České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra řídící techniky

Diplomová práce

Řízení kalibračního vozíku vodoměrných vrtulí

Praha, 2005 Jakub Hamouz

České vysoké učení technické v Praze - Fakulta elektrotechnická

Katedra řídicí techniky *Školní rok:* 2003/2004

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Jakub Hamouz

Obor: Technická kybernetika

Název tématu: Řízení kalibračního vozíku vodoměrných vrtulí

Zásady pro vypracování:

Kalibrační vozík se pohybuje po kolejích nad kanálem naplněným vodou. Na vozíku jsou pevně přichycené kalibrované vodoměrné vrtule, které jsou ponořené do vody a jejich kalibrace se provádí při pohybu vozíku. Rychlost vozíku je řízena tak, aby byl simulován průtok vody kolem vrtulí.

1. Zvolte HW konfiguraci řídicího počítače, operační systém a vývojové prostředí pro řídicí program.
2. Implementujte komunikaci mezi počítačem na vozíku a měničem frekvence, který ovládá pohon vozíku (asynchronní motor).
3. Vytvořte ovládací program pro řízení vozíku.
4. Realizujte komunikaci vozíku s počítačem na velínu, dále zpracování naměřených dat a jejich odeslání na velín.
5. Řiďte vozík pro konkrétní aplikaci kalibrace vodoměrných vrtulí.

Seznam odborné literatury: Dodá vedoucí práce.

Vedoucí diplomové práce: doc. Ing. Jan Bílek, CSc.

Datum zadání diplomové práce: prosinec 2003

Termín odevzdání diplomové práce: květen 2005

doc. Ing. Michael Šebek, DrSc.
vedoucí katedry

L.S.

v zast.
prof. Ing. Vladimír Kučera, DrSc.
děkan

V Praze dne 25.03.2004

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne ……. …….…….
podpis

Abstrakt

Cílem této práce je popsat řídící systém vyvinutý do nového kalibračního vozíku pro Českou kalibrační stanici v Praze. Tento vozík nahrazuje původní vozík, který byl zničen při povodni 2002. Hlavním úkolem vozíku je v současné době kalibrování vodoměrných vrtulí vlečených ve vodním kanále pod vozíkem. Práce je rozdělena na tři hlavní části popisující jednotlivé úrovně řízení. Nižší úroveň, představovaná dvěma deskami s programovatelnými poli Xilinx, zajišťuje ruční řízení vozíku a měření veličin (dráha, čas, rychlost a pulsy od vrtulí) nutných pro kalibraci. Vyšší úroveň řízení, palubní počítač, umožňuje automatické řízení vozíku a celého kalibračního procesu. Výsledky kalibrace jsou ukládány do tohoto počítače a zároveň jsou pravidelně posílány bezdrátovou sítí LAN do počítače na velínu, který reprezentuje třetí, dohlížecí úroveň. Vozík je možné řídit i z velína, kde v tom případě probíhá vizualizace kalibračního procesu.

Abstract

The aim of this work is to describe the control system that was developed for a new calibration trolley for Czech calibration station in Prague. The new trolley replaces the old one, which was destroyed during floods in 2002. Nowadays, the main task of the trolley is a calibration of propellers, which are designed for measuring a water-flow speed. During calibration, propellers are dragged in a water-channel below the trolley. This report is divided into three main parts, each of them concerns one level of the control system. The lower level, consisting of two cards with programmable logic arrays Xilinx, serves for manual control of the trolley and for position, time, speed and propeller pulses measurements, which all are needed for the calibration process. The upper level, made up of an onboard computer, makes it possible to control the trolley and the whole calibration process automatically. The outcomes of measurements are saved into this computer and also are regularly being sent into another computer in the control room through wireless network LAN. The control room represents the third, supervision level, and it is also possible to control the trolley from there. In that case, the calibration process is visualized in the control room.

Poděkování

Na tomto místě bych rád poděkoval Ing. Pavlu Hůlovi a Ing. Janu Kroupovi za jejich rady a podporu v průběhu práce na projektu, Ing. Milanu Bendovi za možnost se tohoto projektu ve firmě Dicont zúčastnit, a Doc. Janu Bílkovi, jakožto vedoucímu práce, za jeho rady a zapůjčení některých školních zařízení.

V neposlední řadě chci poděkovat své rodině za podporu v průběhu celého studia.

Obsah

1 Úvod ... 8
2 Popis kalibračního vozíku a příslušenství ... 10
2.1 Parametry vozíku a kalibračního kanálu ... 10
2.2 Pohony a brzdný systém ... 10
2.3 Přístrojové skříně ... 11
2.4 Akční členy a čidla ... 12
2.5 Řídící systém ... 13
3 Nižší řídící úroveň ... 15
3.1 Ovládací panel řídícího pultu ... 15
3.2 Jednotka logického řízení LC ... 16
3.2.1 Příprava vozíku ... 17
3.2.2 Přepínaní režimu Auto / Ručně ... 17
3.2.3 Start / Stop jízdy ... 18
3.2.4 Jízdní režimy ... 19
3.2.5 Stop režimy ... 20
3.2.6 Řízení převodovek ... 21
3.2.7 Pohyb po dráze ... 22
3.2.8 Řízení otočí ... 22
3.3 Jednotka měření a kalibrace MK ... 23
3.3.1 Kalibrační měření ... 23
3.3.2 Schéma desky MK ... 24
3.3.3 Registry desky MK ... 27
3.3.4 Status registr ... 28
4 Vyšší řídící úroveň ... 30
4.1 Palubní počítač ... 30
4.1.1 Specifikace počítače ... 30
4.1.2 I/O karta ... 31
4.1.2.1 Připojení desky LC na I/O kartu ... 31
4.1.2.2 Připojení desky MK na I/O kartu ... 32
4.2 Spojení počítače s frekvenčními měniči ... 32
4.3 Popis oken řídícího programu ... 34
4.4 Realizace řídícího programu ... 38
4.4.1 Simulátor vozíku ... 39
4.4.2 Obecné třídy ... 40
4.4.3 Hlavní vlákno programu ... 41
4.4.3.1 Přepínání mezi stavy vozíku ... 42
4.4.3.2 Přepínání mezi stavy průběhu měření ... 44
4.4.3.3 Brždění na konci dráhy ... 45
4.4.3.4 Databáze a protokoly o měření ... 46
4.4.4 Komunikace s frekvenčními měniči ... 47
4.4.4.1 Komunikační protokol ... 47
4.4.4.2 Implementace protokolu v programu ... 48
4.4.5 Komunikace s deskou MK ... 50
4.4.6 Sekundární vlákna ... 50
4.4.6.1 Synchronizace vláken ... 50
4.4.6.2 Kontrolní signál watchdog ... 51
4.4.6.3 Monitorování registrů MK a vizualizace procesu ... 51
4.4.6.4 Řízení kalibrace ... 53

4.4.7 Nápověda ... 58
5 Dohlížecí úroveň ... 59
5.1 Počítač na velínu ... 59
5.2 Komunikační protokoly TCP/UDP/IP ... 59
5.3 Programy ve velínovém počítači ... 60
5.4 Komunikace velín - vozík ... 61
5.5 Kalibrace řízená z velína ... 64
6 Závěr ... 66
7 Reference ... 67
Příloha A Zapojení součástek na desce LC ... 68
Příloha B Schéma desky LC ... 73
Příloha C Zapojení součástek na desce MK ... 74
Příloha D Schéma desky MK ... 75
Příloha E Seznam klávesových zkratek v programu ... 76
Příloha F Obsah přiloženého CD ... 77
Příloha G Fotografie ... 78

1 Úvod

Tato práce se zabývá popisem zařízení vyvinutého kolektivem ve firmě Dicont pro Českou kalibrační stanici vodoměrných vrtulí ve Výzkumném ústavu vodohospodářském (VÚV) v Praze 6 - Podbabě. Jedná se o řídící systém multifunkčního vozíku pohybujícího se po kolejnicích nad kanálem napuštěným vodou. Vozík byl projektován tak, aby umožňoval plnění následujících úkolů:

a) plně automatické kalibrování vodoměrných vrtulí vlečených vodou na tyčích, včetně předání výsledků ke zpracování do protokolu
b) kalibrování vodoměrných vrtulí vlečených vodou na zavěšených závažích
c) vlečení jiných předmětů vodou za účelem laboratorních měření
d) provádění hydrodynamických měření vyžadujících blízký přístup k hladině vody v kanále,

přičemž v současné době je aktuální pouze první úkol, ostatní se týkají plánovaných budoucích aplikací vozíku.

Samotný vozík, včetně jeho brzdného systému, byl vyroben spolupracující firmou Inco. Naše firma Dicont zavedla kompletní elektrickou instalaci a vyvinula řídící a měřící systém vozíku.

Vozík je schopen pohybovat se konstantní rychlostí v rozmezí 0,02 – 10 m/s, provádět kompletní kalibrační měření (měření dráhy, času a čítání impulsů od vodoměrných vrtulí při různých rychlostech), výsledky měření předávat počítači na vozíku a průběžně je přenášet do počítače na velínu bezdrátovým přenosem. Samotné měření a ruční řízení vozíku je zajištěno pomocí desek s programovatelnými logickými obvody. Počítač na vozíku zajišťuje vyšší úroveň řízení a obsahuje software pro zadání parametrů měření, řízení celého procesu kalibrace, zpracování výsledků a zobrazování stavu vozíku včetně vizualizace kalibračního procesu. Vozík je možné řídit i z velína, kde se také zobrazuje aktuální stav včetně vizualizace.

Podobný vozík již v ústavu fungoval, byl ovšem zničen při povodni v létě 2002. Cílem nyní bylo vyvinout vozík modernější. Původní vozík byl schopen dosáhnout max. rychlosti pouze 7 m/s, řídící systém byl na reléové bázi a nebyl tedy tak lehce modifikovatelný jako v případě programovatelných polí, neumožňoval plně automatickou kalibraci vrtulí bez zásahu obsluhy, nezobrazoval vizualizaci ani neumožňoval pravidelný přenos dat do velína. Povodeň také ukázala, že musí existovat evakuační zařízení, které v případě další povodňové hrozby vozík včas přesune mimo dosah vody.

V následujících kapitolách je nejprve uveden popis vozíku samotného (kapitola 2) a poté kompletní rozbor řídícího systému v kapitolách 3 až 5. Kapitola 3 se věnuje nižší úrovni řízení (řídící a měřící deska), kapitola 4 vyšší úrovni řízení (palubní počítač) a kapitola 5 dohlížecí úrovni (velín).

Mým hlavním úkolem byla realizace vyšší a dohlížecí úrovně řízení (kapitoly 4 a 5). Nižší řídící úroveň je společným dílem, hlavní zásluhu na jejím vývoji mají Ing. Pavel Hůla a Ing. Jan Kroupa.

2 Popis kalibračního vozíku a příslušenství

Před popisem řídícího systému je nutné se seznámit s vozíkem a jeho jednotlivými částmi. Stručný přehled uvádí tato kapitola.

2.1 Parametry vozíku a kalibračního kanálu

Parametry vozíku jsou následující:

- Délka 3430 mm
- Šířka 3500 mm
- Výška 2700 mm
- Průměr kol 350 mm
- Hmotnost 3600 kg

Popis dílčích částí vozíku je obsahem následujících kapitol (kap 2.2 – 2.5).

Kalibrační kanál se nachází v přízemí budovy VÚV. Je betonový, obdélníkového profilu, 130 m dlouhý, 3 m široký a 3 m hluboký. Na březích kanálu jsou kolejnice, vedoucí po celé jeho délce, po kterých se pohybuje vozík. Kanál je orientován od jihu k severu, na obou jeho koncích se nacházejí stavidla, která ho oddělují od říčního toku Vltavy. Před kalibračním měřením se kanál napustí po okraj a obě stavidla uzavřou. Získá se tak stojatá voda pro co nejpřesnější měření.

Podél celé délky kanálu vedou pod stropem troleje (3x380 V, 50 Hz), ze kterých vozík získává energii pomocí sběračů. V blízkosti trolejí vede pravítko s dírami pro určování polohy vozíku na dráze.

Vozík, jeho části a kalibrační kanál jsou zobrazeny na fotografiích v příloze G.

2.2 Pohony a brzdný systém

Základní funkcí vozíku pro zajištění výše jmenovaných poslání je pohybovat se v podélné ose kalibračního kanálu přesně definovanou rychlostí.

Vzhledem k relativně vysoké horní mezi požadovaného nastavení rychlosti vozíku na jedné straně a k omezené délce kalibračního kanálu na straně druhé, musí být vozík vybaven výkonnými pohony a zejména pak výkonným a spolehlivým brzdným systémem, zabezpečeným automatickými, na člověku nezávislými, ochrannými funkcemi.

Vozík je tedy vybaven dvěma shodnými pohonnými agregáty o výkonu 15 kW (1450 ot/min), umístěnými pod podlahou plošiny. Každý agregát pohání jednu nápravu vozíku o dvou kolech a sestává z elektromotoru, převodovky, rozvodovky a spojovacích spojek a kardanů. Agregáty jsou schopny pracovat samostatně nebo ve dvojici, se vzájemnou regulační vazbou. Každá převodovka má dva řaditelné převodové stupně. Při automatickém řízení vozíku se volba převodového stupně i volba jednoho nebo obou pohonných agregátů provádí automaticky podle nastavené kalibrační rychlosti. Rychlost motorů je řízena statickými měniči frekvence.

Brzdný systém Sistonik je elektronicko – pneumatický, vybavený sofistikovaným řízením, které zaručuje jeho vysokou spolehlivost a bezpečnost brždění za každé situace. Základem jsou dva nezávislé programovatelné automaty Tecomat a Siemens, které pomocí svých čidel permanentně vyhodnocují jízdní situaci vozíku a řídí průběh provozního i havarijního brždění. Automaty se navzájem zálohují. Vůči centrálnímu řídícímu systému vozíku je systém Sistonik spolupracující podsystém, avšak s nezávislými ochrannými funkcemi. Vlastní brzdný efekt zajišťují čtyři pneumatické diskové brzdy, se základním přítlakem čelistí pružinami nezávislými na jiném zdroji energie. Jinak je přítlak čelistí ovládán stlačeným vzduchem z kompresoru.

2.3 Přístrojové skříně

Na plošině vozíku, kryté snímatelnou podlahou, je vybudováno palubní pracoviště operátora vozíku, kde jsou soustředěny prvky elektrických, elektronických a pneumatických obvodů. Veškerá tato technika je podle svého převažujícího zaměření rozdělena do několika přístrojových skříní:

a) rozvaděčová skříň - pro hlavní elektrickou přípojku z trolejí a pro rozvedení, jištění a ovládání jednotlivých obvodů nízkého napětí

b) měničová skříň - pro umístění statických měničů frekvence hlavních pohonných jednotek s příslušenstvím

c) elektrická skříň Sistonik - soustřeďuje elektroniku a řízení brzdného systému, řazení převodovek a ovládání otočí

d) pneumatická skříň Sistonik - soustřeďuje prvky pneumatických ovládacích obvodů

e) kompresorová skříň - obsahuje zdroj stlačeného vzduchu s příslušenstvím

f) řídící pult (ŘP) - na jeho čelních plochách jsou zabudovány prvky pro styk s obsluhou vozíku, tj. ruční tlačítka a klíče, klávesnice s touchpadem, monitor, uvnitř pultu je řídící PC a dále s ním spolupracující jednotka logického řízení (LC) a jednotka měření a kalibrace (MK)

2.4 Akční členy a čidla

Tyto členy jsou umístěny mimo přístrojové skříně, většinou pod plošinou vozíku. Jedná se o:

a) dva pneumatické servopohony řazení s čidly koncových poloh, těmito pohony je zajištěno přeřazování mezi dvěma stupni u planetových převodovek pohonných agregátů

b) tři pneumatické servopohony otočí s čidly koncových poloh, jsou umístěny na vrtulové nástavbě a zajišťují otáčení tyčí s vrtulemi podle směru jízdy vozíku

c) dva magnetické snímače koncových poloh kanálu (koncové spínače KS), každý snímač je tvořen magnetem při pravé kolejnici (ve směru toku řeky) a čidlem pod vozíkem, impuls z těchto čidel má za následek okamžité zastavení vozíku

d) dva magnetické snímače hranice kontroly brždění (spínače KBS), každý snímač je tvořen magnetem při levé kolejnici a čidlem pod vozíkem, slouží brzdnému systému Sistonik, při jejich přejetí začíná kontrola rychlosti podle povolené rychlostní obálky

e) jeden mechanický přepínací snímač dorazů kanálu (spínač DS), dorazy jsou při koncích levé kolejnice

f) jeden optosnímač hrubé polohy vozíku na dráze, je umístěn na konzoli sběrače trolejí a snímá impulsy z děr pravítka podél trolejové dráhy

g) jeden inkrementální snímač otáček pro odměřování dráhy, je umístěn při pravém předním kole vozíku

h) jeden tachometrický snímač otáček pro odměřování dráhy, při pravém zadním kole vozíku

i) dva inkrementální snímače otáček pro regulaci rychlosti a jemné určení polohy vozíku a ujeté dráhy, jsou zabudovány přímo do motorů pohonných agregátů

j) dvě bimetalická čidla ohřátí motorů, zabudovány do motorů pohonných agregátů

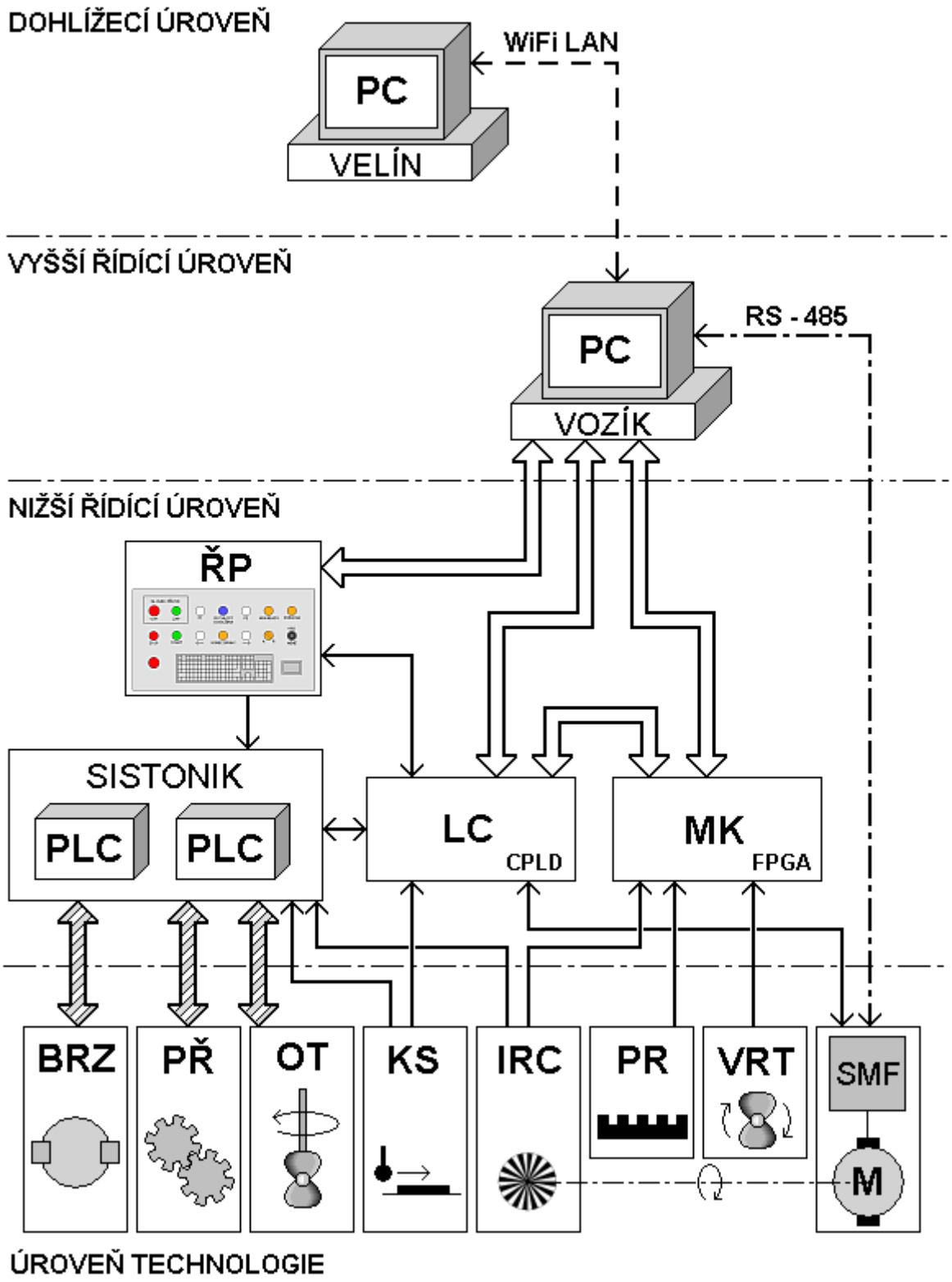
k) jeden mikrospínač sklopení vrtulové nástavby

l) pomocná čidla pneumatického systému Sistonik

Spojovací kabely a pneumatické trubky mezi skříněmi, akčními členy a čidly jsou vedeny převážně v trubkových trasách pod podlahou vozíku. Důsledně jsou odděleny trasy silnoproudé od tras slaboproudých.

2.5 Řídící systém

Schéma celého řídícího systému je zobrazeno na obr. 2.1.



(obr. 2.1 – schéma řídícího systému)

Celé schéma je rozděleno na čtyři úrovně. *Úroveň technologie* zahrnuje akční členy a čidla. *Nižší řídící úroveň* obsahuje všechny prostředky pro ruční řízení vozíku. *Vyšší řídící úroveň* zajišťuje automatickou kalibraci vrtulí. A nakonec nejvyšší, *dohlížecí úroveň*, umožňuje ovládání vozíku z velína.

Ve schématu je použito několik druhů šipek. Jednoduchá šipka značí jeden nebo více vodičů, kterými prochází signály napěťové úrovně 24 V. Tyto signály jsou buď charakteru binárního (např. sepnutí spínače) nebo impulsního (např. signál z inkrementálního čidla nebo z vrtule). Jednoduchá čerchovaná šipka označuje sériovou linku RS-485 a jednoduchá přerušovaná bezdrátovou síť LAN. Dvojitá šipka značí ploché vícežilové kabely pro vzájemné číslicové propojení počítače a desek LC a MK. Těmito kabely prochází signály úrovně TTL. Tato šipka značí též kabely klávesnice a touchpadu. Posledním typem jsou pruhované dvojité šipky, které zobrazují elektronicko-pneumatické spojení mezi Sistonikem a akčními členy.

Úroveň technologie zahrnuje všechny akční členy a čidla. Akčními členy jsou brzdy (BRZ), servopohony řazení převodovek (PŘ), servopohony otočí (OT) a motory (M) se statickými měniči frekvence (SMF). Mezi čidla patří koncové spínače (KS – spínače koncových poloh kanálu, zařazení převodovek i natočení otočí), inkrementální snímače otáček (IRC), pravítko (PR) s optosnímačem, a vrtule (VRT), které jsou kalibrovány.

Nižší řídící úroveň umožňuje ruční řízení vozíku. Je také samozřejmě nezbytnou základnou pro *vyšší řídící úroveň*. Ruční řízení spočívá v ovládání všech hlavních pohybových funkcí vozíku z ručních ovládacích prvků na řídícím pultu (ŘP), přičemž automatické bezpečnostní funkce (havarijní brždění a hlídání konce dráhy) jsou aktivní. Ruční řízení se hodí pro manipulační a servisní účely, v tomto režimu musí být obsluha přítomna na vozíku. Ve schématu vidíme jednotlivé části této řídící vrstvy – jednotku logického řízení LC, jednotku měření a kalibrace MK, brzdný systém Sistonik (obsahující programovatelné logické automaty PLC) a řídící pult ŘP. Této úrovni je věnována *kap. 3*.

Vyšší řídící úroveň, nebo-li palubní počítač, přidává vozíku možnost automaticky provádět kalibrační měření. V tomto režimu zadá uživatel parametry měření do programu v palubním počítači a může opustit vozík. Ten následně provede měření bez jeho přítomnosti. V ručním režimu má počítač také uplatnění, zobrazuje informace o stavu vozíku a umožňuje nastavení některých méně používaných parametrů, které není možné nastavit z řídícího pultu. Není však pro ruční režim nutný a může být případně odpojen. Podrobný popis vyšší řídící úrovně přináší *kap. 4*.

Dohlížecí úroveň slouží k ovládání a monitorování vozíku z velína, umístěného v jiné části budovy. Spojení vozíku a velínu je zajištěno bezdrátovou sítí LAN. Z velínového počítače může uživatel zadávat parametry měření a kontrolovat stav vozíku, podobně jako z palubního počítače. Navíc velínový počítač umožňuje zpracování výsledků měření do protokolů pro zákazníky. Ve velínu má obsluha dále možnost kontrolovat stav vozíku pomocí čtyř televizních kamer, které jsou rovnoměrně umístěny podél celé délky kalibračního kanálu.

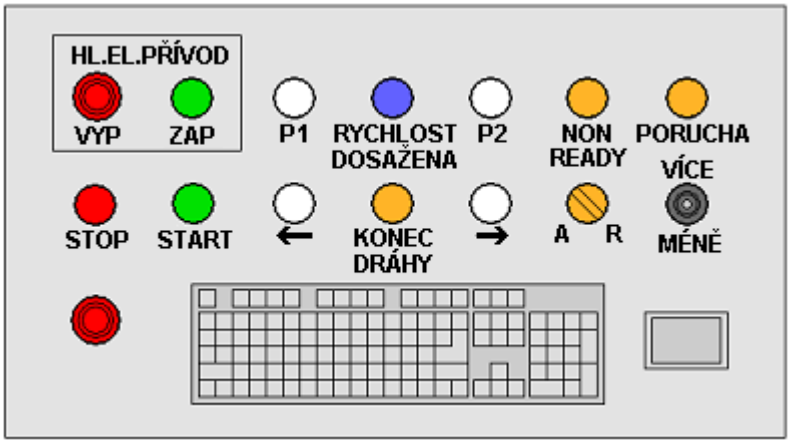
3 Nižší řídící úroveň

Tato kapitola popisuje nižší úroveň řízení, která umožňuje ruční řízení vozíku, tedy ovládání základních pohybových a kontrolních funkcí. Nejprve jsou stručně vysvětleny části řídícího pultu (kap. 3.1) a poté podrobně rozebrány řídící desky LC a MK (kap. 3.2 - 3.3). Brzdný systém Sistonik byl vyvíjen a instalován firmou Inco, z toho důvodu tato práce jeho popis neobsahuje.

3.1 Ovládací panel řídícího pultu

V následujících podkapitolách se hovoří o signálech z řídícího pultu, je proto vhodné se s pultem stručně seznámit.

Ovládací panel řídícího pultu obsahuje prostředky pro zapnutí a vypnutí vozíku, tlačítka a signálky pro ovládání vozíku v ručním režimu, a klávesnici s touchpadem pro ovládání palubního počítače. Panel je umístěn pod monitorem, v mírně šikmé poloze, před sedící obsluhou. Je znázorněn na obr. 3.1.



(obr. 3.1 – řídící pult vozíku)

Význam jednotlivých tlačítek a signálek je následující (zleva doprava a shora dolů):

a) aretační tlačítko VYP – odpojení celého systému od napětí
b) tlačítko se signálkou ZAP – zapnutí napájení do systému
c) tlačítko se signálkou P1 – zařazení prvního převodového stupně
d) signálka RYCHLOST DOSAŽENA - svítí, pokud vozík jede žádanou rychlostí
e) tlačítko se signálkou P2 – zařazení druhého převodového stupně
f) signálka NON READY – svítí, pokud některá z částí vozíku není připravena k funkci
g) signálka PORUCHA – svítí, pokud některá z částí vozíku je v poruchovém stavu
h) tlačítko se signálkou STOP – zastavení vozíku (pomocí motorů, po rychlostní rampě)
i) tlačítko se signálkou START – rozjetí vozíku, svítí po dobu jízdy vozíku
j) tlačítko se signálkou *„šipka doleva“* – navolení směru jízdy na jih

k) signálka KONEC DRÁHY – svítí, pokud vozík přejel některý z koncových spínačů
l) tlačítko se signálkou *„šipka doprava“* – navolení směru jízdy na sever
m) otočný přepínač se signálkou A/R – zvolení režimu jízdy (automaticky nebo ručně)
n) páčkový přepínač VÍCE / MÉNĚ – změna rychlosti, přepínač je zapojen jako motorický potenciometr, tj. velikost změny žádané rychlosti je úměrná době výchylky páčky (přepínač je připojen na speciální vstup frekvenčních měničů, ve skutečnosti se tedy mění žádaná frekvence)
o) druhé aretační tlačítko – způsobí okamžité zabrždění vozíku, toto tlačítko je vyvedené přímo do brzdného systému Sistonik

Většina tlačítek a signálek vyjadřuje vstupy a výstupy do/z desky LC. Názvy na pultu a odpovídající jména signálů v desce LC (viz příští kapitola) jsou buď stejné nebo velmi podobné.[1]

3.2 Jednotka logického řízení LC

Jednotka logického řízení slouží k logickému řízení vozíku. Znamená to, že na vstupu přijímá signály z řídícího pultu, frekvenčních měničů, brzdného systému, převodovek, otočí, počítače a druhé řídící jednotky (MK - jednotky měření a kalibrace), tyto signály upravuje pomocí logických funkcí (součtu, součinu a negace) a klopných obvodů a posílá je na výstup (opět do technologie, počítače a druhé řídící jednotky).

