – Přídavný modul pro řízení rychlosti a polohy v otevřené smyčce pro krokové motory nebo servomotory.
- **Modul pevné telefonní linky** – Přímé připojení automatu k analogové telefonní lince.
- **PROFIBUS-DP Slave**
- **AS-Interface master**
- **PPI/MPI slave** – Tento modul je využíván pro komunikaci s Simaticem vyšší řady, například S7-300 nebo S7-400.

- **Modul RS-485**
- **Modul GSM, GPRS**

1.2.2 S7-200 s CPU 224XP

Pro řízení osobní lodi Lipsko je využíván právě tento typ programovatelného automatu řady Simatic, který vychází z CPU 224 a snoubí všechny oblíbené vlastnosti předcházejících automatů této řady. Že jde o skutečně malé PLC je patrné z rozměrů 140×80×62 mm a jeho hmotnosti pouhých 390 g. Tento typ automatu má integrováno 14 vstupů a 10 výstupů digitálních a 2 vstupy a 1 výstup analogový. Pouzdro tohoto PLC má po obou stranách svorkovnice, na jedné jsou vstupní piny a na druhé výstupní. Tyto svorkovnice jsou plně odnímatelné. Pro rychlou orientaci signálů na vstupech a výstupech je nad každou svorkovnicí panel LED diod. Na pravé straně je umístěno odklápěcí víčko, pod kterým se nachází ruční přepínání režimů programovatelného automatu, analogové potenciometry a konektor pro rozšiřující moduly. Na levé straně uprostřed pouzdra se nacházejí stavové diody, z těch je ihned patrný stav automatu. Pod nimi je umístěn kryt pro konektor na časový modul, modul s bateriemi nebo paměťový modul. Vlevo nahoře je analogová svorkovnice s příslušnými vstupy a výstupy. A konečně v levém dolním rohu jsou umístěny dva konektory pro sériovou komunikaci, RS-485, které mohou být nastaveny na různé komunikační protokoly. Je také možné přes tato rozhraní automat naprogramovat, případně komunikovat s displejem. Umístění automatu přímo v kormidelně lodi je možné vidět na obr. 1.6. Tento automat pracuje s napájecím napětím 24 V DC.

Obr. 1.6: PLC firmy Siemens S7-224XP v kormidelně lodi

Automat s tímto procesorem paměť pro program o velikosti 16 kB (s editací v režimu RUN 12 kB) a paměť pro ukládání dat je široká 10 kB. Zálohování dat je stan-

dardně 100 h, případně typicky 200 dní s použitím přídavného modulu. Lze v něm nastavit až 256 čítačů nebo časovačů, takže je opravdu využitelný i pro náročnější aplikace.

1.2.3 Paměťový prostor S7-200

Data uložená v Simatic S7-200 jsou rozdělena do několika paměťových typů, podle toho zda se jedná o obrazy vstupů, obrazy výstupů, datovou paměť atd.

Typy paměťových oblastí:

- **Registr obrazů vstupů (I):** Automat na začátku každého programového cyklu přečte hodnoty na jeho vstupech, uloží je do tohoto registru a poté jsou využívány v běžícím programu.
- **Registr obrazů výstupů (Q):** Má stejnou funkci jako předchozí registr, ovšem pro výstupy programovatelného automatu.
- **Oblast proměnné paměti (V):** Slouží pro ukládání mezivýsledků operací prováděných řídící logikou programu, nebo pro ukládání dat souvisejících s výkonem programu.
- **Oblast bitové paměti (M):** Využívá se pro ukládání informací o stavech řídících kontaktů.
- **Oblast paměti časovačů (T):** Zde jsou uloženy informace o jednotlivých časovačích, tedy aktuální hodnota časovače a bit časovače, který je v logické jedničce při splnění podmínky kladené na časovač. Záleží tedy na způsobu přístupu, protože aktuální hodnota časovače je uložena v 16 bitové proměnné Word, kdežto bit časovače je jen jediný bit.
- **Oblast paměti čítačů (C):** Tato část paměti má stejnou funkci jako předchozí oblast paměti časovačů.
- **Oblast lokální paměti (L):** Slouží jako zápisníková paměť pro předávání parametrů podprogramům, ukládání ukazatelů a podobně.

Ke všem těmto paměťovým oblastem lze přistupovat těmito způsoby:

- **Přístup k jednotlivým bitům:** [Paměťový typ][adresa bajtu].[adresa bitu]
- **Přístup k blokům dat:** [Paměťový typ][velikost][adresa počátečního bajtu]

Například pro přístup k prvnímu bitu v oblasti bitové paměti je použit zápis M0.1. Nebo například zápis VD4440 říká, že se jedná o přístup k proměnné paměti, bloku dat o velikosti Double Word s počátečním bajtem 4440.

Dále jsou zde méně používané typy paměťových prostor. Vysokorychlostní čítače (HC), Akumulátory (AC) a paměti analogových vstupů. K těm se přistupuje pomocí pořadového čísla. Velmi významnou částí jsou pak Speciální paměťové prostory

(SM), ve kterých je od výrobce předprogramována určitá funkce. Jsou to v podstatě systémové příznakové bity. Více informací o paměťových oblastech lze nalézt v [8].

1.2.4 Norma IEC EN 61131-3

Známá je také pod českým názvem **Programovatelné řídící jednotky – část 3: Programovací jazyky**. Tato norma si klade za cíl sjednotit programovací jazyky různých výrobců PLC, jak jejich syntaxi tak sémantiku pro programování PLC. Součástí normy je i definice společných prvků, jako jsou například deklarace proměnných, datových typů, funkcí a funkčních bloků. Jazyky podle normy jsou systematičtěji a obecněji definovány a to usnadňuje přechody mezi jednotlivými jazyky.

Definované programovací jazyky:

- **Jazyk IL** – Každé instrukci systému odpovídá stejně pojmenovaný příkaz. Tento jazyk využívá typických assemblerovských struktur, jako jsou návěští pro realizaci skoků, symbolická jména pro číselné hodnoty, pojmenování vstupních, výstupních proměnných i vnitřních proměnných, a mnohé další.
- **Jazyk LD** – Je to grafický jazyk vycházející z původní realizace logických funkcí reléovou a kontaktní logikou. Obsahuje například spínací kontakt, jako dvojici svislých čárek, rozpínací kontakt je naopak rozdělen lomítkem uprostřed apod. Složitější funkční bloky, jako jsou čítače, časovače a další jsou znázorněny obdélníkem s popisem uvnitř. Tento jazyk je vhodnější pro starší programátory, kteří byli zvyklí na používání reléové logiky a používá se pro jednodušší funkce, ty složitější se vyplatí programovat ve vyšších jazycích.
- **Jazyk FB** – Také se jedná o grafický jazyk. Pracuje s obdélníkovými bloky, které přizpůsobují svou velikost počtu vstupů. Základními kameny jsou jistě bloky logických funkcí AND a OR, které nahrazují spínací rozhodování reléové logiky. Složitější bloky jsou opět realizovány, jako u jazyka LD.
- **Jazyk ST** – Tento jazyk je obdobou vyšších programovacích jazyků z PC techniky, jako je Pascal nebo C. Umožňuje názorný zápis algoritmů, používání různých funkcí realizujících cykly a podmínky. Je oblíbený zejména mladými programátory. I přesto se části kódů píší v jazyce IL nebo LD.
- **Jazyk SFC** – Tvoří nadstavbu nad výše popsanými jazyky. Umožňuje stavový popis sekvenčních úloh, pomocí přechodového grafu konečných automatů a určité třídy Petriho sítí. Chování v jednotlivých stavech lze obvykle popsat i některým z předchozích jazyků.

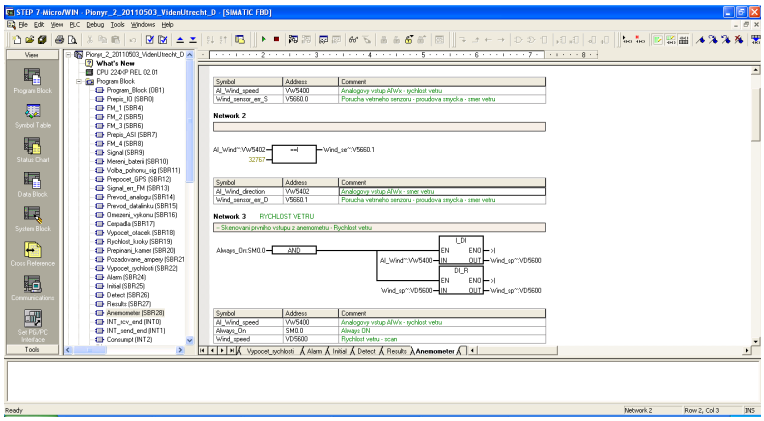
Norma tedy nezaručuje bezchybné efektivní programování, ale přenositelnost kódu mezi jednotlivými zařízeními a platformami. Více o této normě v článku [9]. Přesto

má každá firma zabývající se výrobou a distribucí PLC jazyky své s drobnými odlišnostmi od jazyků normovaných. Ty jsou určeny pouze pro automat dané společnosti.

1.2.5 Vývojové prostředí STEP7/MicroWIN

Jedná se o vývojové prostředí pro vytváření řídících programů k programovatelným automatům firmy Siemens řady S7-200. Mezi největší výhody tohoto softwaru patří možnost jednoduchého programování, přepínání mezi třemi různými programovacími jazyky, podpora programování podle normy IEC EN 61131-3 a řada pomocných průvodců, které usnadňují řešit komplikovanější problémy.

Plocha v prostředí STEP7 je rozdělena na čtyři hlavní části: hlavní okno, boční okno s rychlými volbami, horní lišta rychlého spuštění a dole umístěný stavový řádek, jak je patrné z obr. 1.7. V hlavním okně je zobrazena posloupnost instrukcí tvořící program ve zvoleném jazyce. Boční okno umožňuje rychlý výběr instrukce a zobrazení dalších parametrů programu, případně nastavení komunikace automatu s počítačem. V tomto okně je umístěna tabulka symbolů, ve které se nacházejí proměnné a jejich symbolické názvy. Dále pak tabulka stavů, která slouží programátorovi pro odladění správné funkce aplikace. V horní liště rychlého spouštění se nacházejí ikony pro download programu, upload programu, zapnutí ladícího módu apod. A konečně na spodní stavový řádek jsou vypisovány důležité informace při stažení programu do automatu, případně chyby nebo upozornění.



Obr. 1.7: Rozložení obrazovky ve vývojovém prosstředí STEP7/MicroWIN

Program je rozdělen na samostatné části, tzv. network. V každém network může být jen omezený počet vstupů, logických bloků a výstupů. Toto dělení umožňuje rychlejší a efektivnější programování, protože zajišťuje velmi dobrou přehlednost programu. V rámci jednoho network jsou instrukce vykonány nejprve v prvním řádku

a až potom v řádku následujícím. To je velmi důležité pro správnou představu funkce napsaného kódu.

1.3 Dotykový displej Weintek Easy View MT8150X

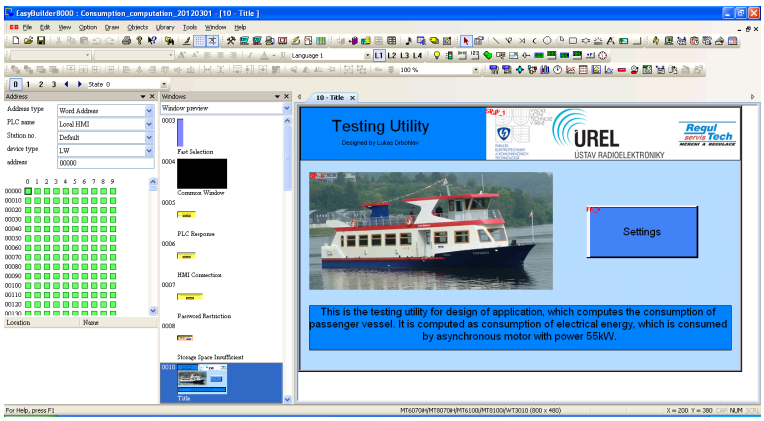
V kormidelně lodi je umístěn 15" dotykový displej od firmy Weintek, který obstarává vizualizaci programu pro řízení lodi a také podprogramu, pro vyhodnocení spotřeby. Umístění displeje v kormidelně je patrné z obr 1.8. Tento panel byl vybrán, protože podporuje přímou komunikaci s řadou PLC a tak není nutné v programu automatu komunikaci speciálně ošetřovat. Další nespornou výhodou oproti konkurenčním panelům je jeho nízká cena a výborné vývojové prostředí EasyBuilder 8000, které má volně šiřitelnou licenci.

Obr. 1.8: Umístění panelu Weintek MT8150X v kormidelně lodi

Displej pracuje v rozlišení 1024x768 a zobrazuje až 65536 barev. Panel obsahuje procesor x86 s taktovacím kmitočtem 500 MHz a operační pamětí DRAM 256 MB. Displej obsahuje množství komunikačních rozhraní ke komunikace s řídícími automaty, nadřazenými systémy apod. Základní komunikační rozhraní jsou tři sériové linky, které jsou programovatelné na různé dva základní typy komunikace: RS-232 a RS-485, ale i na jiné protokoly. Dalším komunikačním nástrojem je rozhraní Ethernet (10/100 Base-T). Panel dále disponuje video vstupním portem, pro zpracování NTSC nebo PAL, audio vstupem a výstupem, dvěma USB host porty verze 2.0 a slotem pro SDHC/SD kartu. Více informací o tomto displeji lze získat na [10].

1.3.1 Vývojové prostředí EasyBuilder EB8000

Všechny displeje od firmy Weintek řady 8000 se programují pomocí vývojového prostředí EasyBuilder EB8000. Velkou výhodou tohoto softwaru je jeho freeware licence. Další předností je pak intuitivní vytváření samotné vizualizace řídícího systému. Program přímo podporuje řadu PLC, například od firem Siemens, Mitsubishi, Allen-Bradley, Danfoss, Hitachi a dalších. Protože pro velkou řadu v současnosti dostupných automatů obsahuje EasyBuilder komunikační knihovny, je spojení takového PLC s displejem prakticky velmi jednoduchá záležitost. Rozložení pracovní plochy je pak zobrazeno na obr 1.9.



Obr. 1.9: Vývojové prostředí Weintek Easy Builder 8000

Rozložení plochy tohoto vývojového prostředí obsahuje od verze 4.50 čtyři základní části. Jedná se o lišty rychlého spouštění, zobrazení využití paměťového prostoru displeje a automatu, přehled vytvořených oken pro zobrazení na displeji a nakonec pracovní plochu, na které se vytváří a upravuje vzhled jednotlivých oken displeje.

Lišty rychlého spouštění jsou velmi užitečné, protože většina důležitých a uživatelem využívaných bloků a nastavení se nachází právě zde. Nejdůležitější ikony rychlého spouštění pro vytváření vizualizace jsou pak kompilace a download programu do displeje. Dále jsou zde také ikony pro vkládání různých zobrazovacích oken, tlačítek, textu, grafů, chybových hlášení a mnoho dalších. V neposlední řadě jsou zde obsažena i nastavení pro vzorkování a uchování dat, která jsou následně použita pro grafické zobrazení nebo uložení na externí paměťové médium, a možnost nastavení řízení běhu programu PLC tím, že jsou v přesně stanovených časových okamžicích vykonávána makra, která například zpracovávají složitější matematické operace, které automat nedokáže s potřebnou přesností provést.

V části, která zobrazuje využití paměťových míst v displeji a v automatu lze nastavit typ paměťového prostoru, který je potřeba vyčíst a přesnou adresu bitu, nebo slova. Potom jsou v dolním segmentu tohoto zobrazení slovní informace o obsahu této sekce paměti.

Další část obrazovky ve vývojovém prostředí slouží přehledu již vytvořených oken, vyskakovacích oken, klávesnic či kalkulaček, které budou do displeje nahrány.

Vzhled aktuálně vybraného okna a možnost jeho úpravy je dominantní částí celého prostředí. Zde si může každý uživatel upravovat grafickou část přesně podle účelu a dané aplikace. V tomto okně se také nastavují jednotlivé vstupní a výstupní parametry zobrazovacích oken, tlačítek, grafů apod.

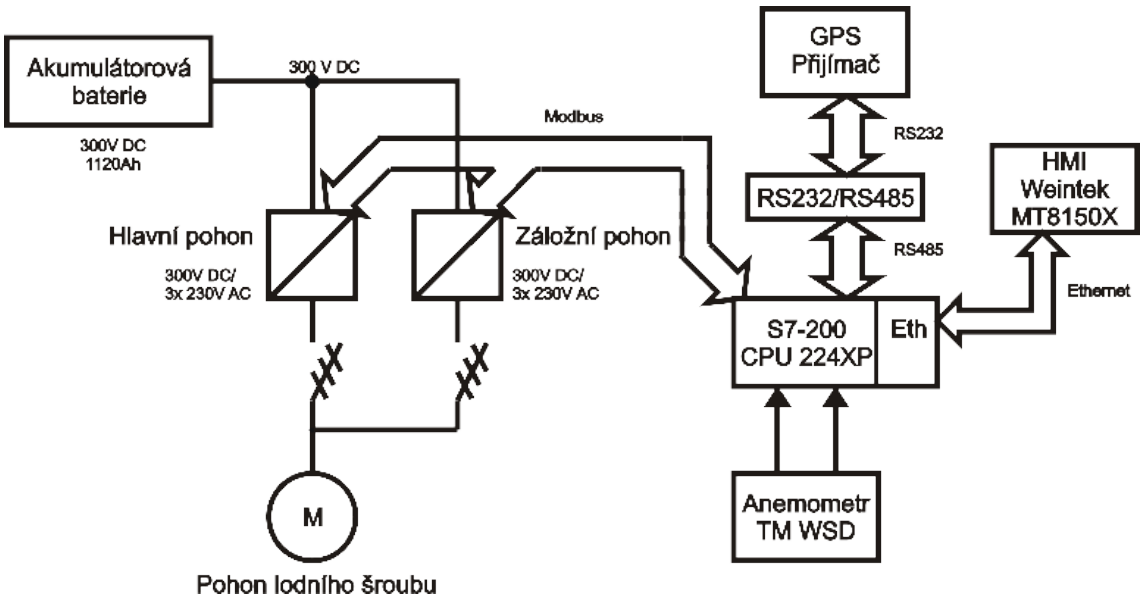
Kompletní manuál popisuje vše od vytvoření prvního vizualizačního programu pro displej, přes připojení automatu, podrobný popis všech prvků, programovací jazyk maker a mnoho dalšího až po diagnostiku a řešení chybových stavů a hlášení. Tento manuál je možné nalézt na [11].

2 PROGRAM PRO VÝPOČET SPOTŘEBY

V této kapitole dochází k podrobnému popisu naprogramovaného softwaru pro výpočet spotřeby lodi a také jeho vizualizace na dotykový ovládací panel. Na začátku je uvedeno schéma zapojení znázorňující sběr dat, která jsou následně vyhodnocována v automatu. A ještě před částí věnované programu je zařazena podkapitola zabývající se teoretickým návrhem programu, který předcházel vlastnímu programování a také úvodní informace k testovacímu vizualizačnímu softwaru, který sloužil k odladění hlavních funkcí programu.

2.1 Schéma zapojení

Všechna zařízení, která jsou využívána softwarem pro vyhodnocení spotřeby osobní lodi jsou zobrazena na obr. 2.1. Základem celého blokového schématu je programovatelný automat Simatic S7-200 s CPU 224XP, do kterého jsou pomocí různých sběrnic a protokolů přenášena provozní data lodi. PLC je rozšířeno o dva přídavné moduly. Jedná se o modul pro sběrnici AS-interface a ethernetový modul. AS-i však slouží pouze pro sběr provozních veličin, jako jsou například stavy frekvenčních měničů, poloha kormidla, napětí baterií a podobně.



Obr. 2.1: Blokové schéma zapojení pro výpočet spotřeby

Z obrázku je patrné, že je trakční baterie, s napětím 300 V DC a 1120 Ah, připojena přes frekvenční měniče k asynchronnímu motoru, který pohání lodní šroub.

Frekvenční měniče zajišťují přeměnu stejnosměrného napětí baterie na třífázové střídavé napětí $3\times230\,\mathrm{V}$, které je přiváděno na motor s výkonem $55\,\mathrm{kW}$. Zdvojení frekvenčních měničů pak zajišťuje hlavní, případně náhradní pohon lodi v momentě, kdy nastane porucha na jednom z měničů.

Frekvenční měniče dodala firma Rockwell Automation, přesněji se jedná o měniče Allen-Bradley PowerFlex 700AC, viz. [12]. K nim byla přikoupena komunikační karta od téže firmy, více v [13]. Komunikace mezi měniči a automatem probíhá po sběrnici RS-485, ale přes komunikační protokol Modbus, který je podrobně popsán v [14]. Tento protokol je založen na komunikaci master-slave, takže automat vysílá dotaz na měnič s určitou adresou a ten odpovídá. Z každého měniče automat postupně vyčítá osm hodnot a po vyčtení těchto hodnot následuje vyčítání hodnot dalšího měniče. Ty jsou na lodi celkem čtyři, pak se celý cyklus vyčítání dat opakuje. Vyčítány jsou hodnoty napětí na svorkách měničů, výstupní proud, teplota atd.

GPS přijímač je připojen k PLC přes převodník sběrnice RS-232 na RS-485, která je již připojena na konektor řídícího systému. Komunikace se provádí tak, že přijímač vysílá na sběrnici stále dokola zprávy, které jsou automatem zachytávány. Protože se přijímají pouze zprávy nesoucí informaci o aktuální poloze, ovládací program v PLC vybírá podle hlavičky pouze pakety, které tuto informaci obsahují a následně je dekóduje. Data jsou přenášena protokolem NMEA, který je používán v GPS komunikaci, podrobnější popis tohoto protokolu lze nalézt v referenčním manuálu [15].

Doposud uvedená komunikace a získávání provozních dat z jednotlivých zařízení byla ošetřena v rámci hlavního řídícího softwaru lodi. Již upravená data jsou následně použita v aplikaci pro vyhodnocení spotřeby.

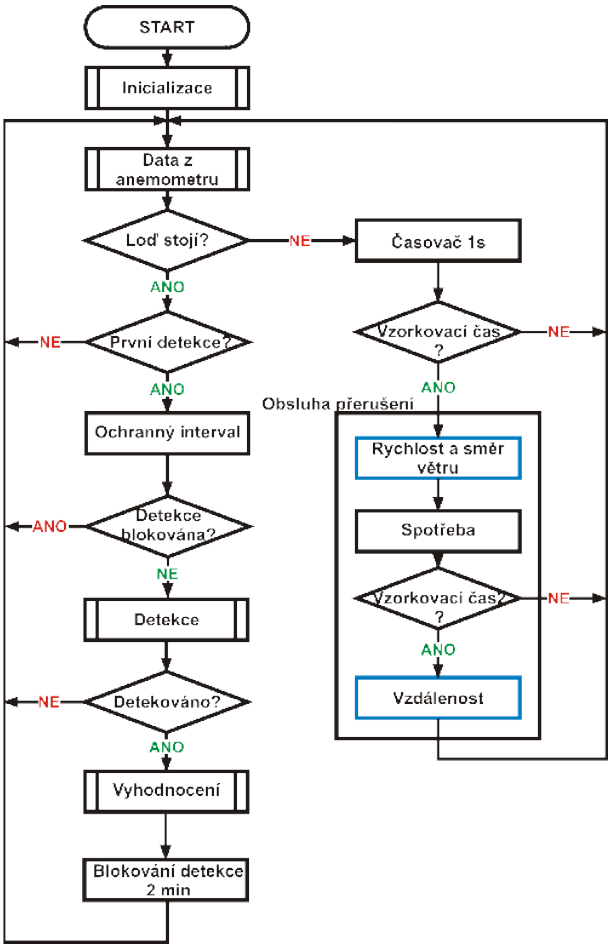
Údaje o větru budou získávány z čidla rychlosti a směru větru. **Anemometr** TM WSD bude připojen k dvojici analogových vstupů programovatelného automatu. Obě snímané veličiny jsou přenášeny jako proud v rozmezí od $4\,\mathrm{mA}$ do $20\,\mathrm{mA}$. Proud přicházející na vstup PLC je převáděn v programu pro výpočet spotřeby na rychlost větru v rozsahu od $0\,\mathrm{m/s}$ do $30\,\mathrm{m/s}$ a směru větru v rozmezí od $0°$ do $350°$. Základní informace o anemometru lze nalézt na [16].

Poslední částí schématu zapojení je **zobrazovací panel** Weintek MT8150X, který zajišťuje vizualizaci řídícího softwaru a aplikace pro vyhodnocení spotřeby lodi. Předávání dat mezi displejem a automatem probíhá přes ethernet. Ve vývojovém prostředí pro programování vizualizace do panelu je možné nastavit z několika typů PLC všech předních výrobců a následná komunikace na nižší úrovni je ošetřena knihovnami displeje, takže ji není potřeba v programu speciálně definovat.

2.2 Návrh algoritmu programu

Zde uváděná část práce se zabývá obecným postupem programu při výpočtu spotřeby a dalších pomocných veličin.

