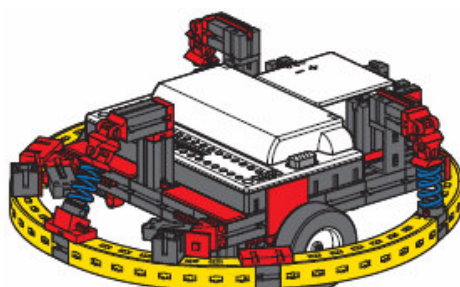


fischertechnik 

ROBO MOBILE SET



učebnice

Obsah

K čemu potřebujeme roboty?	3
Roboti od firmy Fischertechnik	4
Akční členy	4
Senzory	5
Rozhraní ROBO	5
Software ROBO Pro	6
Napájecí zdroj	6
Experimentování	6
První kroky	6
První jednoduchý robot	7
Inteligentní robot s koly	9
Základní model	9
Sledovač světla	11
Stopař	13
Robot detekující překážky	14
Sledovač světla s detekcí překážek	16
Robot s detekcí hran	18
Kráčející robot	20
Možná rozšíření	22
Infračervené dálkové ovládání	22
Datové spojení ROBO RF	22
Rozšíření ROBO I/O	23
Odstraňování problémů	24



K čemu potřebujeme roboty?

■ Karel Čapek poprvé použil slovo robot v románu Golem z roku 1923. Tato uměle vytvořená postava měla převzít těžkou práci člověka, ale zároveň i s jeho schopnostmi.

Ve 30. a 40. letech minulého století se z robota stal spíše automat. Dnes se můžeme ohlédnout a usmát se různým pokusům znázornit jej s některými lidskými rysy, jako např. hlavou s blikajícími světly místo očí apod. Tyto stroje projevovaly jen málo inteligence či pohyblivosti. Jelikož má princip řízení velký vliv na robotiku, stal se návrh robotů s nástupem integrovaných obvodů realističtější. Dnes je inteligence robota spíše otázkou výzkumu a hledání informací.



■ První slibný přístup k řešení problému nabídla tzv. kybernetika. Pojem „kybernetika“ je odvozen od řeckého slova kybernetes. Kybernetes byl navigátorem řeckých lodí a měl za úkol stanovení polohy lodí a navigaci k cíli.

Samozřejmě se počítalo se s tím, že kybernetika učiní roboty „inteligentními“. Ale jak vypadá takové inteligentní chování?

Měli bychom se to pokusit vysvětlit na malém příkladu. Každý asi viděl chování můry ve světle lampy. Můra detekuje zdroj světla, letí k němu, a na poslední okamžik se pokouší vyhnout nárazu do lampy. Je jasné, že v případě simulace tohoto chování musí můra detekovat zdroj světla, stanovit směr a pak k němu letět. Tato schopnost je založena na instinktivním inteligentním chování hmyzu.

Nyní se pokusíme aplikovat tyto schopnosti na technický systém. Musíme detekovat zdroj světla (optickými senzory), provést pohyb (pomocí motorů) a musíme smysluplně propojit detekci a pohyb (programem).



■ Tento experiment provedl v 50. letech minulého století angličan Walter Grey.

S pomocí jednoduchých senzorů, motorů a elektronických obvodů vytvořil řadu kybernetických zvířat, která simulovala velmi specifické chování, např. právě můry. Fotografie ukazuje repliku „kybernetické“ želvy, která je vystavena v Smithsonian Museum v USA ve městě Washington.

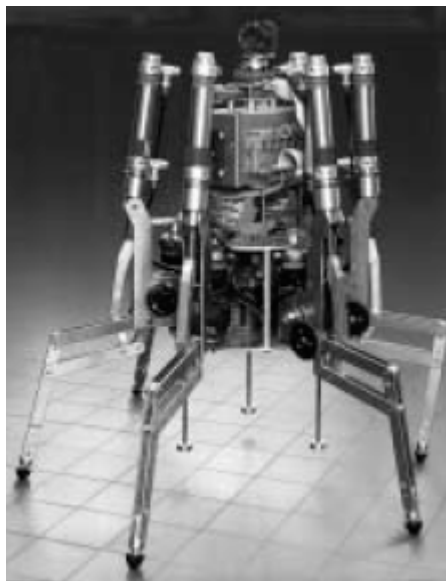
Na základě těchto myšlenek vytvoříme podobné vzorky chování našich robotů ve formě programů.

■ Ale k čemu potřebujeme mobilní roboty? Pokusme se aplikovat chování naší imaginární můry na technické zařízení. Jednoduchým příkladem je hledání světla. Zdroj světla změníme na světlou pásku, která bude nalepena na podlahu sloužit jako cestička, a senzory vpředu otočíme ne dopředu, ale dolů. Pomocí takových cestiček umí mobilní robot najít svou cestu například ve skladu. Další informace v konkrétních místech podél čáry, např. ve formě čárového kódu, říkají robotu, aby v daných místech provedl další akce, např. zvednutí nebo položení palety. Takový robotický systém už existuje. Ve velkých nemocnicích je často potřeba pokrýt značné vzdálenosti pro transport spotřebního materiálu, jako jsou třeba prostěradla. Přesouvání takového materiálu je časově i finančně náročné a často představuje těžkou fyzickou práci. Provádění takových úkolů navíc omezuje čas potřebný pro péči o pacienty.





■ V posledních letech se vědci začali zabývat jinou formou pohybu, která je zcela běžná v přírodě – chůzí nebo během. Byli vyvinuti roboti, kteří se umí pohybovat pomocí nohou. Příkladem šestinohého kráčejícího robotu je elektropneumatický robot „Achille“, vyvinutý Královskou válečnou akademií v Bruselu. Má jednu kameru nahoře a jednu na každé ze šesti nohou a je navržen tak, aby byl schopen reagovat mechanicky na vyvýšené nebo snížené překážky (objekty nebo díry). Takové kráčející stroje mohou být nasazeny tam, kde kolové nebo pásové podvozky nemají šanci, např. v těžkém a lehkém terénu, šplhání přes překážky, stoupání po schodech, překonávání příkopů nebo při práci v nepříístupných nebo nebezpečných oblastech v jaderných elektrárnách, důlních štolách nebo při záchranných operacích. Snadno zjistíte, že mobilní roboty hrají v moderní společnosti důležitou roli.



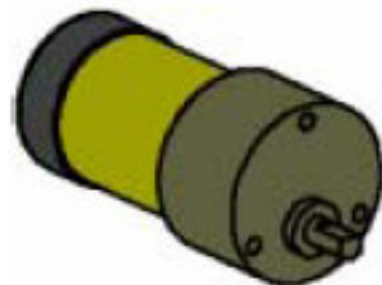
Roboti firmy fischertechnik

■ Jak je možno vyrobit robot ze stavebnice Fischertechnik? K vytvoření robotu potřebujeme senzory (například tlačítka) a pohony (například motory), ale také mnoho mechanických součástek pro vytvoření modelu. Fischertechnik ROBO Mobile Set je ideálním základem. V této stavebnici jsou obsaženy následující senzory a pohony:

Akční členy

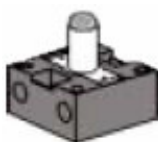
Výkonný motor:

Mobilní roboty pohání dva výkonné stejnosměrné motory (9V/2,4W) s převodovkou s převodovým poměrem 50:1 (což znamená, že na každých 50 otáček motoru připadá jedna otáčka výstupní hřídele).



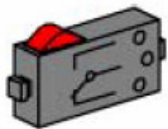
Žárovka s čočkou:

Tato jasně svítící žárovka (9VDC/150mA) umožňuje výstup jednoduchých světelných signálů. Obsahuje integrovanou čočku usměrňující emitované světlo. Při nasměrování světla dopředu můžete pomocí světelného senzoru (fototranzistoru) vytvořit světelnou závoru a měřit odražené světlo. Lampu můžete použít také k zobrazení určitých stavů nebo ke generování varovných zpráv ve formě různého blikání. V této stavebnici je žárovka používána společně se dvěma



fototranzistory jako speciální senzor k rozpoznávání čáry.

Senzory (snímače)

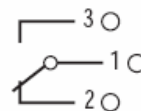


■ Příkladem digitálního senzoru je tlačítko. Digitální hodnoty mají pouze 2 různé stavy. Tyto stavy jsou označovány jako logická 0 nebo 1. Logická 0 tlačítka označuje, že kontakty jsou rozpojené, a logická 1, že jsou spojené.

Tlačítko ze stavebnice Fischertechnik je navrženo jako přepínač. Proto má tři vývody. Když je červené tlačítko stisknuto, je mechanicky aktivován přepínač spojující vývody 1 a 3. Ve stejném okamžiku je přerušeno spojení mezi vývody 1 a 2, které byly v klidovém stavu spojeny. Takto je možné vybrat jednu z počátečních pozic:

Spojeno v klidovém stavu (vývody 1 a 2 jsou spojené)

Rozpojeno v klidovém stavu (jsou spojené vývody 1 a 3).



Fototranzistor může být použit jako digitální i analogový senzor. V prvním případě pro rozpoznávání ostrých přechodů mezi světlem a tmou, například na vyznačené čáře. Ale je možné zjišťovat také množství světla. V takovém případě fototranzistor funguje jako analogový senzor. Analogové hodnoty se mohou pohybovat mezi jejich hraničními hodnotami. Tyto hodnoty musí být převedeny na odpovídající numerické hodnoty, aby mohly být zpracovány počítačem.



