

# Manuál pro robota řazení

Tento manuál tvoří samostatnou přílohu k diplomové práci DP2015 – MV01, Robot řazení, Praha 2015, autora Bc. Jana Rupricha.

## Obsah

<b>Stručný úvod</b> .....	1
<b>Principiální schéma zapojení</b> .....	1
<b>Koncepce řazení</b> .....	2
<b>Nastavení v MAXu</b> .....	3
<b>Vytvoření nového projektu</b> .....	4
<b>Prvotní seřízení robota na novou převodovku</b> .....	6
<b>Podprogramy</b> .....	9
kalibrace.vi.....	10
„vykonavatel“ .....	11
robot.vi.....	12
<b>Výstup z podprogramu robot.vi – kód výsledku</b> .....	13
<b>Požadavky na řídicí SW celého stavu</b> .....	14
<b>Výměna převodovek stejného typu</b> .....	14

## Stručný úvod

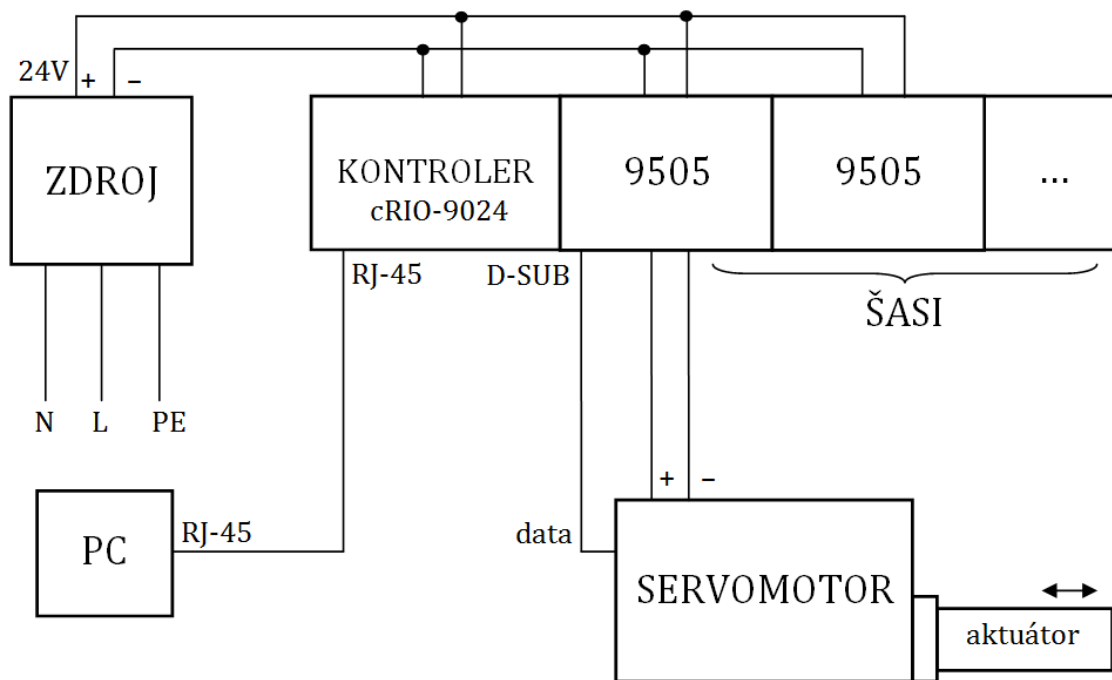
Tento robot řazení je zkonstruován pro ovládání převodovek na uzavřeném zkušebním stavu ve VTP Roztoky. Umí zařadit libovolný převodový stupeň nebo neutrál. Každá převodovka je ovládána svým robotem, přičemž oba dva jsou identické. Akční členy robota jsou dva servopohony, které mají za úkol posouvat bovdeny řazení do správných poloh. Uložení bovdenů řazení na převodovce je zachováno v nezměněném stavu, na straně u řadicí páky jsou bovdeny vyjmuty z tělesa řadicí páky a namontovány na robota.

Řídicí program robotů je proveden v SW Labview. Tento řídicí program lze rozdělit na dvě základní smyčky – smyčku řazení a smyčku proudové ochrany. Řadicí smyčka má za úkol pomocí sekvence operací provést zařazení, zatímco smyčka proudové ochrany má hlídat velikost proudu protékajícího servopohonem při samotném řazení, tzn. při pohybu mezi neutrálem a rychlostním stupněm.

## Principiální schéma zapojení

HW vybavení robota řazení je platforma CompactRIO se čtyřmi speciální moduly NI 9505 (pro každý servopohon jeden modul) a zdroj napájecího napětí, viz obrázek 1.

Zdroj stejnosměrného napětí je na jedné straně připojen k síti a na straně druhé napájí kontroler a všechny čtyři moduly, které jsou zasunuty do šasi. Každý modul ovládá právě jeden pohon. Propojení mezi modulem a pohonem je dvojitý – silové a datové. Silové realizuje napájení motoru a datové přenos signálu z enkodéru do modulu, který je pro tento účel opatřen D-SUB female konektorem. Celý systém pak komunikuje s počítačem pomocí síťového kabelu.



Obr. 1

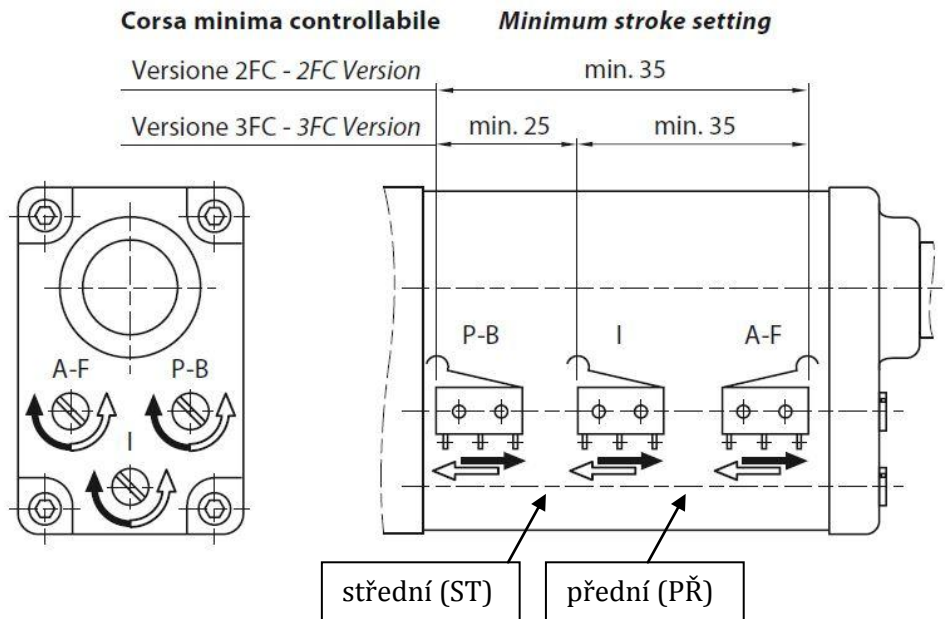
## Koncepce řazení

Tab1 – koncepce ovládní procesu řazení

VOLBA		ŘAZENÍ	
<b>R</b>	mikrospínač (PŘ) a (0)	<b>sudé</b>	mikrospínač (PŘ) a (-1100)
<b>I a II</b>	enkodér (240)	<b>neutrál</b>	enkodér (-620)
<b>III a IV</b>	enkodér (500)	<b>liché</b>	mikrospínač (ST) a (0)
<b>V (a VI)</b>	enkodér (720)	X	