Jednotka je realizována na desce plošných spojů, jejímž jádrem je programovatelné logické pole Xilinx XC95288XL typu CPLD (Configurable Programmable Logical Device). Je to pole typu FLASH, tzn. to co je do něj nahráno v něm zůstává i po odpojení od napájení. 24-voltové vstupy z technologie jsou na optočlenech TLP627 galvanicky odděleny od zbytku desky a převedeny na logiku 3,3 V, která vyhovuje vstupům čipu Xilinx. Výstupy do technologie jsou odděleny optočleny PC815. Vstupy z počítače jsou úrovně TTL (5V) a jsou převedeny na 3,3 V pomocí obvodů 74LS245. Celá deska včetně pole Xilinx je napájena ze spínaného zdroje s obvodem LM2574N-3, který dává výstupní napětí 3,3 V (LVTTL).

Zapojení součástek na desce LC, včetně struktury spínaného zdroje, je zobrazeno v příloze A. Jsou přiloženy i struktury bloků TECHIN (vstupy z technologie), OPTOIN (optické oddělení vstupů), TECHOUT (výstupy do technologie) a FETOUT (optické oddělení výstupů).

V následujícím popisu je celkové logické schéma rozděleno na osm částí podle funkcí, které tyto části zajišťují. Znamená to, že některé výstupní signály znamenají skutečné výstupy, jiné jsou pouze vnitřními signály a vstupují do jiných částí schématu. Celé schéma je realizováno v poli Xilinx. Propojení jednotlivých bloků je možné nalézt v příloze B, kde je plné schéma desky LC.

Jednotlivé bloky jsou popsány v následujících podkapitolách *3.2.1 – 3.2.8.* Vyskytuje se v nich klopný obvod, který s náběžnou hranou signálu C přesouvá vstup D na výstup Q. V případě aktivního prioritního signálu CLR je na výstupu vždy nula.

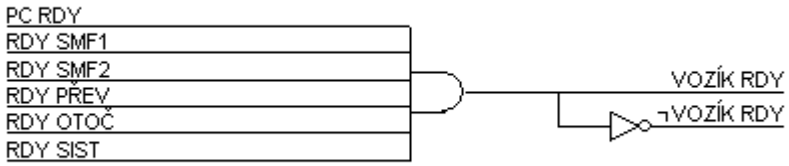
[1] Pro jistotu uvádím ekvivalenty: RYCHLOST DOSAŽENA = VŽ, NON READY = neg. VOZÍK RDY, STOP = STOP JÍZDY, START = JÍZDA, šipka doleva = J, šipka doprava = S, KONEC DRÁHY = STOP DRÁHA

3.2.1 Příprava vozíku

Schéma přípravy vozíku je na obr. 3.2.

Po zapnutí elektrického napájení proběhne automatická kontrola připravenosti (RDY) dílčích částí vozíku (počítače, měničů, převodovek, otočí a brzdného systému). Jsou-li všechny části připraveny, LC vydá signál VOZÍK RDY.

Pozn.: Signál PC RDY je výstupem obvodu, který přijímá watchdog od počítače (impulsní signál PCWD).



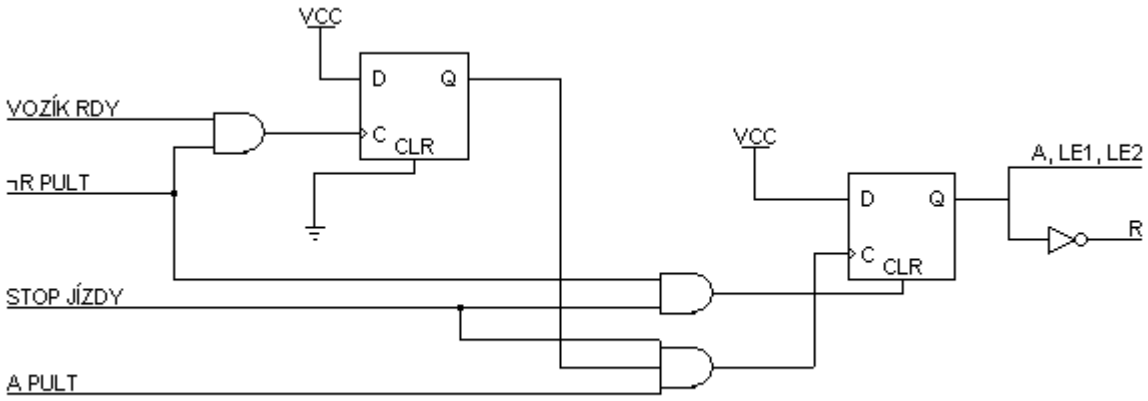
(obr. 3.2 – schéma přípravy vozíku)

3.2.2 Přepínaní režimu Auto / Ručně

Schéma přepínání režimů je znázorněno na obr. 3.3.

Po aktivaci signálu VOZÍK RDY je možné z řídícího pultu vozíku volit ručním přepínačem režim ovládání vozíku (A/R). Z důvodů bezpečnosti musí po náběhu signálu VOZÍK RDY jako první být zadána (nebo již existovat) volba R. Jinak nelze A vůbec zadat.

Při ručním ovládání (R) lze ovládat vybrané základní funkce vozíku z ovládacích prvků řídícího pultu. Cílem ručního ovládání je hlavně provádět manipulační, seřizovací a servisní činnosti na vozíku.



(obr. 3.3 – schéma přepínání režimů)

Při volbě automaticky (A) je zdrojem ovládání palubní počítač, kterému se zadávají výchozí data buď z jeho vlastní klávesnice nebo po sériové lince z velína kalibrační stanice. Zároveň se uvolňuje linka sériového přenosu mezi palubním počítačem a statickými měniči frekvence SMF signály LE1, LE2 (Link Enable).

Výstup signálů R, A, LE je z bezpečnostních důvodů podmíněn stavem STOP JÍZDY.

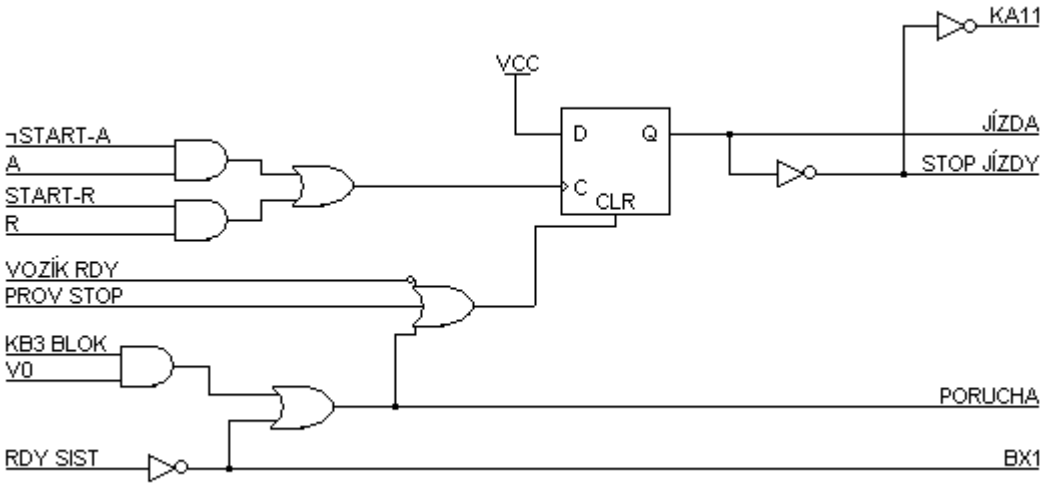
3.2.3 Start / Stop jízdy

Schéma start / stop jízdy je na obr. 3.4.

Tato část schématu vytváří základní podmínky pro rozjezd vozíku (JÍZDA) nebo jeho zastavení (STOP JÍZDY). Zároveň se vysílá do brzdného systému Sistonic binární signál KA11 (KA11 = 1 je povel k odbrzdění, KA11 = 0 povel k zabrzdění). Kromě toho se vydává informace o zastavení vozíku nebo blokování jeho startu v důsledku vzniklé poruchy (PORUCHA) a v některých případech havárie povel měničům k elektromagneticky nebrzděnému zastavení motorů (BX1).

Ke startu jízdy (JÍZDA) je potřeba přítomnost vstupů START–A, START-R[2] spolu s uvolněním příslušného režimu A, R a podmínkou je stav VOZÍK RDY a neexistence PORUCHY.

Ke STOPU JÍZDY je nutný povel k provoznímu zastavení (PROV STOP) nebo k havarijnímu zastavení hlášení ze Sistoniku. Rovněž ztráta aktivace vstupu VOZÍK RDY vede ke STOPU JÍZDY. K zablokování dalšího STARTU JÍZDY po předchozím provozním zastavení slouží vstup KB3 BLOK ze Sistoniku.



(obr. 3.4 – schéma start/stop jízdy)

[2] Písmeno R za názvem signálu znamená, že se jedná o signál z řídícího pultu, písmeno A značí signál z palubního počítače

3.2.4 Jízdní režimy

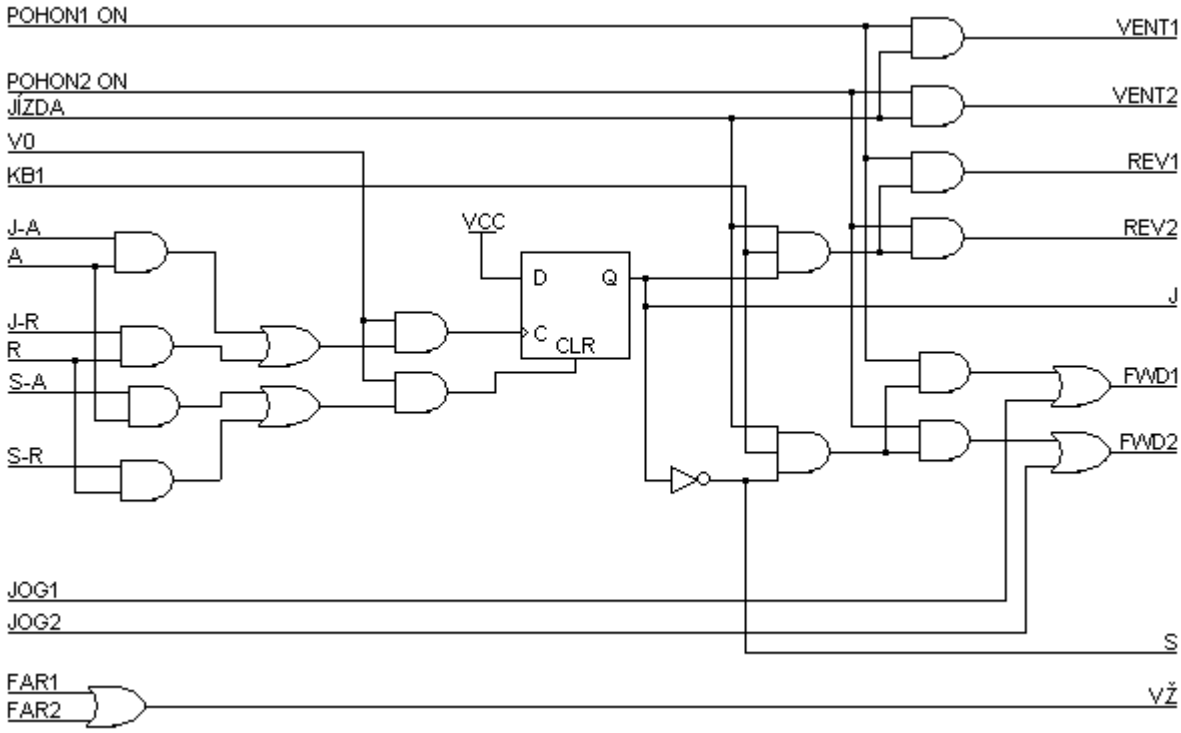
Schéma jízdních režimů je znázorněno na obr. 3.5.

Odezva ze Sistoniku na povel k odbrzdění (signál KB1) spolu s aktivním signálem JÍZDA je v LC vyhodnocena jako povel k jízdě. Tento povel spolu s určením žádaného smyslu otáčení motorů formuje výstupy pro měniče FWD1,2 - vpřed nebo REV1,2 - vzad. Tyto výstupy trvají po celou dobu jízdy až do doby aktivace signálu STOP JÍZDY.

Pomocí signálů z počítače POHON1,2 ON lze samostatně zařadit / vyřadit z funkce jeden nebo druhý pohon (případně oba) tím, že se potvrdí povelová účinnost příslušných výstupů FWD, REV a také tím, že se těmito signály řídí řazení převodovek na záběr nebo na neutrál (viz kap. 3.2.6 - řízení převodovek). Podle těchto signálů se také při jízdě zapínají ventilátory VENT1,2.

Smysl otáčení se zadává vstupy S-A, J-A, S-R, J-R (S - sever, J - jih) spolu s uvolněním příslušného režimu A, R. Zadání je účinné pouze ve stavu nulové rychlosti (signál v0) a je pamatováno až do příští změny v klopném obvodu.

Signály JOG1,2 jsou generovány ve schématu řízení převodovek (kap. 3.2.6) a znamenají, že na daném pohonu ještě nedošlo k přeřazení. Aby se přeřazení usnadnilo, signál JOG rozjede motor na minimální otáčky a zuby v převodovce tak při pohybu hřídele do sebe lépe zapadnou[3].



(obr. 3.5 – schéma jízdních režimů)

[3] Při jízdních zkouškách se občas stávalo, že k přeřazení nedošlo vůbec, protože zuby v převodovce byly vůči sobě v nevhodné poloze a servomotor pro přeřazení to neutáhl. Bylo nutné tedy ručně hýbnout hřídelí pohonu. Tento úkon byl následně zautomatizován signálem JOG.

Výskyt alespoň jednoho ze vstupních signálů z měničů FAR1,2 je vyhodnocen jako dosažení žádané rychlosti – VŽ.

3.2.5 Stop režimy

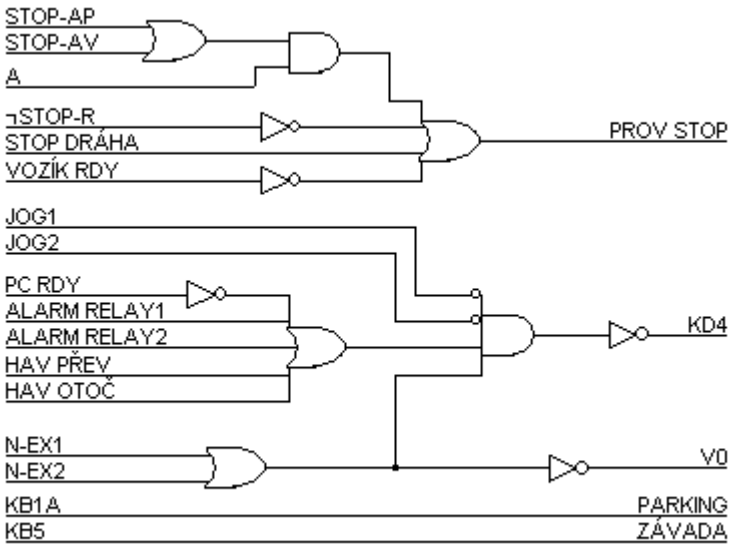
Schéma stop režimů znázorňuje obr. 3.6.

Při provozu vozíku rozlišujeme dva základní režimy zastavení: režim provozního brždění, vyjádřený výstupem PROV STOP, a režim brždění havarijního.

Provozní brždění může být vždy vyvoláno tlačítkem z řídícího pultu (STOP-R), nebo některým ze signálů v automatickém režimu: STOP-AP z počítače, STOP-AV z čítačů impulsů vrtulí (na jednotce měření a kalibrace MK) nebo signálem konce dráhy (STOP DRÁHA).

Havarijní stavy vyhodnocované mimo vlastní kontrolní obvody brzdného systému Sistonik jsou sdruženy do sumárního signálu pro spínání relé KD4 tohoto systému. Jedná se o signál poruchy z PC (neakt. PC RDY), z měničů a jimi sledovaných motorů (ALARM RELAY1,2), od převodovek (HAV PŘEV) a od otočí (HAV OTOČ). Relé KD4 se nesepne, pokud je aktivní alespoň jeden ze signálů JOG.

Kromě uvedených STOP režimů zprostředkuje logické řízení LC informaci o zjištění v Sistoniku nedůležitých závad, které nevedou k bezprostřednímu zastavení vozíku (KB5 - ZÁVADA), informaci pro Sistonik o otáčkách blízkých nulovým obou pohonných motorů v0 (při výskytu alespoň jednoho ze signálů N-EX1,2 z měničů) a informaci ze Sistoniku o zabrzděném stavu (KB1A - PARKING).



(obr. 3.6 – schéma stop režimů)

3.2.6 Řízení převodovek

Schéma řízení převodovek je na obr. 3.7.

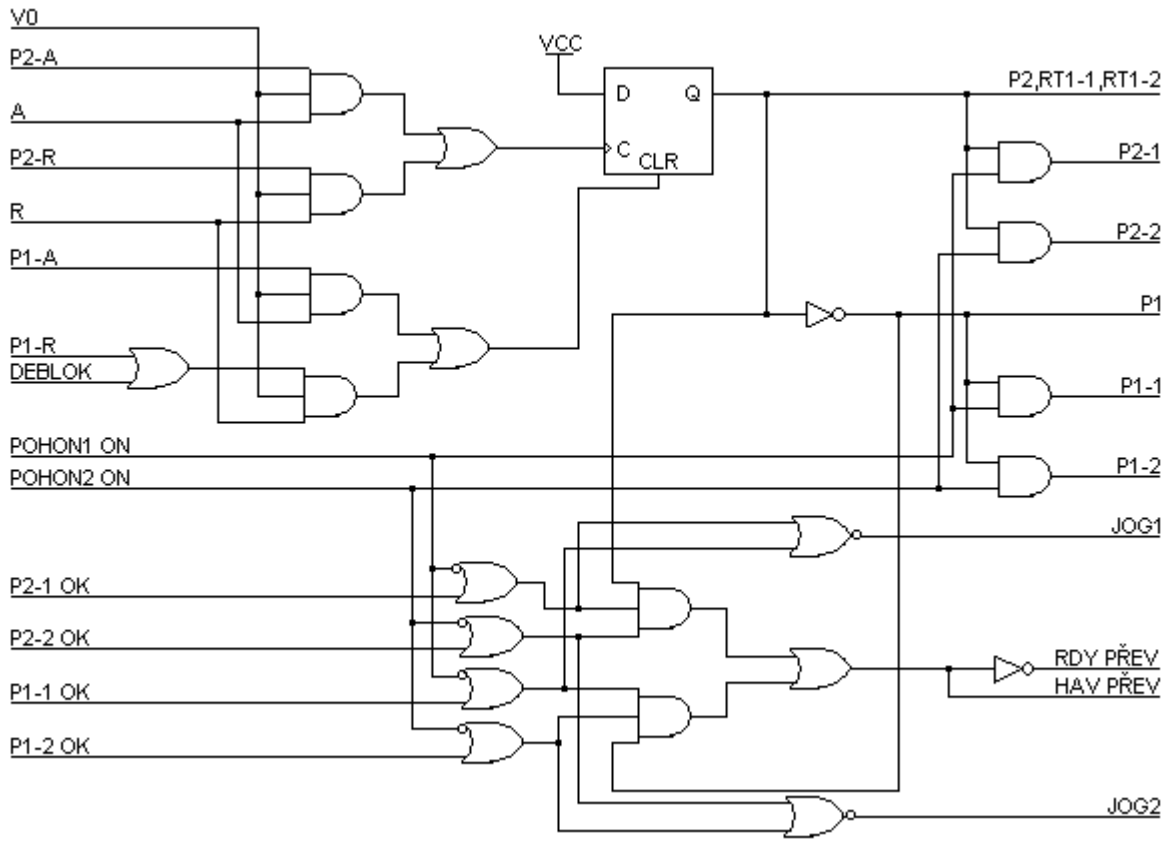
Je-li vozík zastaven (v0), je možné změnit zařazení prvního nebo druhého stupně obou převodovek (u obou převodovek shodně). Děje se to výstupy P1-1,2 nebo P2-1,2. Při řazení na rychlejší stupeň P2 se zároveň výstupem RT1 přepínají parametry akceleračních a deceleračních ramp v měničích.

Řazení na P1 se provede vstupem P1-A v automatickém režimu nebo vstupem P1-R (či DEBLOK) v ručním. Obdobně probíhá řazení na P2. Při současném požadavku na P1 i P2 má P1 prioritu.

Příkazy POHON 1,2 ON z počítače je možné každou převodovku nastavit do funkce řazení nebo naopak odstavit do neutrálu a tím vyřadit náhon příslušné nápravy.

Jakoukoliv změnu řazení doprovází příkazy JOG 1,2 do obou měničů. Na tyto příkazy se začnou motory pohonných agregátů velmi pomalu otáčet, což trvá do okamžiku, kdy přijde zpětná hláška o zařazení. Funkce zlepšuje proces řazení, viz. pozn. v kap. 3.2.4.

Kontrola řazení se provádí porovnáním zpětných signálů (P1-1,2 OK, P2-1,2 OK) se zadaným příkazem P1 nebo P2. Pokud nedojde k souhlasu, není možné vozík rozjet, protože vlivem neaktivního výstupu RDY PŘEV je neaktivní i výstup VOZÍK RDY.



(obr. 3.7 – schéma řízení převodovek)

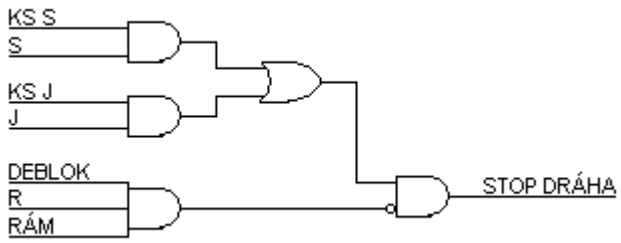
3.2.7 Pohyb po dráze

Schéma pohybu po dráze znázorňuje obr. 3.8.

Na obou koncích dráhy tárovacího žlabu má vozík jízdní omezení, spočívající v zábraně jízdy směrem ven z dráhy. To je vyjádřeno vstupy z koncových spínačů na severním (KS S) resp. jižním (KS J) konci dráhy při jízdě na sever (S) resp. na jih (J). Omezení se projeví zastavením vozíku za koncovými spínači výstupem STOP DRÁHA

Toto omezení lze v ručním režimu zrušit signálem DEBLOK z tlačítka na ŘP (tlačítko je na pultu sloučené se signálkou KONEC DRÁHY), pokud je zároveň zdvižen rám s vodoměrnými vrtulemi (aktivní signál RÁM). Zábrana jízdy se automaticky ruší při obrácení žádaného směru jízdy dovnitř dráhy.

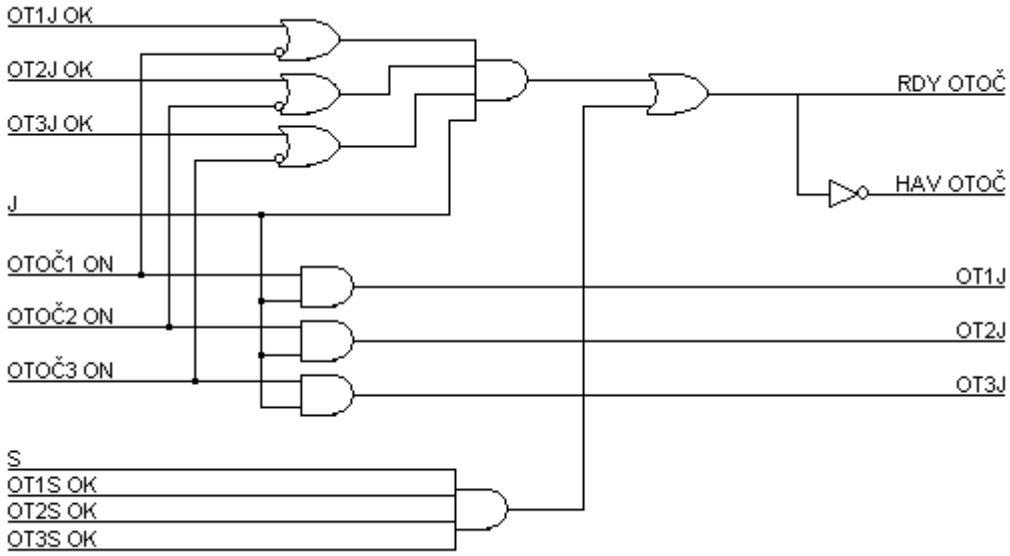
Koncový spínač KS J je zároveň cejchovní značkou pro digitální odměřování dráhy a jeho přejetím od jihu na sever se v desce MK zkalibruje (vynuluje) aktuální poloha na dráze.



(obr. 3.8 – schéma pohybu po dráze)

3.2.8 Řízení otočí

Schéma řízení otočí je znázorněno na obr. 3.9.



(obr. 3.9 – schéma řízení otočí)

Základní poloha otočí je ve směru na sever. Natáčení otočí do protisměru je řízeno vstupem J a příkazy OT1,2,3J. Provedení natočení je sledováno (signály OT1,2,3J OK při otočení na jih a OT1,2,3S OK při otočení na sever) a je podmínkou aktivace signálu RDY OTOČE. Ovládání otočí lze jednotlivě vypnout z PC pomocí deaktivace signálů OTOČ1,2,3 ON.

3.3 Jednotka měření a kalibrace MK

V této kapitole je popsána jednotka MK, která je srdcem celého kalibračního procesu a která zajišťuje měření rychlosti, polohy a času a čítání pulsů od vrtulí.

Jednotka je, podobně jako deska LC, realizována na desce plošných spojů. Jejím jádrem je programovatelné logické pole Xilinx XC2S100 Spartan-II typu FPGA (Field-Programmable Gate Array). Je to pole typu RAM, tzn. po odpojení a následném připojení napájení je nutné pole vždy znovu naprogramovat. Je to možné buď s počítače příslušným kabelem nebo pomocí konfigurační paměti. Pole FPGA mají hustší strukturu a vejde se do nich více logiky než do polí CPLD. Deska MK je výrazně složitější než deska LC, proto je pro ní použito pole FPGA. Při zapnutí se do pole nahrává program z konfigurační paměti XC18V01S020. Optické oddělení vstupů a převodníky napětí jsou řešeny stejně jako na desce LC (kap. 3.2). Deska je napájena ze spínaného zdroje s obvody LM2574N-3 a LM317, který dává napětí 2,5 V pro jádro FPGA a 3,3 V pro I/O bloky FPGA. Zapojení součástek MK je zobrazeno v příloze C.

V následujících kapitolách je nejprve rozebrán průběh kalibračního procesu (kap. 3.3.1), poté je popsáno schéma desky MK (kap. 3.3.2) a na závěr je uveden popis jejích registrů (kap. 3.3.3 - 4).

3.3.1 Kalibrační měření

Vodoměrné vrtule jsou určené pro měření rychlosti průtoku vody v řekách. Před používáním takové vrtule v praxi je však nutné, aby byla nejdříve zkalibrována a schválena. To může v současné době provést jediná instituce v České republice a tou je právě ta instituce, pro kterou je zařízení popisované v této práci vyvinuto.

Pro uskutečnění procesu kalibrace se nabízejí dvě řešení. Jednak je možné použít statickou vrtuli umístěnou v tekoucí vodě plynoucí známou rychlostí, a nebo pohybovat vrtulí danou rychlostí ve stojaté vodě. Vytvářet proud vody o přesné rychlosti toku pro první případ je jistě složitější úloha. Kalibrace se proto v našem případě uskutečňuje aplikací druhého řešení, a to připevněním vrtule na vozík, který ji vleče konstantní rychlostí vodou v kalibračním kanále.

Výsledkem kalibrace vrtule je lineární vztah závislosti rychlosti jízdy vozíku (což je ekvivalentní rychlosti proudění vody) na počtu otáček vrtule za sekundu. Vztah má následující tvar:

$$v = a + bn\,, \tag{3.1}$$

kde v je rychlost proudění vody v m/s, n je počet otáček za sekundu a a, b jsou hledané reálné konstanty.

Měření je nutné provést pro několik rychlostí (obvykle 25) v rozsahu daném výrobcem vrtule. Měření probíhá v následujícím cyklu:

1) rozjetí vozíku na žádanou rychlost
2) v okamžiku dosažení žádané rychlosti je spuštěno měření - začínají se počítat impulsy od vrtule (většina typů vrtulí dává jeden impuls na otáčku, některé typy dva) a zároveň v době příchodu prvního impulsu se uloží aktuální čas a aktuální poloha vozíku
3) jakmile je dosaženo předem zadaného počtu impulsů vrtule, uloží se koncový čas a koncová poloha vozíku a vozík začíná brzdit
4) doba pohybu a ujetá dráha vozíku (daná rozdílem počátečních a koncových hodnot) se uloží do protokolu
5) před dalším měřením vozík čeká po předem nastavenou dobu na ustálení hladiny v kanále
6) → opakuje se bod 1) pro další rychlost

Posledním krokem je pro každé měření spočítat skutečnou průměrnou rychlost a počet impulsů vrtule za sekundu. Tím získáme sadu hodnot počtu impulsů za sekundu $n_1, n_2, \ldots, n_n$ odpovídající sadě rychlostí $v_1, v_2, \ldots, v_n$. Metodou nejmenších čtverců[4] se poté z těchto hodnot spočítají konstanty a a b.