Na obr. 2.2 je naznačen vývojový diagram reprezentující hlavní část algoritmu pro výpočet spotřeby. Ten byl vypracován před samotným programováním, protože značně usnadňuje vytváření zdrojového kódu. Tím, že byl program vytvářen v jazyce FB, je skutečné rozložení bloků programu obdobné, jako na vývojovém diagramu.



Obr. 2.2: Vývojový diagram hlavní části programu pro výpočet spotřeby

Pouze v prvním cyklu programu dochází k inicializaci. Zde jsou nastaveny vynulovány některé proměnné a do výchozích hodnot všechny vytvořené příznakové bity, které zajišťují správný chod programu. Do paměti jsou uloženy hodnoty mezních GPS souřadnic všech kotvišť, které přesně určují jednotlivé zastávky pro detekci. A konečně je nastaveno povolení přerušení od časovače spolu s globálním povolením přerušení.

Za inicializační částí programu následuje vzorkování výstupů z anemometru. Data jsou v každém cyklu programu převedena z hodnoty proudu, přivedené na analogové vstupy automatu, na rychlost respektive směr větru. V tomto podprogramu jsou také umístěny chybové bity, které signalizují rozpojení proudové smyčky čidla, což by znamenalo poruchu jedné jeho části, nebo obou.

Program pokračuje rozhodováním, zda je loď v pohybu nebo ne, tím dochází k větvení programu na dvě části, z nichž je vždy vykonávána buď jedna nebo druhá.

Loď v klidu

V případě, že je rychlost lodi nulová následuje rozhodování, aby bylo zajištěno že se detekce lodi v přístavu provede pouze jedenkrát při uvedení lodi do klidového stavu. Tato kontrola je nutná, protože program u PLC je vykonáván cyklicky viz. obr 1.5 z kapitoly o programovatelných automatech, a v důsledku toho by byla detekce, v případě delšího stání lodi na místě, vykonávána opakovaně a byly by přepisovány výsledné hodnoty tohoto podprogramu. Pokud tedy ještě neproběhla detekce lodi v zastávce, zapíná se časovač, kterým je zajištěn ochranný interval v případě, že by došlo k chvilkovému uvedení lodi do klidové polohy například při přirážení ke kotvišti. Jestliže v průběhu čítání nedojde ke změně rychlosti lodi následuje rozhodování zda je aktivní bit blokující detekci. Tento bit zajišťuje blokování detekce nějaký čas po předchozí úspěšné detekci, tak aby nemohlo dojít několikrát k detekci stejného kotviště v případě, že je například větrné počasí a kapitán má problémy s manévrováním a přistáním k pontonu. Pokud je tento bit neaktivní dojde ke spuštění podprogramu pro detekci lodi v kotvišti. V tomto podprogramu jsou porovnávány GPS souřadnice jednotlivých kotvišť s aktuální polohou lodi. Následuje vyhodnocení spotřeby, času a polohy v podprogramu, který je spouštěn pouze při detekování lodi v kotvišti. Při úspěšné detekci kotviště pokračuje běh programu podprogramem, který ukládá naměřená data do paměti automatu, spouští signalizační bity pro výpočet korekce průměrné rychlosti a směru větru a signalizuje displeji okamžik, ve kterém dochází k zápisu uložených dat na USB paměťové médium. Posledním úkonem v tomto podprogramu je vynulován okamžitých bufferů s hodnotami spotřeby, vzdálenosti, času apod. Po vyhodnocení následuje nastavení bitu, který bude po 2 min blokovat úspěšnou detekci kotviště.

Loď v pohybu

Pokud v prvním rozhodování není rychlost nulová, znamená to, že loď je v pohybu a dochází ke spuštění časovače s dobou časování 1 s. Tento časovač při přetečení nastavené hodnoty vyvolá přerušení a v obsluze k tomuto přerušení dochází k měřením a výpočtům veličin použitých pro vyhodnocení spotřeby. V každé obsluze přerušení,

tedy s periodou 1 s, je proveden výpočet průměrné rychlosti a směru větru a z hodnot frekvenčních měničů určena okamžitá spotřeba v kWh. V případě, že je vzorkovací čas vzdálenosti, 2 nebo 30 s, je v rámci obsluhy přerušení vypočítána také uražená vzdálenost z posledních uložených souřadnic do souřadnic aktuálních. Po výkonu obsluhy přerušení běh programu pokračuje od výkonu podprogramu, který vzorkuje výstupní proud z anemometru.

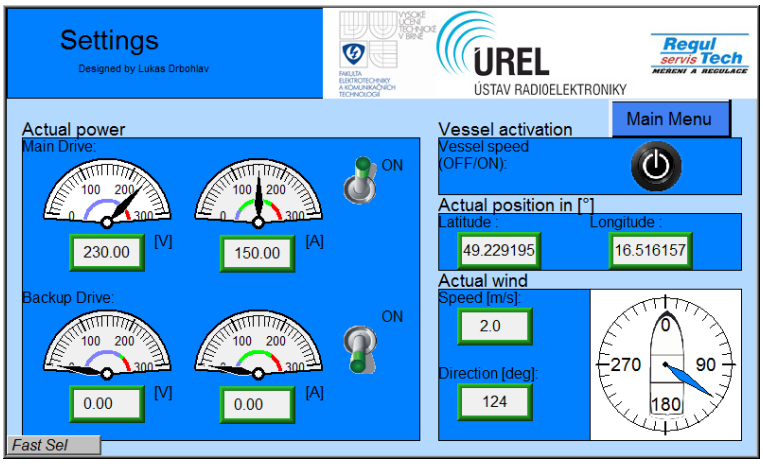
Podle zde uvedeného návrhu následovala realizace vlastního programu v prostředí STEP7/MicroWIN s využitím PLC Simatic S7-224XP a ve vývojovém softwaru EasyBuilder EB8000 pro vizualizaci na ovládací panel Weintek Easy View MT8150X. V této části závěrečné práce je věnována pozornost významným úsekům zdrojového kódu programovacího jazyka pro řídící automaty a jejich podrobnějšímu popisu. V každé následující kapitole je také uveden způsob vizualizace údajů na displej a popis kódu maker, která zajišťují matematicky náročné výpočty, které v PLC není možné přesně realizovat.

2.3 Úvod k testovacímu vizualizačnímu softwaru

V rámci ladění správné funkce programu nahraném v PLC a simulace jeho nejrůznějších stavů, byl vytvořen tzv. Testovací vizualizační software. Součástí této kapitoly bude úvodní seznámení s možnostmi tohoto softwaru a rozbor úvodních obrazovek. Podrobný popis jednotlivých obrazovek bude umístěn v korespondujících kapitolách, které následují v tématických oddílech.

Při testování byl použit dotykový displej nižší řady Weintek MT8070iH, který má sice pouze 7" displej, ale pro simulování programu pro vyhodnocení spotřeby má rovnocenné vlastnosti a rozhraní, které jsou použity u panelu přímo na lodi. Podrobnosti o vlastnostech displeje použitého při testování lze nalézt v datasheetu výrobce [17].

Na obr 2.3 je znázorněna obrazovka testovacího vizualizačního softwaru, která slouží k nastavení parametrů plavby. Na levé polovině obrazovky je umístěna část nastavující okamžitý odběr napětí a proudu motorem. Obě hodnoty lze nastavit po poklepání na číselný blok `Numeric Input Object` a ručně zadat. Rozsah vstupních hodnot je s určitou rezervou od 0 do 300, jak pro napětí tak proud. Optimální hodnoty jsou na části kruhového indikátoru `Meter Display Object` vyznačeny zelenou barvou. Pro hodnotu napětí je to okolo 230 V a proud od 50 do 200 A. Toto lze nastavit pro hlavní i záložní pohon, tedy frekvenční měnič. Aktivace požadovaného frekvenčního měniče se provede nastavení přepínače `Toggle Switch Object` do polohy ON.



Obr. 2.3: Testovací vizualizační software – nastavení

Vpravo první část z vrchu je umístěn přepínač pohybu lodi `Toggle Switch Object`. V případě, že svítí modře loď je v pohybu, pokud není podsvícen loď stojí. Směrem dolů následuje dvojice `Numeric Input Object`, ve kterých lze nastavit aktuální polohu pomocí GPS souřadnic. Poloha je zadávána přímo ve stupních až na šest desetinných míst. A poslední oddíl v pravém dolním rohu slouží pro kontrolu aktuálních parametrů větru – rychlosti a směru. Jsou zobrazeny pomocí `Numeric Display Object`. Navíc je hodnota směru větru pro lepší orientaci zobrazena na kruhovém indikátoru `Meter Display Object`, ve kterém je vyznačena i orientace lodi. Směr odkud vítr fouká ukazuje střelka indikátoru. Nastavování aktuální rychlosti a směru větru je možné pouze na obrazovce věnované anemometrii v kap. 2.47.

2.4 Detekce lodi v kotvišti

Aby mohl výpočet spotřeby mezi jednotlivými zastávkami na linkové trase i mimo ni probíhat automaticky, bylo nutné pomocí GPS souřadnic ověřovat, zda je loď po zastavení v prostoru některého z kotvišť. Je však nutno podotknout, že úspěšné detekování kotviště je velmi důležité nejen pro další zpracování naměřených a vypočítaných dat, ale i pro ukládání hodnot do paměťové médium. A protože manévrování s lodí při přistání k molu je velmi náročná záležitost, která je velmi závislá na podnebních podmínkách, je detekce lodi v kotvišti jedním z nejcitlivějších míst v programu.

2.4.1 Výpočet mezních souřadnic kotvišť

K detekci lodě v kotvišti je zapotřebí mezních GPS souřadnic každého kotviště, tak aby mohly být porovnány s aktuální polohou lodi. Proto byl nejprve zapotřebí matematický aparát, ze kterého se určily meze všech zastávek na Brněnské přehradě. Ten je popsán v této kapitole.

Pro prvotní simulaci funkce tohoto programu, byly stanoveny GPS souřadnice každého kotviště pomocí internetového mapového portálu [18]. Přehledovou mapu Brněnské přehrady se všemi linkovými kotvišti lze nalézt na [19]. Při prvním programování lodi byly pak přímo na přehradě souřadnice upraveny na skutečné hodnoty, které jsou zobrazeny v tabulce 2.1. Protože je přehrada velmi malá plocha ve srovnání s velikostí Země a jejího poloměru, je možné bez větší chyby zanedbat zemské zakulacení při výpočtech vzdáleností. Toto zanedbání vytváří z výpočtů trojrozměrných, výpočty pouze dvourozměrné. Za těchto předpokladů bylo možné odečíst šířku a výšku přehrady v krajních mezích. Z těchto hodnot určit změnu souřadnic na vzdálenosti a vypočítat kolik stupňů zeměpisné šířky a zeměpisné délky připadá na jeden metr.

- **Šířka přehrady:** 4,369 km
  - rozmezí zeměpisných délek: 16°27'23,155" až 16°30'59,344"
  - zeměpisná šířka konstantní
- **Výška přehrady:** 5,025 km
  - rozmezí zeměpisných šířek: 49°13'50,169" až 49°16'32,867"
  - zeměpisná délka konstantní

Tab. 2.1: Skutečné souřadnice kotvišť

ID	Název	Zem. šířka	Zem. délka
0	Bystrc – Servis	49,229130	16,516092
1	Bystrc	49,230110	16,516380
2	Kozí Horka	49,234325	16,508917
3	Sokolské koupaliště	49,240589	16,512278
4	U Kotvy	49,241661	16,501698
5	Osada	49,246033	16,502290
6	Rokle	49,248474	16,493917
7	Cyklistická	49,257680	16,481379
8	Hrad Veveří	49,259565	16,463910
9	Mečkov	49,261162	16,454914
10	Skály	49,265490	16,456861
11	Veverská Bítýška	49,275887	16,456682

Protože program zpracovává reálná čísla místo stupňů a minut, bylo nutné převést tyto jednoty na desetinné číslo. To však zajistila část řídícího softwaru lodi. Z těchto parametrů je již možné vypočítat změnu zeměpisných souřadnic na jednom

metru délky. Pro změnu zeměpisné délky na šířce přehrady platí totiž

$$\Delta longitude\,[^\circ] = longitude2\,[^\circ] - longitude1\,[^\circ], \tag{2.1}$$

kde $longitude2$ je horní mez zeměpisné délky, $longitude1$ je dolní mez zeměpisné délky. Příklad výpočtu pro změřené hodnoty

$$\Delta longitude = 16,516484 - 16,456432 = 0,060052^\circ, \tag{2.2}$$

Potom je změna zeměpisné délky vztažená na vzdálenost 1 m dána vzorcem

$$\Delta longitude\,[^\circ \cdot m^{-1}] = \frac{\Delta longitude\,[^\circ]}{width\,[m]}, \tag{2.3}$$

kde $\Delta longitude$ je změna zeměpisné délka na šířce přhrady a $width$ šířka Brněnské přehrady. Následuje opět příklad výpočtu s dosazením naměřených hodnot

$$\Delta longitude = \frac{0,060052}{4369} = (1,375 \cdot 10^{-5})^\circ \cdot m^{-1}, \tag{2.4}$$

Změna zeměpisné šířky na výšce přehrady by se vypočítala obdobným způsobem s výsledkem

$$\Delta latitude = \frac{0,045194}{5025} = (8,994 \cdot 10^{-6})^\circ \cdot m^{-1}. \tag{2.5}$$

Při detekci lodě v kotvišti program vyhodnocuje, zda se loď nachází ve čtverci se stranou 20 m a středem v daném kotvišti, mimo kotviště Bystrc a servisní molo. Bystrc se detekuje, jako obdélník o rozměrech 35 × 90 m, tak aby v těchto mezích byla skryta i dvě postranní mola určená pro údržbu a nakládku. Servisní kotviště je také detekováno jako obdélník, ovšem s rozměry 50 × 70 m. Je totiž rozděleno na několik částí a loď nestojí vždy u stejného mola, z toho důvodu probíhá při detekci servisního kotviště porovnání se stejnými mezními souřadnicemi pro všechny lodě. Hodnoty mezních souřadnic jednotlivých kotvišť byly vypočítány tak, že přičetlo k přesné souřadnici středu kotviště 10 m a odečetlo 10 m. Výpočet horní mezní zeměpisné délky určitého kotviště lze matematicky zapsat

$$longitude\_h\,[^\circ] = longitude\,[^\circ] + 10 \cdot \Delta longitude\,[^\circ \cdot m^{-1}], \tag{2.6}$$

kde $longitude$ je zeměpisná délka určitého kotviště a $\Delta longitude\,[^\circ \cdot m^{-1}]$ změna zeměpisné délky na jeden metr vzdálenosti. To znamená, že výpočet horní mezní zeměpisné délky pro kotviště Kozí Horka lze vypočítat podle

$$longitude\_h = 16,508917 + 10 \cdot 1,375 \cdot 10^{-5} = 16,509055^\circ. \tag{2.7}$$

Tento postup se opakoval pro všechna kotviště a pro obě zeměpisné souřadnice. Hodnoty mezních zeměpisných šířek a délek pro všechna kotviště jsou uvedeny v tabulce 2.2. Více o GPS systémech lze nalézt v literatuře [20].

Výše uvedené hodnoty slouží tedy pro popis jednotlivých kotvišť v paměti programovatelného automatu. Ten neobsahuje přesné souřadnice, ale definuje kotviště jako souřadnice, které náleží do zmíněných mezí.

Tab. 2.2: Mezní souřadnice kotvišť

Název	Latitude_d	Latitude_h	Longitude_d	Longitude_h
Bystrc – Servis	49,228815	49,229445	16,515748	16,516436
Bystrc	49,229705	49,230515	16,516105	16,516586
Kozí Horka	49,234235	49,234415	16,508780	16,509055
Sokolské koupaliště	49,240499	49,240679	16,512141	16,512416
U Kotvy	49,241571	49,241751	16,501561	16,501836
Osada	49,245943	49,246123	16,502153	16,502428
Rokle	49,248384	49,248564	16,493780	16,494055
Cyklistická	49,257590	49,257770	16,481242	16,481517
Hrad Veveří	49,259475	49,259655	16,463773	16,464048
Mečkov	49,261072	49,261252	16,454777	16,455052
Skály	49,265400	49,265580	16,456724	16,456999
Veverská Bítýška	49,275797	49,275977	16,456545	16,456820

2.4.2 Detekce v PLC

Celý algoritmus pro detekci lodi v kotvišti probíhá pouze v PLC. V displeji je soustředěn pouze přepis ID kotviště, čísla přiřazeného kotvišti podle tab. 2.1, na název a jeho výpis na dotykový ovládací panel.

Inicializace

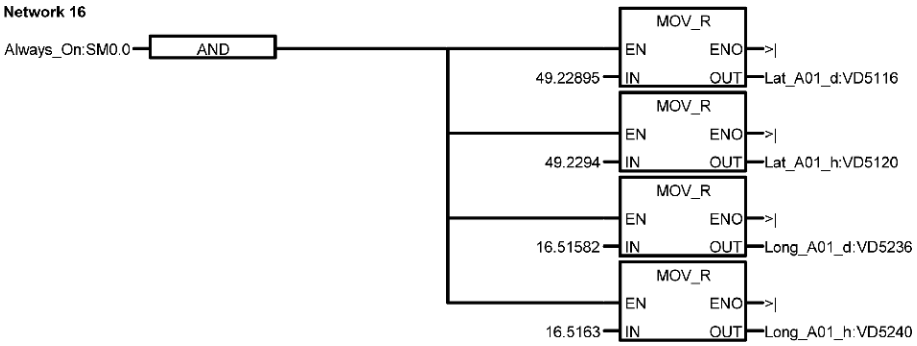
Tento podprogram je volán z hlavní smyčky a vykonáván pouze jednou po spuštění PLC, tedy v prvním cyklu programu. Jeho hlavním úkolem je zajištění správného počátečního nastavení příznakových bitů, uložení GPS souřadnic pro detekci každého kotviště na Brněnské přehradě a také počáteční nulování či přiřazení pomocných proměnných a konstant.

Jsou zde nastaveny do výchozí hodnoty signalizační bity použité pro detekci. Bit, který signalizuje počáteční nebo koncové kotviště úseku je nastaven na `FALSE`. Bit, který zajišťuje, aby proběhla detekce pouze jednou se nastaví na `TRUE`. Bity signalizující detekci zeměpisné šířky a zeměpisné délky jsou nastaveny na úroveň `FALSE`. Bit blokující detekci po určitou dobu po úspěšném detekování kotviště nataven do úrovně `FALSE`. A konečně se na úroveň `TRUE` se nastaví signalizační bit detekce prvního kotviště po zapnutí řídícího automatu.

Pro první cyklus detekce kotviště je použit čas ochranného intervalu před detekcí 0 s. Tato hodnota je důležitá při zapnutí lodě, po první úspěšné detekci se čas nastaví na 5 s a tento interval je používán až do konce plavby. V inicializační části je tedy pomocí instrukce `MOV_W` uložena do proměnné `Ochr_interval` nulová hodnota.

Část kódu, která zajišťuje uložení mezních souřadnic pro jedno kotviště je znázorněna na obr. 2.4. Blok `AND` slouží k rozvětvení jednoho toku programu. Postupně jsou tak vykonány všechny instrukce `MOVE_R`, které přesouvají reálnou hodnotu ze vstupu `IN` na výstup `OUT`. Jejich vykonání probíhá postupně po řádcích od shora dolů.

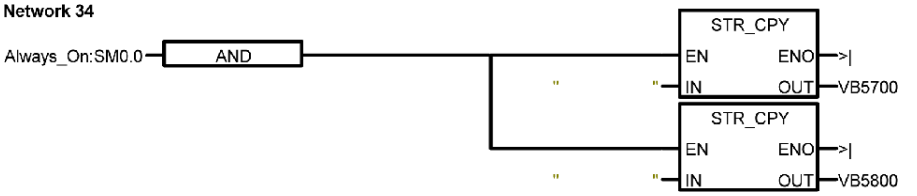
Zde patrnou částí kódu je možné uložit do paměti všechny čtyři mezní souřadnice kotviště. V tomto případě se jedná o souřadnice servisního kotviště Bystrc. A následují totožné bloky instrukcí, které zajišťují uložení mezních souřadnic pro všech 11 kotvišť.



Obr. 2.4: Inicializace – detekce

Blok instrukcí, který nuluje paměťová místa s počátečními a koncovými GPS souřadnicemi je realizován obdobně jako předchozí network. Instrukce `MOV_R` však přesouvají nulové hodnoty a protože je to instrukce, která pracuje s datovým typem `real`, je nutné nuly pro správnou funkci napsat s desetinnou tečkou.

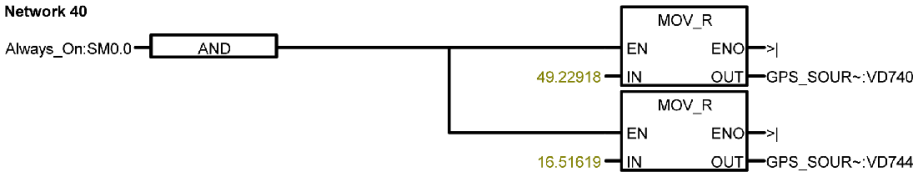
Posledními inicializačními instrukcemi je vymazání textových polí, které na displeji zobrazují aktuální a předchozí detekované kotviště. Tyto instrukce jsou zobrazeny na obr 2.5. Jedná se o dvě instrukce `STR_CPY`, které kopírují řetězec vložený na vstup `IN` do oblasti v paměti zapsané na výstupu `OUT`. Zde je do obou paměťových míst zapsán řetězec 20 znaků mezera, které přemažou celé zobrazovací pole.



Obr. 2.5: Inicializace – vymazání názvů kotvišť

Dále následují jen instrukce, které mají význam pouze při simulaci, protože vytvářejí počáteční podmínky plavby. Nejprve jsou nastaveny hodnoty odpovídající aktuálním GPS souřadnicím na souřadnice středu prvního kotviště – Bystrc-servis. To zajišťují dvě instrukce `MOV_R` zobrazené na obr. 2.6.

V této části kódu se také nastavuje počáteční hodnota bitu `Speed_0`, který signalizuje zda loď stojí, nebo pluje, na úroveň `TRUE`. To znamená, že počáteční stav lodi je, že stojí na místě.

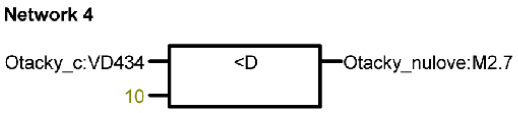


Obr. 2.6: Inicializace – nastavení počátečních souřadnic pro simulaci

Hlavní smyčka programu

Tato část programu probíhá cyklicky stále dokola viz. obr 1.5, na kterém je zobrazen výkon programu programovatelným automatem. V prostředí STEP7/MicroWin se nazývá OB1, tedy organizační blok. V přiloženém programu má symbolické jméno `MAIN`, to bylo zvoleno programátorem, pro lepší přehlednost mezi jednotlivými záložkami. Z tohoto programu jsou dále volána potřebná přerušení – INT nebo podprogramy – SBR. Názvy všech bloků na obrázcích jsou převzaty z testovacího softwaru pro vyhodnocení spotřeby lodi, který musel být ještě upraven a následně vložen do stávajícího řídícího softwaru.

Na obr 2.7 je část hlavního programu, vyhodnocení otáček lodního šroubu. V případě, že budou otáčky v rozmezí od 0 do 10, lze považovat loď za zastavenou. Podmínku nelze nastavit na porovnání otáček s nulou, protože kolem lodního šroubu stále proudí voda a tak, by nedošlo nikdy k vyhodnocení, že loď stojí na místě. Porovnání zajišťuje instrukce `<D`, která dává na výstup úroveň `TRUE`, pokud je horní vstup menší než spodní.

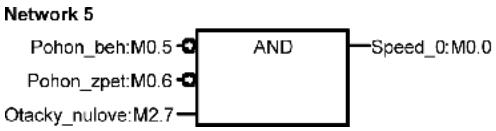


Obr. 2.7: Hlavní smyčka – vyhodnocení otáček

Pro jistější detekci klidového stavu lodi je použita další podmínka. Ta se vztahuje na rychlostní páku umístěnou v kormidelně. Pokud tato páka není nastavena na dopředný ani zpětný běh motoru a zároveň jsou minimální otáčky, pak je loď považována za zastavenou. Tento network je zobrazen na obr 2.8.

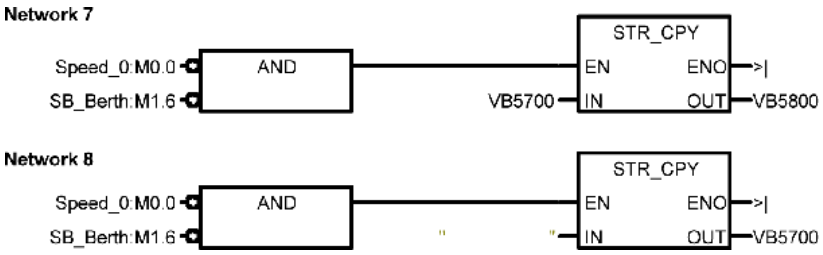
Předchozí dvojice instrukcí je byla v testovacím softwaru vložena mezi příkazy `JMP` a `LBL`. Ty zajišťují jejich přeskok při vykonávání hlavní smyčky programu, protože při simulaci nebyly přítomny žádné z těchto vstupů a v testovacím vizualizačním softwaru byl nastavován pohyb lodi přímo tlačítkem, které tento bit měnilo.

