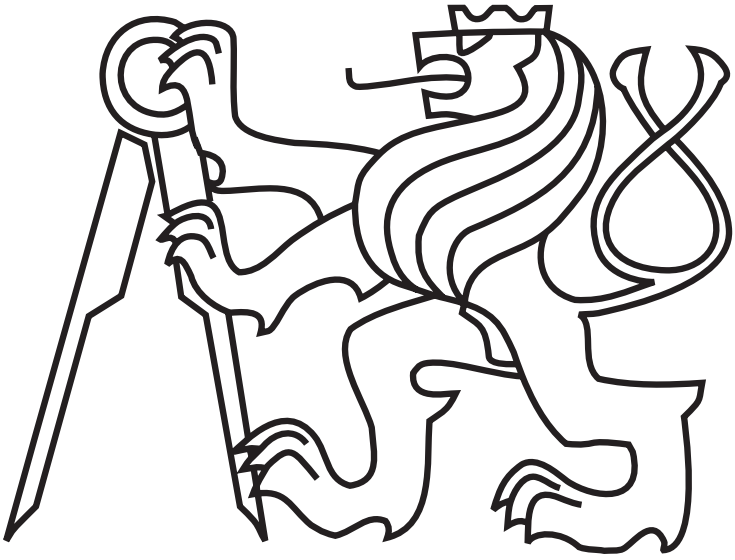
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra řídicí techinky

Diplomová práce



Bc. Vojtěch Pavlík

Měření a řízení spotřeby elektrické energie na výrobních linkách

Vedoucí práce: **Ing. Pavel Burget, Ph.D.**

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Vojtěch Pavlík**

Studijní program: Kybernetika a robotika
Obor: Systémy a řízení

Název tématu: **Měření a řízení spotřeby elektrické energie na výrobních linkách**

Pokyny pro vypracování:

1. Vytvořte zařízení pro měření spotřeby elektrické energie v třífázové síti s připojením k síti Profinet a s možností sběru dat do databáze.
2. Proveďte měření na průmyslovém robotu pro jeho různé provozní stavy včetně stavu se sníženou spotřebou energie.
3. Rozšiřte systém měření spotřeby na reálnou linku v automobilovém průmyslu. Na základě znalosti řídicího programu linky ukládejte do databáze spolu s měřenými hodnotami také provozní stavy linky.
4. Na základě analýzy provozních stavů navrhněte strategii uvádění linky do úsporného režimu a zase zpět do provozního. Strategii implementujte a ověřte.
5. Vytvořte SW modul, který pro danou linku na základě měření spotřeby odhadne její provozní stavy, aniž byste využil znalost stavového automatu pro řízení linky.

Seznam odborné literatury:

Profil PROFIenergy, v1.2. Profibus&Profinet International, 2011.
Profinet System Description. Simatic System Manual. Siemens 07/2010.

Vedoucí: Ing. Pavel Burget, Ph.D.

Platnost zadání: do konce letního semestru 2014/2015

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická
L.S.

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 12. 2. 2014

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 11.5.2014 .....

Abstrakt

Tato práce se zabývá měřením a návrhy pro řízení spotřeby reálné průmyslové linky. Vzhledem k oblasti nasazení je měření realizováno pomocí stanic připojených k síti PROFINET. Změřená data a procesní data z PLC, které řídí linku, jsou odesílána pomocí specializovaného programu do databáze pro případnou další analýzu. Na základě těchto dat a studia průmyslových robotů byl navržen postup pro uvedení linky do úsporného módu a obnovení jejího provozu. Dále byl na zařízeních, které to podporují, vyzkoušen PROFIenergy - profil pro úsporu energie.

Abstract

This thesis covers measurement and management of energy consumption of industrial line. Stations with PROFINET connectivity are used for measurement, because of deployment in automation.For future analysis are measured values and state information from control PLC send to database. This task is accomplished by specialized program, which is described in part of this thesis. Procedure for putting the line into sleep mode and restore its operation was proposed based on this data and study of industrial robots. Profile PROFIenergy for managing energy consumption of certain devices was also tested.

Obsah

1 Úvod **1**

1.1 Motivace 1

1.1.1 Měření 1

1.1.2 Reálná aplikace 2

1.2 Očekáváný přínos práce 2

2 Měření výkonu s připojením k síti PROFINET **3**

2.1 Měření výkonu 3

2.1.1 Výkon harmonického napětí a proudu 3

2.1.2 Měření výkonu v trojfázové síti 4

2.2 Měřicí karty s rozhraním PROFINET 5

2.2.1 Siemens: AI Energy METER ST 5

2.2.2 WAGO: 3-Phase Power Measurement Module 8

2.2.3 Srovnání parametrů měřicích karet 8

2.3 Měřicí rozvaděč 11

2.4 Adresace a paměťové oblasti v PLC Siemens 13

2.4.1 Přímá adresace 13

2.4.2 Nepřímá adresace 14

2.5 Sběr dat 15

2.5.1 Definice požadavků 16

2.5.2 Domluvený formát 17

2.5.3 Komunikace pomocí S7-CPU 17

2.5.4 Popis programu 19

2.5.5 Podrobný popis dekódování 23
2.5.6 Aplikace DEPO 25

3 Měření výkonu průmyslového robotu 26

3.1 Robot KUKA KR5 ARC 27
3.1.1 Spouštění robotu v režimu EXT 29
3.1.2 Programovací jazyk KRL 29
3.1.3 Submit interpreter 30
3.1.4 Trajektorie robotu 30
3.1.5 Datové struktury robotu 31
3.2 Popis profilu PROFIENERGY 31
3.2.1 Obecný popis 32
3.2.2 FB816 PE_START_END 33
3.2.3 Jednotlivé příkazy 34
3.3 PROFIenergy na robotu KUKA 35
3.3.1 PE Identify 35
3.3.2 PE Query Modes 36
3.3.3 Start_Pause 37
3.3.4 End_Pause 37
3.3.5 Go_Sleep_Mode_WOL 37
3.4 Wake On Lan 38
3.5 Měření a sběr dat 38
3.6 Naměřená data 39
3.6.1 Provozní módy 39
3.6.2 Různá rychlost pohybu 41

4 Rozšíření měření na reálnou linku **43**

4.1 Průmyslová linka 43

4.1.1 Digitální továrna Tecnomatix - Process Simulate 44

4.1.2 Základní popis výroby dílu 44

4.1.3 Rozbor proměnných linky 44

4.1.4 Měřicí periferie 45

4.2 Úpravy programu pro sběr dat 46

4.2.1 PLC pro sběr procesních dat 46

4.2.2 PLC pro sběr dat z měření 46

5 Analýza změřených dat **48**

5.1 Prvotní analýza dat 48

5.1.1 Data z měřicího PLC 48

5.1.2 Data z řídicího PLC 50

5.2 Analýza odstávek 50

5.2.1 Detekce pauz pomocí signálu stolu 50

5.3 Analýza spotřeby v jednotlivých režimech 52

5.4 Měření robotu na dílně 53

5.5 Popis VKRC 53

5.5.1 PROFIenergy na VKRC 54

5.5.2 Rozhraní VKRC pro KRC 55

5.6 Strategie uspávání 55

5.6.1 Spuštění úspory energie 55

5.6.2 Postup uspání a probuzení 56

5.7 Ověření funkčnosti 58

6 Softwarový modul **59**

6.1 Zadání a výstup modulů 59

6.2 Implementace jednoduššího modulu (A) pro roboty 60

6.3 Implementace modulu pro další zařízení 60

6.3.1 Postup klasifikace pro dopravníky 61

6.3.2 Postup klasifikace pro lepicí stanici 62

6.4 Ověření funkce klasifikace stavu linky 63

6.5 Analýza spotřeby jednotlivých operací robotů 63

6.6 Implementace modulu klasifikace operací pro roboty 65

6.6.1 Dělení dat 65

6.6.2 K-means 67

6.7 Ověření funkce klasifikace operací 69

7 Závěr **70**

7.1 Robot KUKA KR5 ARC 70

7.2 Nasazení na reálné lince 70

A Seznam zkratek **I**

B Příloha- Tabulky **II**

C Příloha- Obrázky **IV**

D Příloha CD **V**

D.1 Obsah CD V

E Příloha - soubory KRC **VI**

E.1 CELL.SRC VI

E.2 SPS.SUB IX

E.3 CONFIG.DAT X

E.4 OPTION.DAT X

F Příloha - Používání robotu KUKA **XII**

G PE na robotu KUKA **XIII**

G.1 Mód DriveBus power off XIII

G.1.1 Uspání XIII

G.1.2 Probuzení XIII

G.2 Mód hibernace XIV

G.2.1 Uspání XIV

G.2.2 Probuzení XIV

Seznam obrázků

1 Zapojení wattmetrů [2] 4

2 Příklad konfigurace stanice ET200SP [3] 6

3 Příklad nastavení výstupů a dat odeslaných na vstup PLC [5] 11

4 Měřicí rozvaděč 12

5 Příklad konfigurace stanice ET200SP [6] 13

6 Blok FB5630 19

7 Popis programu pro sběr dat 21

8 FC5631: automatický start sběru dat a jejich odesílání 23

9 Robot KUKA KR5 ARC 26

10 Příklad konfigurace industriálního robotu 27

11 Postup pro spuštění v režimu EXT, RQ výstupy PLC, RI vstupy PLC . . . 29

12 Struktura souborů v KRC 32

13 Výpočet doby pauzy podle [13] 33

14 Vstupy a výstupy FB816 [14] 34

15 Příkaz PE:Identify 36

16 Různé provozní módy robotu 39

17 Měření pro různé rychlosti pohybu po stejné trajektorii 40

18 Spotřeba robotu pro jednotlivé pohyby 42

19 Data naměřená pro jeden takt linky 49

20 Délky pauz 51

21 Rozdělení délky pauz s určením prahu 52

22 Různé provozní módy VKRC 54

23 Klasifikace operací otočného stolu 61

24 Detekce stavu dopravníků, fréz a otočného stolu 62

25 Detekce stavu celé linky 64

26 Úprava délky operací v závislosti na změřených datech 66

27 Délka operací a energie během nich spotřebovaná 67

28 Klasifikace operací robotu 1 68

29 Operace z mnoha cyklů IV

30 Automatic system start and normal operation with program number acknowledgement by means of $EXT_START XV

Seznam tabulek

1 Nastavení parametrů karty AI energy Meter ST 7

2 Dataset 0xFD - Active Power L1L2L3 [4] 9

3 Nastavení parametrů karty 750-494 10

4 Paměťové oblasti SIMATIC S7-CPU a příklady adresace 14

5 Mezioblastní ukazatel 15

6 Ukazatel DB 15

7 Ukazatel ANY 16

8 Konkrétní hodnoty pro ukazatel P#MW 4 16

9 Nastavení UDT65 a UDT66 18

10 Parametry bloku FB5630 20

11 Parametry bloku FC5631 22

12 Uživatelské struktury 24

13 PE_Mode 37

14 Spotřeba robotů v klidu 53

15 Porovnání měřicích karet výrobců WAGO a Siemens II

16 Spotřeba a délka operací jednotlivých robotů III

1 Úvod

1.1 Motivace

Otázka úspory energie je téma, kterým se v poslední době zabývá čím dál více lidí. Jedním z důvodů je tlak na snižování výrobních nákladů a maximalizaci zisku. Dalším důvodem je ochrana životního prostředí. Největší neobnovitelné zdroje energie, které lidstvo využívá, dříve či později dojdou a je potřeba jimi šetřit. S tím souvisí i různá nařízení, která musejí výrobci splňovat a tedy úspora energie je pro výrobce důležitá, protože výrobci mohou zvýšit svůj zisk a snáze splnit tato nařízení.

Spořit energii je ve výrobě možné například optimalizací cest dílů v lince, změnou rychlosti pohybů robotů, vypínáním celé linky v době nečinnosti nebo využitím funkcí pro úsporu energie. Pro potřeby úspory energie na průmyslových linkách vznikl profil PROFIenergy pro průmyslové sítě PROFINET. PROFIenergy umožňuje ze zařízení získat informace o tom, jaké součásti PROFIenergy podporuje a dále mezi jakými energetickými stavy může zařízení procházet. Obecně zařízení, které disponuje možností PROFIenergy podporuje několik energetických módů. Konkrétně pak mód pro běžné používání zařízení "Ready to Operate"a další mód, který je energeticky úsporný.

U velkého počtu linek se nevyužívá možnosti vypnutí celé linky. Obecně to totiž přináší více problémů než užitku. Uvést vypnutou linku do provozu je složité. Profil PROFIenergy nevypíná zařízení úplně celé, ale například u robotického kontroléru odpojí napájení servoměničů. Tím je šetřena nějaká část energie a obnovení provozu je potom poměrně rychlé.

1.1.1 Měření

Aby bylo možné určit spotřeby různých zařízení, je nutné instalovat měření. Protože se tato práce zabývá měřením a řízením spotřeby v oblasti průmyslové automatizace, je vhodné využít takové měřicí zařízení, které je možné řídit pomocí průmyslových kontrolérů. Dále je pak vhodné nějakým způsobem sbírat naměřená data pro další analýzu.

Pro vyzkoušení mechanismů úspory energie a práce s robotem je pro studenty dostupný malý průmyslový robot umístěný v budově E na Karlově náměstí. Jedná se o malý manipulátor, který je ale řízen plnohodnotnou řídicí jednotkou s funkcí PROFIenergy.

Aby bylo možné analyzovat a řídit spotřebu celé průmyslové linky, je nejdříve potřeba měřit spotřebu této linky. Protože se linka skládá z mnoha menších zařízení, která provádějí dílčí práce, je vhodné měřit spotřebu právě těchto zařízení. Z té by pak mělo být možné zjistit, co které zařízení dělá. Některá z těchto zařízení podporují profil PROFIenergy nebo je možné je vypnout, aniž by opětovné zapnutí působilo větší problémy.

1.1.2 Reálná aplikace

Jak již bylo zmíněno, jedním z cílů práce by měla být analýza spotřeby reálné linky a návrh strategie pro spouštění úspory energie, kterou by tato linka měla disponovat. Pro reálné nasazení byla vybrána jedna z linek svařovny závodu Škoda Auto. Na těchto linkách jsou nasazeny roboty KUKA, které by měly podporovat PROFIenergy.

1.2 Očekáváný přínos práce

- řešení spotřeby v reálné aplikaci
- analýza spotřeby průmyslové linky
- analýza odstávek průmyslové linky
- tvorba konfigurovatelného (bez restartu PLC) programu pro odesílání dat z PLC
- popis práce s robotem KUKA v laboratoři na Karlově náměsí

2 Měření výkonu s připojením k síti PROFINET

V této kapitole je popsána aplikace pro měření výkonu průmyslového robotu. Jsou zde popsané vztahy pro výkon jako takový. Ty jsou založeny na znalostech z absolvovaných kurzů v průběhu studia viz [1] a [2]. Dále jsou zde popsány měřicí karty, které byly pro měření výkonu použity a případně pak jejich konkrétní nastavení pro měření na robotu v laboratoři. Aplikace pro sběr dat využívá nepřímé adresace PLC a i té je věnována část této kapitoly.

2.1 Měření výkonu

2.1.1 Výkon harmonického napětí a proudu

Okamžitý výkon je možné vypočítat pomocí okamžitého proudu a napětí podle vztahu

$$p(t) = u(t)i(t), \tag{1}$$

kde

$$\begin{aligned} u(t) &= U_m \cos(\omega_0 t + \psi + \varphi) \\ i(t) &= I_m \cos(\omega_0 t + \psi), \end{aligned} \tag{2}$$

kde U_m je maximální hodnota střídavého napětí, I_m maximální hodnota střídavého proudu, ω je frekvence a ψ je počáteční posun fáze.

Úhel φ ve výše uvedených vztazích představuje posun časového průběhu okamžitého napětí oproti průběhu okamžitého proudu. Pokud je úhel φ nenulový, dělí se výkon na činnou a jalovou složku.

Činný výkon vyjadřuje hodnotu práce, kterou vykoná časově proměnný elektrický proud za jednotku času a je možné ho počítat pomocí následujícího vztahu:

$$P = \frac{1}{T} \int_{t_0}^{t_0+T} p(t)\mathrm{d}t = U I \cos\varphi, \tag{3}$$

kde U je efektivní hodnota napětí, I efektivní hodnota proudu. Jalový výkon je potom možné spočítat takto:

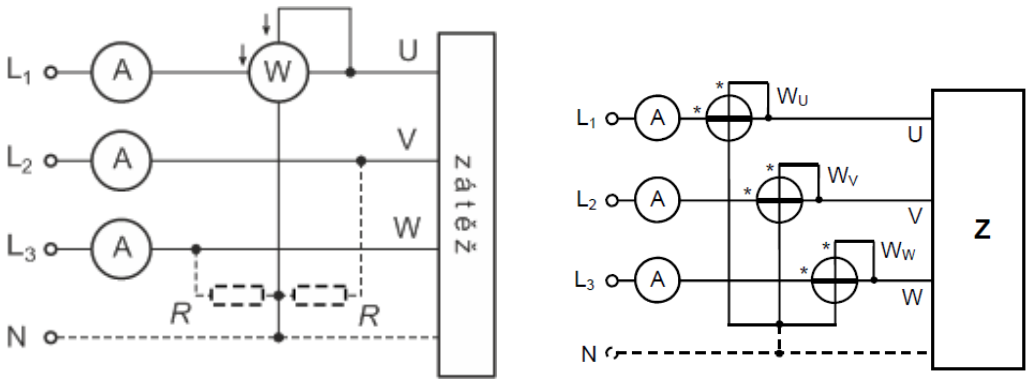
$$Q = UI \sin \varphi, \tag{4}$$

kde U a I značí efektivní hodnoty napětí a proudu.

2.1.2 Měření výkonu v trojfázové síti

Na základě popisu výpočtu v kapitole 2.1.1 je pro měření výkonu elektrického spotřebiče potřeba měřit proud, který protéká spotřebičem a napětí, které je na vstupních svorkách tohoto spotřebiče. Z těchto hodnot je poté možné vypočítat výkon.

V případě třífázové soustavy může být měření celkového výkonu spotřebiče možné použít pouze jeden wattmetr, pokud je zátěž na jednotlivých fázích souměrná. Pokud ne, je potřeba použít 3 wattmetry. Zapojení wattmetrů pro oba případy je vidět na Obrázku 1.



(a) (b)

Obrázek 1: Zapojení wattmetrů [2]

Celkový výkon je potom v pořadí pro jeden wattmetr, tři wattmetry:

$$P_{C1} = 3P_U = 3P_V = 3P_W,$$
$$P_{C3} = P_U + P_V + P_W \tag{5}$$

2.2 Měřicí karty s rozhraním PROFINET

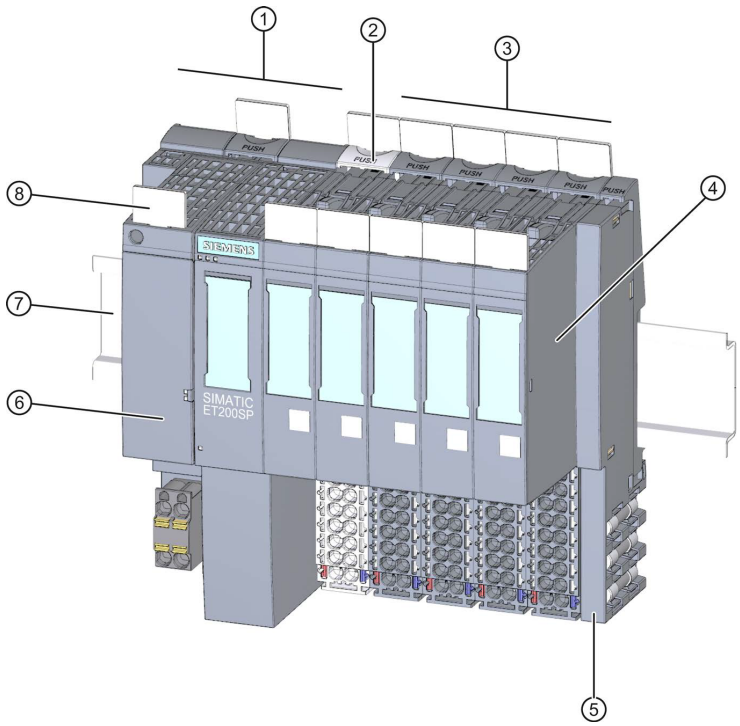
Pro řízení průmyslovými logickými automaty (PLC) existují analogové karty pro měření napětí a proudu. Navíc někteří výrobci nabízí přímo karty specializované na měření výkonu. Obecně tyto karty podporují měření stejnosměrného (DC) a střídavého (AC) proudu a to i pro trojfázové spotřebiče. Oproti analogovým kartám, které měří jen napětí či proud, tyto karty přímo počítají hodnotu jednotlivých složek výkonu, zvládají frekvenční analýzu, diagnostikují parametry sítě nebo počítají spotřebu. Pro měření proudu je u těchto karet využito proudových transformátorů pro převod vyššího proudu na nižší, který odpovídá rozsahu svorek karet. Rozsah je zpravidla 0-1A nebo 0-5A.

Pro měření byly vybrány příslušné speciální karty tří výrobců: Siemens, WAGO a B&R (konkrétně karta X20AP3121). Více karet bylo zvoleno pro možnost komparace jejich vlastností a získání dat do PLC. Poslední zmíněná karta nebyla nakonec testována, pro složitost implementace komunikačního protokolu (FlatStream) mezi PLC a periferií.

Tyto karty byly připojeny pomocí příslušných stanic s rozhraním PROFINET k PLC výrobce Siemens: S7-300 CPU 315-2PN/DP.

2.2.1 Siemens: AI Energy METER ST

Pro měření je využita sestava s připojením k síti PROFINET skládající se z hlavy ET200SP, karty digitálních vstupů 16DIx24VDC ST a analogové měřicí karty AI EnergyMeter ST. Obecná konfigurace stanice ET200SP je vidět na Obrázku 2. Hardwarová konfigurace takovéto stanice se tedy sestává z hlavy (modul s rozhraním) a několika potenciálových skupin s připojenými I/O kartami. potenciálová skupina začíná světlou BaseUnit (držák karty s vyvedenými vstupy či výstupy a propojením na sběrnici stanice) a poskytuje následujícím kartám napájení. Přestože je měřicí karta interně napájena z měřených fází, není možno stanici konfigurovat bez modulu se světlou BaseUnit. Světlá BaseUnit pro měřicí kartu neexistuje. Proto je v použité konfiguraci předřazena karta digitálních vstupů, která tuto BaseUnit má.