Pro zrychlení kalibrování je vozík osazen třemi tyčemi, na každou tyč je možné nasadit jednu vrtuli a měřit tak tři vrtule najednou. Je možné měřit všechny vrtule rozdílné a pro každou zadat jiný požadovaný počet impulsů. (V tom případě pak vozík v bodě 3 výše zmíněného cyklu začíná brzdit až v okamžiku dosažení počtu impulsů u všech vrtulí.) Tyče s vrtulemi jsou umístěny na sklápěcí nástavbě, která umožňuje nasazování a ponořování vrtulí do vody. Otoče (pneumatické servopohony) otáčejí tyče podle směru jízdy tak, aby vrtule byla vždy hlavou (tzv. propelerem) dopředu.

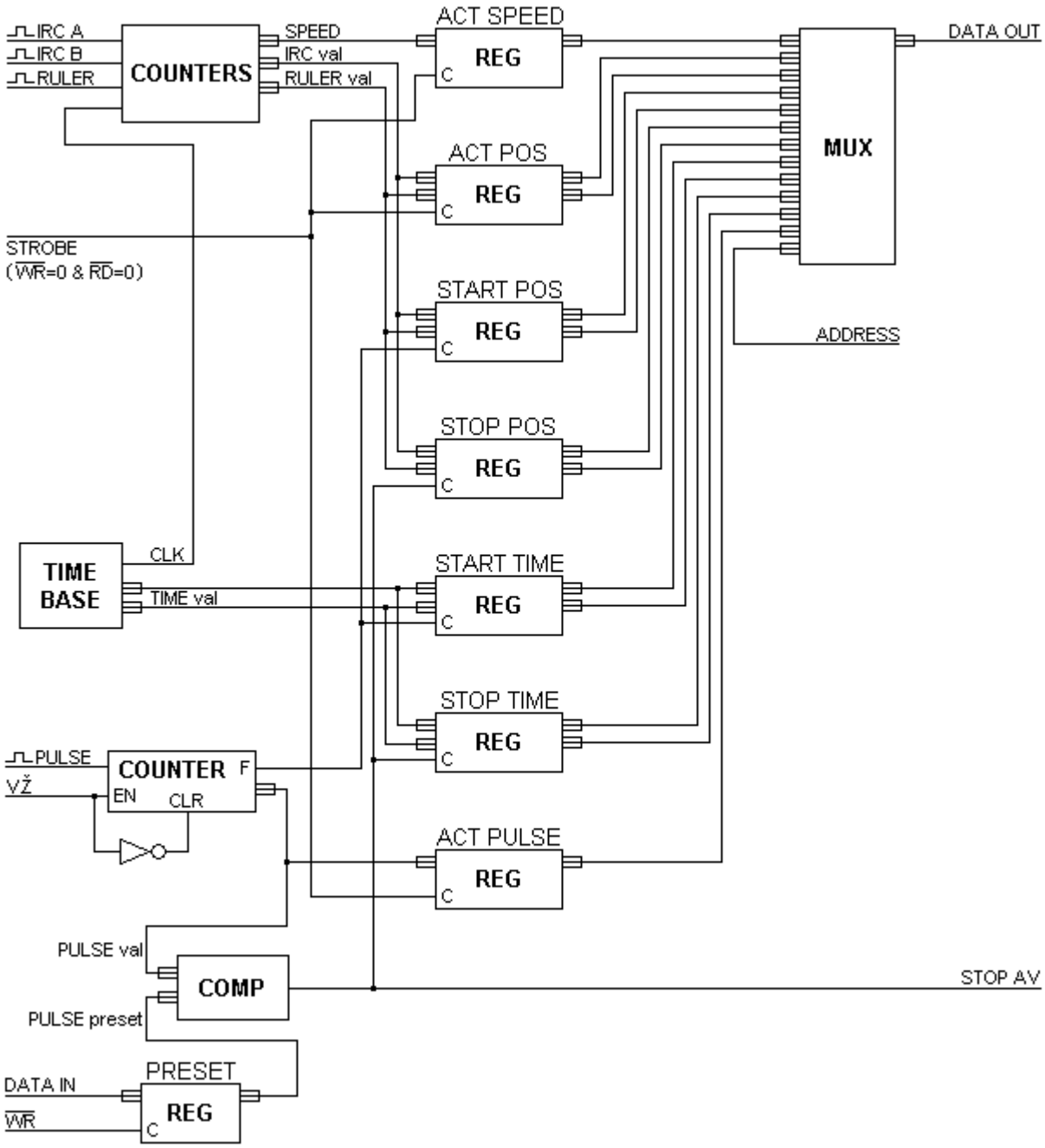
3.3.2 Schéma desky MK

Zjednodušené schéma desky MK je na obr. 3.10. Plné schéma je uvedeno v příloze D. Zapojení všech bloků použitých v plném schématu je možné nalézt na přiloženém CD-ROM.

Popíšeme si nejprve funkci jednotlivých bloků. Blok **COUNTERS**, obsahující především čítače, přijímá na vstupu pulsy z IRC, pulsy z pravítka (RULER) a časové pulsy CLK. Výstupem jsou pak aktuální počet pulsů z pravítka RULERval, počet pulsů z IRC

[4] Metoda nejmenších čtverců (přesněji její nejjednodušší verze) spočívá v proložení bodů v rovině přímkou tak, aby součet kvadratických odchylek všech bodů od této přímky byl minimální. V našem případě jsou tyto body dány souřadnicemi $[n_k, v_k]$ a hledaná přímka je $v = a + bn$.

(IRCval) a hodnota pro výpočet aktuální rychlosti (SPEED). Mezi jednotlivými dírami pravítka je vzdálenost 10 cm, pravítko tedy slouží pro hrubé zjištění polohy vozíku. Pro jemné určení polohy mezi děrami slouží právě impulsy z IRC v motorech, které jsou nulovány při každém pulsu z pravítka. Signály z IRC členu jsou dva a jsou vůči sobě fázově posunuté (IRC A a IRC B). Díky tomu je možné určit směr pohybu, který je nutné znát, aby se správně zpracovaly příchozí pulsy z pravítka, tzn. aby se přičítaly nebo odčítaly podle aktuálního směru. Výstupy z tohoto bloku jsou 16-bitové hodnoty (vícebitové spoje jsou znázorněny na vstupech nebo výstupech jednotlivých bloků vidlicemi).



(obr. 3.10 – zjednodušené schéma desky MK)

Bloky **REG** obsahují datové registry. Náběžnou hranou signálu C se obsah vstupů bloku **REG** zapíše na jeho výstupy. Výstupy těchto bloků vedou přímo do multiplexeru **MUX**, který podle hodnoty ADDRESS (6-bitová adresa) připojuje příslušný vstup na výstup DATA OUT. Lze tak tedy postupně číst všechny registry.

Blok **TIME BASE** slouží jako časová základna pro celou desku. Poskytuje periodický signál CLK a 32-bitovou (2x16 bit) hodnotu času TIMEval.

V bloku **COUNTER** je čítač, který čítá pulsy od vrtule, pokud je aktivní signál VŽ (= rychlost dosažena). V opačném případě se čítač nuluje. Výstupem je 16-bitová hodnota počtu pulsů. Další výstupní signál F svojí náběžnou hranou signalizuje příchod prvního pulsu po vynulování čítače.

Blok **COMP** (komparátor) porovnává vstupní hodnoty a pokud se rovnají, aktivuje výstupní signál.

Deska je napojena plochým kabelem do I/O desky v počítači na tři 8-bitové porty. Dva z nich zabírá 16 bitů vstupních a výstupních dat DATA IN / DATA OUT. Podle aktuálního stavu signálů RD (čtení) a WR (zápis) se tyto porty průběžně přepínají pro čtení nebo pro zápis. K operaci čtení nebo zápisu dochází sestupnou hranou příslušného signálu (oba jsou aktivní v nule). Z důvodu přehlednosti schématu jsou DATA IN a DATA OUT rozděleny, stejně jako je vynecháno ovládání přepínání pomocí signálů RD a WR (plné schéma – viz příloha D). Třetí port je po celou dobu nakonfigurován jako výstupní z počítače a obsahuje 6 bitů adresy ADDRESS a signály RD a WR.

Vstupy IRC, RULER a PULSE jsou vstupy z technologie, VŽ je vstup z desky LC a STOP AV je výstup do LC.

V průběhu jízdy se pravidelně signálem STROBE (= aktivní při RD=0 a WR=0) zapisují aktuální hodnoty rychlosti, dráhy a pulsů od vrtule na výstupy bloků ACT SPEED, ACT POS a ACT PULSE. Mezi dvěma aktivacemi signálu STROBE jsou tyto hodnoty na výstupech bloků „zmraženy“ a jsou načítány (adresováním příslušných registrů) do počítače, kde jsou použity pro vizualizaci procesu kalibrace.

Před měřením se do registru PRESET zapíše požadovaný počet impulsů, který má vrtule při měření dosáhnout. Po rozjetí vozíku a dosažení žádané rychlosti (signál VŽ) začne čítač **COUNTER** načítat pulsy od vrtule. Při příchodu prvního pulsu se signálem F zapíše aktuální počet pulsů z pravítka, z IRC a čas na výstupy bloků START POS a START TIME. Při vypadnutí signálu VŽ, např. při překmitu rychlosti při rozjezdu nebo při zpomalení vlivem nerovnosti na kolejích, se čítač pulsů resetuje. Po opětovném naběhnutí VŽ běží měření znovu a výsledek tak není zatížen chybou, která by při změně rychlosti nastala.

V okamžiku, kdy komparátor **COMP** zjistí shodu nastaveného (PULSEpreset) a aktuálního (PULSEval) počtu pulsů, měření je skončeno a aktuální hodnoty dráhy a času se výstupním signálem komparátoru zapíše na výstupy bloků STOP POS a STOP TIME. Také je aktivován signál STOP-AV, který vede k zastavení vozíku (kap. 3.2.5).

Na závěr jsou do počítače načteny výsledky měření, tedy obsahy registrů START POS, STOP POS, START TIME a STOP TIME.

Je nutné ještě upozornit na to, že zjednodušené schéma na obr 3.10 zpracovává impulsy pouze od jedné vrtule. Ve skutečnosti se však měří vrtule tři, tzn. že bloky START POS, STOP POS, START TIME, STOP TIME, COUNTER, COMP, PRESET a ACT PULSE se v plném schématu vyskytují třikrát.

3.3.3 Registry desky MK

Tabulka 3.1 uvádí přehled registrů pro čtení a tabulka 3.2 přehled registrů pro zápis.[5]

Adresa	Název registru	Obsah registru
0	*Status*	Stavový registr
1	*ActSpeed*	Aktuální rychlost
2	*ActPosIRC*	Aktuální počet pulsů IRC
3	*ActPosRul*	Aktuální počet pulsů z pravítka
4	*AStartIRC*	Počet pulsů IRC na začátku měření vrtule č.1
5	*AStartRuler*	Počet pulsů pravítka na začátku měření vrtule č.1
6	*BStartIRC*	Počet pulsů IRC na začátku měření vrtule č.2
7	*BStartRuler*	Počet pulsů pravítka na začátku měření vrtule č.2
8	*CStartIRC*	Počet pulsů IRC na začátku měření vrtule č.3
9	*CStartRuler*	Počet pulsů pravítka na začátku měření vrtule č.3
10	*AStopIRC*	Počet pulsů IRC na konci měření vrtule č.1
11	*AStopRuler*	Počet pulsů pravítka na konci měření vrtule č.1
12	*BStopIRC*	Počet pulsů IRC na konci měření vrtule č.2
13	*BStopRuler*	Počet pulsů pravítka na konci měření vrtule č.2
14	*CStopIRC*	Počet pulsů IRC na konci měření vrtule č.3
15	*CStopRuler*	Počet pulsů pravítka na konci měření vrtule č.3
16	*AHStartTime*	Horních 16 bitů hodnoty času na začátku měření vrtule č.1
17	*ALStartTime*	Dolních 16 bitů hodnoty času na začátku měření vrtule č.1
18	*BHStartTime*	Horních 16 bitů hodnoty času na začátku měření vrtule č.2
19	*BLStartTime*	Dolních 16 bitů hodnoty času na začátku měření vrtule č.2
20	*CHStartTime*	Horních 16 bitů hodnoty času na začátku měření vrtule č.3
21	*CLStartTime*	Dolních 16 bitů hodnoty času na začátku měření vrtule č.3
22	*AHStopTime*	Horních 16 bitů hodnoty času na konci měření vrtule č.1
23	*ALStopTime*	Dolních 16 bitů hodnoty času na konci měření vrtule č.1
24	*BHStopTime*	Horních 16 bitů hodnoty času na konci měření vrtule č.2
25	*BLStopTime*	Dolních 16 bitů hodnoty času na konci měření vrtule č.2
26	*CHStopTime*	Horních 16 bitů hodnoty času na konci měření vrtule č.3
27	*CLStopTime*	Dolních 16 bitů hodnoty času na konci měření vrtule č.3
28	*AActPulse*	Aktuální počet impulsů od vrtule č.1
29	*BActPulse*	Aktuální počet impulsů od vrtule č.2
30	*CActPulse*	Aktuální počet impulsů od vrtule č.3
32	*APreset*	Požadovaný počet impulsů od vrtule č.1 (pro kontrolu zápisu)
33	*BPreset*	Požadovaný počet impulsů od vrtule č.2 (pro kontrolu zápisu)
34	*CPreset*	Požadovaný počet impulsů od vrtule č.3 (pro kontrolu zápisu)

(tab. 3.1 - přehled registrů desky MK pro čtení)

[5] Registry pro čtení s adresami 0, 1, 2, 3 nesouvisí s registry pro zápis 0, 1, 2, 3, mají jen společné označení adresy. Znamená to, že při operaci zápisu na adresu X se zapisuje jinam, než se při operaci čtení z adresy X čte.

Adresa	Název registru	Obsah registru
0	*Status*	Stavový registr
1	*APreset*	Požadovaný počet impulsů od vrtule č.1
2	*BPreset*	Požadovaný počet impulsů od vrtule č.2
3	*CPreset*	Požadovaný počet impulsů od vrtule č.3

(tab. 3.2 - přehled registrů desky MK pro zápis)

V obou tabulkách se vyskytuje *Status* registr na adrese 0. O tomto registru ještě nebyla řeč a ani nebyl vyznačen ve schématu na obr. 3.10. Jeho popis je náplní následující podkapitoly.

3.3.4 Status registr

Tento registr, pokud z něj čteme, slouží pro zjišťování informací o průběhu měření. V režimu zápisu můžeme do tohoto registru zadat požadavek na měření nebo měření resetovat. Význam jednotlivých bitů registru uvádí tabulka 3.3.

Bit	Čtení	Zápis
0	*MsrActive*	*Reset*
1	*MsrRequest*	*MsrRequest*
2	*MsrDone*	*ResetCalibrated*
3	*VZ*	-
4	*Dir*	-
5	*Calibrated*	-
6 – 12	-	-
13	*CEQ*	-
14	*BEQ*	-
15	*AEQ*	*LED*

(tab. 3.3 – status registr)

Před začátkem měření je nutné nastavit bit *MsrRequest*, aby se měření spustilo[6]. K tomu poté dojde při dosažení žádané rychlosti a příchodem prvního impulsu od vrtule. Stornování měření se provede nastavením bitu *Reset. ResetCalibrated* deaktivuje bit *Calibrated*, který značí, že měření dráhy je zkalibrováno. Poslední bit v režimu zápisu je *LED*, jehož periodickou změnou rozsvěcujeme diodu na desce a můžeme tak vizuálně kontrolovat společnou správnou funkci desky a programu v počítači.

[6] Ještě je nutné naplnit registry APreset, BPreset, CPreset s požadovaným počtem impulsů.

V režimu čtení zjišťujeme, zda měření probíhá (*MsrActive*), zda byl přijat požadavek na měření (*MsrRequest*), zda měření bylo dokončeno (*MsrDone*), zda byla dosažena žádaná rychlost (*VZ*), dále kterým směrem se pohybujeme (*Dir*) a zda je zkalibrováno měření dráhy (*Calibrated*). Bity *AEQ*, *BEQ*, *CEQ* jsou aktivní, pokud příslušná vrtule dokončila měření, tzn. dosáhla požadovaného počtu impulsů.

4 Vyšší řídící úroveň

Tato kapitola je věnována popisu vyšší úrovně řízení palubním počítačem, který umožňuje automaticky provádět kalibrační měření bez účasti obsluhy na vozíku. Nejprve je uvedena specifikace palubního počítače a jeho spojení s periferiemi (kap. 4.1 a 4.2), následuje popis oken programu (kap. 4.3) a poslední kapitola 4.4 podrobně vysvětluje realizaci řídícího programu.

4.1 Palubní počítač

Řídící palubní počítač je průmyslové PC. Od klasického stolního počítače se liší v těchto bodech:

a) je uložen v robustní kovové schránce IAC WALL-MOUNT s výztuhami, obsahující příslušenství pro upevnění počítače
b) pevný disk je umístěn v ochranném antivibračním pouzdru, což je pro tuto aplikaci nutné (vibrace se při rychle jedoucím vozíku objevují)
c) všechny spoje a kontakty uvnitř počítače jsou zalepeny, aby bylo zabráněno jejich uvolňování při vibracích
d) po sestavení počítače a nainstalování operačního systému provedla dodávající firma 3-denní testování funkce počítače (za měnící se okolní teploty apod.)

4.1.1 Specifikace počítače

Palubní počítač obsahuje:

- Procesor Intel Pentim IV 1,8 GHz
- Základní deska ASUS P4GE-VM (onboard LAN), chipset Intel 82845G
- RAM 256 MB
- HDD 40 GB, FDD 3,5“
- CD-ROM MSI C52B
- I/O karta ICP DAS PIO-D96
- Převodník RS-232 / RS-485
- Operační systém MS Windows XP (5.1.2600 service pack 1)
- LCD monitor 15“ AutoCont
- Membránová klávesnice AX7020K + touchpad

Pro tuto aplikaci není nutný operační systém reálného času[7] (jako RT Linux nebo VxWorks). Je to z důvodu, že všechny úkoly, které musí být splněny v minimálním čase, jsou zajištěny deskou MK. Jedná se např. o uložení okamžité polohy vozíku po příchodu posledního impulsu od vrtule. Tato hodnota může být počítačem přečtena kdykoliv později.

Počítač je sice zařazen do procesu řízení vozíku, ovšem časy odezev nejsou tak kritické. Byl proto zvolen dostatečně stabilní systém MS Windows XP. Jedním z důvodů byla také jeho cena a dostupnost programů pro vývoj aplikací.

4.1.2 I/O karta

Tato přídavná karta od firmy ICP DAS má označení PIO-D96 [12]. Obsahuje 96 digitálních TTL vstupů/výstupů. Ty jsou rozděleny do čtyř 24-bitových obousměrných portů. Každý z těchto portů je rozdělen na tři osmibitové skupiny (porty) PA, PB a PC a každá tato skupina může být softwarově nakonfigurována buď jako vstupní nebo jako výstupní. Při připojení napájení jsou všechny porty nastaveny jako vstupní.

Karta obsahuje jeden 37-pinový D-SUB konektor (porty 0,1,2) a tři 50-pinové konektory (porty 3 – 11) pro připojení plochých kabelů. Tyto tři ploché kabely jsou vyvedeny ven z počítače, jeden z nich vede do desky MK, další dva do desky LC. Porty 0,1,2 jsou rezervní, připravené pro možné budoucí použití.

Je možné přistupovat do dvaceti dvou 8-bitových registrů adresovaných vzestupně od základní adresy, která je přidělena při inicializaci karty. Na nízkých adresách jsou pomocné registry a registry pro obsluhu přerušení. Poté následuje 12 registrů s obsahy všech 12 portů. Navíc pro každou trojici portů karta obsahuje jeden konfigurační registr, v jehož třech nejnižších bitech je možné nastavit, který z trojice portů bude vstupní a který výstupní. Toto nastavení lze kdykoliv v běhu programu změnit, což budeme potřebovat.

Softwarové vybavení obsahuje potřebné drivery a knihovny s funkcemi pro ovládání karty. Po nutné inicializaci v programu je pak možné přistupovat na porty (neboli registry karty) pomocí funkcí *OutputByte* a *InputByte*, kde se jako parametr zadá adresa registru.

Karta se instaluje do 5V PCI slotu a zajišťuje „plug and play“.

4.1.2.1 Připojení desky LC na I/O kartu

Deska LC je připojena dvěma plochými kabely na porty 6 – 11 na I/O desce v počítači. Porty 6 a 7 jsou nastaveny jako výstupní a význam jejich bitů ukazuje tab. 4.1. Jsou to signály jdoucí z počítače do desky LC.

[7] Operační systém reálného času zaručuje určité horní hranice dob splnění příkazů.

Bit	7	6	5	4	3	2	1	0
Port 6	S-A	J-A	POHON2 ON	POHON1 ON	START-A	PCWD	-	-
Port 7	-	-	OTOČ3 ON	OTOČ2 ON	OTOČ1 ON	P1-A	P2-A	STOP-AP

(tab. 4.1 – výstupní porty 6 a 7 do desky LC)

Porty 8,10,11 jsou nastaveny jako vstupní a význam bitů ukazuje tab. 4.2. Jedná se o signály jdoucí z desky LC do počítače.

Bit	7	6	5	4	3	2	1	0
Port 8	-	RDY SIST	RDY OTOČ	RDY PŘEV	RDY SMF2	RDY SMF1	P1	VOZÍK RDY
Port 10	-	ZÁVADA	PARKING	-	-	-	-	-
Port 11	VŽ	STOP DRÁHA	STOP JÍZDY	PORUCHA	J	S	A	R

(tab. 4.2 – vstupní porty 8,10,11 z desky LC)

Port 9 je rezervní.

4.1.2.2 Připojení desky MK na I/O kartu

Deska MK je připojena jedním plochým kabelem na porty 3,4,5 na I/O desce. Na portech 3 a 4 se čtou nebo zapisují data a podle toho se tyto porty přenastavují na vstupní nebo výstupní. Port 5 je konstantně nastaven jako výstupní a obsahuje 6-bitovou adresu a signály RD a WR. Porty 3 a 4 vlastně představují sběrnici, přes kterou se adresováním přistupuje k jednotlivým registrům desky MK. Obsazení portů ukazuje tabulka 4.3.

Bit	7	6	5	4	3	2	1	0
Port 3	DATA - NIŽŠÍ BYTE SBĚRNICE							
Port 4	DATA - VYŠSÍ BYTE SBĚRNICE							
Port 5	WR	RD	ADRESA					

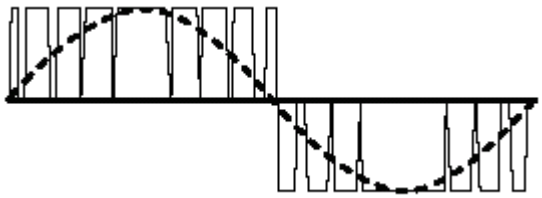
(tab. 4.3 – porty 3,4,5 do/z desky MK)

4.2 Spojení počítače s frekvenčními měniči

Pro řízení motorů jsou použity dva frekvenční měniče Frenic 5000G11S-EN od firmy Fuji [10]. Funkci měničů pro řízení rychlosti asynchronního motoru [9] stručně shrnuje následující odstavec.

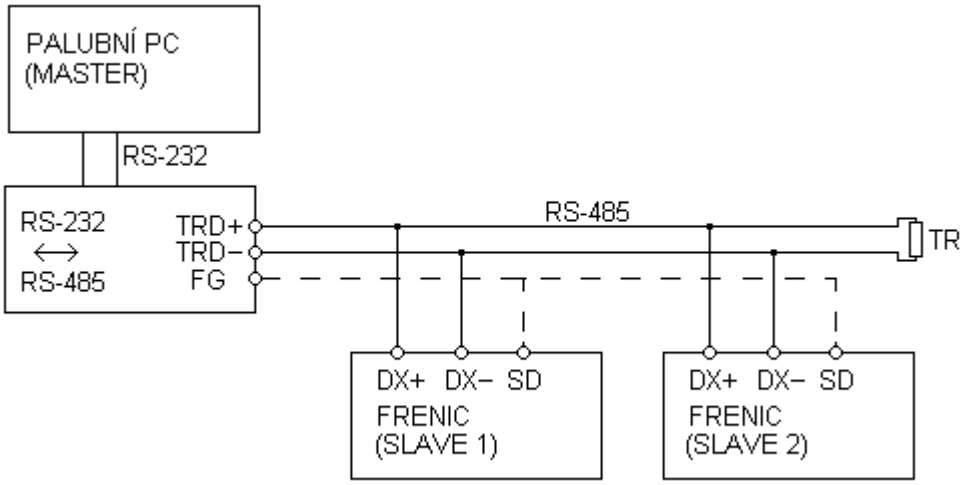
K pohonu motorů se používá trojfázového napájení (3x380 V, 50 Hz), kde jednotlivé fáze jsou vůči sobě posunuté o 120°. Asynchronní motor obecně obsahuje stator a rotor. Na statoru jsou navinuty tři vinutí, každé z nich je připojené na jednu fázi vedení. Po připojení napájení vznikne na statoru točivé elektromagnetické pole (díky fázovému posunu na jednotlivých vinutích), které roztočí rotor spojený s hřídelí. Aby bylo možné měnit rychlost otáčení rotoru, je nutné měnit rychlost otáčení elektromagnetického pole, tzn. měnit frekvenci

střídavého napětí na vinutích. To zajišťuje právě frekvenční měnič. Generování střídavých napětí o různé frekvenci se v měniči provádí pulsně-šířkovou modulací (PWM) stejnosměrného napětí (obr. 4.1), které vzniká usměrněním původního síťového napětí. Z motorů jde zpět do měničů zpětná vazba, informující o skutečné frekvenci, která slouží k regulaci na frekvenci žádanou.



(obr.4.1 – pulsně-šířková modulace PWM)

Spojení palubního počítače a frekvenčních měničů je znázorněno na obr. 4.2.



(obr. 4.2 – sběrnice RS-485 spojující počítač a měniče)

Fyzická vrstva komunikace je zajištěna po průmyslové sběrnici RS-485 (FIELDBUS), kterou podporují frekvenční měniče Fuji. Pro připojení počítače na tuto sběrnici je použito převodníku RS-232/RS-485. RS-232 i RS-485 jsou normy EIA (Electronics Industry Association), definující mechanické, elektrické a logické vlastnosti spojení. Základní rozdíly mezi nimi stručně uvádějí následující dva odstavce.

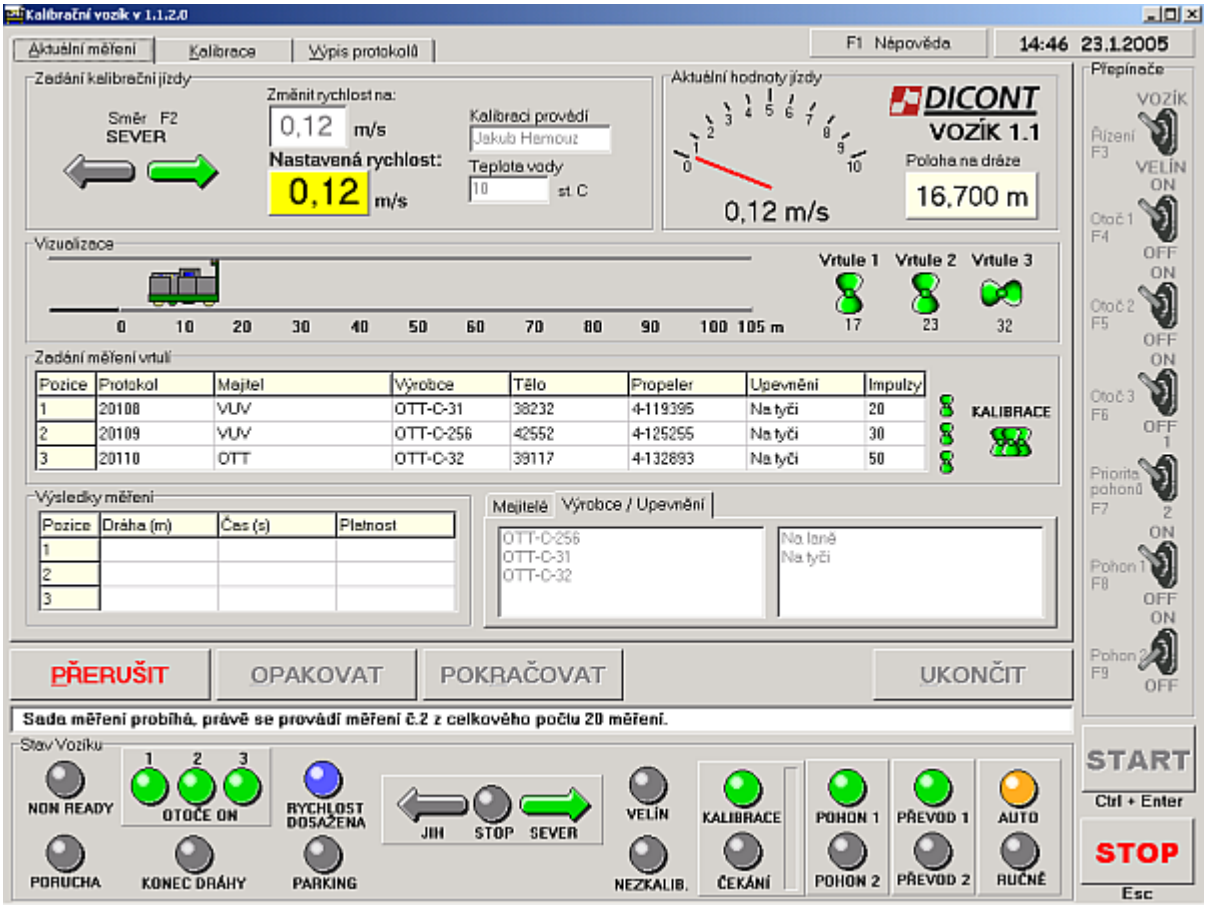
Linka RS-232 [6,7] je standardní sériová linka používaná v osobních počítačích pro připojování myši nebo modemu, je obvykle vyvedená na porty COM1, COM2, …. Je určená pouze pro *point-to-point* komunikaci (tzn. mezi dvěma zařízeními – jeden vysílač, jeden přijímač) a funguje pomalou přenosovou rychlostí (max. 20 kbit/s) na krátkou vzdálenost (max. 30 m). Signály jsou reprezentovány napěťovými úrovněmi vzhledem ke společné zemi obou zařízení, obvody rozhraní jsou nesymetrické a nejsou tak odolné vůči rušení.