Následující obr 2.9 zobrazuje network 7 a 8, které slouží pro kopírování řetězců datového typu `string`. V obou případech, je kopírování provedeno, pokud je nenu-



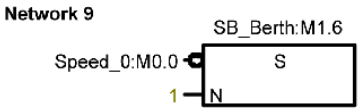
Obr. 2.8: Hlavní smyčka – nulová rychlost lodi

lová rychlost lodi a bit `SB_Berth` rozlišující počáteční a koncové kotviště v log. 0, tedy nastavený na počáteční kotviště. Instrukce `STR_CPY` se tedy vykonají v momentě, kdy loď vyplouvá z kotviště. V tuto chvíli se nakopíruje název aktuálního kotviště do názvu kotviště předchozího a název aktuálního kotviště je přepsán prázdným řetězcem, takže se vymaže. Tato dvojice instrukcí zajišťuje správné zobrazování názvů kotvišť na displeji, to je podrobněji popsáno v části zabývající se vizualizací detekce lodi v kotvišti 2.4.3.



Obr. 2.9: Hlavní smyčka – přesun názvu kotviště

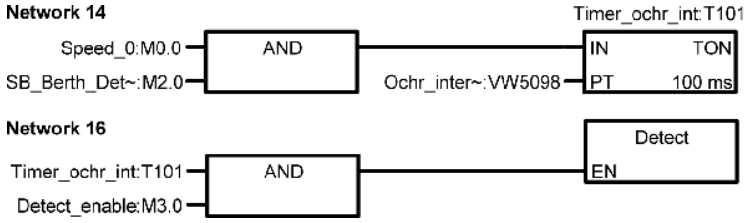
Na obr 2.10 je využit blok instrukce `Set`, který nastaví určitý počet bitů od dané počáteční adresy na log. 1. Tento blok aktivuje povolovací vstup, což je vstup první. Počet bitů je potom zadán na druhý vstup a výstupní adresa je umístěna nad blokem instrukce. V tomto případě se jedná o nastavení jednoho bitu od adresy `SB_Berth`, což je adresa, na které je uložen signalizační bit, který rozlišuje počáteční a cílové kotviště lodi. Na povolovací vstup je přiveden bit `Speed_0`, ve kterém je uložena informace o tom, jestli je loď v klidu nebo v pohybu.



Obr. 2.10: Hlavní smyčka – identifikace počátečního kotviště

Jak je patrné z obr 2.11, hlavním blokem je zde časovač. Jeho první vstup slouží k řízení časování. Na něj je přivedena log. 1 pro spuštění a log. 0 pro vynulování

a zastavení. Druhý vstup určuje do jaké maximální hodnoty bude časovač počítat. V instrukčním souboru automatu S7-224XP jsou časovače, které se inkrementují s krokem 1 ms, 10 ms nebo 100 ms. Proto výsledný čas, do kterého časovač inkrementuje svou hodnotu je dán vynásobením hodnoty zapsané na druhém vstupu a kroku časovače. Zde se jedná o časovač s krokem 100 ms a celkovým časem odpovídajícím proměnné `Ochr_interval` vynásobené s použitým krokem časovače. Tento ochranný interval je zde umístěn především z důvodu přirážení lodi k nástupnímu molu, protože při něm kapitán loď několikrát rozjíždí a zastavuje. Na povolovací vstup časovače je připojen blok s instrukcí `AND`. Na jeho vstupech jsou bity `Speed_0` a `SB_Berth_Detect`. Druhý ze jmenovaných je signalizační bit zajišťující proběhnutí pouze jediné detekce při uvedení lodi do klidové polohy.



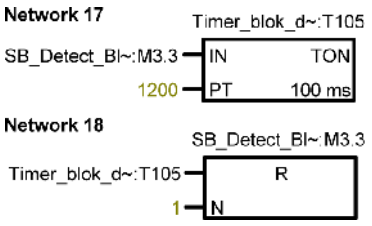
Obr. 2.11: Hlavní smyčka – ochranný interval a detekční podprogram

V následujícím network program vyvolá podprogram pro detekci lodi v kotvišti `Detect`, při nastavení bitu časovače T101 a současně zapnutém povolovacím bitu z displeje. Bit časovače je nastaven až po té co má loď po dobu ochranného intervalu minimální otáčky lodního šroubu a rychlostní páku v klidové poloze. Na druhý vstup povolovací funkce `AND` je přiveden bit `Detect_enable`. Ten je nastavován displejem, v momentě, kdy se při spuštění lodi načte hlavní obrazovka programu displeje, protože je Simatic spuštěn několikanásobně rychleji. V důsledku toho by nedošlo k úplné detekci lodi v kotvišti a ani program pro výpočet spotřeby by nefungoval správně.

Mezi těmito dvěma bloky je vložena ještě instrukce `MOV_W` pro uložení aktuální hodnoty časovače do paměti automatu. Hodnota je zobrazována na displej v testovacím vizualizačním softwaru pro možnost odladění správné funkce ochranného intervalu předcházející detekci.

Poslední část hlavní smyčky programu, která souvisí s detekcí kotviště, je blokování detekce, viz. obr 2.12. Na vstupu prvního network je bit `SB_Detect_Block`, který je nastaven ihned po úspěšné detekci kotviště v podprogramu detekce. Tím je spuštěn časovač T105 s intervalem 2 min. Po dobu dvou minut je blokována úspěšná detekce lodi v kotvišti. A v následujícím network je signalizační bit po uplynutí stanoveného času opět vynulován. Díky tomu je vykonána detekce kotviště jedenkrát

a nemělo by docházet k několikanásobné detekci vlivem příjezdu lodi k molu. To by se projevilo i při ukládání naměřených dat.

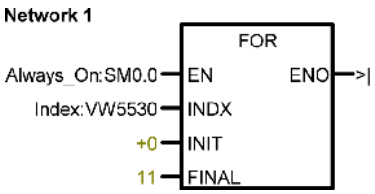


Obr. 2.12: Hlavní smyčka – blokování detekce

Podprogram detekce

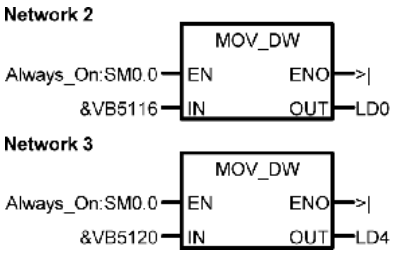
Tento podprogram je spouštěn z hlavního cyklicky vykonávaného programu, v případě, že má loď minimální otáčky lodního šroubu a rychlostní páku v klidové poloze po dobu víc než 5 s. Probíhá vždy jednou v momentě, kdy je rychlost lodi vyhodnocena jako nulová.

Podprogram detekce začíná cyklem `FOR`, zobrazeným na obr 2.13. Ten proběhne celkem 12 krát, od 0 do 11, což odpovídá počtu kotvišť na Brněnské přehradě, která jsou uvedena v tabulce 2.1. Cyklus `FOR` je ukončen návěštím `NEXT`. Mezi těmito network je pak část programu opakována od počáteční hodnoty indexu na vstupu `INIT` po konečnou na vstupu `FINAL`. Index je inkrementován po každém proběhnutí této části kódu. V případě, že index překročí hodnotu stanovenou mezemi, dojde ke skoku na návěští `NEXT` a pokračování v programu od tohoto network.



Obr. 2.13: Detekce – cyklus porovnávání mezních souřadnic kotvišť

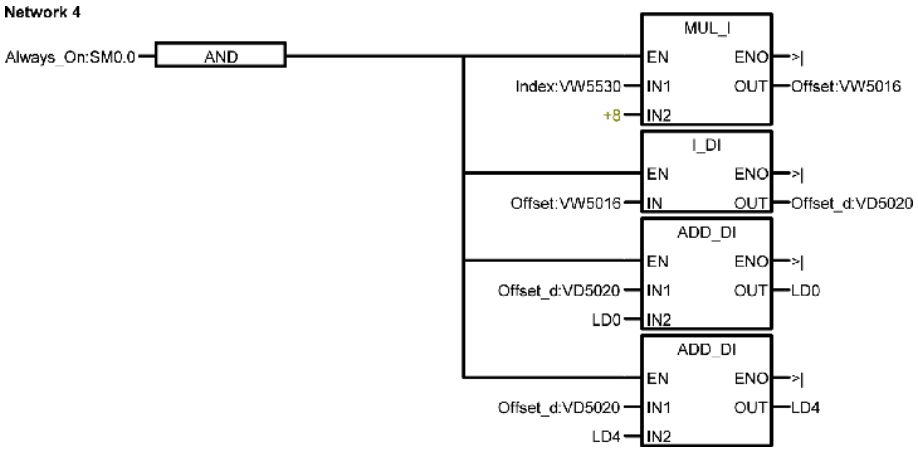
Uvnitř cyklu dochází k uložení adres s mezními zeměpisnými šířkami prvního kotviště do dvojice ukazatelů, pomocí funkce `MOV_DW`. K porovnání mezních souřadnic zeměpisné šířky s aktuální souřadnicí zeměpisné šířky, na které se nachází loď, je nutné použít dva ukazatele. Jeden z nich ukazuje na horní mezní zeměpisnou šířku a druhý na dolní mezní zeměpisnou šířku prvního kotviště. Vše je zobrazeno na obr 2.14.



Obr. 2.14: Detekce – uložení ukazatelů na adresy souřadnic kotvišť

Funkce `MOV_DW` , přesouvá hodnotu rozměru Double Word z adresy na vstupu do adresy uvedené na výstupu tohoto bloku. Vstupní hodnota je označena &, což znamená že se ukládá ukazatel na tuto adresu. Výstupní adresa je umístěna v dočasné lokální paměti, viz. 1.2.3. Protože se obsah lokální paměti po vykonání podprogramu maže, je výkon těchto dvou instrukcí uzavřen v cyklu `FOR`.

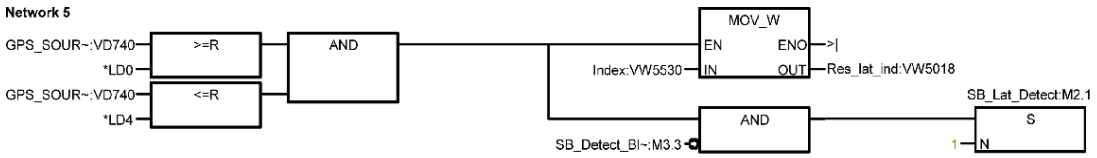
Následující blok instrukcí, uvedený na obr 2.15, slouží k posuvu ukazatelů, podle indexu cyklu `FOR` , postupně na mezní zeměpisné šířky všech kotvišť pro porovnávání se skutečnou mezní zeměpisnou šířkou, na které se loď nachází.



Obr. 2.15: Detekce – posuv ukazatelů po zeměpisných šířkách kotvišť

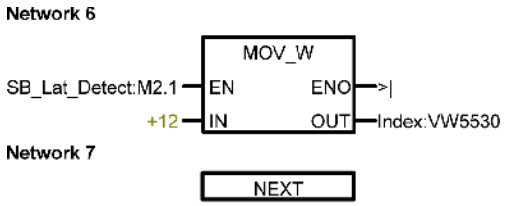
V prvním kroku se do proměnné `Offset` uloží hodnota indexu cyklu `FOR` vynásobená 8 (blok `MUL_I`). Hodnota konstanty odpovídá posuvu o 8 Bajtů, což je velikost dvou hodnot typu `Double Integer`. Proměnná `Offset` tedy udává o kolik je třeba posunout ukazatel na mezní zeměpisnou šířku porovnávaného kotviště. Protože velikost proměnné `Index` je typu `Integer` , je nutná typová konverze hodnoty `Offset` na `Double Integer` instrukcí `I_DI`. Pak už jen stačí přičíst hodnotu `Offset` k oběma ukazatelům dvojicí funkcí `ADD_DI`.

Nyní již následuje porovnání aktuální zeměpisné šířky s oběma ukazateli tak, že aktuální souřadnice musí ležet mezi hodnotami uloženými v obou ukazatelích zároveň. Instrukce, které toto zajišťují jsou uvedeny na obr 2.16. V instrukční sadě automatu simatic se vždy porovnává dvojice hodnot. Proto se nejdříve porovnává aktuální souřadnice s obsahem ukazatele na dolní mezní zeměpisnou šířku a následně až aktuální souřadnice s obsahem ukazatele na horní mezní zeměpisnou šířku. Pokud jsou obě porovnání splněna uloží se hodnota proměnné `Index`. Tato hodnota udává přesně, o které kotviště se jedná. Dále se nastaví vlajkový bit `SB_Lat_Detect`, který signalizuje, že aktuální zeměpisná šířka lodi je v rozmezí některého z kotvišť. To je ovšem podmíněno úrovní `FALSE` bitu blokujícího detekci 2 min po předchozí úspěšné detekci `SB_Detect_Block`.



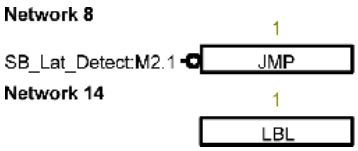
Obr. 2.16: Detekce – porovnání zeměpisné šířky

Na obr 2.17 je zobrazeno ukončení cyklu `FOR` . V případě, že dojde k detekování zeměpisné šířky některého z kotvišť, je tedy aktivní signalizační bit `SB_Lat_Detect`, nastaví se hodnota `Index` cyklu `FOR` na 12. Tedy mimo rozsah cyklu. Proto se v následujícím běhu programu cyklus `FOR` přeskočí až na návěští `NEXT` v network 7.



Obr. 2.17: Detekce – ukončení prohledávacího cyklu

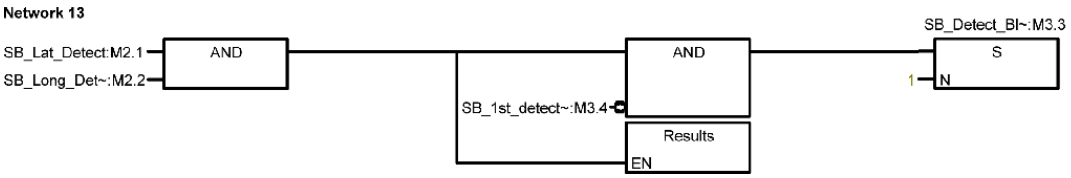
Podprogram pokračuje instrukcí `JMP`, viz. obr 2.18. Tato instrukce zajišťuje skok na návěští, pokud je splněna podmínka na jejím vstupu. Na vstupu použité instrukce je hodnota `SB_Lat_Detect`. Pokud se tedy nedetekuje aktuální zeměpisná šířka, jako souřadnice kotviště, dojde k přeskočení části kódu až na návěští `LBL` s příslušným číslem. Část kódu umístěná v podmínce je detekce zeměpisné délky. Pokud není detekována zeměpisná šířka, jako poloha kotviště, je zbytečné procházet a porovnávat zeměpisné délky.



Obr. 2.18: Detekce – skok, v případě nedetekování zem. šířky

V network 9 následuje detekce zeměpisné délky. Postup je velmi podobný porovnávání aktuální souřadnice se zeměpisnou šířkou. Začíná se instrukcemi `MOV_DW`, které ukládají adresy mezních zeměpisných délek prvního kotviště v paměti procesoru do lokální paměti, stejně jako na obr 2.14. Souřadnice kotviště jsou uspořádané dvojice zeměpisných šířek a délek. Proto je možné po detekování zeměpisné šířky porovnat aktuální zeměpisnou délku s jedinou dvojicí mezních zeměpisných délek. Z tohoto důvodu byla v části pro detekci zeměpisné šířky uložena hodnota indexu cyklu `FOR`, která říká, o jaké kotviště se pravděpodobně jedná. To protože jednotlivá kotviště nemají společnou ani jednu zeměpisnou souřadnici. A s pomocí této hodnoty se nastaví ukazatele na mezní zeměpisné délky odpovídající již detekované zeměpisné šířce. V dalším kroku jsou tyto hodnoty porovnány obdobným způsobem, jako je na obr 2.16, s aktuální zeměpisnou délkou lodi. S tím rozdílem, že již není ukládán index cyklu `FOR`. A pokud je loď v daném intervalu a není detekce blokována bitem `SB_Detect_Block` nastaví se bit `SB_Long_Detect`, který signalizuje správné detekování zeměpisné délky.

To, že se obě souřadnice aktuální polohy lodi nachází v některém z kotvišť, je signalizováno bity `SB_Lat_Detect` a `SB_Long_Detect`, viz obr. 2.19.



Obr. 2.19: Detekce – spuštění podprogramu pro vyhodnocení

Proto, pokud jsou oba tyto bity aktivní je možné přejít na následující podprogram, který řeší vyhodnocení detekce `Result`. Ještě před tím, ale dochází k nastavení bitu blokující úspěšnou detekci po dobu 2 min od této detekce – `SB_Detect_Block`. Jediná vyjímka, kdy není tento bit nastaven je první detekce po spuštění řídícího systému, což zajišťuje v podmínce bit `SB_1st_detect`.

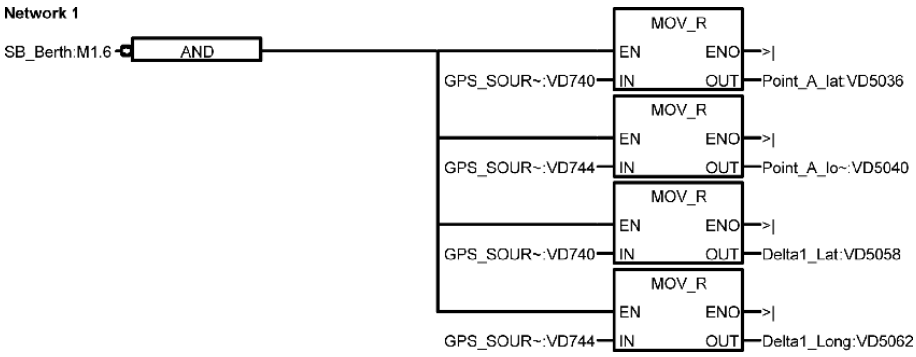
Pokud ovšem zeměpisná délka nesedí do mezních souřadnic žádného z kotvišť, tak se tento krok neprovede a dojde pouze k vynulování bitu `SB_Berth_Detect`,

který zajišťuje, že detekce probíhá pouze jednou při uvedení lodi do klidového stavu. A následně se program vrací do hlavní smyčky.

Podprogram dokončení detekce

Tento podprogram slouží k ukončení detekce lodi v kotvišti, vyhodnocení jejích výsledků a výsledků plavby, uložení naměřených a vypočítaných dat do paměti a vynulování paměťových pozic použitých k ukládání hodnot z dalšího úseku plavby.

Běh podprogramu rozlišuje, zda se jedná o počáteční či koncové kotviště daného úseku plavby a podle toho ukládá hodnoty souřadnic, vypočtené spotřeby a vzdálenosti. Podmínku, zda se jedná o detekci počátečního nebo koncového kotviště na daném úseku plavby zajišťuje signalizační bit `SB_Berth`. Pokud je jeho hodnota log. 0 jsou uloženy aktuální souřadnice lodi do počátečního kotviště, neboli do `Point_A` a také do `Delta1`, pomocí kterého je v průběhu plavby počítána vzdálenost, kterou loď urazila. Hodnota `Point_A` byla použita pouze při vývoji, v současné verzi kódu nemá žádný význam. Ukládání zajišťuje funkce `MOV_R`, která přesouvá hodnoty datového typu `Real` na vstupu bloku do oblasti v paměti určené výstupem bloku. Vše je zobrazeno na obr 2.20.

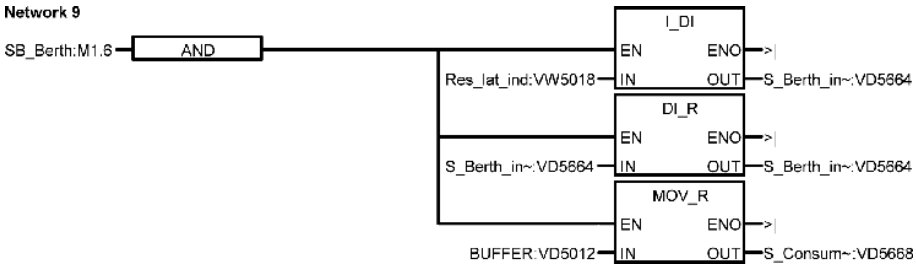


Obr. 2.20: Dokončení detekce – uložení aktuální polohy v počátečním kotvišti

V případě, že se `SB_Berth` rovná log. 1, dochází k výkonu následujícího bloku instrukcí, které souvisí s detekcí lodi v kotvišti. Podprogram pokračuje dvojicí network, kde jsou aktuální souřadnice uloženy do cílového kotviště, tedy `Point_B`, pomocí funkce `MOV_R`. A v následujícím kroku jsou přepsány souřadnice počátečního kotviště souřadnicemi kotviště cílového, protože pro další úsek plavby je cílové kotviště počáteční.

Pro ukládání naměřených a vypočítaných hodnot je nutné data, která mají být uschována na paměťovém médiu, uložit do po sobě jdoucích paměťových míst. Tím je zabezpečeno, že se všechny hodnoty ukládají do stejného souboru. Proto dochází k přetypování čísla indexu `Res_lat_ind` z `integer` na `real` ovšem s mezikrokem

přes `double integer`. Instrukční soubor procesoru programovatelného automatu neobsahuje příkaz, který by to zajistil v jednom kroku. Bloky vykonávající přetypování a uložení čísla kotviště se nacházejí na obr. 2.21.

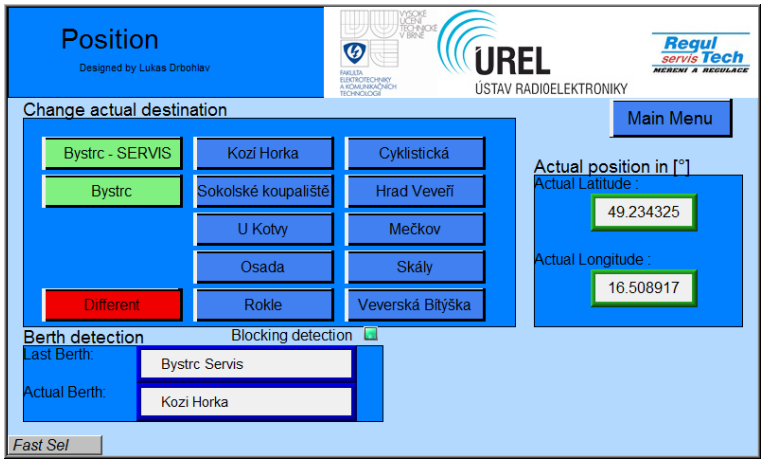


Obr. 2.21: Dokončení detekce – přetypování a uložení čísla kotviště do paměti

Instrukcí `RESET` je pak nulován signalizační bit rozlišující počáteční a koncové kotviště – `SB_Berth`. Posledními instrukcemi se nuluje signalizační bit, který říká, že se jedná o první detekci po zapnutí automatu – `SB_1st_detect` a nastavení doby trvání čekacího intervalu před detekcí. Tento čas je nastaven v inicializačním podprogramu na 0 s, protože se automat zapíná v servisním kotvišti, při přechodu z režimu nabíjení do provozního režimu, a v tu chvíli loď stojí na místě. Mimo první detekci po zapnutí automatu je čas ochranného intervalu nastaven na 5 s.

2.4.3 Vizualizace detekce

Na obr. 2.22 je obrazovka, která ošetřuje vizualizaci detekce lodi v kotvišti pro testovací vizualizační software.



Obr. 2.22: Testovací vizualizační software – detekce

Makro – rychlá změna aktuálních GPS souřadnic

Dominantní část této obrazovky slouží k rychlému přepínání aktuálních GPS souřadnic lodi na polohu středů jednotlivých kotvišť. `Function Key Object` zajišťuje po stisknutí vykonání makra pro dané kotviště. Makro je kód v jazyce podobném jazyku C, který programátorovi umožní naprogramovat si množství dalších libovolných programů. Ty jsou vykonány, jako reakce na nějakou událost. Makro, které uloží do paměti automatu hodnotu předdefinovaných souřadnic, v tomto případě souřadnic spadajících do kotviště Bystrc-Servis, je zobrazeno mezi klíčovými slovy `macro_command` a `end macro_command`.

```
macro_command main()
float a, b

a = 49.229176
b = 16.516193

SetData(a, "SIEMENS S7/200 (Ethernet)", VD, 740, 1)
SetData(b, "SIEMENS S7/200 (Ethernet)", VD, 744, 1)

end macro_command
```

Na začátku kódu jsou deklarovány proměnné typu `float`, do kterých je zapsána zeměpisná šířka a délka kotviště. Tyto hodnoty jsou pak jednoduše pomocí příkazu `SetData` uloženy do paměti automatu na pozici VD740 a VD744.