Fototranzistory patří mezi takzvané polovodičové součástky a jejich elektrické charakteristiky jsou závislé na intenzitě světla. Všichni ví, že solární články používají sluneční světlo ke generování elektrické energie. Fototranzistor je možno chápat jako kombinaci miniaturního solárního článku a tranzistoru. Světelné impulzy (fotony) přijaté fototranzistorem generují velmi malý proud, který je dále zesílen tranzistorem.

Poznámka:

Před připojením fototranzistoru se ujistěte, že má správnou polaritu: červená označuje kladnou elektrodu a jeho maximální napětí je 30V.

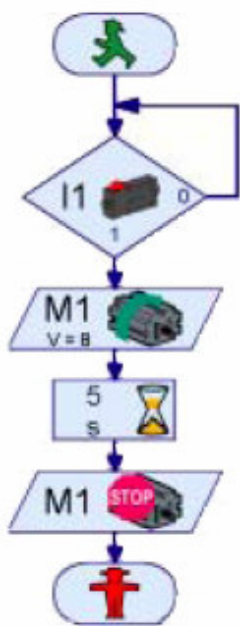
ROBO Interface

■ Pomocí rozhraní ROBO můžeme připojit různé senzory a akční prvky a interpretovat je. Rozhraní ROBO kromě 8 digitálních vstupů nabízí také několik analogových vstupů. Například hodnota elektrického odporu v rozmezí 0 až 5,5 kΩ připojená na vstupy AX a AY je převedena na číselnou hodnotu mezi 0 a 1023. Také mohou být změřeny a dále zpracovány hodnoty světelného senzoru, jako např. fototranzistoru. Na analogových vstupech A1 a A2 je možno měřit napětí mezi 0 a 10V stejnosměrného napětí.

Nejdůležitějším úkolem rozhraní je logické spojení vstupních hodnot. Rozhraní k tomu potřebuje program. Tento program bude na základě vstupních dat a signálů sensorů generovat příslušná výstupní data, řídicí signály motorů atd. Díky rozhraní ROBO máme dostatečnou výpočetní sílu pro návrh i těch nejsložitějších programů.



Software ROBO Pro



■ Grafické programovací rozhraní poskytuje nejefektivnější způsob vytváření potřebných programů pro rozhraní ROBO. Termín „programovací rozhraní“ označuje software umožňující vytvářet programy velmi pohodlným způsobem, pomocí grafických symbolů. Počítač rozhraní ROBO umí ve skutečnosti provádět pouze příkazy v tzv. strojovém kódu. Jsou to velmi jednoduché řídicí struktury, které jsou velmi obtížné pro začátečníky. Proto software ROBO Pro používá grafické prvky, které jsou později přeloženy do jazyka, který může být rozhraním proveden.

Napájecí zdroj

■ Jedinou věcí, již kromě ROBO Mobile Set potřebujete, je Accu Set. Obsahuje baterie sloužící jako mobilní zdroj elektrické energie pro robotické modely a speciální nabíječku baterií. Nejlepší bude baterie hned nabít, aby byly plně nabitě, než začnete experimentovat.



Experimentování

■ Provedeme vás krok za krokem fascinujícím světem mobilní robotiky. Začneme jednoduchou kontrolou základních funkcí rozhraní a senzorů. Pak budeme stavět jednoduché modely s konkrétní funkcí a později se pokusíme o čím dál složitější systémy.

V určitý moment si možná budete říkat, že vytváření vašich vlastních programů je příliš komplikované nebo trvá velmi dlouho. Do rozhraní můžete načíst také hotové vzorové programy a používat je k řízení robotu. Na konci této příručky je kapitola o odstraňování problémů, takže nezapomejte, když se objeví nějaké chyby.

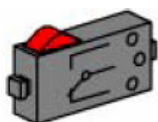
Je velmi důležité, abyste byli obzvláště pozorní při konstruování a počáteční práci robotů. Při zapojování elektrických komponent se držte specifikací a dvakrát či třikrát zkontrolujte správnost zapojení. Co se týká mechanické konstrukce, dávejte pozor na hladký chod a malé vůle v převodech a spojích. Vytváření vlastních programů definujících nové chování je jen na vás a vaši kreativitu. Jste omezeni pouze množstvím paměti a výpočetní silou vašeho hardware. Následující příklady vás mohou inspirovat.

První kroky

■ Nyní jsme si prošli teorii a začneme experimentovat. Někteří z vás už možná chtějí začít, a někteří dokonce chtějí začít velkým krácejícím robotem. To je samozřejmě možné a když se budete pečlivě držet návodu, podaří se vám to na první pokus.

Ale co když to nebude fungovat? V takovém případě musíte při hledání chyby postupovat systematicky. Ale nejdříve zkontrolujte spojení mezi počítačem a rozhraním.

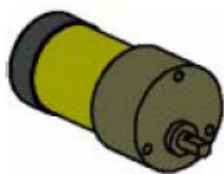
Kapitoly 1 a 2 manuálu k softwaru ROBO Pro popisují, jak nainstalovat řídicí software do PC a jak připojit rozhraní. Pomocí testů rozhraní budeme testovat různé senzory a akční prvky.



Senzor s tlačítkem

Můžeme například připojit senzor s tlačítkem na digitální vstup I1 a zjišťovat, jak se mění stav

vstupu, když je stisknuto tlačítko.



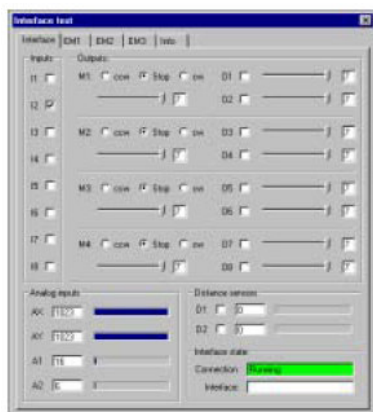
Silný motor

Výstupy budeme testovat připojením motoru k výstupům pro řízení motorů, např. M1. Pomocí levého tlačítka můžeme nyní začít točit motorem a posuvníkem měnit jeho otáčky.

Fototranzistor

K otestování analogových vstupů AX můžete použít fototranzistor.

Při připojování motoru nebo tlačítka polarita nehraje žádnou roli (motor se bude v nejhorším případě otáčet nesprávným směrem), ale fototranzistor je pro jeho správnou funkci důležité zapojit správně. Kontakt tranzistoru s červeným označením by měl být připojen k červenému konektoru a druhý kontakt k zelenému konektoru. Zelený konektor patří do zdířky vstupu AX, který se nachází na kraji rozhraní. Červený konektor patří do zdířky AX umístěné dál od kraje (pozor – při připojování fototranzistoru k digitálnímu vstupu I1-I8 musí být červený konektor zapojen do zásuvky umístěné blíže ke kraji).



Nyní můžeme měnit intenzitu světla dopadajícího na fototranzistor. Bude se při tom měnit hodnota modrého pruhu AX. Jestliže se indikátor nehýbe ze své maximální pozice, měli byste zkontrolovat připojení fototranzistoru. Pokud naopak indikátor zůstává na nule, i když na fototranzistor posvítíte, je asi osvětlení v místnosti příliš jasné. Pozice pruhu se bude při zakrytí fototranzistoru měnit.

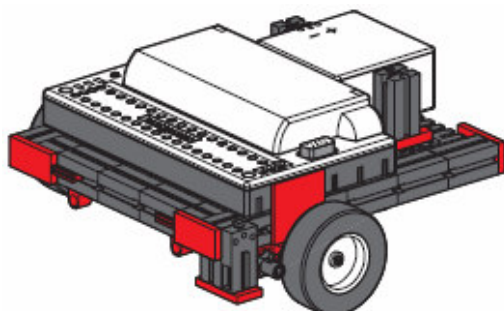
Ještě jednou k barvám na fototranzistoru: při sestavování budeme červený konektor připojovat vždy k červenému konektoru a zelený k zelenému. Jelikož je správná polarita důležitá, budeme vždy používat červený drát pro kladné pole a zelený pro záporné pole. I když se to může zdát příliš pedantské, správné použití barev usnadňuje hledání chyb.

Do oblasti robotiky se dostaneme prostřednictvím jednoduchého programu. Program „Řídící systém garážových vrat“, vysvětlený v kapitole 3 manuálu ROBO Pro, možná nemá nic společného s mobilní robotikou, ale je skvělý pro seznámení se se softwarem ROBO Pro. K vytvoření tohoto programu je potřeba k rozhraní připojit pouze motor a tři tlačítkové senzory. Vše ostatní je popsáno podrobně v manuálu k software.



První jednoduchý robot

- Po otestování rozhraní a řídicího systému garážových vrat se konečně pustíme do práce na prvním robotu. Podle manuálu postavíme model „Simple Robot“ se dvěma motory. To půjde celkem rychle a snadno, jelikož tento model obsahuje pouze několik věcí. Motory připojíme k výstupům M1 a M2.



Spustíme software ROBO Pro a vytvoříme nový program (FILE – NEW). ROBO Pro nabízí různé úrovně obtížnosti práce s ním. Můžete je nastavit v nabídce ROBO Pro LEVEL. V tuto chvíli nám bude stačit Level 1.

Objeví se prázdná pracovní plocha a na levé straně okno s prvky. Zde si můžete vybrat různé prvky programu a umístit je do pracovní oblasti

pomocí levého tlačítka myši. Pomocí pravého tlačítka myši můžete měnit jejich vlastnosti.