RS-485 [6,8] je průmyslová sériová sběrnice. Může na ní být připojeno až 32 přijímačů a neomezený počet vysílačů. Většinou se však připojují zařízení, která mohou vysílat i přijímat. Tato zařízení se dělí na jednotky řídící (MASTER) a jednotky jim podřízené (SLAVE). Většinou je na sběrnici připojen jeden MASTER a několik jednotek typu SLAVE (tzv. *multi-drop* komunikace), kdy MASTER vysílá příkazy příslušným jednotkám (pomocí adresování) a čeká na jejich potvrzení (případně příjem dat). Délka sběrnice může být maximálně 1200 m, nejvyšší přenosová rychlost je 10 Mbit/s. Přenosovým médiem je kroucený dvoudrát se stíněním. Odolnost proti rušení v průmyslových provozech je zajištěna nízkou impedancí a symetrickým uspořádáním.

Na obr. 4.2 vidíme, že v našem případě se jedná o jednu jednotku MASTER (palubní počítač) a dvě jednotky typu SLAVE (frekvenční měniče). Konektory DX+, DX- na měničích značí diferenční vstupy/výstupy. Na konektor SD je připojeno stínění, které je na převodníku RS-232/RS-485 připojeno ke krytu (FG). Zakončovací resistor TR bývá hodnoty přibližně 120 Ω (na druhém konci sběrnice je integrován přímo do převodníku).

4.3 Popis oken řídícího programu

Hlavní okno programu je znázorněno na obr. 4.3.



(obr. 4.3 – hlavní okno programu)

Obrazovka je rozdělena na tři panely. Na největším z nich, zabírajícím většinu uživatelské plochy programu, může obsluha přepínat mezi třemi kartami – *Aktuální měření*, *Kalibrace* a *Výpis protokolů*. Na druhém panelu s názvem *Stav vozíku*, umístěném ve spodní části obrazovky, jsou zobrazovány aktuální hodnoty binárních signálů, informující o stavu vozíku. Poslední panel na pravé straně obsahuje přepínače, kterými se mění stav některých binárních signálu jdoucích do vozíku. Dále je na obrazovce pět tlačítek a informační řádek, kde se uživatel dozvídá slovní informace o stavu kalibračního procesu a o stavu vozíku, včetně diagnostických informací při poruchách. Panely se signálkami, přepínači a tlačítky nejsou překrývány a jsou viditelné po celou dobu běhu programu, aby mohla obsluha kontrolovat stav vozíku a řídit proces kalibrace.

Program se ovládá pomocí klávesnice a touchpadu. Uživateli je umožněno zrychlit si práci s programem pomocí celé řady klávesových zkratek. Např. tlačítko START lze stisknout pomocí touchpadu nebo současným stiskem kláves *Ctrl + Enter.* Podobné zkratky existují pro všechna tlačítka a přepínače, a také pro přepínání mezi kartami. Seznam těchto zkratek je uveden v příloze E.

Vzhled okna, včetně povolení ovládání určitých prvků, se mění podle aktuálního stavu vozíku. Prvky, které jsou v daném okamžiku povolené, jsou graficky zvýrazněné (např. pět horních přepínačů na obr. 4.6), naopak prvky zakázané jsou matné (spodní dva přepínače). Podobně tabulky pro zadávání parametrů měření, které jsou v automatickém režimu povolené, se v ručním režimu zakážou a zmatní, apod.

Podívejme se nyní na jednotlivé panely obrazovky, začneme panelem s přepínači. Uživatel má k dispozici sedm přepínačů. První z nich slouží pro předání řízení vozíku do velína (do dohlížecí úrovně) a zpět. Dalších šest zajišťuje zařazení a vyřazení z provozu některých akčních členů. Tři přepínače *Otoč 1, 2, 3* řídí stejnojmenné signály do LC a je jimi možné zapnout nebo vypnout automatické otáčení tyčí s vrtulemi podle navoleného směru jízdy. Je tedy např. možné vypnout dvě otoče v případě, kdy se měří jen jedna vrtule, nebo vypnout otoče při kalibraci torpéd připevněných k tyči lanem, kdy otáčení není žádoucí. Další přepínač *Priorita pohonů* má význam pouze v automatickém režimu. Určuje, který pohon bude spouštěn při nižších rychlostech, pro které nejsou zapotřebí oba pohony. Význam tohoto přepínače je v rovnoměrném zatěžování obou pohonů a je po určitém časovém úseku přepínán automaticky, pokud tak uživatel sám neučiní. Poslední dva přepínače *Pohon1, 2* jsou funkční pouze v ručním režimu a představují stejnojmenné signály do LC. Pomocí nich může uživatel rozhodnout, zda daný pohon pojede nebo se zařadí do neutrálu.

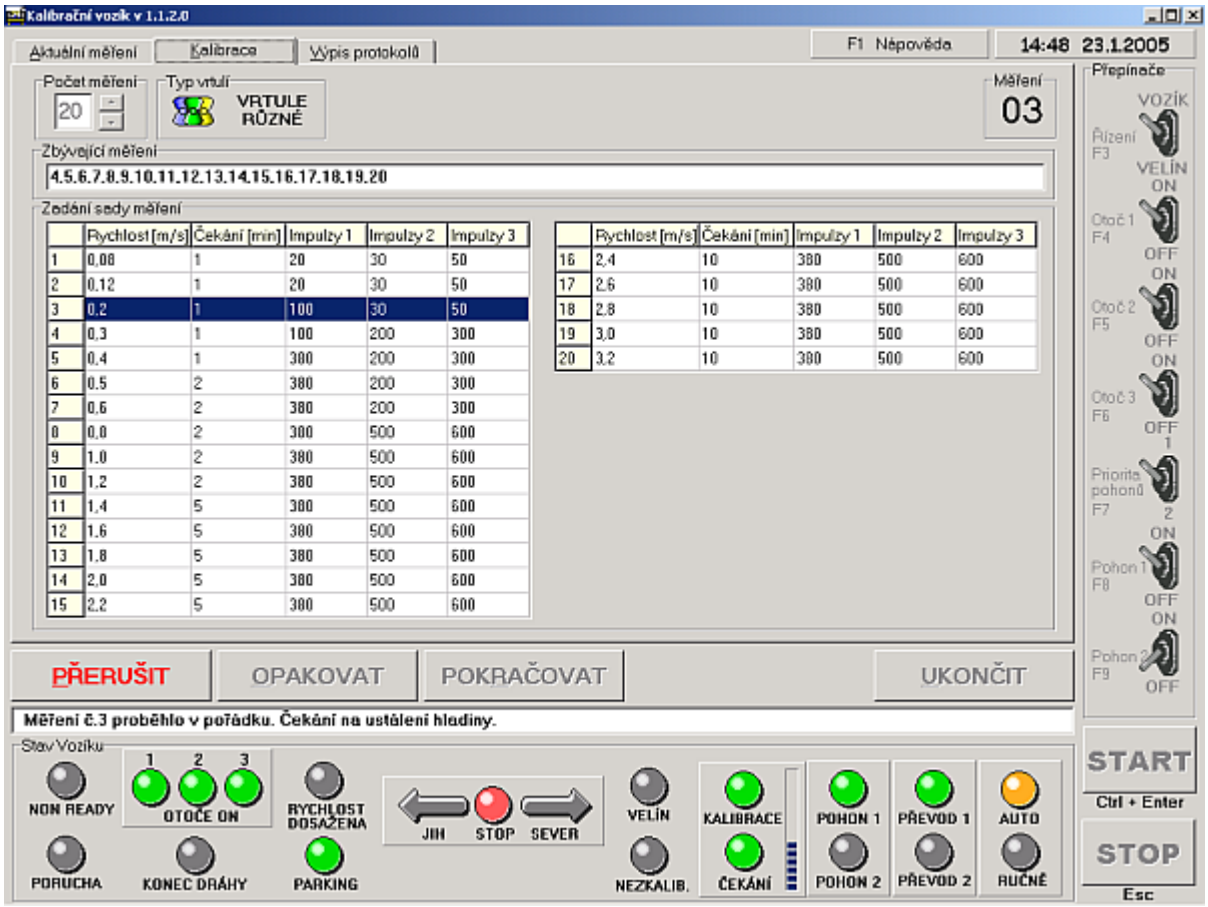
Pod přepínači v levém spodním rohu se nacházejí dvě nejdůležitější tlačítka. Jsou to tlačítka START pro rozjetí vozíku a začátek měření (a současně také start celé kalibrační sady) a tlačítko STOP, kterým se vozík okamžitě zastaví.

Na spodním panelu *Stav vozíku* se nacházejí signálky. Některé z nich jsou přímo výstupy z desky LC, shodné se signály na řídícím pultu, jsou to: NON READY, PORUCHA, KONEC DRÁHY, RYCHLOST DOSAŽENA, STOP, PŘEVOD1 (P1), PŘEVOD2 (P2), AUTO/RUČNĚ (A/R). Pokud je aktivní signál JÍZDA (START), pak svítí příslušná šipka SEVER/JIH podle směru jízdy. Signálka PARKING (zabrzděno) představuje také výstup z LC, ten však není vyveden na ŘP. Ostatní signálky jsou dány výstupem z počítače. POHON 1, 2 ukazují aktivaci pohonů, OTOČ 1, 2, 3 aktivaci otočí. KALIBRACE informuje o tom, že probíhá kompletní automatická kalibrace pro všechny rychlosti a ČEKÁNÍ ukazuje, že se právě čeká mezi měřeními na ustálení hladiny (uplynutý čas se graficky zobrazuje ve sloupci vedle signálek). NEZKALIB. značí, že údaj o absolutní poloze vozíku na dráze není správný a je tedy nutné vozík zkalibrovat projetím jižního koncového spínače. Poslední signálka VELÍN svítí, pokud je vozík řízen z velína.

Na největším panelu může uživatel přepínat mezi kartami. Obr. 4.3 ukazuje aktivní kartu *Aktuální měření.* Její funkce spočívá v zadání parametrů vrtulí, zadání jedné kalibrační jízdy a vizualizaci procesu měření. V horním pravém panelu *Zadání kalibrační jízdy* obsluha zadá směr jízdy, rychlost, své jméno a teplotu vody. Ve vedlejším panelu *Aktuální hodnoty jízdy* se na tachometru zobrazuje rychlost a vedle absolutní poloha vozíku na dráze. V podlouhlém panelu *Vizualizace* se graficky znázorňuje jedoucí vozík na dráze a točící se

vrtule ve vodě. V panelu *Zadání měření vrtulí* je před kalibrací nutné vyplnit parametry vrtulí. Jedná se o číslo kalibračního protokolu pro danou vrtuli, jméno (s adresou) majitele a výrobce vrtule, číslo těla a číslo propeleru (hlavy) vrtule, způsob upevnění (na tyči nebo na laně) a v případě, že provádíme jen jedno měření, je nutné zadat také požadovaný počet impulsů. Kliknutím na zelené vrtulky je možné aktivovat nebo deaktivovat kalibraci jednotlivých vrtulí, aby bylo zřejmé, na kterých pozicích (1,2,3) se bude kalibrovat, a které řádky v tabulce je tudíž nutné kontrolovat pro správnost zadaných údajů. Pokud jsou deaktivovány všechny vrtule, provede se po startu jízda bez měření vrtulí. Trojící větších zelených vrtulí úplně napravo (přepínač JEDNO MĚŘENÍ / KALIBRACE) se určuje, zda se bude provádět jedno samostatné měření pouze pro jednu rychlost nebo celá sada, tedy kompletní kalibrační měření pro všechny rychlosti. V případě nastavení kalibrace se aktivuje záložka *Kalibrace* v horní části obrazovky, jejímž výběrem se zobrazí karta s tabulkami pro zadání kalibračního měření. Ve spodním levém panelu se zobrazují *Výsledky měření*, které právě proběhlo, a jeho platnost. A v posledním pravém dolním panelu se nacházejí databáze *Majitelů*, *Výrobců* a *Upevnění*, které uživateli usnadňují práci při zadávání parametrů vrtulí.

Na obr. 4.4 je aktivní výše zmíněna karta *Kalibrace*.



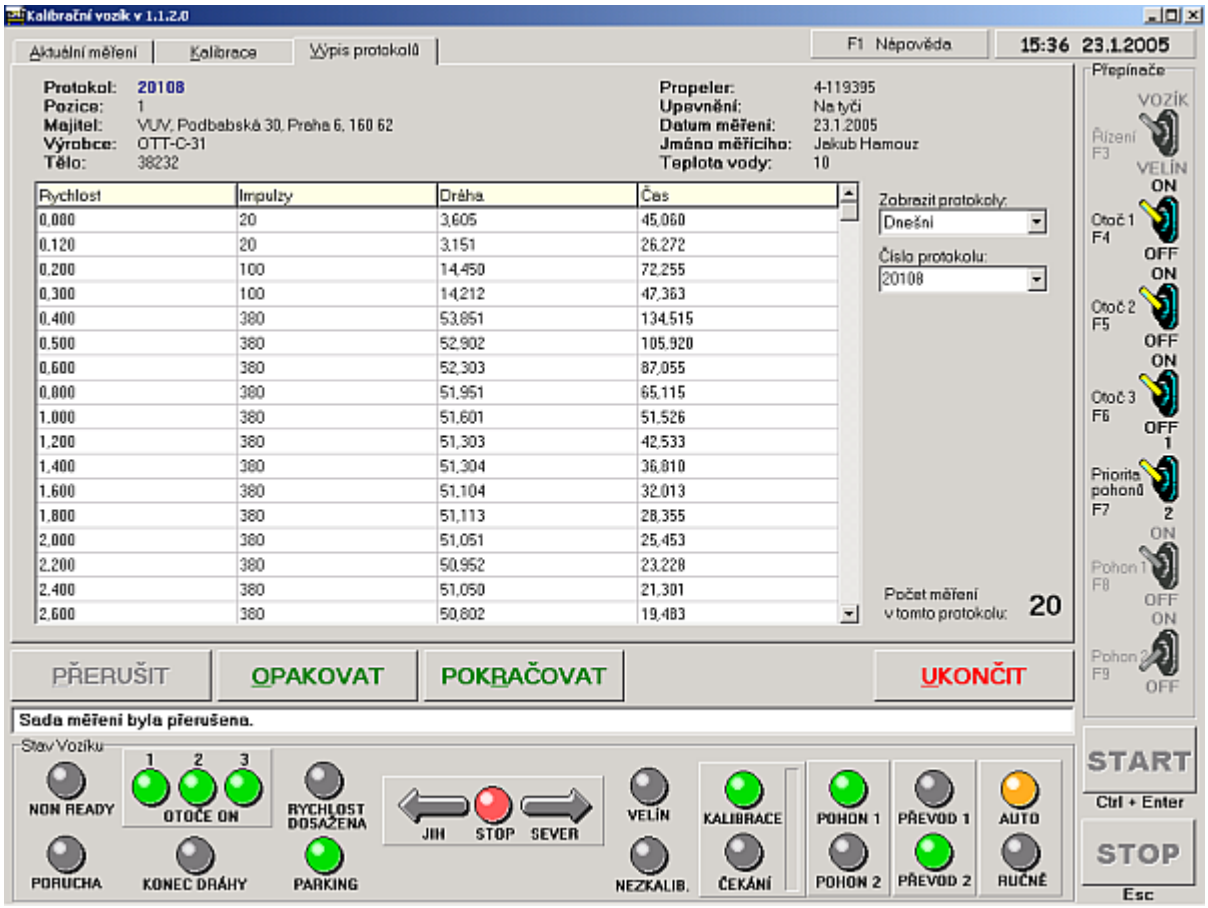
(obr. 4.4 – zadání parametrů kalibrace)

V horní části uživatel zadá počet měření a určí, zda se budou měřit vrtule stejné nebo různé. V případě stejných vrtulí se v tabulce zobrazí jen jeden sloupec pro zadání počtu impulsů, aby uživatel nemusel stejnou hodnotu vyplňovat několikrát. Podle zadaného počtu měření se upraví tabulka. V každém řádku tabulky je pak nutné vyplnit parametry jednoho měření – tedy rychlost, žádané počty impulsů od vrtulí a čekací dobu. Tlačítka pod kartou

umožňují řídit průběh kalibrace. Je možné ji kdykoliv přerušit tlačítkem PŘERUŠIT s tím, že právě prováděné měření se dokončí (pro okamžité zastavení je nutné stisknout tlačítko STOP). V této chvíli může obsluha provést změnu zadání kalibrace nebo tlačítkem OPAKOVAT zadat čísla měření, která se mají zopakovat. Tlačítkem POKRAČOVAT se kalibrace znovu rozjede a tlačítkem UKONČIT je ukončena.

V průběhu kalibrace se v řádku *Zbývající měření* zobrazuje seznam čísel měření, která ještě nebyla provedena a která budou následovat.

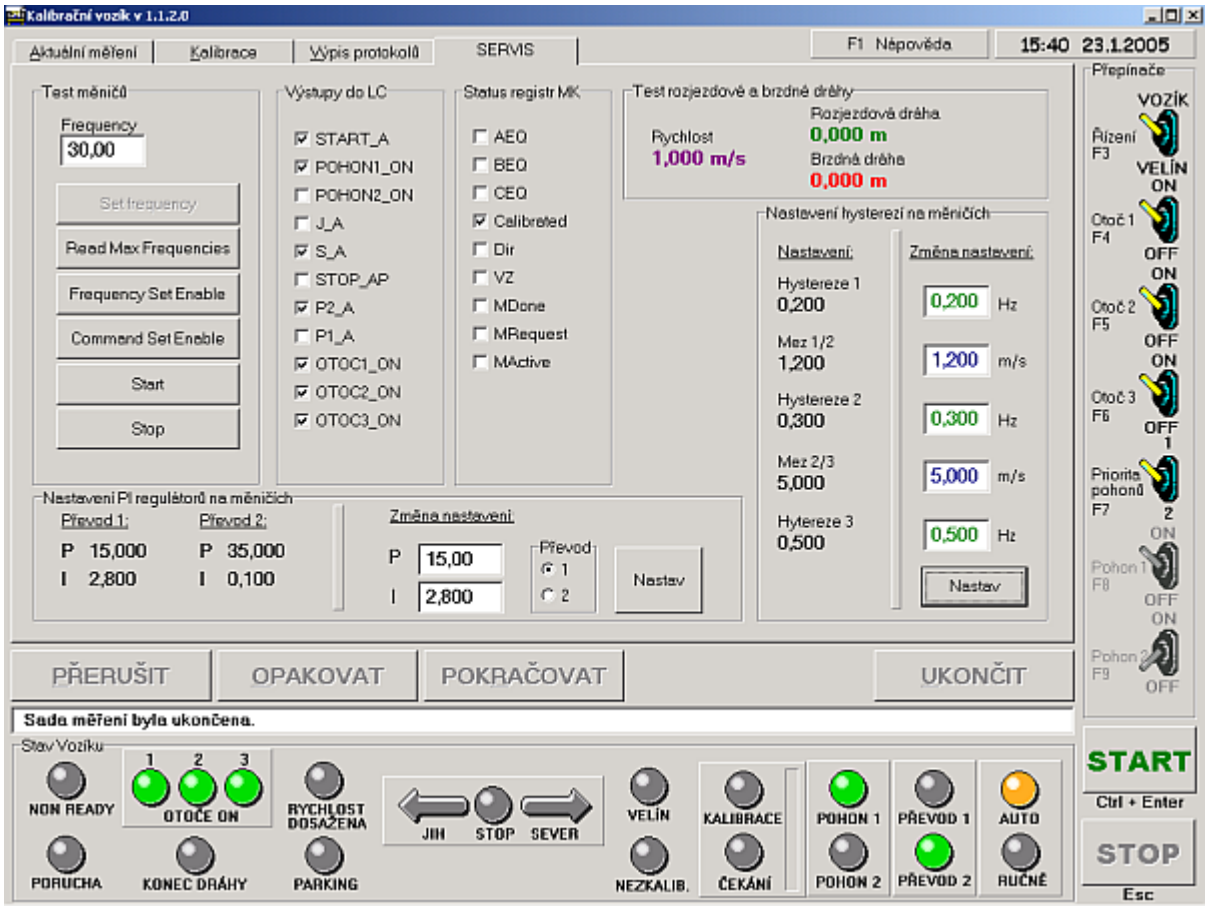
Na obr. 4.5 je karta *Výpis protokolů*, kde si může uživatel prohlížet jednotlivé protokoly. Kliknutím na pole v pravé části *Zobrazit protokoly* se zobrazí tři možnosti - *dnešní*, *za poslední týden* nebo *všechny*. Tento výběr ovlivní, které protokoly se budou vypisovat v poli *Číslo protokolu*, z kterého bude uživatel vybírat požadovaný protokol.



(obr. 4.5 – prohlížení protokolů)

V programu je však možné otevřít ještě jednu kartu, která je skrytá obsluze programu, a je možné ji aktivovat pouze pomocí hesla. Je to servisní karta a je znázorněna na obr. 4.6. Její hlavní funkce byla při oživování vozíku, ale nadále slouží pro servisní účely. Je možné pomocí ní zkoušet funkci měničů na panelu *Test měničů*, tedy nastavení žádané frekvence, maximální frekvence a nastavování módů frekvenčních měničů (FWD, REV a STOP). Na panelu *Nastavení PI regulátorů na měničích* je možné nastavit hodnoty proporcionální (P) a integrální (I) složky regulátoru rychlosti. Toto nastavení se liší pro první a druhý převod a po přeřazení převodovky se v automatickém režimu posílají do měniče příslušné konstanty

regulátoru. Panel *Nastavení hysterezí na měničích* umožňuje nastavení hystereze rychlosti (resp. frekvence) pro různé intervaly rychlosti. Hystereze udává maximální hodnotu, o kterou se může snížit nebo zvýšit skutečná rychlost (frekvence) od žádané, aby byla stále brána jako žádaná. Při změně žádané rychlosti se pak v automatickém režimu posílá příslušná hodnota hystereze do měničů. Pomocí panelu *Výstupy do LC* se přímo ovládají výstupy z počítače do desky LC a na panelu *Status registr MK* se zobrazuje aktuální hodnota jednotlivých bitů stavového registru desky MK. Poslední panel *Test rozjezdové a brzdné dráhy* sloužil při jízdních zkouškách vozíku a ukazuje délku dráhy při rozjezdu na žádanou rychlost a délku dráhy brzdné.



(obr. 4.6 – servisní karta)

4.4 Realizace řídícího programu

Program je kompletně napsán v objektově orientovaném jazyce C++ za použití vývojového prostředí Borland C++ Builder verze 5.0 [3,4,5]. Struktura programu je vhodně rozdělena do objektů tak, aby objekty vyjadřovaly reálné i abstraktní součásti celého systému a bylo tak dosaženo co největší přehlednosti kódu. Do takové struktury se také snadno doplňují nové funkce programu v případě potřeby. Aplikace je psána jako více-vláknová. Popis objektově orientovaného programování za použití vláken v operačním systému Windows není náplní této práce a přepokládá se, že čtenář danou problematiku zná.

Kromě základních funkcí a příkazů jazyka C++ je v programu použita knihovna VCL (Visual Component Library), která je součástí vývojového prostředí, a která obsahuje předdefinované objekty (komponenty). Tyto objekty zjednodušují programování a jsou jimi např. viditelné komponenty jako okno, editační pole nebo tlačítko. Program v hojné míře také využívá funkcí Windows API[8] [2], které jsou nezbytné v případech, kdy knihovna VCL nestačí. Všechny třídy (objekty), které budou v následujícím textu uvedeny, byly napsány přímo pro tuto aplikaci, pokud není uvedeno jinak.

Při psaní kódu byl kladen důraz na to, aby výsledný program splňoval následující podmínky:

1) stabilita – program se nesmí nikdy dostat do stavu tzv. „zamrznutí“[9] nebo způsobit kolizi operačního systému
2) jedná o řídící program, proto musí být všechny stavy vozíku v programu jasně definované, nesmí existovat stav, který program nebude znát
3) uživateli nesmí být dovoleno zadat nepřípustný příkaz, jenž by mohl vést např. ke kolizi vozíku (v nejhorším případě)
4) všechny vstupy od uživatele musí být kontrolovány a v případě jejich nesprávnosti uživateli ohlášena chyba, vč. jejího popisu a způsobu jak ji odstranit
5) program musí být lehce ovladatelný, pomocí touchpadu i klávesových zkratek
6) grafické rozhraní musí být přehledné a uspořádané do logicky souvisejících bloků

4.4.1 Simulátor vozíku

Vývoj řídícího programu pro kalibrační vozík byl započat ve stejnou dobu jako vývoj vozíku samotného včetně jeho ostatních řídících systémů. Z toho vyplývá, že program nebylo možné zkoušet a odlaďovat na reálné aplikaci. Nutností v průběhu vývoje tedy bylo vytvoření simulátoru vozíku. Okno této aplikace (KARTA.EXE) je znázorněné na obrázku 4.7.

Tato aplikace simuluje funkci I/O karty a desek LC a MK, včetně signálů jdoucích do těchto desek z ostatních částí vozíku. Dochází tak k simulaci celé kalibrační jízdy.

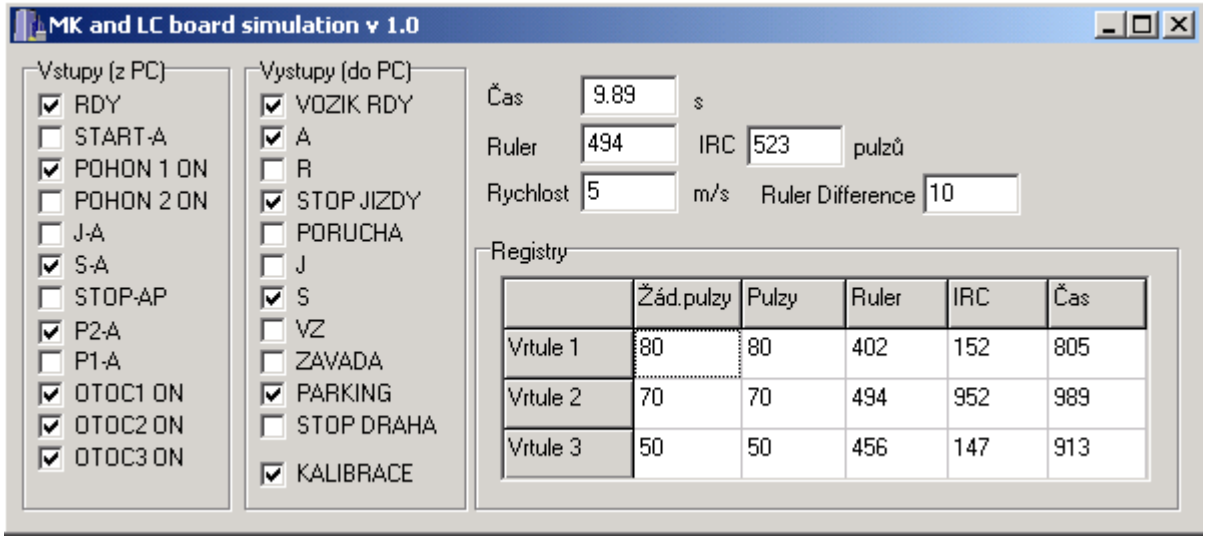
Zdrojový kód programu je možné přeložit buď pro simulaci nebo pro reálný vozík, a to nastavením podmíněného překladu v souboru *compiler.h*. Určité části kódu, ve kterých se komunikuje s I/O deskou, jsou tedy psané dvakrát a vždy se překládá jen příslušná polovina z nich. Části psané pro simulaci obsahují obdoby funkcí *InputByte* (čtení portu na I/O desce) a *OutputByte* (zápis na port I/O desky) nazvané *InPort* a *OutPort,* které nastavují a čtou simulované porty pomocí zasílání zpráv[10] mezi řídícím programem vozíku a simulátorem. Části kódu pro simulaci a pro reálný vozík jsou tedy v podstatě stejné až na použití jiných funkcí pro čtení a zápis na porty.