Servopohony obsahují mikrospínače, viz obrázek 2, které fungují jako koncové rozpínací kontakty a dále rotační inkrementální snímač polohy aktuátoru (enkodér). Tabulka 1 zobrazuje princip řazení. Zpátečka je volena pomocí rozepnutí koncového spínače, ostatní řadící vidličky pomocí enkodéru. Řazení rychlostních stupňů je provedeno rovněž pomocí koncových spínačů, neutrál je najížděn pomocí enkodéru.

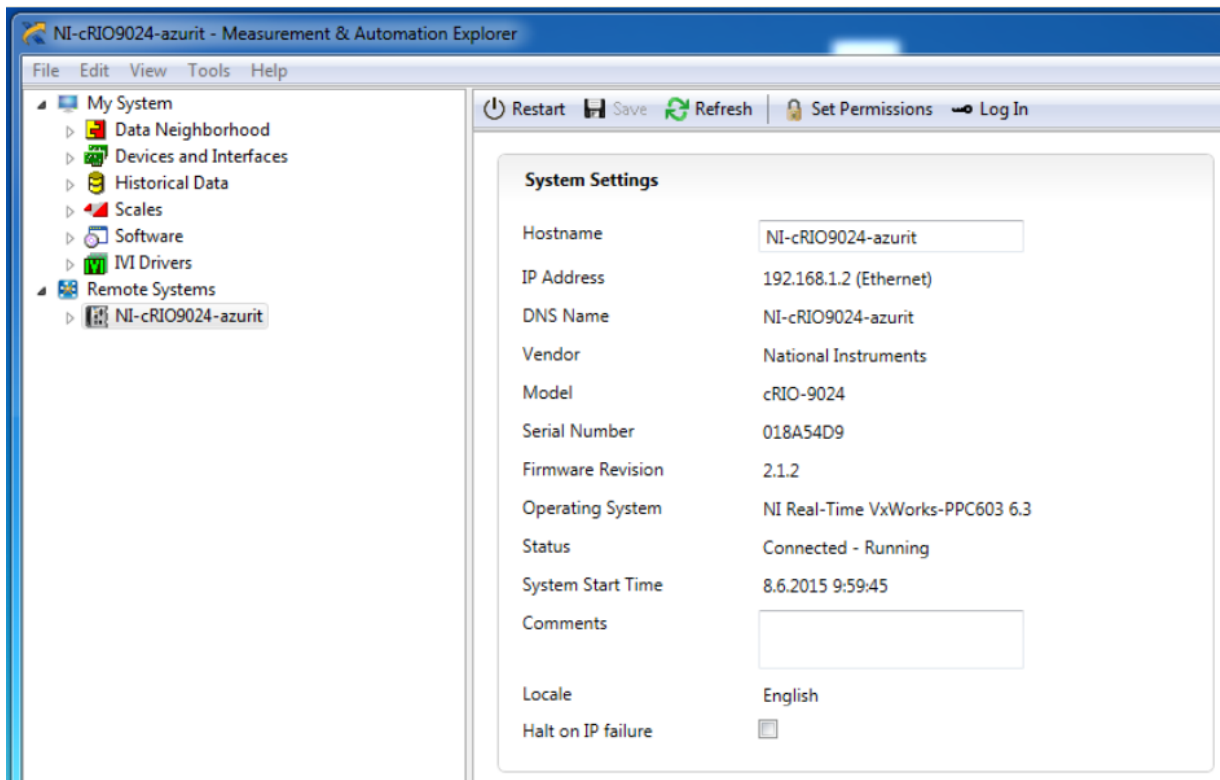
Tabulka výše tuto koncepci shrnuje. Číselné hodnoty v závorkách určují teoretický počet pulsů enkodéru v těchto polohách. **Pro jiný typ převodovky než pro návrhovou MQ 200 budou zde uvedené číselné hodnoty pravděpodobně jiné, princip však zůstává.**



Obr. 2

## Nastavení v MAXu

Po propojení platformy CompactRIO s počítačem pomocí síťového kabelu, musí být v programu MAX tento nový vzdálený systém načten. „DNS Name“ tohoto systému je „NI-cRIO9024-azurit“ a jeho „IP Address“ je 192.168.1.2. Po vyplnění IP adresy je systém načten a musí se zobrazit ve stromu vlevo pod „Remote Systems“, viz obrázek 3.