Obrázek 2: Příklad konfigurace stanice ET200SP [3]

1. Modul s rozhraním
2. Světlá BaseUnit BU..D s propojením napájecího napětí nebo pro vytvoření nové potenciálové skupiny
3. Tmavá BaseUnit BU..B pro vedení potenciálové skupiny dále
4. I/O moduly
5. Server modul (součástí dodávky modulu s rozhraním)
6. Adaptér sběrnice
7. Montážní lišta
8. Referenční identifikační štítky

Měřicí karta se nastavuje pomocí HW konfigurace. Získání požadovaných měřených či vypočítaných veličin je provedeno pomocí vstupů a výstupů karty. Konkrétní nastavení parametrů karty je uvedeno v Tabluce 1. Využit byl měřicí transformátor s převodovým poměrem 60A na 5A, a tedy rozsah karty byl nastaven 5A.

Phase mode	3P4W 3-phase, 4-wire		
Voltage measuring range	230 V		
Energy meter gate	50 Hz		
Line voltage tolerance [%]	10		
Line voltage diagnostics	0		
User data variant	0xFD		
	L1	L2	L3
Diganostics overflow current	0	0	0
Diagnostics overflow cumulative values	0	0	0
Diagnostics of low limit voltage	0	0	0
Diagnostics underflow voltage	0	0	0
Diagnostics overflow voltage	0	0	0
Tolerance factor overcurrent [0.1A]	100	100	100
Tolerance time overcurrent [ms]	40000	40000	40000
Current transformer factor	4	4	4
Current transformer secondary current	5 A	5 A	5 A
Reverse current direction	0	0	0

Tabulka 1: Nastavení parametrů karty AI energy Meter ST

Jak již bylo výše uvedeno, karta komunikuje s PLC pomocí skupiny vstupů a výstupů, kde zabírá vstupní prostor 32 bajtů a výstupní 12. Z pohledu PLC má karta 32I/12Q. Nastavením výrobcem definovaného ID do prvního výstupního bajtu je možno aktivo-

vat vyčítání konkrétního "datasetu"(předdefinovaný výčet měřených/vypočtených hodnot) z karty. Vyčítání je prováděno cyklicky přímo na vstupy PLC. Pro případ měření činného výkonu je nejvhodnější použít dataset s ID 0xFD - Active Power L1,L2,L3. Jeho obsah je vidět v Tabulce 2.

2.2.2 WAGO: 3-Phase Power Measurement Module

Pro měření je využita sestava s připojením k síti PROFINET skládající se z hlavy WAGO 750-350 a analogové karty WAGO 750-494-0001. Základní nastavení karty je provedeno z HW konfigurace ve STEP7. Získání požadovaných měřených či vypočítaných veličin je provedeno pomocí vstupů a výstupů karty. Konkrétní nastavení parametrů karty je uvedeno v Tabulce 3.

Karta v prostoru PLC zabírá 24I/24Q. Tato karta oproti výše uvedené měřicí kartě výrobce Siemens nepoužívá dataset, ale na vstupy PLC odesílá až 4 konfigurovatelné hodnoty. Nastavení konkrétních odesílaných hodnot je provedeno pomocí výstupních bajtů PLC. Na Obrázku 3 je vidět příklad nastavení výstupů a hodnot, které potom karta vrací na vstup. Kolonka "COL_ID"je pro nastavení AC nebo DC měření. Následující kolonky "MET_ID"pak obsahují požadavek na konkrétní měřenou/vypočítanou hodnotu z karty. Pro měření výkonu je nastavení "MET_ID": 007:Active power L1, 008:Active power L2 a 009:Active power L3. Čísla "MET_ID"pro další měřené a vypočtené hodnoty je možné najít v manuálu k měřicí kartě [5].

Karta pracuje s formátem Little Endian, centrála Siemens oproti tomu se systémem Big Endian a je potřeba s tím počítat.

2.2.3 Srovnání parametrů měřicích karet

V Tabulce 15 (příloha A) je vidět porovnání důležitých parametrů měřicích karet. Karta výrobce WAGO je z pohledu měření na tom lépe - má rychlejší vzorkování měřených hodnot

Byte	Allocation	No.
0	User data variant = 253 (0xFD)	-
1	0	-
2...3	Current I L1	7
4...5	Current I L2	8
6...7	Current I L3	9
8...9	Active power L1	23
10...11	Active power L2	24
12...13	Active power L3	25
14...15	Total active power L1L2L3	26
16...19	Total active energy L1L2L3	32
20	Power factor L1	10
21	Power factor L2	11
22	Power factor L3	12
23	Total power factor L1L2L3	13
24	Scaling current L1	-
25	Scaling current L2	-
26	Scaling current L3	-
27	Scaling active power L1	-
28	Scaling active power L2	-
29	Scaling active power L3	-
30	Scaling active power L1L2L3	-
31	Scaling total active energy L1L2L3	-

Tabulka 2: Dataset 0xFD - Active Power L1L2L3 [4]

Watchdog	disabled		
DC measurement	disabled		
Nom. Frequency power supply	50 Hz		
Peak value phase	L3		
Scaling energy consumption	0.05 kWh		
Storage interval energy [s]	60		
	L1	L2	L3
Autoreset min./max. value	enabled	enabled	enabled
User scaling	enabled	enabled	enabled
Current transformer ratio, HB	0	0	0
Current transformer ratio, LB	4	4	4
Peak value meas. Interval [HW]	10	10	10
Min./Max. reset interval [200ms]	1	1	1

Tabulka 3: Nastavení parametrů karty 750-494

Byte	Output data	Input data
0	0x00	0x44
1	0x02 (Status query)	0x02
2	0x00	0x40
3	0x09 (COL_ID)	0x09 (COL_ID_SEL)
4	0x06 (MET_ID_1)	0x06 (MET_ID_1_SEL)
5	0x03 (MET_ID_2)	0x03 (MET_ID_2_SEL)
6	0x4E (MET_ID_3)	0x4E (MET_ID_3_SEL)
7	0x17 (MET_ID_4)	0x17 (MET_ID_4_SEL)
8	---	Voltage, RMS L3-N [7:0]
9	---	Voltage, RMS L3-N [15:8]
10	---	Voltage, RMS L3-N [23:16]
11	---	Voltage, RMS L3-N [31:24]
12	---	Current, RMS L3 [7:0]
13	---	Current, RMS L3 [15:8]
14	---	Current, RMS L3 [23:16]
15	---	Current, RMS L3 [31:24]
16	---	Reactive energy L3 [7:0]
17	---	Reactive energy L3 [15:8]
18	---	Reactive energy L3 [23:16]
19	---	Reactive energy L3 [31:24]
20	---	cos phi L2 [7:0]
21	---	cos phi L2 [15:8]
22	---	cos phi L2 [23:16]
23	---	cos phi L2 [31:24]

Obrázek 3: Příklad nastavení výstupů a dat odeslaných na vstup PLC [5]

a vyšší rozlišení. Karta druhého uvedeného výrobce má potom výhodu v počtu odesílaných dat - najednou je z karty výrobce Siemens možno zjistit více údajů, než z karty WAGO.

2.3 Měřicí rozvaděč

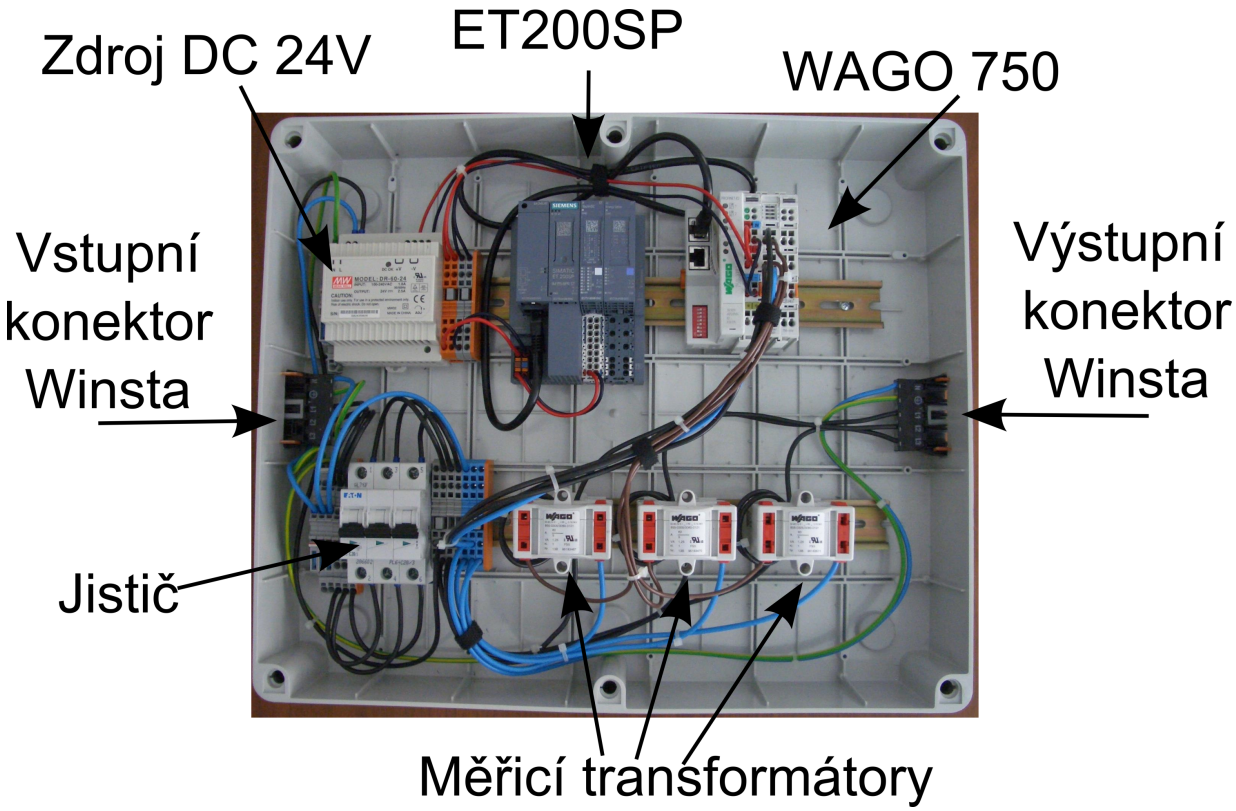
Pro potřeby měření byl vytvořen malý přenosný rozvaděč, který obsahuje měřicí transformátory a příslušné měřicí karty. V rozvaděči jsou umístěny pouze tři měřicí transformátory a pokud je potřeba měřit jinou kartou, je potřeba příslušně přepojit vodiče. Rozvaděč je vidět na Obrázku 4.

V rozvaděči jsou dvě lišty, z nichž jedna slouží pro silovou část a druhá pak přísluší napětí 24V DC. Pro možnost změny měřicí karty jsou vodiče jdoucí od měřicích transformátorů

a ze svorkovnice s fázemi delší.

Vstupní a výstupní konektor je typu Winsta od výrobce WAGO. Tyto konektory jsou certifikované pro proudy až 25A. Z důvodu ochrany měřicí aparatury, je v rozvaděči předřazen třífázový jistič 20A kategorie C. V rozvaděči je také umístěn zdroj 24V DC pro napájení měřicích karet a stanic, aby byl rozvaděč soběstačný..

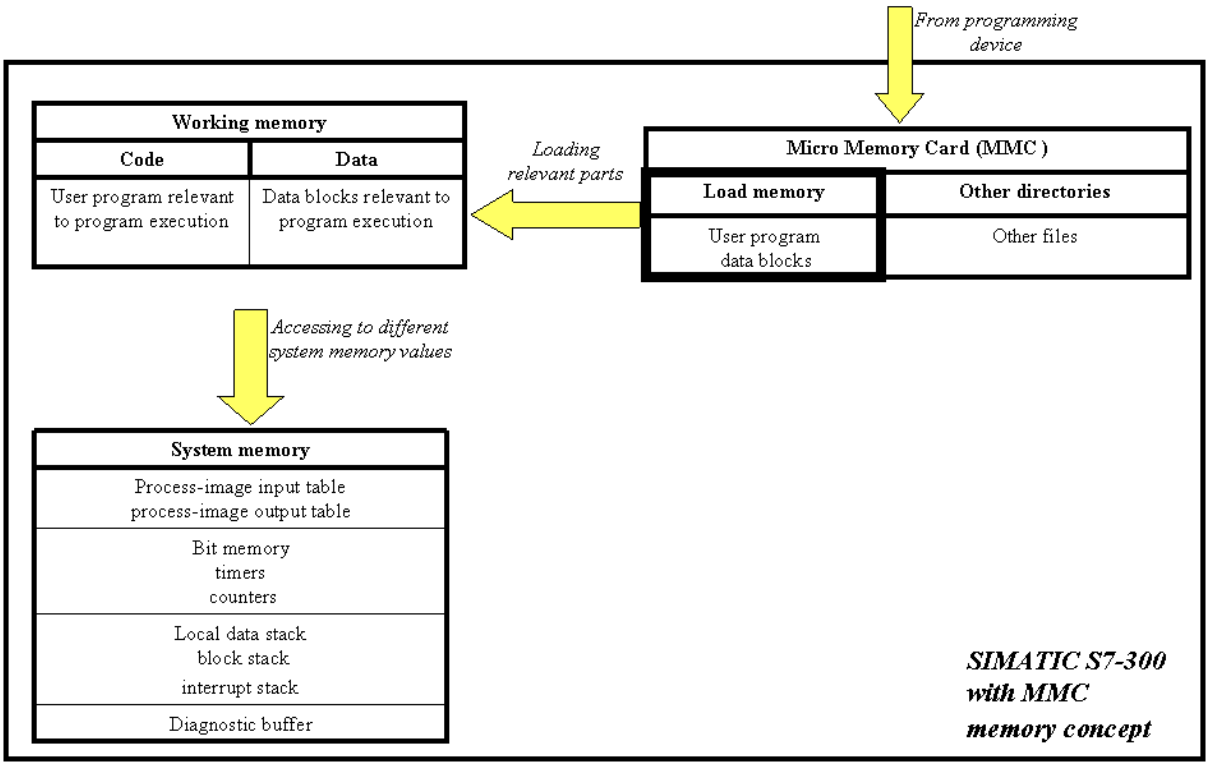
Měřicí transformátory mají převodový poměr 60:1, přičemž použité měřicí karty jsou nastavitelné pro rozsah 0-5A. Vzhledem k těmto parametrům a snaze využít celé pásmo převodníku v kartě, má primární cívka měřicího transformátoru tři závity.



Obrázek 4: Měřicí rozvaděč

2.4 Adresace a paměťové oblasti v PLC Siemens

Paměť, kterou využívá PLC je možné dělit na Working (Pracovní), System (Systémovou) a Load. Koncept paměti pro PLC SIMATIC S7-300 je vidět na Obrázku 5. Z tohoto obrázku je také zřejmé, k čemu je jaká paměť využita. V rámci programu ve STEP7 je možné adresovat proměnné především ze System memory, dále pak i Data z Working memory.



Obrázek 5: Příklad konfigurace stanice ET200SP [6]

2.4.1 Přímá adresace

PLC obsahuje celou řadu oblastí, které je možné adresovat. Konkrétní příklady adresace jsou uvedeny v Tabulce 4. Z pohledu velikosti adresovaných dat, je možno adresovat konkrétní BIT, BYTE, WORD a DWORD (double word).

Název	Relevantní český překlad	Příklad adresace
Inputs	Vstupy	I4.0, IB4, IW4, ID4
Outputs	Výstupy	Q12.0, QB12, QW12, QD12
Markers	Merkery	M8.0, MB8, MW8, MD8
Timers	Časovače	T2
Counters	Čítače	C9
Local	Lokální data	L16.0, LB16, LW16, LD16
Data Block	Datový blok	DB1.DBX7.0, DB1.DBB7, DB1.DBW7, DB1.DBD7

Tabulka 4: Paměťové oblasti SIMATIC S7-CPU a příklady adresace

2.4.2 Nepřímá adresace

Organizace paměti v PLC Siemens společně s programovacím prostředím STEP7 v5.5 umožňuje adresaci pomocí ukazatelů, které mají přesně definovanou strukturu. Tuto strukturu je možno ručně vytvořit, tak aby odkazovala na požadovanou proměnnou v paměti PLC. Tato data je pak možno kopírovat s využitím systémové funkce SFC20: BLKMOV, do které se zadá ukazatel zdroje a ukazatel cílového místa kopírování. Zásadní poznatek je, že funkce SFC20 pracuje jen s ukazateli načtenými z L oblasti (local stack).

Ve STEP7 existuje několik druhů ukazatelů [7]. Liší se především velikostí a tedy i tím, jaké možnosti adresace umožňují.

Základní typ ukazatele má velikost 4 bajty (”DWORD”), přičemž jeho struktura je uvedena v Tabulce 5. Zde první bajt obsahuje informaci o oblasti, ze které bude čteno (z) . Pro potřeby aplikace pro sběr dat byly využity hodnoty pro přístup do paměťových oblastí M,I,Q a DB. Spodní tři bity ukazatele obsahují informaci o příslušném bitu (x), zbytek potom pořadové číslo BYTE (y).

Dalším typem ukazatele je ukazatel do určitého datového bloku. Jeho struktura je vidět v Tabulce 6. Délka tohoto ukazatele je 6 bajtů, kdy spodní 4 bajty (byte n+2 až byte n+5) jsou stejné jako pro předchozí typ ukazatele a vrchní dva bajty (byte n a byte n+1) slouží

Byte n								Byte n+1								Byte n+2								Byte n+3							
1	0	0	0	0	z	z	z	0	0	0	0	0	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	x	x	x

Tabulka 5: Mezioblastní ukazatel

pro určení čísla datového bloku.

byte n	Číslo datového bloku
byte n+1	
byte n+2	Ukazatel oblasti
byte n+3	
byte n+4	
byte n+5	

Tabulka 6: Ukazatel DB

Nejvíce možností má ukazatel typu ANY, ten je dlouhý 10 bajtů, kdy jeho spodních 6 bajtů opět odpovídá popisu jednoduššího ukazatele (ukazatele pro datový blok). Jeho struktura je vidět v Tabulce 7. První bajt obsahuje hlavičku, která je pro ukazatel typu ANY definována jako 10H. Následuje číselný kód základní adresované jednotky: BYTE, WORD, DWORD,... Další slovo obsahuje počet (rozsah) adresované oblasti. Jedinou nevýhodou ukazatele ANY je nemožnost adresovat jednotlivé bity. Minimální adresovanou jednotkou je BYTE. Konkrétní příklady je možno najít v knize [7]. Pro názornost je v Tabulce 8 ukázáno konkrétní nastavení největšího ukazatele.

2.5 Sběr dat

Jak již bylo výše popsáno, změřené hodnoty výkonu je možné po konfiguraci karet vyčíst přímo z PLC z jeho vstupů. Pro řízení a rozhodování o aktuálním stavu linky, je vhodné využít i další data - informace o aktuálním stavu linky. Ty jsou uloženy v některých

Byte n	16#10
Byte n+1	Typ
Byte n+2	Počet jednotek 1/2
Byte n+3	Počet jednotek 2/2
Byte n+4	Číslo datového bloku 1/2
Byte n+5	Číslo datového bloku 2/2
Byte n+6	Ukazatel oblasti 1/4
Byte n+7	Ukazatel oblasti 2/4
Byte n+8	Ukazatel oblasti 3/4
Byte n+9	Ukazatel oblasti 4/4

Tabulka 7: Ukazatel ANY

Byte n	16#10
Byte n+1	16#04
Byte n+2	16#00
Byte n+3	16#01
Byte n+4	16#00
Byte n+5	16#00
Byte n+6	16#83
Byte n+7	16#00
Byte n+8	16#00
Byte n+9	16#20

Tabulka 8: Konkrétní hodnoty pro ukazatel P#MW 4 (čtení jednotky velikosti WORD, 1x, číslo DB: 0)

vnitřních proměnných PLC nebo se jedná přímo o vstupy a výstupy PLC.

Tato data je potřeba nějakým způsobem ukládat a tedy pro potřeby této práce bylo zvoleno odesílání dat do databáze.

2.5.1 Definice požadavků

Po zvážení možností, které by měl sběr dat splňovat, vzešly tyto požadavky:

- pro sběr procesních dat i pro sběr měřených dat použít stejný program
- rekonfigurovatelnost čtených adres bez nutnosti restartu PLC
- odeslání dat pomocí UDP paketu v domluveném formátu
- řízení zajištěno centrálou Siemens
- program napsaný v prostředí Siemens STEP7 v5.5

- automatické nastavení parametrů a inicializace odesílání dat po startu PLC

Vzhledem k těmto požadavkům je výsledný program postaven na adresování pomocí ukazatele typu "ANY"pro možnost rekonfigurace čtených adres a odesílána jsou data velikosti "WORD"- to kvůli ušetření místa ve zprávě a zároveň dostatečné velikosti pro odeslání změřené či stavové hodnoty, která má zpravidla rozsah "WORD". Pro komunikaci je pak využit protokol UDP, který podporují některé centrály či specializované karty.

2.5.2 Domluvený formát

Pro jednoduchost zpracování v PLC byl formát zprávy domluven takto: prvních 8 bajtů obsahuje časovou značku, přečtenou přímo z PLC, dále následuje 1-90 údajů, které mají následující formát: STRING(8) ':' WORD ';' . Dvojtečka a středník jsou přidány pro rozpoznání konce zprávy a začátek čtené hodnoty. STRING(8) obsahuje název proměnné, slovo potom obsahuje odesílanou hodnotu.

2.5.3 Komunikace pomocí S7-CPU

Pro odeslání změřených hodnot je použit protokol UDP. Správu ethernetového spojení a odeslání je možné realizovat několika způsoby. Jednou z možností je použití specializované komunikační karty. Druhá možnost je využít integrovaného rozhraní na PLC. Pro tuto funkci jsou dostupné bloky ze skupiny "Open Communication". Konkrétně se jedná o bloky FB65: TCON , FB66: TDISCON a FB67: TUSEND. K těmto blokům pro konfiguraci připojení a nastavení přísluší uživatelské struktury UDT65: TCON_PAR a UDT66: TADDR_PAR.