---

[8] Ve zdrojovém kódu jsou označeny čtyřtečkou, např. *::WaitForSingleObject( )*

[9] Stav „zamrznutí“ je ve více-vláknových aplikacích snadno dosažitelný, např. v případě, že jedno vlákno čeká na výskyt události, která je aktivována z druhého vlákna, a současně toto druhé vlákno čeká na událost aktivovanou z prvního vlákna (dojde k tzv. *deadlocku*)

[10] API funkce SendMessage

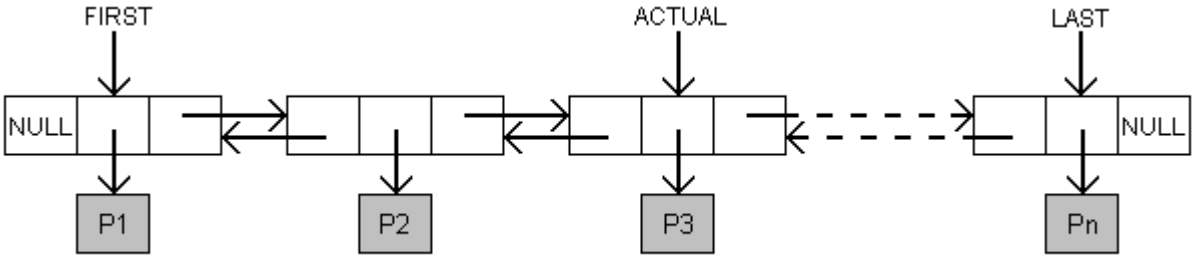


(obr.4.7 – simulátor vozíku)

4.4.2 Obecné třídy

Tato podkapitola uvádí stručný přehled vyvinutých tříd, které mají obecné použití a od kterých se v programu dědí další třídy s konkrétním použitím.

Šablona (*template*) třídy **T_List<p>** implementuje dynamický obousměrný seznam znázorněný na obr. 4.8., kde proměnným parametrem *p* je typ prvku seznamu. Pomocí této šablony tedy můžeme vytvářet třídy reprezentující seznamy prvků libovolných typů (např. **T_List<int>** je definice seznamu celých čísel). Metody **T_List<p>** umožňují všechny základní operace se seznamem, především vkládání a odebírání prvků.



(obr. 4.8 – dynamický obousměrný seznam, šipky představují ukazatele, šedá políčka jednotlivé prvky)

Šablona třídy **T_Queue<p>** implementuje obecnou frontu (s vlastnostmi podobnými šabloně třídy **T_List<p>**).

Třída **TRowList** je základní třídou pro databáze[11] použité v programu. Umožňuje tvořit tabulku, kde každým prvkem tabulky je nějaký řetězec znaků (AnsiString). Tato třída je zděděna od třídy **T_List<TRow>**, kde pomocná třída **TRow** obsahuje jeden řádek tabulky

[11] Jiným řešením je použití komponent VCL pro databázové aplikace, jako **TDatabase**, **TDataSource**, **TTable** apod. Pro tuto aplikaci je však toto řešení zbytečně sofistikované.

(jednorozměrné pole prvků typu AnsiString). Jinak řečeno, jedná se o dynamický seznam řádků tabulky. Počet sloupců (neboli počet prvků v každém řádku) se zadává v konstruktoru třídy **TRowList**. Kromě zděděných vlastností obsahuje tato třída také možnost vytvoření „nultého řádku“ se všeobecnými informacemi o tabulce a dále ukládání tabulky do souboru a její opětovné načítání.

Třída **TsThread**[12] je společným předkem pro všechna sekundární vlákna v programu. Je třídou abstraktní, tzn. že není možné přímo vytvořit její instanci. Je nutné nejdříve vytvořit potomka a v něm definovat funkci *ThreadProc*, která obsahuje smyčku vlákna. Použitím sekundárních vláken je umožněn paralelní běh několika smyček programu, což je pro naší aplikace nezbytné. Hlavním důvodem je nutnost současné komunikace s několika periferiemi.

4.4.3 Hlavní vlákno programu

Hlavní vlákno programu je spuštěno v základní funkci *WinMain* a zároveň je otevřeno hlavní okno implementované třídou **TformMainVUV**. Ta je potomkem třídy **TForm** z VCL, jenž je předkem pro všechna okna. Tato třída vytváří grafické rozhraní aplikace a zapouzdřuje všechny prvky umístěné v okně, jako jsou editační pole, tabulky, tlačítka apod.

Nejdůležitějším prvkem v hlavním vlákně je časovač **timerTrolleyStates**. Ten každých 100 ms spouští obslužnou funkci, která nejprve načte aktuální vstupy z desky LC. Na základě těchto signálů pak vyhodnocuje podmínky pro přechody mezi stavy vozíku. V průběhu vývoje programu se ukázaly jako vhodné dvě skupiny stavů, které můžeme vyjádřit dvěma stavovými diagramy, a jejichž popis je náplní následujících dvou podkapitol 4.4.3.1 – 2. Obslužná funkce časovače dále zajišťuje provozní brždění (kap. 4.4.3.3).

Aktuální vstupy z desky LC se ukládají do třídy **TTrolley**, která se stará o rozsvěcování příslušných signálek.

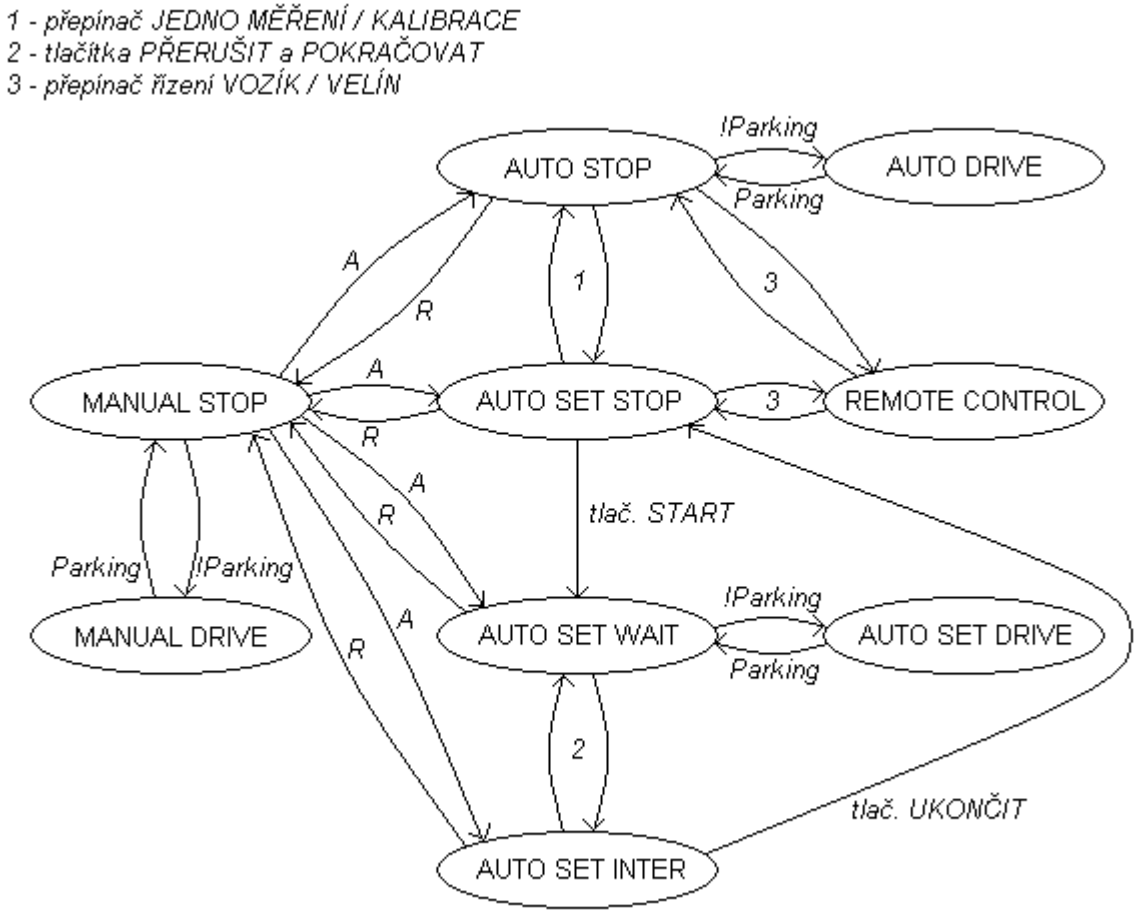
Dalšími funkcemi hlavního vlákna je spouštění obslužných funkcí v reakci na události od prvků v okně (např. změna editačního pole nebo stisk tlačítka), dále obsluha klávesových zkratek (realizovaná funkcí volanou při stisku jakékoliv klávesy), kterými se zrychluje ovládání programu. U těchto zkratek je nutné zajistit, aby nebyly použity v okamžiku, kdy je otevřeno nějaké modální okno (např. pro zadání opakování měření). V tom případě totiž dojde k „zamrznutí“ programu, protože aktivní modální okno zmizí, ale stále existuje, a hlavní okno pak nelze ovládat. V hlavním vláknu také dochází ke zpracování databází a protokolů (kap. 4.4.3.4).

Při nabíhání programu (resp. hlavního vlákna) probíhá řada inicializačních operací včetně načítání konstant pro měření a jiných parametrů ze souboru *VUV.INI* uloženého v adresáři s operačním systémem WINDOWS. Mimo jiné je např. zkontrolováno, jako dlouho již uživatel nezměnil prioritu motorů. Pokud tato doba přesáhne určitou nastavenou hodnotu (14 dní), je priorita automaticky změněna.

[12] Je možné také použít předdefinovanou třídu **TThread** z knihovny VCL. Při vytváření této aplikace však byla snaha o napsání vlastního kódu za použití API funkcí všude tam, kde to bylo jen možné. Odměnou za to byla vyšší míra kontroly nad programem a větší přehlednost vlastního kódu.

4.4.3.1 Přepínání mezi stavy vozíku

Vozík se v každém okamžiku své práce nachází v nějakém stavu. Program tento stav musí znát a podle toho povolovat nebo blokovat určité vstupy od uživatele. Z toho důvodu byla vytvořena sada devíti stavů. Jejich názvy a přechody mezi nimi znázorňuje stavový diagram na obr. 4.9.



(obr. 4.9 – stavový diagram znázorňující přechody mezi stavy vozíku)

Základní stav je MANUAL STOP, kdy je vozík v ručním režimu a stojí. Pokud uživatel vozík rozjede pomocí tlačítek na řídícím pultu (což se projeví v deaktivaci signálu PARKING z desky LC), přejde vozík do stavu MANUAL DRIVE. Po zabrždění se vrací zpět do MANUAL STOP.

Ve stavu MANUAL STOP je možné z řídícího pultu přepnout řízení vozíku z ručního režimu do automatického (deaktivace signálu R z desky LC a aktivace signálu A). Tím vozík přechází do jednoho ze čtyř stavů – AUTO STOP, AUTO SET STOP, AUTO SET WAIT nebo AUTO SET INTER. Vždy přejde do toho stavu, ze kterého se naposledy přešlo do MANUAL STOP. Po spuštění programu je tímto stavem implicitně AUTO STOP, který značí, že vozík je v automatickém režimu, stojí a bude se provádět jedno samostatné měření pro jednu rychlost nebo, pokud jsou všechny vrtule deaktivovány, servisní jízda bez kalibrace

vrtulí. Po rozjetí tlačítkem START vozík přejde do stavu AUTO DRIVE (podobně jako v případě ručního režimu).

Ze stavu AUTO STOP je dále možné přepínačem JEDNO MĚŘENÍ / KALIBRACE přejít do stavu AUTO SET STOP, který značí, že se bude provádět kompletní kalibrace vrtulí pro všechny rychlosti.

Tlačítkem START přejde vozík ze stavu AUTO SET STOP do AUTO SET WAIT a tím se zároveň rozjede vlákno **TThreadMsrSet**, které se stará o průběh celé kalibrace.

Ve stavu AUTO SET WAIT se vozík nachází, pokud zrovna čeká před dalším měřením na ustálení hladiny, nebo pokud čeká na odchod obsluhy z vozíku na začátku nebo po přerušení kalibrace. Pokud se právě provádí jedno ze sady měření nebo se dojíždí na konec dráhy před dalším měřením, nachází se ve stavu AUTO SET DRIVE. Přechody mezi těmito stavy má opět na svědomí signál PARKING.

V průběhu kalibrace je možné ji kdykoliv přerušit tlačítky PŘERUŠIT nebo STOP. Pokud se vozík nachází ve stavu AUTO SET WAIT, přejde okamžitě do stavu přerušení AUTO SET INTER. Pokus se nachází v AUTO SET DRIVE, přejde po zastavení do AUTO SET WAIT a teprve pak do AUTO SET INTER. Při použití tlačítka PŘERUŠIT se právě prováděné měření dokončí a pak se vozík zastaví, při použití STOP se zastaví okamžitě.

Ve stavu přerušení kalibrace je možné upravovat zadání kalibrace nebo definovat, která měření se mají opakovat. Tlačítkem POKRAČOVAT se uvede kalibrace opět do chodu. Tlačítkem UKONČIT se kalibrace ukončí a zároveň se ukončí běh vlákna **TThreadMsrSet**.

Pokud je vozík v automatickém režimu a stojí, může uživatel přepnout režim na ruční (na řídícím pultu), čímž vozík přejde do stavu MANUAL STOP. V tu chvíli je možné vozíkem pohybovat pomocí řídícího pultu. Po návratu do automatického režimu se vozík vrátí do stavu, ve kterém byl před přepnutím do ručního řízení.

Poslední stav REMOTE CONTROL vyjadřuje řízení vozíku z velína.

V tabulce 4.4 je shrnuto povolení nebo zablokování určitých vstupů od uživatele podle toho, v kterém stavu se vozík nachází. Vstupy od uživatele rozumíme zadávání textů a čísel do tabulek a editačních polí, přepínání přepínačů a stisk tlačítek. Znak + v tabulce znamená, že daný prvek je povolen, znak – znamená jeho zablokování, a znak **s** značí zablokovaný a zároveň ztmavený prvek pro větší grafické odlišení automatického a ručního režimu.

Při každé změně stavu se okamžitě aktualizuje povolení a blokování prvků.

	MAN. STOP	MAN. DRIVE	AUTO STOP	AUTO DRIVE	AUTO SET STOP	AUTO SET WAIT	AUTO SET DRIVE	AUTO SET INTER	REM. CONT.
Zadání parametrů vrtulí + databáze, teplota, kalibrující	S	S	+	-	+	-	-	+	-
Výsledky měření	S	S	-	-	-	-	-	-	-
Zadání směru a rychlosti	S	S	+	-	-	-	-	-	-
Přepínače otočí	+	-	+	-	+	-	-	+	-
Přepínač velín	-	-	+	-	+	-	-	-	+
Přepínač motorů	+	-	-	-	-	-	-	-	-
Přepínač priority motorů, aktivace vrtulí, zadání kalibrace	-	-	+	-	+	-	-	+	-
Přepínač Jedno měření / Kalibrace	-	-	+	-	+	-	-	-	-
Tlačítko Přerušit	-	-	-	-	-	+	+	-	-
Tlačítka Opakovat, Pokračovat a Ukončit	-	-	-	-	-	-	-	+	-
Tlačítko Start	-	-	+	-	+	-	-	-	-
Tlačítko Stop	-	-	-	+	-	-	+	-	-

(tab. 4.4 – povolení a blokování vstupů od uživatele)

4.4.3.2 Přepínání mezi stavy průběhu měření

Další sadou stavů jsou stavy popisující jednu jízdu vozíku. Průběh jízdy můžeme rozdělit do pěti stavů: PARKOVÁNÍ, PŘIPRAVENO, ROZJEZD, MĚŘENÍ a BRŽDĚNÍ. Stavový diagram ukazuje obr. 4.10.

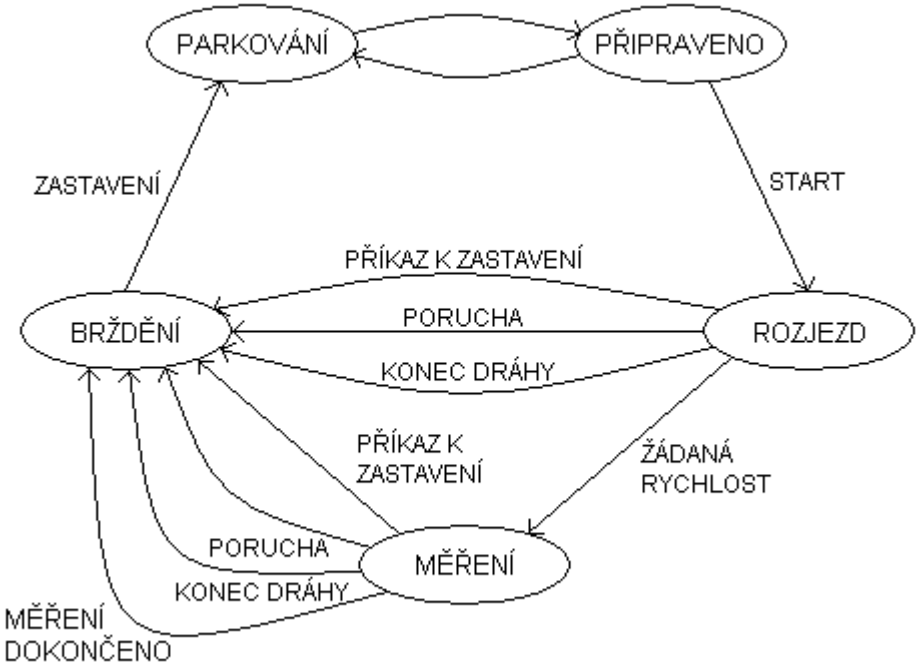
Přechody mezi stavy PARKOVÁNÍ a PŘIPRAVENO je zajištěn pomocí jednoho ze se sekundárních vláken **TThreadMsrSet**, **TThreadMsrOne** nebo **TThreadMsrNone**, která budou vysvětlena dále. Přechody mezi ostatními stavy zajišťuje časovač **timerTrolleyStates** a jsou tedy závislé na signálech z desky LC.

Ze stavu PŘIPRAVENO do stavu ROZJEZD se přechází deaktivací signálů PARKING a STOP JÍZDY, tzn. že vozík je odbržděn a jede. Po dosažení žádané rychlosti (signál VŽ) je dosaženo stavu MĚŘENÍ, kdy probíhá měření vrtulí. Při dokončení měření je deskou MK vydán signál STOP-AV, která následně aktivuje signál STOP JÍZDY, čímž vozík přejde do stavu BRŽDĚNÍ. V okamžiku zabrždění je aktivací signálu PARKING dosaženo stavu PARKOVÁNÍ.

Tato smyčka může být ve stavu ROZJEZD nebo MĚŘENÍ přerušena a předčasně tak může být dosaženo stavu BRŽDĚNÍ. To může nastat ve třech případech. Za prvé uživatel má možnost jízdu kdykoliv přerušit tlačítkem STOP na řídícím pultu nebo v programu. Dále může být předčasně brzděno z důvodu poruchy (signál PORUCHA). A poslední možností je

dosažení konce dráhy před dokončením měření. Brždění na konci dráhy je náplní následujícího odstavce 4.4.3.3.

Po přechodu do stavu PARKOVÁNÍ se kontroluje, zde měření proběhlo v pořádku.

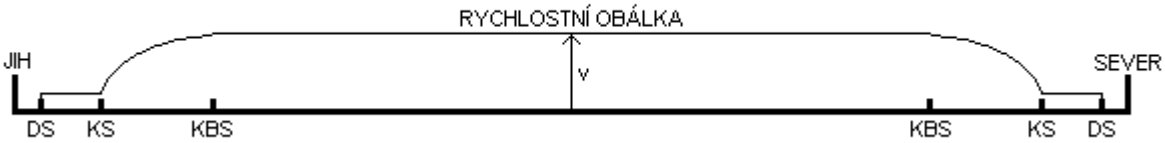


(obr. 4.10 – stavový diagram znázorňující přechody mezi stavy kalibrační jízdy vozíku)

V případě, že jsou všechny vrtule deaktivovány, probíhá tzv. automatická jízda bez měření vrtulí. Hodí se pro dojezd vozíku na konec dráhy nebo pro servisní účely. Vozík automaticky zastaví na konci dráhy anebo před koncem, pokud uživatel stiskne tlačítko STOP.

4.4.3.3 Brždění na konci dráhy

Vozík jezdí až rychlostí 10 m/s (36 km/h) na poměrně krátké dráze. To vyžaduje vysoce bezpečný automatický brzdný systém působící na koncích dráhy. Schéma dráhy je uvedeno na obr. 4.11.



(obr. 4.11 – umístění spínačů v kanálu)

Na každé straně kanálu je jeden snímač dorazu (DS), za kterým je úplně blokována jízda vozíku. Mezi těmito spínači je rychlost kontrolována rychlostní obálkou brzdného systému Sistonik podle obr. 4.11. Jakmile rychlost vozíku v kterémkoliv místě dráhy překročí vymezenou obálku, dojde k okamžitému zastavení vozíku.

Dráha použitelná pro kalibraci je vymezena dvěma koncovými spínači (KS). Mezi body DS a KS na každé straně kanálu je povolená rychlost vozíku max. 0,5 m/s.

Snímače hranice kontroly brždění (KBS) určují místa, od kterých se snižuje rychlostní obálka směrem ke konci dráhy. Snížení obálky se uplatňuje pouze při pohybu vozíku směrem k bližšímu konci dráhy. Mezi snímači KBS má obálka lineární charakter.

Brzdný systém Sistonik však neslouží pro provozní, nýbrž pro havarijní brždění. Pokud tedy rychlost překročí obálku a vozík je havarijně zastaven, je ohlášena porucha a obsluha musí brzdný systém odblokovat. Provozní brždění proto zařizuje automaticky počítač. Ten zajišťuje, aby rychlost vozíku v automatickém režimu nikdy nepřesáhla rychlostní obálku. Tím je vlastně zajištěna dvoustupňová kontrola rychlosti vozíku.

Počítač má v každém okamžiku jízdy k dispozici aktuální hodnotu polohy vozíku na dráze s_{akt} a aktuální hodnotu rychlosti v_{akt}, které získává z desky MK (kap. 4.4.6.3). Pokud je vozík ve stavu ROZJEZD nebo MĚŘENÍ, provádí se v každém cyklu časovače **timerTrolleyStates** kontrola vzdálenosti od konce dráhy, ke kterému se vozík přibližuje. K zastavení vozíku dojde v těchto dvou případech:

a) vozík se pohybuje směrem na sever a neplatí podmínka

$$s_{akt} \leq l_{dr} - 3v_{akt} ,$$

b) vozík se pohybuje směrem na jih a neplatí podmínka

$$s_{akt} \geq 3v_{akt} ,$$

kde l_{dr} je celková délka dráhy (mezi spínači KS). Nulová hodnota s_{akt} je u jižního konce dráhy. Konstanta 3 byla zjištěna pokusně tak, aby při žádné rychlosti nedošlo k protnutí obálky.

Pokud tedy vozík při procesu kalibrace dorazí na konec dráhy, aniž by bylo aktuální měření dokončeno, dojde k provoznímu zastavení bez použití brzdného systému. Poté se pokračuje v opačném směru a měření se opakuje.

Pozn.: hodnoty s_{akt} a v_{akt} slouží také pro vizualizaci jízdy vozíku.

4.4.3.4 Databáze a protokoly o měření

V průběhu zadávání parametrů do tabulek může uživatel použít tří databází. První z nich, **TOwnerList** (zděděná od **TRowList**) obsahuje seznam majitelů vrtulí s jejich adresami. Další dvě, **lbProducerList** a **lbFixingList**, obsahují seznam výrobců vrtulí a

seznam typů upevnění. Všechny databáze je možné libovolně rozšiřovat. Ukládají se na disk do souborů typu *.lst*.

Celkové kalibrační protokoly se ukládají do instancí třídy **TProtocol** (zděděná od **TRowList**). „Nultý řádek“ (viz kap. 4.4.2) obsahuje parametry vrtule a informace o kalibraci (datum, kalibrující osoba, …). Každý řádek tabulky pak představuje jedno měření, tzn. nastavenou rychlost jízdy, počet impulsů vrtule, ujetou dráhu a dobu jízdy. Protokoly se ukládají do souborů *.msr*.

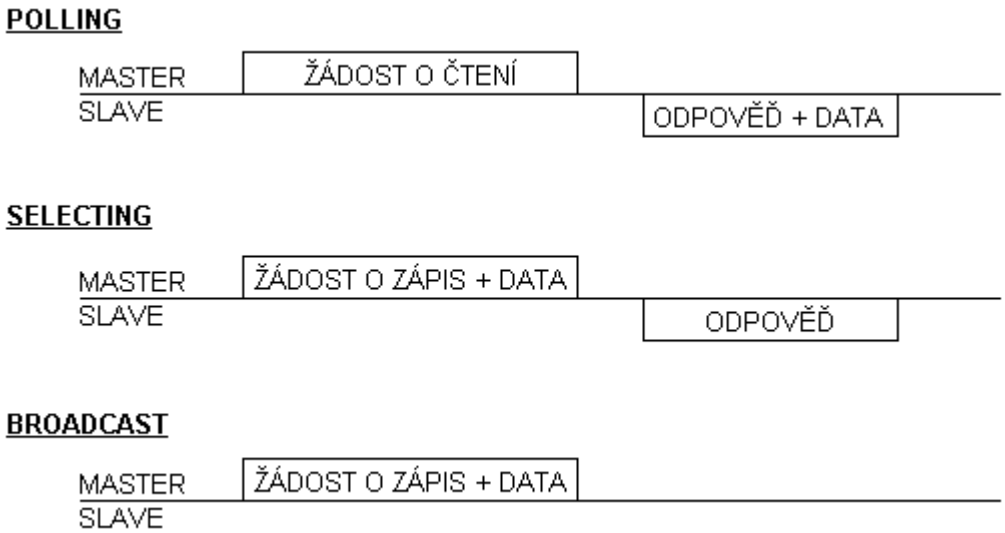
Soubory s protokoly jsou poté zpracovány programem pro tisk protokolů, vyvinutým na jiném pracovišti.

4.4.4 Komunikace s frekvenčními měniči

4.4.4.1 Komunikační protokol

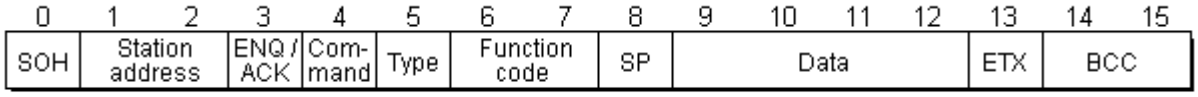
Tato kapitola popisuje protokol komunikace mezi palubním počítačem a měniči frekvence po sběrnici RS-485 (popis v kap. 4.2).

Existují tři metody posílání paketů (zpráv), a to žádost nadřízené jednotky o čtení z jedné z podřízených jednotek (*polling*), žádost o zápis do jedné z podřízených jednotek (*selecting*) a žádost o zápis do všech podřízených jednotek najednou (*broadcast*). Graficky je posílání znázorněno v obr. 4.12:



(obr. 4.12 – metody přenosu paketů)

Struktura paketu je stejná pro žádosti i odpovědi. Je možné použít standardní rámec, ve kterém můžeme použít všechny komunikační funkce, nebo zkrácený rámec, ve kterém je počet funkcí omezen, zato však umožňuje vysokorychlostní komunikaci. Struktura plného standardního rámce je složena ze 16 bytů a je ukázána na obr. 4.13.



(obr. 4.13 – struktura standardního rámce)

Význam jednotlivých bytů je uveden v tabulce 4.5. Pokud je hodnotou znak, pak se do rámce umísťuje jeho ASCII hodnota. Pokud je hodnotou číslo (v hexadecimálním tvaru), rozdělí se na cifry a použijí se ASCII hodnoty jednotlivých cifer (kromě SOH, ENQ, SP a ETX – do těchto bytů se píše přímo daná číselná hodnota).

Byte	Hodnota	Popis
SOH	01_H	Start Of Header – počáteční byte
Station address	0 – 39, 99	Adresa stanice, které je byte určen (99 = broadcast)
ENQ / ACK	05_H / 06_H (15_H)	ENQ = Žádost, ACK = pozitivní odpověď (NAK neg.)
Command	R,W,A,E	R = čtení (polling), W,A = zápis (selecting), E = reset
Type	F,E,C,P,H,A,o,S,M	Typ funkce
Function Code	0 – 49	Kód funkce
SP	20_H	Space – nepoužitý byte
Data	0 – $FFFF_H$	Data
ETX	03_H	End Of Text – poslední byte před kontrolním součtem
BCC	0 – FF_H	Kontrolní součet

(tab. 4.5 – význam bytů v rámci)

Počítač (MASTER) vysílá pakety s třetím bytem ENQ, měniče (SLAVEs) posílají zpět paket s třetím bytem buď ACK (Acknowledge – potvrzení) nebo NAK (zamítnutí). Pokud měniče přijmou paket necelý nebo s neplatným kontrolním součtem, neposílají zpět nic. Ke kolizím na sběrnici zde nedochází, protože počítač je vždy první, kdo „oslovuje“ daný měnič, čeká na jeho odpověď a teprve potom vysílá dál.

4.4.4.2 Implementace protokolu v programu

Počítač vysílá a přijímá pakety na sériovém portu COM (linka RS-232). Pro sériový port, jakožto hardwarovou periferii, je nutné naprogramovat obsluhu. Platforma WINDOWS NT, tedy operační systémy Windows NT 4.x, 2000 a XP, využívá tzv. protected (chráněný) mód procesoru, který již nepodporuje přímý zápis do registrů periferií (jako tomu bylo u MS-DOSu nebo platformy WINDOWS 9.X), ale je nutné k těmto periferiím přistupovat přes jejich ovladače. Je tak zabráněno tomu, aby neautorizovaným přístupem několika paralelně běžících procesů na stejnou periferii nedošlo ke vzájemné kolizi těchto procesů nebo dokonce celého operačního systému. Pro tvorbu těchto ovladačů slouží aplikační rozhraní *Windows application interface* (API), obsahující tisíce funkcí a služeb, mezi nimi také funkce pro obsluhu hardwarových periferií. Tím, že API funguje jako univerzální rozhraní, získáváme nezávislost na tom, jak je daný hardwarový blok fyzicky realizován. API funkce se proto v každé verzi Windows přizpůsobují vývoji periferií a jejich poslední verze Windows XP by měla mít sadu API funkcí nejúplnější.