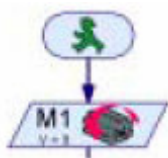
Úkol 1 (úroveň 1)

Náš jednoduchý robot by měl jet 5 sekund rovně, pak se 2 sekundy točit na místě a poté se zastavit.

Tipy:

Budeme společně krok za krokem programovat první robot:

- Začneme malým zeleným panáčkem GO, jenž představuje začátek programu.
- Pak vezmeme symbol motoru z okna s prvky a vložíme jej pod prvek start, což způsobí automatické nakreslení propojovací čáry. V okně s vlastnostmi nastavíme výstup motoru na „M1“ a směr rotace na „ccw“ a pak potvrdíme tlačítkem OK.



- Stejným způsobem umístíme další symbol motoru pod ten první a nastavíme motor 2.
- Pro čekání po určitou dobu se používá prvek Time Delay, který umístíte pod symbol motoru a nastavíte čas na 5 sekund.
- Poté nastavíme M2, aby se otáčel v opačném směru (cw), pak počkáte 2 sekundy a nakonec oba motory vypnete. Program se ukončuje symbolem End s malým červeným panáčkem STOP. Obrázek ukazuje dokončený vývojový diagram programu.

Pokud si nejste jisti, zda jste vše provedli správně, můžete porovnat svůj program s již hotovým vzorovým programem. V tom případě nejprve uložte svůj program a pak otevřete soubor Simple Robot 1.rpp uložený v adresáři se vzorovými programy ROBO Pro (implicitně v adresáři C:\Program Files\ROBO Pro\Sample Programs\ROBO Mobile Set).

Pokud je vše v pořádku, můžete program uložit do rozhraní. Po klepnutí na tlačítko Download se objeví nové okno. V něm zvolíte, že program má být uložen do paměti FLASH 1 a že by měl být po uložení do paměti spuštěn.

Hned po uložení do paměti se model rozjede rovně, pak se bude chvíli otáčet na místě a nakonec se zastaví. Chcete-li program spustit znovu, stiskněte krátce tlačítko Prog na rozhraní. Dioda LED Prog1 bude blikat po celou dobu provádění programu. Po jeho ukončení zůstane rozsvícená. Program zůstane v paměti FLASH v rozhraní i po přerušení napájení. Můžete si to vyzkoušet odpojením baterií. Po opětovném připojení baterií zvolte uložený program - mačkejte tlačítko Prog dokud se nerozsvítí LED Prog1. Program spustíte jednoduše opětovným stiskem tlačítka.

Náš robot toho ale přece může dělat více, ne? Co si úkol trochu rozšířit?



Úkol 2 (úroveň 1)

Aby se robot nezastavil hned po 7 sekundách, naučíme jej tancovat.

- Necháme jej jet rovně, zatočit doprava, zatočit doleva a couvat po různou dobu a různou rychlostí speeds.
- To by mělo pokračovat dokud program nebude zastaven tlačítkem Prog na rozhraní.

Tipy:

- Jednoduše ponechte opačnou polaritu motorů, aby robot jel požadovanými směry.
- Rychlost motorů je možno nastavit mezi 1 a 8 v okně vlastností pro každý motor. Jestliže se budou M1 a M2 otáčet stejným směrem, ale různými rychlostmi, robot bude zatáčet.
- Nakreslete propojující čáru z konce posledního prvku programu k čáře vedoucí k prvnímu prvku, aby se program prováděl opakovaně.
- Hotový příklad najdete v souboru Simple Robot 2.rpp.

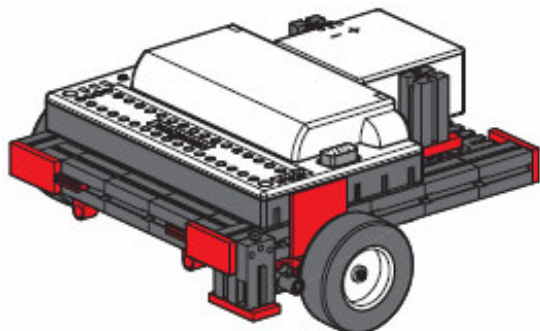
Gratulujeme, nyní jste postavili svůj první robot a sami jste jej naprogramovali. Není nijak zvlášť inteligentní, protože neumí rozpoznávat překážky a když jej nebudete hlídat, spadne ze stolu. Ale to se časem změní.



Inteligentní robot s koly

■ Roboti potřebují k poznávání svého okolí senzory. Následující modely ukazují několik různých mobilních robotů, na nichž si můžete vyzkoušet různé senzory. Je nezbytné, aby byly propojené interní stavy robotu, např. měření ujeté vzdálenosti počítáním pulsů z enkodérů, a externí signály, např. sledování světla nebo čáry. Pro každý model byly vytvořeny různé úkoly. Jsou navrženy tak, aby vás inspirovaly a seznámily s podstatou úkolu. Programy pro každý úkol najdete v adresáři ROBO Pro, v podadresáři \Sample Programs\ROBO Mobile Set\. Ale klidně můžete pro dané modely vytvářet vlastní úkoly. Po skončení následujících příkladů vás určitě ještě něco napadne.

Základní model



■ Ve srovnání s naším prvním jednoduchým modelem je základní model stabilnější a robustnější. kromě toho obsahuje 2 senzory pro měření ujeté vzdálenosti, skládající se z tlačítkového senzoru a enkodérového kolečka. Enkodérové kolečko je připevněno ke hřídeli motoru a aktivuje tlačítko čtyřikrát za každou otáčku motoru. Tento model slouží jako základ pro jiné modely mobilních robotů.

Sestavte základní model podle stavebního návodu. Při stavbě buďte velmi pečliví. Až bude mechanická část konstrukce hotová, zkontrolujte hladký chod obou motorů jejich přímým připojením k bateriím bez použití rozhraní.



Úkol 1 (úroveň 1)

- Naprogramujte rozhraní tak, aby robot jel 40 pulsů rovně.
- Pulzy počítejte pomocí čítače na vstupu I1.
- Změřte ujetou vzdálenost a vypočítejte vzdálenost ujetou na jeden impuls.
- Tento test opakujte 3 krát a zaznamenejte si rozdíly v hodnotách do tabulky.

Tipy:



- Nejprve zapněte oba motory (s otáčením motoru doleva).
- Pomocí prvku čítače Pulse Counter spočítejte impulsy na vstupu I1.
- Počítejte obě hrany impulsů (0-1 při stisku, 1-0 při uvolnění). To se nastavuje v okně vlastností jako druh impulsů. Tím bude měření ujeté vzdálenosti přesnější.
- Pak vypněte motory a ukončete program.
- Hotový program najdete v souboru Basic Model1.rpp.



Výsledek :

	Počet impulsů	Ujetá vzdálenost	Vzdálenost/impuls
Test 1	40		
Test 2	40		
Test 3	40		

Dá se říci, že tento model ujede na jeden impuls zhruba jeden centimetr (0,394 palce).

Nyní už také budete vědět, jaký směr rotace musíte nastavit pro každý motor, aby model jel určitým směrem. Poznamenejte si, co jste se naučili, do níže uvedené tabulky, abyste nad tím nemuseli pokaždé znovu přemýšlet, když budete chtít změnit směr otáčení. Jestliže zapojíte kabely přesně tak, jak je uvedeno v manuálu, budou se při nastavení otáčení doleva točit oba motory dopředu. Takto jsou naprogramovány motory ve všech vzorových programech.



Doplňte tabulku:

Směr pohybu	Směr otáčení M1	Směr otáčení M2
Dopředu	ccw	ccw
Dozadu		
Doleva		
Doprava		
Stop		

cw = ve směru hodinových ručiček, **ccw** = proti směru hodinových ručiček

Normálně byste museli umístit při každé změně směru pohybu uvést symboly obou motorů.

Tomu se můžete vyhnout vytvořením podprogramu pro každý pohyb. To enormně zjednoduší programování. V kapitole 4 manuálu k software ROBO Pro je podrobně popsáno vytváření podprogramů. Po přečtení této kapitoly budete připraveni provést další úkol. Nyní můžete přejít k úkolu **úrovně 2** v ROBO Pro.



Úkol 2 (úroveň 2)

- Vytvořte podprogram pro každý směr pohybu.
- Naprogramujte model tak, aby jel do čtverce s hranou délky jednoho metru (3.28 ft.).
- Jak vysoká je přesnost opakování?

Tipy:

- Nejprve vytvořte podprogram pro jízdu rovně. Ostatní programy můžete vytvořit zkopírováním toho prvního. Budete muset změnit pouze název a směr rotace motorů.
- Snížení rychlosti při zatáčení doleva a doprava zvýší přesnost.
- Znovu zapojte čítač impulsů a senzor tlačítka na vstup I1.
- Nejprve načtěte program do RAM a zjistěte, kolik impulsů je potřeba k otočení o 90°. Načítání do paměti RAM je jednak mnohem rychlejší než načítání do paměti FLASH memory a navíc má paměť Flash omezenou životnost přibližně 100.000 zápisů.
- Hotový program se jmenuje Basic Model 2.rpp.



