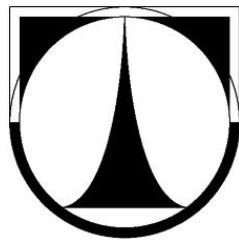

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2612 - Elektrotechnika a informatika

Studijní obor: 2612R011 - Elektronické informační a řídicí systémy



Robotino – výukové úlohy

Robotino – education tasks

Bakalářská práce

Autor:	David Mrázek
Vedoucí práce:	Ing. Josef Černohorský, Ph.D.
Konzultant:	Ing. Jan Strnad

V Liberci 3.5.2013

Zadání bakalářské práce

Příjmení a jméno studenta (osobní číslo - nepovinné)	Mrázek David
Zkratka pracoviště	MTI
Datum zadání BP/DP	10. 10. 2012
Plánované datum odevzdání	17. 5. 2013
Rozsah grafických prací	Dle potřeby dokumentace
Rozsah průvodní zprávy	cca 35 – 40 stran
Název BP/DP (česky)	Robotino – výukové úlohy
Název BP/DP (anglicky)	Robotino – education tasks
Zásady pro vypracování BP/DP	
<ol style="list-style-type: none"> 1. Seznamte se systémem Festo Robotino, prostudujte dostupnou dokumentaci k základním úlohám 2. Seznamte se s možnostmi navigace mobilních robotů ve vnitřním prostředí. Proveďte rešerši používaných metod se zaměřením na Festo Robotino. 3. Vytvořte sadu výchozích projektů, pro snazší použití studenty v rámci výuky a odborných prací 4. Zhodnoťte možnosti stávající konfigurace Festo Robotino, případně navrhnete rozšíření pro potřeby řešení složitějších úloh. 	
Seznam odborné literatury	
<p>Dokumentace systému Festo Robotino, dokumentace systému North Star Sensor, dokumentace projektu OpenRobotino</p> <p>Grepl R, Kinematika a dynamika mechatronických systémů, VUT v Brně, ISBN 978-80-214-3530-8</p>	
Vedoucí BP/DP	Ing. Josef Černohorský, Ph.D.
Konzultant BP/DP (u externích pracovníků uveďte plný název pracoviště – firmy)	Ing. Jan Strnad

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis

Poděkování

Chtěl bych poděkovat vedoucímu mé bakalářské práce Ing. Josefu Černoorskému, Ph.D., který mi vždy ochotně pomohl v řešení nastalých problémů, a časově se mi vždy přizpůsobil. Poděkování patří samozřejmě také mým přátelům a přítelkyni, kteří mi několikrát pomohli a také mé rodině, která mě v mém studiu podporuje.

Abstrakt

Cílem této bakalářské práce je vytvořit sadu úloh, pro studenty Technické univerzity v Liberci. Pro větší přehlednost práce, byla rozdělena do čtyř hlavních částí.

První část práce je teoretická a zabývá se celkovým seznámením se s mobilním systémem Festo Roborino (dále jen "Robotino"). Jsou zde všechny důležité informace o jeho konstrukci a možnostech.

Druhá část práce je také teoretická a zabývá se možnostmi navigace mobilních robotů ve vnitřním prostředí. Naleznete zde řešerši všech možností navigace mobilních robotů ve vnitřním prostředí.

Třetí část práce je nejdůležitější a obsahuje vypracované úlohy pro studenty na hodiny základy robotiky. Všechny tři úlohy jsou vytvořeny tak, aby se studenti naučili naprogramovat mobilní robotický systém Festo Robotino a zároveň aby se seznámili s možnostmi navigace mobilních robotů ve vnitřním prostředí.

Čtvrtá část práce obsahuje zhodnocení stávající konfigurace Robotina. Také zde naleznete návrhy pro zlepšení konfigurace Robotina a celkové zhodnocení práce.

Klíčová slova

Robot, Robotino, Robotino® View, Robotino® SIM, navigace, senzory

Summary

The main goal of this work is to create a set of tasks for the students of the Technical University in Liberec. To make it easier to read, the work was divided into four main parts.

The first part is theoretical and this part also introduces us to the mobile system Festo Robotino (mentioned as "Robotino"). There are all the important informations about its design and capabilities.

The second part is theoretical, too. It is about the possibilities of navigation of indoor mobile robots. You will find there all types of navigation.

The third part is the most important and it contains prepared tasks for students in class called Basics of robotics. All three tasks are created in that way that students can learn how to program mobile robotic system Festo Robotino and to become familiar with navigation of indoor mobile robots.

The last part contains an assessment of the current configuration Robotino. You will also find suggestions for improvement of Robotino configuration and overall assessment of the work.

Keywords

Robot, Robotino, Robotino® View, Robotino® SIM, navigation, senzors

Obsah

1	Seznámení se se systémem Robotino	10
1.1	Podvozek	10
1.2	Pohonná jednotka	11
1.3	Řídící jednotka	11
1.4	Senzory.....	12
1.4.1	Inkrementální snímač	13
1.4.2	Anti-kolizní senzor	13
1.4.3	Infračervené snímač	13
1.5	Programování robota	13
1.6	Parametry	14
2	Možnosti navigace mobilních robotů ve vnitřním prostředí	15
2.1	Navigace výpočtem (Dead reckoning).....	15
2.1.1	Odometrie.....	15
2.2	Inerciální navigace	16
2.2.1	Gyroskopy	17
2.2.2	Akcelerometry	18
2.3	Sledování vodící čáry.....	18
2.3.1	Optický systém	19
2.3.2	Magnetický systém.....	19
2.4	Navigace pomocí taktilních a proximních senzorů.....	19
2.4.1	Taktilní senzory	19
2.4.2	Proximitní senzory	20
2.5	Northstar navigační systém.....	20
2.6	GPS - Global positionong systém	21
2.6.1	Satelitní segment	22
2.6.2	Kontrolní segment	23

2.6.3	Uživatelský segment.....	23
2.6.4	System Assisted GPS (A-GPS).....	24
3	Vypracování úloh pro studenty	25
3.1	Sledování vodící čáry	26
3.1.1	Popis programu.....	26
3.1.2	Spouštění programu.....	28
3.2	Navigace pomocí odometrie	29
3.2.1	Popis programu.....	29
3.2.2	Spouštění programu.....	31
3.3	Navigace pomocí dotykových a bezdotykových senzorů	32
3.3.1	Popis programu.....	33
3.3.2	Spouštění programu.....	36
4	Zhodnocení stávající konfigurace Festo Robotino	37
4.1	Držák pro uchopení	37
4.2	Některé další snímače	37
4.3	Infračervené senzory	38

Seznam obrázků

Obr. 1: Podvozek [2]

Obr. 2: Motor (1), Inkrementální čidlo (2), Všesměrový válec (3), Zařízení (4), Řemen (5) [2]

Obr. 3: Řídící jednotka [3]

Obr. 4: Inkrementální snímač (1), Anti-kolizní snímač (2), Infračervený snímač (3) [2]

Obr. 5: Gyroskop [6]

Obr. 6: Vlevo Northstar projektor, vpravo senzor [14]

Obr. 7: Rozložení bodů v prostoru [14]

Obr. 8: Satelitní segment [13]

Obr. 9: Bludiště

Obr. 12: Nastavení kamery

Obr. 10: Navigace pomocí sledování vodící čáry

Obr. 11: Testovací prostředí Robotino® SIM

Obr. 13: Main program

Obr. 14: Reset funkčního bloku "Odometrie"

Obr. 15: Krok 1

Obr. 16: Krok 2

Obr. 17: Globální osový systém

Obr. 18: Rozložení snímačů

Obr. 19: Main program - Paralelní zapojení

Obr. 20: První krok programu

Obr. 21: Tabulka pro převod voltů na cm [2]

Obr. 22: Druhý krok programu - rotace kolem osy z

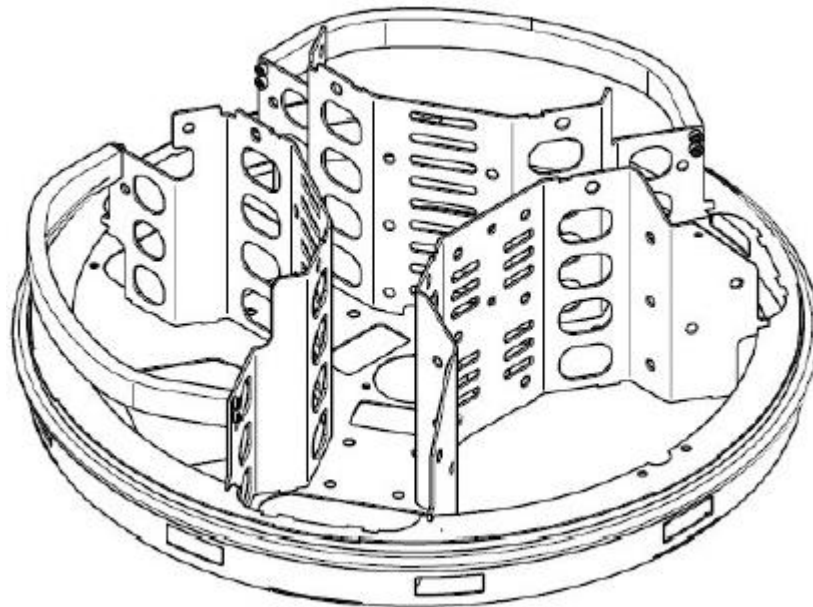
Obr. 23: Držáky na uchopení od firmy Festo [15]

1 Seznámení se se systémem Robotino

Robotino je plně funkční, vysoce kvalitní mobilní robotický systém se všesměrovým pohonem, určený pro výukové účely. Slouží nejenom k výuce a vzdělání, ale i k výzkumu. Je skvělou pomůckou k praktickému vyučování a velmi vhodný pro výuku žáků a studentů studujících mechatroniku, automatizaci a příbuzné předměty. Mezi jeho přednosti patří zejména jednoduché a logické naprogramování, například přes notebook, nebo dokonce i přes iPad. [1]

1.1 Podvozek

Podvozek má tvar kruhu, jehož průměr je 37 cm. Je velmi robustní, neboť se skládá z laserem svařované korozi-vzdorné ocelové platformy. To se ovšem na druhé straně projevilo na hmotnosti celého robota, která je 11 kg. [1]

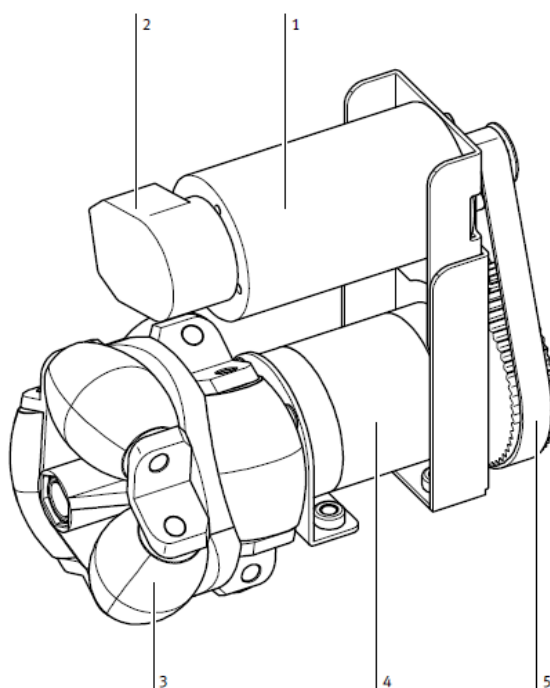


Obr. 1: Podvozek [2]

1.2 Pohonná jednotka

Robotino je všesměrový robot. Všesměrovost zde zajišťují tři všesměrová kola Mecanum uspořádaná v úhlu 120°. Všechna jsou samostatně ovládatelná. Každé všesměrové kolo je přímo připevněno na svou převodovku. Převodovky jsou poháněny stejnosměrným motorem přes řemen a dvě ozubená kola. [2,4]

Každá z těchto pohonných jednotek se skládá ze základních částí, které můžeme vidět na Obr. 2.