Přístupů, jak napsat ovladač pro sériový port, je mnoho. V našem případě byl použit postup podle [1]. Byly převzaty dvě třídy, **TThreadComm** a **TCommEventThread**, které tuto obsluhu zajišťují.

TThreadComm je objekt obsluhy sériového portu a zajišťuje otevření portu, nastavení pracovních parametrů portu a operace čtení a zápisu. Po otevření je tento port obsazen naší aplikací a ostatní paralelně běžící aplikace ho nemohou sdílet. Je na něj hleděno jako na soubor a operace s ním se od operací se souborem v podstatě neliší. Je důležité ho otevřít v tzv. režimu *overlapped* (překrytý), aby bylo možné v jednom okamžiku číst a zároveň zapisovat z/do portu (tzv. *full-duplex mód* = komunikace mezi zařízeními současně v obou směrech).[13] Znamená to, že operace čtení a zápisu mohou běžet jako dva paralelní procesy. Nastavením pracovních parametrů se myslí nastavení přenosové rychlosti, řídících znaků, parity, timeoutů při operacích čtení a zápisu, apod.

Třída **TCommEventThread** je potomkem třídy **TThread** z knihovny VCL a řeší obsluhu událostí, které generuje hardware sériového portu, jako je např. událost přijetí znaku. Podrobný popis obou tříd je uveden v [1].

Pro kompletní zajištění komunikace mezi počítačem a měniči byly dále napsány následující třídy: **TPacket**, **TThreadSend**, **TSerialLink** a **TInverterLink**.

TPacket se stará o složení, odeslání, parsování[14] a kontrolu přijatého paketu.

TSerialLink usnadňuje komunikaci po sériové lince. Zapouzdřuje objekt sériového portu **TThreadComm**, objekt fronty paketů **T_Queue<TPacket>** a vlákno pro odesílání zpráv **TThreadSend**. V konstruktoru **TSerialLink** se sériový port otevře, vytvoří se fronta paketů a rozjede se vlákno pro jejich odesílání. Třída **TSerialLink** je však třídou abstraktní, protože obsahuje abstraktní metodu *MyOnRxChar*, která je použita jako parametr při vzniku objektu sériového portu a je volána vždy, pokud je přijat znak po sériové lince. Tato metoda je vždy unikátní pro konkrétní aplikaci. Je tedy nutné od třídy **TSerialLink** podědit novou třídu a v té pak definovat metodu *MyOnRxChar*. Takto byla vytvořena třída **TInverterLink** pro komunikaci s několika frekvenčními měniči.

Na frekvenčních měničích lze sériovou linkou přenastavit v podstatě všechny parametry, kterých je přibližně 200. Tyto parametry lze také nastavovat na panelu umístněném na skříni měniče, aniž by byla sériová linka zapojená. V našem případě se bude nastavovat parametrů jen několik. Parametr je vždy určen písmenem (Type – viz obr. 4.13 struktura rámce) a číslem (Function Code).

TInverterLink obsahuje funkce pro vytváření různých paketů - zpráv pro měniče (např. nastavení frekvence[15], nastavení hystereze[16] apod.) a definuje funkci *MyOnRxChar*, která přijímá odpovědi od měničů a reaguje na ně. Všechny zprávy, které jsou připraveny k odeslání, jsou uloženy do fronty **T_Queue<TPacket>**. Vlákno **TThreadSend** pravidelně

[13] V našem případě komunikace s měniči by však stačil i tzv. *half-duplex* mód, při kterém je v jednom okamžiku možné buď číst a nebo přijímat (dochází k časovému sdílení kanálu - směr komunikace se přepíná). Pro doplnění uveďme že nejjednodušší variantou spojení je tzv. *simplex*, kdy je přenosová cesta realizována jen v jednom směru a zařízení je buď vysílačem nebo přijímačem po celou dobu přenosu.
[14] Parsování je rozložení paketu na jednotlivé elementy.
[15] Type = S, Function Code = 01
[16] Type = E, Function Code = 30

kontroluje, zda jsou ve frontě nějaké zprávy a pokud ano, postupně je odesílá. Před odesláním nové zprávy však vždy čeká po časový interval ($T_{REPLY_TIMEOUT}$ = 3s) na přijetí odpovědi od měničů. Pokud odpověď nepřijde nebo přijde s chybami (tzn. nesouhlasí kontrolní součet), pokusí se o poslání zprávy ještě dvakrát a až pak, v případě neúspěchu, ohlásí chybu komunikace. Pokud přijde negativní odpověď (bit č.3 = NAK), tzn. měnič odmítl provést příkaz, je chyba ohlášena ihned.

4.4.5 Komunikace s deskou MK

Jednotka měření a kalibrace MK je podrobně popsána v kapitole 3.3 a její spojení s počítačem v kapitole 4.1.2.2. Komunikace s ní probíhá přístupem do jejích registrů přes sběrnici (porty 3 a 4). K tomu slouží třída **TRegister**, která pomocí své property *Value* zajišťuje vstupně/výstupní operace s registry desky MK.

Operace zápisu probíhá v následujících krocích:

1. Nastavení sběrnice jako výstupní z počítače (tedy do režimu zápisu)
2. RD = 1, WR = 1 (tyto signály jsou aktivní v nule)
3. Připravení dat na sběrnici a nastavení adresy
4. WR = 0, sestupnou hranou WR dojde k zápisu do registru
5. Nastavení sběrnice zpět do režimu čtení

Před operací čtení je sběrnice již nastavena do režimu čtení, protože se do něj vždy automaticky vrací po operaci zápisu. Čtení pak probíhá následovně:

1. RD = 1, WR = 1
2. Nastavení adresy
3. RD = 0, sestupnou hranou RD dojde k přesunu dat z daného registru na sběrnici
4. Čtení dat ze sběrnice

Těla funkcí pro operace s registry ve třídě **TRegister** musí být ochráněna kritickou sekcí, aby byl zajištěn přístup na sběrnici v jednom okamžiku vždy jen z jednoho vlákna programu.

4.4.6 Sekundární vlákna

4.4.6.1 Synchronizace vláken

Ve vícevláknové aplikaci je nutné zařídit synchronizaci vláken. Jsou k tomu používány předdefinované objekty – např. objekt kritické sekce (Critical Section) nebo objekt události (Event) [2].

Objekt kritické sekce je použit tam, kde je nutné zařídit, aby s určitými daty pracovalo v jeden okamžik jen jedno vlákno. Úseky kódů ve vláknech, kde se s danými daty pracuje,

jsou označeny jako kritická sekce a v této sekci pak může být jen jedno vlákno. Ostatní jsou uspána a čekají, až bude kritická sekce uvolněna.

Objekt události je použit tam, kdy je nutné mezi vlákny předávat informace o výskytu nějakých událostí. Např. jedno vlákno přijímá vstupy z periferie a jiné vlákno čeká na aktivaci nějakého signálu z této periferie. Toto vlákno je dočasně uspáno a čeká na výskyt události, který bude aktivován z prvního vlákna v okamžiku, kdy bude zjištěna změna daného signálu z periferie.

Při programování za použití více vláken pod Windows bylo zjištěno, že každé vlákno musí mít ve svém cyklu alespoň minimální čas, kdy je suspendováno (neboli kdy spí[17]). Pokud to u některého vlákna neplatilo, plánovač, starající se o přepínání mezi vlákny, byl nadměrně zatížen a výkon procesoru stoupl na 100%. To po nějaké době vedlo k „zamrznutí“ programu a zabrždění vozíku.

4.4.6.2 Kontrolní signál watchdog

V průběhu provozu vozíku se může stát, že nastane doposud neobjevená chyba v programu, která způsobí jeho „zamrznutí“ nebo dokonce havárii celého operačního systému. Chyba může nastat nejen softwarového charakteru, ale i hardwarového, kdy se při jízdě vlivem otřesů např. přeruší nějaký kontakt uvnitř počítače[18]. Všechny tyto případy vedou ke ztrátě kontroly nad vozíkem a musí dojít k jeho okamžitému zastavení.

Z toho důvodu sekundární vlákno **TThreadWatchDog** vysílá impulsní signál PCWD (watchdog) do LC. Pokud přestane program fungovat, přestane fungovat i watchdog, deska LC to vyhodnotí jako ztrátu signálu PC RDY (a tedy i VOZÍK RDY) a dojde k zastavení vozíku.

Z toho plyne, že pokud nefunguje program, vozík se nemůže rozjet. Při servisu vozíku je však nutné, aby existovala možnost s vozíkem hýbat i při vypnutém palubním počítači. Deska LC proto nabízí možnost překlenutí watchdogu pomocí jumperu.

4.4.6.3 Monitorování registrů MK a vizualizace procesu

Vedle watchdogu a vláken pro komunikaci po sériové lince je dalším sekundárním vláknem, které běží po celou dobu běhu programu, také **TThreadRegRead**. Toto vlákno periodicky vyčítá obsahy registrů na desce MK. Běží v tomto cyklu:

1. Aktivace signálu STROBE, tzn. nastavení RD = 0 a WR = 0, čímž dojde k uchování aktuálních hodnot registrů v desce (viz kap. 3.3.2).

[17] Např. API funkce Sleep

[18] K tomu skutečně došlo při jízdních zkouškách. Paměť DIMM, která patrně nebyla při instalaci na základní desku počítače správně zasazena do slotu, se při otřesech za jízdy postupně více uvolňovala a způsobovala tak zamrznutí nebo reset počítače. Okolnost, že k resetu počítače docházelo pouze při otřesech, dopomohla k nalezení závady.

2. Načtení registrů *Status*, *ActSpeed*, *ActPosIRC*, *ActPosRul*, *AActPulse*, *BActPulse*, *CActPulse.*
3. Grafické znázornění točení vrtulí podle obsahu registrů s aktuálním počtem pulsů od vrtulí *(AActPulse, BActPulse, CActPulse)*.
4. Grafické znázornění polohy vozíku podle obsahu registrů s počtem pulsů od IRC *(ActPosIRC)* a od pravítka *(ActPosRul)* pomocí komponenty VCL **TPaintBox**.
5. Grafické znázornění rychlosti vozíku podle obsahu registru *ActSpeed* pomocí objektu **TTachometer**.
6. Kontrola *Status registru* – v případě, že se naplní bity *AEQ*, *BEQ* nebo *CEQ*, které udávají konec měření jednotlivých vrtulí, pošle se zpráva[19] hlavnímu oknu programu, ve kterém se výsledky vyhodnotí a uloží do protokolu

Pro výpočet aktuální polohy je použito následujících vzorců:

$$s_{akt} = K_{ruler} n_{ruler} + K_{IRC1} n_{IRC} \quad \text{pro první převodový stupeň a}$$
$$s_{akt} = K_{ruler} n_{ruler} + K_{IRC2} n_{IRC} \quad \text{pro druhý převodový stupeň,}$$

kde s_{akt} je aktuální poloha [m], n_{ruler} je stav registru *ActPosRul*, n_{IRC} je stav registru *ActPosIRC* a K_{ruler}, K_{IRC1} a K_{IRC2} jsou konstanty. Při výpočtu těchto konstant se vycházelo z toho, že kolo vozíku má průměr 0,35 m, na jednu otočku motoru připadá 1024 pulsů IRC, vzdálenost mezi dírami pravítka je 0,1 m a platí následující vztahy:

$$n_k = \frac{3}{49} n_m \quad \text{pro první převodový stupeň a}$$
$$n_k = \frac{3}{7} n_m \quad \text{pro druhý převodový stupeň,}$$

kde n_k jsou otáčky kola a n_m otáčky motoru. Tyto vztahy jsou dány převodovkami - každý pohon obsahuje jednu převodovku s konstantním převodem 3:1 a jednu planetovou řaditelnou s převody 1:7 a 1:49. Výsledné konstanty jsou:

$$K_{ruler} = 0{,}1$$
$$K_{IRC1} = 65{,}74203376.10^{-6}$$
$$K_{IRC2} = 0{,}46019423636.10^{-3}$$

Do registru *ActSpeed* deska MK ukládá počet impulsů IRC za konstantní časový okamžik $t_M = 0{,}1310715$ s (je to doba trvání 2^{22} pulsů hodin – $f_{CLK} = 32$ MHz). Z tohoto údaje a za pomoci konstant K_{IRC1} a K_{IRC2} můžeme určit konstanty K_{speed1} a K_{speed2}. Aktuální rychlost se potom počítá podle vztahů:

$$v_{akt} = K_{speed1} n_M \quad \text{pro první převodový stupeň a}$$
$$v_{akt} = K_{speed2} n_M \quad \text{pro druhý převodový stupeň,}$$

kde n_M je aktuální obsah registru *ActSpeed*. Rychlostní konstanty jsou:

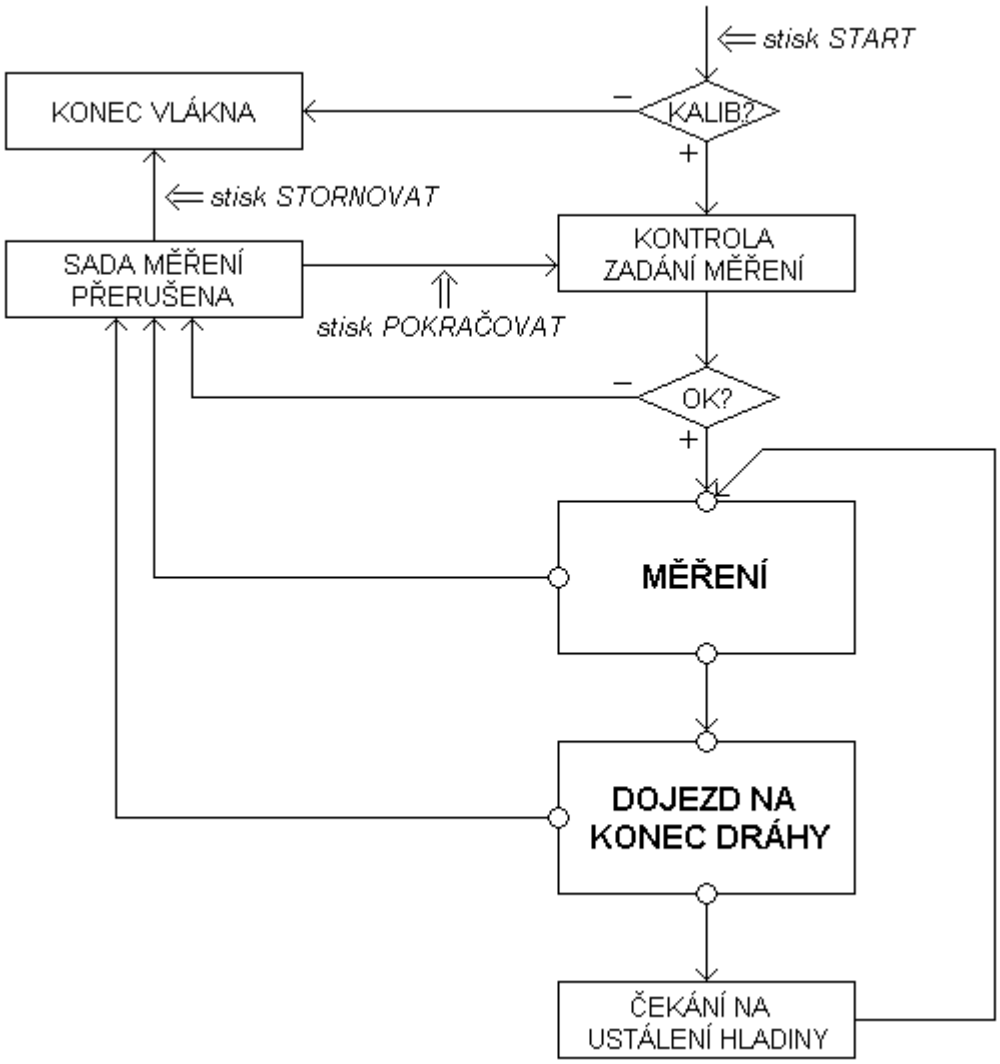
$$K_{speed1} = 5{,}01573826194.10^{-04}$$
$$K_{speed2} = 3{,}51101678371.10^{-03}$$

[19] Pomocí API funkce PostMessage

4.4.6.4 Řízení kalibrace

Průběh kompletní kalibrace je řízen sekundárním vláknem **TThreadMsrSet**. Vývojový diagram tohoto vlákna je znázorněn na obr. 4.14.

Je potřeba nejprve uvést, že pro všechny následující vývojové diagramy platí: pokud na jednoduchou přechodovou šipku mezi jednotlivými akcemi ukazuje dvojitá šipka, je tento přechod podmíněn splněním toho, co je u dvojité šipky uvedeno. Buď je to stisk určitého tlačítka nebo výskyt události[20], která se stane v jiném vlákně. Pokud dvojitá šipka ukazuje na kosočtvercový blok s podmínkou, pak to, co je u dvojité šipky uvedeno, hraje roli v hodnocení platnosti podmínky.



(obr. 4.14 – vývojový diagram vlákna TThreadMsrSet)

Vlákno **TThreadMsrSet** se spouští stiskem tlačítka START, když ovšem bylo předtím přepínačem JEDNO MĚŘENÍ / KALIBRACE přepnuto na kalibraci. Pokud vozík není zkalibrovaný (resp. měření dráhy není zkalibrováno), vlákno se okamžitě ukončí s oznámením obsluze, že je nutné vozík zkalibrovat (tzn. projet jižním koncovým spínačem). V opačném případě vlákno pokračuje kontrolou zadání parametrů měření. Pokud některý

[20] Pokud před jméno události vložíme text „hEvent“, tedy např. z TrolleyReady vytvoříme hEventTrolleyReady, dostaneme přímo jméno proměnné ve zdrojovém kódu (umístěném na přiloženém CD-ROM).

parametr chybí nebo je nepřípustný, je to obsluze ohlášeno chybovým oknem s popisem, co je potřeba opravit, a vlákno je přerušeno, tzn. čeká se na začátku cyklu vlákna na stisk tlačítka POKRAČOVAT.[21]

Pro aktuální parametry měření jsou použity třídy **TMeasure** a **TPropeller**. Další třída **TMsrList** (zděděná **od T_List<int>**) obsahuje dynamický seznam čísel měření, která ještě nebyla provedena.

Jakmile jsou všechny parametry v pořádku, nastává vlastní měření. V obr. 4.14 je tato část vyjádřená blokem MĚŘENÍ, jehož vnitřní struktura bude popsána dále. Měření může být z různých důvodů kdykoliv přerušeno. Poté následuje dojezd na konec dráhy (tento proces bude také popsán dále) a čekání na ustálení hladiny. Po uběhnutí časového intervalu čekání se cyklus opakuje a probíhá další měření.

Na obr. 4.15 je znázorněn vývojový diagram bloku MĚŘENÍ. Nejprve dojde k zařazení převodů podle žádané rychlosti. Rozhoduje se podle následujících dvou vztahů:

- pokud platí $v \leq v_{m\arg 2}$, kde v_{marg2} = 2,5 m/s, pojede jen jeden pohon (ten, který má nastavenou prioritu přepínačem PRIORITA MOTORŮ), druhý bude zařazen do neutrálu; v opačném případě pojedou oba
- pokud $v \leq v_{m\arg 1}$, kde v_{marg1} = 1 m/s, bude zařazen první převodový stupeň, jinak druhý převodový stupeň

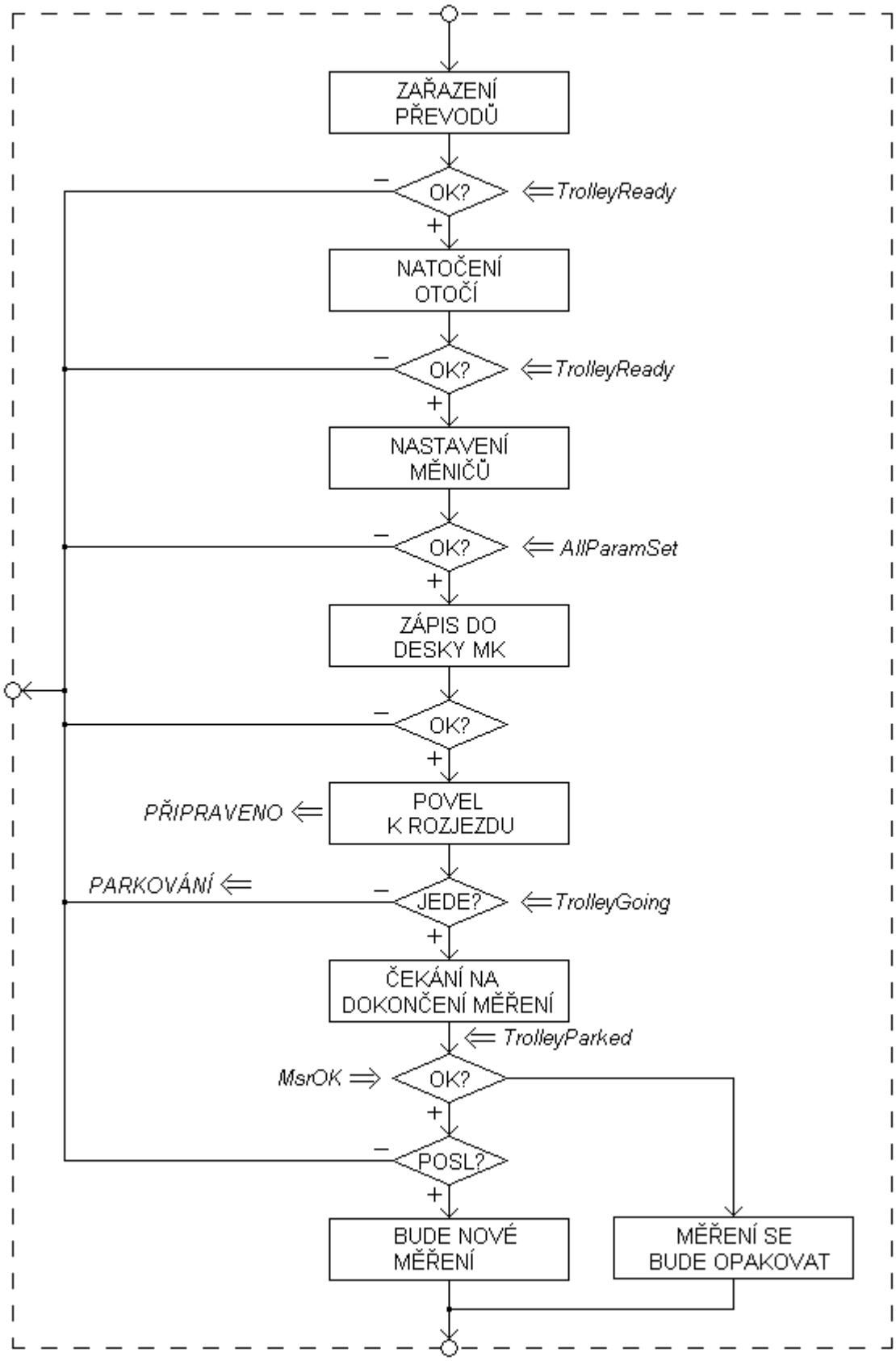
Převodové stupně se řadí signály P1-A, P2-A, zařazení do neutrálu proběhne deaktivací signálů POHON1 ON nebo POHON2 ON.

Poté se po daný časový interval ($T_{WAIT_FOR_READY}$ = 6s) čeká na aktivaci signálu VOZÍK RDY (neboli RDY PŘEV – převodovky připraveny – viz kap. 3.2.6). Tato skutečnost je oznámena z hlavního vlákna (z časovače **timerTrolleyStates**) událostí *TrolleyReady*. Pokud událost nenastane do vypršení časového intervalu, je sada měření přerušena s ohlášením závady.

Toto čekání mezi úkony bylo přidáno po jízdních zkouškách, kdy bylo zjištěno, že pokud se provádělo více úkonů najednou (řazení převodovek, točení otočí apod.), docházelo někdy k nedostatku stlačeného vzduchu a vypadávání systému Sistonik. Tím byla kalibrace přerušena.

Dále proběhne zjištění směru jízdy podle aktuální polohy na dráze a podle toho nastavení signálu J-A nebo S-A. Tím dojde k přetočení otočí a opět se čeká na *TrolleyReady* (resp. VOZÍK RDY a RDY OTOČ).

[21] Tato skutečnost je vláknu oznámena událostí *MsrSetAllowed* z hlavního vlákna, které registruje stisk tlačítek.



(obr. 4.15 – vývojový diagram bloku MĚŘENÍ)

Následuje nastavení parametrů měničů, předně nastavení žádané frekvence (resp. rychlosti). Na měničích je nutné nastavit frekvenci podle následujících vzorců:

$$f[\mathrm{Hz}] = K_{freq1}.v[\mathrm{m/s}] \text{ pro první převodový stupeň a}$$

$$f[\mathrm{Hz}] = K_{freq2}.v[\mathrm{m/s}] \text{ pro druhý převodový stupeň,}$$

kde v je požadovaná rychlost, $K_{freq1,2}$ jsou konstanty a f je spočtená frekvence pro měniče. Při výpočtu konstant $K_{freq1,2}$ se vycházelo z toho, že motor má při 50 Hz jmenovité otáčky 1460 ot/min, a ze vztahů uvedených v kapitole 4.4.6.3 o vizualizaci. Konstanty vyjdou[22]:

$$K_{freq1} = 30{,}5228657989 \text{ a}$$
$$K_{freq2} = 4{,}36040939984.$$

Kromě toho se do měničů posílá nastavení hysterezí a nastavení PI regulátorů rychlosti. Hodnoty složek PI regulátoru byly pokusně nastaveny při jízdních zkouškách a to různě pro první a druhý převodový stupeň. Podle nastaveného převodu se tedy měniče přenastaví. Následuje nastavení hystereze, tzn. maximální hodnoty, o kterou se může snížit nebo zvýšit skutečná frekvence od žádané, aby byla stále brána jako žádaná. Byly pokusně určeny tři hodnoty hystereze pro tři intervaly frekvencí. Tyto hodnoty lze měnit na servisní kartě nebo přímo v *.INI* souboru.

V okamžiku přijetí potvrzení nastavení frekvence a všech parametrů (výskyt události *AllParamSet*, která nastává v metodě volané událostmi na sériové lince *MyOnRxChar* v objektu **TInverterLink**) se pokračuje v běhu vlákna. Tato událost musí opět nastat do časového limitu, jinak se vlákno přeruší.

Nyní přichází připravení desky MK pro měření (deska je popsána v kapitolách 3.3, 4.1.2.2 a 4.4.5). Nejprve je nastavením bitu *Reset* ve *Status registru* desky MK smazán obsah registrů, do kterých se budou ukládat výsledky měření. Potom jsou do registrů *APreset*, *BPreset* a *CPreset* zapsány požadované hodnoty impulsů pro jednotlivé vrtule, pokud jsou aktivní a budou se měřit. Pokud ne, zapíše se nula. Zpětným přečteným registrů je zkontrolován zápis hodnot a poté je nastaven bit *MsrRequest* ve *Status registru*.

V tuto chvíli je vše připraveno a vozík dostává signál k rozjedu START-A. Zároveň je přepnut stav měření do stavu PŘIPRAVENO (viz. kap. 4.4.3.2). Pokud se vozík rozjede (výskyt události *TrolleyGoing* z hlavního vlákna (časovače **timerTrolleyStates**)), začíná měření.

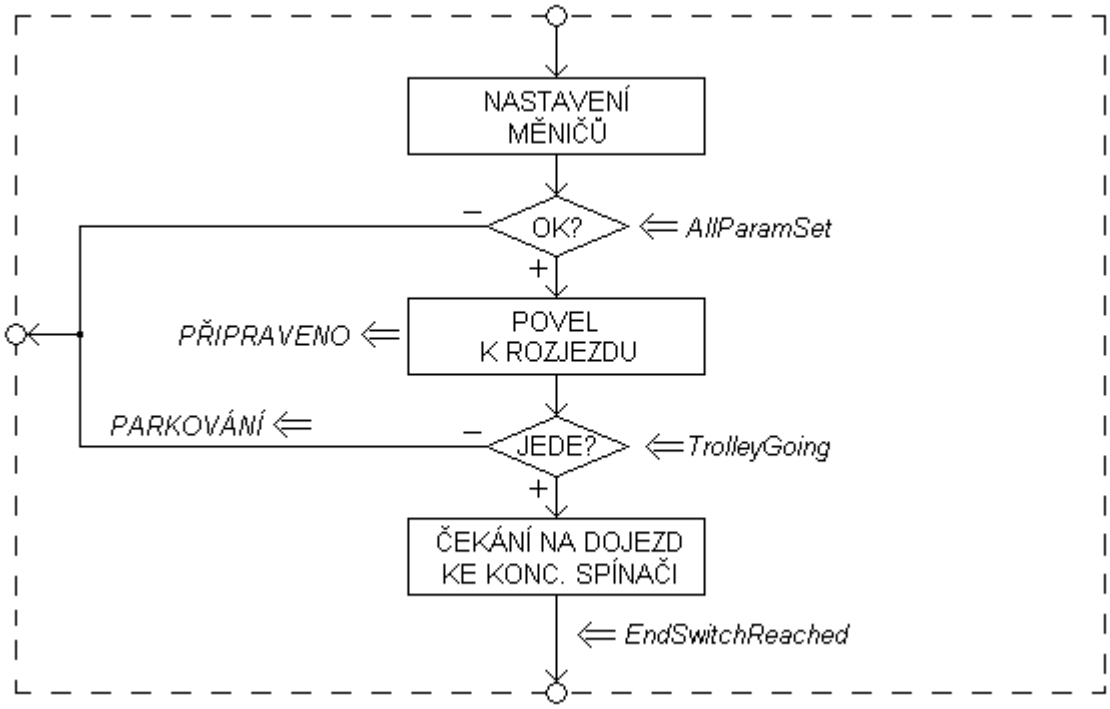
Proces měření je nyní kontrolován hlavním vláknem, které přepíná mezi stavy průběhu kalibrační jízdy (kap. 4.4.3.2). V okamžiku, kdy je některá z vrtulí doměřena, je z vlákna **TThreadRegRead** (kap. 4.4.6.3) odeslána zpráva hlavnímu oknu, ve kterém je tím spuštěna funkce pro přidání měření do protokolu (objekt **TProtocol**). V případě, že jsou všechny vrtule doměřeny, aktivuje hlavní vlákno událost *MsrOK*, která po zastavení vozíku (událost *TrolleyParked*) rozhodne, zda měření proběhlo v pořádku nebo zda bude nutné ho zopakovat. V případě, že měření bylo poslední, je kalibrace přerušena.