Sledovač světla



■ Dosud jste pracovali se základním modelem a nyní nastal čas naučit robot reagovat na signály z jeho okolí. Podobně jako mýra, která zjišťuje zdroj světla a letí k němu. Stavebnice obsahuje 2 fototranzistory, které budeme používat jako detektor světla. Každý senzor bude ovlivňovat jeden motor, čímž umožní robotu jít za zdrojem světla. Tento program se skládá ze dvou částí. Jedna část se stará o hledání zdroje světla, zatímco druhá část provádí řízení směrem ke zdroji světla. Budeme k tomu opět používat podprogramy. Po zapnutí bude aktivován podprogram pro vyhledávání světla. Tento podprogram bude pokračovat dokud nebude detekován zdroj světla. Hlavní program se pokouší vést robot směrem ke zdroji světla. Kdykoliv se směr pohybu robotu více odchýlí z ideálního směru, jeden ze senzorů nebude osvětlen zdrojem světla a robot změni směr tak, aby byly osvětlené oba senzory.

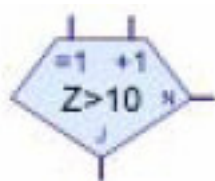
Nejprve podle manuálu sestavte model sledovače světla.



Úkol 1 (úroveň 2)

- Nejprve naprogramujeme funkci vyhledávání světla. Robot by se měl pomalu otáčet alespoň o 360° dokud nenajde zdroj světla a poté se zastaví. Jinak se bude otáčet opět 360°, ale opačným směrem. Pokud stále nedokáže detekovat zdroj světla, počká 5 sekund a pak zase začne hledat.
- Až úspěšně nadetekuje zdroj světla, měl by se model k němu vydat. Pokud se zdroj světla přesune doleva nebo doprava, robot by jej měl následovat. Když robot ztratí kontakt, měl by se vrátit k hledání zdroje světla. Zkuste robot přitáhnout pomocí baterky.

Tipy:



• Použijte podprogramy pro pohyb různým směrem, jež jste vytvořili pro základní model. Až načtete program Basic model 2.rpp, najdete program Basic model 2 a jeho podprogramy v okně s prvky ROBO Pro pod načtenými programy. Tyto podprogramy můžete jednoduše vložit do svého nového programu.

• Pro podprogram vyhledávání světla použijte prvek Count Loop (popis tohoto prvku najdete v manuálu ROBO Pro).

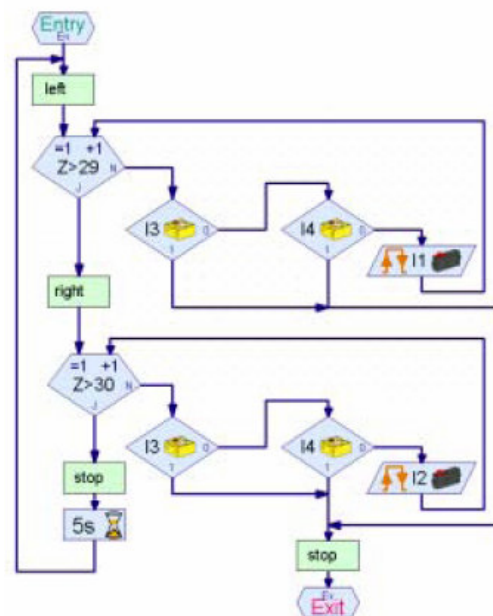
• Ve smyčce mezi „N“ a „+1“ zjišťujete stav fototranzistorů a načtete jeden impuls ze senzoru na vstupu I1. Smyčka se opakuje dokud robot neobjeví zdroj světla nebo se neotočí o 360°. Jednoduše musíte zkusit, kolik průběhů cyklem zabere, než se robot zcela otočí. Pak zadejte tuto hodnotu „Z“ do cyklu čítání.

• Pak musíte stejným způsobem naprogramovat druhou smyčku pro hledání druhým směrem.

• Pokud robot detekuje zdroj světla, zastaví se a opustí podprogram.

• Zde je kompletní podprogram pro vyhledávání světla:

• V hlavním programu znovu zjistíte stav fototranzistorů a budete řídit motory podle toho, které fototranzistory detekovaly světlo:



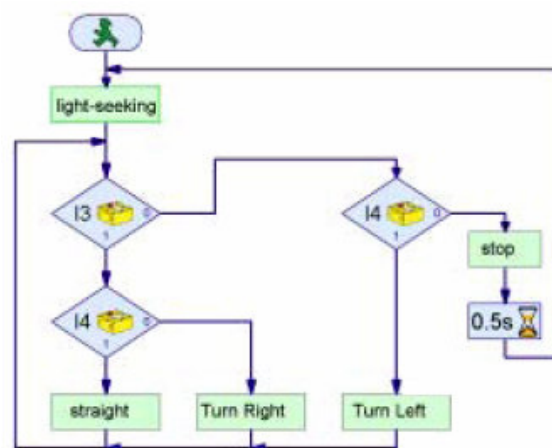
Světlo na I3 a I4	Rovně
Světlo pouze na I3	Zatočit doprava
Světlo pouze na I4	Zatočit doleva
Nebylo detekováno světlo	Zastavit a vrátit se do podprogramu pro hledání světla

• Zatočení doleva nebo doprava můžete dosáhnout nastavením různých rychlostí M1 a M2 se stejným směrem rotace. Jízda robotu pak bude mnohem hladší.

• Hlavní program nakonec vypadá takto:

• Hotový program najdete v souboru **Lightseeker.rpp**.

• Jako zdroj světla použijte baterku. Pozor aby kužel světla nebyl moc široký a neosvětloval oba fotosenzory. Pamatujte si, že ve velmi osvětleném prostředí může jiný zdroj světla, například slunce, svítit více než baterka a robot může následovat silnější zdroj světla.

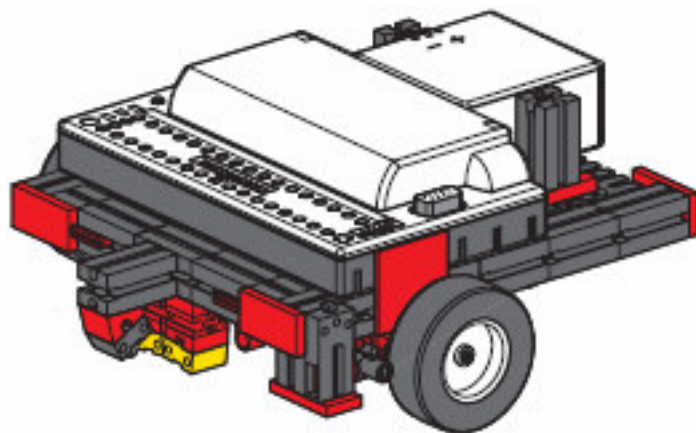


Stopař



■ Najdi a něco dělej je základní charakteristika inteligentního života. V předchozí části jste vytvořili a naprogramovali sledovač světla, neboli robot, který je schopen reagovat na přímé signály svého cíle.

U stopaře použijeme jiný způsob vyhledávání. Místo cíleného postupu ke zdroji světla vyznačíme černou čarou cestu, po které by měl robot jet. Toho lze dosáhnout relativně snadno pomocí fototranzistorů. Ty budou měřit odražené světlo a podle něj budou řízeny motory. Aby to fungovalo, bude čára osvětlena žárovkou. Pozor, aby světlo z žárovky nesvítilo přímo na fotosenzory.



Nyní podle návodu sestavte model Stopař.



Úkol 1

- Nejprve napište podprogram pro nalezení cesty. Za tímto účelem by se měl robot jednou otočit.
- Pokud není schopen nalézt cestu, měl by jet chvíli rovně a začít znovu hledat. K detekování cesty se používají fototranzistory.
- Pokud robot našel cestu, měl by se jí držet.
- Jestliže cesta skončila nebo ji robot ztratil, třeba z důvodu prudké změny směru, měl by ji začít znovu hledat.

Tipy:

- Po zapnutí lampy musíte chvíli (asi jednu sekundu) počkat, než budete moci snímat fototranzistory. Jinak budou fototranzistory detekovat „tmu“, což znamená cestu, tam, kde není, protože čtení je provedeno dříve než je žárovka plně rozsvícena.
- Jako cestu můžete použít černou pásku na vodovodní potrubí, který je asi 20mm široká nebo prostě nakreslete černou cestičku o této šířce na bílý papír. Zatáčky by neměly být moc ostré, aby robot neztrácel často stopu. Nejprve se pomocí rozhraní ujistěte, že fototranzistory detekují vaši cestu. Nezapomeňte při tom rozsvítit lampu.
- Připevněte lampu tak, aby oba fototranzistory měly na výstupu hodnotu 1 na světlém podkladu i když jsou oba motory M1 a M2 zapnuté. Pokud je baterie už dost vybitá, žárovka bude svítit o něco méně při spuštěných motorech. Kdyby žárovka nebyla správně připevněna, mohl by fototranzistor číst „tmu“ i kdyby nenašel cestu.
- Sledování čáry funguje podobně jako hledání světla. Musíte jen upravit hledání tak, aby model před dalším hledáním popojel kousek rovně.
- Pamatujte si, že tento model pojede rovně když bude na výstupu obou fototranzistorů 0 (= tma).
- Hotový program najdete v souboru **Tracker.rpp**.

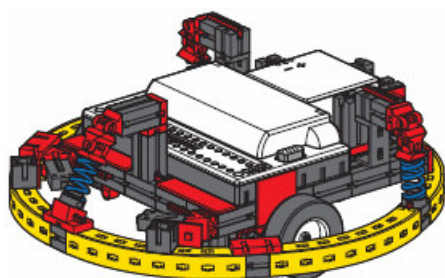




Úkol 2

- Vytvořte trasu s různými stupni zakřivení. S jakým nejmenším poloměrem bude model ještě pracovat?
- Při změnách trasy experimentujte s různými rychlostmi motorů M1 a M2. Jaká kombinace poskytuje nejlepší výsledky?
- Vytvořte kruhovou cestu. Pokuste se optimalizovat rychlosti tak, aby robot dosáhl co nejlepšího času na kolo. Zkušenosti z tohoto úkolu se vám budou hodit při soutěžích robotů.