Obr. 2: Motor (1), Inkrementální čidlo (2), Všesměrový válec (3), Zařízení (4), Řemen (5) [2]

Robotino může dosáhnout rychlosti až 2 m/s. Tak vysoké rychlosti se ovšem nedoporučují používat. Robot je totiž při tak vysoké rychlosti velmi nestabilní, a to i na rovném povrchu. Navíc již v těchto větších rychlostech nezvládá vyhodnotit údaje o překážkách.

1.3 Řídící jednotka

Řídící jednotka je postavena na průmyslovém PC 104. Tento počítač má procesor s rychlostí 300 MHz, paměť 128 MB SDRAM a podporuje operační systémy typu Windows, Linux, DOS a jiné. Další výhodou je množství rozhraní, která počítač vlastní, jsou to například integrovaná síťová karta 10/100 LAN (na tento port je připojen Wi-Fi adaptér), 2x USB 2.0 (na jeden port je připojena webkamera), VGA. [2,3]



Obr. 3: Řídící jednotka [3]

1.4 Senzory

Robotino disponuje třemi základními typy senzorů, který zajišťují bezpečný pohyb. Dále je vybaven kamerou s rozlišením VGA. Všechny tři typy senzorů můžeme vidět na Obr.3.



Obr. 4: Inkrementální snímač (1), Anti-kolizní snímač (2), Infračervený snímač (3) [2]

1.4.1 Inkrementální snímač

Skutečná rychlost každého motoru se měří v otáčkách pomocí inkrementálního snímače. Pokud se aktuální rychlost motoru odchyluje od požadované hodnoty, může být upravena tak, aby odpovídala požadované hodnotě pomocí PID regulátoru, jehož parametry jsou nakonfigurovány pomocí Robotino®View softwaru. [2]

1.4.2 Anti-kolizní senzor

Anti-kolizní čidlo je tlakovým senzorem, který je ukrytý pod gumou. Tato guma vede kolem celého obvodu podvozku. Spínací komora se nachází uvnitř plastového profilu. Uvnitř komory jsou umístěny dva vodivé povrchy. Tyto plochy jsou zkratovány, i když dojde na pásu k minimálnímu tlaku. Spolehlivě rozeznatelný signál je potom přenášen na řídicí jednotku, která Robotino uvede do klidu. [2]

1.4.3 Infračervené snímač

Robotino je vybaveno devíti infračervenými senzory pro měření vzdálenosti. Ty jsou namontovány na podvozku v úhlu 40° vůči sobě. Robotino může pomocí těchto senzorů zkoumat všechny okolní plochy. Díky tomu se může vyhýbat překážkám. Senzory jsou schopny přesného nebo relativního měření vzdálenosti k objektům na vzdálenost 4 až 30 cm. Připojení senzoru je obzvláště jednoduché, zahrnují jen jeden analogový výstupní signál a napájení. Senzory vyhodnotí vzdálenost a přečtou si ho jako analogový signál. Bohužel jsou standardně umístěné infračervené senzory příliš těsně nad zemí a to brání robotu rozeznat ty překážky, které stojí na nohách cca od 4 cm výše (například skříň na nohách). Překážky je proto při experimentech vhodnější simulovat například papírovými krabicemi. [2, 3]

1.5 Programování robota

Vlastní připojení počítače k Robotinu a jeho programování je poměrně jednoduché. Poté co Robotino zapneme, načte se vlastní OS (v našem případě se jedná o Linux) a vytvoří díky zabudovanému Wi-Fi adaptéru vlastní bezdrátovou počítačovou síť, ke které se student připojí ze svého laptopu nebo školního počítače. Zapneme program Robotino®View a přihlásíme ke konkrétnímu Robotinu, které je určeno vlastní IP adresou. Dále jen vytvoříme logický program, podle kterého má robot fungovat, spustíme jej a můžeme jezdit. IP adresu lze u každého robota změnit, proto je možné využít při výuce či jiné akci několik těchto mobilních robotů najednou. [3]

Ke tvorbě logického programu je Robotino vybaven velmi přehledným softwarem pro programování, vhodným k praktickému pochopení toho, jak fungují

jednotlivé části systému, například senzory. Robot se programuje pomocí softwaru Robotino®View. Ten je určen zejména pro méně zkušené uživatele. Jedná se o grafické prostředí vývojových diagramů, kde se program vytváří vkládáním „Ikon“ do prostoru k tomu určenému a jejich následným propojováním. Tím vytvoříme potřebné vztahy pro požadovanou funkci programu, respektive požadované chování robota v té které situaci. Dvojitým poklepáním na ikonu lze otevřít a projít detaily nastavení jednotlivých částí podprogramů, a pokud jsme si jistí správným zapojením, můžeme program spustit. Po spuštění programu se Robotino začne chovat tak, jak je dáno programem. [3]

Zkušenější uživatelé mohou Robotino naprogramovat pomocí programovacích jazyků C, C++, Java, Net či v prostředí Matlab, Simulink a LabVIEW. Robota lze samozřejmě naprogramovat i tak, aby nebyl závislý na programu z PC. Ten se nahraje do paměti robota. Pak již stačí robota zapnout a ten provede daný logický program, můžeme například pomocí kamery rozlišovat černou pásku na zemi a řídit se podle ní anebo se jen vyhýbat překážkám. [3]

Při zkoušení nového programu je vhodné Robotino umístit na podstavec, který zvedne kolečka robota do vzduchu a z kterého nám robot nikam neuteče. Z vlastních zkušeností mohu toto řešení doporučit, studenti nejsou neomylní a v programu se může kdykoliv objevit chyba, která by mohla způsobit poškození Robotina.

1.6 Parametry

- Hmotnost: 11 Kg
- Šířka: 37 cm
- Výška: 26 cm (včetně kamery)
- Rychlost: až 2 m/s
- Napájecí napětí: 24 V DC, 4,5 A (2x AKU 12V, 4Ah)
- Výstupy: 8x digitální
8x analogové (0 - 10 V)
- Vstupy: 8x digitální
2x relé

2 Možnosti navigace mobilních robotů ve vnitřním prostření

Pro navigaci robotů ve vnitřním prostření se nejčastěji využívá.

- Navigace výpočtem
- Inerciální navigace
- Sledování vodící čáry
- Navigace pomocí dotykových a bezdotykových senzorů
- Northstar navigační systém
- GPS systém

Navigace tvoří u mobilního robota jeho elementární dovednost. Tím se rozumí jak je robot schopný plánovat cestu z bodu A do bodu B. Robot se pohybuje v pracovním prostření, ve kterém se vyskytují různé překážky. Proto je cesta plánována s ohledem na některé požadavky. Nejdůležitějším požadavkem pro mobilního robota je bezkolizní cesta. Dále jsou také kladeny požadavky na bezpečnost a délku cesty. [5]

Při navigaci robota se setkáváme s několika problémy. Jedním z nich je problém lokalizace robota, tedy určení jeho přesné pozice. Polohu robota v třírozměrném prostoru určujeme pomocí tří lineárních stupňů volnosti (x, y, z) a pomocí tří stupňů volnosti rotace (Θ, ψ, φ). [5]

Jedním z možných způsobů lokalizace kolového robota jako je Robotino je odometrie, která počítá ujetou vzdálenost z odvalování kol robota. Tato metoda se ovšem nedá moc dobře aplikovat u našeho Robotina, protože má všesměrový podvozek, u kterého se nedá dostatečně přesně spočítat odvalení kol. U všesměrového podvozku jaký se nejčastěji využívá IRC snímače v kombinaci se snímači inerciálními. [6]

2.1 Navigace výpočtem (Dead reckoning)

Matematická procedura pro určování současné pozice robota, pomocí postupného přičítání díky známému kurzu a rychlosti v průběhu času. (Nejjednodušší implementací této metody je odometrie) [8]

2.1.1 Odometrie

Odometrie je věda, která se zabývá odhadem aktuální polohy u mobilních robotů. Základem odometrie je znalost geometrického modelu robota. V podstatě měří otáčení kol, čímž určí vzdálenost, kterou vozidlo urazilo od začátku pohybu. Z toho vyplývá, že tato metoda se používá u robotů s kolovým podvozkem. Nejčastěji bývá

realizována pomocí inkrementálních čidel, za předpokladu že otáčky kol mohou být převedeny na lineární posuv vzhledem k podlaze. To nám ovšem způsobí nárůst chyb, který způsobují nepřesnost této metody. [6, 7]

Systematické chyby

- Rozdílné průměry kol
- Aktuální průměr rozdílný od normálního
- Výchytky kol
- Omezený rozsah enkodéru
- Omezená vzorkovací frekvence enkodéru

Ostatní chyby

- Jízda po nerovném povrchu
- Jízda přes neočekávané překážky
- Prokluz kol při kluzkém povrchu, rychlé otáčení atd.

Nejčastěji používané sensory pro dead reckoning jsou: tachodynamy, potenciometry, optické, magnetické, induktivní a kapacitní enkodéry, magnetometry a gyroskopy. [8]

Použití této metody patří mezi ty levnější, ale i přesto můžeme docílit velmi přesného měření pomocí inkrementálních enkoderů. Avšak problémem je neustále narůstající chyba, kterou není možné eliminovat. [8]

2.2 Inerciální navigace

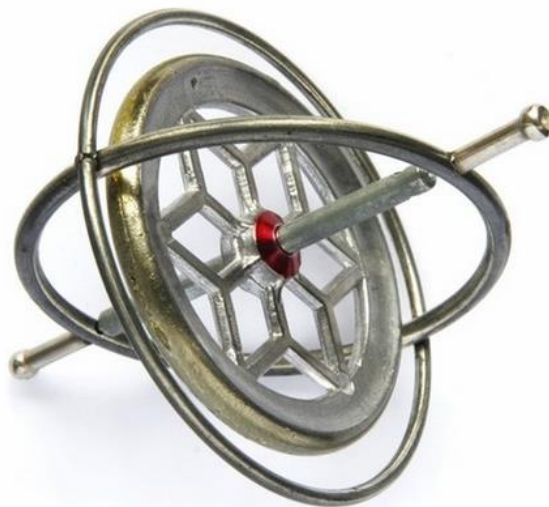
Využívá inerciálních senzorů, jako jsou například gyroskopy a akcelerometry. Jejichž pomocí následně měří rychlost pohybu vozidla a poté určuje jeho pozici díky zrychlení. Celý princip inerciální navigace funguje tak, že pokud známe startovací pozici vozidla a zaznamenáme-li během jeho jízdy všechna zrychlení ve všech osách, můžeme dopočítat jeho současnou rychlost a pozici.