[22] Tyto konstanty byly mírně upraveny při jízdních zkouškách na hodnoty K_{freq1} = 29,7508657989 a K_{freq2} = 4,248040939984 tak, aby žádaná rychlost odpovídala rychlosti měřené.

Nyní vozík dojíždí na konec dráhy podle vývojového diagramu na obr. 4.16. Je to vlastně zjednodušení diagramu z obr. 4.15. Směr vozíku se nemění, rychlost je nastavena na 1 m/s a vozík jede do té doby, než je zastaven automatickým brzděním na konci dráhy (kap. 4.4.3.3). Poté se podle diagramu 4.14 čeká po daný časový interval na ustálení hladiny a následně se pokračuje novým měřením – cyklus se vrací do bloku MĚŘENÍ.

V průběhu kalibrace je možné kdykoliv stisknout tlačítko PŘERUŠIT. Tento požadavek je zaregistrován a v nejbližší možný okamžik je kalibrace přerušena (blok SADA MĚŘENÍ PŘERUŠENA v diagramu 4.14). Tzn. pokud se provádí měření, počká se na jeho dokončení a pak se kalibrace přeruší. Podobně při dojezdu na konec dráhy. Stiskem tlačítka STOP dojde k zastavení a přerušení kalibrace okamžitě.

V přerušeném stavu je možné změnit parametry měření, přidat měření nebo opakovat měření tlačítkem OPAKOVAT (změna **TMsrList**).

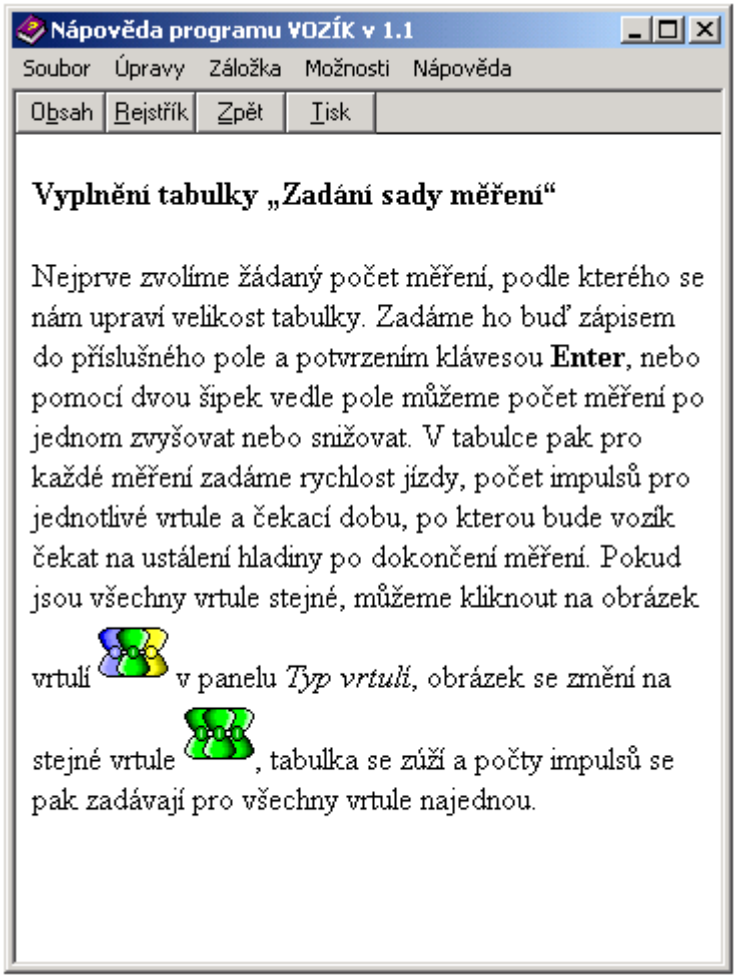


(obr. 4.16 – vývojový diagram bloku DOJEZD NA KONEC DRÁHY)

Program ještě spouští dvě podobná vlákna. Jedno samostatné měření (pokud je přepínačem JEDNO MĚŘENÍ / KALIBRACE přepnuto na jedno měření) je řízeno vláknem **TThreadMsrOne** a jízda bez vrtulí vláknem **TThreadMsrNone**. Jsou to zjednodušení vlákna **TThreadMsrSet** bez opakovací smyčky.

4.4.7 Nápověda

Při stisku klávesy F1 se ukáže kontextová nápověda s popisem ovládání programu. Příklad jedné stránky nápovědy je ukázán na obrázku 4.17. V nápovědě lze vyhledávat i pomocí rejstříku.



(obr. 4.17 – nápověda)

5 Dohlížecí úroveň

Tato kapitola popisuje řízení kalibrace a její monitorování z velína. Nejprve je popsán počítač na velíně (kap 5.1), kapitola 5.2 shrnuje komunikační protokoly a kapitoly 5.3 – 5.5 popisují samotné řízení z velína a komunikace s vozíkem.

5.1 Počítač na velínu

Počítač na velíně obsahuje tyto komponenty:

- Procesor Intel Pentium IV 2,8 GHz
- Základní deska OP 3000N (onboard LAN)
- RAM 256 MB
- HDD 80 GB, FDD 3,5“
- DVD-R/RW Toshiba DL
- Operační systém MS Windows XP Pro, CZ
- LCD monitor 19“ Sony
- Klávesnice + myš

Propojení počítače na velíně s počítačem na vozíku je provedeno prostřednictvím bezdrátové síně LAN. Z velína, který je situován ve druhém patře nad kanálem, je veden kabel ke směrové anténě Buffalo 2,4 GHz, umístěné na jižní straně kanálu na stěně. Další anténa je umístěna na vozíku.

Komunikace po síti LAN probíhá pomocí komunikačních protokolů popsaných v následující kapitole.

5.2 Komunikační protokoly TCP/UDP/IP

Komunikační protokoly Transmission Control Protocol (TCP), User Datagram Protocol (UDP) a Internet Protocol (IP) zajišťují komunikaci po libovolném počtu vzájemně spojených sítí. Jsou vhodné pro sítě LAN (Long Area Network) i WAN (Wide Area Network).

Internet Protocol (IP) odpovídá síťové vrstvě v OSI modelu[23]. Zajišťuje směrování paketů (tzv. routing) po síti od jednoho nódu (= uzlu) k druhému. Cílový nód je určen 4-bytovou IP adresou.

Transmission Control Protocol (TCP) odpovídá transportní vrstvě v OSI modelu a zajišťuje tzv. „spolehlivou komunikaci“ po síti. Mezi dvěma nódy je pomocí tohoto protokolu navázáno spojení a otevřena konverzace, je vytvořen tzv. virtuální spoj. TCP rozděluje zprávy (pakety) na segmenty a předává je nižším vrstvám (IP) k odeslání. Zároveň kontroluje bezchybné doručení segmentů a opětovné složení celé zprávy. V případě, že dojde k chybě, opakuje odeslání segmentu do té doby, než je v pořádku doručen. TCP zajišťuje komunikace typu full-duplex.

User Datagram Protocol (UDP) také odpovídá transportní vrstvě a oproti TCP je to jednodušší, ale méně spolehlivý protokol. Pakety (tzv. datagramy) jsou odesílány podobně jako poštovní dopisy. Po odeslání už se o ně odesílatel nestará a pokud se ztratí, odesílatel o tom není informován. Mezi dvěma nódy není navázána konverzace jako v případě TCP. Na druhou stranu jsou odesílané pakety menší než při použití TCP, protože k nim není přibalováno tolik přídavných informací. UDP se tedy používá tam, kde není nutná vysoká spolehlivost přenosu, zejména tam, kde se o kontrolu doručení paketů starají vyšší vrstvy (aplikace). To je také náš případ.

V naší aplikaci můžeme použít protokol UDP, protože aplikace na velíně i na vozíku se samy starají o kontrolu přenosu pomocí potvrzovacích zpráv.

5.3 Programy ve velínovém počítači

Ve velínovém počítači jsou dva programy. První (VELÍN.EXE) slouží pro vzdálené řízení vozíku, druhý (PROTOKOL.EXE) zpracovává získané výsledky měření pro tisk. PROTOKOL.EXE byl vyvinut spolupracující firmou, proto se jím tato práce nezabývá.

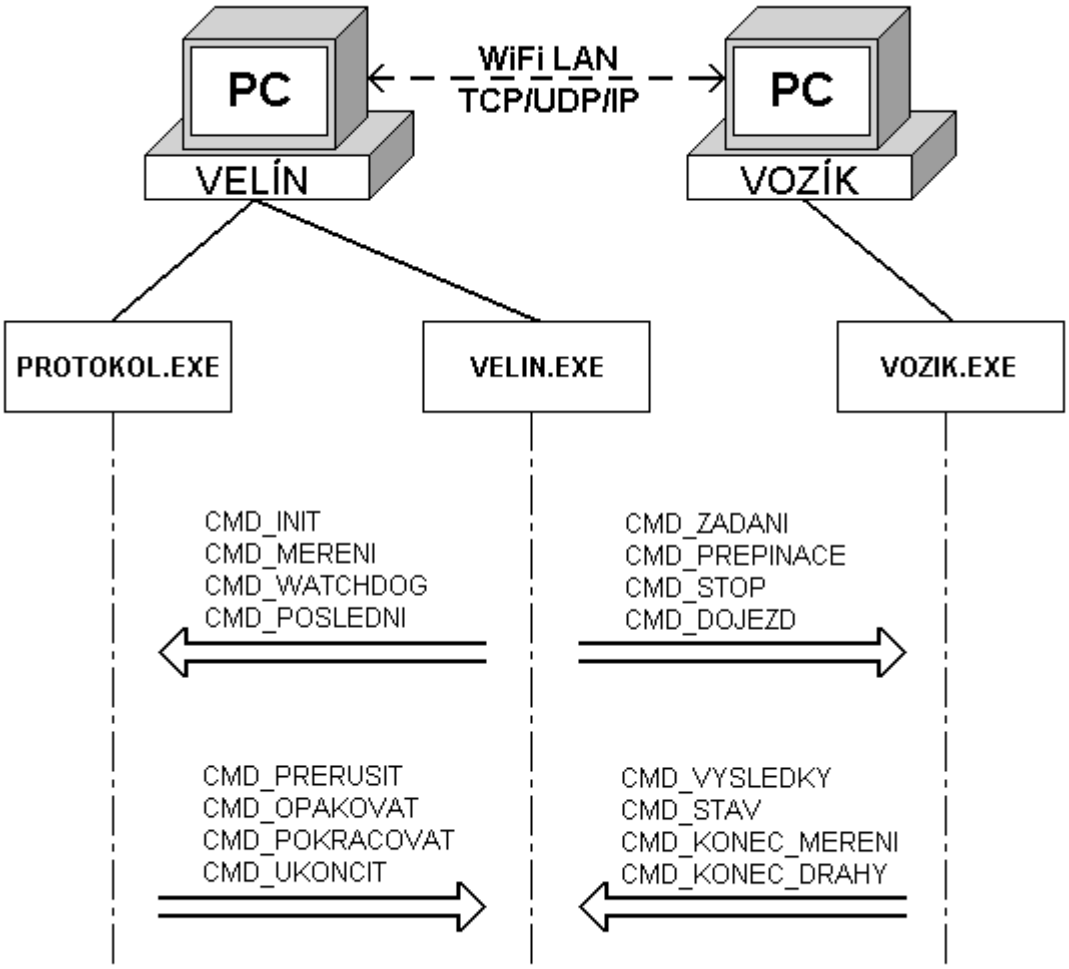
Program řízení vozíku z velína je vizuálně v podstatě identický s programem v palubním počítači na vozíku, ovšem co se týká jeho struktury, je jednodušší. Je rozdílný v následujících bodech:

- Neobsahuje komunikaci s deskou LC, MK ani frekvenčními měniči (tzn. postrádá příslušné třídy)
- Stavy vozíku (svit signálek, aktuální poloha, rychlost, pulsy od vrtulí) jsou získávány pravidelným posíláním zpráv z vozíku do velínu
- Nastavení přepínačů je při změně posíláno z velínu do vozíku
- Jednotlivá měření jsou řízena nadále programem z vozíku, program na velíně pouze pošle zadání a čeká na výsledky (tzn. třída **TThreadMsrSet** je na velínu jednodušší – viz příští kapitola)

[23] OSI (Open System Interconnection) model rozděluje komunikační problémy do sedmi vrstev: 1. fyzické (přenosové médium), 2. spojovací (přístup na sběrnici), 3. síťové (směrování paketů), 4. transportní (rozklad zpráv do paketů), 5. relační (nástroje pro dialog), 6. prezentační (komprese, utajení), 7. aplikační (programy).

5.4 Komunikace velín - vozík

Jak bylo uvedeno v předchozích kapitolách, komunikace obou počítačů probíhá po bezdrátové síti LAN pomocí protokolu UDP/IP. Posílání paketů po síti je zobrazeno na obr. 5.1.



(obr. 5.1 – komunikace velín – vozík)

Jak je vidět s obrázku, program VELIN.EXE komunikuje jak s programem VOZIK.EXE, tak s programem PROTOKOL.EXE. I když se programy PROTOKOL.EXE a VELIN.EXE nacházejí na jednom počítači, je ke komunikaci použit také protokol UDP/IP (tyto programy se navzájem adresují jako „host“ – tedy adresou 127.0.0.1).

Jednotlivé zprávy jsou různě dlouhé. Jejich typ je určen prvním bytem, který obsahuje identifikační číslo zprávy (např. CMD_ZADANI = 0, CMD_PREPINACE = 1, atd.). Dále každá zpráva obsahuje určitý počet parametrů = proměnných různých typů. Typy *char*, *short*, *integer* a *float* jsou ukládány ve stejném formátu jako v jazyce C. Typ řetězec znaků *string* je vždy ukončen znakem středník (;). Zpráva CMD_OPAKOVAT může obsahovat libovolný počet parametrů typu *string*, proto je na konci ukončena znakem mřížka (#).

Struktura jednotlivých zpráv posílaných mezi aplikacemi VOZIK.EXE a VELIN.EXE je následující:

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Obsah	CMD_ZADANI	Rychlost				Čekání				Pulsy 1		Pulsy 2		Pulsy 3	
Typ	Char	Float				Float				Short		Short		Short	

Byte	0	1
Obsah	CMD_PREPINACE	Stav přepínačů
Typ	Char	Char

Byte	0
Obsah	CMD_STOP
Typ	Char

Byte	0
Obsah	CMD_DOJEZD
Typ	Char

Byte	0	1	2	3	4	5	6	7	8	9
Obsah	CMD_VYSLEDKY	Číslo vrtule	Dráha				Čas			
Typ	Char	Char	Float				Float			

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Obsah	CMD_STAV	Stav signálek				Akt. rychlost				Akt. poloha				Pulsy 1		Pulsy 2		Pulsy 3	
Typ	Char	Integer				Float				Float				Short		Short		Short	

Byte	0	1
Obsah	CMD_KONEC_MERENI	V pořádku?
Typ	Char	Char (bool)

Byte	0	1
Obsah	CMD_KONEC_DRAHY	V pořádku?
Typ	Char	Char (bool)

Struktura posílaných zpráv mezi aplikacemi VELIN.EXE a PROTOKOL.EXE je následující:

Byte	0	…	…	…	…	…	…	…	…	…
Obsah	CMD_INIT	Protokol	Pozice	Majitel	Výrobce	Tělo	Propeler	Upevnění	Datum	Jméno
Typ	Char	String	String	String	String	String	String	String	String	String

Byte	0	…	…	…	…	…	…
Obsah	CMD_MERENI	Číslo měření	Protokol	Rychlost	Pulsy	Dráha	Čas
Typ	Char	String	String	String	String	String	String

Byte	0
Obsah	CMD_WATCHDOG
Typ	Char

Byte	0
Obsah	CMD_POSLEDNI
Typ	Char

Byte	0
Obsah	CMD_PRERUSIT
Typ	Char

Byte	0	…	…	…	…	…
Obsah	CMD_OPAKOVAT	Číslo měření	Číslo měření	…	Číslo měření	#
Typ	Char	String	String	…	String	Char

Byte	0
Obsah	CMD_POKRACOVAT
Typ	Char

Byte	0
Obsah	CMD_UKONCIT
Typ	Char

Každá aplikace, která přijme jakoukoliv zprávu (kromě CMD_WATCHDOG), odešle odesílací aplikaci potvrzení v následujícím tvaru:

Byte	0	1
Obsah	CMD_CFM	Číslo přijaté zprávy (CMD_…)
Typ	Char	Char

Popis použití zpráv je uveden v kapitole 5.5.

Samotná realizace komunikace v protokolu je umožněna pomocí knihovny WINSOCK 1.1 (zkr. WINdows SOCKets, hlavičkový soubor *winsock.h*) [13]. Tato knihovna obsahuje množství API funkcí a zastřešuje komunikaci pomocí protokolů TCP/IP. *Socket* je koncový bod internetové komunikace a je určen IP adresou a číslem portu. Na jednom počítači tak můžeme vytvořit řadu socketů, které budou komunikovat s jinými sockety v síti.

Pro tuto komunikaci byla vytvořena třída **TTCPComm**, která umožňuje komunikaci mezi dvěma sockety. V konstruktoru vytvoří objekty místního socketu (místo, kde se bude přijímat) a vzdáleného socketu (místo, kam se bude posílat). Její potomkové slouží pro konkrétní aplikaci a každý z nich obsahuje dvě vlákna – přijímací a vysílací. Vysílací vlákno bere zprávy z fronty, kam je ukládá aplikace, a odesílá je vzdálenému socketu. Po každé odeslané zprávě čeká na potvrzení přijetí po určitý časový interval. Pokud doručení nepřijde, zkouší zprávu odeslat ještě dvakrát a až poté, v případě opakovaného neúspěchu, ohlásí chybu komunikace. Přijímací vlákno čeká na přijetí zpráv, zpracovává je a odesílá potvrzení.

V aplikaci VOZIK.EXE je pro komunikaci určena třída **TTCPCommWithMaster** (potomek **TTCPComm**), obsahující přijímací vlákno **TThreadRecvFromMaster** a vysílací vlákno **TThreadSendToMaster**). Obdobné třídy jsou v aplikaci VELIN.EXE.

5.5 Kalibrace řízená z velína

Při kalibraci z velína je nutné v programu na vozíku přepnout na řízení z velína, tím se ovládání programu na vozíku zablokuje (dostane se do stavu REMOTE CONTROL). Poté obsluha nasadí vrtule a odejde do velína.

Kalibrace pak probíhá stejně jako je znázorněno na obr. 4.14 v kap. 4.4.6.4, tedy pomocí pozměněného vlákna **TThreadMsrSet**. Obsluha ve velínovém počítači zadá veškeré parametry kalibrace a stiskne tlačítko START. Po kontrole parametrů se ve vývojovém diagramu 4.14 dostáváme do bloku MĚŘENÍ. Ten je oproti řízení z vozíku jednodušší. Zprávou CMD_ZADANI se pošlou vozíku parametry aktuálního měření a čeká se na přijetí zpráv CMD_VYSLEDKY obsahujících výsledky měření jednotlivých vrtulí. Ve vozíku zatím probíhá měření řízené vláknem **TThreadMsrOne**. Po dokončení měření pošle vozík zprávu CMD_KONEC_MERENI s parametrem, zda měření proběhlo v pořádku. Následuje skok do bloku DOJEZD NA KONEC DRÁHY v obr. 4.14, kdy velín odešle příkaz CMD_DOJEZD a čeká na přijetí zprávy CMD_KONEC_DRAHY s parametrem, zda se opravdu dojelo ke koncovému spínači. Mimoto se pravidelně posílá do velínu informace o stavu vozíku CMD_STAV, která slouží pro vizualizaci a funguje také jako watchdog správné funkce vozíku. Vozík lze kdykoliv okamžitě zastavit posláním zprávy CMD_STOP.

Před kalibrací se pošlou aplikaci PROTOKOL.EXE parametry všech vrtulí prostřednictvím zpráv CMD_INIT. Po každém skončeném měření se pro každou vrtuli posílá zpráva CMD_MERENI, obsahující výsledky měření vč. čísla vrtule a čísla protokolu. Po posledním měření je poslána zpráva CMD_POSLEDNI. Pravidelně se také posílá zpráva CMD_WATCHDOG.

Aplikace PROTOKOL.EXE má možnost řídit průběh kalibrace pomocí zpráv CMD_PRERUSIT, CMD_OPAKOVAT, CMD_POKRACOVAT a CMD_UKONCIT, které mají stejný význam jako stisk příslušných tlačítek v aplikaci VELIN.EXE. Ve zprávě

CMD_OPAKOVAT se posílá seznam čísel měření, která se mají zopakovat. Tato aplikace tedy průběžně dostává výsledky měření a ihned je graficky zobrazuje spolu s ostatními měřeními. V případě, že obsluha pohledem na graf zjistí, že některé měření jasně neodpovídá ostatním měřením (např. z důvodu dočasného zadrhnutí vrtule apod.), může kalibraci přerušit v jednom z programů (VELIN.EXE nebo PROTOKOL.EXE) a dané měření zopakovat.

6 Závěr

Řídící systém, popsaný v této práci, je v současné době (leden 2005) plně funkční až po vyšší řídící úroveň. Zajišťuje tedy automatickou kalibraci vodoměrných vrtulí, což znamená, že obsluha zadá v palubním počítači parametry kalibrace a může vozík opustit. Ten sám provede všechna měření a uloží je do protokolu.

Zprovozňování dohlížecí úrovně je nyní ve stádiu testů programů na bezdrátové síti za jízdy vozíku a bude uvedeno do plného provozu do března 2005. Je tomu tak z důvodu zpožďování zakázky vinou všech zúčastněných stran. Dosavadní testy však ukázaly, že i za nejvyšších rychlostí vozíku funguje bezdrátová komunikace bez chyb na přenosové lince.

Nový vozík včetně jeho řídícího systému je modernější než původní a bude umožňovat větší spektrum úkolů, které se po něm v budoucnu budou chtít. Prvním tímto úkolem bude kalibrace torpéd zavěšených na lanech za vozíkem. Dalším úkolem by mělo být měření odporu různých křovin tažených vodou pro zjišťování, které z nich jsou vhodné pro vysázení na březích řek jakožto ochrana proti povodni. Dále měření odporů modelů lodí tažených za vozíkem apod.

Současný řídící systém je vytvořen tak, aby jeho změna pro další úkoly byla co nejjednodušší. Programovatelná pole Xilinx mají rezervy a může se tak zvyšovat počet vstupů, výstupů i zaplnění vnitřních bloků pole. Objektově orientovaný a více-vláknový přístup psaní softwaru programů do obou počítačů zajišťuje modularitu a přehlednost programu a je tak velmi jednoduché přidat v budoucnu další funkce. Hardware počítačů byl volen tak, aby byl po delší dobu pokud možno dostačující i pro plánované úkoly.

7 Reference

[1] Václav Vacek. Sériová komunikace ve Win32. Ben, Praha 2003.

[2] Charles Petzold. Programování ve Windows. Computer Press, Praha 1999.

[3] David Matoušek. C++ Builder – vývojové prostředí. Ben, Praha 2000.

[4] David Matoušek. C++ Builder – vývojové prostředí, 2. díl. Ben, Praha 2001.

[5] Miroslav Virius. C++ Builder – kompletní průvodce. Grada, Praha 2000.

[6] Vladimír Haasz, Miloš Sedláček. Elektrická měření. Vydavatelství ČVUT, Praha 1998.

[7] EIA Standard RS-232-C. Electronic Industries Association, Washington 1969.

[8] EIA Standard RS-485. Electronic Industries Association, Washington 1969.

[9] Josef Janoušek, Vladimír Suchánek. Základy silnoproudé elektrotechniky. Vydavatelství ČVUT, Praha 1999.

[10] Frenic 5000G11S-EN Instruction Manual. Fuji Electric Co., 2004.

[11] Miroslav Virius. Od C k C++. Kopp, České Budějovice 2000.

[12] PIO-D96 User Manual. ICP DAS, 2003.

[13] Martin Hall, Mark Tofwig, Geoff Arnold, David Treadwell, Henry Sanders. Windows Sockets, An Open Interface For Network Programming Under Microsoft Windows. Microdyne, 1993.

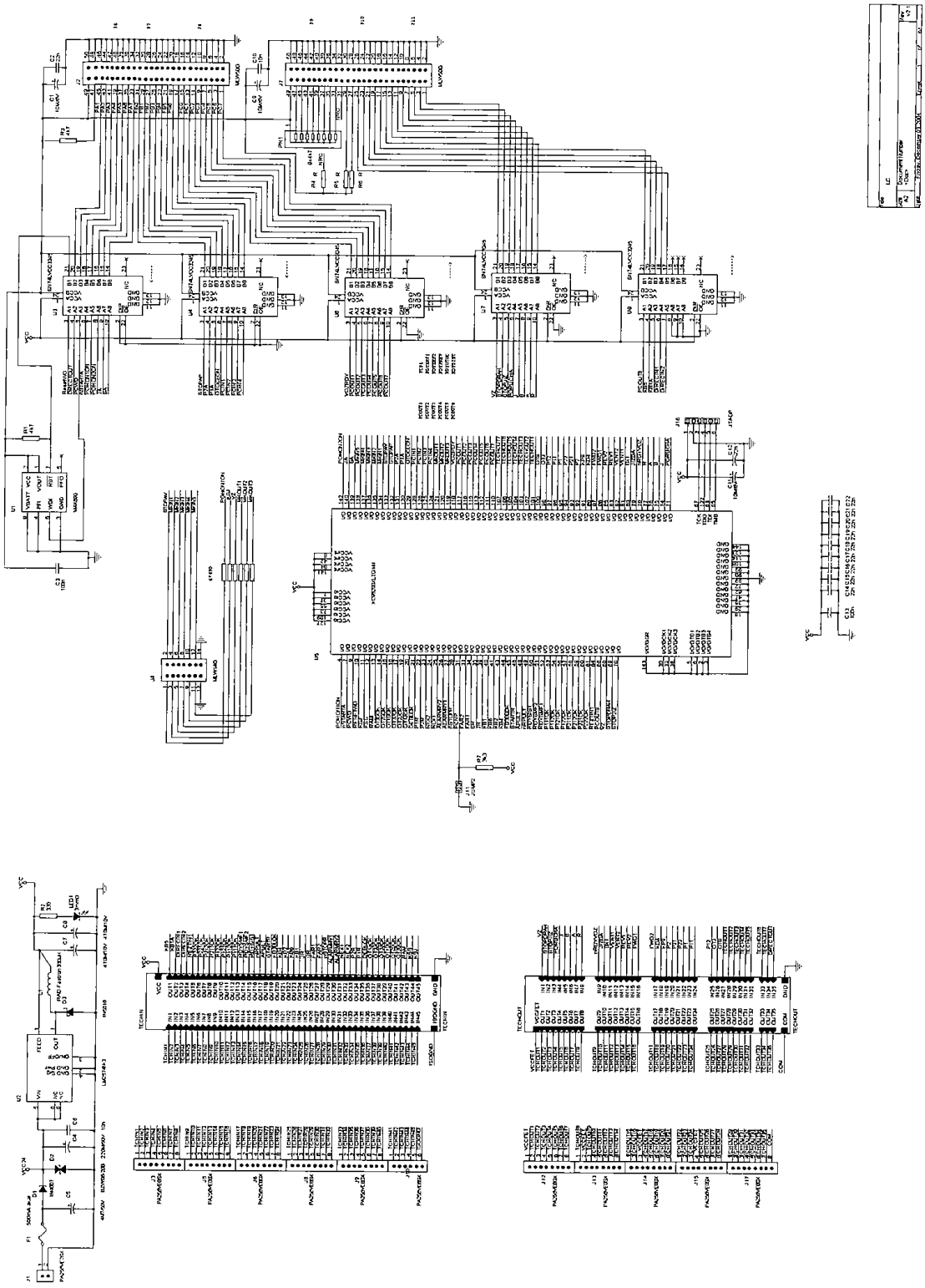
[14] Richard Šusta. Programování pro řízení ve Windows. Vydavatelství ČVUT, Praha 1999.