Konkrétní význam všech parametrů a možných nastavení je uveden v manuálu [8]. Nicméně nejdůležitější parametry a jejich nastavení pro tuto aplikaci jsou uvedeny v Tabulce 9. Mnohé problémy mohou plynout ze špatného nastavení parametru "local_device_id". Ten je pro každý typ centrály jiný a špatně nastavený parametr znemožní otevření komu-

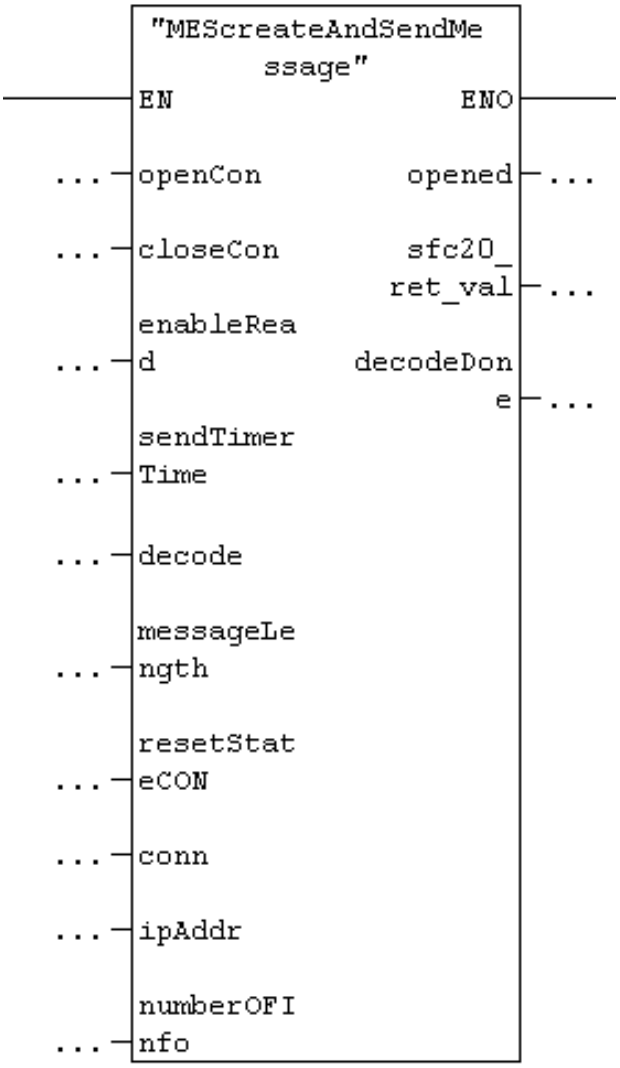
nikace. Jeho správnou hodnotu pro dané PLC je možné najít ve výše zmíněném manuálu. Pro PLC SIMATIC řady 315 je nastavení parametru 16#02, PLC řady 319 potom 16#03.

UDT65			UDT66	
block_length	W#16#40		rem_ip_addr[1]	B#16#C0
id	W#16#02		rem_ip_addr[2]	B#16#A8
connection_type	B#16#13		rem_ip_addr[3]	B#16#65
active_est	FALSE		rem_ip_addr[4]	B#16#2E
local_device_id	B#16#02		rem_port_nr[1]	B#16#07
local_tsap_id_len	B#16#02		rem_port_nr[2]	B#16#D0
rem_subnet_id_len	B#16#00			
rem_staddr_len	B#16#00			
rem_tsap_id_len	B#16#00			
next_staddr_len	B#16#00			
local_tsap_id[1]	B#16#07			
local_tsap_id[2]	B#16#D0			

Tabulka 9: Nastavení UDT65 a UDT66

2.5.4 Popis programu

Program je zabalen v jednom funkčním bloku, kdy po příchodu signálu pro dekódování jsou vytvořeny ukazatele v závislosti na zadaných parametrech z konfiguračního datového bloku (DB). Čísla bloků byla zvolena po prozkoumání parametrů pro programy pro průmyslovou linku, na které je program nasazen. Blok FB5630 s vysvětlením parametrů je na Obrázku 6 a v Tabulce 10.

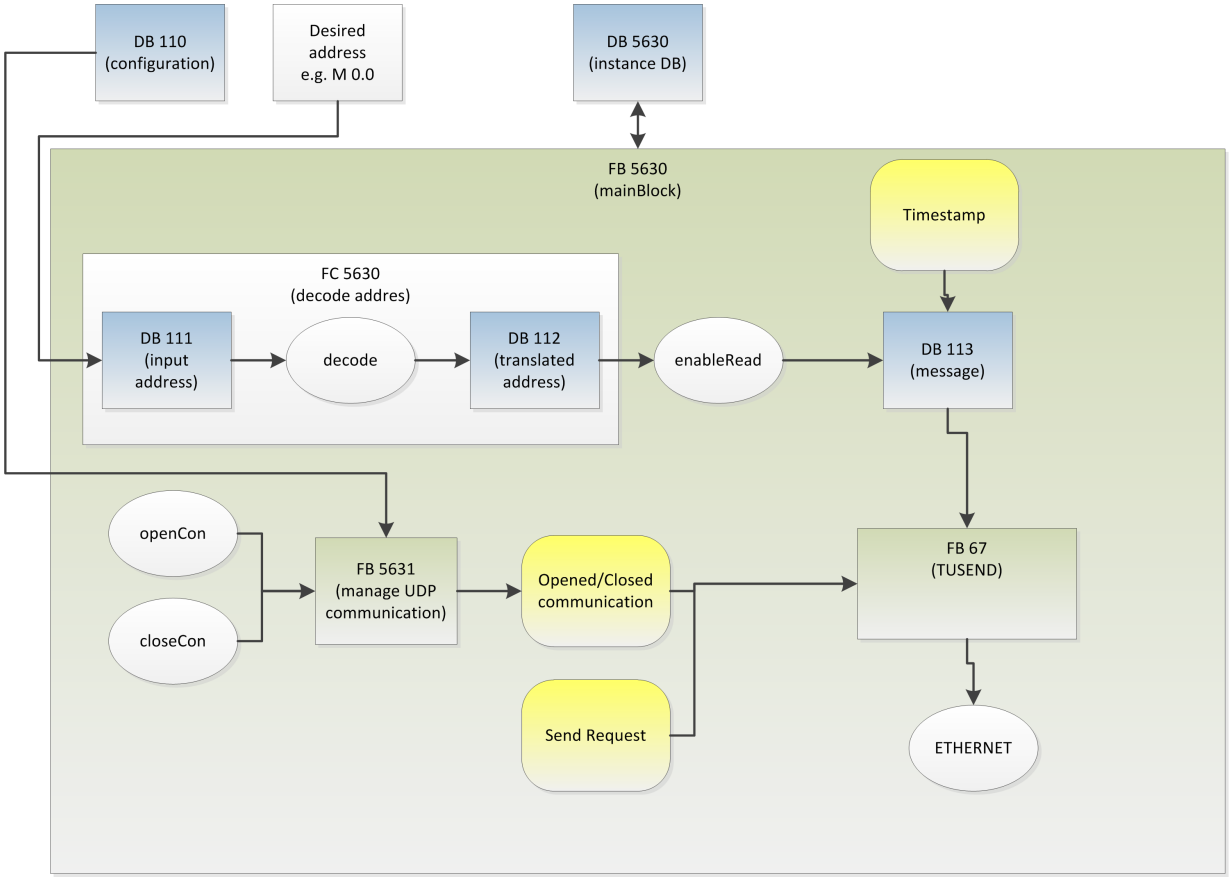


Obrázek 6: Blok FB5630

Parametr	**Deklarace**	**Data typ**	**Popis**
openCon	IN	BOOL	Na vzestupnou hranu inicializuje otevření spojení
closeCon	IN	BOOL	Na vzestupnou hranu inicializuje zavření spojení
enableRead	IN	BOOL	enableRead = 0: data nejsou čtena ani kopírována enableRead = 1: data jsou čtena pomocí ukazatelů a kopírována do odesílané zprávy
sendTimerTime	IN	TIME	Nastavení časovače pro odesílání zpráv. Perioda odesílání = 2x sendTimerTime
decode	IN	BOOL	Na vzestupnou hranu inicializuje dekódování adres z konfiguračního DB do adresního DB (vytvoření ukazatelů z konfiguračních dat)
mesageLength	IN	INT	Délka odesílané zprávy v bajtech (24-1448)
resetStateCON	IN	BOOL	Na vzestupnou hranu je vnitřní stav spojení resetován
conn	IN	ANY (UDT65)	struktura pro tvorbu spojení
ipAddr	IN	ANY (UDT66)	struktura pro nastavení komunikačního partnera
numberOFInfo	IN	INT	Počet dekódovaných a čtených dat
opened	OUT	BOOL	STATUS: 0: spojení uzavřeno 1: spojení otevřeno
sfc20_ret_val	OUT	INT	Zpětná hodnota bloku pro kopírování dat
decodeDone	OUT	BOOL	STATUS: 0: dekódování nebylo provedeno 1: dekódování provedeno

Tabulka 10: Parametry bloku FB5630

Program pro sběr dat je tedy složen z několika částí, kterým přísluší jednotlivé bloky. Symbolický průběh programu je uveden na Obrázku 7. S ohledem na parametry karet viz kapitola 2.2.3, byla zvolena základní vzorkovací frekvence 40ms.



Obrázek 7: Popis programu pro sběr dat

- **konfigurační bloky**
  - DB111 - požadované adresy k přečtení
  - DB110 - konfigurace komunikace, vstupů a výstupů hlavního FB
  - UDT5630 - konfigurační struktura (požadavek na oblast čtených dat, číslo BYTE, číslo DB, název proměnné)
- **překlad adresy**

  - FC5630 - z konfiguračního DB vytvoří překlad do ukazatelů, kopíruje název do zprávy
  - DB112 - datový blok obsahuje přeložené adresy
  - UDT5631 - struktura přeložené adresy

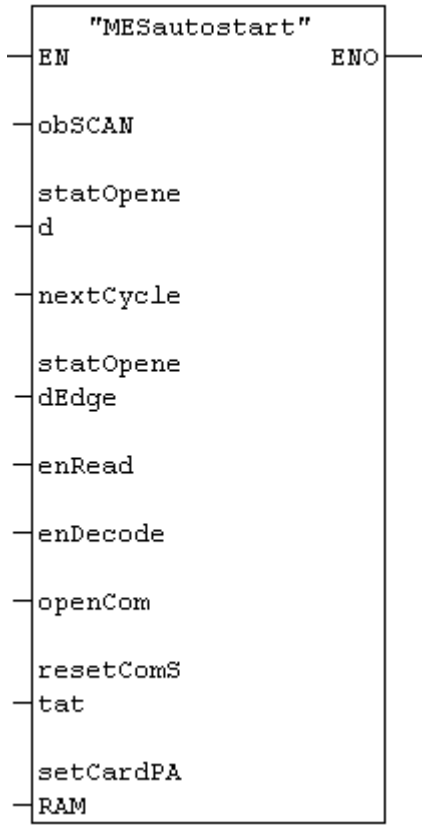
- **čtení dat z ukazatelů a odeslání zprávy**
  - FB5630 - hlavní blok programu
  - DB113 - obsahuje zprávu k odeslání
  - UDT5632 - struktura jedné zprávy

- **správa připojení**
  - FB 5631 - blok obsluhuje připojení a odpojení komunikace

Pro splnění požadavku na automatický start čtení a odesílání adres byla vytvořena funkce FC5631: MESautostart. Blok funkce FC5631 je uveden na Obrázku 8, popis parametrů potom v Tabulce 11.

Parametr	**Deklarace**	**Data typ**	**Popis**
obSCAN	IN	BYTE	Číslo SCANu PLC
statOpened	IN	BOOL	Příslušný signál bloku FB5630
nextCycle	INOUT	BOOL	Stavová proměnná funkce: následující cyklus
statOpenedEdge	INOUT	BOOL	Stavová proměnná funkce: detekce hrany
enRead	INOUT	BOOL	Příslušný signál bloku FB5630
enDecode	INOUT	BOOL	Příslušný signál bloku FB5630
openCom	INOUT	BOOL	Příslušný signál bloku FB5630
resetComStat	INOUT	BOOL	Příslušný signál bloku FB5630
setCardPARAM	INOUT	BOOL	Požadavek na nastavení parametrů měřicích karet

Tabulka 11: Parametry bloku FC5631



Obrázek 8: FC5631: automatický start sběru dat a jejich odesílání

2.5.5 Podrobný popis dekódování

Dekódování je provedeno pomocí funkce FC5630: MESdecodeAddrFC. Tato funkce je kompletně napsaná v STL (strojový kód pro PLC Siemens) a to z důvodu většího rozsahu možností. Především pak snazší programování cyklů a práce přímo s pevně adresovanými daty v paměti PLC.

Pro práci s datovými strukturami je možné využít dva adresní registry (AR1 a AR2). Dále je možné otevřít nejvýše dva datové bloky - jeden jako globální DB, druhý potom jako instanční DB. Potom je možné kopírovat s využitím adresních registrů mezi těmito bloky a využít k tomu ukazatele a posun o pevně zvolený ukazatel. Při práci s adresními registry je nutno dbát na jejich zálohování a obnovení. Konkrétně adresní registr AR2 je obecně

využíván programem napsaným ve STEP7 a bez řádného uložení a obnovení registru, může dojít k chybné adresaci a pádu programu.

Pro potřeby programu byly vytvořeny uživatelské struktury (UDT), které usnadňují konfiguraci a práci s ukazateli. Tyto struktury jsou uvedeny v Tabulce 12.

UDT5630: MESaddr		
Název	**Typ**	**Počáteční hodnota**
Oblast	CHAR	’ ’
nazev	STRING[8]	”
adresaByte	INT	0
adresaDB	INT	0
UDT5631: MEStransAddrUDT		
Název	**Typ**	**Počáteční hodnota**
transAddr	ARRAY[1..10] of BYTE	B#16#10, B#16#4, B#16#0, B#16#1
varName	STRING[8]	”
value	WORD	W#16#0
UDT5632: MESoneInfo		
Název	**Typ**	**Počáteční hodnota**
varName	STRING[8]	’ ’
dPoint	CHAR	’:’
value	WORD	W#16#0
strednik	CHAR	’;’

Tabulka 12: Uživatelské struktury

Konfigurační datový blok je vytvořen z jednotlivých konfigurovaných adres, co adresa to jedna struktura UDT5630. Proměnná Oblast může nabývat hodnot M,Q,I a D. Jedná

se o paměťové oblast PLC, D je potom zkratka pro DB (program porovnává jeden znak). Řetězec "nazev"potom obsahuje název, s jakým bude proměnná odeslána, "adresaByte" určuje adresu bajtu, "adresaDB"určuje číslo datového bloku.

Pro strukturu UDT5631 je důležité počáteční nastavení proměnné "transAddr". Toto pole v sobě nese ukazatel popsaný v kapitole 2.4.2. První 4 bajty ukazatele jsou pro všechny čtené hodnoty stejné a mohou být tedy nastaveny takto globálně.

Překlad adres probíhá ve dvou krocích:

1. překlad adres z konfiguračního DB do DB s přeloženými adresami (úprava konkrétního záznamu formátu UDT5631 na základě dat z konkrétního UDT5630)
2. kopírování názvu proměnné z konfiguračního DB do DB obsahujícího zprávu (kopírování proměnné "nazev"z konkrétní UDT5630 do proměnné "varName"v konkrétní UDT5632)

2.5.6 Aplikace DEPO

V předchozích kapitolách byl popsán sběr procesních dat a hodnot výkonu až po odeslání UDP paketu přes Ethernet na konfigurovanou adresu. Pro příjem dat vytvořil pan Ondřej Fiala aplikaci, která na cílovém zařízení tato data sbírá.

Aplikace je napsaná v jazyce RUBY a pro ukládání dat využívá databáze mongoDB. Data jsou ukládána surová tzn. příchozí paket je uložen jako jeden záznam. Výhodou tohoto přístupu je rychlost programu pro ukládání. Dále je možno celou databázi uložit (zálohovat) na externí médium a poté přenést z místa instalace na místo, kde budou data zpracována.

Export dat z databáze je opět proveden programem napsaným v jazyce RUBY na základě konfiguračního souboru, který obsahuje požadované proměnné, jejich převodní poměr a typ číselné reprezentace.

3 Měření výkonu průmyslového robotu

Tato kapitola je věnována popisu průmyslového robotu KUKA KR5 ARC a testování programu pro sběr dat a ovládání robota pomocí PLC. Dále je zde popsána súptřeba robotu v různých provozních stavech. Aby bylo možné měřit spotřebu v různých provozních stavech a různé druhy pohybu, bylo nutné naučit se robot ovládat z PLC a také prozkoumat různá nastavení a konfigurace samotného robotu.

Pro potřeby této práce byl využit robot KUKA KR5 ARC s příslušnou řídicí jednotkou. Tento robot je instalovaný ve strojovně na Karlově náměstí a je vidět na Obrázku 9.

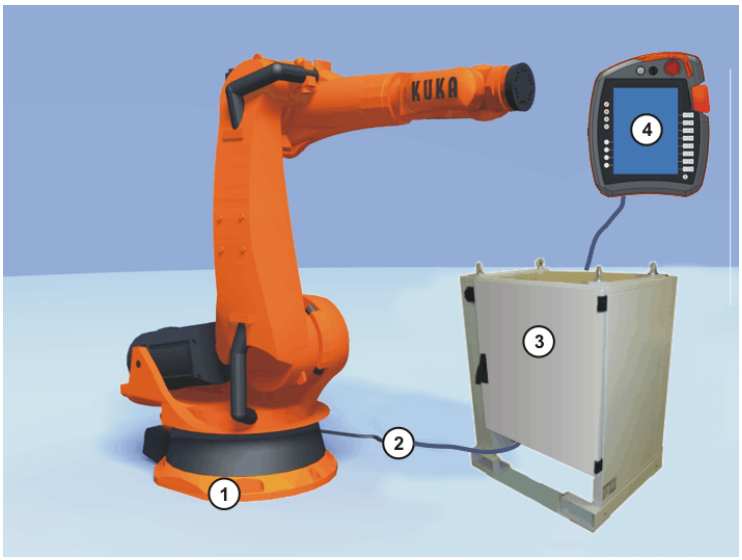
Obrázek 9: Robot KUKA KR5 ARC

Pro potřeby této práce tedy byla vytvořena konfigurace obsahující PLC, robot a měřicí

periferie. Tato konfigurace obsahuje také bezpečnostní část, bez které není možné robotem hýbat. Dále pak byly vytvořeny různé programy pro pohyb robotu.

3.1 Robot KUKA KR5 ARC

KUKA KR5 ARC je šestiosý manipulátor pro nízké zátěže. KR5 v označení robota značí řadu a mezní zátěž, která u tohoto robotu činí 5kg. ARC v názvu značí, že předpokládaná aplikace robotu je ve svařování. Na Obrázku 10 je vidět obvyklá konfigurace robotu. Ta se skládá z manipulátoru (v našem případě KR5 ARC), propojovacích kabelů, řídicí jednotky (KR C4) a ovládacího panelu (KCP). Řídicí jednotka potom disponuje připojením k síti PROFINET a celá sestava se z pohledu této sítě jeví jako jediné zařízení.



Obrázek 10: Příklad konfigurace industriálního robotu: 1-Manipulátor, 2-Propojovací kabely, 3-Robotický kontroler, 4-Ruční ovládací panel [9]

Celá sestava se konfiguruje pomocí programu WorkVisual. Jedná se o prostředí dodané přímo výrobcem robotu, ve kterém je možné nastavit parametry zařízení pro síť PROFINET, bezpečnostní konfiguraci robotu a konfigurovat systémové soubory. Takto vytvořenou konfiguraci je potom možno nahrát do KRC. Systémové soubory obsahují velké

množství parametrů, které je možné nastavit. Z pohledu spotřeby jsou zajímavé parametry, které určují délku doby nečinnosti, než robot sepne brzdy.

S KRC je možno komunikovat pomocí konfigurovaných bitových signálů, které jsou vyměňovány mezi řídicí jednotkou a PLC. Signály od PLC jsou:

- MOVE_ENABLE - povolení pohybu, je možné konfigurovat v souboru option.dat
- EXT_START - inicializace řízení z PLC
- PGNO - číslo programu v souboru cell.src
- PGNO_VALID - potvrzení čísla programu
- CONF_MES - kvitování chyb
- DRIVES_ON - zapnutí pohonů
- KONEC_PRACE - potvrzení dokončení úlohy

Signály odesílané z KRC do PLC potom:

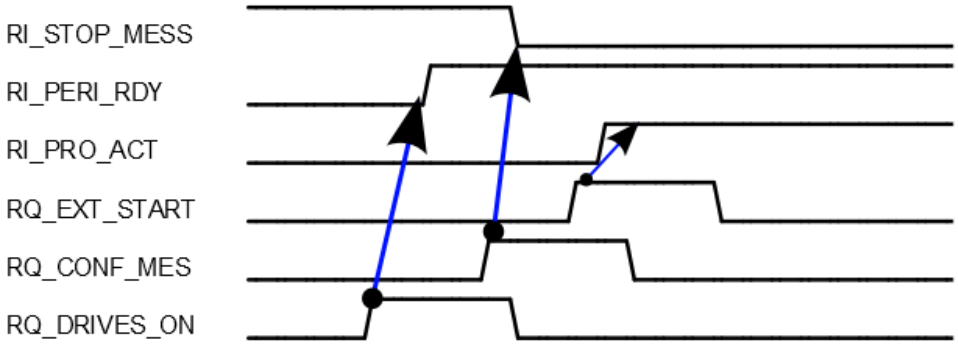
- STOP_MESS - indikátor přítomnosti chyby
- EXT - indikace režimu EXT
- PRO_ACT - indikace běžícího programu cell.src
- T1 - indikace režimu T1
- T2 - indikace režimu T2
- PGNO - zvolené číslo programu
- ON_PATH - robot je na definované trajektorii
- PERI_RDY - pohony jsou připraveny
- KONEC_PRACE_POZADAVEK - robot dokončil program, čeká na kvit konce práce

3.1.1 Spouštění robotu v režimu EXT

Robot může být provozován v několika pracovních režimech:

- T1: mód se sníženou rychlostí, ruční režim
- T2: mód bez omezení rychlosti, ruční režim
- AUT: řízení porgramu KRC bez nadřazeného PLC
- EXT: program řízen z PLC

Pro potřeby této práce je nutné mít robot spuštěn v režimu EXT. Přepínáni mezi jednotlivými režimy je provedeno pomocí panelu KCP a klíčku na něm umístěném. Přepnutí samotné nestačí k tomu, aby se robot začal pohybovat na povely PLC. Proto je potřeba provést sekvenci uvedenou na Obrázku 11. Pokud je signál "STOP_MESS"v logické 0, stačí provést pouze poslední uvedenou operaci: nastavit signál "EXT_START"do logické 1. Poté již běží program



Obrázek 11: Postup pro spuštění v režimu EXT, RQ výstupy PLC, RI vstupy PLC

3.1.2 Programovací jazyk KRL

Robot se v externím režimu pohybuje na základě programů napsaných v jazyce KRL (KUKA Robot Language). Tyto programy je možno vytvářet buď pomocí prostředí WorkVi-

sual, nebo přímo v ručním panelu KCP. Struktura programu a některé příkazy jsou popsány v manuálu [10]. Složitější příkazy a příklady jsou popsány v příručce [11]. Popis je sice pro starší verzi KRC, nicméně většina příkazů funguje stejně. Řídicí jednotka v laboratoři využívá verzi jazyka 8.3, v manuálu je popsána v8.2, nicméně většina příkazů zůstala stejná a tedy je starší manuál stále použitelný.

3.1.3 Submit interpreter

V robotickém kontroleru běží dvě paralelní úlohy

- Robot interpreter : Spouští robotický program.
- Submit interpreter : Spouští program SPS.SUB, který běží cyklicky. Pro jednodušší řízení může nahrazovat řízení nadřízeným PLC. Umožňuje zápisy na vstupy a výstupy mimo aktuální bežící robotický program.