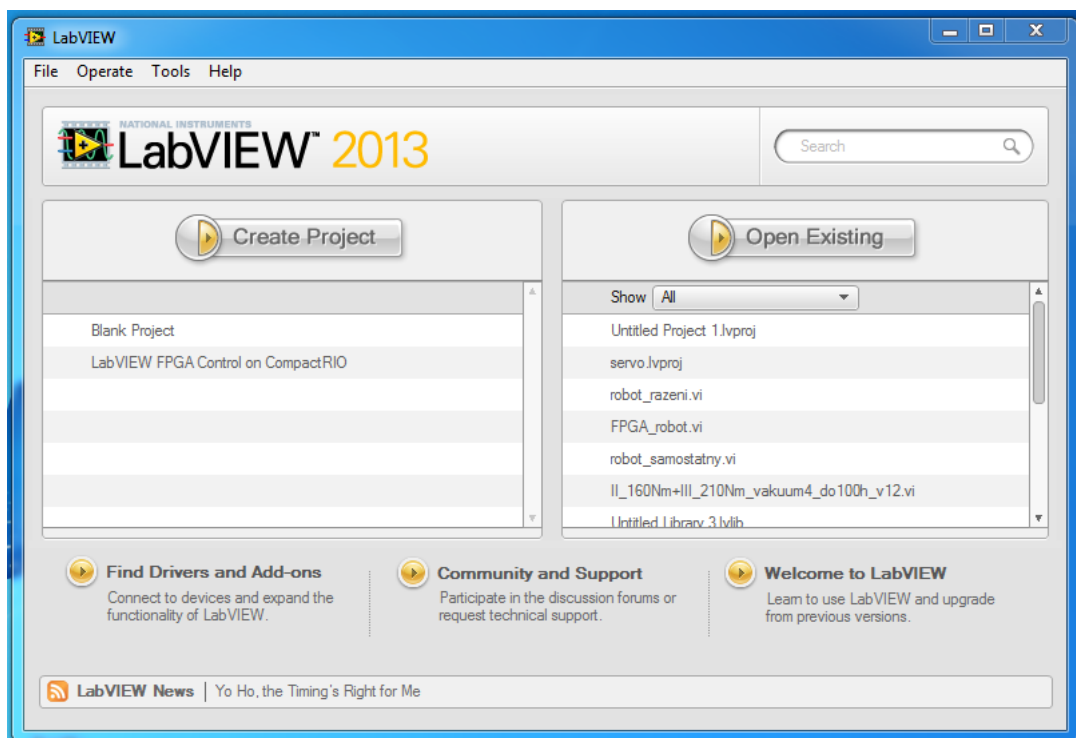


Obr. 3

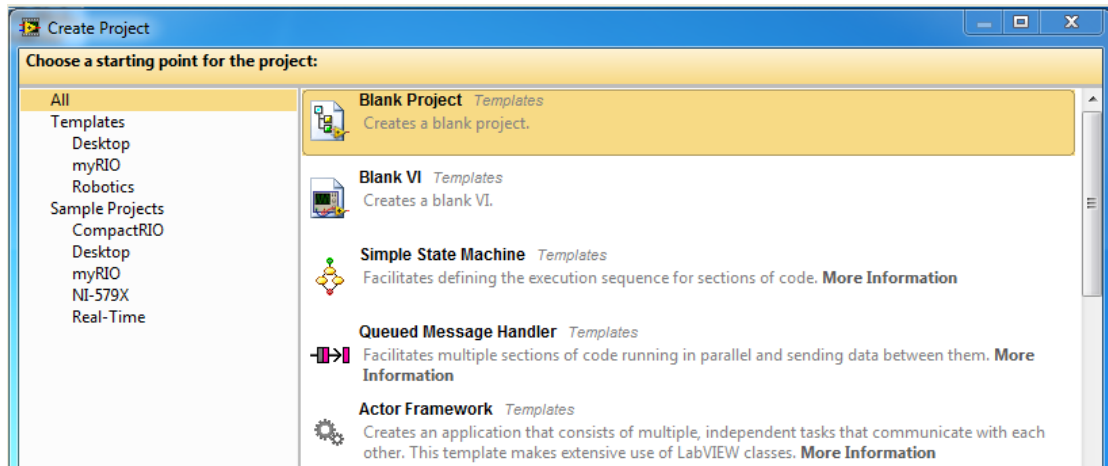
## Vytvoření nového projektu

Aby mohla fungovat komunikace mezi platformou CompactRIO a počítačem, musí být vytvořen projekt, kde jsou spravovány všechny programy, které používáme pro běh naší aplikace v „CompactRlu“. Rovněž zde intuitivně probíhá správa všech zásuvných modulů. Na několika obrázcích nyní popíši vytvoření nového projektu a přidání „CompactRla“ do tohoto projektu.

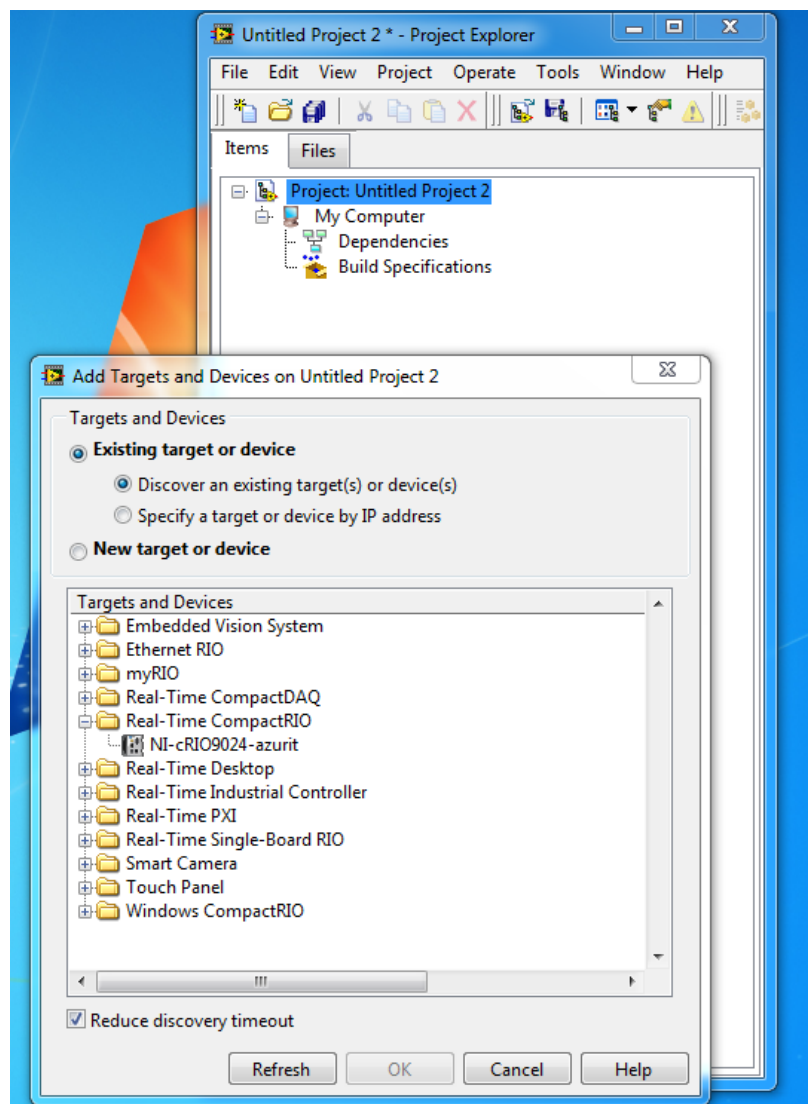
1. Otevřít program LabVIEW a zvolit „Create Project“ (obr. 4)
2. Vybrat „Blank Project“ a potom kliknout na „Finish“ (obr. 5)
3. Otevře se nové okno s prázdným projektem. Pravým tlačítkem myši kliknout na horní položku ve stromu „Project: nazev\_projektu“, vybrat „New“ a dále „Targets and Devices“. Je-li CompactRIO připojené, pak jej ve složce „Real-Time CompactRIO“ lze vybrat. Operaci potvrdit tlačítkem OK. (obr. 6)
4. Vybrat programovací mód „LabVIEW FPGA Interface“, potvrdit „Continue“ (obr. 7)
5. Do projektu se načte kompletní struktura „CompactRla“ (obr. 8)