Ze zrychlení je možné poměrně jednoduše zjistit pomocí integrace rychlost a pomocí další integrace pozici. Problém integrace je v tom, že je integrováno nejen zrychlení objektu, ale i všechny chyby vznikající při měření objektu. Proto byla technologie inerciální navigace neustále vylepšována v průběhu minulých desetiletí a nyní je již široce používána a to zejména v letadlech, lodích, tancích, ponorkách, raketách a střelách. Nevýhodou této navigace je její vysoká cena. [6, 8]

2.2.1 Gyroskopy

Gyroskopy jsou již dlouhou dobu známy a využívány pro měření a určování změny polohy nebo natočení libovolného předmětu, ke kterému jsou připevněny. Přesnost gyroskopu pak závisí na stabilitě udržení jeho otáček. Dříve však bylo možné použít jejich mechanické provedení, případně optické s využitím světla, světlovodných vláken. Dnes je již možné najít je i v integrované podobě klasických součástek obsahujících mimo samotný snímač i celou škálu vyhodnocovacích obvodů a logiky. Výstup je pak analogový, digitální nebo obojí. Díky tomu lze gyroskopy použít i v běžných aplikacích, nejen ve vědě a výzkumu. Například dnes běžně používané určování pozice a sledování pohybu objektů přes GPS by bez gyroskopů nebylo možné. [9]

Gyroskop je vlastně setrvačnick. Může být popsán jako těžké kolo otáčející se v ložiscích s nepatrným třením. Otáčející se setrvačnick má moment hybnosti, takže jeho osa bez působení vnějších sil udržuje stále stejný směr. Je to tedy zařízení, které dokáže určit svoji orientaci v prostoru a to vše díky tomu, že kolo nemění při rotaci svou polohu a tím dává informaci o orientaci dopravního prostředku v prostoru. [10]



Obr. 5: Gyroskop [6]

Gyroskopy lze dále rozdělit podle použitého fyzikálního principu na:

- Mechanické
- Kvantové
- Jaderné
- Elektrické
- Optické

2.2.2 Akcelerometry

Akcelerometry jsou senzory využívající se pro měření statického a dynamického zrychlení, naklonění těles a také měření vibrací. [11]

Pracují na principu, který využívá setrvačnosti hmoty pro měření rozdílu mezi kinematickým zrychlením vzhledem k nějakému inerciálnímu prostoru a gravitačním zrychlením.

V dnešní době se již klasické mechanické senzory dávají na ústup a jsou nahrazovány součástkami vyrobenými pomocí MEMS technologie. Díky této technologii je možné dosahovat mnohem menších rozměrů součástek, menší spotřeby energie a také nižších cen. Tato technologie má samozřejmě i nevýhody a tou největší je nedostatečná přesnost pro mnohé aplikace.

Akcelerometry se rozdělují podle použité technologie na:

- Piezoelektrické akcelerometry - využívají piezoelektrický krystal, který generuje náboj úměrný působící síle, která při zrychlení působí na každý objekt.
- Piezorezistivní akcelerometry - využívají mikrokřemíkovou mechanickou strukturu, kde zrychlení odpovídá změně odporu
- Akcelerometry s proměnou kapacitou - využívají mikrokřemíkovou mechanickou strukturu, kde zrychlení odpovídá změně kapacity

2.3 Sledování vodící čáry

Jde o jednu z nejjednodušších metod pro navigaci robotů. Princip činnosti lze obecně popsat tak, že robot reaguje na snímanou pozici nějaké externí trasy. Je jasné, že metoda má dvě značná omezení. První omezení je v modifikaci prostředí ohledně instalace vodících čar. Druhé omezení je takové, že robot se nemůže dostat na libovolné místo, pouze tam, kde jsou instalované vodící čáry. I přes tyto omezení se jedná o jednu z nejpoužívanějších metod v průmyslu. Důvodem oblíbenosti této metody je jednoduchost, nízká cena a velmi vysoká spolehlivost. Při využívání této metody je možno použít prakticky neomezené množství strojů, pracujících současně bez nebezpečí vzájemného rušení. [8]

Dnes nejčastěji používané metody sledování vodící čáry zahrnují zpravidla nějaký typ pevně nainstalovaného vodícího drátu nebo pruhu a senzory, které jsou umístěny v přední části stroje. Nejčastěji se využívá optických a magnetických systémů navádění. [8]

2.3.1 Optický systém

Optické vodící čáry jsou velmi jednoduché na instalaci a je jednoduché je změnit. Díky tomu, že vodící čáry jsou v tomto případě pasivní, není limitovaná délka vodících čar. Na rozdíl od systémů s naváděním pomocí vodičů zde nevádí malá přerušování vodící čáry. Hlavní nevýhodou těchto systémů je to, že optický naváděč je zpravidla nekrytý a obvykle navíc vyčnívá nad povrch a může tedy být snadno poškozen, ušpiněn nebo zakryt. [8]

2.3.2 Magnetický systém

Druhým nejpoužívanějším principem pro navádění je magnetická páska, která je připevněna nebo zapuštěna do podlahy. Magnetické pole je pak snímáno magnetometrií či halovými sondami umístěnými opět v přední části robota. Výhody tohoto principu jsou nenáročnost na údržbu, značná odolnost, imunita vůči přerušování čáry a neomezená délka naváděcích čar. Magnetických čar je také možno využít při vytváření mřížek a při spojení s odometry dojde k podstatnému omezení chyb. [8]

2.4 Navigace pomocí taktilních a proximálních senzorů

2.4.1 Taktilní senzory

Taktilní senzory jsou u mobilních robotů pro poskytování informací při kolizi robota s překážkou. Navigace pomocí těchto senzorů je založena na fyzickém kontaktu s překážkou. Jsou to tzv. dotykové senzory. Tyto senzory se nejčastěji připevňují na robota tam, kde dochází ke kontaktu. Umísťují se tedy například na držáky pro uchopení, do nohou robotů, kteří jsou pochoduující nebo do nárazníku robota, jako je to i v našem případě u Robotina.

Taktilní senzory se rozdělují na tři nejpoužívanější skupin:

- Taktilní antény
- Taktilní nárazníky
- Pole snímačů

Taktilní senzory jsou velmi jednoduché a zároveň i velmi levné, což je jejich výhodou, ale nalezneme samozřejmě i určité nevýhody, jako je koroze kontaktů a deformace míst, kde dochází ke kontaktu. I přes tyto nevýhody jsou taktilní senzory velmi používané a oblíbené senzory.

2.4.2 Proximitní senzory

Proximitní senzory jsou u mobilních robotů pro poskytování varování o přítomnosti překážky. Na rozdíl od taktilních senzorů, jsou proximitní senzory bezdotykové a nemusí tedy dojít k přímému kontaktu senzoru s překážkou.

Proximitní senzory se dělí do několika nepoužívanějších skupin, kde každá z těchto skupin má jinou oblast použití, která je dána rozlišením, dosahem, frekvencí snímání a prostředím. Jsou to kapacitní, indukční, magnetické, mikrovlnné, optické a ultrazvukové.

Proximitní a taktilní senzory se zpravidla používají jenom jako doplněk k dalším senzorům, které jsou na robotovi umístěny. Výjimkou jsou pak ultrazvukové a optické proximační senzory, které jsou často využívány i samostatně k navigaci robotů ve vnitřních prostorech. [8]

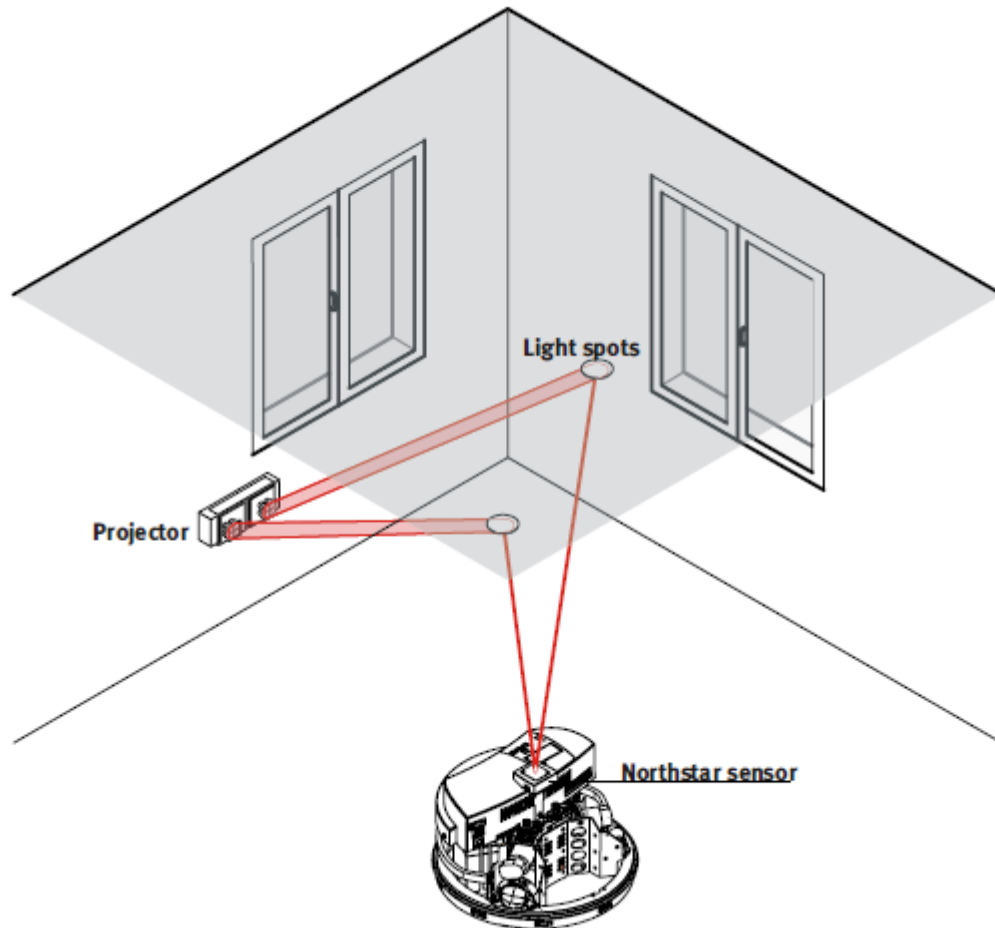
2.5 Northstar navigační systém

Northstar je infračervený systém sledování, který využívá infračervené světelné body jako body orientace. Infračervený sensor zjistí svou pozici a orientaci založenou na umístění dvou světelných bodů. Systém Northstar se skládá z Northstar projektoru a Northstar senzoru, které můžeme vidět na Obr. 6.