Submit interpreter je spuštěn automaticky po zapnutí robotického kontroléru. Jeho spuštění, či vybrání jiného SPS.SUB je možné udělat pomocí panelu KCP. Další popis a nastavení je možné najít v manuálu [12]. V rámci Submit interpreteru je tedy možno přidávat různé další příkazy, které se cyklicky vykonávají. Pro potřeby této práce byl do souboru SPS.SUB přidán kód který umožnil odesílání různých vnitřních proměnných kontroleru do PLC.

3.1.4 Trajektorie robotu

Pro potřeby měření je poměrně zajímavá také informace o aktuální poloze robotu, konkrétně jeho koncového bodu. Informace o poloze vůči zvolené bázi obsahuje vnitřní proměnná kontoleru *$POS_ACT*. Konkrétně se jedná o strukturu, která obsahuje hodnoty aktuálního natočení kolem os X,Y a Z, a stejně tak i posunutí v těchto osách. Tyto proměnné jsou typu REAL (32 bitů).

Tyto proměnné mohou být odesílány do PLC na vstupy, které jsou v hardwarové konfiguraci mapované pro KRC. Aby bylo možné po bitových výstupech poslat 32bitovou

hodnotu, je potřeba nejdříve tyto výstupy spojit do skupiny "signálu". Toto je možné udělat v souboru "config.dat". Dále, aby byla data aktualizována, je potřeba v souboru "sps.sub"(který je spouštěn Submit interpretrem) přidat část kódu, která přiřazuje hodnotu zvolené rotace či posunu (například $POS_ACT.X) do příslušného "signálu".

Popis jednotlivých souborů, které byly v průběhu práce upravovány je uveden v kapitole 3.1.5.

3.1.5 Datové struktury robotu

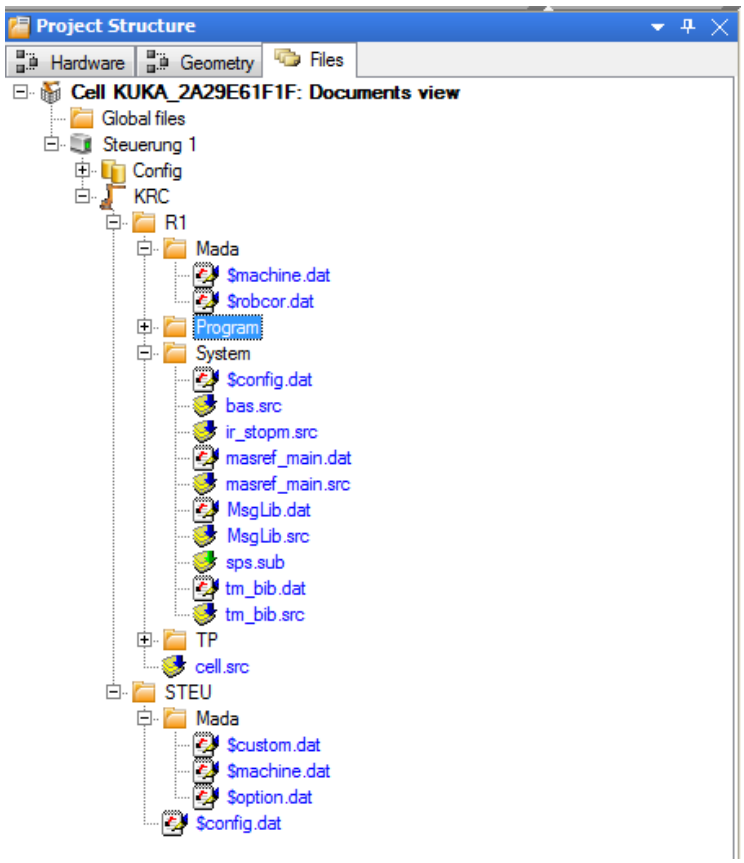
Na Obrázku 12 je vidět stromová struktura souborů v KRC, tak jak je zobrazena v programovacím prostředí WorkVisual. V Příloze E jsou uvedeny výpisy jednotlivých souborů v aktuální konfiguraci KRC testovacího robotu.

Soubory zásadní pro tuto práci jsou:

- KRC/R1/Mada/machine.dat : strojová data robotu, obsahují proměnnou pro nastavení doby nečinnosti, než robot přejde na brzdy
- KRC/R1/System/sps.sub :soubor spouštěný submit interpretrem, zde je potřeba přiřadit proměnné, pokud je žádáno jejich odesílání na výstup
- KRC/R1/cell.src :hlavní program pro spuštění v režimu EXT
- KRC/STEU/config.dat :zde je možno definovat proměnné a spojovat je do signálů
- KRC/STEU/Mada/option.dat :zde jsou různá uživatelská nastavení, například povolení kontroly "move enable"(kontrola sinálu z PLC pro uvolnění pohybu)

3.2 Popis profilu PROFIENERGY

Různých provozních stavů z pohledu spotřeby energie je u tohoto robotu možno docílit využitím profilu PROFIenergy.



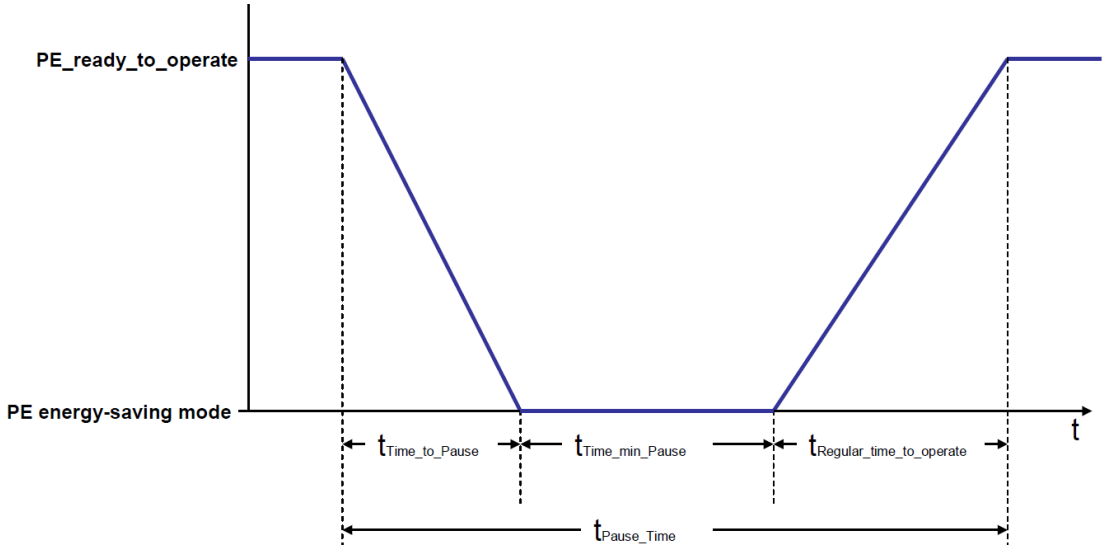
Obrázek 12: Struktura souborů v KRC

3.2.1 Obecný popis

Jedná se o úsporu energie, které je možno dosáhnout odpojením určitých periferií. Proto má každé zařízení, které podporuje PROFIenergy (PE), vytvořeno několik energetických profilů, ve kterých se může nacházet. Dále musí takovéto zařízení podporovat určité příkazy, které je možno posílat pomocí specializovaných programových bloků.

Celý koncept je tedy založen na jasně definovaném "rozhraní", které musí všechna kompatibilní zařízení podporovat a které je specifickým způsobem ovládáno z PLC. Ideou profilu je, aby bylo možno zařízení uspat a probudit v definovaný čas a případně bylo možno i měřit jejich aktuální spotřebu. Uspáním a probuzením je myšlen přechod mezi dvěma energetickými módy daného zařízení, z nichž je jeden výchozí a druhý úsporný. Těchto

energetických módů může mít zařízení více, minimálně ale dva. Na obrázku 13. je uveden výpočet pauzy a definice jednotlivých časů.



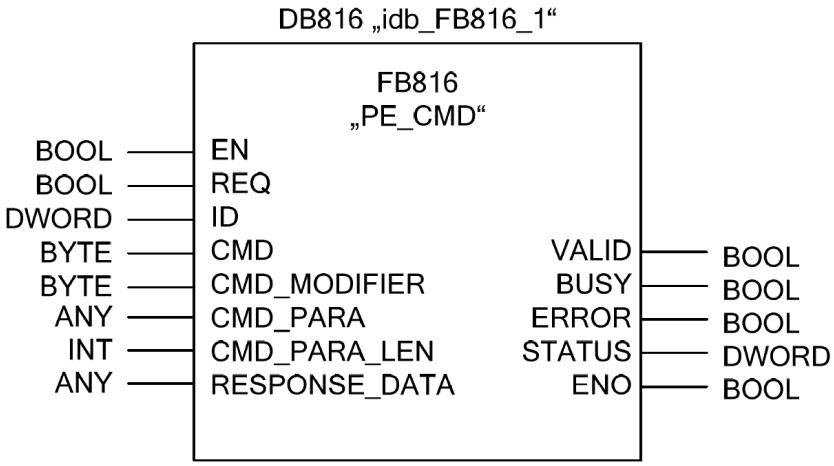
Obrázek 13: Výpočet doby pauzy podle [13]

Pro práci s PE je připraveno několik funkčních bloků pro STEP 7 (i pro TIA-Portal), které ovládají veškerou funkcionalitu jedná se o bloky: *FB815 "PE_START_END"*, *FB816 "PE_CMD"*, *FB53 "DS3_WRITE"*. Z nich je důležitý blok *FB816*, který umožňuje poslat většinu potřebných příkazů.

3.2.2 FB816 PE_START_END

Tento blok by se dal brát, jako hlavní blok pro spravování PE. Sdružuje v sobě funkce pro měření, informace o profilech zařízení a příkazy k přechodu mezi energetickými stavy. Na Obrázku 14 je vidět blok jako takový. Popis jednotlivých signálů je uveden v následujícím výčtu:

- REQ - odeslání příkazu
- ID - adresa zařízení na PROFINETu



Obrázek 14: Vstupy a výstupy FB816 [14]

- CMD - identifikátor příkazu (skupina příkazů)
- CMD_MODIFIER - upřesnění příkazu
- CMD_PARA - vstupní parametry
- CMD_PARA_LEN - délka vstupních parametrů
- RESPONSE_DATA - výstupní data
- VALID - indikace validity dat
- BUSY - indikace práce bloku
- ERROR - indikace chyby
- STATUS - stavová informace

3.2.3 Jednotlivé příkazy

Hexadecimální číslo uvedené před příkazem označuje vstupní parametr bloku FB816 (parametr CMD). Ne všechny tyto příkazy musí být zařízením podporovány. Mělo by ale podporovat minimálně prvních pět.

- 0x01 Start_Pause (vyvolá přechod do energeticky úsporného módu)
- 0x02 End_Pause (vyvolá přechod do energeticky méně úsporného módu)
- 0x03 Query_Modes (zaobaluje skupinu příkazů pro zjištění dostupných energetických módů a jejich vlastností)
- 0x04 PEM_Status (zjistí aktuální energetický stav zařízení a jeho vlastnosti)
- 0x05 PE_Identify (zjistí ze zařízení, jaké podporuje příkazy-paramtery pro volání bloku FB816)
- 0x06 Query_Version (indikuje implementovanou verzi PROFIenergy profilu)
- 0x10 Query_Measurement (pomocí tohoto příkazu je možno zjistit, jaká měření zařízení podporuje a také získat měřené hodnoty)
- 0x20 Info_Sleep_Mode_WOL (zjistí metodu probuzení ze speciálního úsporného módu)
- 0x21 Go_Sleep_Mode_WOL (vyvolá přechod do speciálního úsporného módu)

3.3 PROFIenergy na robotu KUKA

Řídicí jednotka robotu podporuje PROFIenergy, s tím, že je nutné v konfiguraci (v prostředí WorkVisual) této jednotky možnost PE povolit. Hlavním předpokladem, pro funkci příkazů k uspání, je přepnutí robota do externího režimu (*EXT*). V následujících podkapitolách je popsáno použití jednotlivých příkazů.

3.3.1 PE Identify (CMD=0x05)

Ke zjištění podporovaných funkcí bloku *FB816*, je možno poslat příkaz s parametry: *CMD=5, CMD_MOD=0, PARA_LEN=0*. Odezva na tento příkaz je vidět v červeném rámečku na obrázku 15. Z toho je tedy zřejmé, že robot podporuje 8 různých příkazů a to CMD=0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x20, 0x21. Obecně se jedná o obsluhu

zapínání a vypínání pauzy, funkcionalitu WOL(Wake On Lan) probuzení a identifikaci energetických módů.

kuka_safety_new_strojovna_vojtaWOL ▸ robot.f-cpu [CPU 315F-2 PN/DP] ▸ Watch and force tables ▸ Watch table_pe

	Name	Address	Display format	Monitor value	Modify value	Comment
1	"pe_id"	%MD118	DEC_unsigned	2032	2032	
2	"pe_startcmd"	%M110.3	Bool	FALSE	FALSE	
3	"pe_valid"	%M100.0	Bool	FALSE		
4	"pe_busy"	%M110.1	Bool	FALSE		
5	"pe_error"	%M110.2	Bool	FALSE		
6	"pe_status"	%MD114	Hex	16#0000_0000		
7	"pe_cmd_in"	%MB122	Hex	16#05	16#05	21
8	"pe_cmd_mod"	%MB123	Hex	16#00	16#00	0
9	"pe_cmd_para_len"	%MW124	DEC_unsigned	0	0	0
10	"e-srm-reference"	%I0.3	Bool	TRUE		
11	"PE_data".get_modes[1]	%DB7.DBB0	Hex	16#FF	16#FF	
12	"Tag_20"	%MW222	DEC_unsigned	20680		
13	"Tag_22"	%MW224	DEC_signed	0	3	
14		%DB7.DBD0	Time	T#-4H_39M_37S_21...	T#0MS	
15	"PE_start_end_data_change".pe_s_e_start	%DB9.DBX0.0	Bool	FALSE	FALSE	
16	"PE_start_end_data_change".pe_s_e_end_in	%DB9.DBX0.1	Bool	FALSE		
17	"PE_start_end_data_change".pe_s_e_id	%DB9.DBD2	DEC_unsigned	0	2032	
18	"PE_start_end_data_change".pe_s_e_pause_time	%DB9.DBD6	Time	T#5M	T#5M	
19						
20	"PE_data".decode.PE_Mode_ID	%DB7.DBB244	Hex	16#08		
21	"PE_data".decode.PE_Mode_Attributes	%DB7.DBB245	Hex	16#01		
22		%DB7.DBB246	Hex	16#02		
23		%DB7.DBB247	Hex	16#05		
24		%DB7.DBB248	Hex	16#03		
25		%DB7.DBB249	Hex	16#04		
26		%DB7.DBB250	Hex	16#06		
27		%DB7.DBB251	Hex	16#21		
28		%DB7.DBB252	Hex	16#20		

Obrázek 15: Příkaz PE:Identify

3.3.2 Query Modes (CMD=0x03)

Tato skupina příkazů zaobaluje funkce, pro zjištění možností různých energetických profilů. V případě poslání *CMD=3, CMD_MOD=1, PARA_LEN=0*, je odpovědí řídicí jednotky počet energeticky úsporných stavů a jejich hexadecimální identifikátory. A tedy robot podporuje tři různé módy: *0xFF, 0xFE, 0x01*.

V případě poslání *CMD=3, CMD_MOD=2, PARA_LEN=1, PARA="PE_ID"*, kde PE_ID je výše zmíněný identifikátor energetického módu, vrátí řídicí jednotka informace o spotřebě a dobách pauzy pro daný mód. Tato data pro všechny podporované módy jsou shrnuta v Tabulce 13, přičemž význam většiny časů je možné najít na Obrázku 13. Zde je ovšem dát pozor na definici času: čas t_{Pause_Time} je u KRC představován "Time_min_Pause"a $t_{Time_min_Pause}$ je "Time_min_length_of_stay".

Name	Drive_Bus_OFF	Hibernate	Ready_to_Operate
PE_Mode_ID	0x01	0xFE	0xFF
Time_min_Pause	25s	1m 50s	0
Time_to_Pause	5s	50s	0
Time_to_Operate	20s	50s	0
Time_min_length_of_stay	0	10s	0
Time_max_length_of_stay	-1ms	-1ms	-1ms
Mode_Power_Consumption	0,15 kW	0,03 kW	0,22 kW
Energy_Consumption_to_pause	0	0	0
Energy_Consumtion_to_operate	0	0	0

Tabulka 13: PE_Mode

3.3.3 Start_Pause (CMD=0x01)

Tento příkaz slouží k přechodu do úsporného režimu. V parametrech je potřeba zadat také délku pauzy. *CMD=1, CMD_MOD=0, PARA_LEN=4, PARA=time*

3.3.4 End_Pause (CMD=0x02)

Příkaz slouží k probuzení z úsporného módu, použití: *CMD=2, CMD_MOD=0, PARA_LEN=0, PARA=/*

3.3.5 Go_Sleep_Mode_WOL (CMD=0x21)

Příkaz převede robot do nejúspornějšího režimu- hibernace. Použití: *CMD=21, CMD_MOD=0, PARA_LEN=0, PARA=/*. Robot se uspí a reaguje pouze na příchod Wake

On Lan (WOL) paketu.

3.4 Wake On Lan

Jedná se o funkcionalitu známou ze síťových zařízení a počítačů, kdy je zařízení vypnuto a napájeno je co nejméně součástí včetně síťového adaptéru, který čeká na speciální probouzecí paket - Wake On Lan (WOL) paket. Tento paket je možné poslat pomocí určitých programů přímo z PC. WOL paket (též nazývaný Magic Packet) se skládá z hlavičky, která obsahuje 6 jedničkových bajtů: FF:FF:FF:FF:FF:FF , následuje šestnáctkrát zopakovaná MAC adresa probouzeného zařízení.

Takto vytvořený paket je možné odeslat i z PLC pomocí postupu a bloků pro komunikaci, které byly popsány v kapitole 2.5.3. Důležité je správně nastavit IP adresu. Paket musí být odeslán na takzvaný Broadcast (adresa platná pro všechna zařízení v dané síti - paket dorazí všem).

3.5 Měření a sběr dat

Robot je připojen k řídicí jednotce, která má možnosti úspory energie (splňuje profil PROFIenergy), navíc motory jsou řízeny servozesilovačem, který je obsažen v této jednotce. A tedy je vhodné měřit spotřebu robota a řídicí jednotky jako celku. Měření tedy bylo provedeno na přívodním kabelu řídicí jednotky.

Řídicí jednotka je připojena třífázovým vodičem s vodičem N a PE. Nicméně po prozkoumání dokumentace a zapojení bylo zjištěno, že vodič N není v řídicí jednotce k ničemu připojen. Měřen je tedy proud a napětí jednotlivých fází L1, L2 a L3.

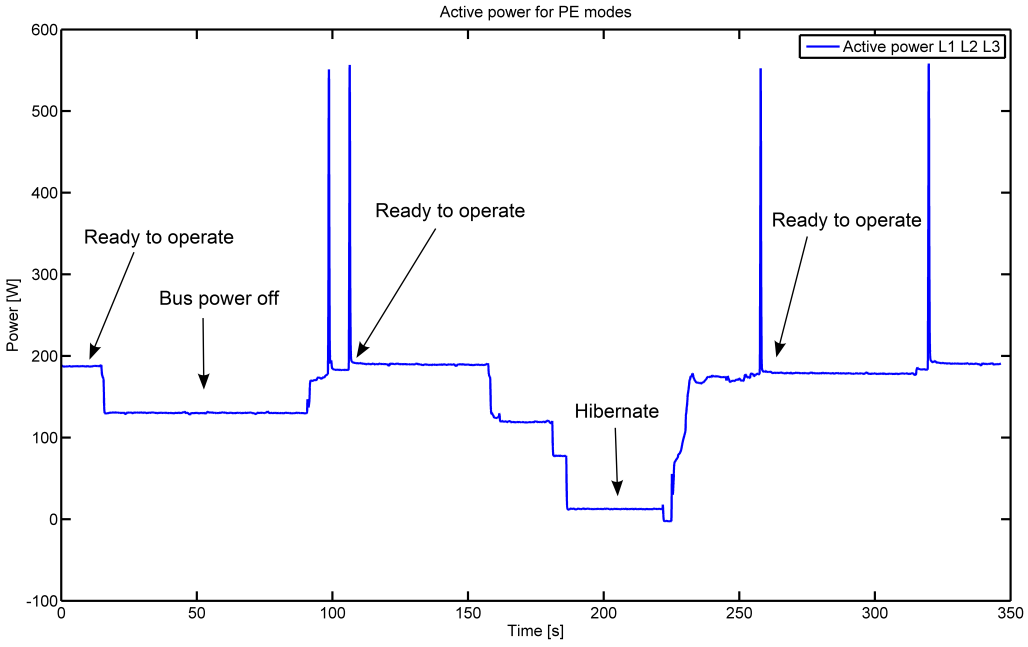
Data byla odesílána pomocí programu popsaného v kapitole 6.6.1. Pro měření byla využita karta WAGO pro její rychlejší vzorkování výkonu a protože pro analýzu spotřeby stačily 4 odesílané hodnoty.

3.6 Naměřená data

Z PLC bylo možné posílat jak data změřená měřicí kartou, tak i další stavové informace například hlášení z programu robotu, že dokončil či započal pohyb.

3.6.1 Provozní módy

Jak bylo výše zmíněno, testovaný robot podporuje tři energetické módy. Přechod mezi nimi a spotřeba v těchto stavech je vidět na Obrázku 16. Měřeno bylo pro postupné změny: Ready to operate - Bus power off - Ready to operate - Hibernate - Ready to operate.