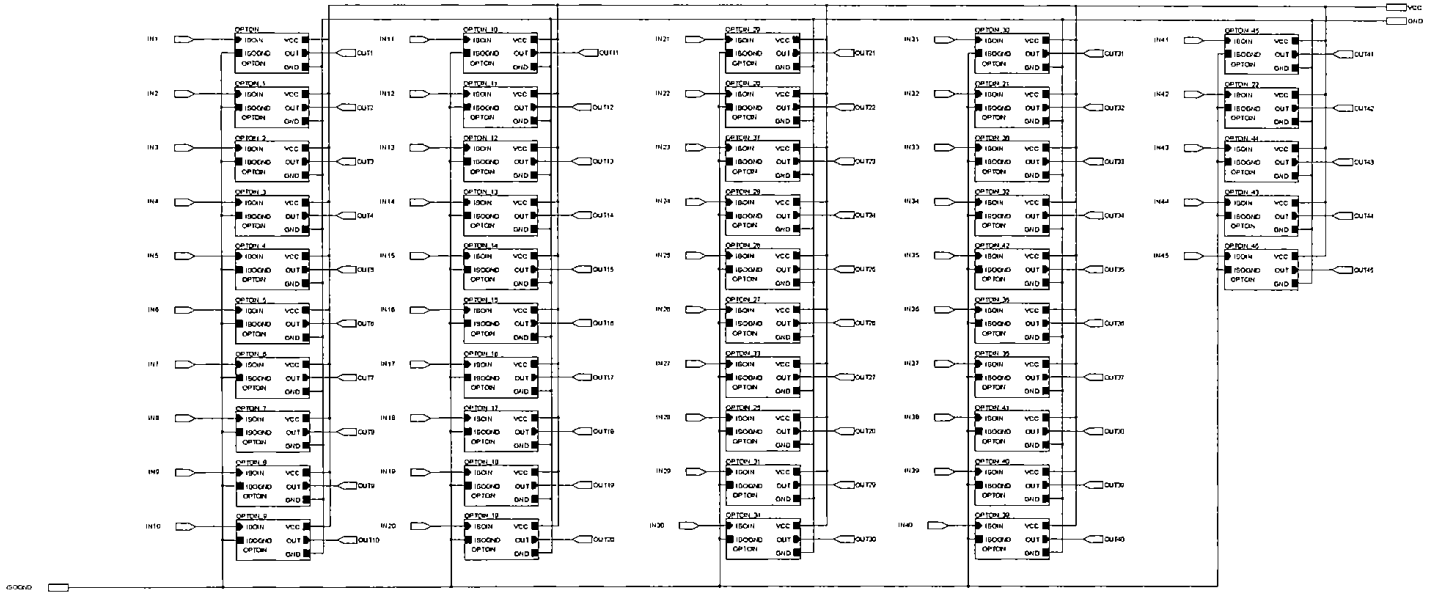
[15] Spartan-II 2.5 V FPGA Family: Functional Description. Xilinx, 2003.

[16] XC9500 In-System Programmable CPLD Family. Xilinx, 2003.

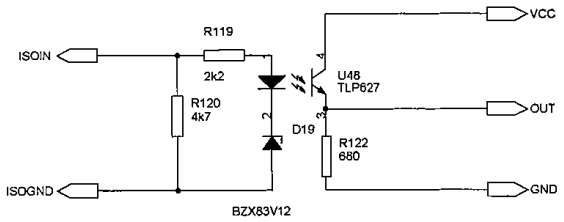
Příloha A Zapojení součástek na desce LC

včetně bloků TECHIN, OPTOIN, TECHOUT, FETOUT

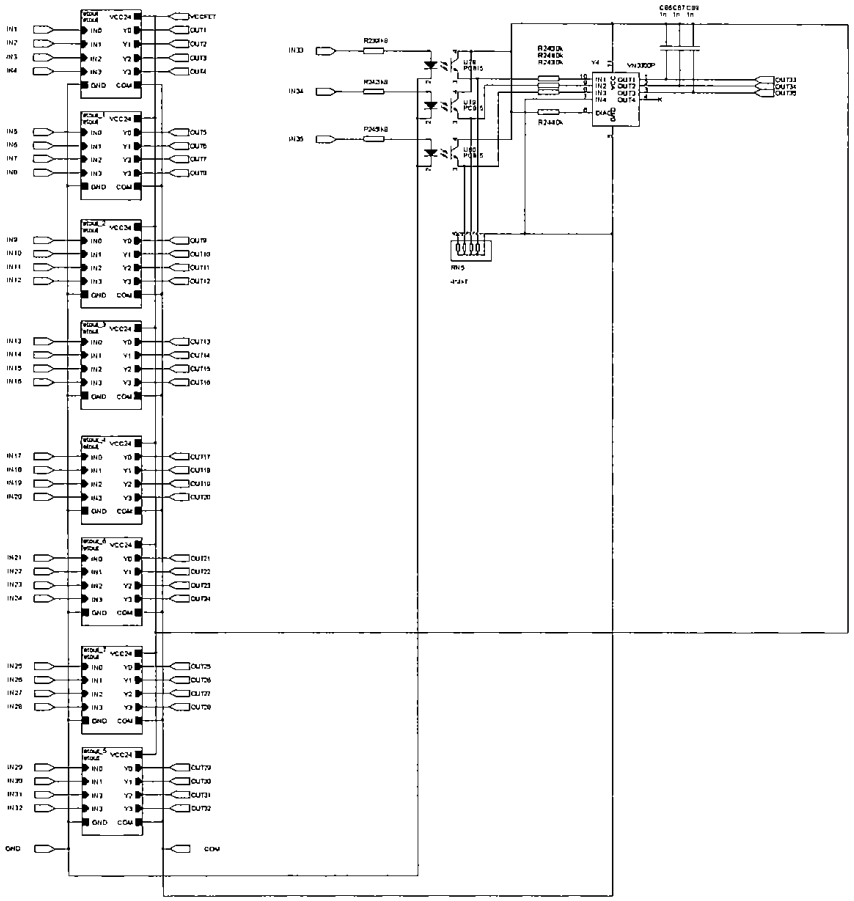




Title	LOTECHN	
Size A2	Document Number <Doc>	Rev v1.0
Date	Friday, June 11, 2004	Sheet 7 of 7

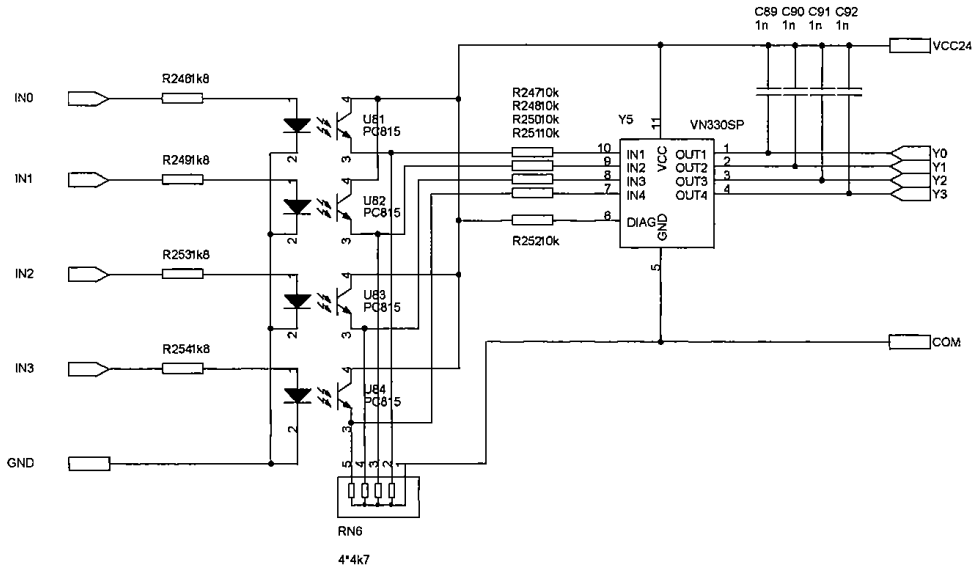


Title	LC/TECHIN/OPTOIN	
Size A4	Document Number <Doc>	Rev v2.0
Date: Friday, June 11, 2004	Sheet 28 of 63	



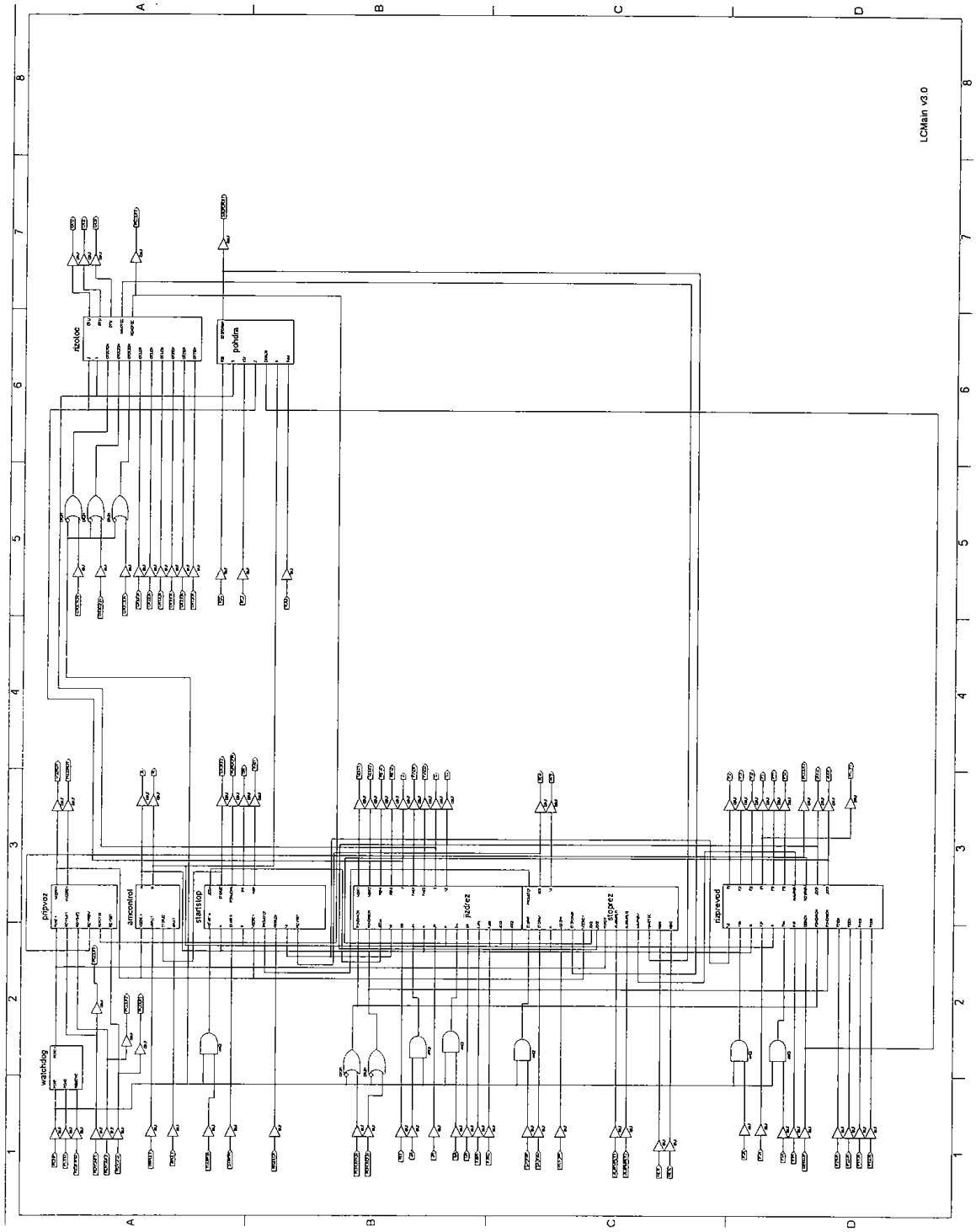
LGTECHOUT

Size A2	Document Number <Doc>	Rev v2.0
Date: Friday, July 11, 2003	Sheet 9 of 63	

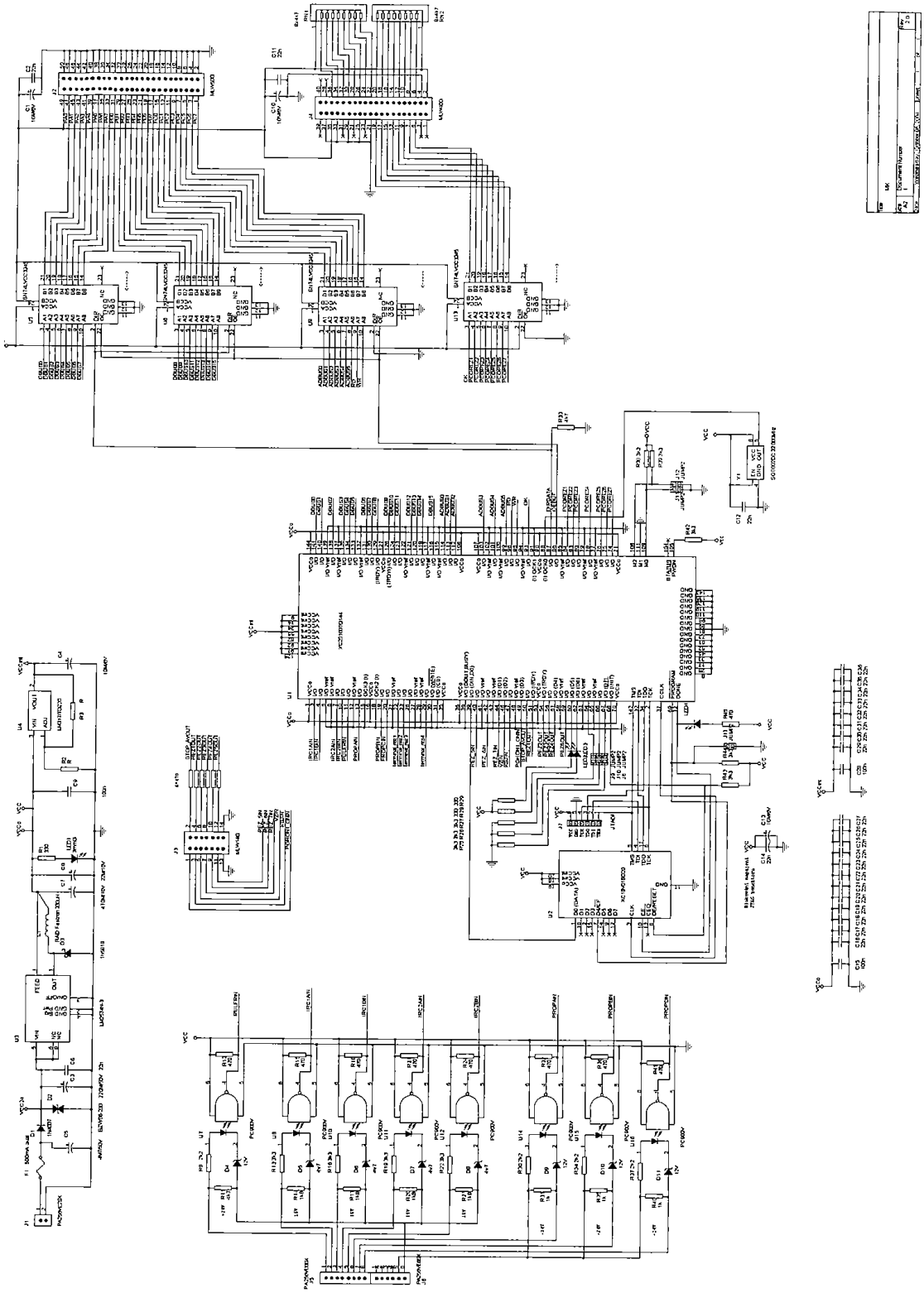


Title	LC/TECHOUT/FETOUT		
Size A4	Document Number <Doc>		Rev v2.0
Date: Friday, June 11, 2004	Sheet 59 of 63		

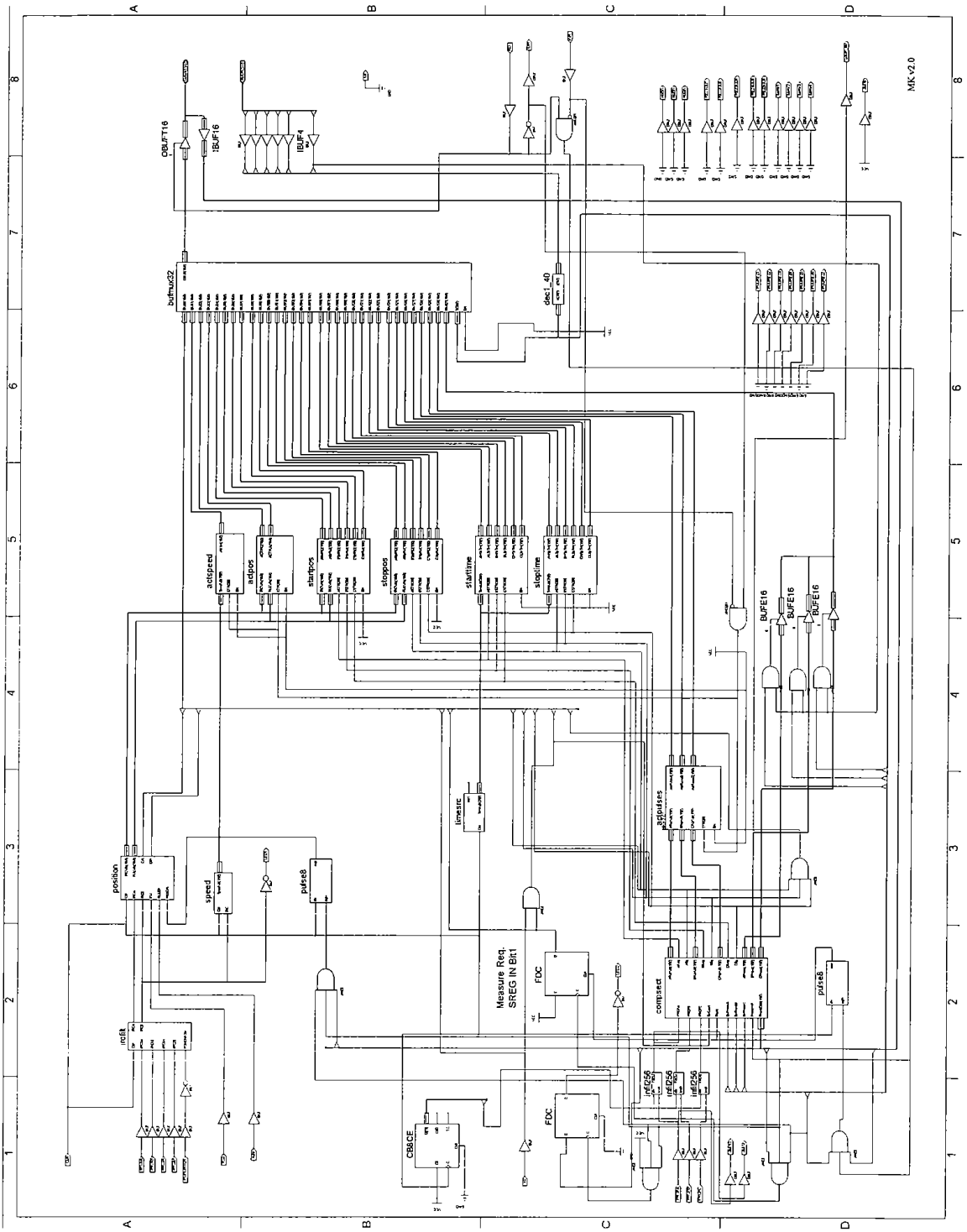
Příloha B Schéma desky LC



Příloha C Zapojení součástek na desce MK



Příloha D Schéma desky MK



Příloha E Seznam klávesových zkratek v programu

F2 - volba směru vozíku
F3 - přepínaní mezi řízením z vozíku a z velína
F4 - zapnutí / vypnutí otoče č.1
F5 - zapnutí / vypnutí otoče č.2
F6 - zapnutí / vypnutí otoče č.3
F7 - volba priority motorů
F8 - zapnutí / vypnutí pohonu č.1
F9 - zapnutí / vypnutí pohonu č.2

Ctrl + Enter - tlačítko START
Esc - tlačítko STOP

Ctrl + P - tlačítko PŘERUŠIT
Ctrl + O - tlačítko OPAKOVAT
Ctrl + R - tlačítko POKRAČOVAT
Ctrl + U - tlačítko UKONČIT

Ctrl + A - karta *Aktuální měření*
Ctrl + K - karta *Kalibrace*
Ctrl + V - karta *Výpis protokolů*

Ctrl + C - nastavení stavu *poloha nezkalibrována* (signálka NEZKALIB.), použije se v případě, že z nějakého důvodu poloha vozíku na obrazovce zjevně nesouhlasí s reálnou polohou, po stisku této kombinace a následném projetí jižního koncového spínače (směrem na sever) se znovu poloha zkalibruje

Příloha F Obsah přiloženého CD

V adresáři VOZIK se nachází zdrojové kódy programu pro palubní počítač a v adresáři KARTA zdrojové kódy pro simulátor vozíku. V adresáři EXE jsou uloženy spustitelné soubory VOZIK.EXE a KARTA.EXE včetně potřebných .DLL knihoven a obrázků. Pro simulaci jsou nutné obě spuštěné aplikace (spouští se navzájem). Je vhodné adresář EXE zkopírovat na pevný disk, přejmenovat a změnit cestu ve VUV.INI (viz dále), protože se do tohoto adresáře ukládají databáze.

Adresář MERENI obsahuje ukázku souboru s výsledky kalibrace.

V kořenovém adresáři je také soubor VUV.INI, který je nutné nahrát do adresáře WINDOWS (nebo WINNT) a případně změnit tyto cesty:

[PATHS]
Main=c:/Vozik/ => cesta k samotné aplikaci VOZIK.EXE a k databázím
Protocols=c:/Mereni/ => cesta k adresáři, kam se ukládají výsledky měření

V adresáři MANUALY jsou uloženy dokumenty k desce I/O, polím Xilinx a ke knihovně WinSock.

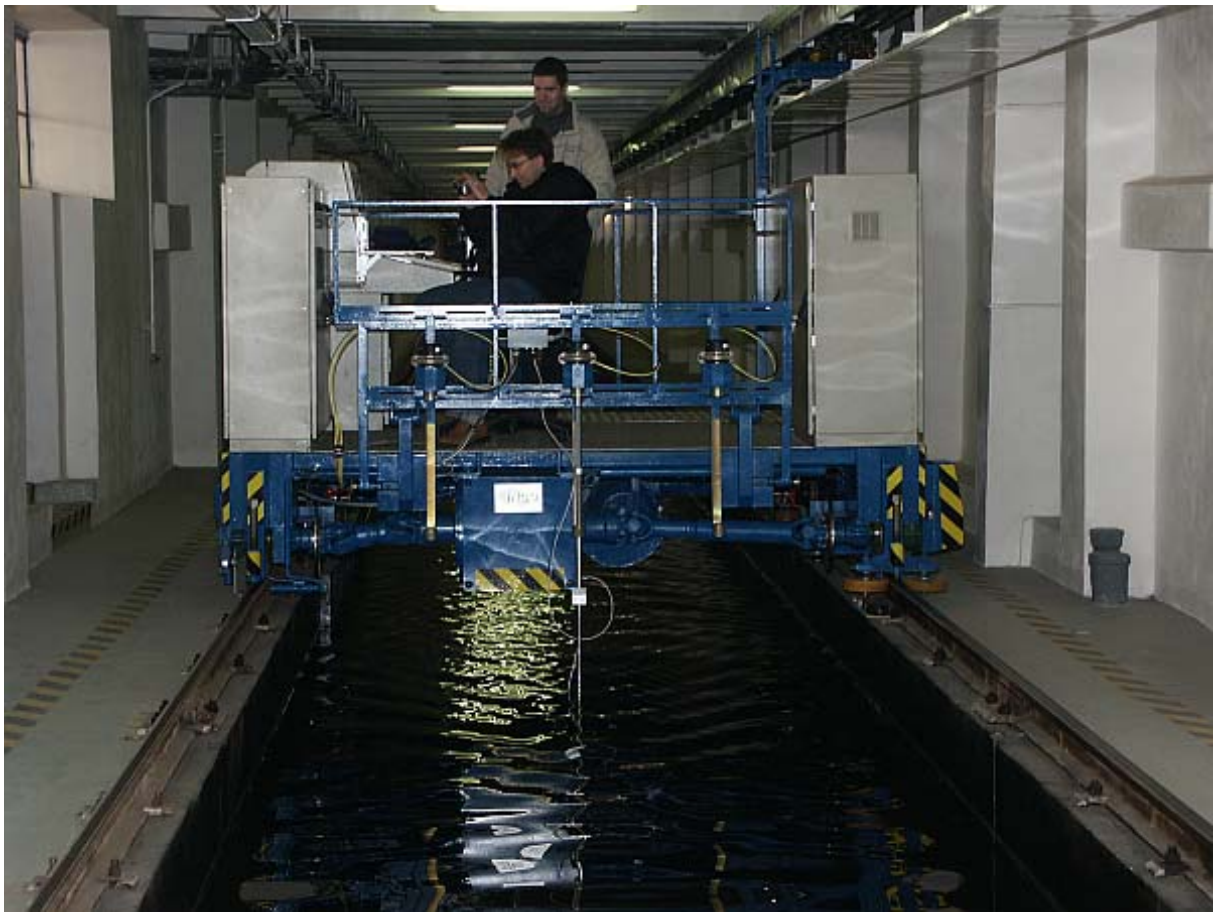
Adresář FOTKY obsahuje množství fotek objektu VÚV, kalibračního kanálu, vozíku a jeho příslušenství.

Pozn.: program pro velín, který je stále ve vývoji, není přiložen na CD.

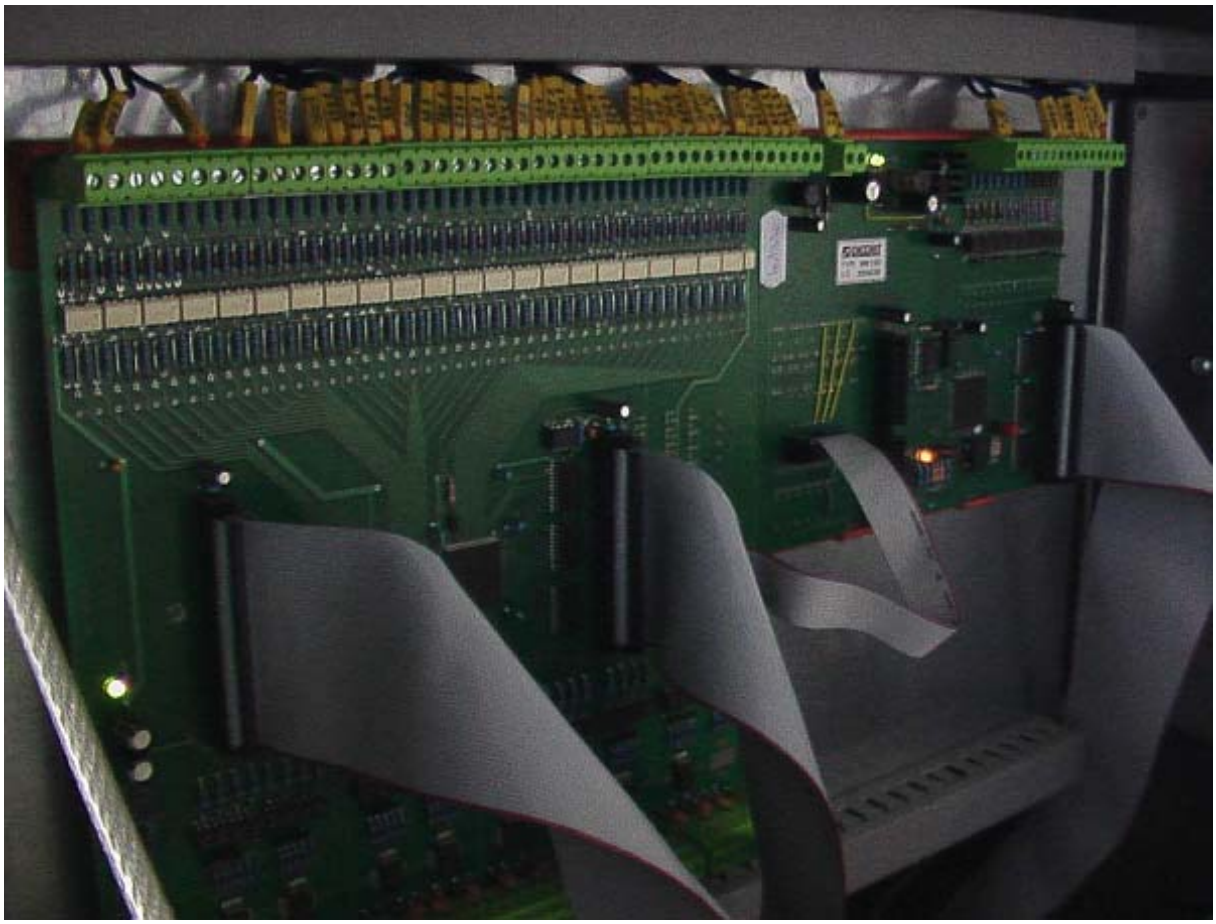
Příloha G Fotografie

Popis:

1) kalibrační kanál
2) vozík
3) na vozíku probíhá kalibrace vrtule na prostřední tyči
4) řídící pult (vpravo), brzdný systém Sistonik (vlevo)
5) deska MK (vpravo), deska LC (vlevo)
6) frekvenční měniče
7) vrtule
8) vozík jede k severnímu konci kanálu



DICONT



FRENIC
FRENIC