Obr. 6: Vlevo Northstar projektor, vpravo senzor [14]

Rozlišování mezi dvěma nebo více různými projektory v daném Northstar systému je zajištěno pomocí různě blikajících frekvencí na jednotlivých projektorech. Umístění dvou bodů světla vůči sobě navzájem tak může být použito pro určení polohy.



Obr. 7: Rozložení bodů v prostoru [14]

System Northstar se používá pro navigaci nebo pro sledování objektů. Pro účely navigace, je projektor namontován v nehybném stavu poblíž aplikace, ale senzor je připevněn přímo na Robotinu. V případě sledování objektů se senzor a projektor musí prohodit. [14]

2.6 GPS - Global positionong systém

GPS je globální polohový systém, kterým lze určit polohu objektu kdekoli na Zemi bez ohledu na denní dobu či nepřízeň počasí. Polohou se rozumí určení souřadnic severní či jižní šířky a východní či západní délky, ale zároveň i určení nadmořské výšky. Systém je tedy schopen určit polohu v třírozměrném prostoru, a to s velkou přesností (cca 10m). Jelikož bylo GPS primárně navrženo pouze pro vojenské účely, tak

byla tato přesnost v dřívějších letech záměrně zhoršována, jako jedno z mnoha opatření, proti úmyslnému zneužití. Toto záměrné zhoršování přesnosti bylo vypnuto v květnu roku 2000 prezidentem USA, jako nejvyšším ozbrojeným velitelem.

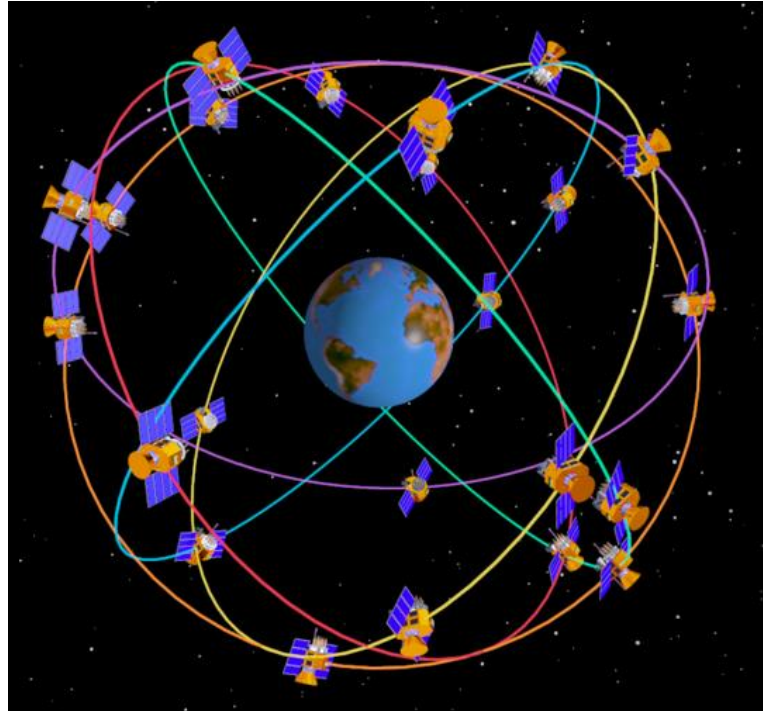
Funkce, kterou GPS nabízí, nachází v posledních letech čím dál více uplatnění v různých oborech. Nejširší využití našla v dopravě a to až už v určování polohy plavidel, letadel či pozemních dopravních prostředků, tak i jako navigace v neznámých oblastech v případě turistiky, nebo ke sledování zásilek či vyhledávání kradených automobilů. [8, 12]

System GPS lze rozdělit do tří základních segmentů:

- Satelitní
- Kontrolní
- Uživatelský

2.6.1 Satelitní segment

Satelitní segment je tvořen družicemi, které obíhají kolem Země. V dřívějších letech bylo na oběžné dráze 24 družic, které obíhali pravidelně, nyní je na oběžné dráze 32 družic, které obíhají nepravidelně ve výšce přibližně 20 200 km. Každá družice oběhne Zemi za 11 hodin a 58 minut. Oběžných drah pro družice je 6 a jsou navrženy tak, aby z jakéhokoliv místa na Zemi bylo v jakýkoliv čas 6 viditelných družic. Pro určení přesné polohy je potřeba pouze 4 družice, ty další dvě jsou pouze rezerva pro místa, kde výhled zakrývají například budovy či terénní nerovnosti.



Obr. 8: Satelitní segment [13]

2.6.2 Kontrolní segment

Kontrolní systém se skládá ze stanic umístěných na zemském povrchu. Tvoří ho pět řídicích a kontrolních stanic rozmístěných po celém světě. Z toho jsou čtyři stanice automatické bez obsluhy a jedna je hlavní řídicí s lidskou obsluhou, která je umístěna v Coloradu Springs. Tyto stanice provádějí permanentní sledování družic satelitního segmentu a spravují celý systém GPS.

2.6.3 Uživatelský segment

Uživatelský systém je koncové zařízení, které je umístěno přímo u uživatele. V současné době je na světě mnoho značek, které vyrábí pro uživatele svoje vlastní zařízení a je jen na uživateli jaké z nich si vybere. Každé zařízení se samozřejmě liší výbavou, ale všechny pracují na stejném principu.

Uživatelský segment na základě přijímaného signálu provádí výpočet souřadnic polohy zařízení. V signálu, který byl odeslán pomocí družice ke koncovému zařízení, je zakódována informace o čase. Druhý čas, který potřebujeme pro výpočet, je čas, kdy byl signál přijat koncovým zařízením. Když tyto dva časy od sebe odečteme, získáme čas, který byl potřebný pro dopravení informace ke koncovému zařízení. Tento čas ještě musíme vynásobit rychlostí světla a poté dostaneme vzdálenost uživatelského zařízení od družice. Poté abychom mohli vypočítat tři požadované hodnoty, které přesně definují naši polohu, stačí sestavit tři rovnice o třech neznámých a pomocí matematických

postupů je dopočítat. Problém však nastává u odchylky atomových hodin družice, protože koncové zařízení má méně přesné hodiny z důvodu vysoké ceny atomových hodin. Tento problém je vyřešen pomocí signálu ze čtvrté družice, který doplní soustavu rovnic o čtvrtou neznámou, kde bude právě tato odchylka naší neznámou.

2.6.4 Systém Assisted GPS (A-GPS)

Systém GPS pro určení přesné polohy potřebuje znát rozmístění viditelných satelitů na obloze a mít synchronizované vnitřní hodiny. Každý satelit vysílá informace o své přesné poloze jednou za 30 sekund, rychlostí 50 b/s a proto v ideální situaci potřebuje přijímač minimálně tuto dobu pro získání kompletní zprávy. Systém A-GPS nám umožňuje rychlejší zjišťování pozice. Pro rychlejší určování pozice máme dvě možnosti. [16]

- Získání důležitých informací pomocí internetu
- Výpočet pozice pomocí výkonného serveru, namísto interního procesoru

Kvůli přesnosti GPS navigace v řádu desítek metrů, není tato navigace moc vhodná, jako navigace pro mobilní roboty ve vnitřním prostředí. Ale právě díky systému A-GPS a jemu podobným, kdy dochází k propojování GPS s internetem, se nám tato přesnost zlepšuje až na jednotky metrů. Pokud by se v budoucnosti tato přesnost ještě zlepšila, mohla by se GPS navigace začít používat i ve vnitřních prostorech.

V méj bakalářské práci tuto navigaci zmiňuji proto, že momentálně je to nejrozšířenější druh navigace a v budoucnu by mohla mít ještě spoustu dalších využití.

3 Vypracování úloh pro studenty

Pro studenty jsem vytvořil 3 úlohy, na kterých se naučí navigovat Robotino třemi rozdílnými navigacemi. Úlohy jsou zvolené tak, aby se studenti naučili ovládat všechny senzory, které Robotino v základním vybavení nabízí.

K samotnému programování jednotlivých úloh jsem využil program Robotino® View, který nabízí firma Festo Didactic zdarma ke stažení na jejich internetových stránkách. K simulaci úloh jsem využil programu Robotino® SIM, který je ovšem zdarma pouze v demo verzi, kde není možné posouvat překážky, nebo například měnit vodící čáru využitou v programu "Sledování vodící čáry". K otestování funkčnosti programu je dostačující i demo verze a není tedy nutnost zakupovat verzi Professional.

Součástí vypracování úloh je i návrh bludiště, na kterém by si studenti mohli na hodinách robotiky vyzkoušet všechny tři úlohy a porovnat, jaká navigace je pro Robotino nejvýhodnější.



Obr. 9: Bludiště

Robotino je se svými 32cm v průměru docela robustní robot. Bylo tedy jasné, že i samotné bludiště bude muset být docela velké. Zvolil jsem čtvercový rozměr 2x2 metry a navrhl ho tak, aby zde bylo možné simulovat různé situace.

Při vytváření byl kladen důraz na to, aby bylo bludiště možné po každé ukázce rozložit na menší části a uskladnit. Proto je bludiště sestaveno ze čtyř čtvercových desek o velikosti 1x1 m, které jsou k sobě přidělány pomocí spony, kterou lze otevřít. Bludiště je tedy možné rozebrat a uskladnit.

3.1 Sledování vodící čáry

Jako první úlohu jsem zvolil jednu z nejjednodušších navigací, která je na Robotinu možná naprogramovat. Přes svoji jednoduchost, možná i kvůli své jednoduchosti se jedná o nejpoužívanější navigaci v průmyslu. Je to navigace, pomocí sledování vodící čáry.

3.1.1 Popis programu

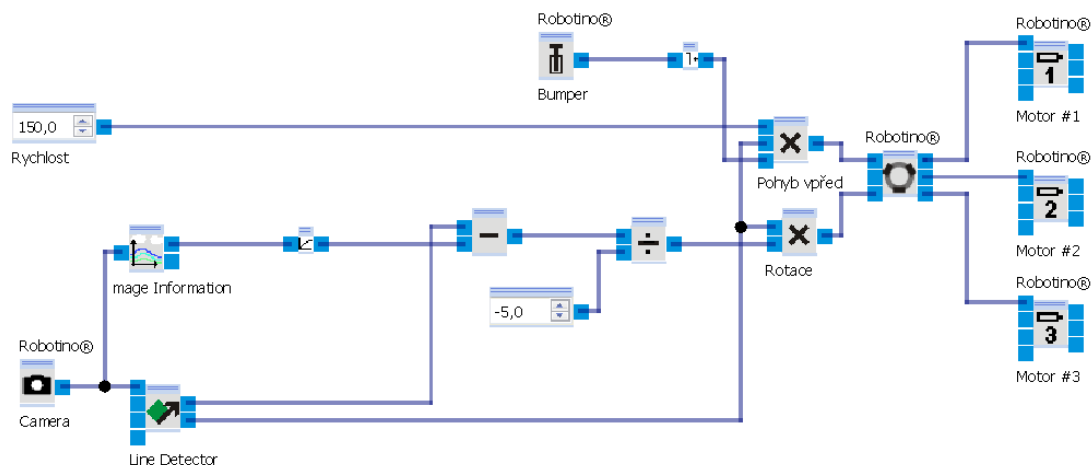
Rozhodně nejdůležitější komponentou je zde "Camera", kterou přenášíme samotný obraz pro Robotino. Před samotným programováním programu, je vhodné otestovat funkčnost kamery. Otestování funkčnosti kamery je velmi jednoduché, stačí k tomu pouze funkční blok "Camera", který nemusíme ani nikam připojovat. Po navázání spojení s Robotinem a spuštění programu, bychom měli na kameře vidět přesně to, co vidí Robotino.