Robot detekující překážky



■ Všechny roboty, které jsme dosud stavěli uměly ujet nějakou vzdálenost nebo sledovat světelný zdroj či cestu. Ale co se stane, když bude na cestě nějaká překážka? Dobře, překážka bude odtlačena stranou nebo se o to robot bude nesmyslně pokoušet dokud nebude baterie prázdná. Samozřejmě by bylo mnohem inteligentnější, kdyby robot uměl rozpoznat překážku a vyhnout se jí. Z tohoto důvodu je robot vybaven flexibilním kruhovým nárazníkem se třemi senzorovými tlačítky. Pomocí tohoto nárazníku je robot schopen určit, za je překážka nalevo, napravo nebo před ním. To, jak zareaguje na takovou překážku, je pouze otázkou programu.

Nejprve sestavte model „???Robot with Obstacle Detection“. K měření ujeté vzdálenosti je potřeba pouze jedno senzorové tlačítko (I1). Z toho důvodu můžeme ze základního modelu odstranit senzorové tlačítko I2 a použít je pro detekci překážek.



Úkol 1 (úroveň 2)

- Robot by měl jet nejprve rovně. Pokud narazí na překážku nalevo (I4), měl by se vrátit kousek zpátky a pak zahnout doprava.
- Pokud narazí na překážku vpravo (I3), měl by se kousek vrátit a pak zahnout doleva.

Tipy:

- Detekování překážek při couvání necháme až na později.
- Hlavní program bude číst senzorová tlačítka. V závislosti na to, které senzorové tlačítko je aktivováno, bude model objíždět překážku zleva nebo zprava. V obou případech to bude prováděno v podprogramu.
- Počet impulsů při zatáčení doprava by měl být jiný než počet impulsů při zatáčení doleva (např. 3 impulsy doprava, 5 impulsů doleva). Jinak se může stát, že model zajede do rohu a tam se zarazí, protože se bude otáčet o stejné množství impulsů doleva i doprava.
- Hotový program se jmenuje **Obstacle 1.rpp**.

Robot ještě neumí dvě věci: neumí rozpoznávat překážky při couvání a rovněž zatím neumí rozeznat, zda je překážka přímo před ním. Ale mohl by. Pokud bude stisknuto I5 při couvání, překážka je za modelem. Pokud jsou vstupy I3 a I4 při pohybu vpřed aktivovány současně, překážka je přímo před robotem. V takovém případě by se měl robot hned otočit o 90°. Dohromady nyní máme následující možnosti, na něž by měl robot reagovat:

Překážka	Senzor (tlačítko)	Reakce
vpravo	pouze I3	zatočit doleva (otočení o přibl. 30°)
vlevo	pouze I4	zatočit doprava (otočení o přibl. 45°)
vpředu	I3 a I4	zatočit doleva (otočení o přibl. 90°)
vzadu	I5	kontrolováno pouze při couvání. Zastavit a pokračovat v původním plánu.

Tento problém vám mohou pomoci vyřešit některé nové prvky programu jako Operators (např. AND, OR) z ROBO Pro Level 3. Level 3 umožňuje výměnu dat mezi odlišnými prvky pomocí oranžových šipek. Abyste mohli využít výhod těchto možností, přepněte software na úroveň Level 3. Nyní by bylo dobré si vzít příručku ROBO Pro a pečlivě si přečíst kapitolu 5. Poté budete připraveni na další úkol.



Úkol 2 (úroveň 3)

- Upravte program pro rozpoznávání překážek tak, aby model reagoval podle popisu ve výše uvedené tabulce.
- Využijte možností ROBO Pro Level 3.

Tipy:

- V podprogramu pro zjišťování překážek jsou zjišťovány různé kombinace senzorů pomocí operátorů. Podprogram má pro každou z možností oddělený výstup.

Datový vstup SR = senzor vpravo

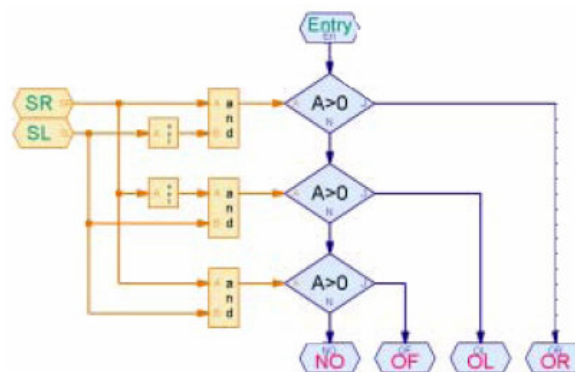
Datový vstup SL = senzor vlevo

Výstup NO = žádná překážka

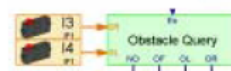
Výstup OF = překážka vpředu

Výstup OL = překážka vlevo

Výstup OR = Překážka vpravo

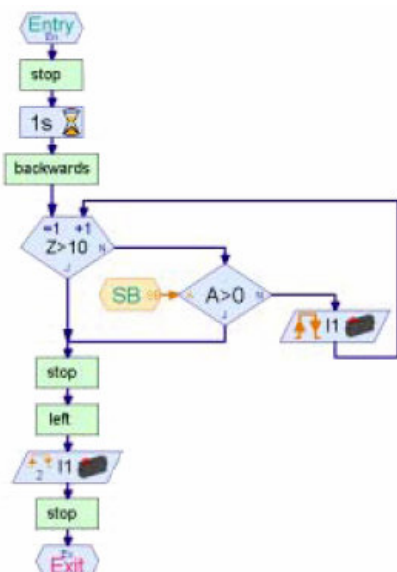


- Vložte do hlavního programu oranžové prvky Sensor a připojte je k podprogramu pomocí datových vstupů, abyste hned viděli, který senzor je čten.



- Vstup I5 je zjišťován při couvání ve všech podprogramech pro vyhýbání se překážkám. Robot bude couvat dokud nebude dosaženo stanoveného počtu impulsů nebo dokud nebude zapnut vstup I5. Vstup I5 je znovu umístěn do hlavního programu, abyste hned viděli, ve kterém podprogramu je senzor čten.
- Kompletní program najdete v souboru **Obstacle 2.rpp**.

Výhodou techniky programování použité v tomto úkolu je to, že můžete sledovat přímo v hlavním programu, který senzor je čten v podprogramu. Pokud byste chtěli změnit vstup, můžete to provést v jediném bodě bez potřeby prohledávání všech podprogramů. Použití operátorů kromě toho umožňuje vytváření velmi přehledných logických operací. Principiálně je to možné provést také pomocí prvků větvení. Ale to



může při více podmínkách vést velmi rychle ke zmatku.

Sledovač světla s detekcí překážek



■ Ještě zdaleka nejsme u konce možností stavebnice ROBO Mobile Set. Proto nyní zkombinujeme funkci vyhledávání světla s vyhýbáním se překážkám. Z čistě vědeckého pohledu bude robot vybaven dvojím druhem chování. Jelikož oba vzorky chování nemohou být aktivní současně, dostanou různé priority. Robot by měl vyhledávat světlo a když narazí na překážku, měl by se jí vyhnout. Až se jí vyhne, bude moci pokračovat ve vyhledávání světla.

Profesionální vývojáři software, když řeší takovou úlohu, se nebudou zabývat jen programováním. Použijí speciální strategii pro vývoj programu. Jednou z těchto metod je tzv. návrh „shora dolů“. Při tomto přístupu je systém definován jako celek shora dolů bez zabývání se detaily na začátku. Tuto metodu použijeme i my.



Úkol 1 (úroveň 3)

Naučte robot následující vzorky chování:

- Najdi zdroj světla.
- Až jej najdeš, jeď k němu.
- Pokud narazíš na překážku, vyhni se jí.
- Pak znovu začni hledat zdroj světla.

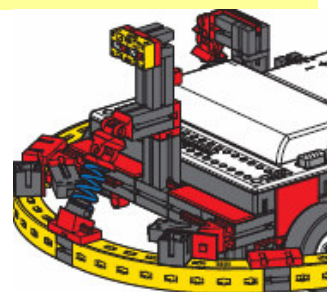
Při hledání řešení použijte prvky programu ROBO Pro level 3.

Při úkolu používejte postup shora dolů.

Tipy:

Nejprve si úkol rozdělíme na tři části:

- Zjištění, zda robot vidí zdroj světla (podprogram „Light“)
- Zjištění překážky (podprogram „Obstacle“)
- Sdělení robotu, co v těchto případech dělat (podprogram „Driving“)



Nyní zvažte různé situace, kdy je robot schopen využít podprogramy „Light“ a „Obstacle“. Každé ze situací přiřaďte numerickou hodnotu. Tato hodnota je uložena v proměnné pomocí prvku Command. Každé situaci bude odpovídat nějaká reakce, která je provedena v řídicím podprogramu.

Podprogram Light (světlo)

Číslo	Situace	Stav senzorů	Reakce
0	Není přítomen žádný zdroj světla	I6=0; I7=0	hledání světla
1	Zdroj světla přímo před robotem	I6=1; I7=1	jízda rovně
2	Zdroj světla nalevo od robotu	I7=1	otočení doleva
3	Zdroj světla napravo od robotu	I6=1	otočení doprava

Podprogram Obstacle (překážka)			
Číslo	Situace	Stav senzorů	Reakce
4	Překážka přímo před robotem	I3=1; I4=1	otočení se o 90°
5	Překážka napravo od robotu	I3=1	otočení doleva
6	Překážka nalevo od robotu	I4=1	otočení doprava

Nyní musíte jednoduše zadat tyto výsledky do programu ROBO Pro pomocí prvků programu.