Vpravo na obrazovce je pro lepší přehlednost uvedena dvojice `Numeric Display Object`, která zobrazuje aktuální polohu lodi.

Jako doplňující informace byly na displej přidány bloky zobrazující název kotviště, tzv. `Ascii Display Object`. Na displeji je zobrazován název aktuálního kotviště v případě, že loď má minimální otáčky, rychlostní páku v nulové poloze a aktuální souřadnice jsou detekovány, jako souřadnice některého z možných kotvišť. Po rozjetí lodi se název zobrazený v aktuálním kotvišti vymaže a přesune se do názvu kotviště předchozího. Tímto způsobem je po celou dobu plavby zobrazováno v jakém kotvišti se loď naposledy pohybovala.

Makro – výpis názvů detekovaných kotvišť

Ukládání názvů kotvišť do paměti PLC zajišťuje makro. V `PLC Control Object` je nastaveno přesné spouštění makra změnou úrovně bitu `SB_Long_Detect`, tento bit je nastaven pouze v případě, že je detekováno na aktuálních souřadnicích kotviště.

Následuje úryvek kódu tohoto makra:

```
macro_command main()
short Res_lat_ind, reset = 0
char berth0[20] = "Bystrc Servis"
...
char clear[20] = ""

GetData(Res_lat_ind, "SIEMENS S7/200 (Ethernet)", VW, 5018, 1)

select case Res_lat_ind
  case 0
    StringSet(clear[0], "SIEMENS S7/200 (Ethernet)", VW, 5700, 20)
    StringSet(berth0[0], "SIEMENS S7/200 (Ethernet)", VW, 5700, 20)
    break
  case 1
    ...
    break
  ...
  break
end select

SetData(reset, "SIEMENS S7/200 (Ethernet)", M, 2.1, 1)
SetData(reset, "SIEMENS S7/200 (Ethernet)", M, 2.2, 1)

end macro_command
```

Datový typ zde definovaných proměnných je `short`, který má rozměr 8 bitů. V deklaraci proměnných také dochází k uložení názvů všech 12 kotvišť do polí řetězců. Poslední pole je pole mazací, protože obsahuje 20 znaků mezery. To je nutné pro smazání obsahu okna `Ascii Display Object`, před zapsáním názvu dalšího kotviště.

Vlastní program makra začíná instrukcí `GetData`, která přečte data z uvedeného programovatelného automatu. Data mají počátek v adrese VW5018, jedná se o jeden 16 bitový registr, a jsou uložena do proměnné `Res_lat_int`. Prakticky se jedná o hodnotu indexu cyklu `FOR` po úspěšné detekci kotviště v automatu, protože tato hodnota přímo udává, o které kotviště se jedná. Viz. kap. 2.4.2.

Následuje větvení makra pomocí struktury `select case` – `end select`. Jako vstupní parametr byla použita proměnná `Res_lat_ind`, podle které dochází k rozhodování, v jakém kotvišti se loď po detekci nachází, respektive jaký název má být

vypsán na displej. Ve výše uvedeném kódu je patrné, že při vyhodnocení podmínky `Res_lat_ind` == 0, jako splněné, dochází nejprve ke smazání předchozího názvu kotviště a následně k uložení do stejného paměťového prostoru proměnné `berth0`, ve které je uložen název kotviště. K uložení řetězce do paměti procesoru Simatic se používá funkce `StringSet`, která má parametry postupně – proměnná, kterou chceme zapsat do automatu, použitý automat, část paměti v automatu a velikost bloku, počáteční adresa a počet znaků v řetězci. Každý případ `case` je nutné ukončit klíčovým slovem `break`, což zajistí přeskočení ostatních větví `case` až do bodu `end_select` v kódu.

Po skončení větvící podmínky ještě dochází k vynulování signalizačních bitů automatu, `SB_Lat_Detect` a `SB_Long_Detect`, které informují o úspěšné detekci zeměpisné délky a zeměpisné šířky. To je provedeno instrukcí `Set_Data`, která se používá pro zápis jednotlivých bitů do paměti procesoru. Tím se ošetřuje předběhnutí automatu oproti displeji.

Posledním důležitým prvkem této obrazovky je `Bit Lamp Object`, který se rozsvítí po úspěšné detekci kotviště a signalizuje blokování další úspěšné detekce po dobu 2 min. Informuje tedy o stavu bitu `SB_Block_Detect`.

2.5 Výpočet spotřeby

K výpočtu spotřeby dochází s 1 s intervalem v momentě, kdy není splněna klidová podmínka lodi. Rychlostní páka se nachází v poloze vpřed nebo vzad a otáčky lodního šroubu jsou vyšší než minimální klidové otáčky (10 otáček/min). Potom je vzorkována hodnota napětí na svorkách a výstupní proud aktivního frekvenčního měniče. Aktuální spotřeba se vypočítá podle vztahu

$$P = U \cdot I, \tag{2.8}$$

kde U je napětí na svorkách aktivního frekvenčního měniče a I jeho výstupní proud do motoru. Protože byla zadavatelem požadována hodnota spotřeby v kWh je nutné výsledek předchozího vztahu převést na kW, podělením hodnoty 1000. Druhá úprava vypočteného příkonu se týká převodu na část hodiny, protože je aktuální spotřeba počítána s periodou 1 s, je nutné spotřebu v kWh vydělit konstant 3600, která odpovídá počtu sekund v hodině. Výsledný vzorec je

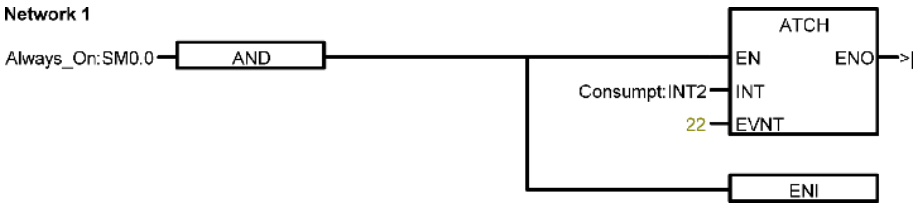
$$P[kWh] = \frac{P}{1000 \cdot 3600}, \tag{2.9}$$

kde P je aktuální spotřeba počítaná každou 1 s.

2.5.1 Spotřeba v PLC

Inicializace

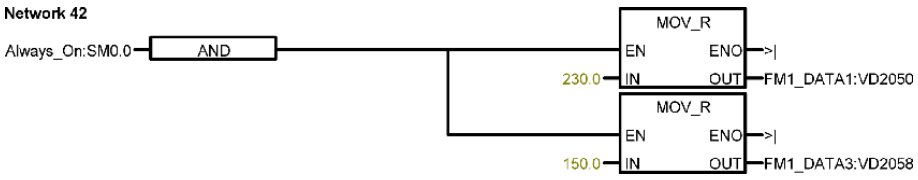
V inicializačním podprogramu dochází nejprve k vykonání instrukce `ATCH`, která slouží k povolení přerušení od události zadané na třetím vstupu. Obsluhu tohoto přerušení vykoná program, jehož symbolické jméno je uvedeno na druhém vstupu. V tomto případě se jedná o přerušení při události číslo 22, což je přetečení časovače T96, které je obslouženo programem `Consumpt`. Druhý v pořadí bude vykonán blok instrukce `ENI`. Ten slouží ke globálnímu povolení přerušení. Network je zobrazen na obr 2.23.



Obr. 2.23: Inicializace – povolení přerušení od přetečení časovače

Následuje nastavení signalizačních bitů, spojených s výpočtem spotřeby, do požadovaných počátečních stavů. Bit, který zajišťuje nulování časovače T96 (obsluha přerušení po 1 s), se nastaví na úroveň `FALSE`. Dále dochází k počátečnímu vynulování paměťového místa, do kterého je ukládána spotřeba na aktuálním, předchozím úseku plavby a celková denní spotřeba.

V samotném závěru podprogramu `Initial` se nastavují počáteční hodnoty pro simulaci, jak je patrné z obr. 2.24. První instrukce `MOV_R` nastavuje napětí na svorkách hlavního frekvenčního měniče na 230,0 V a druhá instrukce `MOV_R` pak výstupní proud frekvenčního měniče na 150 A.

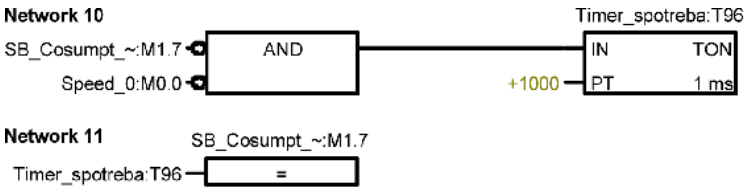


Obr. 2.24: Inicializace – nastavení hodnot pro simulaci

Hlavní smyčka programu

V hlavní smyčce programu jsou obsaženy pouze dva network, které jsou využity

při výpočtu spotřeby. Základem prvního je časovač T96, jak je patrné z obr 2.25.



Obr. 2.25: Hlavní symčka – časovač spouštějící výpočet spotřeby

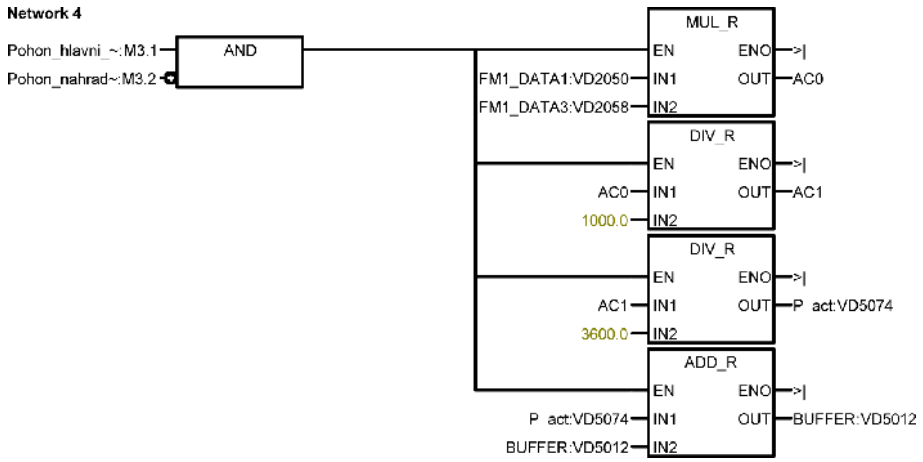
V použitém zapojení počítá časovač do hodnoty 1 s, pokud je na jeho povolovací vstup přivedena log. 1. Na povolovací vstup je připojen blok instrukce `AND` se vstupními bity `SB_Consumpt_T_Clear` a `Speed_0`. To zajišťuje, že časovač je v normálním časovacím módu jen v případě nenulové rychlosti lodi a současně nulového bitu pro restart časovače. S tímto network je spojen i network následující, který ukládá do nulovacího bitu časovače hodnotu bitu časovače. Bit časovače je nenulový pouze v momentě, kdy je na časovači shodná hodnota s požadovanou. Proto tento blok instrukce přiřazení `=` zajišťuje nulování hodnoty časovače po každém načítání do 1 s. Od tohoto časovače se odvíjí obsluha přerušení v případě přetečení jeho hodnoty, přes údaj nastavený na vstupu.

Obsluha přerušení

Jedná se o obsluhu přerušení od přetečení časovače T96. Obsluha se provádí každou 1 s a tak často je tedy počítána spotřeba lodi. Výsledek se potom přepočítá na kilowatty a přičte se k obsahu bufferu.

Na obr 2.26 dochází k výpočtu spotřeby asynchronního motoru v kWh. Na povolovacím vstupu jsou dva vstupní bity automatu. `Pohon_hlavni` je v log. 1, když je motor napájen z hlavního frekvenčního měniče. `Pohon_nahradni` svým nastavením indikuje, že motor napájí náhradní frekvenční měnič. Tyto dva vstupy samozřejmě nemohou být nikdy současně aktivní. V simulačním programu jsou vstupy automatu nahrazeny bity z jeho paměti, kvůli možnosti ladění a přepisování.

Tento blok instrukcí je vykonán v případě napájení motoru z hlavního frekvenčního měniče. Blok instrukce `MUL_R` je určen pro násobení dvou reálných hodnot, v tomto případě se násobí napětí na svorkách hlavního frekvenčního měniče s jeho výstupním proudem a výsledek je ukládán do akumulátoru, tedy do dočasného místa v paměti, určeného pro ukládání mezivýsledků. Následující instrukce `DIV_R` zapíše na výstup hodnotu prvního vstupu poděleného vstupem druhým. V tomto případě dochází k dělení nejprve hodnotou 1000, což znamená převod na kW, a poté je ještě



Obr. 2.26: Výpočet spotřeby – v kWh při napájení hlavním frekvenčním měničem

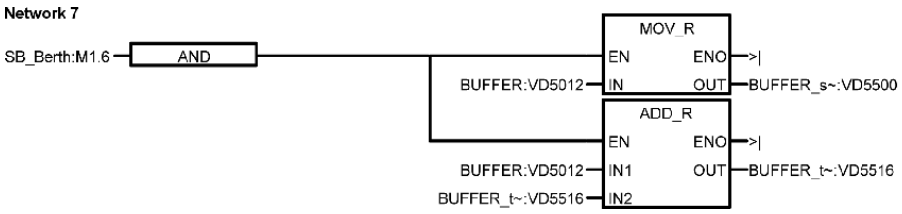
tato hodnota dělena 3600, což lze brát jako váhování odečtené hodnoty spotřeby. Pokud totiž je odečítána hodnota spotřebované energie každou sekundu a ukládá se do bufferu, kde se všechny hodnoty sčítají, je tato energie právě 1/3600 celkové energie spotřebované za hodinu. Blok instrukce `ADD_R`, již počítá součet vstupních dat a ukládá ho do paměti na výstupu. Tento blok realizuje neustálé přičítání do hodnoty `BUFFER`, což je celková spotřeba na aktuální trase.

Pro záložní frekvenční měnič je blok instrukcí velmi obdobný, s tím rozdílem, že se na vstup bloku `MUL_R` přivádí napětí a proud z měniče záložního.

Dokončení detekce

Po dojetí do cílového kotviště, tedy když je `SB_Berth` v log. 1, dochází také k uložení hodnoty spotřeby na daném úseku plavby do oblasti v paměti nazvané `Buffer_save`. Následuje přičtení spotřeby uložené v `Buffer` k `Buffer_total` pomocí instrukce `ADD_R`. To je celková spotřeba za celý den od uvedení lodi do provozu. Všechny tři hodnoty jsou vypisovány na displej, pro lepší přehled aktuální spotřeby, spotřeby na posledním úseku plavby a celkové denní spotřeby. Blok instrukcí zajišťující právě popsané kroky je uveden na obr. 2.27.

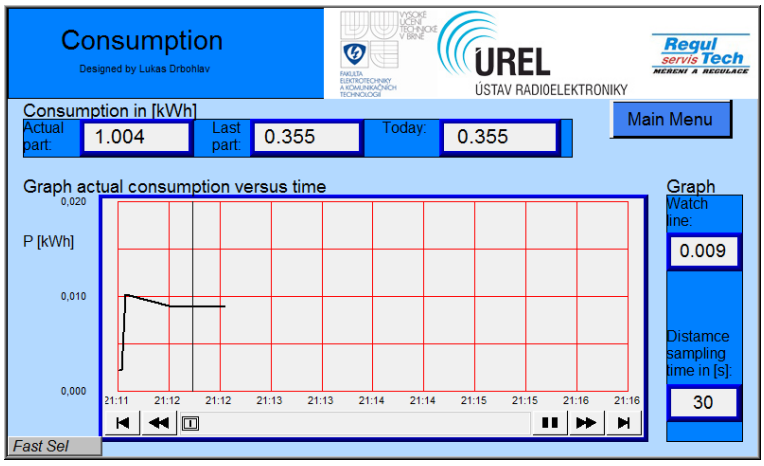
Poslední dvě instrukce, které zajišťují část programu zabývající se výpočtem spotřeby, jsou pro uložení hodnoty spotřeby na aktuálně skončeném úseku trasy `BUFFER` do paměťového bloku, ze kterého bude spolu s dalšími informacemi ukládána na USB médium připojené k displeji. Bloky zajišťující právě popsanou funkci lze nalézt v předchozí kapitole na obr. 2.21. A druhá instrukce, která zajišťuje vynulování hodnoty `BUFFER`, tak aby v další části plavby mohla být proměnná využívána pro ukládání celkové spotřeby na aktuálním úseku trasy v kWh, tak jako tomu bylo doposud.



Obr. 2.27: Dokončení detekce – uložení spotřeby na předchozí trase do paměti

2.5.2 Vizualizace spotřeby

Obrazovka testovacího vizualizačního softwaru, která se zabývá zobrazením údajů o spotřebě na displej je zobrazena na obr. 2.28. V horní části plochy jsou umístěny tři prvky `Numeric Display Object`. První zobrazuje paměťové místo v PLC, kde je uložen údaj o aktuální spotřebě na právě probíhajícím úseku trasy. Do druhého je vypisována spotřeba na předchozím úseku plavby a poslední blok zobrazuje akumulační hodnotu spotřeby za celý den.



Obr. 2.28: Testovací vizualizační software – spotřeba

Většinu zobrazitelné plochy zaujímá grafická závislost aktuální spotřeby v závislosti na čase. Aby se požadovaná data vykreslovala podle zadání bylo nejdřív nutné nastavit v `Data SamplingObject` vzorkování požadovaných hodnot. Vzorkování je spuštěno aktivací bitu `SB_Display`, který je zmiňován v souvislosti s výpočtem vzdálenosti, více v kapitole věnované této tématice. 2.6.2. Dále se nastavuje paměťové místo, ze kterého budou data přečtena, tedy místo s vypočtenou aktuální spotřebou v kWh, která je počítána každou 1 s. Maximální počet vzorků je nastaven na 45000 pro mezní případ, že by docházelo k vykreslení do grafu každou druhou sekundu dne. A formát dat byl zvolen `Real`. Posledním krokem je nastavení bitu, který když

je nastaven na úroveň `TRUE`, dojde k vyčištění oblasti grafu. Tento bit je `SB_Berth` a k vymazání obsahu vzorkovací paměti dojde při rozjetí lodi od správně detekovaného kotviště na další úsek trasy.

Pak již následuje nastavení v bloku `Trend Display Object` vzorkovací objekt, ze kterého mají být požadovaná data vykreslena a parametry zobrazení grafu. Dochází zde také k určení paměťového místa, nejlépe v displeji, do kterého bude ukládána hodnota grafu v místě kurzoru. `Numeric Display Object`, který tuto hodnotu zobrazuje je umístěn na straně obrazovky. A pod ním hodnota užitečná především pro ladění softwaru, tedy údaj po jaké době dochází aktuálně k vzorkování vzdálenosti, ale i s tím spojené vykreslování aktuální spotřeby do grafické závislosti.

2.6 Výpočet vzdálenosti

Měření a výpočet tohoto parametru plavby je poměrně důležité, protože z uražené vzdálenosti lodí mezi dvěma kotvišti a času odpovídajícímu této plavbě se vypočítá odhad průměrné rychlosti plavidla. Průměrná rychlost poslouží ke korekci průměrné rychlosti a směru větru. To znamená, že na přesnosti měření a výpočtu tohoto parametru závisí přesnost měření rychlosti a směru větru.

2.6.1 Výpočet vzdálenosti ze dvou GPS souřadnic

GPS souřadnice jsou v simaticu již přepočítány řídícím programem lodi z hodnot ve stupních, minutách a sekundách na reálná čísla s rozměrem ve stupních. Jelikož se jedná o sférické souřadnice – zeměpisnou délku a šířku, je nejprve nutné je přepočítat na souřadnice kartézské. K tomu slouží následující tři rovnice:

$$x\,[km] = \mathrm{R}\,[km] \cdot sin(\theta\,[^\circ]) \cdot cos(\lambda\,[^\circ]), \tag{2.10}$$

$$y\,[km] = \mathrm{R}\,[km] \cdot sin(\theta\,[^\circ]) \cdot sin(\lambda\,[^\circ]), \tag{2.11}$$

$$z\,[km] = \mathrm{R}\,[km] \cdot cos(\theta\,[^\circ]), \tag{2.12}$$

kde R je poloměr Země navýšený o nadmořskou výšku v místě brněnské přehrady, což je pro výpočet na tak malé ploše bráno, jako konstanta. λ je zeměpisná délka ve stupních. θ je úhel od osy z zmenšený o velikost zeměpisné šířky. Jak je patrné z následujícího vzorce

$$\theta\,[^\circ] = 90^\circ - \varphi\,[^\circ], \tag{2.13}$$

kde je φ zeměpisná šířka. Po převedení souřadnic počátečního i koncového bodu do kartézského systému stačí už jen vypočítat vzdálenost v km, po dosazení všech hodnot také v km, podle vzorce

$$D = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2}, \tag{2.14}$$

kde jsou hodnoty s indexem A kartézské souřadnice počátečního bodu výpočtu a hodnoty s indexem B kartézské souřadnice koncového bodu úseku, na kterém je počítána vzdálenost.

Jako příklad výpočtu byla vypočítána vzdálenost kotviště v Bystrci (49,230033 N, 16,516101 E) a druhého kotviště v pořadí Kozí Horka se souřadnicemi (49,234747 N, 16,507811 E) a porovnána s vzdáleností naměřenou pomocí mapového serveru [18]. Tyto souřadnice ještě neodpovídají reálným hodnotám, jedná se pouze o údaje zjištěné pomocí mapového serveru. Jako první je vypočítán úhel θ počátečního bodu, podle vzorce 2.13

$$\theta_A = 90 - 49,230033 = 40,769967^\circ, \tag{2.15}$$

Následuje převod hodnot ve sférických souřadnicích prvního bodu do souřadnic kartézských podle 2.10, 2.11 a 2.12, které vypadají po dosazení takto

$$x_A = 6378,234 \cdot sin(40,769967) \cdot cos(16,516101) = 3993,283\,km, \tag{2.16}$$

$$y_A = 6378,234 \cdot sin(40,769967) \cdot sin(16,516101) = 1184,085\,km, \tag{2.17}$$

$$z_A = 6378,234 \cdot cos(40,769967) = 4830,475\,km, \tag{2.18}$$

Stejně se postupuje i pro koncový bod úseku, na kterém se počítá vzdálenost z GPS souřadnic. Po tomto výpočtu už je možné určit požadovanou vzdálenost dosazením do vzorce 2.14

$$D = \sqrt{(3993,283 - 3993,074)^2 + (1184,085 - 1183,394)^2 + (4830,475 - 4830,818)^2}, \tag{2.19}$$

$$D = 0,799078\,km. \tag{2.20}$$

Vypočtená vzdálenost se pak při porovnání s vzdáleností naměřenou liší v řádově desítkách cm, což je v případě GPS systému, který určuje polohu s přesností okolo nízkých jednotek metrů naprosto postačující přesnost výpočtu.

Dosažitelná přesnost výpočtu

V programovatelném automatu, ale i v displeji, je obsažen procesor, který pracuje s konečným počtem desetinných míst a tím má omezenou přesnost. Proto bylo nutné zjistit, který procesor bude pro výpočet vzdálenosti ze dvou GPS souřadnic vhodnější. Zvýšené nároky na přesnost jsou dány velkým počtem desetinných míst, které je třeba u GPS souřadnic brát v potaz, vzhledem k tomu, že je přehrada v globálním měřítku velmi malá a hodnoty souřadnic se tak mění napříč celou přehrady až v řádu tisícin, spíše desetitisícin stupně.

Porovnání přesnosti výpočtu vzdálenosti je pro význačné hodnoty výpočtu uvedeno v tab. 2.3. Z této tabulky je patrné, že nejpřesnější výpočet tabulkovým procesorem nedosahuje ani jeden z možných procesorů.

Ovšem při podrobnějším porovnání lze dojít k závěru, že hodnoty vypočtené displejem se blíží ideálním tabulkovým více než hodnoty z automatu. Protože dochází k odlišnosti ve výsledku goniometrických funkcí je odchylkou postižen celý další výpočet a chyba se tak akumuluje. Výsledkem toho je, že celková vypočtená vzdálenost se liší při výpočtu v PLC o necelých 50 m na vzdálenosti 800 m. Naproti tomu hodnoty získané z displeje obsahují nepřesnost v řádu jednotek centimetrů, což při měření polohy globálním pozičním systémem GPS již z principu nemá žádný význam.

Tab. 2.3: Porovnání přesnosti výpočtu vzdálenosti

	Tabulkový procesor	**PLC**	**Displej**
x_A	3993,283993	3993,309	3993,28418
y_A	1184,085346	1184,101	1184,085449
z_A	4830,475525	4830,46	4830,475586
x_B	3993,074215	3993,06	3993,073731
y_B	1183,394623	1183,386	1183,394409
z_B	4830,818195	4830,842	4830,818359
D [km]	**0,799079112**	**0,8480172**	**0,7995741**

Z výše uvedených důvodů byl výpočet vzdálenosti ze dvou GPS souřadnic zapracován do vizualizačního programu, přesněji do makra vizualizačního programu, které tento výpočet v přesně daných časových intervalech zajišťuje.