Podle návodu, který firma Festo Didactic dodává k Robotinu, bylo v předchozích verzích programu Robotino® View možné na kameře měnit různé barevné moduly, se kterými kamera pracovala a také rozlišení kamery. V nejnovější verzi 2.8.4, se kterou jsem pracoval, jsem ovšem toto nastavení nikde nenašel a tak se musíme smířit s pevně nastaveným barevným modulem RGB, který podle testů vycházel pro sledování vodící čáry nejlépe. Možnost nastavení rozlišení zde zůstala a můžeme si vybrat z rozlišení 320x240, nebo 640x480.

Další velmi důležitou komponentou, které využívám v tom projektu je " Line Detector". Tento funkční blok slouží k detekci sledované linky na vstupním obrazu, který mu dodává kamera. Využil jsem zde "Imput", kam jsem připojil kameru, abych měl obraz a dále taky oba výstupy "x" a "Line found". Výstup "x" určuje x-pozici zjištěného řádku v pixlech. To znamená, že pokud je ve středu jak má, indikuje 0, je-li ovšem na jiné pozici, indikuje hodnoty od -160 do 160, podle toho na jaké straně je a podle toho se Robotino poté natáčí. Výstup "Line found" říká, zda Robotino vidí, nebo nevidí vodící čáru.

Poslední velmi důležitou komponentou, které využívám speciálně u této navigace je "Image Information". Pomocí "Image Information" získám výšku a šířku obrazu pořízeného kamerou. V tomto programu využívám pouze šířku obrazu, výšku řešit nemusím. Šířku je 320, protože jsem zvolil rozlišení kamery 320x240. Momentálně tedy získávám hodnoty od 0 do 320, ale jak jsem psal již výše, potřebujeme hodnoty od -160 do 160, abych od sebe mohl odečíst hodnoty z výstupu "Line Detector" a "Image Information". Tuto hodnotu bych již mohl posílat přímo na rotační vstup "Omnidrive", ale je vhodné ji ještě vydělit, aby rotace Robotina byla co nejmenší. Čím větší budeme mít nastavenou hodnotu rotace, tím pravděpodobněji dojde ke ztrátě linky a Robotino zastaví. Při testování na bludišti bylo zjištěno, že ideální je dělit hodnotou "-3". Při větších hodnotách, docházelo k tomu, že Robotino ztrácelo vodící čáru, při průjezdu zatáčkami.

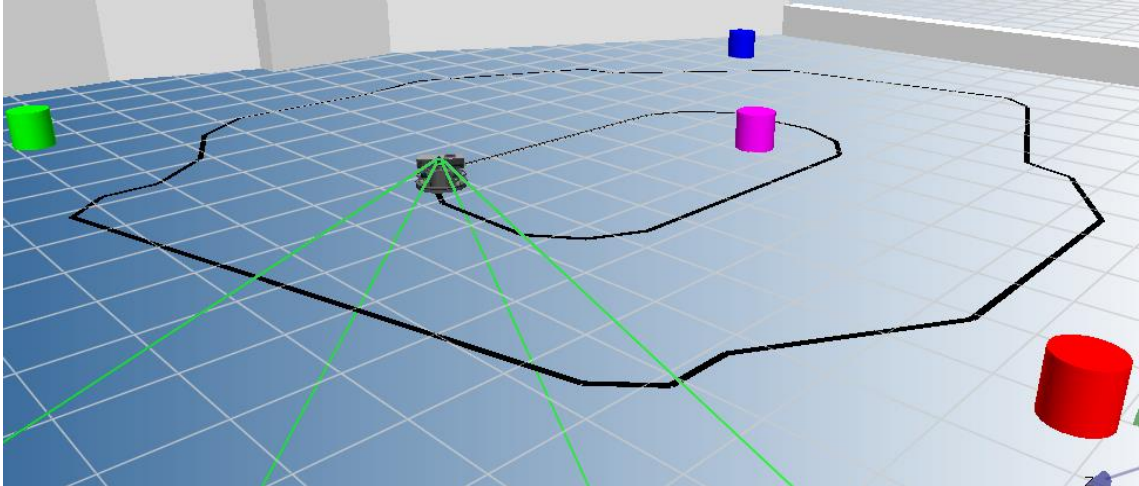
Pokud se podíváme na kompletní program na Obr.10, tak ještě stojí za zmínku funkční blok s názvem "Bumper" a samozřejmě "Omnidrive". Bumper slouží k indikaci nárazu Robotina do stěny. Při nárazu indikuje jedničku na výstupu, která je převedena invertorem na nulu. Bumper je v součinu společně s rychlostí a "Line found". Pokud je jeden z těchto funkčních bloku nulový, Robotino okamžitě zastaví. Poslední ze zmíněných bloků, což ovšem neznamená, že nejméně důležitým je "Omnidrive". U tohoto bloku využívám dvou vstupů i všech tří výstupu. Pokud se podíváme blíže, zjistíme, že vstupy jsou pohyby v osách. První vstup je pohyb v ose x v mm/s, potom máme pohyb v ose y, také v mm/s a jako poslední je rotace kolem osy z v deg/s. "Omnidrive" pomocí těchto vstupních veličin vypočítává rychlosti pro jednotlivé motory, kterou posílá na jednotlivé výstupy v rpm. Pokud máme takto sestavený program a myslíme, že je správně sestavený, můžeme zahájit testování.



Obr. 10: Navigace pomocí sledování vodící čáry

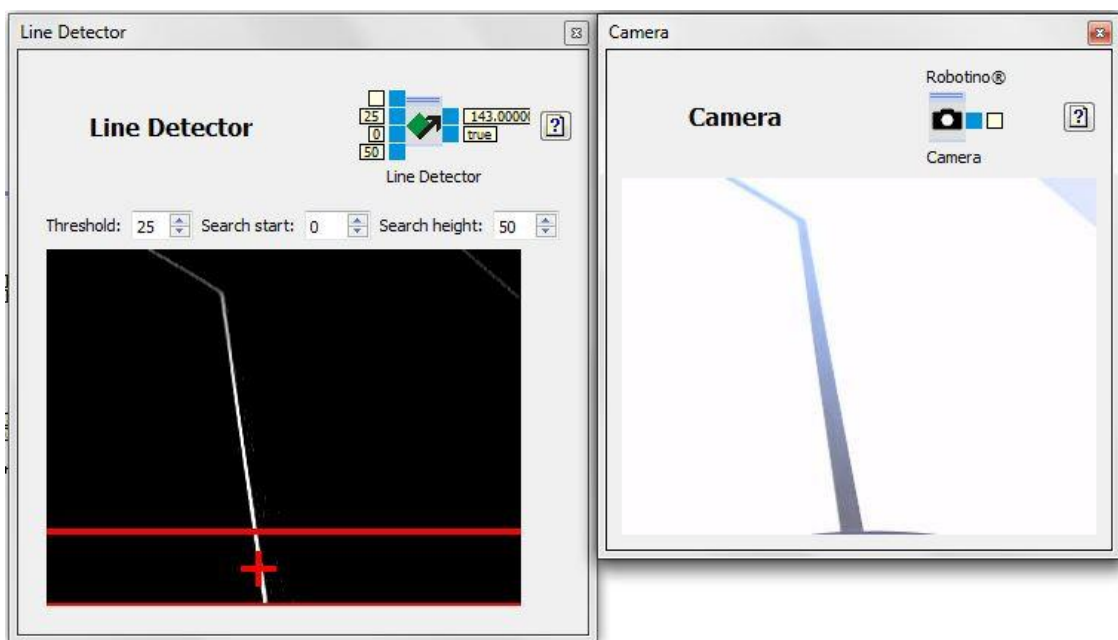
3.1.2 Spouštění programu

Jako první je dobré testovat hotový program v programu Robotino® SIM, kde neriskujeme, že zničíme Robotino špatným naprogramováním. Taky zde nemusíme řešit prostor, kterého Robotino pro svůj pohyb potřebuje docela dost.



Obr. 11: Testovací prostředí Robotino® SIM

Poté co program otestujeme v Robotino® SIM, můžeme Robotino postavit na start bludiště. Jako první, ještě před samotným startem, musíme nastavit kameru tak, abychom na ní viděli naši snímanou vodící čáru a zároveň abychom neviděli žádné kabely a ani podvozek. Pokud toto neuděláme správně, tak se Robotino nerozjede. Jak už jsem vysvětloval výše, je to z důvodů, že je Robotino nastavené tak, aby po ztrátě vodící čáry zastavilo a zabránilo se tak nechtěnému bourání do okolních předmětů. Jak správně nastavit kameru, můžeme vidět na Obr.12.



Obr. 12: Nastavení kamery

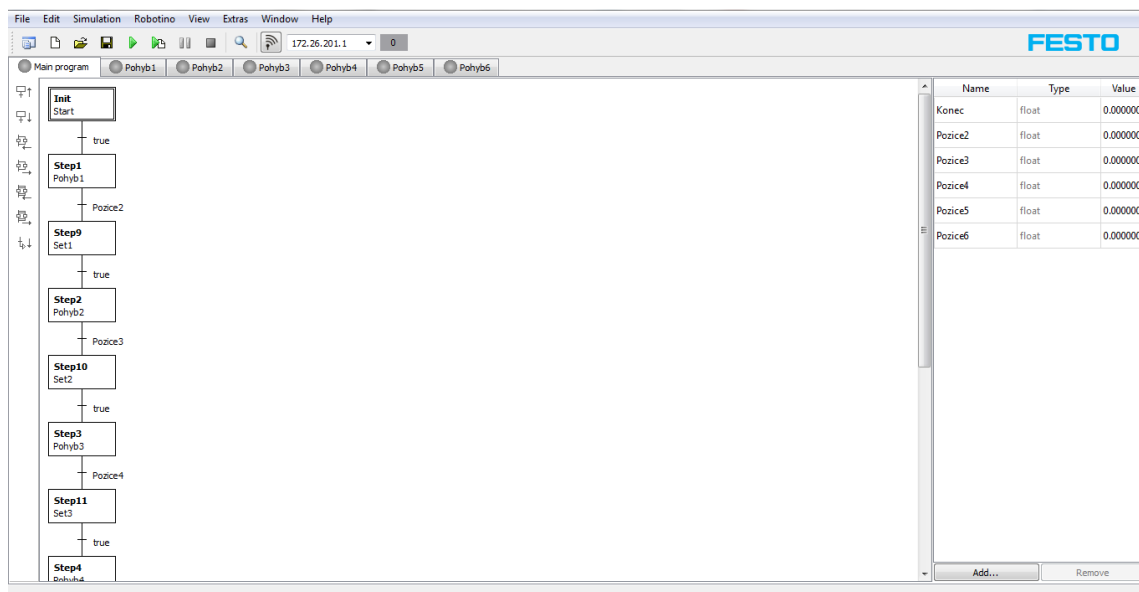
Rychlost doporučuji nastavit maximálně na 200 mm/s, i když je Robotino schopno jet rychlostí až 2 m/s. Při větších rychlostech než 200 mm/s hrozí ztráta vodící čáry a tedy zastavení Robotina.