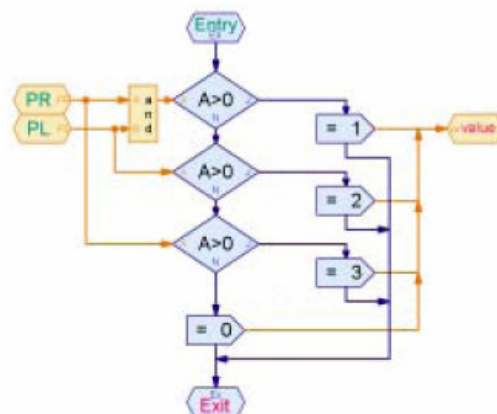
Podprogram Light (světlo):

PR=fototranzistor vpravo

PL=fototranzistor vlevo

Stejně jako dříve zadejte do hlavního programu prvky pro čtení fototranzistorů, abyste mohli hned sledovat, co se děje v podprogramu.

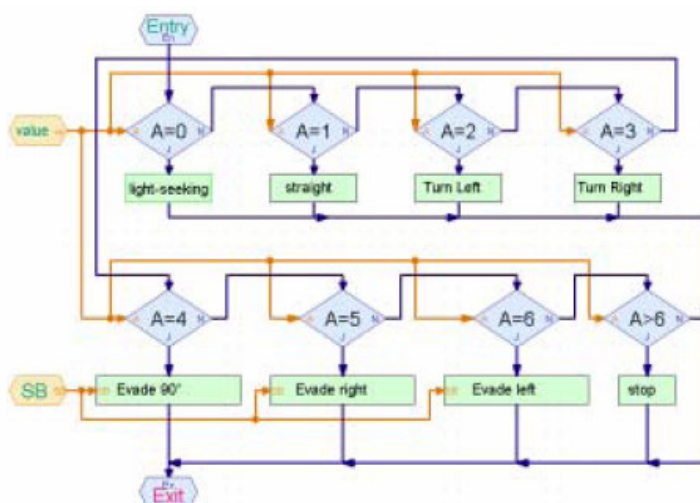
Do hlavního programu musíte vložit také proměnnou pro uložení hodnoty z příkazových prvků. Ta bude používána v několika pod programech. S podprogramem ji spojíte pomocí datového výstupu.



Vytvořte podprogram „Obstacle“ (překážka) stejně jako podprogram „Light“ (světlo).

V podprogramu „Driving“ (řízení) budete používat aktuální hodnotu proměnné v prvcích pro větvení program a provedení odpovídající reakce:

SB=senzor vzadu



Teď už zbývá jen maličkost – vytvořit podprogramy. Ale moment! Většina z nich už existuje. Podprogram pro hledání světla může být například zkopírován z programu pro model Sledovač světla. Pokud si nevzpomínáte, jak se to dělá, přečtěte si kapitulu 4 manuálu ROBO Pro.

Ale pozor:

U modelu Sledovače světla byly fototranzistory připojeny na vstupy I3 a I4. nyní jsou na vstupech I6 a I7. Kromě toho jsme měli tlačítkové senzory připojené na I1 pro zatáčení vlevo a I2 pro zatáčení vpravo. Nyní impulsy počítáme pouze na vstupu I1, což funguje také celkem dobře. To znamená, že budete muset podprogram pro hledání světla po zkopírování upravit. Jelikož je čtení senzoru skryto v podprogramu, je snadné se splést. K tomu nedojde, když umístíte vstupy do hlavního programu a připojíte je do podprogramu pomocí datových vstupů.

Ale to jste ještě nevěděli, když jste pracovali na Sledovači světla.

Podprogramy pro vyhýbání se úřekázkám také existují, psali jsme je pro model Robot detekující překážky. Tady byl senzor na I5, čtený dodatečně při couvání, již umístěn do hlavního programu.

Můžete se podívat na hotový program v souboru **Obstacle-Light.rpp**. Hlavní program vypadá na první pohled velmi přehledně a jednoduše. Ale v podprogramech je skryto mnoho těžké práce. Ale i složitý program můžete projít postupně krok za krokem shora dolů a pochopit jej.

Mimochodem, pokud byste měli kamaráda, který má také stavebnici ROBO Mobile Set, mohli byste experimentovat ještě více. Můžete například připevnit zdroj světla na každý z robotů a roboti se mohou navzájem hledat.

Robot s detekcí hran



■ V předchozím příkladu jsme viděli, jak postupovat při vytváření složitějšího programu. Nyní můžete robot naučit další velmi důležité chování. Je potřeba jej naučit, aby nespadl ze stolu. Náraz do překážky robotu ve většině případů neublíží. Ale pád ze stolu z výšky jednoho metru by jej mohl zničit, i když jsou stavební prvky Fischertechnik velmi robustní. Z toho důvodu vybavíme robot senzory umožňujícími rozpoznávání hran. Detektory hran se skládají z tlačítkového senzoru, který je aktivován otáčejícím se kolečkem. Toto kolečko se může pohybovat také nahoru a dolů. Když se kolečko dostane za hranu stolu, klesne a tlačítkový senzor již není aktivován, a program tak zjistí, že model dojel k hraně a zareaguje. Robot má celkem 4 detektory hran, které mu umožňují zjišťovat možnost pádu na každé straně při pohybu dopředu i dozadu. Model díky tomu nebude mít senzor pro měření ujeté vzdálenosti. Ujetá vzdálenost bude zjišťována z času zapnutí motorů.

Nejprve sestavte model podle návodu.

Pečlivě zkontrolujte, zda detektory hran správně fungují: ???

- když se model přiblíží k hraně edge stolu a tlačítko je znovu stisknuto
- když je kolo zpátky na stole.

Pokud to bude nutné, jedno z tlačítek může být posunuto o kousek nahoru nebo dolů.



Úkol 1 (úroveň 3)

- Nejprve se zamyslete, jak by měl robot reagovat při dojetí k hraně.
- Při bližším prozkoumání zjistíte, že existuje mnoho kombinací senzorů.
- Aktivován by mohl být jen jeden ze 4 senzorů, 2 či 3 různé senzory najednou, nebo dokonce všechny čtyři senzory.
- Jak by měl robot reagovat na každou z těchto možností?

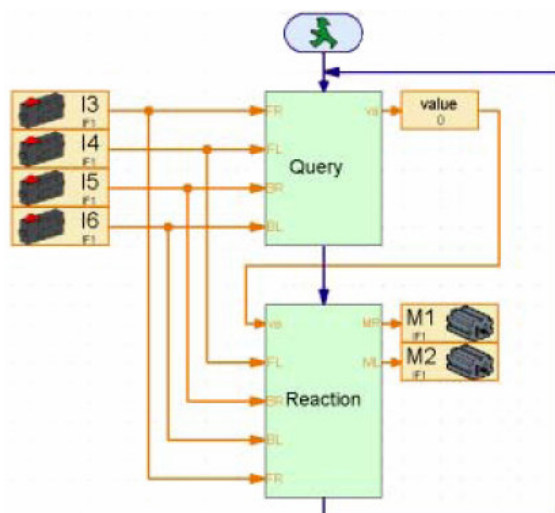
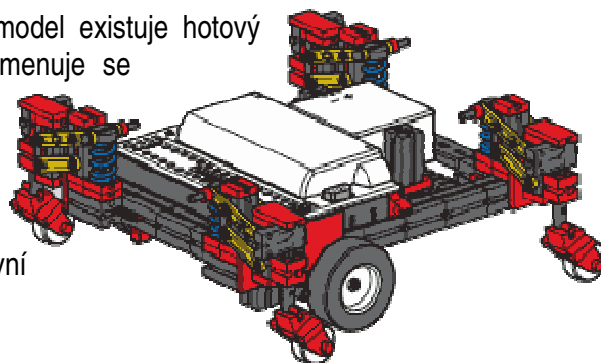
Tipy:

Řešení najdete v následující tabulce. Senzory, které jsou za hranou (tlačítkový senzor=0), jsou vyznačeny ●. Každá kombinace má své číslo. V programu, který budeme vytvářet později, bude robot reagovat na každou ze situací na základě tohoto čísla. Ale o tom si povíme později. Nejprve se zamyslíme nad tím, jak robot musí stát, aby konkrétní kombinace nastala.

Č.	Pravý přední (I3)	Levý přední (I4)	Pravý zadní (I5)	Levý zadní (I6)	Reakce
0					Rovně (žádný senzor není za hranou)
1	●	●	●	●	Zastavit (všechny 4 senzory jsou za hranou)
2	●	●	●		Zatočit doprava
3	●	●		●	Zatočit doleva
4	●		●	●	Zatočit doleva
5		●	●	●	Zatočit doprava
6	●	●			Nejprve dozadu, pak zatočit doprava
7	●		●		Zatočit doleva
8	●			●	Zatočit doleva
9		●	●		Zatočit doprava
10		●		●	Zatočit doprava
11			●	●	Dopředu
12	●				Nejprve dozadu, pak doleva
13		●			Nejprve dozadu, pak doprava
14			●		Dopředu
15				●	Dopředu

Docela náročné, že? Ale nebojte se, pro tento model existuje hotový program využívající všech výhod ROBO Pro. Jmenuje se **Edges.rpp**.