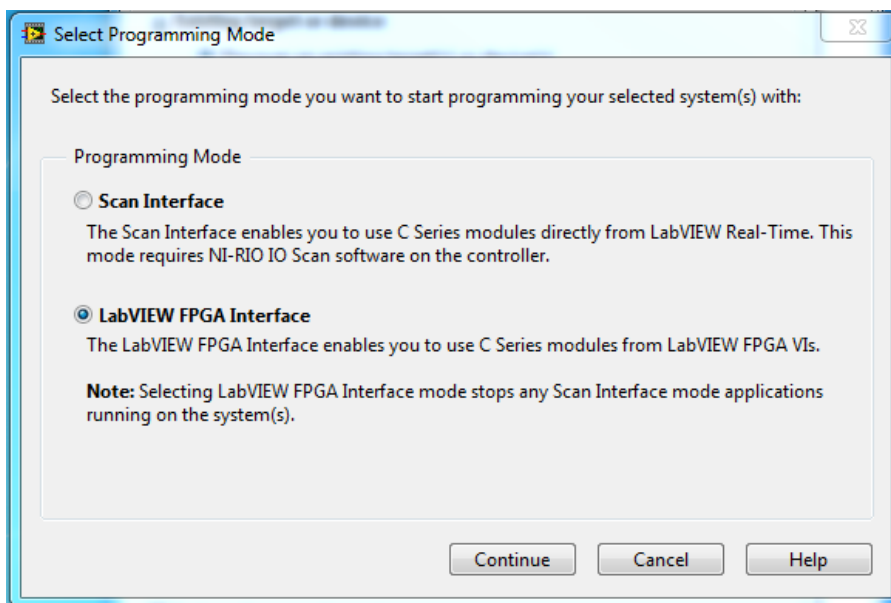
Obr. 4



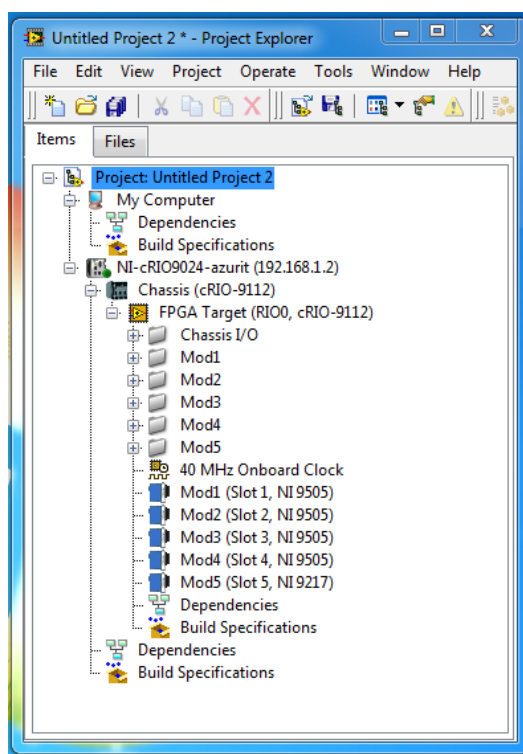
Obr. 5



Obr. 6



Obr. 7



Obr. 8

## Prvotní seřízení robota na novou převodovku

Z robota se odmontuje samojistící matice (poz. 23) na obou vozících, viz výkres sestavy robota řazení v Příloze A k diplomové práci, čímž jsou motory uvolněny a lze je odklopit a vhodným předmětem podložit tak, aby se vozíky mohly volně pohybovat po vedení. Bovdeny řazení jsou již zajištěny v *opěrném plechu* (poz. 11).

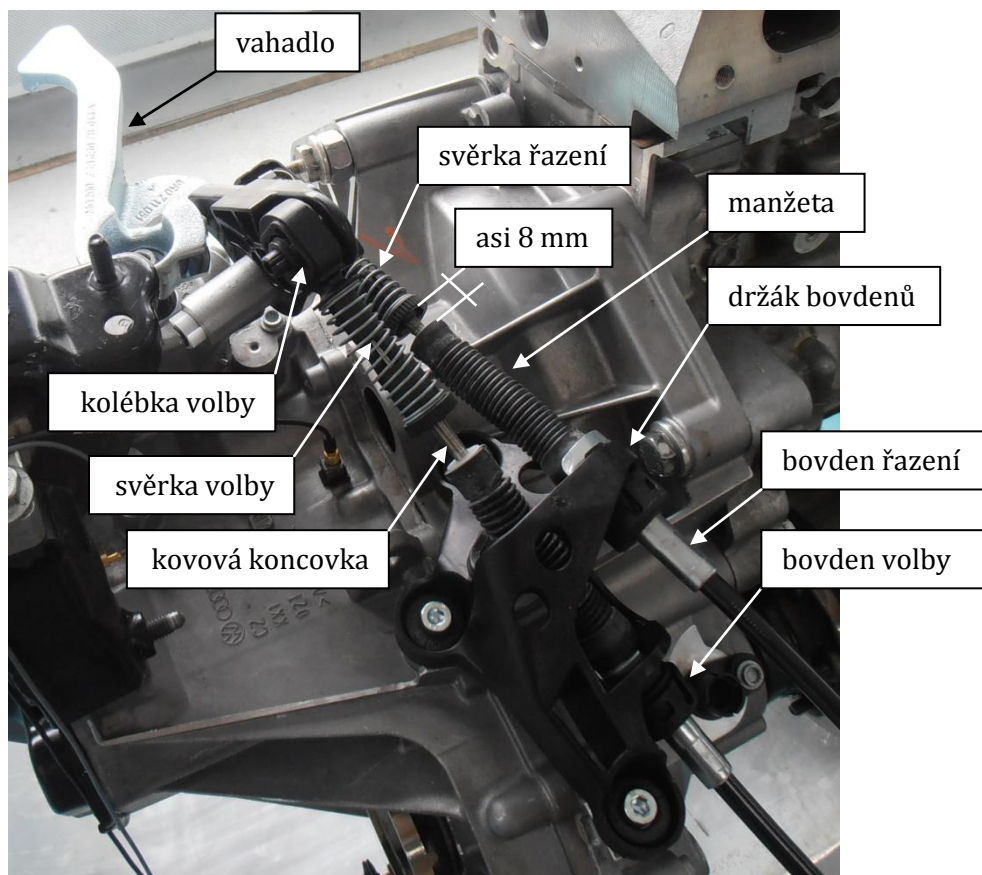
Nyní připevníme plastový držák bovdenu k převodovce pomocí třech šroubů M8x30. Ve vhodných krajových polohách řadicí páky na převodovce (vahadlo) umístíme kovové koncovky bovdenu do otevřených svěrek. Pro zařazený V. stupeň by vozík

bovdeny volby měl mít vůli asi 5 až 8 mm od konce lineárního vedení. Zařadíme tedy V. stupeň a vymezíme tuto vzdálenost. Kovovou koncovku bovdenu volby sevřeme pomocí svěrky volby. Následně kovovou koncovku bovdenu řazení sevřeme pomocí svěrky řazení tak, aby vzdálenost mezi touto svěrkou a pryžovou manžetou bovdenu byla asi 8 mm, viz následující obrázek 9. Pokud teď budeme ručně pohybovat vahadlem, tyto pohyby se budou zpětně přenášet přes bovdenu až k vozíkům na lineárních vedeních.