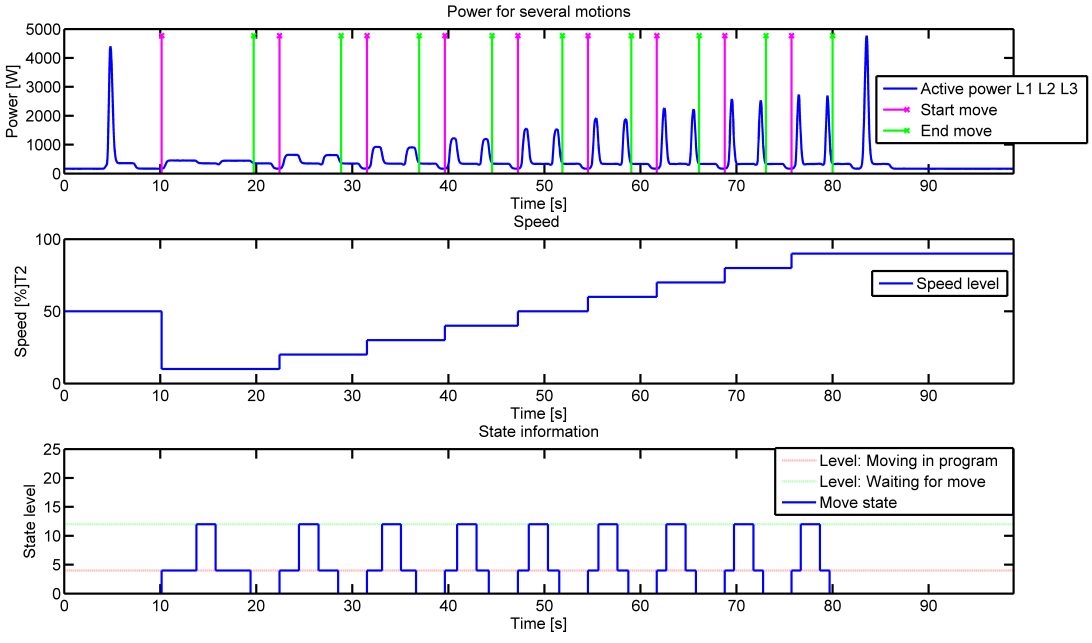
Obrázek 16: Různé provozní módy robotu

Výkonové špičky, které se objevují po probuzení souvisí s aktivací některé z vnitřních součástí KRC - ve chvíli, kdy dojde k výkonové špičce, je z jednotky slyšet bzučivý zvuk. Podle těchto špiček je možné vysledovat, kdy došlo k dokončení obnovení jednotky do módu Ready to operate - tomu odpovídá první špička po změně. Druhá špička potom odpovídá

ručnímu kvitování chyb a obnovení normálního provozu. Dále bylo zjištěno, že se tyto špičky snižují s tím, jak dlouho je jednotka v provozu.

Další zajímavost je, že při přechodu do stavu Hibernate prochází nejdříve jednotka podobným stavem, jako je Bus power off (BPO). To je možné vysvětlit tím, že jsou jednotlivé součásti KRC odepínány postupně. Mezi prvními jsou potom odepínány servozesilovače a tedy výkon je po nějakou dobu podobný jako v režimu BPO.

Ve srovnání s hodnotami, které uvádí výrobce (to je možné zjistit pomocí funkce Diagnostický monitor na KCP nebo pomocí PE příkazu) pro jednotlivé módy jsou změřené hodnoty nižší cca o 20-30W. To může být způsobeno periferiemi, které jsou připojené "za"kontrolérem robotu. V našem případě je zde pouze jediná stanice Siemens-ET200 a výrobce může počítat s více připojenými zařízeními. Roli může též hrát využití a stáří robotu, které je v laboratorních podmínkách odlišné od normálního provozu na průmyslové lince.



Obrázek 17: Měření pro různé rychlosti pohybu po stejné trajektorii

3.6.2 Různá rychlost pohybu

Z pohledu spotřeby může být zajímavé zjistit, jak je to se spotřebou energie při různě rychlém, stejném pohybu. Předpoklad je takový, že serva robotu mají vyšší spotřebu při rychlejším pohybu, nicméně při pomalejším pohybu může hrát velkou roli spotřeba řídicí jednotky. Pro ověření byl vytvořen program, který provedl stejný pohyb několikrát po sobě v různých rychlostech. K měřeným datům navíc byla odesílána hlášení z robotického programu o dokončení pohybu, aby bylo možné přesně detekovat čas, kdy pohyb probíhal.

Program je symbolicky popsaný Algoritmem 1, doby čekání jsou vloženy, aby bylo možné snáze detekovat ukončení pohybu.

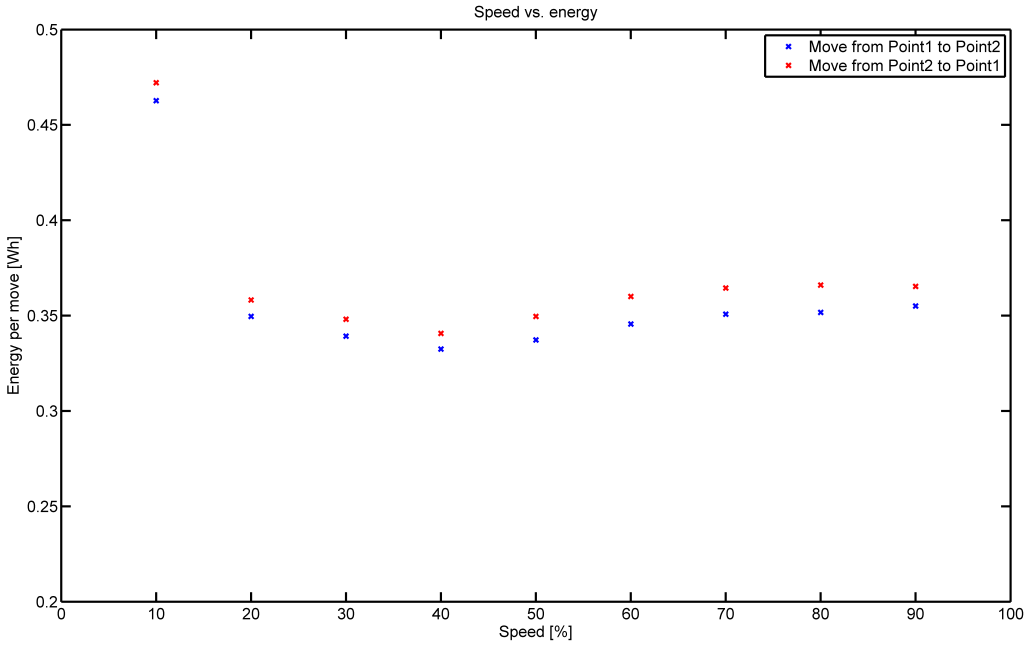
```
MOVE $HOME      // přesun do domácí pozice - příkaz je povinný;
MOVE Point2                    // příprava na pohyb;
WAIT 2s;
for k = 10; k ≤ 100 do
    WAIT 3s;
    SET OUT[1]=1              // hlášení stavu - v pohybu;
    MOVE Point1, velocity = k%T2;
    SET OUT[2]=1       // hlášení stavu - čekání na zpětný pohyb;
    WAIT 2s;
    SET OUT[2]=0                  // čekání ukončeno;
    MOVE Point2, velocity = k%T2;
    SET OUT[1]=0                   // pohyb dokončen;
    k = k + 10;
MOVE $HOME;
```

Algoritmus 1: Program pro testování spotřeby v různých rychlostech

Na Obrázku 17 je vidět měřená spotřeba pro stejnou trajektorii vykonávanou robotem při různých rychlostech pohybu. Měření bylo spuštěno po celou dobu programu. tj. od jeho navolení a spuštění až po jeho ukončení.

Na Obrázku 18 je potom vidět vypočtená spotřeba během jednotlivých pohybů. Důvod, proč spotřeba při pohybu z Bodu 1 do Bodu 2 je nižší než při pohybu opačném je způsobena tím, že na první zmíněný pohyb robot startoval ze zabrzděných motorů a na druhý nikoliv. To je způsobeno nastavením konstanty doby, než robot v průběhu programu přejde na brzdy. A tedy při delším čekání robot na brzdy přešel, při kratším nikoliv.

Dále je z Obrázku 18 patrné, že pro nejpomalejší pohyby hraje velkou roli nejspíše spotřeba řídicí jednotky a udržení robota na servech. Tento vliv postupně klesá. Na druhou stranu rychlejší pohyby spotřebovávají více energie a nejméně energie tedy robot spotřebuje při rychlostech blízkých 40% rychlosti režimu T2.



Obrázek 18: Spotřeba robotu pro jednotlivé pohyby

4 Rozšíření měření na reálnou linku

Program pro sběr dat byl rozšířen na reálnou linku, která je součástí větší komplexu celé svařovny daného podniku, konkrétně na linku Zadní příčník ve Škoda Auto v Mladé Boleslavi. Následující kapitola se zabývá úpravami programu, které musely být udělány, a popisem linky. Vzhledem k tomu, že se jedná o linku externí firmy, nebudou některé informace záměrně uvedeny.

4.1 Průmyslová linka

Reálná linka, na kterou byl program nasazen, má název Zadní příčník. Před postavením byla linka navržena v prostředí digitální továrny Tecnomatix

Jedná se o svařovací linku, kde se z několika dílů plechu a úchytů sedaček svařuje díl "zadní příčník"pro automobil. Do linky tedy vstupují plechové výlisky, úchyty, matky a další díly. Všechny tyto díly jsou ručně vkládány operátory buď do dopravníků, nebo na otočný stůl. Výstupem z linky je potom hotový svařený díl. Operátoři zajišťují zásobování linky a případné řešení problémů, operace svařování a lepení dílu jsou prováděny roboty.

Linka se skládá z několika stanovišť, na kterých jsou prováděny dílčí operace, jako je svařování kleštěmi připevněnými k robotu či svařování stacionárními kleštěmi nebo lepení. Některá z těchto stanovišť jsou napojena na dopravníky a otočný stůl. Předání výrobku mezi stanovišti je zajištěno pomocí stolů s upínkami, kam je nedokončený díl pokládán z prvního stanoviště a potom je přebírán druhým stanovištěm.

Z pohledu větších spotřebičů se linka skládá z:

- 6x robot KUKA KR270 R2700 včetně řídicí jednotky a panelu
- 1x otočný stůl
- 3x vstupní dopravník
- 1x výstupní dopravník

- několik brusek čepiček svařovacích kleští
- několik stacionárních svařovacích kleští
- lepicí stanoviště

4.1.1 Digitální továrna Tecnomatix - Process Simulate

Jedná se o vývojové prostředí firmy Siemens pro simulaci výrobních procesů. Prostředí se skládá z několika součástí, ve kterých je možné navrhnout linku z hlediska fyzické topologie, pohybů a operací jednotlivých zařízení. Dále je možné v tomto prostředí simulovat chod linky a odhadnout délku operací a ty pak exportovat do Ganttova diagramu.

4.1.2 Základní popis výroby dílu

Operátor založí základní díl do otočného stolu a k němu přiloží další součásti. Stůl se otočí dovnitř linky a roboty postupně přivaří a přilepí příslušné díly. V průběhu několika taktů linky (doby než díl projde celou linkou) je potřeba obrousit čepičky svařovacích kleští, nebo je úplně vyměnit. Některé z dílů jsou doplňovány do dopravníků a ty potom fungují jako zásobníky pro linku. Některé roboty na lince zastávají více funkcí - využívají několik nástrojů, obvykle uchycovací přípravky a svařovací kleště. A jsou tak schopny nejen svařovat, ale také přesouvat díl mezi jednotlivými stanovišti.

4.1.3 Rozbor proměnných linky

Linka je řízena jedním PLC Siemens SIMATIC S7-319-F 3PN/DP. To v sobě, jak je vidět z názvu, (písmeno F) již zahrnuje bezpečnostní funkce. S ostatními periferiemi je propojeno pomocí sítě PROFINET. Program pracuje řádově s tisíci vstupy, výstupy a merkery. Pro rozbor stavů linky byl k dispozici pouze program pro prostředí STEP7 v5.5 . V tomto prostředí je možno zobrazit takzvanou tabulku symbolů, která obsahuje přiřazení

symbolických jmen jednotlivým fyzickým adresám. Z těchto popisů se potom vycházelo při rozboru činnosti linky.

Z pohledu zjišťování stavu linky, je vhodné znát stav každého pracoviště. Protože jsou tato pracoviště úzce svázána s roboty a jejich operacemi, je výhodné sledovat stav programu robotu. Kontroléry robotů vyměňují s PLC několik signálů, které značí "konec práce", "číslo běžícího programu"či "procesní bod". Z těchto proměnných by mělo být dobře vidět, co robot právě dělá.

I stanoviště otočný stůl odesílá signál o dokončení práce. Dále některá stanoviště hlásí, že se svařuje, či další možné volby jako používaný nástroj.

Výše popsané signály společně rozumně popisují v jakém stavu je celá linka z pohledu operací zařízení. Dále je v proměnných možné najít velké množství signálů, kterými je detekováno správné založení dílů, přítomnost dílů, bezpečnostní stavy, blokování předávacích pozic jednotlivými stanovišťmi. U těchto signálů je složité jasně říci, co se děje, jen z tabulky symbolů a bylo by potřeba důkladného vysvětlení programátora.

4.1.4 Měřicí periferie

Bylo rozhodnuto, že měření výkonu bude zajišťovat samostatná jednotka, aby měření neovlivňovalo cyklus řídicího PLC linky a aby tak proběhly co nejmenší zásahy do tohoto funkčního celku. Pro měření tedy bylo využito PLC Siemens SIMATIC S7-319-F 3PN/DP společně se stanicí WAGO 750-377 a Siemens ET200SP. Kdy stanice výrobce WAGO obsahuje 5 měřicích karet a stanice ET200SP potom 8 měřicích karet a jednu kartu digitálních vstupů. Takto bylo vytvořeno 13 měřicích bodů, které měří:

- 6x robotický kontrolér
- 3x dopravník
- 2x frézy čepiček
- 1x otočný stůl

- 1x lepicí stanice

Data jsou odesílána do průmyslového PC výrobce B&R, konkrétně APC910.

Elektrickou instalaci měřicí aparatury zajistila Škoda Auto sama. Instalace měřicího PLC a jeho programu byla provedena na konci února roku 2014. Přímo na místě bylo řešeno několik problémů, které vyvstaly s prvotním nastavením uživatelských struktur UDT a nastavení datových bloků v offline režimu.

4.2 Úpravy programu pro sběr dat

Jak již bylo zmíněno v kapitole 2.5.4, bylo potřeba dodržet standard Škoda Auto, pro psaní programu pro PLC. Především se tento standard projevil v číslech bloků. Ve standardu je jasně dané, která čísla bloků co znamenají a která rozmezí je možné využít. Dále se standard projevil v názvech proměnných a zařízení v síti PROFINET a IP adresách jednotlivých zařízení.

4.2.1 PLC pro sběr procesních dat

Do programu řídicího PLC byl přidán program popsaný v kapitole 2.5.4. Dále byly v konfiguračním datovém bloku konkrétně nastaveny požadované čtené adresy. Posledním z úkonů bylo nastavení synchronizace času PLC s NTP serverem.

Důležité bylo nastavit řádně parametr pro komunikaci pomocí konkrétního PLC (viz kapitola 2.5.3).

4.2.2 PLC pro sběr dat z měření

Pro sběr dat bylo potřeba vytvořit kompletní projekt pro PLC včetně hardwarové části, která obsahuje příslušné stanice a měřicí karty.V programu byla přidána inicializace měřicích karet, aby odesílaly žádoucí hodnoty popsané v kapitole 2.2. I zde bylo potřeba

správně nastavit komunikační parametry a nakonfigurovat odesílané proměnné. Také byla nastavena synchronizace času s NTP serverem.

5 Analýza změřených dat

Tato kapitola se věnuje konkrétním naměřeným datům na průmyslové lince. Data byla měřena a ukládána od konce února roku 2014. Prvotní analýza dat byla prováděna nad daty z doby 22.2.2014 až 26.2.2014. Přesnější analýza potom probíhala nad daty od 26.2.2014 do 5.3.2014.

Jednou z provedených analýz byla analýza odstávek linky, protože převádět linku do úsporného režimu má smysl, pokud nastane právě takováto situace.

Reálná výrobní linka je nejprve navrhována v prostředí digitální továrny Tecnomatix a potom až postavena. V tomto prostředí je vytvořena celá linka včetně operací a výrobních prostředků. Z posloupnosti a délky operací je pak možné exportovat Ganttův diagram pro operace celé linky. Tento diagram jednotlivých operací konkrétních zařízení a jejich návazností byl dodán a na jeho základě bylo možné provádět další analýzy.

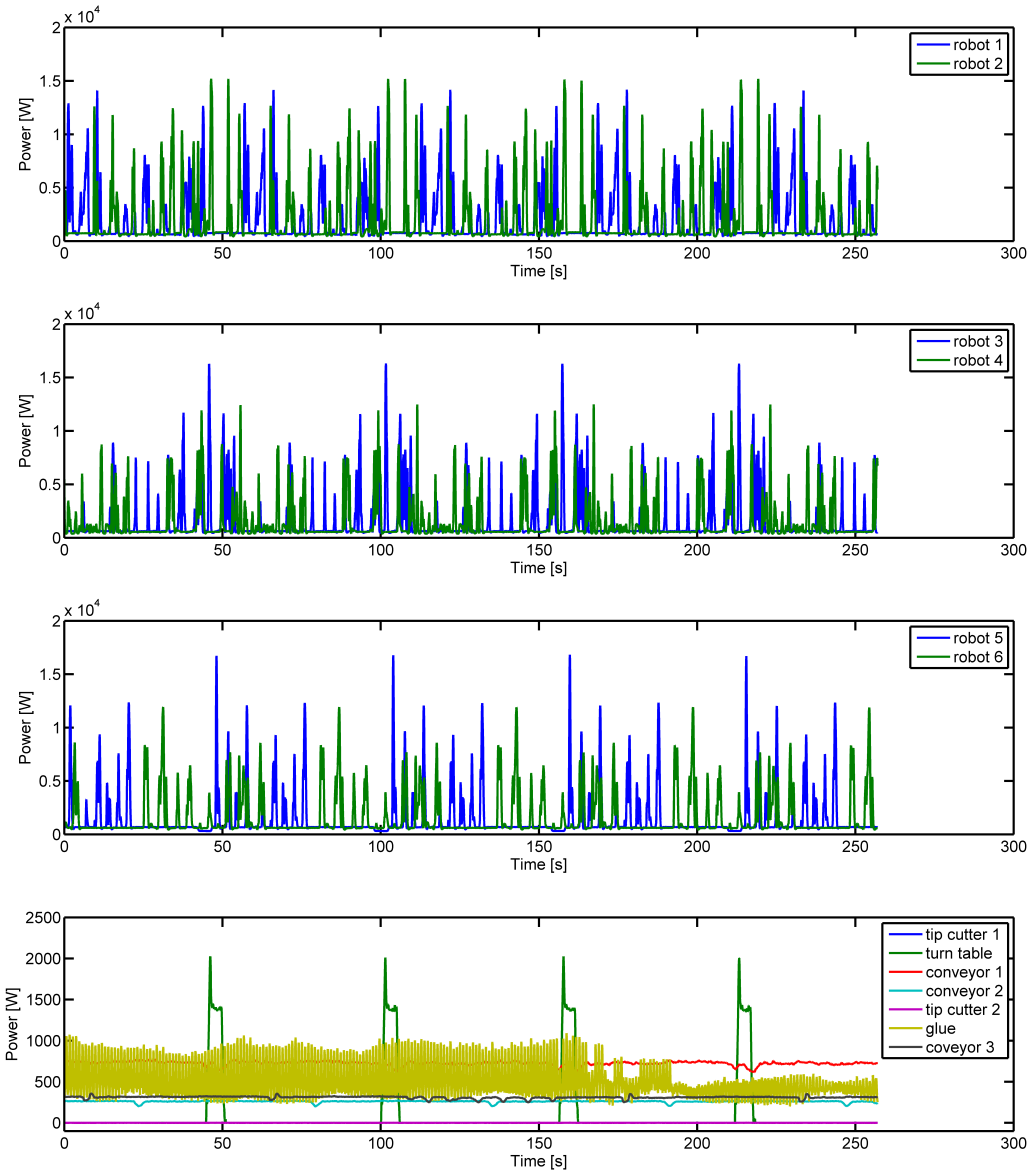
Exportovaná data získaná z týdenního měření jsou 6GB velká a je poměrně složité s nimi pracovat. Proto byla tato data rozdělena do menších částí po 2 000 000 vzorcích s tím, že spotřeba za jednotlivé fáze byla sečtena do souhrnné spotřeby za celé zařízení.

5.1 Prvotní analýza dat

5.1.1 Data z měřicího PLC

První data ukázala, že průběh výkonu otočného stolu při jeho otočení je lichoběžníkový a jinak nulový a tedy je podle něj možno dobře "zorientovat"(zarovnat časovou osu signálu, aby souhlasil se začátkem cyklu) Ganttův diagram.

Dále po srovnání pohybů s Ganttovým diagramem, který byl k lince dodán, je zřejmé, že linka zpracovává několik výrobků najednou. V lince jsou přítomny 4 kusy výrobku v různém stádiu dokončení. Tím je využití výrobních pracovišť v lince zvýšeno, nicméně to znesnadňuje analýzu operací jednotlivých měřených výrobních prostředků. Data pro jeden takt linky jsou vidět na Obrázku 19.



Obrázek 19: Data naměřená pro jeden takt linky

Z Obrázku 19. je dále patrné, že lepicí stanice má velmi proměnlivý signál. Frekvenční analýza v programu Matlab ukázala, že se jedná o signál s frekvencí 1Hz a zřejmě tedy nejde o rušení, ale o cyklické spínání nějaké části lepičky - nejspíše ohřevu lepidla.

5.1.2 Data z řídicího PLC

Ukázalo se, že některé zvolené proměnné se v průběhu programu nemění. Nicméně z ostatních, které se mění, je možné dobře vyčíst, co který robot dělá a v jakém stavu je jeho program.

Také se ukázalo, že je čas řídicího PLC navíc synchronizován pomocí programu s dalším časovým serverem. A tedy je čas tohoto PLC synchronizován dvěma různými metodami a navíc s různými NTP servery. Ukázalo se, že časové servery měly nastavené různé časové pásmo a tedy PLC bylo střídavě synchronizováno na jeden a pak na druhý čas. V již naměřených datech tedy bylo potřeba upravit časovou osu tak, aby odpovídala synchronizaci se stejným NTP serverem, jako je synchronizováno měřicí PLC.

5.2 Analýza odstávek

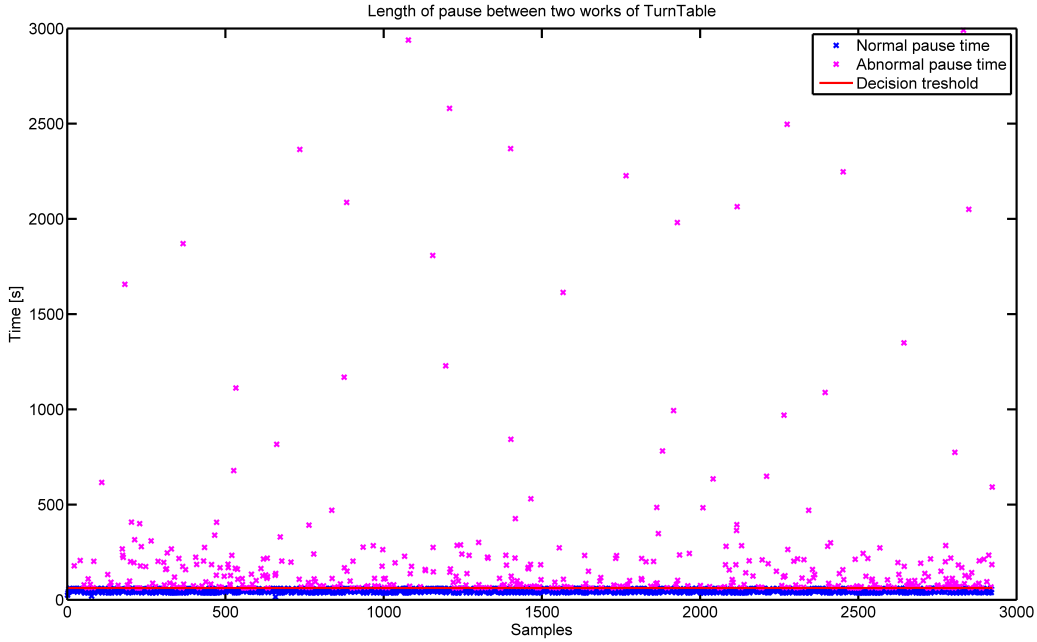
Odstávek (pauz) se může vyskytnout několik druhů. Plánované odstávky probíhají o víkendu a pak během směny. Neplánované potom mohou být způsobeny špatným založením dílů, nedoplněním dopravníků, frézováním a výměnou svařovacích čepiček, hardwarovými či softwarovými chybami výrobních prostředků.