Nejdůležitější prvky jsou umístěny v hlavním programu, což usnadňuje pochopení sekvence jako celku. Složitě čtení senzorů a řízení motorů jsou skryté v podprogramech. Nejprve hlavní program:



Sekvence začíná čtením 4 senzorů. Po celou dobu můžete sledovat, který ze senzorů je čten. Jsou připojeny k podprogramům pomocí datových vstupů. Podprogram pro čtení „Query“ určuje, které tlačítko je stisknuté a přiřazuje hodnotu podle výše uvedené tabulky. Tato hodnota je přiřazena do stejnojmenné proměnné, kterou najdete v hlavním programu. Hodnota proměnné je pak odeslána do podprogramu „Reaction“, který na základě této hodnoty řídí oba motory. Senzory jsou poté znovu čteny v podprogramu „Reaction“, aby se mohl robot vyhnout překážce.

V případě potřeby nyní můžete na rozhraní měnit přiřazení senzorů i výstupů motorů, aniž byste se museli dívat do podprogramů a hledat, kde by se mohl skrývat vstupní prvek nebo symbol motoru. Každý vstup a každý výstup je uveden pouze jednou.

Tato technika programování je užitečná zejména když byste chtěli používat podprogram pro mnoho různých modelů a zatím nevíte, které vstupy a výstupy rozhraní v konkrétním případě použijete.

Pokud jste zvědaví, jednoduše se podívejte do podprogramů a snažte se je pochopit. Způsob programování je podobný jako u modelu Sledovač světla s detekcí překážek.



Úkol 2 (úroveň 3)

Načtete program do rozhraní a nechte model jezdit po stole.

- Reaguje model vždy správně?
- Měl by se v případě nějaké kombinace chovat jinak?
- Podle potřeby upravte program.

Kráčející robot

- Nyní se místo robotu s koly zaměříme na jiný druh pohybu – na chůzi.

Chůze hmyzu je perfektním modelem pro pohyb mechanických šestinohých robotů. V průběhu tzv. tříbodové chůze jsou vždy tři ze šesti nohou současně na zemi. Přední a zadní noha na jedné straně se pohybuje zároveň s prostřední nohou na straně druhé.

Tříbodová chůze

Nohy stojící na zemi (znázorněné černě) tvoří stabilní trojnožku, takže je model vždy stabilní a při chůzi nepřepadne.



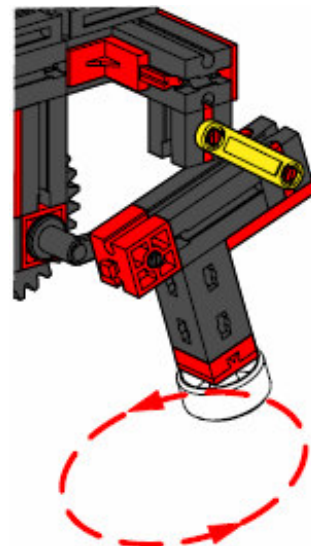
Nohy kráčejícího robotu od firmy Fischertechnik jsou vyrobeny se čtyřmi klouby. Tento návrh se nazývá mechanismus klikové hřídele. Pohyblivé části převodů poháněné klikovou hřídelí vytváří kmitavé pohyby. Vzdálenost mezi jednotlivými klouby a pozice spodní části nohy jsou vytvořeny tak, aby noha prováděla eliptický pohyb, když se hřídel otáčí. Výsledkem je pohyb podobný chůzi.

6 ojníc pohánějících nohy musí být připevněno přesně tak, jak je uvedeno v manuálu. Tři nohy dotýkající se ve stejný moment podlahy odpovídají stejné pozici klikové hřídele. Ojnice tři nohou, jež jsou v daný moment ve vzduchu, jsou otočeny o 180° proti zbývajícím třem. Správnou pozici ojníc vůči ostatním zajišťuje možnost chůze modelu se správným pořadím kroků.

Šrouby použité k zabezpečení převodů na hřídele musí být pevně dotažené, aby se ojnice při chůzi neposunuly.

každou stranu modelu pohání jeden motor (to je nezbytné pro chůzi). Z toho důvodu musíte zajistit, aby prostřední noha na jedné straně byla vždy ve stejné pozici jako dvě vnější nohy na straně druhé. Software řídí synchronizaci pomocí senzorů I1 a I2.

Nejprve sestavte model podle manuálu. Důkladně zkontrolujte všechny senzory a motory pomocí testu rozhraní, abyste se ujistili, že jsou zapojené správně. Směr otáčení motorů: otáčení doleva = pohyb rovně.





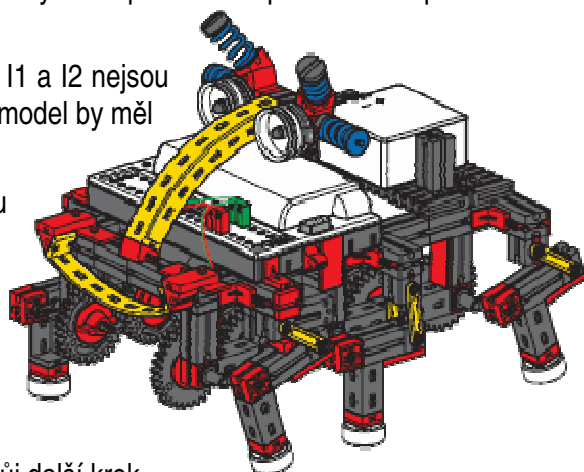
Úkol 1 (úroveň 1)

Naučte robot chodit.

- Naprogramujte model tak, aby chodil rovně pomocí tříbodové chůze.
- Pomocí senzorů I1 a I2 sesynchronizujte levé a pravé nohy.
- Při tom se ujistěte, že jsou dvě vnější nohy na jedné straně a prostřední noha na druhé straně ve stejné pozici.

Tipy

- Nejprve přesuňte nohy na levé i pravé straně do výchozí pozice. To provedete zapnutím motorů (při otáčení doleva).
- Sekvence by měla pokračovat pouze když tlačítka I1 a I2 nejsou uvolněna???depressed (toto čtení je nutné, protože model by měl provést druhý krok).
- Nechte běžet motory dokud nebudou znovu uvolněna příslušná tlačítka (I1 pro M1, I2 pro M2). Je velmi důležité, aby model nezačal další krok dokud nebudou oba senzory uvolněny. Protože pouze pak budou nohy ve správné pozici. Samozřejmě za předpokladu, že klikové hřídele pohánějící nohy jsou přiděleny podle manuálu.
- Nyní může sekvence začít a robot může provést svůj další krok.



Model bude nyní do ukončení programu chodit rovně.

- Hotový program najdete v souboru **Walking Robot 1.rpp**.

Podobně jako u základního modelu i zde můžete nechat model chodit doleva nebo doprava změnou směru otáčení motorů. K počítání kroků můžete používat I1 nebo I2.

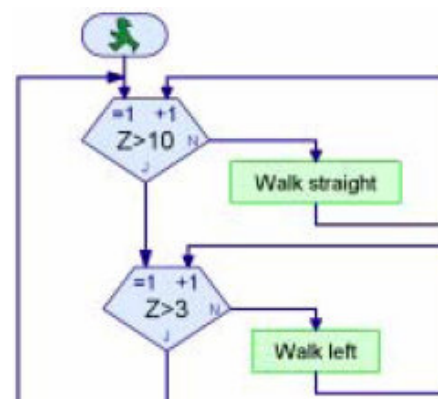


Úkol 2 (úroveň 2)

- Naprogramujte svůj model tak, aby šel 10 kroků rovně, 3 kroky doleva, 3 kroky doprava a 10 kroků dozadu.
- Pro každý směr vytvořte samostatný podprogram.
- K počítání kroků použijte prvek čítače Count Loop.

Tipy

- Jednoduše zkopírujte program **Walking Robot 1.rpp** do podprogramu.
- Pro každý směr pohybu vytvořte samostatnou kopii tohoto podprogramu. V každém podprogramu změňte směr rotace motorů, aby se model pohyboval požadovaným směrem.
- Pomocí prvku čítače Count Loop spočítejte kroky potřebné pro otočení. V každém cyklu podprogramu model udělá jeden krok. Když program projde cyklem 10krát, model udělá 10 kroků.

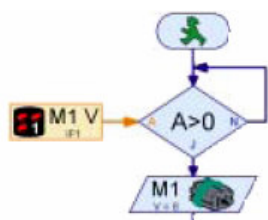


Takto můžete váš robot naučit jakoukoliv sekvenci kroků (). Rozpoznáváním překážek jsme se podrobně zabývali již u robotu s koly. Takže to nemusíme znovu opakovat. Ale proč to neaplikovat na kráčejší robot? Potřebné senzory ve stavebnici jsou. Při programování můžete jako příklad používat robot s koly. Držíme vám palce!

Možná rozšíření

■ Rozhraní ROBO Interface nabízí mnohem více funkcí, než zde bylo uvedeno. Ale abyste je mohli využít, budete potřebovat další komponenty, které nejsou obsažené ve stavebnici. A protože jsou velmi zajímavé, o několika z nich si alespoň řekneme.

Infračervené dálkové ovládání



■ Rozhraní ROBO Interface má infračervený přijímač s diodou pro ovladač obsažený ve stavebnici IR Control Set Art No. 30344. To vám umožní číst tlačítka vysílače v software ROBO Pro jako digitální vstupy a například pro zapínání a vypínání motorů.