Poloha bovdenu vůči převodovce a robotu řazení lze ovlivnit dvěma způsoby:

1. mírou zasunutí kovové koncovky bovdenu do svěrek
2. posuvem *opěrného plechu* (poz. 11) vůči *základní desce* (poz. 6).

Zkoušením různých variant vzájemných posuvů, lze seřízení měnit. Po úspěšném nastavení jsou šrouby (poz. 19) dotaženy.



Obr. 9

**!!! UPOZORNĚNÍ:** Aktuátor servomotoru musí být vždy veden, nesmí se nechat volně otáčet. Ruční manipulace šroubováním je nepřipustná. Rotace aktuátoru v blízkosti mikrospínačů může způsobit jejich zničení.

Tab. 2 – přiřazení funkce jednotlivým servomotorům

Označení servomotorů	Co ovládá?
Mot1	řazení na zkoušené převodovce
Mot2	volbu na zkoušené převodovce
Mot3	řazení na technologické převodovce
Mot4	volbu na technologické převodovce

Veškeré ovládání robota řazení je prováděno pomocí **čelního panelu podprogramu pro FPGA** (*FPGA\_robot.vi*), viz obrázek 10. Vahadlem zařadíme zpátečku a provedeme seřízení motoru volby, resp. jeho mikropsínače (PŘ) a následně i mikropsínače (ST). Postup je svým způsobem iterační. Otáčením šroubu A–F doprava či doleva, viz obr. 2, **(jedna otáčka odpovídá 0,7 mm posuvu)** posuneme mikropsínač (PŘ) do předpokládané polohy, následně ručně nastavíme rychlost motoru na maximum a čekáme, až mikropsínač (PŘ) rozepne a motor tak zastaví. Vzdálenost, ze které motor najíždí na mikropsínač, musí být dostatečně dlouhá, aby bylo dosaženo plné rychlosti. Nyní motor sklopíme zpět a pokusíme se nasadit *mezikus* (poz. 8) na *čep se závitem* (poz. 14). Tento postup opakujeme, dokud se to nepodaří tak, aby v bovdenu nevznikala žádná zbytková síla, resp. *mezikus* musí jít nasadit na *čep se závitem* zcela volně. Mikropsínač (ST) seřídíme cca 10 mm za polohu pro V. (případně VI.) převodový stupeň, aby se nemohl účastnit běžného provozu a sloužil pouze jako pojistka pro vypnutí motoru. Seřízení mikropsínačů motoru řazení provedeme analogicky, a to na zařazeném III. a IV. převodovém stupni.

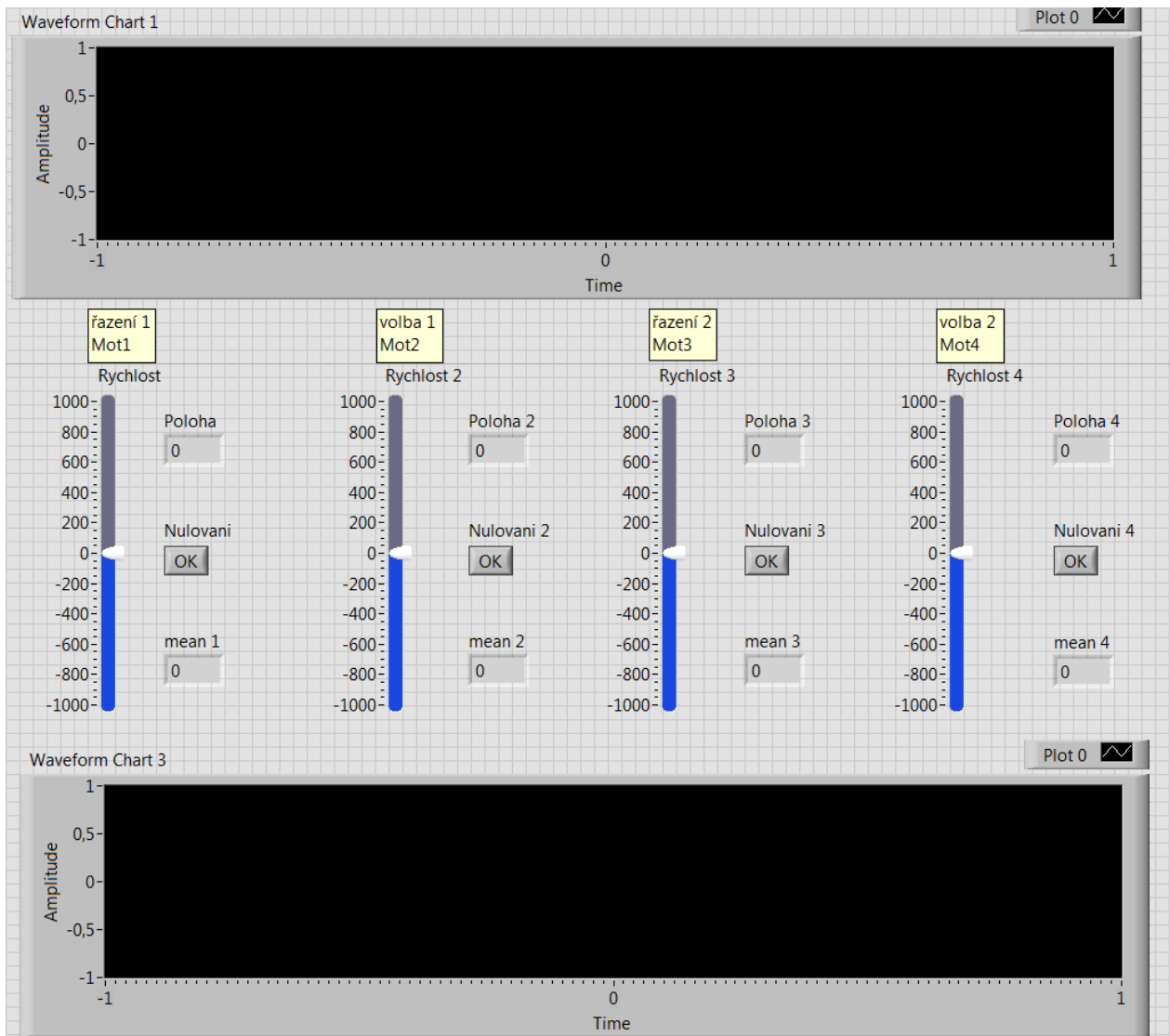
Nastavení enkodérů je opět velmi podobné. Nejprve popíši polohu neutrálu z pohledu bovdenu řazení. Motor řazení je v pracovní poloze, tedy je nasazen na čepu, a motor volby je odklopen. Ručně nastavíme plnou rychlost vzad pro motor řazení, čímž je zařazen III. převodový stupeň. Nyní ručně pomocí tlačítka na čelním panelu vynulujeme polohu enkodéru. Motor řazení odklopíme, abychom mohli vahadlem zařadit neutrál. Další postup je iterační a podobný postupu pro nastavení mikropsínačů. Ručně ovládáme rychlost tohoto motoru, dokud se nepodaří volně nasadit *mezikus* na *čep se závitem*. Až je to splněno, odečteme z čelního panelu už správnou hodnotu udávající polohu neutrálu a poznamenejeme si ji.