5.2.1 Detekce pauz pomocí signálu stolu

Jak bylo výše zmíněno, výkon stolu má vhodný průběh pro snadnou detekci probíhajícího otáčení stolu. Pro detekci neobvyklé délky cyklu je možné využít informaci o době mezi dvěma pracemi (otočeními) otočného stolu, který je umístěn na začátku výrobního procesu. Tato doba bude při normálním cyklu mít hodnotu blízkou čtvrtině délky taktu linky

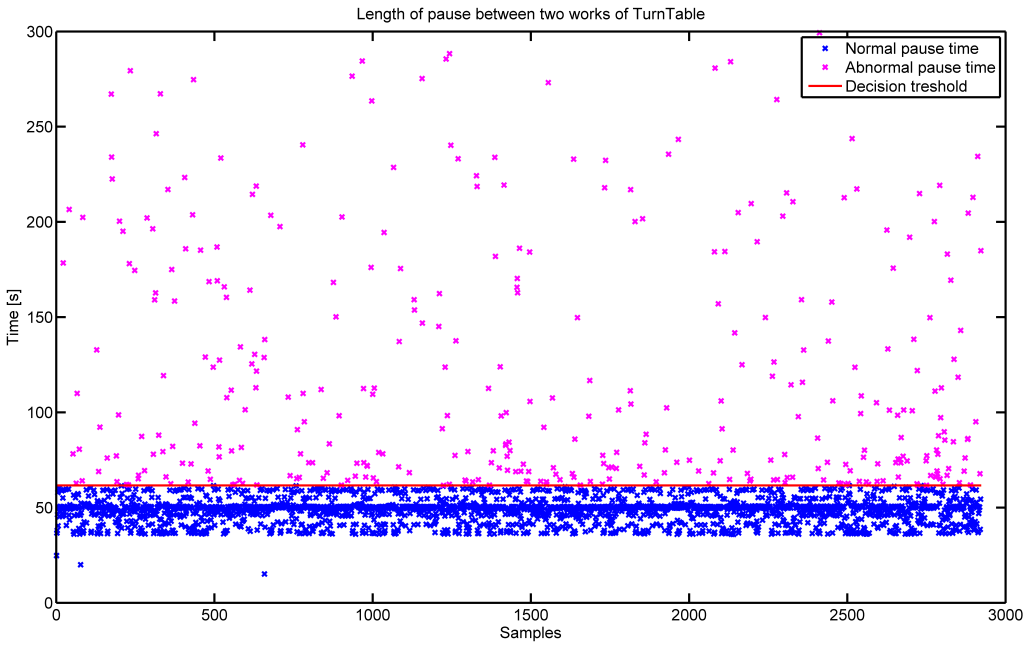
(v lince jsou najednou zpracovávány čtyři výrobky). Na Obrázku 20 jsou vidět doby mezi dvěma pracemi stolu v průběhu 67 hodin, kdy byla linka v provozu. Jsou zde vidět i velmi dlouhé doby mezi dvěma pracemi stolu. Na Obrázku 21 je potom vidět výřez předchozího obrázku, aby bylo možné detailněji určit rozhodovací práh, kdy jsou práce stolu děleny na normální cyklus a abnormálně dlouhý cyklus (pauza). U tohoto obrázku je posunuta časová osa tak, aby byly vidět cykly kratší, než 5 minut.

Rozhodovací práh byl určen tak, aby normální cykly pokrývaly hustou část množiny doby mezi dvěma pracemi otočeného stolu. Z této analýzy vychází, že abnormálních cyklů je cca 14% z celkového počtu cyklů. Vzhledem k délce abnormálních cyklů, je poměr celkové délky těchto cyklů a celkové doby vyšší (záměrně není uvedena přesná hodnota).



Obrázek 20: Délky pauz

Dále je vidět, že pauz kratších než 5 minut je poměrně hodně, nicméně delších pauz už je málo. Uspávat zařízení má smysl, pokud je pauza plánovaná či všechna zařízení nebudou po nějakou dobu nic dělat. V případě neplánovaného zdržení cyklu z důvodu frézování



Obrázek 21: Rozdělení délky pauz s určením prahu

čepiček či výměny čepiček, by uspané zařízení mohlo znemožnit tuto akci.

5.3 Analýza spotřeby v jednotlivých režimech

Ze zařízení umístěných na lince by měly pouze roboty disponovat nějakou možností úspory energie. V kapitole 5.4 je popsáno jakou. Další zařízení podle dostupných informací úsporou energie nepodporují.

Pro nasazení úspory energie bylo potřeba přesvědčit majitele linky, že má toto řešení smysl a tedy bylo potřeba analyzovat, jakou spotřebu mají roboty pokud nepracují a o kolik je tuto spotřebu možné snížit pomocí úspory energie. Spotřeba robotů na lince je popsána v Tabulce 14. Úsporné režimy testovaného robotu jsou potom uvedeny v kapitole 5.3.

	Spotřeba na servech [W]	Spotřeba na brzdách [W]
Robot 1	736	237
Robot 2	740	249
Robot 3	640	268
Robot 4	557	216
Robot 5	680	248
Robot 6	611	239

Tabulka 14: Spotřeba robotů v klidu

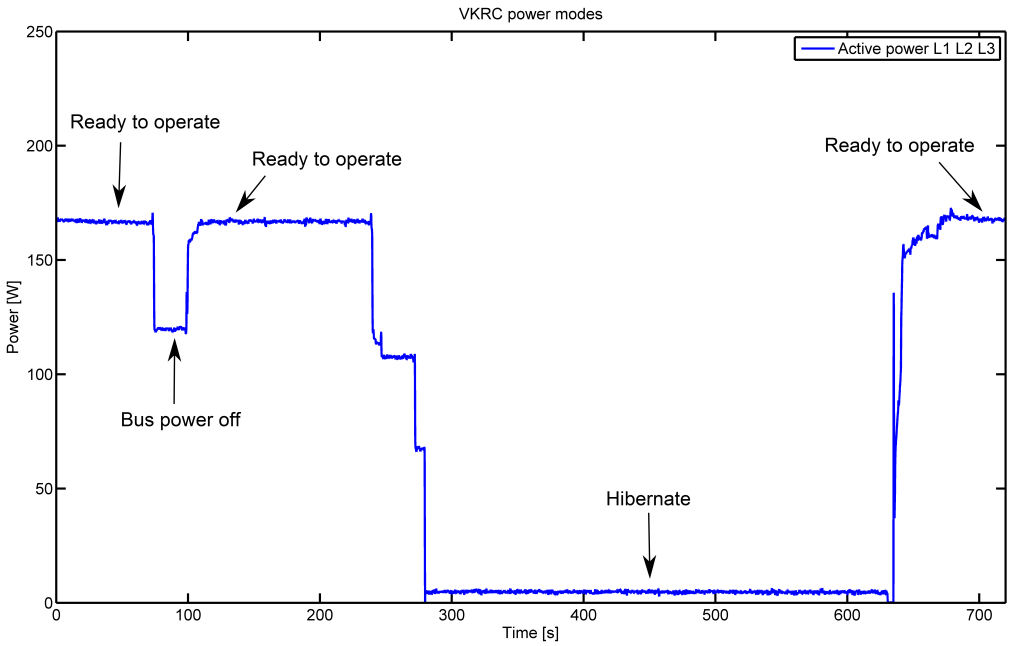
5.4 Měření robotu na dílně

Pro vyzkoušení ovládání a měření spotřeby různých režimů řídicí jednotky VKRC, bylo provedeno měření na dílně ve Škoda Auto v Mladé Boleslavi. Pro testování byl zpřístupněn robot podobný robotu na lince Zadní příčník. Konkrétně se jednalo o robot KUKA KR220.

V průběhu testování byly zjištěny odlišné vlastnosti řídicích jednotek VKRC a KRC popsané v kapitole 5.5. Navíc byl vyzkoušen program pro odeslání WOL paketu i ze stanice ET200: IM151-8F PN/DP CPU. Na Obrázku 22 je vidět spotřeba VKRC v různých provozních režimech. Ve srovnání s KRC je spotřeba o několik wattů nižší.

5.5 Popis VKRC

Roboty na lince se skládají, podobně jako testovací robot na Karlově náměstí, z robotu (na lince jde o robot KUKA KR270 R2700), řídicí jednotky, přívodních kabelů a ručního panelu. Řídicí jednotka je upravena pro potřeby automobilky Volkswagen a její označení je VKRC. Oproti jednotce KRC se liší především jiným softwarem. Ovládání programu je prováděno pomocí několika bitových vstupů, kdy pro spuštění programu je potřeba nastavit tyto bity všechny na 0 a poté navolit číslo programu. Oproti kvitování chyb u KRC, kdy je potřeba nastavit několik signálů postupně na logickou 1, je u VKRC vstup pro kvitování



Obrázek 22: Různé provozní módy VKRC

jenom jeden.

5.5.1 PROFIenergy na VKRC

Po testování robotu na dílně bylo zjištěno, že VKRC aktuálně nedisponuje plnou funkcionalitou PROFIenergy. VKRC ale určitou úsporu energie podporuje a to v režimech podobných KRC. V podstatě se zdá, že softwarový modul pro PROFIenergy, který je v KRC, jen rozšiřuje rozhraní úspory energie, kterou řídicí jednotka už disponuje.

VKRC má vyhrazeny dva vstupy pro úsporu energie. Jeden ovládá mód podobný "Bus power off"(BPO) a druhý vstup potom ovládá mód podobný "Hibernate"(HIB). Módy podporované KRC byly popsány v kapitole 3.3.

Přechod a setrvání v módu BPO je dán pozitivní hranou a setrváním příslušného vstupu v logické 1. Přechod zpět potom negativní hranou. A tedy pokud je signál v logické 0, je

robot v režimu "Ready to operate"(RTO). Přechod do režimu HIB je podobný, nicméně probuzení probíhá pomocí WOL paketu.

5.5.2 Rozhraní VKRC pro KRC

Protože VKRC se poněkud liší od KRC, byl vytvořen převodní blok v prostředí STEP7, aby bylo možné testovat uspávání v laboratoři tak, jak bude fungovat na průmyslové lince a tamním kontroléru. Převodní blok má tedy vstupy pro požadavek úspory energie HIB a BPO. Uvnitř sebe potom blok obsahuje logiku, která ovládá PROFIenergy příkazy a posílá je řídicí jednotce.

Dále byl vytvořen blok, který provede nutnou sekvenci ke kvitování chyb, na základě jediného signálu. To odpovídá výše zmíněnému chování VKRC.

5.6 Strategie uspávání

Jak již bylo popsáno v předchozích kapitolách, úsporou energie disponují roboty. A tedy tyto roboty je možné uspávat a probouzet za účelem snížení spotřeby celé linky.

5.6.1 Spuštění úspory energie

Na základě analýzy dat z linky a zjištění délky abnormálně dlouhých cyklů je možno popsat pravidla pro rozhodnutí o spuštění úspory energie těmito body, kde "X"značí počet minut, který je vhodné určit až z reálných testů na lince a případných požadavků obsluhy:

- Rozhodnutí o přepnutí zařízení do úsporného režimu je vhodné vytvořit na základě předpokládané doby odstávky. Pokud je doba odstávky menší než X minut nebo je potřeba v průběhu odstávky hýbat robotem, nemá smysl uspávat zařízení.
- Pokud se jedná o odstávku delší než X minut a není potřeba robotem hýbat, je vhodné ho převést do režimu Bus power off (BPO).

- V případě dlouhé odstávky (několik hodin), například víkendové odstávky, má smysl přepnout všechna zařízení do režimu Hibernate (HIB)

V případě odstávky kratší než X minut je pravděpodobné, že se jedná o neplánovanou odstávku, kdy jsou řešeny problémy se svařovacími čepičkami a odpojení napájení serv by mohlo způsobit více problémů, než by byl zisk z úspory energie.

5.6.2 Postup uspání a probuzení

V kapitole 5.5.1 bylo popsáno, jak funguje úspora energie na VKRC. Po testech na robotu na Karlově náměstí, vznikla tato doporučení :

- doba pro uspání musí být dostatečně dlouhá, aby bylo možné robot uspat i probudit
- uspávat robot je možné i v průběhu programu, nicméně se nesmí v té chvíli pohybovat - pohyb vede k chybě
- při přechodu a obnovení z režimu BPO i HIB je možné pokračovat v původně běžícím robotickém programu.
- po probuzení je potřeba kvitovat chybu ("Odpojení od sběrnice napájení"u režimu BPO), u režimu HIB je navíc potřeba zajistit bezpečnost (robot obsahuje bezpečnostní funkci, kterou je po obnovení připojení s PLC řádně zprovoznit)

Na základě těchto doporučení, znalostí o cyklech linky a testech jak na KRC tak VKRC je možné popsat postup probuzení a uspání:

- **Úsporný režim Bus power off**
  1. zhodnocení situace, jako vhodné pro aktivaci pauzy BPO
  2. dokončení programu nebo přechod do čekací smyčky v rámci programu(robot se nehýbe)

3. nastavení příslušného vstupu pro režim BPO do logické 1 a inicializovat tak přechod RTO-BPO
4. robot je v úsporném režimu
5. požadavek na obnovení provozu
6. nastavení příslušného vstupu pro režim BPO do logické 0 a inicializovat tak přechod BPO-RTO
7. kvitování chyb, po čase definovaném jako doba pro probuzení z BPO
8. obnovení původního provozu

- **Úsporný režim Hibernate**

1. zhodnocení situace, jako vhodné pro aktivaci pauzy HIB
2. dokončení programu nebo přechod do čekací smyčky v rámci programu(robot se nehýbe)
3. nastavení příslušného vstupu pro režim HIB do logické 1 a inicializovat tak přechod RTO-HIB
4. zadání požadavku na odpojení PROFINET zařízení od PLC
5. robot je v úsporném režimu
6. požadavek na obnovení provozu
7. nastavení příslušného vstupu pro režim HIB do logické 0
8. odeslání WOL paket z PLC a inicializovat tak přechod HIB-RTO
9. zadání požadavku na opětovné připojení PROFINET zařízení k PLC
10. obnovení bezpečnosti linky
11. kvitování chyb, po čase definovaném jako doba pro probuzení z HIB
12. obnovení původního provozu

5.7 Ověření funkčnosti

Funkčnost programu byla nejprve testována na robotu na Karlově náměstí. Zde se ukázalo, že program pro uspání a probuzení funguje, jak bylo předpokládáno včetně kvitování chyb. Robot bylo možné uspat v průběhu programu, kdy byl program zastaven a čekal na signál od PLC. Po probuzení robotu a kvitování chyb, pokračoval robot v programu tam, kde skončil.

Ověření na reálné lince nebylo možné provést, protože v době psaní této práce, nebyly dokončeny všechny přípravy na reálné lince a nebylo tak vydáno povolení k testování programu a úpravě programu řídicího PLC linky. Původní odhady byly tedy příliš optimistické, ale alespoň bylo testováno na dostupných robotech tj. na robotu v dílně ve Škoda Auto a na robotu na Karlově náměstí.

6 Softwarový modul

Z hlediska spotřeby jsou zajímavé stavy "linka v provozu"a "linka v odstávce". Pokud je linka v odstávce, má smysl spouštět úsporu energie. Pro ověření funkce modulu byl zvolen Matlab, z důvodu jednoduché práce s maticemi naměřených dat a jejich vizualizace.

Dále vzhledem k tomu, že je objem dat odesílaných měřicím PLC velký (cca 4GB za den), je složité dlouhodobě měřit a ukládat spotřebu linky včetně procesních dat. Navíc není nutné zaznamenávat detailně spotřebu běžných operací. Smysl má shromažďovat celkovou spotřebovanou energii a detailní informace o abnormálních cyklech či operacích. Pokud by se podařilo takto detailně detekovat operace, mohl by sběr dat běžet mnohem déle a data by se potom snáze načítala do programů.

Operace je možné klasifikovat za pomoci nějakých, pro ně charakteristických veličin. V případě, kdy je měřen výkon zařízení, jsou těmito parametry délka operace a energie spotřebovaná v průběhu této operace.

Po naprogramování několika klasifikátorů a vyzkoušení různých metod, se zdá, že klasifikace stavů linky by sama o sobě mohla vydat na téma samostatné diplomové práce.

6.1 Zadání a výstup modulů

Vstupy:

- Ganttův diagram operací pro každé zařízení
- Změřená data výkonu
- Klasifikace operace v závislosti na délce operace a energie spotřebované touto operací (případně práh pro rozhodnutí)
- Normální doba mezi dvěma cykly zařízení

Výstupy pro klasifikaci stavů a klasifikaci operací robotů:

A) Klasifikace stavu linky - pracuje, pauza, nečinnost

B) Klasifikace operace robotů - konkrétní operace, abnormální operace

6.2 Implementace jednoduššího modulu (A) pro roboty

Robot se aktuálně může nacházet v několika stavech a to :

- pracuje
- nečinnost
- stojí na brzdách

V budoucnu mohou přibýt stavy různých energetických úspor. V každém z výše uvedených stavů má robot různý výkon. Při pohybu je výkon největší, při stání na brzdách nejmenší. Pro stání na motorech a brzdách byl výkon změřen a popsán v kapitole 5.3 v Tabulce 14. Tyto hodnoty je možné použít jako rozhodovací prahy pro určení jednotlivých stavů robotu. Takto se vytvoří tři výkonové oblasti, ve kterých se robot může nacházet. Podle oblasti, do které spadá aktuální výkon robotu, je potom klasifikován jeho aktuální stav.

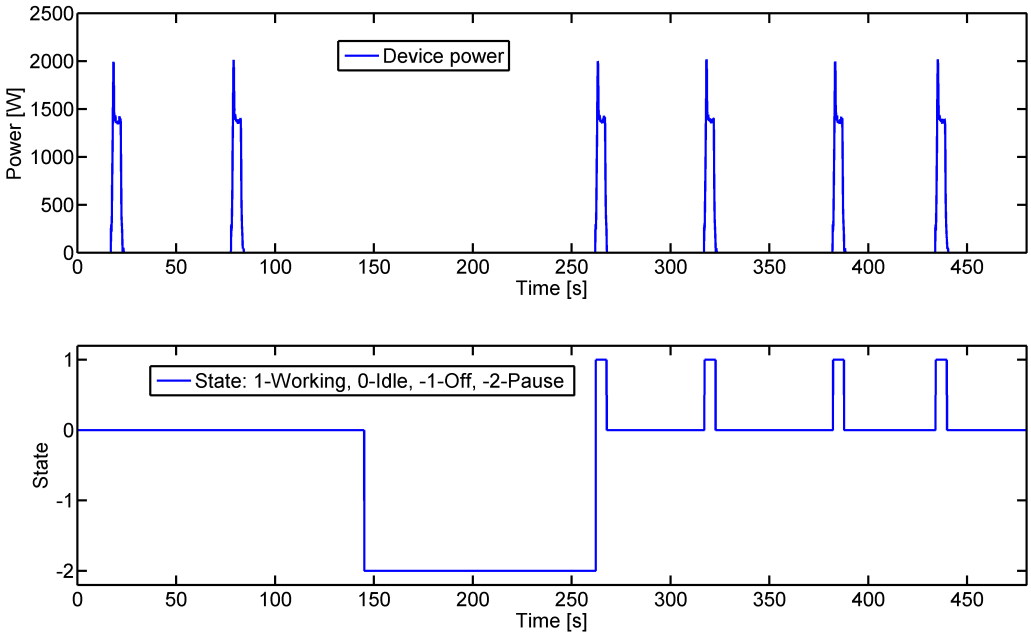
6.3 Implementace modulu pro další zařízení

Zbylá zařízení (jiná než roboty) provádějí jednodušší operace. Otočný stůl se pouze jednou za cyklus otočí, dopravníky se hýbou, pokud je potřeba posunout další díly do linky. Frézování čepiček probíhá po několika cyklech.

Tato zařízení tedy buď dělají jednu určenou operaci, nebo čekají, než tuto operaci budou moci dělat, nebo jsou vypnutá. Do těchto tří množin je možné operace klasifikovat v závislosti přechodu signálu výkonu přes zvolené prahy. Při činnosti zařízení je spotřeba vyšší. Některá zařízení mají spotřebu blízkou nule či nulovou, i když jsou pouze v nečinnosti a mohou se tedy nacházet ve dvou stavech: ”práce”a ”vypnuto”. V závislosti na době

otočení stolu je dále možné pro celou linku detekovat, jestli došlo k odstávce. Detekce operací stolu je vidět na Obrázku 23. První dvě operace nejsou klasifikovány, protože klasifikace byla spuštěna nad oknem délky 100s. A klasifikátor klasifikuje aktuálně probíhající operaci tzn. prvních 100s vzorků není klasifikováno.

Od otočení stolu se odvíjí, jestli byl do linky vložen nový díl. Pokud nebyl díl vložen, stůl se neotočí a do linky nepřibývá tak další díl ke zpracování. Nějakou dobu potom je ještě možné, že některé roboty a pracoviště dodělávají díly, které už byly rozpracované. Ve chvíli, kdy jsou všechna zařízení v klidu, nastává pauza.