3.2 Navigace pomocí odometrie

Odometrie je naprosto rozdílná navigace od předchozí navigace. U odometrie si nastavíme cestu, kterou chceme, aby Robotino provádělo a to bez jakékoliv vodící čáry. Odometrie vypočítává z odvalování kol od začáteční pozice, vzdálenost ujetou do další pozice. Proto si můžeme nastavit přesné vzdálenosti jaké Robotino ujede a podle toho plánovat jeho cestu přesně tam kam chceme.

3.2.1 Popis programu

Při programování odometrie, jsem se musel na program Robotino® View podívat trochu jinak. Jako první jsem zjistil, že budu potřebovat skoky, které tento program také samozřejmě nabízí. Skoky řešíme v části, která se nazývá "Main program", kterou můžeme vidět na Obr.13.

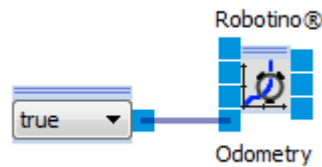


Obr. 13: Main program

Zde si musíme jako první jednotlivé skoky vytvořit. Skoky se vytváří v pravém sloupci, kde se vytváří pomocí tlačítka "add". Když si vytvoříme skoky, můžeme začít vytvářet schéma, kde budeme vytvářet samostatné kroky pomocí ikon v pravém sloupci. Vždy se vytvoří jeden krok a před ním i za ním umístíme daný skok. Pomocí těchto ikon vytvoříme schéma, kde mohou být skoky jak paralelní, tak sériové. Můžeme skákat z prvního na poslední, z posledního na prostřední, je to jedno, vše záleží nato, jak si skoky rozdělíme uvnitř kroků.

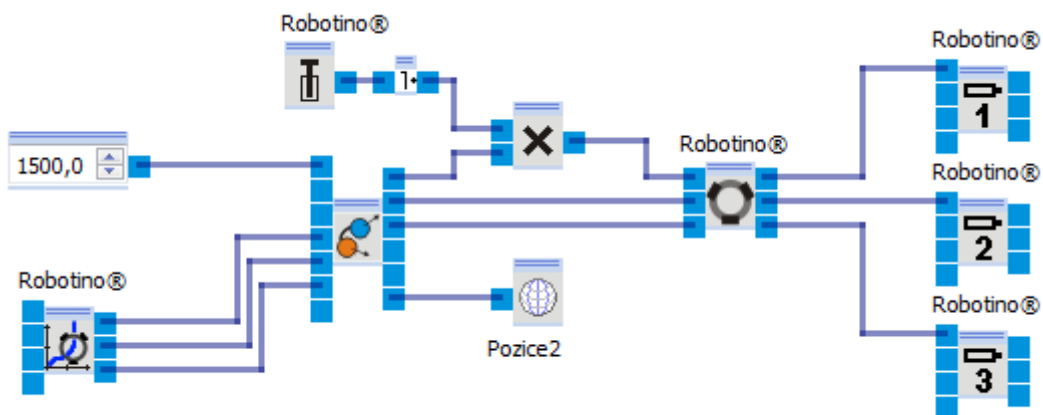
Jak můžeme vidět na Obr.13, můj program má 12 kroků a také 12 skoků. V každém kroku je umístěný program, který končí skokem před jiný krok. Takže pokaždé, co Robotino provede nějaký posun, dojde ke skoku na jiný krok, kde se zase provede další posun.

Hned na začátku programování jsem narazil na problém, který byl spojený s resetem odometrie. Odometrie nám slouží k počítání ujeté vzdálenosti Robotina. Proto bylo logické, aby po každém pohybu došlo ke smazání odometrie a já nemusel přičítat další posun k předchozímu. Ovšem při resetu v každém kroku, docházelo ke špatné navigaci robota. Proto jsem musel vytvořit samostatné kroky s názvem "Set". Ke skoku na tyto kroky dochází po každém ukončeném pohybu a poté dojde k nepodmíněnému skoku na další pohyb.

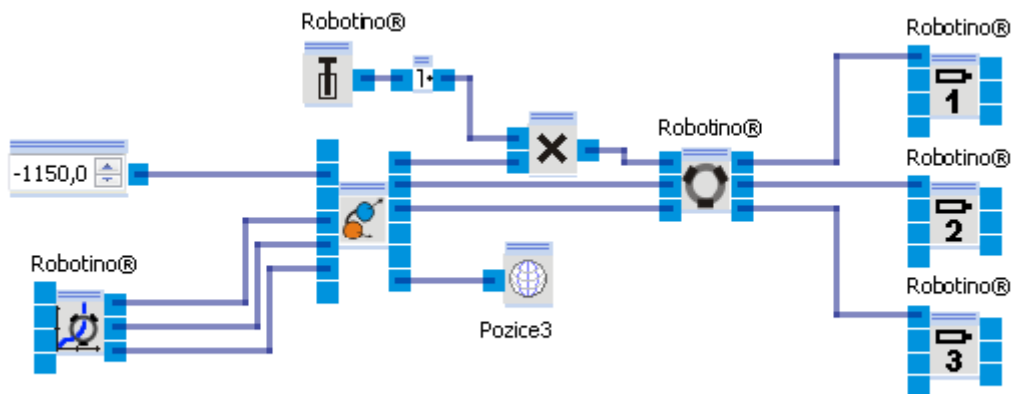


Obr. 14: Reset funkčního bloku "Odometrie"

Když se podíváme blíže na jednotlivé kroky programu, tak zjistíme, že schémata jsou všechna stejná. Jediný v čem se kroky vždy liší, je že každý krok provádí jiný pohyb, tudíž jsou vždy jiné konstanty přivedeny k "Position driver", jak můžete vidět na Obr. 15 a 16. Na Obr.15 je první pohyb Robotina, což je pohyb v ose x o 150 cm. Poté dojde ke skoku na reset funkčního bloku "Odometrie" a po resetu k nepodmíněnému skoku na druhý pohyb. V druhém pohybu je vidět, že Robotino vykoná pohyb 115 cm vpravo a takhle to pokračuje, dokud Robotino neprojde celé bludiště.



Obr. 15: Krok 1

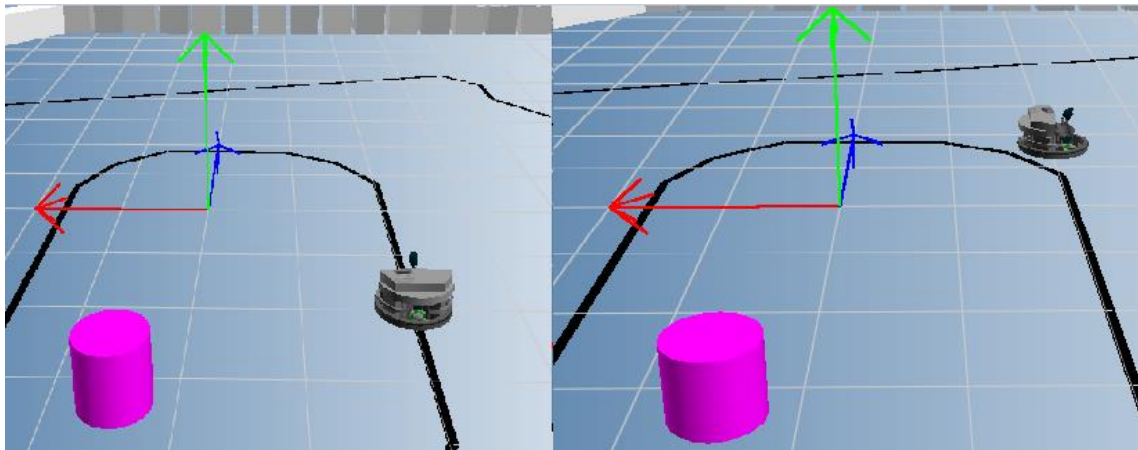


Obr. 16: Krok 2

Nyní se podívat blíže na jednotlivé komponenty, které jsou v tomto programu použity. Nejdůležitější komponentou je zde "Odometry", která se jmenuje podle této metody. Funkční blok "Odometry" vypočítává aktuální polohu Robotina. Další blok, který je velmi důležitý, porovnává danou aktuální polohu s požadovanou polohou a jmenuje se "Position Driver". Dokud ve funkčním bloku "Position Driver" nedojde ke shodě aktuální a požadované polohy, Robotino cestuje nastavenou cestou. Pokud dojde v "Position Driver" ke shodě, dojde ke skoku například na "Krok 2" jak můžeme vidět na Obr.16.

3.2.2 Spouštění programu

Hned na začátku programování tohoto programu se ukázala velká výhoda testování programu nejprve v Robotino® SIM. Poté co jsem spustil první verzi programu, Robotino nedělalo vůbec to, co mělo a kdybych to zkusil přímo na Robotinu, určitě by někam nabouralo. Po zkoumání a dlouhém trápení se s programem, jsem zjistil, že osový kříž je pro Robotino globální. Což můžete vidět na Obr. 17. Tedy, že se osy nenatáčejí spolu s Robotinem. Počátek souřadného systému je nastaven pomocí resetu odometrie na hodnoty (0,0,0). Muselo tedy dojít ke kompletnímu předělání pohybu Robotina. Tento problém jsem zjistil jenom díky programu Robotino® SIM, kde jsem si v "Options" nastavil zobrazení osového kříže a nevím, co bych dělal bez tohoto programu, protože nikde v návodu na Robotino ani samotný program Robotino® View o tom není ani jedna poznámka.



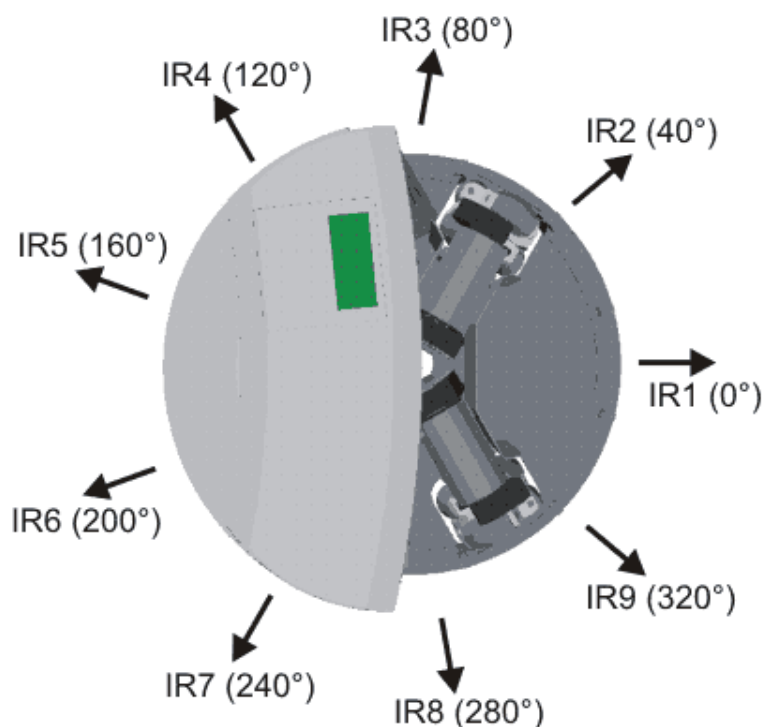
Obr. 17: Globální osový systém

Po odzkoušení můžeme nahrát program přímo do Robotina a vyzkoušet na vytvořeném bludišti stejně jako předchozí program. Jelikož se oba programy testují na stejném bludišti, můžeme testovat jaká navigace je rychlejší a lepší.