Například jsme ve vzorovém programu naprogramovali dálkové ovládání pro kráčejší robot. Pomocí 4 oválných tlačítek na dálkovém ovladači můžete řídit model tak aby šel dopředu, dozadu, doleva a doprava. Než ho začnete používat, musíte pouze načíst program **Walking Robot IR.rpp** do rozhraní.

Jiným geniálním programem spolupracujícím s dálkovým ovladačem je program **Mobile-Teach-IR.rpp**. Pomocí tohoto programu můžete řídit robot s koly, např. Jednoduchý robot nebo Základní model. Tento model si bude pamatovat cestu, kterou ujel a bude ji podle potřeby moci zopakovat. Ale po zastavení programu je uložená cesta vymazána.

Toto umožňuje prvek programu „List“ ROBO Pro. Do tohoto prvku je možno uložit mnoho hodnot a pak je znovu vyvolat (podívejte se také do manuálu k ROBO Pro). Program sám o sobě může být celkem složitý, ale použití tohoto prvku je velmi jednoduché:

1. Načtete program Mobile-Teach-IR.rpp do paměti Flash rozhraní ROBO a spustíte jej.
2. Stisknete tlačítko M1 ► / ►► na dálkovém ovládání. Tím se spustí proces učení.
3. Pomocí oválných tlačítek se šipkami přesuňte model do požadovaného směru.
4. Stisknete tlačítko M2 ► / ►►. Tím se uloží cesta, kterou robot urazil.
5. Stisknete tlačítko M3 ► / ►►. Robot pak projde uloženou cestu.

Pomocí aplikace jako je tato je programování robotů rychlé. Pamatujte si, že uložená cesta bude odstraněna po ukončení programu pomocí tlačítka Prog na rozhraní.

Datové spojení ROBO RF

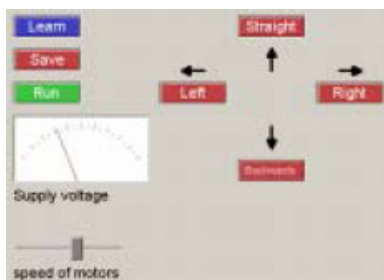
■ Rádiové rozhraní ROBO RF Data Link Art No. 93295 nahrazuje kabel mezi rozhraním a PC. To je samozřejmě dobrá věc. Zprvč nebudete muset při každém načítání dat do rozhraní pořád připojovat a odpojovat kabel. Zadruhé budete moci programy spouštět v online režimu bezdrátově. Tak bude mnohem snazší najít chyby než pomocí opakovaného načtení software do rozhraní. A kromě toho budete moci řídit mobilní roboty v online režimu na vaší obrazovce pomocí panelu v software ROBO Pro, podobně jako když používáte infračervený dálkový ovladač. Ale na rozdíl od dálkového ovladače obrazovka také ukazuje data předaná rozhraním,

jako např. hodnoty proměnných nebo analogových vstupů, napájecí napětí baterií a rychlost motorů.

Například jsme připravili demonstrační program a nyní je možno řídit robot s koly pomocí panelu v software. Tento program je uložen v souboru Mobile-Teach-RF.rpp. Samozřejmě to můžete zkusit i pomocí kabelu k rozhraní. Ale to může být poměrně nepohodlné. Akční rádius modelu je omezený, kabel se zamotává a robot nezatáčí tak, jak má. Po prvním takovém pokusu si pravděpodobně rychle objednáte RF Data Link.

Načtěte program **Mobile-Teach-RF.rpp**.

Přepněte se na panel v pruhu funkcí hlavního programu. Pak program spusťte v online režimu. Nyní budete moci řídit a programovat model pomocí tlačítek na panelu.



1. Stiskněte tlačítko „Learn“. Tím se spustí proces učení.
2. Pomocí šipek navedte model do požadovaného směru.
3. Stiskněte tlačítko „Save“. Tím se uloží cesta, kterou robot urazil.
4. Stiskněte tlačítko „Run“ a robot projde uloženou cestu.

Cesta bude ovšem po ukončení program ztracena.

Více informací o vytváření panelů najdete v manuálu k ROBO Pro.

Rozšíření ROBO I/O

■ Pokud chcete vytvořit model s tolika senzory a modely, že nestačí vstupy a výstupy na rozhraní ROBO Interface, můžete použít rozšiřující modul ROBO I/O-Extension Art No. 93294. Tím získáte dalších 8 digitálních vstupů, 4 výstupy pro motory a jeden analogový odporový vstup. K tomuto rozšíření můžete připojit druhý a třetí modul, které budou tak řízeny rozhraním ROBO Interface. Díky tomu můžete získat celkem až 16 výstupů pro motory, 32 digitálních vstupů, 5 analogových odporových vstupů, 2 analogové napěťové vstupy a 2 vstupy pro senzory vzdálenosti.

Pokud to stále ještě nestačí, můžete z počítače řídit v online režimu několik rozhraní. Jedno například připojené přes sériový port COM , jedno přes port USB nebo dvě rozhraní připojená přes USB a všechny mohou mít až 3 rozšíření ROBO I/O! To už by mohlo stačit, že? Informace o tom, jak to celé funguje, najdete v kapitole 6 manuálu k ROBO Pro.



Odstraňování problémů

■ Experimentování je zábava. Tedy, dokud vše funguje. Problémy většinou nebudete muset odstraňovat, ale bohužel ne vždy.

Když model nebude fungovat správně, musíte zjistit, zda jste správně pochopili mechaniku a zda jste schopni najít chybu.

U mechanických chyb můžete alespoň něco vidět (nesprávné sestavení) nebo cítit (když se to nebude hýbat).

Ale u problémů s elektrickým zapojením je to složitější.

Profesionálové používají při odstraňování problémů řadu různých měřících přístrojů, například voltmetr nebo osciloskop. Ale ne každý má takový přístroj. Z toho důvodu se chceme zkusit zaměřit na jednoduché chyby a jejich řešení.



Sestavování kabelů

Než začnete experimentovat, připravte si komponenty ze stavebnice. Konektory jsou například přidělané na jednotlivé kabelové segmenty. Nejprve zkrátíme kabely. Odměříme si požadovanou vzdálenost a odpovídající segmenty odstraníme. Každý kabel by měl být po sestavení otestován pomocí baterie a lampy. Pokud žárovka po připojení k baterii svítí, kabel je v pořádku. Také byste měli zkontrolovat, zda jsou správně barvy kabelů, červený konektor na červeném kabelu a zelený konektor na zeleném kabelu.

Test rozhraní

Jestliže program (dokonce ani dodaný program) ve spojení s modelem nefunguje, můžete spustit test rozhraní. Tento nástroj umožňuje otestovat každý vstup a výstup samostatně. Fungují senzory? Točí se motory správným směrem? U všech našich mobilních motorů jsou motory připojené tak, že se kolečko nebo noha budou pohybovat dopředu při směru otáčení ccw. Jestliže je vše v pořádku, budete muset začít hledat nějakou mechanickou příčinu.

Ztráta spojení

Ztráta spojení je nepříjemná chyba. Možná, že jsou konektory málo zasunuty. V takovém případě prostě jen dotáhněte šroubky malým šroubovákem. Ale opatrně. Pokud byste je dotáhli moc, kontakty se mohou zlomit. Další příčinou ztráty spojení mohou být svorky, kde je konektor připevněn ke kabelu šroubkem. Opatrně je dotáhněte. Také se podívejte, zda měděný drátek není někde zlomený.

Zkrat

Nesprávným zapojením kabelů také můžete vytvořit zkrat. V takovém případě nebude nic fungovat tak, jak by mělo. Baterie mají vestavěnou pojistku, která přeruší proud při příliš vysoké teplotě nebo proudu. Výstupy rozhraní budou také při přehřátí odpojeny.

Ke zkratu může dojít také když nesprávně dotáhněte šroubek konektoru na kabelu. Šroubek může přecházet přes hranu konektoru. Jestliže pak zapojíte dva konektory do rozhraní hned vedle sebe a jejich šroubky se dotknou, vznikne zkrat. Z toho důvodu by měly být šroubky dotahovány vždy s citem. Když zapojujete konektory, pokaždé sledujte, zda se nemohou šroubky dotknout.

Napájecí zdroj

Dochází-li při práci k nevysvětlitelným přerušením, mohou být vybité baterie. Při připojení zátěže na krátkou dobu poklesne napětí pod povolenou mez a to způsobí reset procesoru rozhraní. Když se rozsvítí červená kontrolka na rozhraní ROBO Interface, je baterie slabá a baterie musí být dobity.

Chyba v programu

Pokud se chyba objevuje ve vámi napsaném programu a nemůžete najít její příčinu, zkuste načíst jedem z dodaných programů. Takto budete moci vyloučit elektrické nebo mechanické chyby. Při online režimu můžete sledovat běh programu na obrazovce. Jestliže se program v určitém bodě zastaví, pak byste měli na tomto místě hledat příčinu. Možná jste vybrali nesprávný vstup nebo motor či podmínku.

Pokud se vám problém nepodaří vyřešit, můžete nás kdykoliv kontaktovat e-mailem na adrese info@stavebnice.com nebo se podívejte na adresu www.fischertechnik.de, kde najdete diskusní fórum a chat a můžete se zde také zdarma zaregistrovat do klubu Fischertechnik Fan-Club.

Věříme, že s touto stavebnicí strávíte mnoho hodin zábavy, zažijete s ní mnohá překvapení a naučíte se mnoho nových věcí.