Obrázek 23: Klasifikace operací otočného stolu

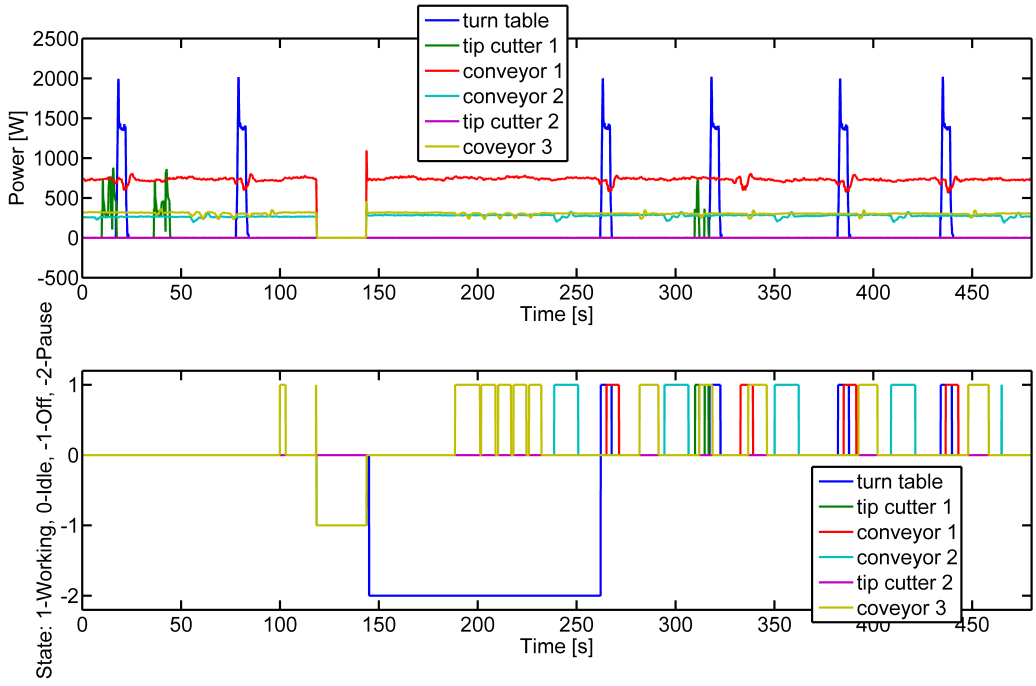
6.3.1 Postup klasifikace pro dopravníky

Protože dopravníky mají stálý odběr v řádu stovek wattů a pohyb nejprve tuto spotřebu na chvíli sníží a poté zvýší, je potřeba detekovat je jinak, než otočný stůl a frézy čepiček. Detekce využívá velikosti změny signálu. Pokud je změna správně orientovaná (záporná)

a vyšší než určitý práh, je detekován začátek pohybu dopravníku. Nicméně tato změna nesmí být příliš velká - potom se jedná o vypnutí dopravníku.

6.3.2 Postup klasifikace pro lepicí stanici

Jak bylo výše zmíněno, obsahuje měřený výkon lepicí stanice složku o frekvenci 1Hz s amplitudou v řádu stovek wattů. Po použití filtru se zdálo, že v signálu není nic dalšího a že je měřeno ohřívání lepidla. Pokud se v signálu vykytuje něco dalšího, má to tak malou amplitudu, že je to těžko rozlišitelné od ohřevu. A tedy jsou u lepičky detekovány dva stavy - práce a vypnuto.



Obrázek 24: Detekce stavu dopravníků, fréz a otočného stolu

6.4 Ověření funkce klasifikace stavu linky

V předchozích kapitolách byl popsán postup klasifikace každého zařízení. Pro určení stavu celé linky je nutné tyto stavy spojit a vytvořit jeden výsledný stav za celou linku. Informace o tom, že mohla nastat pauza, vychází z klasifikace stavů otočného stolu. Pokud stůl hlásí, podmínky vhodné pro pauzu a zároveň žádné zařízení není ve stavu "pracuje"(ideálně potom ve stavu "vypnuto", u robotu "stojí na brzdách "), je výsledek klasifikace, že nastala pauza.

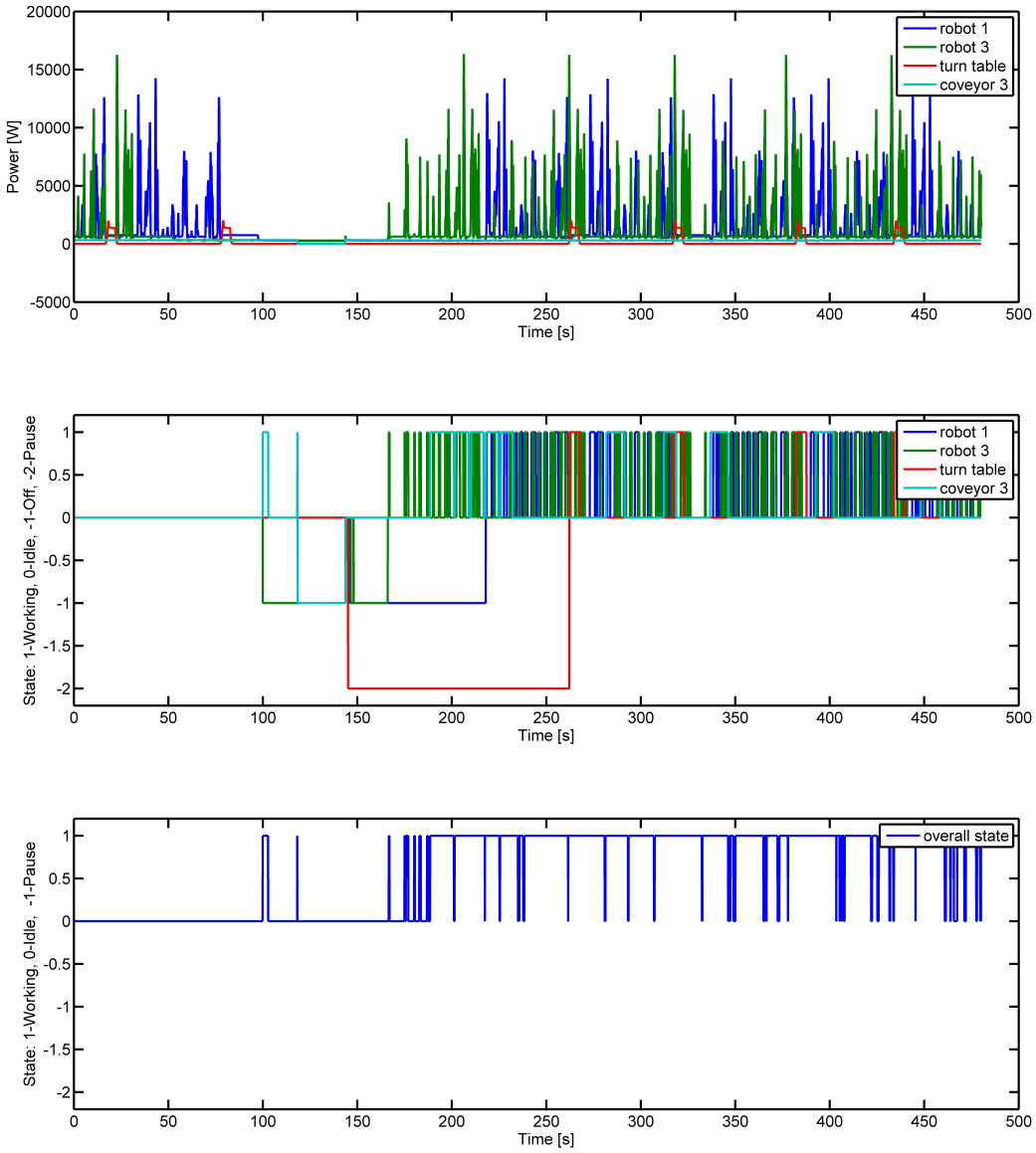
Na Obrázku 24 je vidět výsledek detekce pro menší zařízení. Lepička není v grafu obsažena, protože její signál překrýval ostatní signály a její stav po celou dobu těchto dat byl stejný a to "práce". Dobře je zde vidět, že je potřeba spojit data ze všech zařízení pro výsledné určení, jestli nastala pauza. V době, kdy je otočný stůl ve stavu pauzy, se ještě hýbou dopravníky.

Na Obrázku 25 je potom vidět celková klasifikace stavu linky. Je zde dobře patrné, že linka není ve stavu pauzy, přestože to detekce otočného stolu hlásí. Důvodem jsou ostatní zařízení, která už pracují.

6.5 Analýza spotřeby jednotlivých operací robotů

Aby bylo možné vytvořit prototypy operací, bylo nutné analyzovat konkrétní operace robotů. To bylo možno udělat na základě dodaného Ganttova diagramu. Při analýze se ukázalo, že délky jednotlivých operací robotů se mírně liší oproti popisu v diagramu. To může být způsobeno tím, že návrh z digitální továrny, pro který je Ganttův diagram vytvořen, zřejmě počítá s jedním dílem v lince, ale v reálu jsou v lince přítomny čtyři díly a ty ovlivňují různé doby čekání na uvolnění pracovišť. Dále by pak mohly hrát roli modely robotů, které byly při návrhu použity.

Způsob detekce operací pro jeden cyklus linky (od vstupu dílu do linky po výstup hotového výrobku z tohoto dílu) byl zvolen takto:



Obrázek 25: Detekce stavu celé linky

1. nalezení otočení stolu v datech výkonu otočného stolu
2. zarovnání dat a časové osy, tak, aby otočení stolu odpovídalo času v diagramu
3. podle diagramu nalézt začátek cyklu robotu
4. na základě diagramu určit časy operací
5. posunutí konců operací tak, aby byly na konci pohybů (na kraji výkonové špičky)
6. integrace výkonu pro zjištění energie spotřebované během operace

Při této analýze se ukázalo, že některé roboty začínají dělat práci dříve než je popsáno diagramem - to je dáno tím, že operace probíhají jakmile, jsou dostupné prostředky pro provedení této operace (materiál, robot, volný pracovní stůl). Dále u některých robotů není v diagramu postižen čas, kdy robot čeká na uvolnění některého z výrobních prostředků a to může ztížit vyhodnocení dat. Spotřeba jednotlivých operací z jednoho cyklu je uvedena v Tabulce 16 (Příloha B).

6.6 Implementace modulu klasifikace operací pro roboty

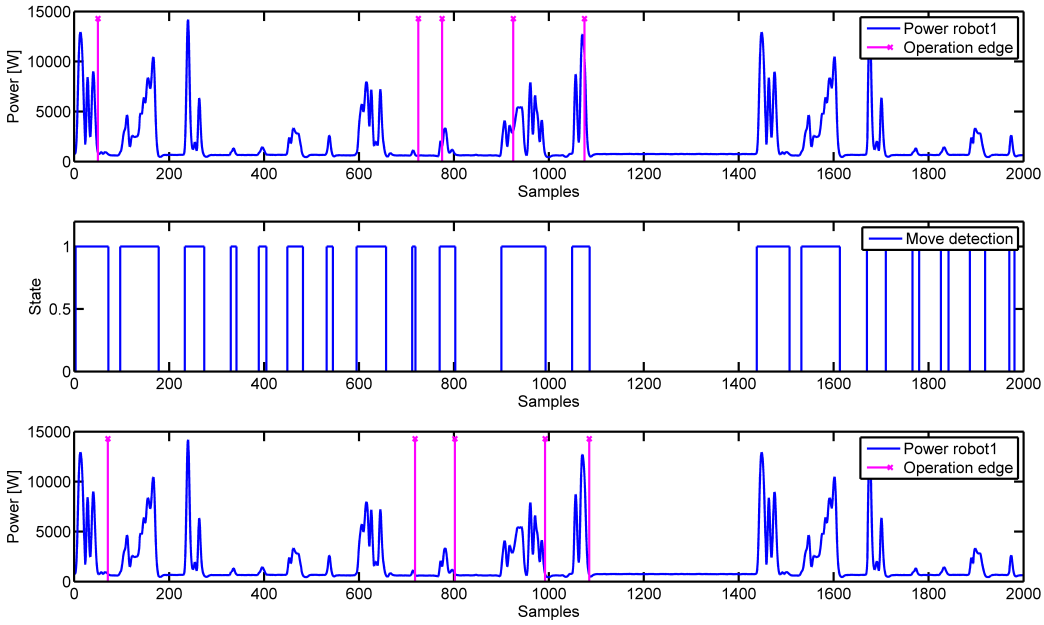
Pro klasifikaci operací je možné použít metodu nejbližšího souseda. Kdy se klasifikovaná operace porovnává s prototypy, u kterých je známo, jakou operaci představují. Operace je klasifikována jako prototyp, který je v použité metrice této operaci nejbližší.

Nejprve bylo nutné rozdělit data výkonu robotu podle jednotlivých operací. Dále bylo potřeba vytvořit množinu trénovacích dat a vytvořit prototypy. Pro obecnost je vhodné použít více prototypů stejné operace, a z těch potom nějakým způsobem vytvořit jeden, aby bylo porovnávání rychlejší.

6.6.1 Dělení dat

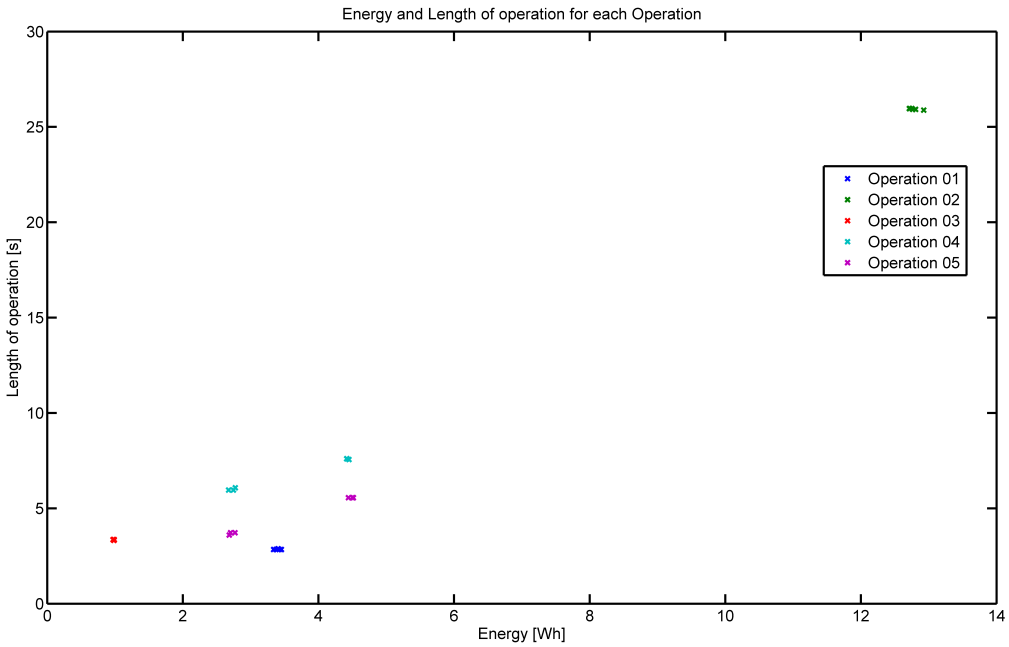
V předchozí kapitole byly délky operací upravovány ručně. Pro automatické zpracování je potřeba aby i tato úprava probíhala automaticky. Implementován byl následující postup:

1. vyhlazení signálu mediánovým filtrem
2. nalezení začátku práce zařízení
3. zarovnání dat a časové osy, tak, aby začátek cyklu zařízení odpovídal času 0
4. určení časů operací na základě diagramu
5. detekce pohybů (výkonových špiček) na základě přechodu filtrovaného signálu přes definovaný práh
6. posunutí konců operací tak, aby byly na konci pohybů
7. integrace výkonu pro zjištění energie spotřebované během operace a výpočet délky operace



Obrázek 26: Úprava délky operací v závislosti na změřených datech

Na Obrázku 26 je vidět signál s operacemi vyznačenými v časech daných diagramem a potom signál s posunutými operacemi. Na Obrázku 27 jsou vidět délky operací a energie spotřebovaná během více cyklů při jednotlivých operacích.



Obrázek 27: Délka operací a energie během nich spotřebovaná

6.6.2 K-means

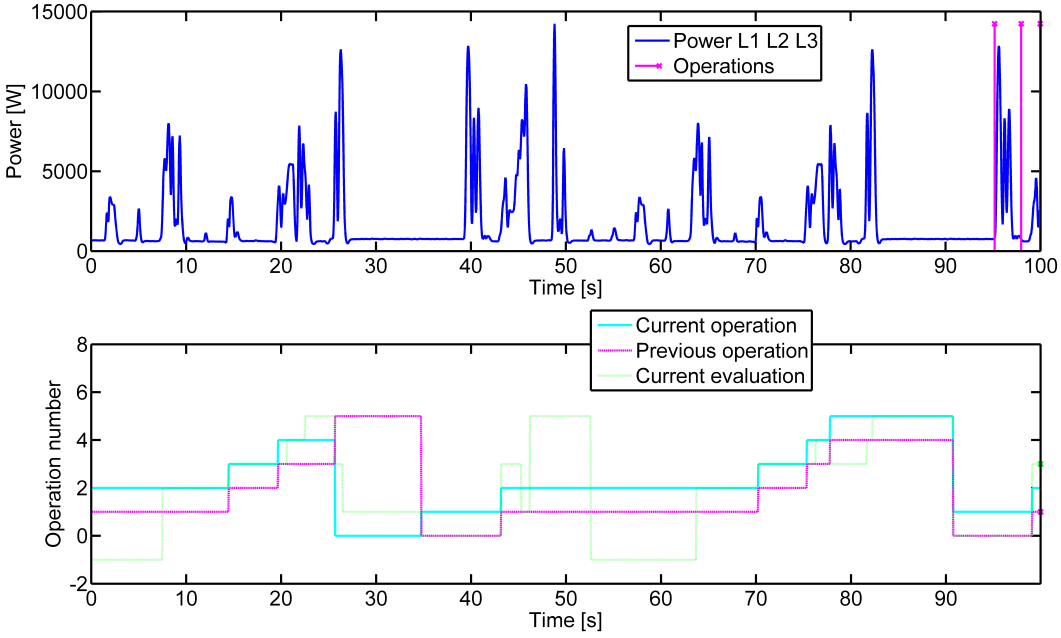
Při vytváření prototypů operací byla nejprve použita data z měření dlouhého 22h (dáno zpracováním dat na menší celky) provozu linky. Pro velké množství cyklů dochází při detekci operací k chybám a také některé cykly jsou významně delší než ostatní kvůli frézování čepiček či nějaké další chybě a výsledná získaná data potom netvoří snadno oddělitelné shluky viz Obrázek 29 (Příloha C)

Tyto špatné detekce by mohlo jít oddělit pomocí klasifikační metody K-means. Operace by se nechaly klasifikovat pomocí této metody a poté by se porovnala tato klasifikace s klasifikací, která byla určena při dělení operací. Prototypy, u nichž by se klasifikace lišila,

by potom byly považovány za špatně detekované a odstraněny. Ze zbylých prototypů by potom byly vytvořeny prototypy pro klasifikaci.

Ukázalo se, že využití K-means je příliš složité a mnohdy nepřesné. Vzhledem ke startu algoritmu z náhodného bodu a vzhledem k použité metrice (eukleidovská metrika v souřadném systému (energie spotřebovaná operací,délka operace)), kdy jsou některé shluky příliš blízko sebe a je složité je pomocí této metody separovat. Bylo tedy rozhodnuto ustoupit od této metody. Při nalezení vhodné metriky a využití apriorní informace o délce každé operace a její spotřeby, by metoda K-means mohla být výhodná.

Pro určení prototypu byl použít medián délky a energie operací určených, pro menší soubor hodnot, kde se vyskytovalo jen velmi málo vychýlených hodnot. Na Obrázku 28 je vidět průběh klasifikace pro robot 1. Je zde vidět, že aktuální odhad operace, který je závislý na datech, které postupně přicházejí, se často mění a je nepřesný. To je dáno postupným načítáním času a spotřebované energie dokud neproběhne celá operace.



Obrázek 28: Klasifikace operací robotu 1

6.7 Ověření funkce klasifikace operací

Klasifikace stavů linky funguje dle předpokladu. Pokud je linka celá v klidu, modul hlásí stav pauzy. Pokud je doba mezi dvěma pracemi stolu dostatečně dlouhá, pro hlášení stavu pauzy, ale některé ze zařízení stále nebo opět pracuje, není pauza hlášena.

Ukázalo se, že navržený postup pro detekci jednotlivých operací robotu není dobře použitelný úplně pro všechny roboty. Robot 4 například v průběhu programu čeká a to není postiženo v Ganttově diagramu. Začátek práce robotu je detekován pomocí doby nečinnosti - předpokládá se, že robot několik vteřin nic nedělá, než začne další úkol. To odpovídá Ganttově diagramu. Ve skutečnosti roboty, které čekají v průběhu operace, navazují dalším cyklem téměř ihned a je složité detekovat tento přechod. Naprogramovaný algoritmus pro dělení dat je potom při dělení dat nepřesný a špatně rozdělená data vedou ke špatné klasifikaci operací. Určení začátku cyklu takovéhoto robotu by mohlo být provedeno podle závislosti operací (z Ganttova diagramu) ostatních robotů, pro které algoritmus funguje dobře. Tato možnost nebyla implementována pro časovou náročnost implementace. Prioritu při implemetace měl takový modul, který detekuje práci a odstávky linky a který je zásadní pro řízení spotřeby linky. A tento modul byl vytvořen.

7 Závěr

Cílem této práce bylo řešení úspory energie na průmyslové lince. Pro měření výkonu trojfázového zařízení byl vytvořen malý přenosný rozvaděč, který disponuje připojením k síti PROFINET a obsahuje měřicí karty dvou různých výrobců pro případné porovnání. Pomocí tohoto rozvaděče byl měřen robot KUKA KR5 ARC. Na tomto robotu byl testován také profil PROFIenergy pro úsporu energie.

Data z měření byla odesílána pomocí speciálního programu a sbírána do databáze, aby následně mohla být vyhodnocena.

7.1 Robot KUKA KR5 ARC

Při testování robotu v laboratoři na Karlově náměstí byly prozkoumány jeho možnosti z pohledu profilu PROFIenergy. Dále byla vytvořena konfigurace pro PLC, aby bylo možné robota ovládat a hýbat s ním. Z testování robotu vzešlo velké množství poznatků, jak co udělat, pro další studenty, kteří na tomto robotu budou pracovat.

7.2 Nasazení na reálné lince

Systém měření spotřeby byl rozšířen na reálnou linku ve Škoda Auto. Tato linka svařuje díl karoserie automobilu a mimo jiné obsahuje 6 robotů KUKA, které obstarávají většinu úkonů při tvorbě tohoto dílu. Ukázalo se, že roboty (kontroléry robotů) nedisponují modulem PROFIenergy. Nicméně podporují úsporu energie pomocí tří energeticky různě náročných módů. Tyto módy se velmi podobají módům, které podporuje kontrolér v laboratoři s funkcionalitou PROFIenergy. Případné použití PROFIenergy u robotů na lince tak příliš neovlivní aktuálně navržené postupy, spíše to zjednoduší používání úspory energie u těchto robotů.

Z naměřených dat bylo možné analyzovat odstávky linky a parametry jednotlivých operací, které jsou na lince prováděny. K tomu byly využity informace z Ganttova diagramu

operací, které jsou nutné pro vytvoření dílu.