2.6.2 Vzdálenost v PLC

I přesto, že při porovnání přesnosti výpočtu vzdálenosti ze dvou GPS souřadnic, byl přijat jednoznačný závěr, že procesor v displeji je přesnější, dochází k částem výpočtu uražené vzdálenosti v řídícím automatu. Jedná se zejména o spouštění výpočtu pomocí makra displeje, nastavením signalizačního bitu, tak aby byl výpočet proveden v přesně stanovený čas.

Inicializace

Jako první je v této části programu uložena úroveň `FALSE` do bitu signalizujícího výpočet vzdálenosti v aktuálním cyklu programu. Bit udávající zda rozdíl dvou po sobě jdoucích vzorků spotřeby je nižší než zadaný práh, je nastaven na úroveň `FALSE`. Bit, který signalizuje displeji začátek výpočtu vzdálenosti je nastaven do úrovně `FALSE`.

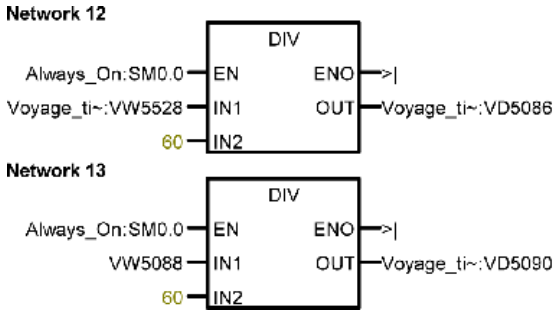
Do všech čtyř paměťových míst, ve kterých se budou nacházet souřadnice aktuální a předchozí pro výpočet vzdálenosti je uložena nulová hodnota. Počáteční čas vzorkování vzdálenosti je nastaven na 2 s. Hodnota, do které je ukládán předchozí vzorek spotřeby je vynulována, stejně jako hodnota vzdálenosti na aktuálním úseku.

Dále je nulována hodnota vzdálenosti na předchozím úseku a celková denní vzdálenost. A v neposlední řadě i hodnota odhadu průměrné rychlosti lodi na předchozím úseku je nulována.

Protože v této části programu je zahrnut i výpočet čistého času plavby na aktuální a předchozí trase, jsou v rámci inicializace vynulovány i hodnoty času plavby v sekundách, ale také hodnoty času předchozího úseku v minutách a hodinách, které jsou ukládány až při detekci koncového kotviště daného úseku. Aktuální čas plavby se počítá z počtu sekund, proto nemusí být nulovány při inicializaci.

Hlavní smyčka

Jedinými instrukcemi v hlavní smyčce programu spadajícími do této kapitoly jsou bloky vykonávající převod času aktuální plavby v sekundách na minuty a hodiny, které jsou následně zobrazovány na displej posádce. Tyto instrukce jsou zobrazeny na obr. 2.29.



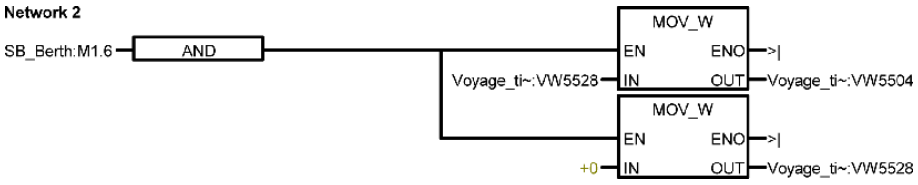
Obr. 2.29: Hlavní smyčka – převod času v sekundách na minuty a hodiny

Konkrétně se jedná o instrukce `DIV`. Tato instrukce dělí dvě 16-bitová čísla s 32-bitovým výsledkem, přičemž významnějších 16 bitů(nižší adresa) v sobě nese zbytek po dělení a méně významných 16 bitů(vyšší adresa) celočíselný podíl dvou vstupních hodnot. Povolovací bit obou instrukcí je systémový bit `Always_On`, který je stále v log. 1. To je možné proto, že počet sekund plavby mezi dvojicí kotvišť je počítáno pouze pokud je loď v pohybu a k vynulování této hodnoty dojde až po detekci koncového kotviště aktuálního úseku trasy. Na vstupu první instrukce `DIV` je hodnota `Voyage_time_count`, ve které je uschován počet sekund od vyplutí z posledního detekovaného kotviště. Po celočíselném vydělení této hodnoty 60, bude výsledkem zbytek **(=počet sekund)** a nějaký výsledek. Tento výsledek je vložen na vstup následující instrukce `DIV` a na jejím výstupu je zbytek **(=počet minut)** a zbytek **(=počet hodin)**.

Dokončení detekce

Podprogram dokončení detekce začíná instrukcí, která při detekci počátečního kotviště, uloží hodnoty aktuálních GPS souřadnic do počátečních souřadnic první části pro výpočet vzdálenosti. To odpovídá první detekci po zapnutí lodi do režimu plavby. Jedná se o instrukce `MOV_R` se vstupy `GPS_SOURAD_1` a `GPS_SOURAD_2`, které jsou uloženy do výstupů `Delta1_Lat` a `Delta1_Long`. To vše je podmíněno úrovní `FALSE` bitu signalizujícího počáteční nebo koncové kotviště úseku. Tento blok instrukcí je zobrazen v kapitole 2.4.2 na obr. 2.20.

V případě, že je detekováno koncové kotviště aktuálního úseku, následuje uložení počtu sekund předchozí plavby do paměti, viz. obr. 2.30.

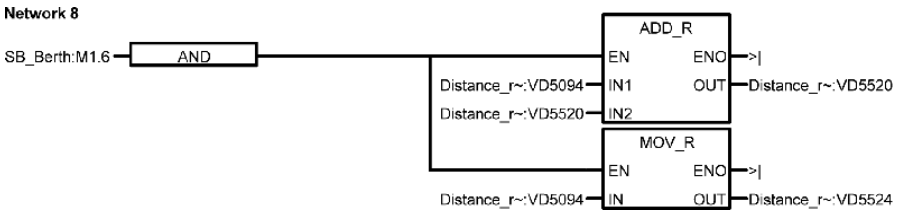


Obr. 2.30: Dokončení detekce – uložení a nulování počtu sekund plavby

K uložení i nulování je využita funkce `MOV_W`, protože `Voyage_time_count` je typu `Word`. Tato hodnota je první instrukcí uložena do hodnoty `Voyage_time_count_save`, ze které je čas za poslední úsek plavby převeden na hodiny, minuty a sekundy a zobrazen na displeji v kormidelně, jako čas předchozí plavby. A hodnota další instrukcí je hodnota `Voyage_time_count` nulována.

Podprogram pokračuje dvojicí instrukcí `DIV`, které z této hodnoty vypočtou čas posledního úseku plavby, obdobně jako na obr. 2.29. Výsledkem instrukcí je zbytek po celočíselném dělení 60 a postup převodu na minuty a hodiny je totožný s postupem uvedeným výše.

Vzdálenost je počítána makrem přímo v displeji, ale její hodnota se posílá přes rozhraní ethernet a ukládá do paměti PLC. Nejprve je ale přičtena k přičtena k obsahu celkové denní vzdálenosti `Distance_result_total` instrukcí `ADD_R`, jak je patrné z obr. 2.31.



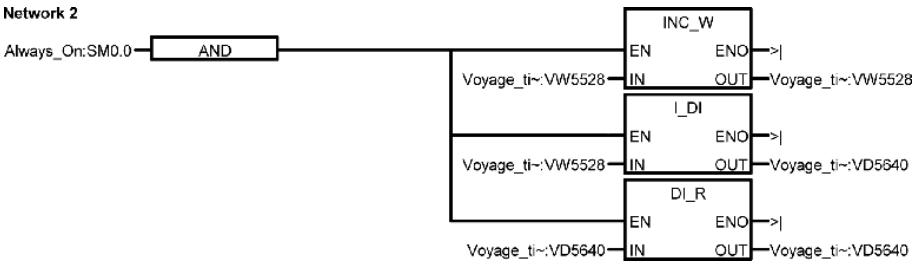
Obr. 2.31: Dokončení detekce – uložení vzdálenosti na aktuálním úseku

Stejná vzdálenost je také uložena do oblasti v paměti tak, aby nebyla přepsána následujícím cyklem výpočtu celkové vzdálenosti na následujícím úseku plavby. Stejně jako v případě spotřeby jsou tyto hodnoty zobrazovány na displeji.

Po vykonání této části podprogramu pro dokončení detekce se do paměti na místo `Distance_result` uloží nulová hodnota tak, aby se na dalším úseku plavby mohla vzdálenost měřit od nuly. A poslední instrukce, která zajišťuje výpočet vzdálenosti v podprogramu dokončení detekce lodi, je nastavení času pro vzorkování vzdálenosti `Get_distance_time` zpět na 2 s pro další úsek plavby.

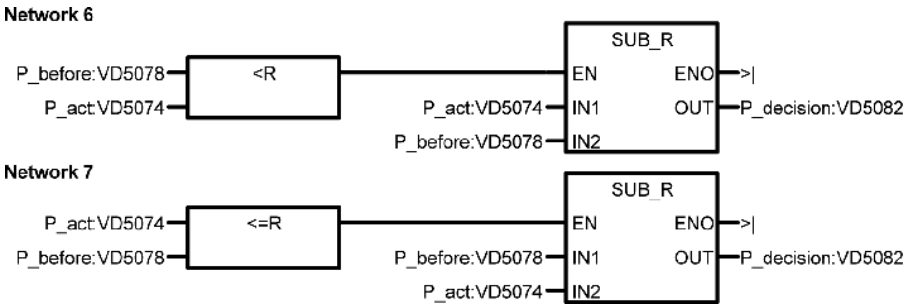
Obsluha přerušení

První blok obsahuje instrukce inkrementace, konverze z datového typu `Interger` na `Double Integer` a následně na typ `Real`, jak je patrné z obr. 2.32. Instrukce `INC_W` slouží k inkrementaci proměnné datového typu `Word`. Na jeho povolovací vstup je přiveden signál, který má stále hodnotu log. 1, proto se vykoná při každém vstupu do obsluhy přerušení. Na datový vstup je přivedeno slovo, které se má inkrementovat a na datový výstup potom oblast v paměti, kam ho má procesor uložit. Proto je na vstupu i výstupu slovo `Voyage_time_count`. Tato hodnota je v displeji ukládána na vložené paměťové médium, ale aby mohla být ukládána do jednoho sešitu spolu s ostatními daty musí být datového typu `Real`, proto je nutné přetypování pomocí instrukcí `I_DI` a dále `DI_R`. Proměnná `Voyage_time_count` v sobě ukládá hodnotu čistého času plavby v sekundách, protože je inkrementována pouze při obsluze přerušení a ta nastává jen když není splněna klidová podmínka lodi. Takže i když se například loď zastaví na nějakou dobu a pak dopluje ke kotvišti, bude zde uložena hodnota o času plavby bez této zastávky.



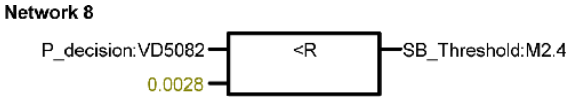
Obr. 2.32: Obsluha přerušení – čas plavby v sekundách

V další části obsluhy přerušení dochází k porovnání absolutní hodnoty rozdílu aktuální a předchozí spotřeby s prahovou spotřebou. Protože procesor programovatelného automatu neobsahuje funkci pro výpočet absolutní hodnoty, musel být rozdíl vypočten pomocí dvou network. Ty vždy zabezpečí, aby se odečítalo nižší číslo od vyššího. Jak je zobrazeno na obr 2.33.



Obr. 2.33: Obsluha přerušení – absolutní hodnota rozdílu spotřeb

Na obr 2.34 je již porovnání absolutní hodnoty rozdílu po sobě jdoucích hodnot spotřeby s prahovou spotřebou. Jako práh byla zvolena hodnota aktuálního příkonu 10 kW, což odpovídá po přepočtu na jednu sekundu spotřebě 0,0028 kWh. Výsledek instrukce `<R`, která porovnává dvě reálná čísla, je uložen do signalizačního bitu `SB_Threshold`. Podle zápisu instrukce je patrné, že pokud bude rozdíl dvou po sobě jdoucích vzorků příkonu nižší než zadaná mez, bude bit nastaven na úroveň `TRUE`.

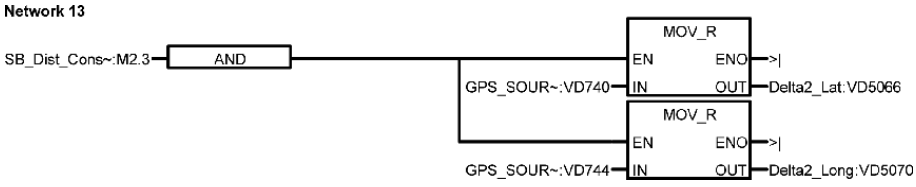


Obr. 2.34: Obsluha přerušení – porovnání rozdílu spotřeb s prahovou úrovní

Následující instrukce řeší případy, kdy je bit `SB_Threshold` aktivní a kdy ne, respektive kdy je rozdíl po sobě jdoucích vzorků spotřeby nižší nebo vyšší zvolená než prahová úroveň. V případě, že je rozdíl aktuální a předchozí spotřeby nižší než práh, nastaví se hodnota `Get_Distance_time` na 30 s. Vychází to z toho, že se když se po sobě jdoucí hodnoty výkonu motoru nemění jede loď konstantní rychlostí a stačí vzorkovat vzdálenost jednou za 30 s. V opačném případě, když je rozdíl hodnot vyšší než práh, dojde k nastavení času `Get_Distance_time` na 2 s a dochází tedy k častějšímu vzorkování uražené vzdálenosti.

V dalším kroku obsluhy přerušení je ošetřeno, kvůli výpočtu uražené vzdálenosti, že po určité době dochází k uložení aktuálních souřadnic polohy lodi do souřadnic koncového bodu úseku na němž je vzdálenost počítána. Koncové souřadnice jsou uloženy v paměti pod názvem `Delta2_Lat` a `Delta2_Long`. Okamžik uložení nastává, když je hodnota `Voyage_time_count` beze zbytku dělitelná časem uloženém v `Get_distance_time`. V takovém případě se po splnění této podmínky nastaví bit `SB_Dist_Consumpt` a vykonají se instrukce na obr 2.35. V dalším network se také nastaví bit `SB_Display`, který signalizuje displeji čas ke spuštění makra

pro výpočet vzdálenosti. Po vykonání těchto instrukcí se do signalizačního bitu `SB_Dist_Consumpt` uloží hodnota `FALSE`.

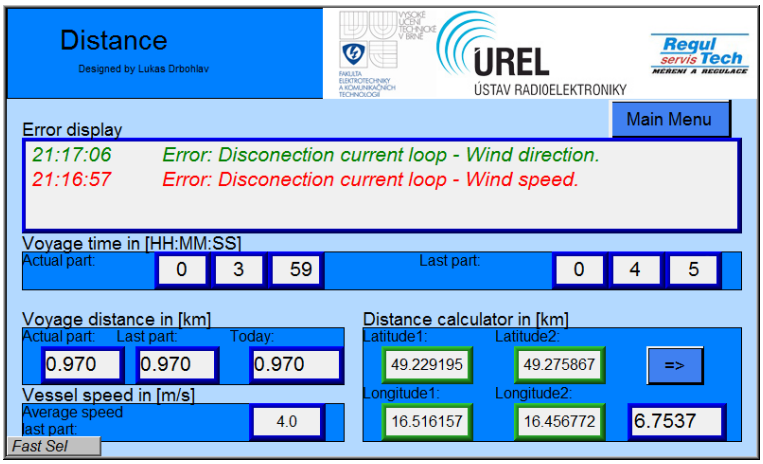


Obr. 2.35: Obsluha přerušení – uložení aktuálních souřadnic

Nakonec každé obsluhy přerušení, od přetečení časovače T96, se přesune hodnota aktuálního vzorku spotřeby `P_act` do hodnoty spotřeby předchozí vzorku `P_before`. Aby bylo možné v následujícím cyklu provádět opět ty samé zde uvedené operace.

2.6.3 Vizualizace vzdálenosti

Na obr. 2.36 je obrazovka testovacího vizualizačního softwaru, která slouží především pro výpočet a zobrazení hodnot vzdálenosti. Dalšími zobrazovanými údaji je čas plavby a signalizace poruchových stavů. Poruchové stavy však budou vysvětleny v kapitolách, které s poruchami přímo souvisí. A výpočet rychlosti lodi na předcházejícím úseku plavby je zajištěn až makrem pro korekci rychlosti a směru větru v kapitole 2.7.4.



Obr. 2.36: Testovací vizualizační software – vzdálenost

Prvním blokem zobrazovaných hodnot na této obrazovce je čistý čas plavby. Hodnota aktuální odpovídá času na právě probíhajícím úseku plavby. Tento čas je

počítán pouze, když je loď v pohybu. Takže pokud je plavba přerušena delší zastávkou uprostřed přehrady a pak doplutím do kotviště tak čas, který bude zobrazen na displeji posádce odpovídá času plavby mezi těmito dvěma kotvišti bez zastávky. Druhá soustava bloků `Numeric Display Object` zobrazuje čas na předchozím úseku plavby.

Makro – pro výpočet vzdálenosti ze dvou GPS souřadnic

Jak již bylo zmíněno v kapitole 2.6.1, byla hlavní část výpočtu uražené vzdálenosti při plavbě zajištěna makrem v ovládacím displeji. Vzdálenost je počítána vždy jednou za určitý časový úsek a spuštění makra je řízeno změnou stavu bitu `SB_Display`, jak je patrné v nastavení `PLC Control` v programu EasyBuilder. Větší část makra pro výpočet vzdálenosti je uvedena v příloze A.1.

Makro začíná definicí proměnných a konstant a jejich datových typů. Pomocí funkce `GetData` jsou načteny aktuální souřadnice počátečního a koncového bodu úseku, na němž má být vzdálenost vypočtena. Tyto souřadnice jsou postupně podle velikosti změny spotřebovaného výkonu měněny softwarem v řídícím automatem.

Další dvojice příkazů slouží k vypočtení úhlu θ, což je úhel doplňující zeměpisnou šířku do 90°, výpočet probíhá podle vzorce 2.13.

Nyní již následuje převod sférických souřadnic počátečního bodu počítané vzdálenosti, zeměpisné šířky a délky, do souřadného systému katézského, podle vzorců 2.10, 2.11 a 2.12. Goniometrické funkce jsou vždy zapsány tak, že první údaj je úhel, pro který se funkce počítá a druhý údaj je proměnná, do které má být výsledná hodnota uložena. Tato posloupnost příkazů je následně provedena i s GPS souřadnicemi koncového bodu.

Výpočet vzdálenosti podle vzorce 2.14 byl rozdělen do několika kroků, protože instrukce displeje umožňují najednou pracovat pouze se dvěma proměnnými. Hodnoty odpovídajících si kartézských souřadnic jsou od sebe nejprve odečteny a uloženy do pomocné proměnné, tato proměnná je umocněna na druhou instrukcí `POW`. Takto vypočítané pomocné proměnné jednotlivých souřadnic se sečtou a vypočítá se z nich druhá odmocnina funkcí `SQRT`. Ta má stejnou syntaxi, jako goniometrické funkce.

Ze simaticu je funkcí `GetData` načtena hodnota prozatím uražené vzdálenosti, k níž je přičten aktuálně spočítaný úsek. Celková vzdálenost na dané trase je pak uložena instrukcí `SetData` zpátky do automatu. A ještě jsou uloženy souřadnice koncového bodu výpočtu vzdálenosti do souřadnic počátečního bodu pro výpočet souřadnic následující. To zajišťují další dvojice funkcí `GetData` a `SetData`.

Hodnoty vypočtené makrem jsou zobrazeny vlevo dole pod údaji o čase. Automaticky je zobrazována hodnota aktuální uražené vzdálenosti na probíhajícím úseku

plavby. Tato hodnota je v kotvišti uložena a zobrazena do vedlejšího bloku `Numeric Display Object` – vzdálenost na předchozím úseku trasy. Konečně uražené vzdálenosti na předchozím úseku jsou akumulovány a zobrazována třetím blokem, jako denní vzdálenost.

Vedle údajů o spotřebě se nacházejí bloky aplikace Distance Calculator. Ten umožňuje si pro zadané souřadnice do čtyř `Numeric Input Object` ověřit vzdálenost. Aplikace je užitečná zejména pro urychlení výpočtů při kontrole správnosti indikovaných výsledků. Hodnoty, se kterými se pracuje jsou umístěny v paměti displeje.

2.7 Výpočet rychlosti a směru větru

Pro objektivní vyhodnocení spotřeby jednotlivých kapitánů dopravního podniku, je nutné porovnávat i podmínky, za kterých byly hodnoty spotřebované elektrické energie naměřeny. Rychlost a směr větru jsou měřeny a vzorkovány s periodou 1 s. Během trasy jsou tyto údaje průměrovány a na konci každého úseku upraveny o rychlost lodi, která má na údaje získané z anemometru velký vliv. Vzhledem k tomu, že jsou obě veličiny získávané ze senzoru větru vzájemné závislé, nejde je jednoduše aritmeticky zprůměrovat. Proto je nutné zavést příslušný matematický aparát.

2.7.1 Převod výstupních hodnot z anemometru

Na analogový vstup programovatelného automatu je přiveden signál v rozsahu 0–20 mA, který převádí vstupní ADC převodník na hodnotu od 0 do 32000. Podíl těchto mezních hodnot odpovídá proudu na jeden bod hodnoty z ADC a proto se vstupní hodnota přivedená do paměti automatu přepočítá na vstupní proud v mA podle vztahu

$$I = ADC\_val \cdot \frac{\Delta I\_ran}{\Delta ADC\_ran} = ADC\_val \cdot \frac{20 - 0}{32000 - 0} = ADC\_val \cdot 6,25 \cdot 10^{-4}, \tag{2.21}$$

kde ΔADC_val je hodnota z ADC, ΔI_ran je rozsah vstupní hodnoty proudu z anemometru a ΔADC_ran rozsah hodnot ze vstupního ADC převodníku automatu.

V dalším kroku je nutné přepočítat proud na danou veličinu. Jelikož se jedná o lineární závislost lze realizovat výpočet parametru pomocí obecné rovnice přímky, která je dána vztahem

$$ax + by + c = 0, \tag{2.22}$$

kde a je x-ová souřadnice normálového vektoru této přímky, b je y-ová souřadnice a c je konstanta, která je dopočítávána, tak aby vztah platil. Souřadnice normálového

vektoru se určí ze vztahů

$$n_1 = \Delta Wind\_Speed\_Range, \tag{2.23}$$

$$n_2 = -(\Delta Anem\_Range), \tag{2.24}$$

kde $\Delta Anem\_Range$ je rozsah proudů z anemometru a $\Delta Wind\_Speed\_Range$ rozsah rychlosti větru odpovídající rozsahu proudů ze senzoru. Po dosazení hodnot dostaneme

$$n_1 = 30 - 0 = 30\,m/s, \tag{2.25}$$

$$n_2 = -(20 - 4) = -16\,mA, \tag{2.26}$$

Dále jsou výsledky dosazeny do obecné rovnice přímky 2.22

$$30x - 16y + c = 0. \tag{2.27}$$

Z tohoto vztahu je vypočtena konstanta c a rovnice se upraví do tvaru

$$y = 1,875x - 7,5, \tag{2.28}$$

kde v tomto případě y reprezentuje rychlost větru v m/s a x výstupní proud z anemometru. Postup odvození vztahu pro výpočet směru větru z přivedeného proudu do automatu je obdobný s výsledkem

$$y = 21,875x - 87,5, \tag{2.29}$$

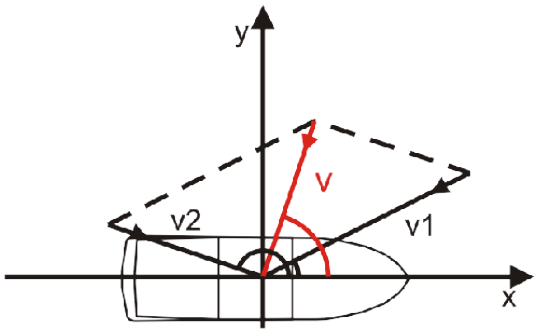
zde je hodnota y směr větru v $^\circ$ a x vstupní proud přivedený na analogový vstup PLC.

Poslední dvě rovnice 2.28 a 2.29 slouží k převedení hodnoty proudu, přiváděného na analogový vstup automatu, na rychlost respektive směr větru. Takto se musí vstupní hodnoty proudu upravit před každým vzorkováním pro výpočet průměrné rychlosti a směru větru.

2.7.2 Výpočet parametrů větru

Vítr je popsán dvěma měřenými parametry – rychlostí a směrem. V tomto případě není možné počítat průměrnou hodnotu rychlosti nebo průměrný směr algebraickým průměrováním. Hlavním důvodem je to, že pro výpočet průměrného úhlu je nutno použít goniometrické funkce pro rozložení větru do jednotlivých os a výsledek upravit podle předem daných kritérií. Dalším důvodem pak může být provázanost obou veličin, takže nelze počítat průměr odděleně každého z parametrů.