Nyní popíši nastavení enkodéru motoru volby. Motor řazení je odklopen a motor volby je v pracovní poloze. V tomto okamžiku nesmí být zařazen žádný převodový stupeň. Ručně nastavíme plnou rychlost vpřed pro motor volby, čímž je zvolena řadičí vidlička zpátečky. Nyní ručně pomocí tlačítka na čelním panelu vynulujeme polohu enkodéru. Další postup už je analogický postupu předcházejícímu a musí být proveden celkem třikrát, viz tabulka výše.

Výsledné polohy, do nichž mají motory podle enkodérů najíždět, jsou také shrnuty v tabulce 2 stejně jako hodnoty zařazených převodových stupňů.

Seřízení druhého robota je shodné.

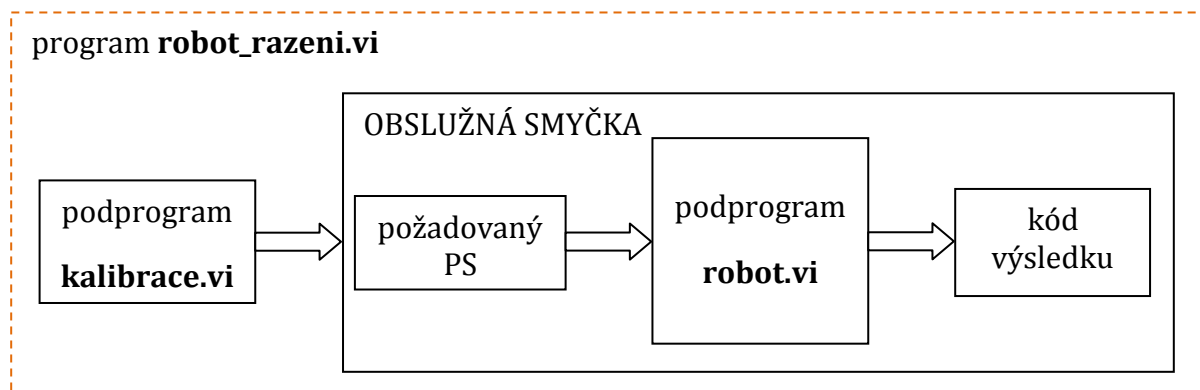




Obr. 10

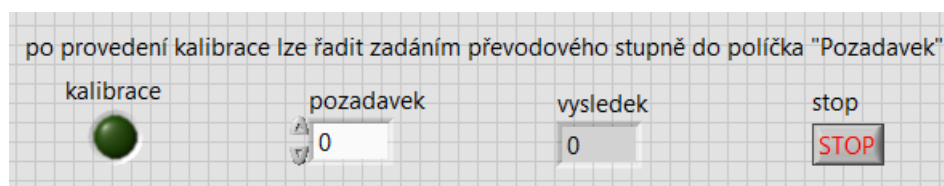
## Podprogramy

V následujícím textu budou uvedeny pouze údaje relevantní pro používání robota.



Obr. 11

Abych mohl ověřit, že podprogram robot.vi funguje, musel jsem vytvořit nový hlavní program (robot\_razeni.vi), jehož schéma je znázorněno na obrázku 11 a jeho hlavní panel na obrázku 12.



Obr. 12

Požadovaný převodový stupeň je nutné zadávat do okna „pozadavek“ podle následující tabulky 3. Robot následně provede zařazení (nebo resp. pokus o něj) a jako výstup odešle číselnou hodnotu do okna „vysledek“. Tato koncepce je vysvětlena v kapitole Výstupy z podprogramu robot.vi – kód výsledku.

Tab. 3 – přiřazení proměnné „pozadavek“ k požadovanému PS

Požadovaný PS	1	2	3	4	5	6	N	R
„pozadavek“	1	2	3	4	5	6	0	7

Nyní k jednotlivým podprogramům:

### **kalibrace.vi**

Inkrementální charakter snímače v enkodéru neumožňuje odečítat polohu motoru v absolutním slova smyslu, což má za následek, že při spuštění řídicí aplikace software nemá informaci, v jaké poloze se jednotlivé motory nacházejí. Proto při každém novém spuštění musí dojít k tzv. kalibraci.

Prvním pohybem je pohyb motorů řazení plnou rychlostí vzad, čímž dojde k zařazení některého lichého převodového stupně nebo zpátečky. Do údaje enkodéru o poloze motorů řazení je zapsána nula. Následuje pohyb týchž motorů do neutrálu pomocí speciálního podprogramu nazvaného „vykonavatel“ (dále jen „vykonavatel“), jehož popis bude uveden níže. Nyní se už mohou rozjet motory volby plnou rychlostí směrem ke zpátečce. Tak jako předtím se samy zastaví po rozpojení mikropsínače. Do údaje o poloze motorů volby je zapsána nula. Opět následuje pohyb týchž motorů do neutrálu pomocí „vykonavatele“. Na závěr sekvence se zkontroluje, zda neutrál je dosažen s požadovanou přesností, kterou jsem zvolil na  $\pm 5$  pulsů enkodéru na všech čtyřech motorech.

## **DOPORUČENÍ**

*Obsluha by po provedení kalibrace, resp. před prvním řazením pomocí robota, měla fyzicky zkontrolovat, zda je na obou robotech zařazen neutrál. To lze zjistit tak, že servomotory je možno snadno podklopit a také tím, že při ručním protáčením horního hřídele stavu zůstává spodní hřídel nehybný. Pokud by byl z jakéhokoli důvodu posunut kalibrační koncový spínač, pak by se kalibrace provedla chybně a SW by to nemohl zjistit.*

*Rovněž doporučuji spouštět kalibraci z neutrálu, tzn. vypínat řídicí program celého stavu právě při zařazeném neutrálu, aby následně mohl být z tohoto stavu znovu spuštěn. Pokud by před kalibrací byl z nějakého důvodu zařazen lichý převodový stupeň (kalibrační) ne úplně přesně, motory řazení by se při jejich kalibraci pouze vrátily do neutrálu a tím by byl do řazení vnesen určitý offset.*