Protože se kontrolér v laboratoři a robot liší od robotů a kontrolérů, které jsou umístěny v průmyslové lince, bylo provedeno měření na testovacím robotu ve Škoda Auto v tamní dílně. Na základě analýzy dat a testování tohoto robotu potom vznikla strategie a postup pro uvádění linky do energeticky úsporného stavu. Konkrétní časová konstanta pro rozhodnutí nebyla určena, protože se nepodařilo zajistit testování úspory energie přímo na reálné lince. Původní časový plán byl příliš optimistický a v čase daném pro vytvoření této práce se nepodařilo domluvit testování strategie na lince. A to především z důvodu, že nasazení programu, který by spouštěl energeticky úsporné módy, je příliš velký zásah do linky (z hlediska výroby a stávajícího řídicího programu). Vyzkoušené postupy ale pokrývají situaci, kdy by se uspávaly roboty na reálné lince během víkendu a jsou připravené k nasazení.

Dále byl vytvořen softwarový modul, který z měřených dat odhaduje stav linky. Konkrétně stav, jestli linka prochází nějakým druhem odstávky a bylo by tedy vhodné ji převést do energeticky úsporného stavu. Dále byl modul testován pro detekci jednotlivých operací různých zařízení. Navržený modul dobře fungoval pro menší zařízení. Pro roboty, které provádějí složité operace, fungoval pouze v případě, že se robot držel diagramu operací a tedy před začátkem pohybu byl určitou dobu v klidu.

Reference

[1] Zemánek Ivan. Studijní materiály k předmětu A3B31EOP, kapitola 5.8. , 2010.

[2] Ripka Pavel. Studijní materiály k předmětu A3B38SME, přednáška č. 3. , 2011.

[3] Siemens. ET200SP distributed I/O system, System Manual ,dostupné z: http://support.automation.siemens.com/WW/view/en/58649293. , 2014.

[4] Siemens. ET 200SP: Analog input module: AI Energy Meter ST ,dostupné z: http://www.siemens.com/simatic-tech-doku-portal. , 2013.

[5] WAGO Kontakttechnik GmbH & Co. KG. WAGO-I/O-SYSTEM 750: 3-Phase Power Measurement Module, dostupné z: http://www.wago.com/wagoweb/documentation/navigate/nm0_-dxe.htm. , 2013.

[6] Siemens. Memory concepts for SIMATICS S7-300 CPUs and for C7 Devices. , 2006.

[7] Berger Hans. Automatizace se STEPem 7 v AWL. , 1998.

[8] Siemens. System Software for S7-300/400 System and Standard Functions. , 2006.

[9] KUKA Roboter GmbH. KR C4 Operating Instructions. , 2012.

[10] KUKA Roboter GmbH. KUKA System Software 8.2: Návod k obsluze a programování pro konečné uživatele. , 2012.

[11] KUKA Roboter GmbH. KR C2/ KR C3: Expert Programming. , 2005.

[12] KUKA Roboter GmbH. KUKA System Software 8.2: Operating and Programming Instructions for System Integrators. , 2011.

[13] Siemens. Technical Specification for PROFINET. , 2012.

[14] Siemens. Siemens AG, PROFIenergy with ET200S, dostupné z: http://support.automation.siemens.com/WW/view/en/41986454. , 2011.

A Seznam zkratek

BPO bus power off

HIB hibernate

KCP KUKA control panel

KRC KUKA robotic controler

KRL KUKA robot language

NTP netvork time protocol

PLC programmable logic controller

RMS root mean square

RTO ready to operate

tzn. to znamená

WOL wake on lan

B Příloha- Tabulky

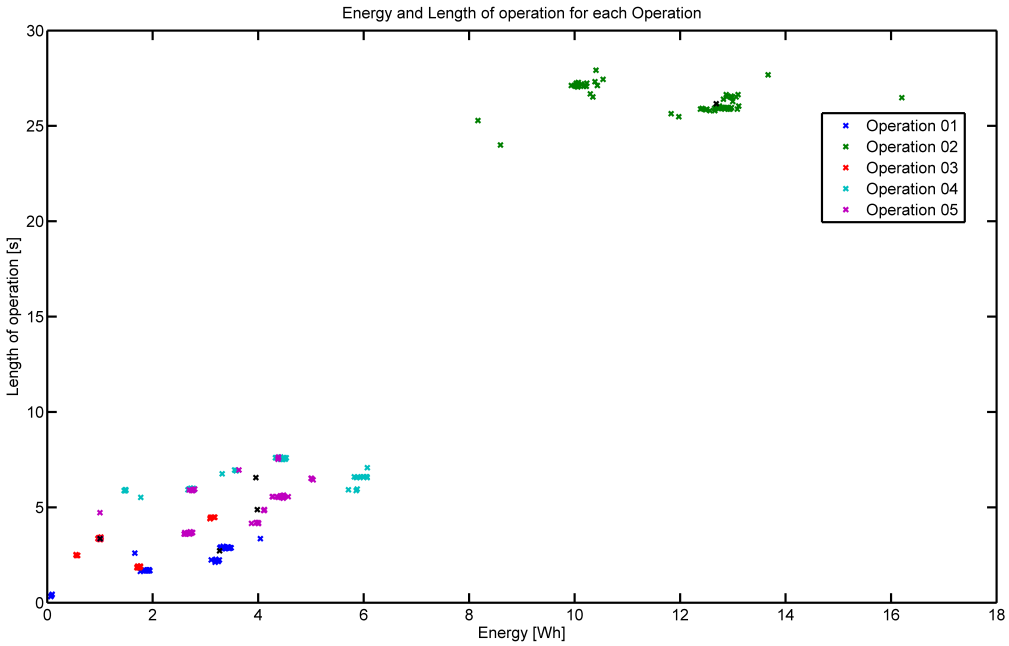
Parametr	**Siemens: AI Energy METER ST**	**WAGO: 750-494-0001**
Měřicí vstupy	6 (3x proud, 3x napětí)	6 (3x proud, 3x napětí)
Vstupní napětí	230V - tři fáze	480V
Měření pouze napětí	NE	ANO
Měření pouze proudu	ANO	ANO
Měření fázových úhlů	ANO	ANO
Počet hodnot najednou odeslaných	12	4
Prostor v PLC I	32	24
Prostor v PLC Q	12	24
Vzorkovací frekvence		
RMS napětí Lx-N	50ms	40ms
RMS napětí Lx-Ly	50ms	340ms
RMS proud Lx	50ms	40ms
Činný výkon Lx	200ms	40ms
Jalový výkon Lx	200ms	340ms
Zdánlivý výkon Lx	200ms	40ms
Rozlišení		
Napětí	0,01V	0,01V
Proud	0,001A	0,0005A
Činný výkon	1W (aritmetická střední hodnota)	0,05W
Jalový výkon	1VAR (aritmetická střední hodnota)	0,05VAR
Zdánlivý výkon	1VA (aritmetická střední hodnota)	0,05VA

Tabulka 15: Porovnání měřicích karet výrobců WAGO a Siemens

Zařízení	typ prace	energie [Wh]	doba [s]
Robot 1	operace1	3,57	2,80
	operace2	13,63	25,80
	operace3	0,44	2,40
	operace4	1,35	5,20
	operace5	4,36	6,00
Robot 2	operace1	2,40	5,20
	operace2	2,61	2,80
	operace3	4,41	5,60
	operace4	2,44	2,40
	operace5	2,64	4,00
	operace6	1,51	7,40
	operace7	11,71	24,20
	operace8	3,11	2,40
Robot 3	operace1	6,17	6,17
	operace2	1,00	5,80
	operace3	0,70	3,00
	operace4	1,07	6,20
	operace5	2,29	2,80
	operace6	5,45	19,20
	operace7	3,09	4,80
	operace8	2,76	4,00
	operace9	2,96	2,40
Robot 4	operace1	1,95	5,20
	operace2	1,49	2,60
	operace3	2,32	5,00
	operace4	0,71	1,72
	operace5	5,57	13,28
	operace6	1,28	1,40
	operace7	1,72	11,32
	operace8	3,30	3,48
	operace9	0,91	5,60
	operace10	4,62	3,60
Robot 5	operace1	18,14	31,48
Robot 6	operace1	0,00	0,00
	operace2	3,45	6,00
	operace3	3,85	2,60
	operace4	7,38	19,80
	operace5	1,56	2,40
	operace6	4,01	6,00
	operace7	0,17	1,00
	operace8	1,49	1,32
	operace9	0,26	0,40
	operace10	0,65	1,28

Tabulka 16: Spotřeba a délka operací jednotlivých robotů

C Příloha- Obrázky



Obrázek 29: Operace z mnoha cyklů

D Příloha CD

D.1 Obsah CD

- /
  - `dp` ..... diplomová práce v pdf formátu
  - `KUKA_KR5_ARC/`
    - `KRC` ..... soubory řídicí jednotky KRC robota v laboratoři
  - `STEP7/`
    - `Kuka-safety1.zip` .. projekt robotu v laboratoři bez odesílání dat do databáze

E Příloha - soubory KRC

E.1 CELL.SRC

```
&ACCESS RVO
&REL 19
&COMMENT HANDLER on external automatic
DEF  CELL ( )
  ;EXT EXAMPLE1 ( )
  ;EXT EXAMPLE2 ( )
  ;EXT EXAMPLE3 ( )

  ;FOLD INIT
  ;FOLD BASISTECH INI
  ;FOLD CHECK HOME
  ;FOLD PTP HOME Vel=5 %

  ; inicializace vystupu
  so_konec_prace = False
   MasRef_Main()  ; overeni reference
   BRAKETESTREQ() ; test brzd
  ;FOLD AUTOEXT INI

  LOOP
    POO (#EXT_PGNO,#PGNO_GET,DMY[],0 )
    MasRef_Main()  ; overeni reference
    BRAKETESTREQ() ; test brzd
    SWITCH  PGNO ; Select with Programnumber

    CASE 1
      POO (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
      plc1_polystyren( )
      so_konec_prace = True
```

```
    wait for si_konec_prace
    so_konec_prace = False

  CASE 2
    P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
    plc1_polystyren_zpet( )
    so_konec_prace = True
    wait for si_konec_prace
    so_konec_prace = False

  CASE 3
    P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
    ;EXAMPLE3 ( ) ; Call User-Program

  CASE 4
    P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
    pwsv1( )
    so_konec_prace = True
    wait for si_konec_prace
    so_konec_prace = False

  CASE 5
    P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
    progchar( )
    so_konec_prace = True
    wait for si_konec_prace
    so_konec_prace = False

  CASE 6
    P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
    progchar2vyska( )
    so_konec_prace = True
    wait for si_konec_prace
```

```
      so_konec_prace = False

    CASE 7
      P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
      ct2014v2( )
      so_konec_prace = True
      wait for si_konec_prace
      so_konec_prace = False

    CASE 8
      P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
      ptp2014( )
      so_konec_prace = True
      wait for si_konec_prace
      so_konec_prace = False

    CASE 9
      P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
      krempa( )
      so_konec_prace = True
      wait for si_konec_prace
      so_konec_prace = False

    DEFAULT
      P00 (#EXT_PGNO,#PGNO_FAULT,DMY[],0 )
    ENDSWITCH
  ENDLOOP
END
```

Výpis 1: cell.src

E.2 SPS.SUB

```
&ACCESS RVO
&REL 20
&COMMENT PLC on control
DEF  SPS ( )
  ;FOLD DECLARATIONS
  ;FOLD INI

  LOOP
     WAIT FOR NOT($POWER_FAIL)
     TORQUE_MONITORING()

    ;FOLD USER PLC
    ;Make your modifications here
$DOPOSACT_X=$POS_ACT.X
$DOPOSACT_Y=$POS_ACT.Y
$DOPOSACT_Z=$POS_ACT.Z

;$DOPOSACT_X=$AXIS_ACT.A1
;$DOPOSACT_Y=$AXIS_ACT.A2
;$DOPOSACT_Z=$AXIS_ACT.A3
;$COUNTVEL=$AXIS_ACT.A4
;$POSOUT2=$AXIS_ACT.A5
;$POSOUT3=$AXIS_ACT.A6

    ;ENDFOLD (USER PLC)
  ENDLOOP
  ;FOLD ;%{H}
  ;ENDFOLD

;FOLD USER SUBROUTINE
```

Výpis 2: sps.sub

E.3 CONFIG.DAT

```
&ACCESS RV$
&REL 12
DEFDAT $CONFIG

SIGNAL $DOPOSACT_X $OUT[33] TO $OUT[64]
SIGNAL $DOPOSACT_Y $OUT[65] TO $OUT[96]
SIGNAL $DOPOSACT_Z $OUT[97] TO $OUT[128]

SIGNAL $COUNTVEL $OUT[161] TO $OUT[192]
;SIGNAL $POSOUT1 $OUT[161] TO $OUT[192]
SIGNAL $POSOUT2 $OUT[193] TO $OUT[224]
SIGNAL $POSOUT3 $OUT[225] TO $OUT[256]

ENDDAT
```

Výpis 3: config.dat

E.4 OPTION.DAT

```
&ACCESS  RV
&PARAM DISKPATH = MADA
&PARAM VERSION = 1.0.0
&PARAM VERSION=1.0.0
&REL 3
DEFDAT  $OPTION PUBLIC
CHAR $V_OPTION[32]
$V_OPTION[]="V1.0.0/KUKA8.2" ;VERSIONSKENNUNG
BOOL $TECH_OPT=TRUE ;FUNKTIONSGENERATOR
BOOL $TCP_IPO=TRUE ;GREIFERBEZOGENE INTERPOLATION
BOOL $SEP_ASYNC_OV=FALSE ;Schalter fuer asynchrone Hand-Overrides
BOOL $LOOP_CONT=FALSE
CHAR $LOOP_MSG[128]
```

```
BOOL $IDENT_OPT=TRUE
INT $SINGUL_STRATEGY=0 ;0=NONE, 1=APPROX
BOOL $MOT_STOP_OPT=FALSE ;AKTIVIERUNG "BLOCK EXTERNAL START"
BOOL $CHCK_MOVENA=FALSE ;AB-UND EINSCHALTEN DER UEBERPRUEFUNG DER
   EINGANGSNUMMER VON $MOVE_ENABLE
BOOL $COLLISAVOID=FALSE ;
BOOL $MOTIONCOOP=FALSE ;
BOOL $PROGCOOP=FALSE ;
BOOL $T2_OV_REDUCE=TRUE ;TRUE = Overridereduzierung auf 10 % in Mode T2
BOOL $VAR_TCP_IPO=FALSE
BOOL $SPL_VEL_MODE_OPT=TRUE ; Defaulteinstellung fuer $SPL_VEL_MODE
BOOL $IMPROVEDMIXEDBLENDING=TRUE ; verbessertes gemischtes Ueberschleifen
ENDDAT
```

Výpis 4: option.dat

F Příloha - Používání robotu KUKA

Užitečné tipy

- Před přehráním projektu v KRC je vhodné tento projekt zálohovat. Pokud by došlo k chybnému nahrání. Pokud spadne nahrávání projektu z důvodu nekompatibility některých modulů, je možné obnovit projekt takto:

  1. nahrát projekt, který je kompatibilní
  2. restartovat KRC.
  3. nahrát nekompatibilní projekt - dojde k pádu
  4. restartovat KRC
  5. nyní je nahrán hybrid obou projektů a je potřeba zkontrolovat, že je bezpečnost v pořádku
  6. robot by měl fungovat

- v aktuálním projektu je nastavena bezpečnost tak, že i při režimu T1 a T2 se chová porušení bezpečné zóny (něco projde optickým závěsem) jako bezpečnostní zastavení. V provozu je obvyklé, že v režimu T1 a T2 hýbe robotem zkušený pracovník, který hlídá bezpečnost a potřebuje přístup k robotu a tedy světelný závěs při přerušení nespustí bezpečnostní zastavení.

G PE na robotu KUKA

Robot KUKA KR5 ARC podporuje úsporu energie podle profilu PROFIenergy (přesněji řečeno je PROFIenergy podporováno jeho řídicí jednotkou- KRC). Posílání příkazů je možno vyřešit pomocí specializovaného funkčního bloku *FB816:PE_CMD*.
Robot podporuje tři energetické módy:

- Ready to operate -mód pro běžné používání robota
- DriveBus power off
- Hibernate

Předpokladem pro přechod mezi energetickými módy je řízení robotu z PLC tzn. robot je v režimu EXT.

G.1 Mód DriveBus power off

V tomto módu jsou osy zastaveny na brzdách a navíc je odepnuto napájení sběrnice pro napájení serv. Čili servoměniče umístěné ve skříni KRC jsou přepnuty na úsporu energie.

G.1.1 Uspání

Jde provést pomocí bloku *FB816:PE_CMD*. Parametry:

- pe_cmd: 1
- pe_cmd_para_length:4
- pe_cmd_para.dw1: time of pause (time)

pozn.: VAT tabulka "PE"

G.1.2 Probuzení

Jde provést pomocí bloku *FB816:PE_CMD*. Dále je potřeba kvitovat hlášení, aby bylo možno s robotem hýbat. Parametry:

- pe_cmd: 2

pozn.: VAT tabulka "PE"
Kvitování probíhá pomocí následující sekvence signálů, která odpovídá obrázku na straně 181 manuálu (KSS_82_SI_en_v4.pdf) viz Obrázek 30.:

1. Q:*$DRIVES_ON*: 1 → I:*$PERI_RDY*: 1
2. Q:*$CONF_MESS*: 1 → I:*$STOP_MESS*: 0
3. Q:*$EXT_START*: 1 → program running: 1

V tuto chvíli by se mělo ovládání programu stejně, jako před posláním příkazu k uspání.
pozn.: VAT tabulka "AUT_EXT"

G.2 Mód hibernace

V tomto módu je vypnuto vše, co je možné vypnout, aby řídicí jednotka ještě mohla reagovat na vnější povely. Zařízení se také odpojí od PN sítě.

G.2.1 Uspání

Jde provést pomocí bloku *FB816:PE_CMD*. Parametry:

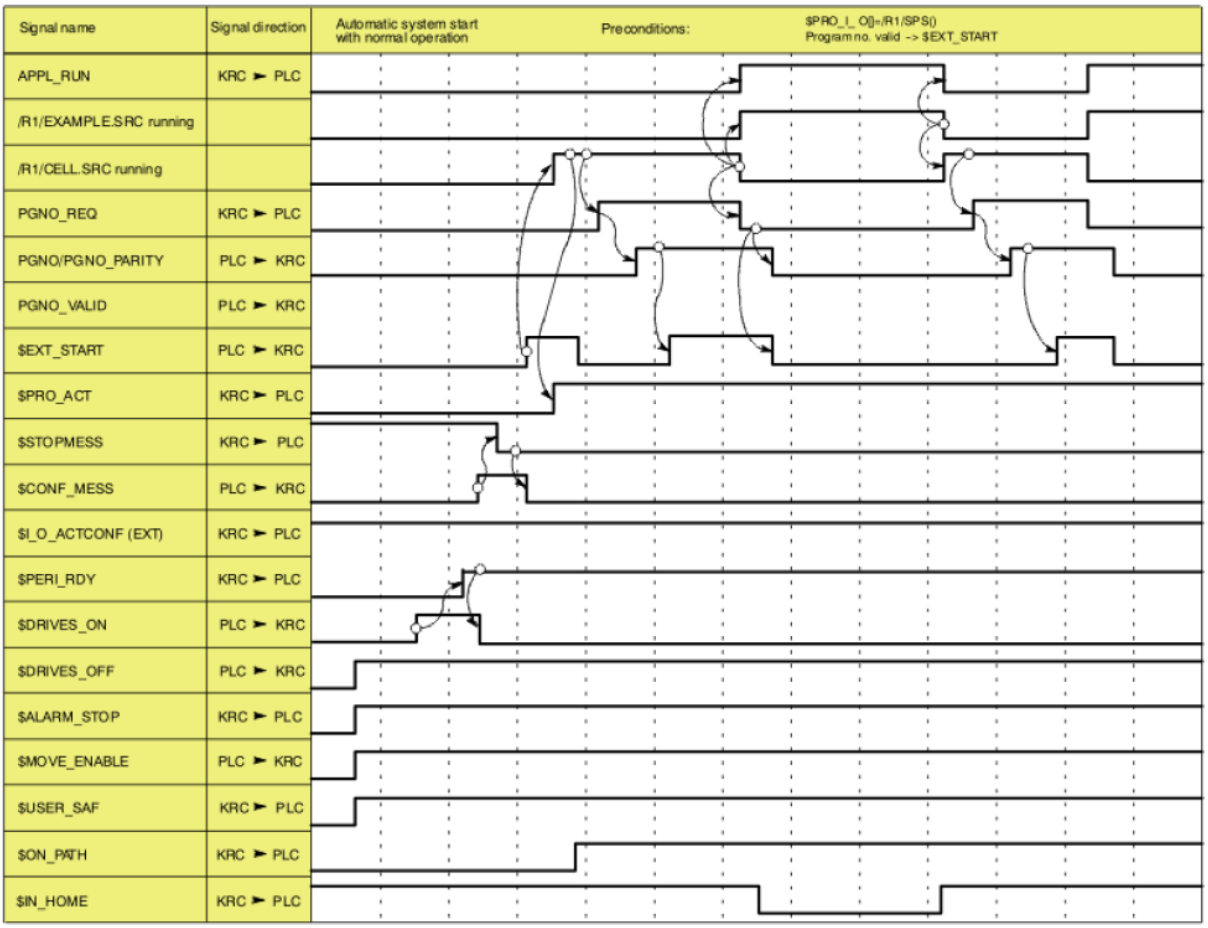
- pe_cmd: 21

pozn.: VAT tabulka "PE"

G.2.2 Probuzení

Takto uspané zařízení je možno probudit pomocí WOL paketu (Wake On Lan). Tento paket obsahuje MAC adresu probouzeného zařízení a je potřeba ho poslat na "broadcast"konkrétní podsítě.
pozn.: VAT tabulka "WOL" V této tabulce je potřeba nastavit správnou MAC adresu a IP adresu probouzeného zařízení. Přepnutím proměnné "makerq"(DB32.DBX6.1) do logické 1 vytvoří WOL paket s ohledem na vloženou adresu. Proměnná "sendrq"(DB32.DBX6.0) potom iniciuje automatické připojení-odeslání paketu-odpojení.
Po tomto by se zařízení mělo probrat a je potřeba ještě provést kvit chyb. Opět postup odpovídá obrázku na straně 181 manuálu (KSS_82_SI_en_v4.pdf) viz Obrázek 30. s tím, že nejprve je nutno zmáčknout fyzické tlačítko pro kvitování chyby:

1. zmáčknout fyzické tlačítko pro kvitování chyby (poté správně naskočí ProfiSafe komunikace)
2. Q:*$DRIVES_ON*: 1 → I:*$PERI_RDY*: 1
3. Q:*$CONF_MESS*: 1 → I:*$STOP_MESS*: 0
4. Q:*$EXT_START*: 1 → program running: 1



Obrázek 30: Automatic system start and normal operation with program number acknowledgement by means of $EXT_START