Grafické znázornění způsobu výpočtu výsledného směru a rychlosti větru, při působení dvou větrů o různých směrech a rychlostech, je zobrazen na obr. 2.37. Pro možnost algoritmizace tohoto výpočtu je nutné jej definovat matematicky.



Obr. 2.37: Výpočet průměrné rychlosti a směru větru

V čase vzorkovaný vítr je popsán rychlostí v_1 a směrem α_1, vítr vzorkovaný v další vzorkovací periodě je určen rychlostí v_2 a směrem α_2. Oba větry musí být následně rozloženy podle příspěvků do osy x a y podle vztahů

$$x_n = v_n \cdot cos(\alpha_n), \tag{2.30}$$

$$y_n = v_n \cdot sin(\alpha_n), \tag{2.31}$$

kde x_n je vektor n-tého větru rozložen do osy x a y_n vektor n-tého větru rozložen do osy y. Po rozložení vektorů do jednotlivých os se pokračuje výpočtem průměrného příspěvku do každé osy podle vzorečků

$$\bar{x} = \frac{1}{M}\sum_{i=1}^{M} x_i, \tag{2.32}$$

$$\bar{y} = \frac{1}{M}\sum_{i=1}^{M} y_i, \tag{2.33}$$

kde M je počet průměrovaných hodnot. V tomto případě je $M = 2$. x_i je i-tý příspěvek do osy x a y_i je i-tý příspěvek do osy y. Nyní je možné vypočítat velikost průměrného větru, na obrázku označeného jako v, dané vztahem

$$v = \sqrt{\bar{x}^2 + \bar{y}^2}, \tag{2.34}$$

kde hodnota $\bar{x}$, $\bar{y}$ jsou hodnoty průměrného větru rozložené do os x a y. Z hodnoty průměrné rychlosti výsledného větru je vypočten průměrný směr obou vstupních větrů podle rovnice

$$\theta = sin\left(\frac{\bar{y}}{v}\right), \tag{2.35}$$

kde je $\bar{y}$ hodnota výsledného větru v ose y a v velikost výsledného větru, neboli průměrná rychlost odpovídající vstupním větrům.

Protože je funkce *sin* periodická, docházelo by při takto vypočteném průměrném směru větru k nejistému určení, do kterého kvadrantu má být daný směr přiřazen.

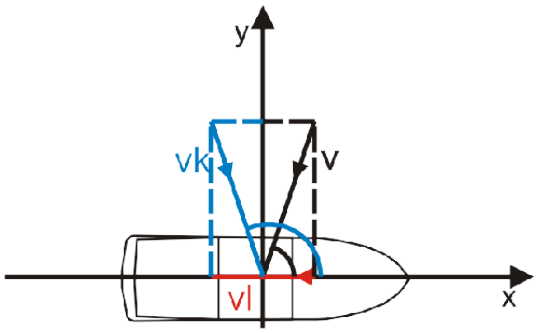
Proto musí být ještě výsledek porovnán se dvěma pravidly. První pravidlo ošetřuje posunutí výsledku do správného kvadrantu

$$cos(\theta) > 0 \quad \wedge \quad sin(\theta) > 0 \quad \Rightarrow \quad \theta = 0° + |\theta|, \tag{2.36}$$
$$cos(\theta) < 0 \quad \wedge \quad sin(\theta) > 0 \quad \Rightarrow \quad \theta = 180° - |\theta|, \tag{2.37}$$
$$cos(\theta) < 0 \quad \wedge \quad sin(\theta) < 0 \quad \Rightarrow \quad \theta = 180° + |\theta|, \tag{2.38}$$
$$cos(\theta) > 0 \quad \wedge \quad sin(\theta) < 0 \quad \Rightarrow \quad \theta = 360° - |\theta|. \tag{2.39}$$

Touto sadou podmínek však výpočet nekončí. Jelikož se může stát, že bude třeba průměrovat vítr ze dvou směrů, jejichž výsledný *sin* nebo *cos* bude roven nule. Pak by nebyl směr přiřazen do správného kvadrantu, proto je nutné upravit algoritmus ještě následující podmínkou

$$sin(\theta) = 0 \quad \wedge \quad \bar{x} < 0 \quad \Rightarrow \quad \theta = 180°, \tag{2.40}$$
$$sin(\theta) = 0 \quad \wedge \quad \bar{x} > 0 \quad \Rightarrow \quad \theta = 0°, \tag{2.41}$$
$$cos(\theta) = 0 \quad \wedge \quad \bar{y} < 0 \quad \Rightarrow \quad \theta = 270°, \tag{2.42}$$
$$cos(\theta) = 0 \quad \wedge \quad \bar{y} > 0 \quad \Rightarrow \quad \theta = 90°. \tag{2.43}$$

Takto provedený a upravený výsledek udává přesnou a správnou hodnotu průměrného směru větru v případě, že je vítr měřen na pevnině. Protože ale bude anemometr umístěn přímo na lodi, bude rychlost lodi zanášet do výsledku vždy nějakou nepřesnost. Ta bude větší v momentě, kdy bude foukat vítr menší nebo srovnatelný s rychlostí lodi a naopak bude menší pokud bude mít vítr vyšší rychlost, než je rychlost lodi. Je tedy nutné zavést korekci upravující výslednou rychlost a směr větru. Postup výpočtu je graficky znázorněn na 2.38.



Obr. 2.38: Výpočet korekce průměrné rychlosti a směru větru

Ten lze matematicky vyjádřit vztahem

$$\bar{x_k} = \bar{x} - v_l, \tag{2.44}$$

kde $\bar{x}$ je výsledný vítr rozložený do osy x ze vztahu 2.32 a v_k je rychlost lodi v daném momentu. K odečtení rychlosti lodi dochází proto, že směr lodi bude vždycky ve směru osy x, tedy pod úhlem 0°. Výpočet pak pokračuje od vzorce 2.34. Takto upravený výsledný směr větru již bude výrazně přesnější.

PŘÍKLAD VÝPOČTU

Nechť má první vítr rychlost $v_1 = 2\,m/s$ a směr $\alpha_1 = 160°$, druhý vítr $v_2 = 2\,m/s$ a $\alpha_1 = 160°$, tedy vítr má při dvou po sobě jdoucích vzorkováních konstantní rychlost a směr. Rychlost lodi je $v_l = 2,78\,m/s$. Rozložení prvního vzorkovaného větru do osy x a osy y je provedeno podle vztahů 2.30 a 2.31

$$x_1 = v_1 \cdot cos(160) = -1,8794 = \bar{x}, \tag{2.45}$$

$$y_1 = v_1 \cdot sin(160) = 0,6840 = \bar{y}. \tag{2.46}$$

Protože jsou parametry obou vzorků stejné odpovídají ve směru jednotlivých os x a y přímo průměrným hodnotám $\bar{x}$ a $\bar{y}$. Následuje výpočet korekce příspěvku do osy x o rychlost lodi podle 2.44

$$\bar{x_k} = -1,8794 - 2,78 = -4,6594. \tag{2.47}$$

Průměrná rychlost větru po korekci vycházející z 2.34, se rovná

$$v_k = \sqrt{(-4,6594)^2 + 0,6840^2} = 4,7093\,m/s. \tag{2.48}$$

A nakonec průměrný směr větru, který je dán dosazením do vztahu 2.35

$$\theta = sin\left(\frac{0,6840}{4,7093}\right) = 8,351°. \tag{2.49}$$

Tím dostaneme výsledný směr větru v prvním kvadrantu, který je ještě nutno upravit prvním pravidlem a protože byla hodnota *cos* záporná a *sin* kladná, úhel bude ve druhém kvadrantu a výsledný směr je pak dán vztahem

$$\theta = 180° - \mid 8,351° \mid = 171,65°. \tag{2.50}$$

Z příkladu je patrné, že i přes to, že podle senzoru by měl být průměrný vítr stejný, jako vítr vzorkovaný (protože byl vítr po dobu obou vzorkování konstantní), došlo po korekci pohybu lodi k úpravě rychlosti a také směru větru a výsledky se od vstupních vzorků liší. Tento postup však zajišťuje znatelně větší přesnost naměřených a vypočtených hodnot.

2.7.3 Rychlost a směr větru v PLC

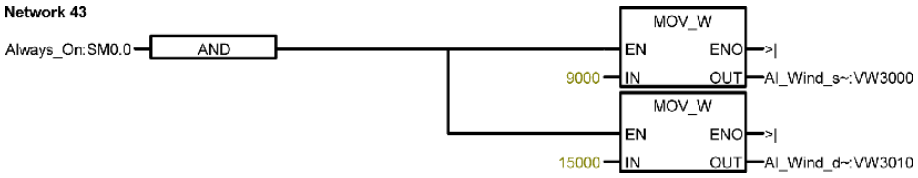
Jak je patrné z předchozí kapitoly, průměrování rychlosti a směru větru se skládá z několika goniometrických funkcí. Jak již bylo zmíněno v kapitole 2.6.1, není procesor programovatelného automatu se svou architekturou a instrukční sadou vhodný pro jejich výpočet. Proto je v řídícím systému pouze spouštěcí část, která signalizuje displeji, v jaký čas má začít vykonávat funkci maker, která průměrování rychlosti a směru větru a korekci těchto hodnot zajišťují.

Inicializace

Protože při výkonu algoritmu pro výpočet průměrné rychlosti a směru větru je majoritní část výpočtu umístěna v displeji, nedochází k nastavování tolika inicializačních hodnot. Nejprve je nataven na úroveň `TRUE` bit, který zajišťuje nulování hodnot, souvisejících s výpočtem větru, v obsluze přerušení. Bit signalizující do displeje start výpočtu průměrných hodnot větru s periodou 1 s je resetován do stavu `FALSE`. A bit, který signalizuje displeji, při doplutí do koncového kotviště úseku, start výpočtu korekce průměrných hodnot větru je nastaven na úroveň `FALSE`.

Dále dochází k nulování hodnot průměrné rychlosti a směru větru na aktuálním úseku trasy, včetně rozložení průměrného větru do příspěvků od osy x a osy y. Také vymazání hodnot průměrné rychlosti a směru větru na posledním úseku trasy s korekcí i bez ní. Tedy se jedná o nulování čtyř proměnných.

Jako počáteční stav anemometru pro simulaci byly do paměťových míst představujících analogové vstupy automatu, uloženy hodnoty v očekávaném rozsahu, viz. obr 2.39. Protože Simatic převádí analogovou vstupní hodnotu z rozsahu 0–20 mA na 0–32000, je do paměti uložena po převodu rychlost 3 m/s a směr větru 118 °. Převod je podrobněji popsán v kapitole 2.7.1.

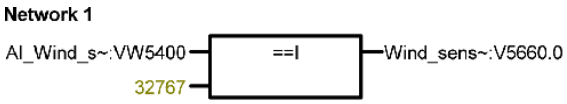


Obr. 2.39: Inicializace – parametry větru pro simulaci

Anemometer

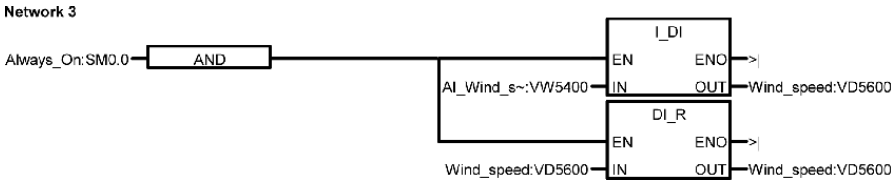
Jedná se o podprogram, který převádí výstupní proud z anemometru na rychlost a směr větru, jak bylo popsáno v kapitole 2.7.1, případně indikuje rozpojení proudové smyčky senzoru, tedy jeho poruchu. Tento podprogram je spouštěn z hlavní smyčky v každém cyklu programu.

První instrukcí je funkce porovnání hodnoty z analogového vstupu automatu, který představuje rychlost větru, s hodnotou poruchy, jak je patrné z obr 2.40. Funkce `==I` porovnává dvě hodnoty datového typu `Integer`. Pokud se vstupy rovnají, nastaví bit na svém výstupu do úrovně `TRUE`. Na vstupu instrukce je hodnota odpovídající analogovému vstupu řídícího systému a hodnotou 32767 hlásí Simatic rozpojení proudové smyčky na analogovém vstupu.



Obr. 2.40: Anemometer – rozpojení proudové smyčky, porucha

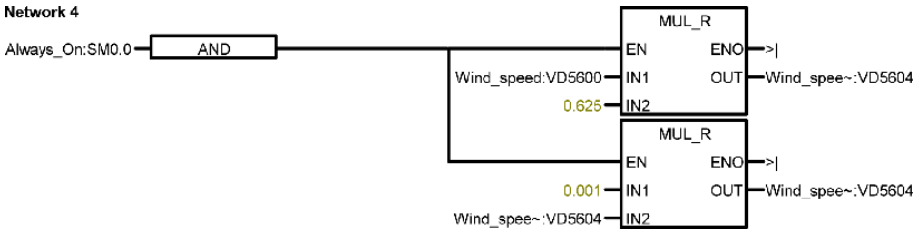
Stejná instrukce je provedena ještě v následujícím network pro druhý analogový vstup, který představuje směr větru. Pak již následuje převod výstupního proudu z rozmezí 4 mA až 20 mA na rychlost větru od 0 m/s do 30 m/s. Na obr. 2.41 je dvojice instrukcí, která zajišťuje přetypování.



Obr. 2.41: Anemometer – přetypování hodnoty ze vstupu na reálné číslo

První blok `I_DI` přetypovává hodnotu z analogového vstupu v datovém typu `Integer` na `Double Integer`. Druhý blok pak ukládá již přetypovanou hodnotu v datovém typu `Real`.

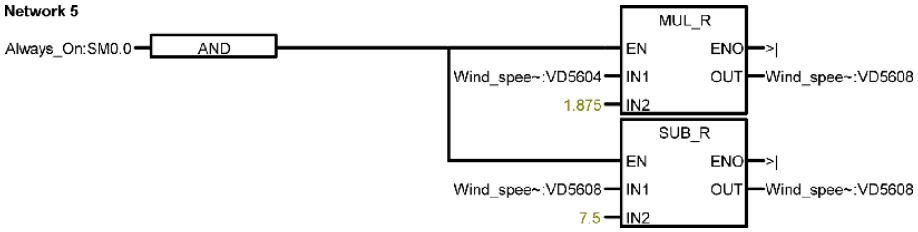
Následující network je zobrazen na obr. 2.42. Program dále kopíruje postup přepočtu hodnoty z ADC převodníku na rychlost větru.



Obr. 2.42: Anemometer – převod hodnoty z ADC na proud

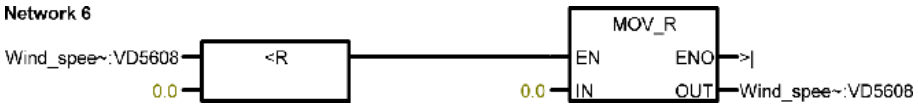
Do dvojic instrukcí `MUL_R`, je pro větší přesnost rozděleno násobení konstantou do dvou bloků. Celkově tento network zajišťuje převod hodnoty z analogového vstupu automatu na proud, který je popsán vztahem 2.21.

Výpočet pokračuje realizací rovnice 2.28, která zajišťuje převod proudu získaného z anemometru na rychlost větru v m/s. Bloky realizující tuto funkci jsou zobrazeny na obr. 2.43. Nejprve je hodnota proudu vynásobena konstantou pomocí instrukce `MUL_R` a následně je od této hodnoty odečtena konstanta příkazem `SUB_R`.



Obr. 2.43: Anemometer – převod proudu na rychlost větru

Poslední instrukcí v tomto bloku je ošetření záporných hodnoty rychlostí větru, viz obr. 2.44. Porovnávací instrukce `<R` zjistí, že je rychlost větru nižší než nula a vykoná příkaz `MOV_R`, který přesune do příslušného místa v paměti nulovou hodnotu.

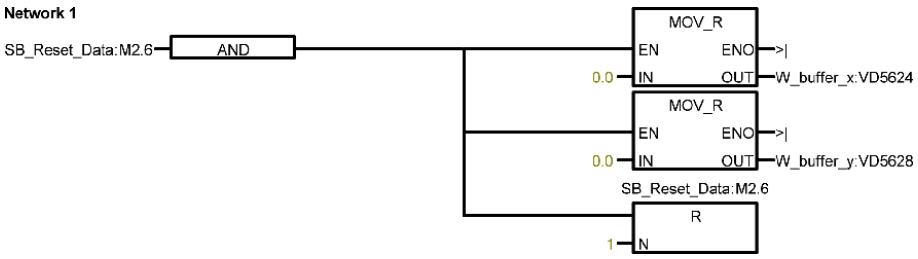


Obr. 2.44: Anemometer – ošetření záporných hodnot rychlosti větru

Od obr 2.41 až po 2.44 se budou použité instrukce opakovat ještě jednou i dále. Všechny tyto bloky jsou totiž použity pro převod druhé výstupní hodnoty z anemometru na směr větru ve °. V instrukcích budou ovšem použity jiné konstanty.

Obsluha přerušení

V obsluze přerušení dochází hned v prvním network k vynulování paměťových míst, která v sobě ukládají hodnotu průměrného větru rozloženého do os x a y, to je zobrazeno na obr. 2.45. Na povolovacím vstupu instrukce `AND` je umístěn bit, který je při dokončení detekce nastaven na úroveň `TRUE` a tak dojde k proběhnutí této části kódu ihned po rozjetí lodi směrem k dalšímu kotvišti. Proměnné jsou nulovány funkcemi `MOV_R`. Nakonec je resetován i bit `SB_Reset_Data`. Z toho vyplývá, že jsou tyto instrukce vykonány pouze jednou.

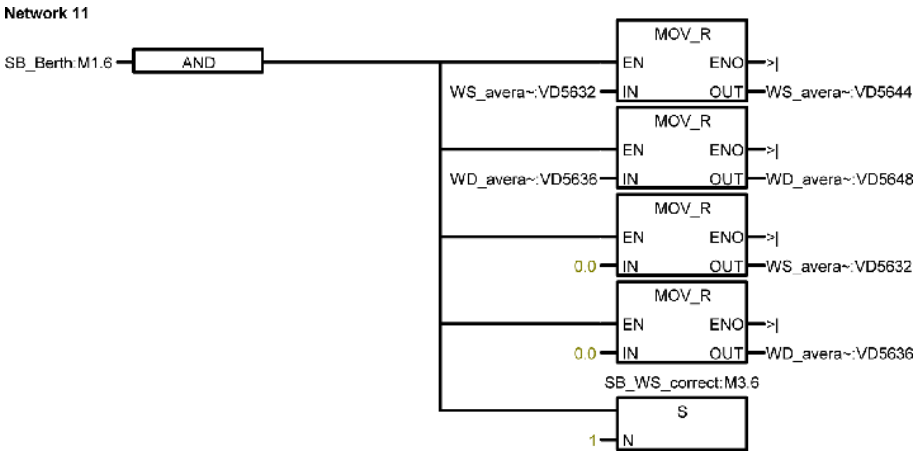


Obr. 2.45: Obsluha přerušení – nulování příspěvků průměrného větru

Nejdůležitější částí týkající se výpočtu větru v obsluze přerušení je network, ve kterém je pravidelně v každém přerušení nastavován bit `SB_Wind_comp`, který signalizuje displeji start makra pro výpočet průměrné rychlosti a směru větru bez korekce na právě probíhajícím úseku plavby.

Dokončení detekce

V podprogramu dokončení detekce se nachází další velice významná část kódu pro výpočet průměrné rychlosti a směru větru. Ta je vykonána pouze při detekci koncového kotviště úseku plavby. V první části je pomocí instrukcí `MOV_R` uložena rychlost a směr větru na právě skončeném úseku do paměti procesoru, pro možnost zobrazení hodnot na předcházejícím úseku plavby bez korekce. Tyto hodnoty jsou pak z původních paměťových míst vymazány, aby uvolnili prostor pro další údaje na následujícím úseku trasy. Poslední instrukce `SET` slouží k nastavení bitu `SB_WS_correct` na úroveň `TRUE`. Signalizační bit spouští v displeji makro, které obsluhuje výpočet korekce průměrné rychlosti a směru větru o průměrnou rychlost lodi na právě skončeném úseku trasy. Celková část kódu je vykreslena na obr. 2.46.



Obr. 2.46: Dokončení detekce – uložení, nulování větru a korekce

2.7.4 Vizualizace rychlosti a směru větru

Základem této obrazovky je dvojice trojice maker, která jsou vykonávána v definovaných okamžicích nebo je jejich výkon spouštěn stisknutím tlačítka.

Makro pro výpočet průměrné rychlosti a směru větru

Část kódu tohoto makra je umístěn v příloze A.2. Výkon instrukcí makra kopíruje výpočet průměrné rychlosti a směru větru, který byl popsán v kapitole 2.7.2.

Výpočet začíná deklarací proměnných v paměti. Dalším krokem je načtení aktuální rychlosti v m/s a směru větru ve stupních instrukcemi `GetData`. Následující čtyři instrukce realizují rozložení aktuálního větru na příspěvky od základních os podle vztahů 2.30 a 2.31. Tento výpočet je rozdělen na určení hodnoty goniometrické funkce úhlu a vynásobení příslušnou rychlostí větru.

Z paměti programovatelného automatu jsou načteny hodnoty bufferů příspěvků od osy x a osy y pomocí dvojice instrukcí `GetData`. Do hodnot bufferů se přičtou příspěvky od základní os aktuálního větru a buffery jsou uloženy zpátky do paměti PLC příkazy `SetData`. Další instrukcí `GetData` je načteno počet sekund aktuální plavby. Hodnoty v jednotlivých bufferech jsou vyděleny počtem sekund plavby, což odpovídá počtem průměrovaných vzorků větru. V teoretickém výpočtu je tato část kódu reprezentována vztahy 2.32 a 2.33.

Pokračuje instrukcemi realizujícími vztah 2.34, který určuje průměrnou rychlost větru bez korekce. Tato hodnota je uložena do paměti příkazem `SetData`. Dále je vypočten směr instrukcí `ASIN`, podle vztahu 2.35. Jelikož instrukční soubor maker displeje neobsahuje funkci pro výpočet absolutní hodnoty, je tato matematická operace realizována umocněním na druhou a následným odmocněním hodnoty směru větru, která má být převedena do absolutní hodnoty.

Dalším krokem ve výpočtu jsou podmínky, které zaručují správné určení směru větru. Nejprve se pomocí instrukcí `IF - END IF` porovnávají hodnoty odpovídající příspěvkům od základních os k průměrnému větru s nulovou hodnotou. Jednotlivé podmínky jsou popsány vztahy 2.36, 2.37, 2.38 a 2.39. Další dvojice podmínek `IF` zajišťuje přiřazení hodnoty úhlu v místě, kde je funkce `ASIN` nedefinována, tedy že příspěvek od jedné nebo druhé ze základních os je nulový. To je popsáno podmínkami 2.40, 2.41, 2.42 a 2.43.

Na konci kódu makra dochází k uložení průměrného směru větru do paměti PLC a k vynulování signalizačního bitu `SB_Wind_comp`, který spouští celý tento výpočet.

Makro – korekce průměrné rychlosti a směru větru

Korekce průměrných parametrů větru je vypočítána makrem, které je spuštěno bi-

tem `SB_WS_correct`. Toto makro má téměř stejný sled instrukcí jako makro pro výpočet průměrných hodnot větru zobrazené v příloze A.2.

Makro začíná deklarací a načtením hodnot z paměti. Navíc je načtena z paměti hodnota uražené vzdálenosti na předchozím úseku trasy pomocí instrukce `GetData`. Dále je výkon makra rozšířen o následující dva řádky kódu

```
...

vessel_speed = (distance*1000)/voyage_time_s
SetData(vessel_speed, "SIEMENS S7/200 (Ethernet)", VD, 5656, 1)

x_avr_C = x_avr - vessel_speed
pom = x_avr_C*x_avr_C + y_avr*y_avr

...
```

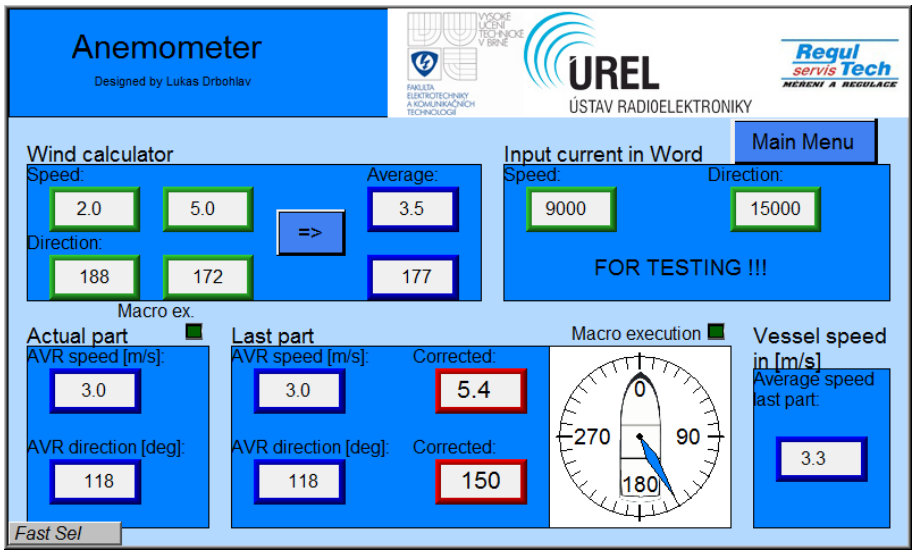
První řádek odpovídá výpočtu průměrné rychlosti lodi na předchozím úseku. Aby bylo dosaženo větší přesnosti, dochází k výpočtu až v cílovém kotvišti aktuální trasy plavby. Tato hodnota je uložena do paměti řídícího automatu. Na třetím řádku výpočet pokračuje korekcí průměrného příspěvku od osy x o průměrnou rychlost lodi podle vzorce 2.44. S použitím takto korigované hodnoty se následně počítá průměrná rychlost a také směr. Průměrný směr je na konci makra opět podroben podmínkám, tak aby bylo jednoznačně správně určeno o jaký výsledný směr se jedná.