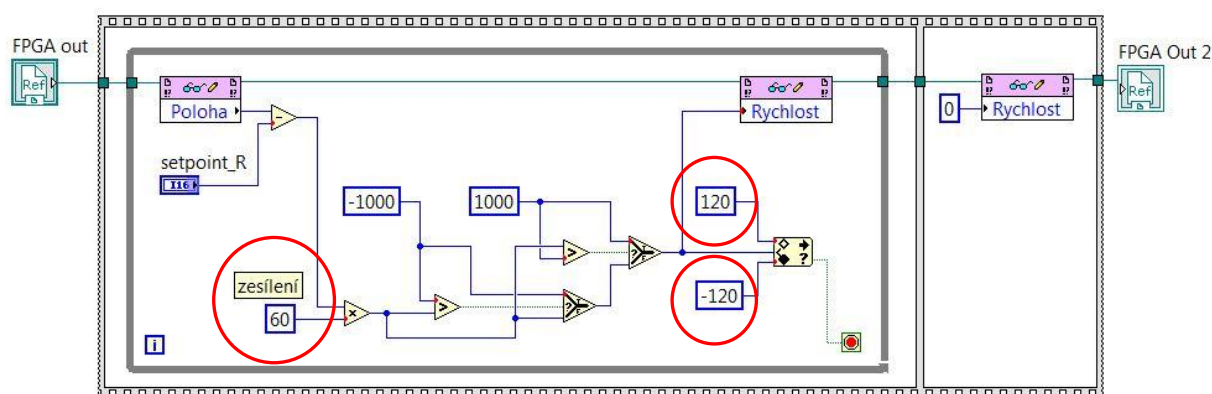
### „vykonavatel“

„Vykonavatel“ je krátký program, jehož úkolem je přemístit aktuátor z výchozí polohy do polohy požadované (setpoint) pouze ze znalostí údajů z enkodéru. Programů „vykonavatel“ existuje celkem šest, jejich použití shrnuje tabulka 4.

**„R“ v názvu znamená řazení, „V“ znamená volbu, druhé „V“ v pořadí symbolizuje použití pro volbu V. stupně a „2“ na konci je určena pro robota u technologické převodovky.**

Tab. 4 – použití programů „vykonavatel“

Název programu.vi	Motor	Co obsluhuje	Zesílení
vykonavatel_R	Mot1	vyřazení do lokálního neutrálu	60
vykonavatel_R_2	Mot3	vyřazení do lokálního neutrálu	60
vykonavatel_V	Mot2	volba ŘV I., II. a ŘV III., IV.	60
vykonavatel_V_2	Mot4	volba ŘV I., II. a ŘV III., IV.	60
vykonavatel_V_V	Mot2	volba ŘV V.	90
vykonavatel_V_V_2	Mot4	volba ŘV V.	90



Obr. 13 – vykonavatel\_R pro Mot1

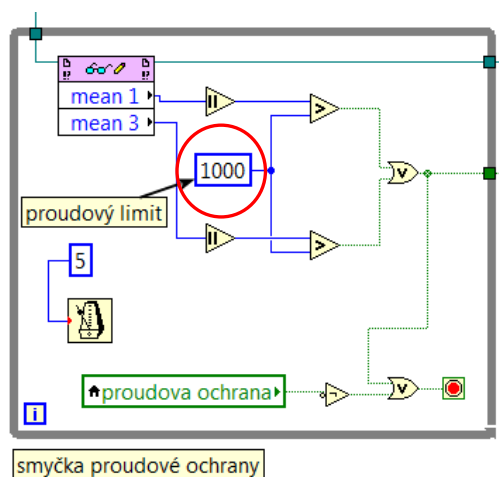
Nastavitelný parametr tohoto programu je jeden, a to zesílení P-regulátoru při zpomalování servopohonu do polohy setpoint. Pokud požadujeme razantnější pohyb motoru, je třeba použít zesílení vyšší a nechat tak více působit dynamiku systému, zatímco pro přesné najetí do požadované polohy zesílení nižší. Použití vyššího zesílení má však za následek, že motor požadovanou polohu může přejet. Horní a dolní limity rychlosti pro ukončení smyčky, viz obrázek 13, by měly být celými násobky zadaného zesílení.

### robot.vi

Tento podprogram má za úkol provést vlastní řazení od okamžiku načtení požadavku až do výpočtu kódu výsledku. Vstupy do něj jsou tři – booleovská hodnota kalibrace (T/F), požadavek na řazení a reference pro komunikaci s FPGA, výstupy jsou dva – kód výsledku v proměnné výsledek, což je celé číslo v intervalu 0 až 255, a výstup reference pro komunikaci s FPGA.

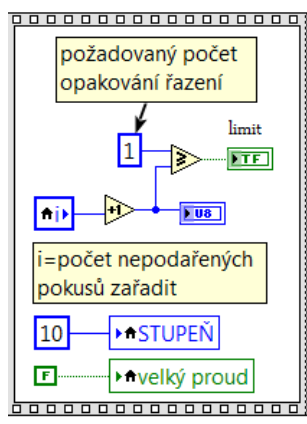
#### ***Jaké parametry lze měnit uvnitř programu robot.vi?***

1. Číselné hodnoty z tabulky 1. Jsou to vlastně jednotlivé body setpoint pro podprogramy „vykonavatel“. Nutno upravit u každého použití tohoto podprogramu, např. při změně typu zkoušené převodovky.
2. Proudový limit ve smyčce proudové ochrany. Program robot obsahuje sledování dovolené hodnoty síly v bovdenu řazení během samotného řazení, tedy při pohybu z lokálního neutrálu směrem k požadovanému převodovému stupni. Tato síla je jednoznačně přiřaditelná k proudu, který právě protéká servomotorem dle vztahu  $I_{limit} = 3,75 \cdot F + 500$  [mA]. **Tento limit je nastaven na 1000 mA, což odpovídá cca 150 N v bovdenu.**

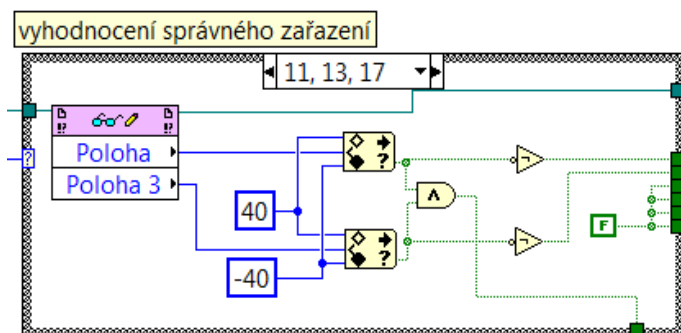


Obr. 14

3. Požadovaný počet opakování řazení. Pokud je během prvního pokusu o zařazení detekován nadproud v některém ze servopohonů, řazení je okamžitě přerušeno. Následuje vyřazení do neutrálu na obou robotech a opětovný pokus o zařazení. Kolik bude těchto opětovných pokusů o zařazení, definuje konstanta v červeném kruhu na obrázku 15. Toto nastavení budiž výchozí, tedy opakování bude jen jedno.