Doporučuji Robotino v bludišti nastavit co nejvíce na střed a rovně, protože zde nejsou zapojeny žádné snímače proti nárazu, je zde zapojen pouze "Bumper" jako v předchozím programu, takže po nárazu se Robotino zastaví.

3.3 Navigace pomocí dotykových a bezdotykových senzorů

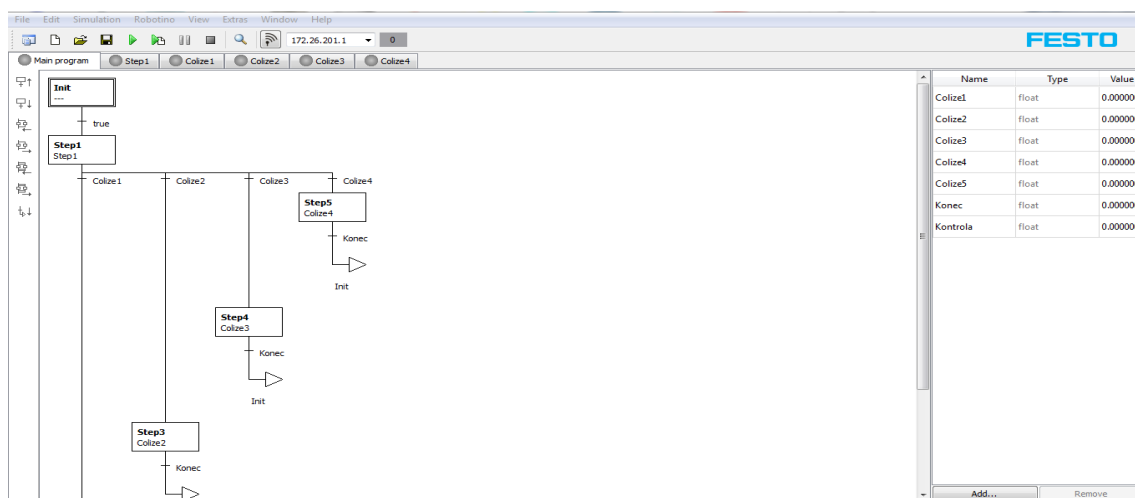
Navigace pomocí dotykových a bezdotykových senzoru je třetí navigací, kterou jsem vybral pro svoji bakalářskou práci. Vybral jsem jí proto, že se tato navigace zase liší od předchozích dvou navigací. Ano sice jsem zde také využíval například skoků, jako v předchozím programu, ale samotná navigace je jiná. Využil jsem infračervených senzorů, které jsou umístěny kolem dokola celého Robotina, ve výšce 4 cm a vzájemné vzdálenosti 40°, jak můžete vidět na Obr.18. Tyto snímače mají možnost snímat ve vodorovné poloze ve vzdálenosti od 1 do 40 cm. Bohužel umístění snímačů ve výšce 4 cm je dosti nepraktické, protože snímače nedokáží reagovat například na skříně, které jsou umístěny na nohou, vyšších než 4 cm. Z tohoto důvodu má naše bludiště zdi vyrobené z desky, která začíná u země a je vysoká 15 cm, aby je senzory naznamenaly.



Obr. 18: Rozložení snímačů

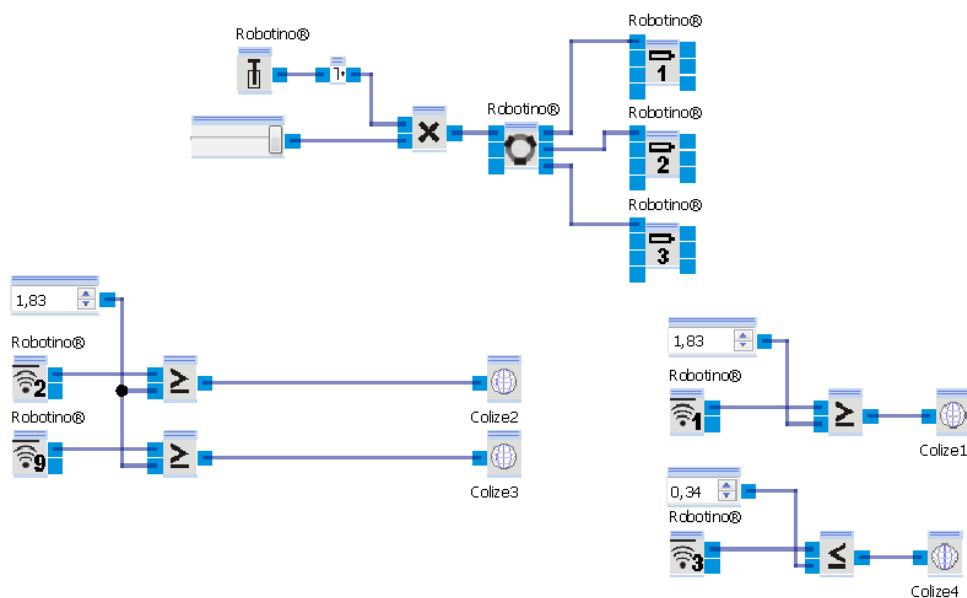
3.3.1 Popis programu

Jak se již psal, i v tomto programu musíme využívat skoků. V předchozím programu jsem již vysvětloval, jak se takové skoky vytváří, takže tuto část můžu přeskočit. Ale i přesto se ještě na skoky tohoto programu podíváme, protože zde stojí za zmínku, že v porovnání s předchozím programem, tady skoky vypadají jinak. Skoky jsou umístěny paralelně. Vše funguje úplně stejně, krom toho, že veškeré skoky probíhají již v prvním kroku a po skoku z následujícího kroku se vždy vracíme do prvního kroku. To vše se opakuje dokola, dokud program nevypneme.



Obr. 19: Main program - Paralelní zapojení

Když se podíváme blíže na jednotlivé kroky programu, hned si můžeme všimnout, že v první kroku jsou čtyři schémata, která jsou na sobě nezávislá. (Obr.20)



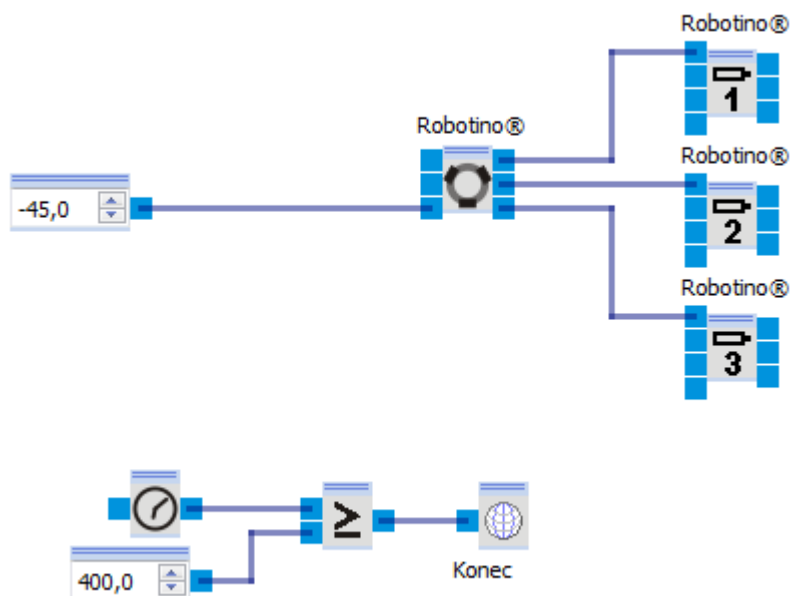
Obr. 20: První krok programu

První schéma zajišťuje pouze pohyb Robotina vpřed, což není nic, co by nebylo použito již v předchozích úlohách. V dalších schématech se objevují funkční bloky, které využívám poprvé. Jsou to funkční bloky "Distance". Těchto bloků nabízí Robotino® View hned devět, což je stejný počet, jako je na Robotinu umístěných infračervených snímačů, jak můžeme vidět na Obr.18. Jak jsem psal výše, snímače mají schopnost snímat oblast o velikosti 1 až 40 cm a pokud v této oblasti zaznamenají jakoukoliv překážku, indikují na výstupu hodnotu o její vzdálenosti. Hodnota je však pouze napětí a je ve voltech. Tato hodnota by nám sama o sobě moc neřekla, proto byla vytvořena tabulka, kde můžeme vidět převod hodnot napětí na cm. Tato tabulka je vidět na Obr.21. Zvolil jsem vzdálenost 6 cm, tedy velikost napětí 1,83V a pokud je hodnota na výstupu "Distance" větší nebo rovna této hodnotě, provede se skok na jiný krok, podle toho jaký senzor zaznamenal překážku dříve. U "Distance 3" je hodnota nastavena na 0,34 V. Tento senzor je využíván ve třetí a čtvrté zatáčce navrženého bludiště, protože na rozdíl od prvních dvou zatáček jsou tyto zatáčky levotočivé a Robotino se musí tedy točit opačným směrem.

Distance (cm)	Voltage (V)	Distance (cm)	Voltage (V)
1	1.9	21	0.57
2	2.54	22	0.55
3	2.54	23	0.53
4	2.51	24	0.5
5	2.13	25	0.48
6	1.83	26	0.46
7	1.61	27	0.45
8	1.42	28	0.44
9	1.28	29	0.42
10	1.19	30	0.41
11	1.08	31	0.39
12	1.01	32	0.38
13	0.92	33	0.36
14	0.87	34	0.64
15	0.81	35	0.34
16	0.75	36	0.32
17	0.71	37	0.31
18	0.67	38	0.3
19	0.63	39	0.28
20	0.61	40	0.28

Obr. 21: Tabulka pro převod voltů na cm [2]

V dalších krocích je pak již jenom změna pohybu Robotina, pomocí rotace kolem osy z a skok zpět na začátek programu, což můžeme vidět na Obr.22.



Obr. 22: Druhý krok programu - rotace kolem osy z

3.3.2 Spouštění programu

Při spouštění tohoto programu v Robotino® SIM, vzniká problém s přemísťováním překážek. Bohužel demo verze tuto možnost nenabízí a tak jsem si musel vystačit s překážkami tak, jak už byli v Robotino® SIM rozmístěné. Ale i tak si myslím, že funkčnost se zde dá výborně vyzkoušet, před ostrým zkoušením přímo na Robotinu.