Poslední dvojice instrukcí v makru slouží k vynulování signalizačních bitů. Jedná se o instrukce `SetData` a dochází k nulování bitů, které spouští výpočet průměrné rychlosti a směru větru v sekundových intervalech a také korekci průměrných hodnot větru. Jsou to bity se symbolickými názvy `SB_Wind_comp` a `SB_WS_correct`.

Obrazovka zajišťující indikaci parametrů souvisejících s výpočtem průměrné rychlosti a směru větru je zobrazena na obr. 2.47.

Upravené makro, které počítá průměrnou rychlost a směr větru, je použito v aplikaci Wind Calculator, která je umístěna v levém horním rohu obrazovky. Toto makro nepracuje s žádnými paměťovými místy v PLC, pouze s pamětí displeje. Tato aplikace umožňuje pro dva větry, jehož hodnoty budou zadány prostřednictvím `Numeric Input Object`, vypočítat průměrnou rychlost a směr výsledného větru. Výsledné hodnoty jsou zobrazeny blokem `Numeric Display Object`.

V pravém horním rohu jsou umístěny dva bloky `Numeric Input Object`, které slouží pro zadávání hodnot, jaké jsou získány z ADC převodníku analogového vstupu automatu. Takže ručně zadaná hodnota je podle vztahů zmíněných v kapitole 2.7.1



Obr. 2.47: Testovací vizualizační software – rychlost a směr větru

přepočítána na odpovídající rychlost a směr větru. Převedené hodnoty jsou pak zobrazeny na obrazovce s nastavením v testovacím vizualizačním softwaru viz. 2.3.

Pomocí makra dochází k automatickému průměrování rychlosti a směru větru v průběhu plavby. Zobrazované hodnoty objektem `Numeric Display` jsou umístěny na obrazovce v levém dolním rohu, jedná se o průměrné hodnoty větru bez korekce na aktuálním úseku plavby.

Ve spodní části obrazovky uprostřed jsou zobrazeny hodnoty větru na předchozím úseku trasy. V modrých blocích `Numeric Display Object` jsou po detekci koncového kotviště daného úseku trasy uloženy hodnoty průměrné rychlosti a směru větru na předchozím úseku bez korekce. Hned vedle jsou bloky zobrazující rychlost a směr větru na posledním úseku trasy po korekci o průměrnou rychlost lodi. Korekce je provedena vykonáním makra po připlutí do koncového kotviště daného úseku. V tomto makru pro korekci vypočtena průměrná rychlost lodi na předchozím úseku. Rychlost lodi v m/s je zobrazena v bloku hned vedle.

Poruchy

V rámci vizualizace podprogramu počítajícího průměrnou rychlost a směr větru byl vytvořen `Event Display Object`, který signalizuje poruchové události. Tento objekt se dvěma poruchovými stavy je zobrazen na obr. 2.36, který zobrazuje vizualizační plochu podprogramu pro výpočet vzdálenosti v kapitole 2.6.3.

Poruchy zobrazované tímto displejem se musejí v EasyBuilderu zadat do nastavujícího `Alarm (Event) Log`. Zde je možné nastavit bit, na který displej reaguje zobrazovanou zprávou v `Event Display Object`.

Výsledek tohoto nastavování je, že v případě nastavení signalizačního bitu automatem dojde ke zobrazení zprávy s červeným fontem. Pokud porucha odezní a bit změní svou hodnotu na úroveň `FALSE` text se zprávou o poruše zezelená, ale zůstane zobrazen. Tím je zajištěna indikace nápravy poruchového stavu a zároveň paměť stavů, které během plavby nastaly.

V souvislosti s výpočtem průměrné rychlosti a směru větru dochází k výpisu poruchového stavu pouze v momentě kdy je přerušená proudová smyčka anemometru. Ať jedna nebo druhá (rychlost/směr).

2.8 Ukládání naměřených a vypočtených hodnot

Nedílnou součástí programu pro vyhodnocení spotřeby osobní lodi je automatické ukládání naměřených a vypočítaných dat na paměťové médium. Provozní hodnoty tak mohou být dále zpracovány a vyhodnocovány pracovníkem Dopravního podniku.

Ukládání zajišťuje prvek `Data Sampling Object`. Spouštění ukládání dat reaguje na sestupnou hranou bitu `SB_Wind_correct`, který informuje displej, kdy vypočítat korekci rychlosti a směru větru. Po skončení makra s výpočtem korekce je tento bit resetován do úrovně `FALSE`. Dále se nastavuje počáteční paměťové místo od kterého jsou hodnoty ukládány, datový typ, počet hodnot a maximální počet uložení za den. Jako úložné médium je nastaven USB1, což znamená první USB paměťový disk, který je k displeji připojen. Doba zachování je stanovena na 365 dní.

Makro – signalizace nedostatku paměti pro ukládání

Toto makro je spuštěné na pozadí všech obrazovek aplikace. Zajišťuje správné nastavení bitů, které spouští, zavírají, nebo potvrzují vyskakovací okno signalizující nedostatek paměti na USB pro ukládání dalších hodnot.

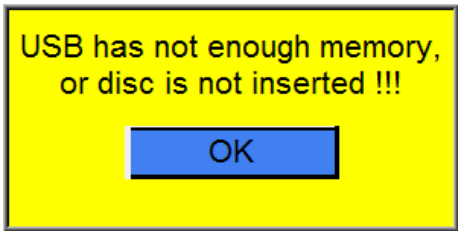
```
macro_command main()
int a
bool bit0 = 0, bit1 = 1
bool accept, error, disable

GetData(a, "Local HMI", LW, 9076, 1)
GetData(accept, "Local HMI", LB, 330, 1)

if a == 0 then
   SetData(bit1, "Local HMI", LB, 320, 1)
   else
   SetData(bit0, "Local HMI", LB, 320, 1)
```

```
   accept = false
end if

GetData(error, "Local HMI", LB, 320, 1)

if error == 1 and accept == 0 then
   disable = true
end if

if error == 1 and accept == 1 then
   disable = false
end if

SetData(disable, "Local HMI", LB, 340, 1)
SetData(accept, "Local HMI", LB, 330, 1)

end macro_command
```

Na začátku dochází k deklaraci a inicializaci proměnných. Následuje uložení velikosti volného místa na USB1 a také uložení hodnoty stavu bitu, který potvrzuje vyskakovací okno. Oba úkoly jsou provedeny instrukcí `GetData`. V podmínce `IF` je testována velikost paměťového místa. Pokud je nulová dojde k uložení jedničky do bitu `error`, který signalizuje varování. V opačném případě se nastaví bit `error` do nuly a navíc se bit potvrzující varování nastaví na úroveň `FALSE`.

V další dvojici podmínek `IF` je nejprve testováno zda je aktivní bit varování a neaktivní bit potvrzující poruchu. V takovém případě se nastaví bit zobrazující vyskakovací okno do úrovně `TRUE`. Druhá podmínka je platná pokud jsou oba vstupní bity aktivní. To je pak bit, který zajišťuje zobrazení vyskakovacího okna s varováním, nastaven a úroveň `FALSE`.

Na konci následuje uložení bitu, který zobrazuje vyskakovací okno a bitu, který potvrzuje přijetí varování. Vyskakovací poruchové okno je zobrazeno na obr. 2.48.



Obr. 2.48: Testovací vizualizační software – nedostatek paměti na USB

3 REALIZACE A VYHODNOCENÍ

Program pro vyhodnocení spotřeby byl zakomponován do řídícího softwaru celé lodi. Stejně tak tomu bylo i s vizualizací vypočítaných a naměřených hodnot touto aplikací. V této kapitole jsou popsány nezbytné úpravy, které musely být v této souvislosti provedeny, vzhled aplikace přímo na palubě lodi a v neposlední řadě se zde nachází naměřená data při zkušebních plavbách.

3.1 Úprava programu před implementací do PLC

Program pro vyhodnocení spotřeby byl ještě před implementací upraven, tak že byly vymazány nadbytečné části, které sloužili pouze pro simulaci a paměťová místa byla upravena tak, aby odpovídala přiřazenému rozsahu v řídícím softwaru.

V hlavní smyčce bylo nutné odebrat instrukce `JMP` a `LBL`, které v simulaci zajišťují přeskok vyhodnocení pohybu lodi. Dále musel být změněn časovač pro ochranný interval a blokování detekce, tak aby nedocházelo ke kolizi s již použitými časovači v řídícím softwaru.

V podprogramu pro inicializaci byly odstraněny network, které souvisely s počátečním nastavením simulačních podmínek. První instrukce, které byly ubrány z celkového kódu jsou patrné z obr. 2.6. Ty zajišťovaly nastavení počátečních souřadnic lodi. Následovala instrukce `SET`, která nastavovala bit `Speed_0`, což je bit zajišťující klidový stav lodi po spuštění PLC. Počáteční nastavení napětí a proudu do motoru viz. obr. 2.24 bylo také odstraněno. A nakonec bylo smazáno i nastavení vstupních hodnot z anemometru instrukcemi na obr. 2.39.

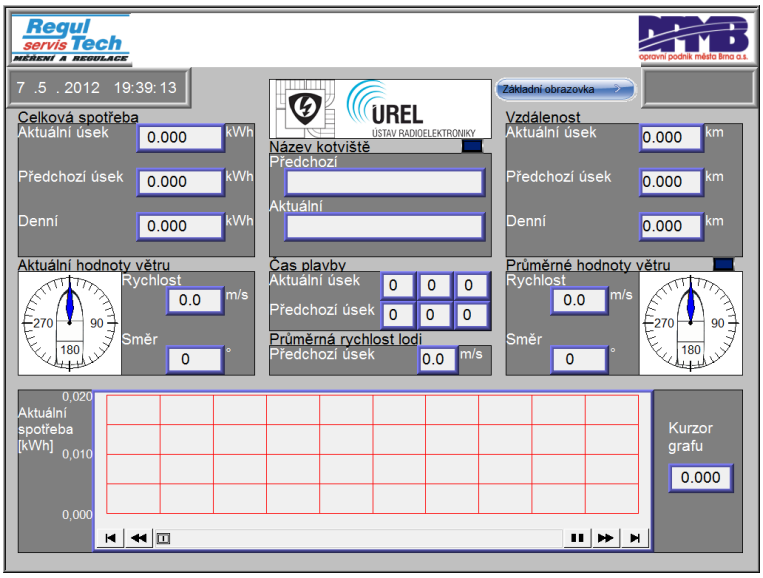
Paměťová místa, která nahrazovala analogové vstupy řídícího automatu v instrukcích převodu proudu na rychlost a směr větru, byla zaměněna skutečnými analogovými vstupy automatu. Nastavení pro simulaci bylo patrné z obr. 2.40 a 2.41.

Poslední změnou bylo přepsání paměťových míst, která na obr. 2.26 spouštějí výpočet spotřeby pro hlavní frekvenční měnič. Tyto proměnné byly nahrazeny digitálními vstupy, které jsou připojeny k frekvenčním měničům a signalizují, jaký z nich je aktivní.

Takto upravené bloky programu mohly být zkopírovány do řídícího softwaru lodi. Ještě před tím, ale došlo k vložení tabulky symbolických adres, která zajišťuje možnost pojmenování paměťových míst snadno zapamatovatelnými jmény. Tabulka symbolických adres je k dispozici v příloze A.3.

3.2 Vizualizace vyhodnocení spotřeby

Do již vytvořené vizualizace pro kompletní řídící software lodi byla vytvořena obrazovka indikující důležité informace o vyhodnocení spotřeby. Tato obrazovka je k dispozici posádce, která tak může sledovat aktuální podmínky výpočtu spotřeby. Vizualizace vyhodnocení spotřeby osobní lodi je zobrazena na obr. 3.1. Objekty a makra, které jsou na vizualizaci použity vycházejí přímo z nastavení v testovacím vizualizačním softwaru.



Obr. 3.1: Vizualizace programu pro vyhodnocení spotřeby na lodi Lipsko

V levém horním rohu se nachází část **Celková spotřeba**. Obsažené bloky vycházejí z kapitoly 2.5.2. Posádka zde nalezne hodnoty spotřeby na aktuálním úseku, na úseku předcházejícím a záznam o celkové denní spotřebě. Všechny uvedené hodnoty jsou v kWh.

Vedle spotřeby v části **Název kotviště** jsou dva bloky zobrazující ASCII znaky, ve kterých se objevují názvy detekovaných kotvišť, viz. 2.4.3. Ve spodním bloku je zobrazen název kotviště, ve kterém se loď aktuálně nachází. Horní blok pak ukazuje název kotviště, které bylo detekováno, ale loď z něho již vyplula. Modrá dioda nad tímto blokem signalizuje blokování detekce lodi v kotvišti. V momentě kdy je rozsvícena, nemůže být detekováno kotviště. Po 2 min je blokující bit vynulován a detekce probíhá opět normálně. Toto opatření slouží proti několikanásobným detekcím.

Pod názvy kotvišť je umístěn **Čas plavby**. Jedná se o čistý čas plavby, který se inkrementuje pouze za předpokladu, že má loď rychlostní páku v jiné než nulové poloze. První údaj odpovídá čistému času aktuálního úseku plavby a údaj pod ním

času plavby na úseku předcházejícím. Výpočet času je součástí kapitoly o vizualizaci vzdálenosti 2.6.3.

Ještě pod názvy kotvišť je možné nalézt údaj **Průměrná rychlost lodi**. Je to hodnota, která je vypočítána z celkové vzdálenosti uražené na předchozím úseku a času plavby na tomto úseku. Tato hodnota potom slouží ke korekci rychlosti a směru větru. Výpočet této hodnoty vychází z kapitoly 2.7.4.

Na stejné úrovni jako předchozí části, ale úplně na levé straně obrazovky se nachází část **Vzdálenost**. Bloky, které se zde vyskytují jsou blíže vysvětleny v kapitole 2.6.3. Zobrazují vzdálenost uraženou lodí na aktuálním úseku trasy, na úseku předcházejícím a konečně celkovou denní uraženou vzdálenost. Všechny hodnoty jsou vypisovány v km.

V prostřední části nejvíce vlevo je blok **Aktuální hodnoty větru**. Ten obsahuje dva objekty, které vypisují rychlost a směr aktuálního větru. Hodnota směru je pak navíc znázorněna kruhovým indikátorem, pro lepší orientaci směru větru vzhledem k lodi. Na opačné straně se nachází blok **Předchozí hodnoty větru**. Také jsou zde bloky vypisující rychlost a směr větru ovšem na předchozí trase upravené o korekci průměrnou rychlostí lodi. Směr je stejně jako v předchozím případě indikován kruhovým indikátorem. Údaje jsou zobrazeny stejně jako v testovacím vizualizačním softwaru v kapitole 2.7.4. Modrá dioda zobrazuje bit signalizující průměrování hodnot větru bez korekce, při plavbě je tedy zapínána a vypínána se střídou 1 s.

Celou spodní polovinu obrazovky zaujímá grafické znázornění měnící se aktuální spotřebu v kWh v čase. Více je popsáno v kapitole 2.5.2. Nový bod do grafu je vzorkován s periodou 2 s nebo 30 s. To závisí na rozdílu aktuální spotřeby ve dvou po sobě jdoucích vzorcích. Na pravé straně je navíc blok, ve kterém je zobrazena hodnota kurzoru grafu.

3.3 Postup odladění

Samotné programování a ladění softwaru přímo na lodi probíhalo v několika dnech. První den byla na zkušební plavbě odladěna první verze programu pro vyhodnocení spotřeby. Druhý den byl tento software implementován i do ostatních lodí na Brněnské přehradě. Následovalo několikatýdenní sbírání údajů z pravidelných linkových plaveb lodí Dopravního podniku. Po těchto dvou týdnech došlo k vyhodnocení dat, zjištění a úpravě nedostatků.

Nejprve byla do lodi nahrána první verze programu. Následoval testovací okruh po celé Brněnské přehradě, aby bylo možné odečíst přesně hodnoty GPS souřadnic jednotlivých kotvišť. V průběhu plavby se vyskytl problém s výkonem makra v displeji, který zabraňoval pravidelnému výpočtu uražené vzdálenosti lodí a tedy

i výpočtu průměrné rychlosti lodi na předchozím úseku trasy. Tento problém byl odstraněn změnou podmínky pro spuštění všech maker v programu vizualizace, tak aby makro reagovalo na úroveň nikoli pouze na hranu spouštěcího bitu. Pokud makro reagovalo na hranu, došlo pouze k jeho částečnému výkonu. V momentě, kdy displej potřeboval vykonat jinou operaci s vyšší prioritou přiřadil se procesor této události, ale podmínka pro spuštění makra již odezněla a tak se procesor k dokončení makra už nevrátil.

Po první zkušební plavbě byly do řídícího systému zadány skutečné souřadnice jednotlivých kotvišť a tento software byl implementován i do ostatních lodí. Následoval několikatýdenní zkušební provoz, který měl ukázat chyby a nedostatky programu pro vyhodnocení spotřeby.

Při následujícím programování, byla stažena data z jednotlivých lodí a vyhodnocena. Naměřená data ukázala, že docházelo k několikanásobným detekcím kotviště vlivem přirážení lodi k pontonu. Odstranění této závady si vyžádalo vložení intervalu blokujícího detekci po dobu několika minut od úspěšné detekce kotviště. To je popsáno v kapitole 2.4.2 a zejména znázorněno na obr. 2.12.

Dalším nedostatkem je občasná chyba ukládané hodnoty. Hodnoty jsou ukládány procesorem displeje na paměťové médium. Ovšem pokud je v momentě, kdy jsou hodnoty zapisovány do paměti rychlostní páka nastavena do jiné polohy než nulové dojde v automatu ke spuštění dalšího cyklu měření hodnot mezi následujícími kotvišti. A tak hodnoty, které ukládá displej do paměti jsou vynulovány řídícím systémem. Proto je možné, že se občas uloží nesmyslná hodnota. Tento problém lze řešit uložením hodnot zapisovaných na paměťové médium do jiné oblasti v paměti, tak aby nedocházelo k jejich přepisování automatem.

Tento den byla také vyzkoušena funkce zapůjčeného anemometru. Ten byl provizorně přidělán na tyči k zábradlí lodi. A v rámci jednoho testovacího okruhu byla odladěna funkce měření a ukládání průměrné rychlosti a směru větru a také uložena naměřená data. Podle těchto hodnot dojde k ustanovení, zda se anemometr pořídí na každou loď.

3.4 Naměřená data

V této části diplomové práce jsou shromážděny grafické závislosti měřených veličin na úseku plavby lodi. Data byla naměřena v období od 6. 4. 2012 do 3. 5. 2012. Z nich byly vybrány hodnoty pro grafické znázornění změny měřených veličin. Na všech grafech jsou zobrazena ID, která jsou přiřazena jednotlivým kotvištím v tab. 2.1.

Hodnoty z denního záznamu o spotřebě jsou graficky zobrazeny na obr. 3.2. Údaje byly naměřeny 1. 5. 2012 na lodi Dallas. Z osy x je patrné, že tyto hodnoty byly

naměřeny ještě před použitím blokovacího intervalu, protože řada kotvišť zde byla detekována vlivem přistávání k pontonu několikrát. Dále lze pozorovat, že ID kotvišť nejdou popořadě. Tento záznam byl pořízen během linkové plavby, při které by měla loď zastavit ve všech stanicích, ale v momentě, kdy na pontonu některého z kotvišť nečekají žádní cestující, se kotvištěm pouze proplouvá a tak není detekováno. Údaj vždy indikuje spotřebu, kterou měla loď na úseku trasy končícím v daném kotvišti.

Po vložení blokovacího intervalu do řídícího systému, viz. kap. 2.4.2 a obr. 2.12, byly při testovacím okruhu s lodí Vídeň dne 3. 5. 2012 naměřeny nové hodnoty spotřeby. Na obr. 3.3 lze pozorovat, že bylo skutečně každé kotviště detekováno pouze jednou. Z tohoto obrázku je také patrné, že cesta do posledního kotviště probíhala se zastávkami, zatímco zpáteční cesta do Bystrce byla přímá – bez zastávek. Nepřítomnost prvních kotvišť je vysvětlena tím, že během této části testovací plavby byl poopraven kód a tak přehrán vizualizační software v displeji i řídící software v automatu.

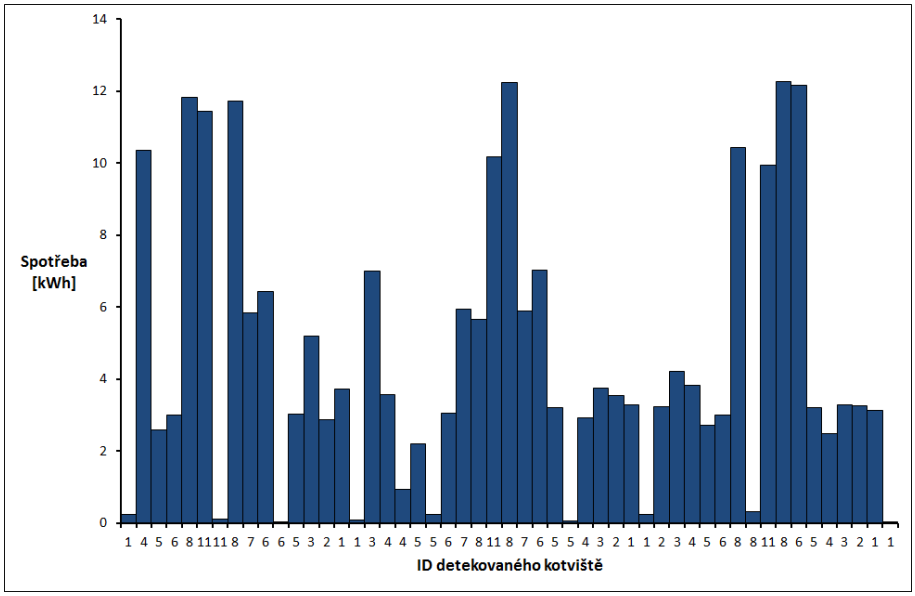
Z naměřených dat stažených z lodí byly vybrány porovnatelné údaje o spotřebě tří lodí. Podmínkou bylo, že údaje musely být nasbírány ve stejný den. Dále museli být lodi na linkové trase přibližně ve stejný čas, tak aby byly zajištěny podobné podmínky při měření. Úplně stejného časového úseku nelze docílit, protože lodi na linkové plavbě vyráží s půlhodinovými rozestupy. Taková shoda byla nalezena například v datech stažených 1. 5. 2012 přibližně mezi 11 a 16 hodinou. Srovnání spotřeby těchto tří lodí je zobrazeno na obr. 3.4.

Údaje spotřeby v prvním kotvišti odpovídají úseku z kotviště druhého do prvního při návratu lodi z předchozího linkového okruhu. Z grafického znázornění je patrné, že spotřeby jednotlivých lodí jsou srovnatelné. V případě, že má v nějakém kotvišti jedna loď spotřebu dramaticky vyšší než další dvě, je to způsobeno průjezdem jedné nebo více zastávek. Jiná situace nastává při detekci prvního kotviště po poslední zastávce ve Veverské Bítýšce. Do této spotřeby se započítává i energeticky velmi náročné otáčení lodi, které probíhá téměř na místě a tak jsou údaje o spotřebě vyšší a rozdíly mezi jednotlivými loděmi mohou být znatelnější.