Obr. 15



Obr. 16

4. Limity pro vyhodnocení správného zařazení. Na konci každého řazení je provedeno porovnání počtu načtených pulsů enkodéru s hodnotou teoretickou pro dané polohy, viz tabulka 1. Jednotlivé případy a výchozí nastavení limitů shrnuje tabulka 5. Na obrázku 16 je znázorněn případ „11, 13, 17“ jako výstřižek z programu.

Tab. 5 – limity pro kontrolu dle teoretické hodnoty

Případ	Teoretická hodnota dle tab. 1	Horní limit	Dolní limit
Default, 10	odeslána hodnota True		
11, 13, 17	0	40	-40
12, 14	-1100	-1060	-1140
15	není	-10	-80

### Výstup z podprogramu robot.vi – kód výsledku

Pro bezproblémový běh řídicí aplikace a pro snazší hledání případných chyb tento program obsahuje diagnostikování poruchových stavů, které je ukryto právě v kódu výsledku.

Koncepce byla zvolena následující. Vybral jsem osm různých poruch, které mohou nastat dle možností odečítání různých hodnot z robotů řazení. Porucha budiž definována booleovskými hodnotami T/F. Z těchto osmi hodnot jsem sestavil pole o osmi prvcích, které následně převádím do desítkové soustavy. Přičemž pozice 1 odpovídá nejméně významnému bitu (LSB), zatímco pozice 8 nejvíce významnému bitu (MSB). Následující tabulka 6 přiřazuje jednotlivé chyby k pozicím v poli. Výsledný kód výsledku je tedy dán součtem všech dílčích hodnot. Libovolná kombinace chyb je tak jednoznačně určena a na základě této informace může dále řídicí program celého stavu patřičně reagovat. **V případě, že řazení proběhlo úspěšně, je kód výsledku nula.**

Tab. 6 – přiřazení chyby k pozicím v poli

Pozice bitu č.	Obecný popis chyby	Konkrétní místo	Odpovídající hodnota
1	počet načtených pulsů enkodéru neodpovídá hodnotě teoretické, což uvnitř programu znamená <i><u>fatal error</u></i> = T	Mot1 při řazení I, III nebo R	$2^0 = 1$
2		Mot3 při řazení I, III nebo R	$2^1 = 2$
3		Mot1 při řazení V	$2^2 = 4$
4		Mot3 při řazení V	$2^3 = 8$
5		Mot1 při řazení II nebo IV	$2^4 = 16$
6		Mot3 při řazení II nebo IV	$2^5 = 32$
7	detekován nadproud při posledním pokusu o zařazení v rámci jednoho požadavku, <i><u>velký proud</u></i> = T	Mot1	$2^6 = 64$
8		Mot3	$2^7 = 128$

### Požadavky na řídicí SW celého stavu

Vzhledem k tomu, že uzavřený zkušební stav včetně robotů by měl fungovat plně automaticky, je nutné pro řídicí program definovat sekvenci úkonů po obdržení jakéhokoli výsledku z robota řazení. Tyto postupy shrnuje tabulka na obrázku 17.

### Výměna převodovek stejného typu (viz obr. 9)

#### Demontáž:

1. odjistit obě svěrky
2. povolit šrouby na držáku bovdenů
3. vysunout oba kovové konce bovdenů z otevřených svěrek
4. zavřít obě svěrky
5. odmontovat koncovku řazení se svěrkou
6. odmontovat plastovou kolébku volby se svěrkou
7. odšroubovat vahadlo z převodovky

#### Montáž:

1. přišroubovat vahadlo k převodovce
2. připevnit plastovou kolébku volby se svěrkou
3. nasadit koncovku řazení se svěrkou na vahadlo
4. ručně zařadit vahadlem III. převodový stupeň
5. zařadit na robotu pomocí jeho ovládacího programu také III. převodový stupeň
6. zasunout oba kovové konce bovdenů do otevřených svěrek
7. dotáhnout šrouby na držáku bovdenů
8. zajistit (zacvaknout) obě svěrky

## Odpovědi robotů na požadavek a nároky na řízení stavu

Kód výsledku	co to znamená?	předpoklad	řešení												
<b>0</b>	nebyl indikován nadproud a aktuátor dosáhl požadované polohy	není zařazen požadovaný PS na obou převodovkách	<p><b>možné situace</b>  <b>A:</b> oba PS zařazeny – vše OK  <b>B:</b> Pož. PS + neutrál – rychlost nebyla správně zařazena nebo vypadla  <b>C:</b> Pož. PS + jiný PS – byla zařazena nesprávná rychlost</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;"><b>pro A</b></td> <td style="width: 25%;">Moment je konst. (0 Nm)</td> <td style="width: 25%;">Předepnout na 10 Nm</td> <td style="width: 25%;">moment se nemění (10 Nm)</td> </tr> <tr> <td><b>pro B</b></td> <td></td> <td></td> <td>moment se nezačne zvyšovat</td> </tr> <tr> <td><b>pro C</b></td> <td>Původní nulový moment se zvyšuje</td> <td>dle barvy</td> <td>dle barvy</td> </tr> </table> <p>pootočení vstupním hřídelem pomocí hlavního el. motoru bez předpětí</p>	<b>pro A</b>	Moment je konst. (0 Nm)	Předepnout na 10 Nm	moment se nemění (10 Nm)	<b>pro B</b>			moment se nezačne zvyšovat	<b>pro C</b>	Původní nulový moment se zvyšuje	dle barvy	dle barvy
<b>pro A</b>	Moment je konst. (0 Nm)	Předepnout na 10 Nm	moment se nemění (10 Nm)												
<b>pro B</b>			moment se nezačne zvyšovat												
<b>pro C</b>	Původní nulový moment se zvyšuje	dle barvy	dle barvy												
<b>64 nebo 128</b>	byl "i" krát indikován nadproud	<b>možná příčina</b> problém na převodovce	je vyřazeno na N nelze zařadit 2x totéž může být řazen jiný PS než požadovaný, např. neutrál												
<b>1 až 63</b>	aktuátor nedosáhl požadované polohy	posunul se koncový spínač vlivem provozu vypadl kabel datový vypadl kabel napájení	pootočení vstupním hřídelem pomocí hlavního eM bez předpětí opětovný pokus o zařazení požadovaného PS cyklus možno několikrát opakovat, jinak nutný zásah obsluhy												
<b>ČASOVÝ LIMIT PRO ZPĚTNOU VAZBU Z ROBOTY</b>	ovládací program stavu nedostal v daném limitu žádnou informaci z roboty	nedokončení podprogramu „vykonavatel“ vypadl kabel datový vypadl kabel napájení	může být řazen jiný PS než požadovaný, např. neutrál opětovný pokus o zařazení požadovaného PS cyklus možno několikrát opakovat, jinak nutný zásah obsluhy												

Obr. 17