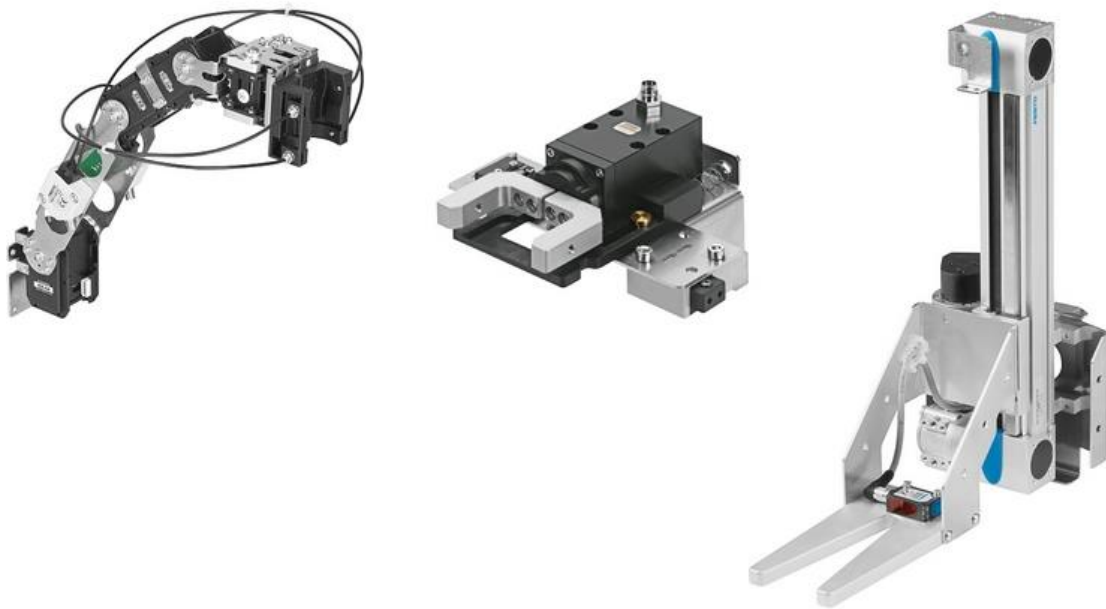
U toho programu bych upozornil více jak u kteréhokoliv jiného programu, na rychlost Robotina. Pokud byla nastavena rychlost Robotina vyšší jak 200 mm/s docházelo k tomu, že infračervené senzory nestíhaly vyhodnocovat vzdálenost překážky a docházelo k nárazům. Je tedy nutné, aby rychlost byla nastavena maximálně na 200 mm/s.

4 Zhodnocení stávající konfigurace Festo Robotino

Když se podíváme na stávající konfiguraci systému Festo Robotino, přijdeme na pár nedostatků, které nám brání ve vykonávání dalších zajímavých úloh.

4.1 Držák pro uchopení

Největší nevýhoda Robotina, bude zaručeně absence jakéhokoliv držáku pro uchopení. Máme hned dvě možnosti jak držák na uchopování získat. Jedna z možností je, vyrobit si přesně takový držák, který by se nám nejvíce hodil, což by mohlo být například část zadání pro diplomovou práci na Robotinu. Ale máme tu ještě druhou možnost a to zakoupení držáku na uchopení přímo u firmy Festo, která takových držáků nabízí hned několik.



Obr. 23: Držáky na uchopení od firmy Festo [15]

Pokud si podle specifikací vybereme držák, který nám nejvíce vyhovuje, stačí ho už jenom vložit do košíku a firma Festo vám ho zašle. Jediný problém je, že firma Festo nezveřejňuje cenu držáku. Pokud chcete cenu zjistit, musíte se na každý držák samostatně zeptat pomocí emailu, nebo telefonicky.

4.2 Některé další snímače

Stejně jako si můžeme koupit u firmy Festo držáky pro uchopení, můžeme si na internetových stránkách koupit i snímače, které nebyli v základním balíčku. Naše stávající konfigurace má většinu dostupných senzorů, ale přesto ne všechny.

Jeden ze dvou snímačů, které naší konfiguraci chybí je indukční analogový snímač. Tento snímač nahrazuje kameru v úloze "Sledování vodící čáry". V této úloze, by muselo samozřejmě dojít ještě k jedné změně a to k nahrazení černé pásky, páskou magnetickou.

Druhý ze dvou snímačů, které naší konfiguraci chybí je Laser scanner. Tento snímač umožňuje vytváření map, lokalizaci a navigaci. Funkci vytváření map, žádný jiný senzor dodávaný firmou Festo nenabízí a proto jeho absence v naší konfiguraci velký nedostatek.

4.3 Infračervené senzory

Poslední z problémů, na který jsem při práci s Robotinem narazil je špatné umístění infračervených senzorů. Tento nedostatek jsem již jedno v práci zmiňoval, ohledně toho, že je nutno při testování využít krabic jako překážek. Jelikož jsou infračervené snímače umístěny ve výšce 4 cm a snímají pouze vodorovně nalezené objekty, je problém například se stoly, nebo skříněmi, které jsou umístěny na nahou vyšších 4 cm, protože je potom infračervený senzor nezaznamená. Tento problém je bohužel řešitelný pouze pro firmu Festo a to tak, že u dalších modelů Robotina by umístili infračervené senzory níže.

Závěr

V této práci jsem vytvořil sadu tří základních úloh, na kterých si studenti mechatroniky naučí navigovat a programovat Robotino. Tyto úlohy zahrnují všechny základní senzory, které firma Festo k Robotinu dodává a studenti si tedy mají šanci vyzkoušet různé navigace. Během práce navíc vznikl nápad, vytvořit bludiště, kterým by muselo Robotino projet bez nabourání a my mohli na všechny tři druhy navigace porovnat na jedné a té samé cestě.

Program Robotino® View 2 je velice intuitivní a lehce pochopitelný program, za což bych rád firmě Festo poděkoval. Kdybych jí měl něco vytknout, tak by to bylo pár změn, které udělali v poslední verzi programu, protože zmizelo několik funkcí, které byly k výuce užitečné. Ale jak jsem již říkal, celý program je velmi dobře vytvořený a na výuku se mi zdá vhodný, jako studentovy, který se už s několika programy učil.

Druhý zmiňovaný program v práci byl program Robotino® SIM. Tento program není už tak povedený jako Robotino® View. Nevím jaký je program v plné verzi, protože tu si musíte u firmy Festo koupit. Ale práce s Robotino® SIM demo je trochu složitější. Největší problém je zde, že si nemůže posouvat překážky, ani měnit vodící čáru. Ale i přes tyto problémy je program dostačující k vyzkoušení programů, před ostrým puštěním na Robotinu.

Robotino samo osobě se vždy chovalo přesně tak jak mělo, jediným jeho problémem jsou baterie, které ho pohání. Tyto baterie, jak jsem se dočetl i v několika článcích odchází dost rychle. Slovem odchází, nemyslím zrovna, že by se vybíjeli rychle, ale to, že po několika použitích se už nedají nabít a musíte koupit nové baterie. Firma Festo samozřejmě na internetových stránkách nabízí nové baterie, ale s těmi to pravděpodobně dopadne úplně stejně a proto je vhodnější koupit jinou variantu baterií, které vydrží trochu víc, protože přeci jen ve výuce se budou baterie nabíjet docela často a nebylo by dobré, kdyby odešli v půlce roku a studenti nemohli dodělat své práce.

Úkolem této bakalářské práce bylo sestavit sadu výchozích projektů na mobilním robotovy od firmy Festo, pro studenty mechatroniky na Technické univerzitě v Liberci. Tento úkol se povedl bez větších problémů. Pokud se nějaké problémy vyskytly, tak to byl většinou problém se špatným pochopením manuálu, nebo nastavením v programu, za což samozřejmě firma Festo nemohla. Všechny problémy byly zdárně vyřešeny a práce mohla být dokončena.

Použitá literatura

- [1] ONDRÁČEK, Zdeněk a Jiří KAFKA. Mobilní robot pro výuku: Robotino. *ROBOT REVUE*. roč. 2010, č. 2. DOI: ISSN.
- [2] WEBER, Ralph-Christoph a Marcus BELLENBERG. *Festo*. 11/2009. Germany: Festo Didactic GmbH & Co., 2009, 49 s. ISBN 544305. Dostupné z: www.festo-didactic.com
- [3] LOCKER, Martin. Robotino. [online]. 2008 Dostupné z: <http://www.vosrk.cz/robotika/articles/robotino/cs>
- [4] Mecanum wheel. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-02-19]. Dostupné z: http://en.wikipedia.org/wiki/Mecanum_wheel
- [5] PYTLÍČEK, Tomáš. *Navigace mobilního robotu v úlohách hledání utajené cesty*. Praha, 2009. Bakalářská práce. České vysoké učení technické v Praze.
- [6] LACHNIT, Zdeněk. *INERCIÁLNÍ SNÍMACE PRO ZPRESNOVÁNÍ ODOMETRIE MOBILNÍCH ROBOTU*. Praha, 2007. Bakalářská práce. České vysoké učení technické v Praze.
- [7] WIKLER, Zbyněk. Odometrie. In: [online]. [cit. 2013-05-07]. Dostupné z: <http://robotika.cz/guide/odometry/en>
- [8] ŠOLC, František a Luděk ŽALUD. *ROBOTIKA*. Brno, 2002. České vysoké učení technické v Brně.
- [9] VOJÁČEK, Antonín. Integrované MEMS GYROSKOPY. In: [online]. [cit. 2013-05-07]. Dostupné z: <http://automatizace.hw.cz/integrované-mems-gyroskopy>
- [10] Fyzmatik. [online]. [cit. 2013-05-07]. Dostupné z: <http://fyzmatik.pise.cz/>
- [11] VOJÁČEK, Antonín. Principy akcelerometrů - 1. díl - Piezoelektrické. In: [online]. [cit. 2013-05-07]. Dostupné z: <http://automatizace.hw.cz/clanek/2007011401>
- [12] KUS, Martin. *GPS*. Jindřichův Hradec, 2007. Bakalářská práce. Vysoká škola ekonomická v Praze.
- [13] Kosmo.cz. [online]. [cit. 2013-05-07]. Dostupné z: <http://www.kosmo.cz/data/1025/gps.jpg>
- [14] WEBER, Ralph-Christoph a Markus BELLENBERG. Robotino®: Handbuch/Manual Northstar™ Sensor 2. Denkendorf,: Festo Didactic GmbH & Co. KG, 2009.
- [15] FESTO. [online]. [cit. 2013-05-07]. Dostupné z: <http://www.festo-didactic.com>

[16] KURUC, Jíří. Pořádek ve zkratkách: co znamená A-GPS. [online]. [cit. 2013-05-10]. Dostupné z: <http://navigovat.mobilmania.cz/clanky/poradek-ve-zkratkach-co-znamená-a-gps/sc-3-a-1314495>