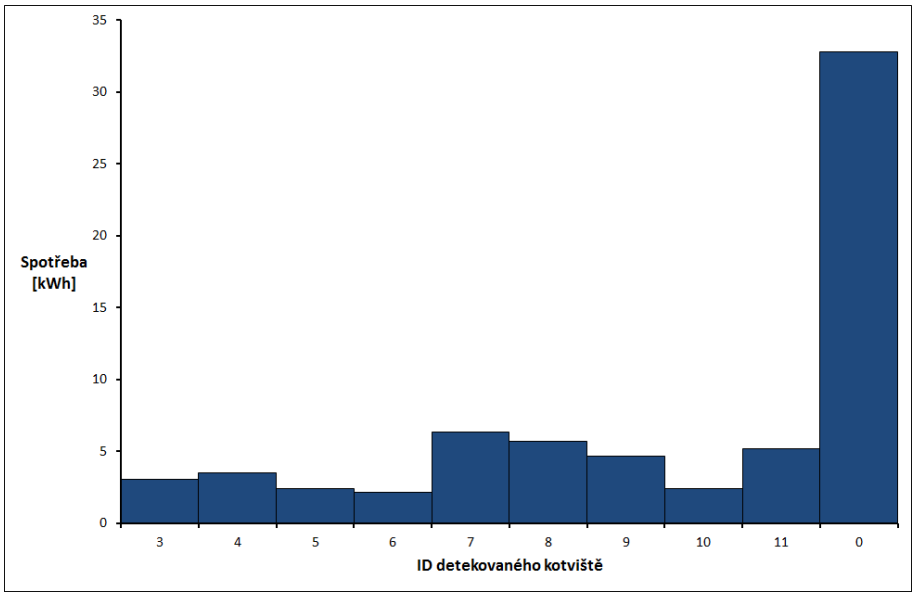
Pro vyhodnocení spotřeby byla měřena a počítána řada pomocných veličin. První takovou veličinou je výpočet vzdálenosti uražené lodí mezi jednotlivými kotvišti. Tato hodnota v praxi závisí na trajektorii, kterou loď mezi jednotlivými zastávkami zvolí. Není zcela konstantní, ale její hodnota se mění pouze v řádu desítek až stovek mětrů. Pro představu vzdáleností naměřených mezi stanicemi při linkové plavbě je na obr. 3.5 znázorněna tato závislost. Ta byla naměřena na lodi Stuttgart dne 1. 5. 2012.

Z naměřené vzdálenosti a doby plavby mezi jednotlivými kotvišti je v každém kotvišti počítána průměrná rychlost lodi. Grafické znázornění průměrné rychlosti lodi na úseku trasy je na obr. 3.6. Data byla stažena 1. 5. 2012 z lodi Stuttgart.

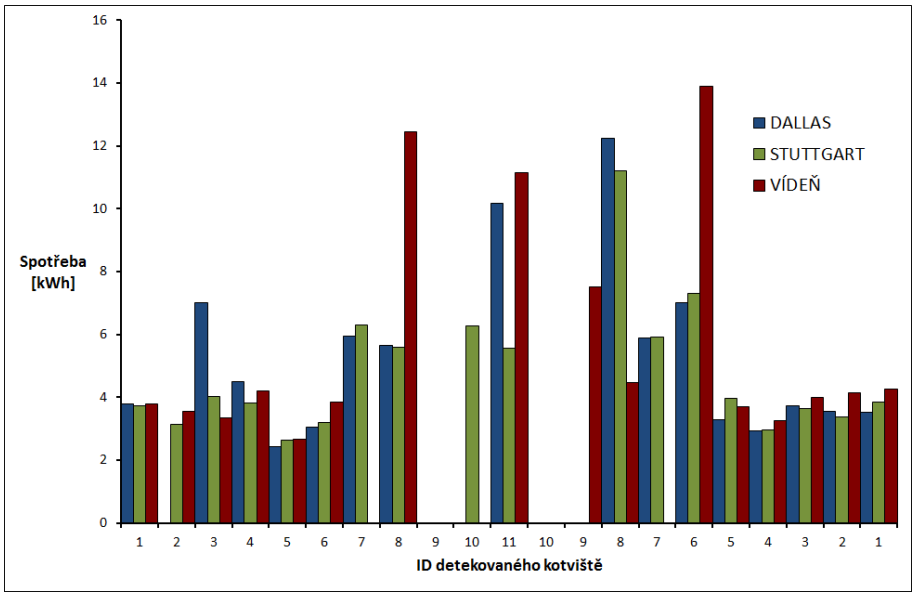
Poslední dvojice grafů je spjata s měřením větru. Dne 3. 5. 2012 byl zapůjčen anemometr, se kterým byl absolvován testovací okruh na lodi Vídeň. Na obr. 3.7 je zobrazena změna průměrné rychlosti větru s korekcí podél okruhu plavby a na obr. 3.8 potom průměrný směr větru s korekcí. Největší rychlosti nabýval vítr mezi stanicí Rokle a Cyklistická, kde podle dlouhodobých zkušeností kapitánů skutečně vítr fouká nejvíce a tudíž je největší problém s lodí u pontonu přistát. Vzhledem k tomu, že se celý měřící den vítr na ploše přehrady velmi otáčel, je z grafické závislosti směru větru patrné, že průměrný vítr foukal po dobu plavby směrem k Veverské Bítýšce lodi do zad. Naproti tomu při návratu zpátky do kotviště Bystrc–Servis foukal celou cestu proti vítr, což je patrné z posledního sloupce grafu.



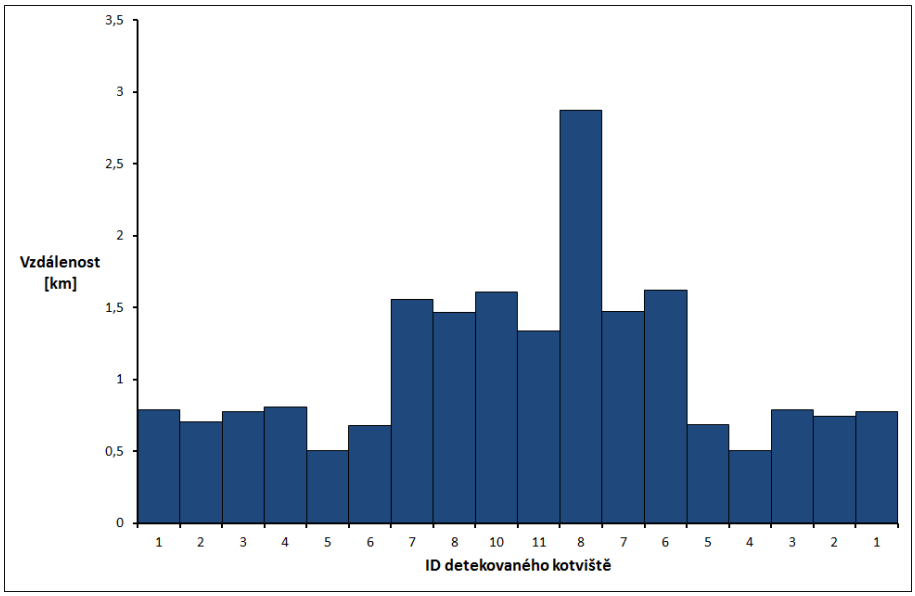
Obr. 3.2: Denní záznam o spotřebě – DALLAS (1.5.2012)



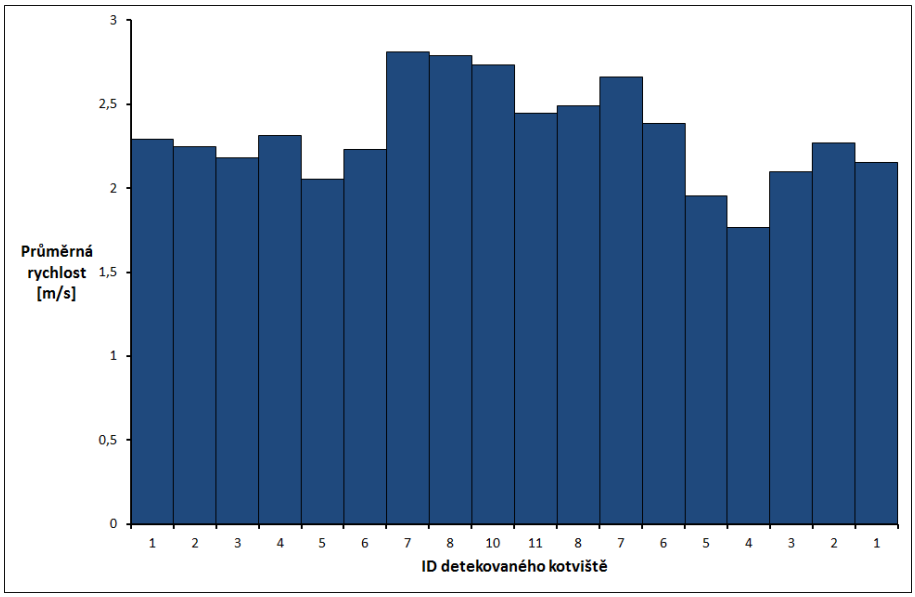
Obr. 3.3: Záznam o spotřebě z testovací plavby – VÍDEŇ (3.5.2012)



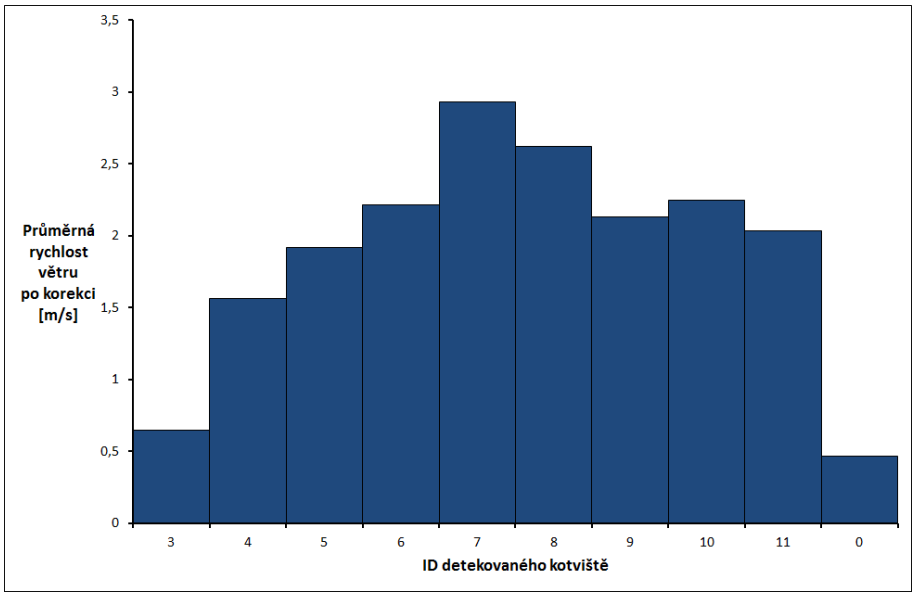
Obr. 3.4: Porovnání spotřeby – DALLAS, STUTTGART a VÍDEŇ (1.5.2012)



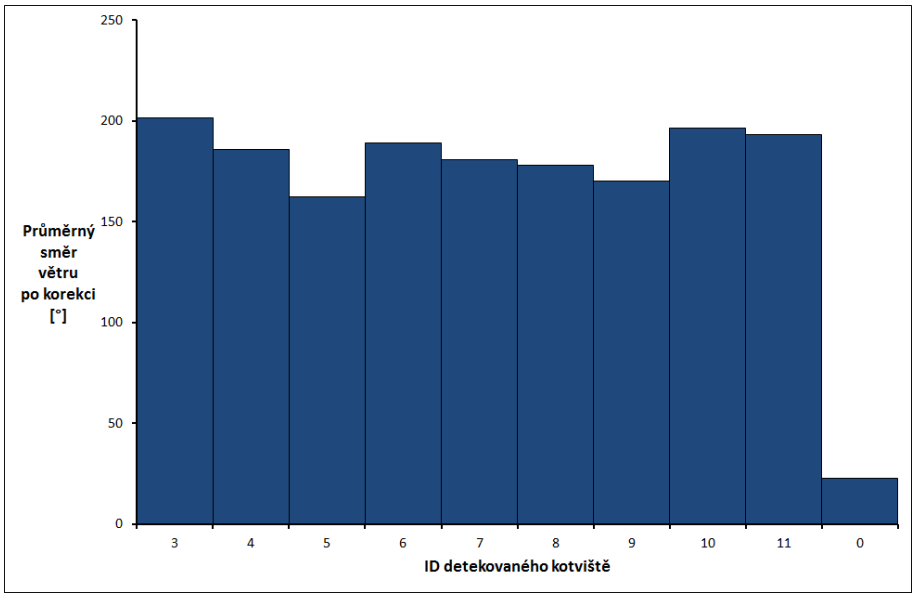
Obr. 3.5: Změřená vzdálenost mezi kotvišti – STUTTGART (1.5.2012)



Obr. 3.6: Změřená rychlost mezi kotvišti – STUTTGART (1.5.2012)



Obr. 3.7: Změřená rychlost větru mezi kotvišti – VÍDEŇ (3.5.2012)



Obr. 3.8: Změřený směr větru mezi kotvišti – VÍDEŇ (3.5.2012)

4 ZÁVĚR

Úvod zadání diplomové práce, seznámit se s programovatelnými automaty a prostředím pro vývoj jejich programů, byl splněno v rámci první kapitoly. Zde byly podrobně rozebrány typy programovatelných automatů, jejich parametry a funkce. V následujících podkapitolách potom přišlo na řadu vývojové prostředí od firmy Siemens STEP7/MicroWIN a popis normy definující programovací jazyky pro PLC. V závěru teoretické části byl zmíněn i dotykový displej Weintek Easy View MT8150X, který zajišťuje vizuální přehled o stavu lodi pro posádku a vývojové prostředí pro vizualizaci EasyBuilder EB8000.

Dále byl vytvořen software pro PLC Simatic, který zajišťuje vyhodnocení spotřeby osobní lodi. Tento program zajišťuje detekci lodi v kotvišti, protože díky tomu může být výpočet spotřeby na daném úseku trasy naprosto automatický a nezávislý na obsluze. V případě, že loď stojí na místě, má minimální otáčky a rychlostní páku v nulové poloze je považována za stojící. Po uplynutí čekacího časového intervalu následuje porovnání GPS souřadnic, na kterých se loď právě nachází s bankou mezních souřadnic všech kotvišť a podle toho se určuje, zda se loď nachází v kotvišti. Rozšíření algoritmu detekce spočívá v doplnění o blokovací časovač, který aktivuje a deaktivuje bit, zabraňující proběhnutí úspěšné detekce určitou dobu po předchozí úspěšné detekci. To odstraňuje mnohonásobné detekce a tím i ukládání falešných dat do zálohovacího souboru na paměťovém médiu.

Program dále každou sekundu vzorkuje napětí a proud odebírané asynchronním motorem lodi a převádí ho na hodnotu spotřeby v kWh. Součástí je i signalizace startu výpočtu vzdálenosti, výpočtu průměrné rychlost a směru větru a výpočet korekce průměrné rychlosti a směru větru o průměrnou rychlost lodi.

Rozšířením programu v řídícím systému lodi, bylo vytvoření testovacího vizualizačního softwaru pro displej MT8070iH, který simuloval spolupráci dotykového panelu a programovatelného automatu. V rámci této simulace byly navrženy a odladěny funkce jednotlivých částí programu, které z důvodu nižší přesnosti nemohly být implementovány do procesoru Simatic. Jedná se o výpočet vzdálenosti a průměrné rychlosti a směru větru. Výpočet uražené vzdálenosti probíhá s periodou 2 nebo 30 s, podle toho jaký je rozdíl po sobě jdoucích aktuálních hodnot spotřeby motoru. V ten moment jsou uloženy aktuální souřadnice lodi a spolu se souřadnicemi předchozího bodu měření vytvoří úsečku jejíž délka je vypočtena algoritmem pro výpočet vzdálenosti.

Algoritmus pro měření rychlosti a směru větru je ve finálním programu nahraném v lodích obsažen, ale na lodích prozatím nejsou k dispozici anemometry, které by tyto veličiny měřily. V případě připojeného čidla pro měření rychlosti a směru větru dochází v každém cyklu automatu k převodu proudu přivedeného ze senzoru

na analogový vstup PLC na hodnotu dané veličiny. Vzorkování rychlosti a směru větru pro průměrování probíhá s periodou 1 s. V koncovém kotvišti daného úseku trasy se pak vypočtou korigované hodnoty o průměrnou rychlost lodi.

Takto připravené softwary byly implementovány do řídícího automatu S7-200 a také do dotykového ovládacího panelu Easy View MT8150X. Obrazovka v panelu byla upravena, tak aby zobrazovala všechny podstatné měřené veličiny. Jedná se tedy o spotřebu, vzdálenost, čas, název aktuálního a naposledy detekovaného kotviště, aktuální rychlost a směr větru, průměrná rychlost a směr větru po korekci a průměrná rychlost lodi na předcházejícím úseku trasy. Ve spodní části je také zobrazen graf zachycující změnu aktuální spotřeby v čase.

Při vyhodnocení naměřených dat během testovacího provozu aplikace pro vyhodnocení spotřeby bylo zjištěno několik nedostatků, které byly odstraněny. Největší změna bylo vložení bitu, který blokuje detekci 2 min po předchozí úspěšné detekci. Tím se zamezilo vzniku několikanásobných detekcí v jednom kotvišti při přistávacím manévru. Dále byly přesunuty některé proměnné, které jsou ukládány na paměťové médium, tak aby nedocházelo k jejich nechtěnému přepisu v průběhu ukládání.

Navržené řešení tedy měří s největší přesností spotřebu. Přesnost uražené vzdálenosti závisí na přesnosti GPS přijímače a zaokrouhlování při výpočtech. Aktuální rychlost a směr větru je zatížena chybou, která vzniká tím, že se měří na pohybující se lodi. Tato chyba je odstraněna až ve výsledných hodnotách pomocí korekce o rychlost lodi.

V rámci testování programů pro vyhodnocení spotřeby a odladění chyb byly programy nahrány i do ostatních čtyř lodí Dopravního podniku města Brna. Tak aby bylo získáno co největší množství dat z linkových plaveb, která mohou být vyhodnocena a z jejichž rozložení lze udělat závěr o funkčnosti navrženého řešení, případně opravit chyby, které tato data odhalí.

LITERATURA

[1] Je tu Lipsko! *Šalina* [online]. 2010, roč. 4, č. 5, s. 3 a 7 [cit. 2012-05-07]. Dostupné z:
<http://www.e-salina.cz/f/salina/p/05%20-%202010/05%20-%202010.pdf>.

[2] ŠTOHL, R.; ZAHRADNÍK, J. Uplatnění AS-Interface v lodní dopravě. *Automa: časopis pro automatizační techniku* [online]. Praha: FCC Public, 2011, roč. 18, č. 3, s. 56-57 [cit. 2012-05-07]. ISSN 1210-9592.
Dostupné z: <http://www.odbornecasopisy.cz/res/pdf/43207.pdf>.

[3] FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ VUT V BRNĚ. *AS-Interface Česká Republika* [online]. Brno, 2003 [cit. 2012-05-07]. Dostupné z: <http://www.as-interface.cz/>.

[4] KLOS, O. Co je systém AS-INTERFACE. *MM Průmyslové spektrum* [online]. Praha: SEND Předplatné s.r.o, 2007, roč. 11, č. 3, s. 40 [cit. 2012-05-07]. ISSN 1212-2572. Dostupné z:
<http://www.mmspektrum.com/clanek/co-je-system-as-interface.html>.

[5] BECKER, R. *AS-Interface Řešení pro automatizaci : příručka, technika, funkce, aplikace*. AS-International Association, 2004. 184 s. ISBN 80-214-2958-5.

[6] ŠMEJKAL, L; MARTINÁSKOVÁ, M. *PLC a automatizace: 1. Základní pojmy, úvod do programování*. 1. vyd. Praha: BEN - technická literatura, 1999, 223 s. ISBN 80-860-5658-9.

[7] ZEZULKA, F. *Prostředky průmyslové automatizace*. 1. vyd. Brno: Nakladatelství VUTIUM, 2004. 176 s. ISBN 80-214-2610-1.

[8] SIEMENS. *SIMATIC: S7-200 Programmable Controller System Manual* [online]. Germany, 2005 [cit. 2012-05-07]. Dostupné z:
<http://www1.siemens.cz/ad/current/content/data_files/automatizacni_systemy/mikrosystemy/simatic_s7200/manual_s7_200_2005_en.pdf>.

[9] URBAN, L. Programování PLC podle normy IEC EN 61131-3 – víc než jednotné jazyky. *Automa: časopis pro automatizační techniku* [online]. Praha: FCC Public, 2005, roč. 12, č. 2 [cit. 2012-05-07]. ISSN 1210-9592. Dostupné z:
<http://www.odbornecasopisy.cz/index.php?id_document=30310>.

[10] WEINTEK LABS., Inc. *MT8150X: X86 CPU Core touch panel computer with 15.0" XGA TFT display* [online]. Taiwan, 2011. [cit. 2012-05-07]. Dostupné z: <http://www.weintek.com/Download/MT8000/eng/DataSheet/X_series /MT8150XV2WK_DataSheet_110512.pdf>.

[11] WEINTEK LABS., Inc. *Easy Builder Pro User's Manual* [online]. Taiwan, 2012 [cit. 2012-05-07]. Dostupné z: <http://www.tecon.cz/pdf/EB8000_User_Manual.pdf>.

[12] ROCKWELL AUTOMATION. *PowerFlex 700 AC Drives: Vector Control Firmware 4.001 & Up, Frames 0...10* [online]. USA, 2011 [cit. 2012-05-07]. Dostupné z: <http://literature.rockwellautomation.com/idc/groups/literature/documents/um/20b-um002_-en-p.pdf>.

[13] ROCKWELL AUTOMATION. *20-COMM-H RS-485 HVAC Adapter: Firmware Version 2.xxx* [online]. USA, 2009 [cit. 2012-05-07]. Dostupné z: <http://literature.rockwellautomation.com/idc/groups/literature/documents/um/20comm-um009_-en-p.pdf>.

[14] MODBUS ORGANIZATION, Inc. *Modbus application protocol specification: V1.1b* [online]. USA, 2006 [cit. 2012-05-07]. Dostupné z: <http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf>.

[15] SIRF TECHNOLOGY, Inc. *NMEA Reference Manual* [online]. USA, 2008 [cit. 2012-05-07]. Dostupné z: <http://www.usglobalsat.com/store/downloads/NMEA_commands.pdf>.

[16] BAGHIRRA S.R.O. *Čidlo rychlosti a směru větru: TM WSD* [online]. Praha, 2011 [cit. 2012-05-07]. Dostupné z: <http://www.baghirra.cz/data/file/stahujTMWSD.pdf>.

[17] WEINTEK LABS., Inc. *MT8070iH : Human Machine Interface with 7" TFT LCD display* [online]. Taiwan, 2011 [cit. 201205-07]. Dostupné z: <http://www.weintek.com/Download/MT8000/eng/DataSheet/i_series/MT-8070iHV2WK_DataSheet_ENG_110607.pdf>.

[18] Seznam: vyhledávání na internetu. *Mapy.cz* [online]. 2002 [cit. 2012-05-07]. Dostupné z: <http://www.mapy.cz/>.

[19] Brněnská přehrada: informace, fotografie, mapa. *Mapa Brněnské přehrady* [online]. Brno, 2011 [cit. 2012-05-07]. Dostupné z: <http://www.brnenskaprehrada.cz/mapa280f.html#m280fd>.

[20] *Understanding GPS: Principles and applications*. Editor Elliot Kaplan. Boston: Artech House, 1996. 554 s. ISBN 0-89006-793-7.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AC Alternate current – střídavá elektrická veličina

ADC Analog to digital converter – analogově digitální převodník

ASCII American Standard Code for Information Interchange – tabulka základních znaků

AS-i Actuator/Senzor interface

CPU Central Processing Unit – centrální procesorová jednotka

DC Direct current – stejnosměrná elektrická veličina

DRAM Dynamic Random Access Memory – Dynamická paměť s náhodným přístupem

EN Enable Input – povolený vstup

ENO Enable Output – povolený výstup

FB Function Block Diagram – programovací jazyk PLC, jazyk logických schémat

FTP File Transfer Protocol – datový síťový protokol

GPRS General Packet Radio Service – datový přenos v mobilních komunikacích

GPS Global Positioning System – globální družicový polohový systém

GSM Global System for Mobile Communications – systém pro mobilní komunikaci

HMI Human Machine Interface – operátorské panely

IL Instruction List – programovací jazyk PLC, jazyk mnemokódů

IN Input – vstup

INT Interruption – přerušení

LD Ladder Diagram – programovací jazyk PLC, jazyk kontaktních schémat

LSB Least Significant Bit (Byte) – nejméně významný bit (byte)

MPI Multi Point Interface – průmyslová sběrnice

MSB Most Significant Bit (Byte) – nejvýznamnější bit (byte)

NTSC National Television System(s) Committee – Norma pro analogové televizní vysílání

OB Organizační blok

OS Operating system – operační systém

OUT Output – výstup

PAL Phase Alternating Line – Norma pro analogové televizní vysílání

PC Personal Computer – osobní počítač

PLC Programmable Logic Controller – programovatelný logický automat

PPI Point to Point Interface – průmyslová sběrnice

SBR Subroutine – podprogram

SD Secure Digital – typ paměťové karty

SDHC Secure Digital High Capacity – typ paměťové karty

SFC Sequential Function Chart – programovací jazyk PLC, jazyk sekvenčního programování

ST Structured Text – programovací jazyk PLC, strukturovaný text

USB Universal Serial Bus – univerzální sériová sběrnice

WWW World Wide Web – internetový protokol