



Střední průmyslová škola technická

Jablonec nad Nisou, Belgická 4852, příspěvková organizace



Středoškolská technika 2016

**Setkání a prezentace prací středoškolských studentů
na ČVUT**

VYUŽITÍ SENZORŮ ROBOTINA®

Daniel Drobník

Střední průmyslová škola technická

Belgická 4852, Jablonec nad Nisou

PŠT

Anotace

Tato práce se zabývá jak návrhem, tak virtuální simulací a samotnou realizací souboru úloh využívajících senzory robotu, za použití programovacího softwaru.

Klíčová slova: robot, robotino, programování, senzor, virtualizace

Annotation

This work is about designing, virtual simulation and the implementation itself of a set of tasks utilizing the robot sensors, using the programming software.

Keywords: robot, robotino, programming, sensor, virtualization

Obsah

Obsah.....	3
Použité značky a symboly.....	7
1. Úvod.....	7
2. Teoretická část.....	10
2.1. Modularita.....	10
2.2. Pohon.....	10
2.2.1. Konstrukce pohonného modulu	Chyba! Záložka není definována.
2.3. Řídicí jednotka	11, 12
2.4. Rozšiřující deska	13
2.5. Senzorika.....	14
2.5.1. Infračervené triangulační dálkoměry	14
2.5.2. Optické inkrementální snímače úhlu natočení	15
2.5.3. Optické limitní snímače a indukční analogový snímač	16
2.5.4. Ostatní senzory.....	17
2.6. Navigace v prostoru	17
3. Praktická část.....	Chyba! Záložka není definována.
3.1. Využitý software	Chyba! Záložka není definována.
3.1.1 Robotino® View 3	18
3.1.2 Robotino® SIM Demo	18
3.2. Základy programování Robotina.....	Chyba! Záložka není definována.
3.2.1 Práce s programovacím prostředím.....	19
3.2.2 Simulace.....	25
3.3. První úloha - "Bezpečná jízda po pruhu"	26
3.3.1 Cíl úlohy.....	26
3.3.2 Použité programovací bloky	26

3.3.3 Ukázka programu	27
3.4. Druhá úloha - "Kopírování zdi"	30
3.4.1 Cíl úlohy.....	30
3.4.2 Použité programovací bloky	30
3.4.3 Ukázka programu	30
3.5. Třetí úloha - "Provokace"	32
3.5.1 Cíl úlohy.....	32
3.5.2 Použité programovací bloky	32
3.5.3 Ukázka programu	32
4. Závěr	35
5. Seznam použité literatury a zdrojů informací	36
5.1. Zdroje informací.....	36
5.2. Literatura	36
6. Seznam použitého software	36
7. Seznam příloh	37
8. Licenční ujednání.....	37

Použité značky a symboly

F – síla,	F	[N]
I – elektrický proud,	I	[A]
m – hmotnost,	m	[kg]
M – moment síly,	M	[Nm]
P – výkon,	P	[W]
R – elektrický odpor,	R	[Ω]
U – elektrické napětí,	U	[V]
ρ – rezistivita,	ρ	[Ωm]
ω – úhlová rychlost,	ω	[$\Omega\text{ s}^{-1}$]
atd.	a tak dále	
mj.	mimo jiné	
tj.	to jest	
IR	infra-red = infračervené	
CPU	central processing unit – procesor	
®	označení registrace	
™	trademark – označení obchodní značky	

1. Úvod

Tato maturitní práce by měla být nápomocná z informativního hlediska případným zájemcům o problematiku robotiky na studijní či laické úrovni, a to jak svou teoretickou částí, tak i tou aplikovanou. FESTO™ Robotino®, mobilní robotický systém německé nadnárodní společnosti (dále jen „robotino“), je díky své modularitě a zvoleným komponentám, převážně veřejně dostupným v sortimentu prodejen elektrotechnických součástí, určený především pro studijní účely, pro které byl vybrán již mnoha středními i vysokými školami k profesionálnější výuce oborů zabývajících se robotikou či mechatronikou.

Společnost FESTO™ byla založena roku 1925 Albertem Fezerem s Gottliebem Stollem. Původně šlo o společnost působící v dřevozpracujícím průmyslu, konkrétně se zabývala výrobou dřevozpracujících nástrojů používaných tehdejšími obyvateli i továrnami. Prvním významnějším produktem společnosti byla světově první přenosná, benzinem poháněná motorová pila. Později firma expandovala svým polem působnosti také do oblasti automatizace.

Automatizace označuje používání řídicích systémů pro řízení technologických procesů a zařízení. Cílem automatizace je snížit potřebu přítomnosti člověka při vykonávání určité, zpravidla repetitivní činnosti vyžadující stabilní dlouhodobou přesnost. Moderní pojetí automatizace upřednostňuje pohled na automatizaci procesu výroby nikoliv jako inovaci za účelem plného nahrazení lidského faktoru (což je v praxi neuskutečnitelné), nýbrž za účelem koordinované kooperace člověk – robot, díky které jsou stavěni do role pracovních partnerů.

Roku 2000 byla firma rozdělena, celý segment zabývající se zmíněnými dřevozpracujícími nástroji se přesunul do nové společnosti Festool. Co se týče dnešní doby, ve FESTU pracuje přes 15 500 zaměstnanců, za rok 2013 činil obrát 2,9 miliard eur a výkonným ředitelem společnosti (CEO, Chief Executive Officer) je Eberhard Veit.

Aktuálně má tato společnost v nabídce přes 28 000 katalogových výrobků, nicméně často se zabývá také specifickými požadavky individuálních klientů. Působíště a činnost firmy se v dnešní době dá rozdělit do tří kategorií: 1) **Automatizace výroby**: Společnost sleduje světové trendy a na základě toho vyvíjí nové výrobky. Ty mají především za cíl úsporu energie a ohleduplnost k životnímu prostředí. Každoročně tak

vzniká katalog nového sortimentu firmy a k němu příslušný software, elektronické příručky a další online nástroje. 2) **Automatizace procesů**: Firma poskytuje specifické koncepce pro konkrétní výrobu, logistiku, manipulaci a manipulaci s tekutinami, plyny a materiály o konzistenci pasty či pevnými látkami. Jde také o koncepce určené do extrémních podmínek – velmi vysoké / nízké teploty, drsné a korozivní prostředí. Velmi významným odběratelem produktů FESTO™ jsou vedoucí řetězce v technologickém průmyslu, které si od FESTA™ pravidelně objednávají jak jednotlivé součásti, tak i kompletní systémy. Jde o technologické průmysly zabývající se petrolchemikou, výrobou papíru, těžbou a zpracováním ropy, plynárenství, výrobou elektrické energie, získávání kovů a hornictví. 3) **Školení a vzdělávání** FESTO™ poskytuje konzultace a vzdělávací prostředí ve 26 světových jazycích. Takzvané vzdělávání FESTO™ Didactic® je firmou poskytováno po celém světě ve specifických školicích zařízeních a účastní se jich cca 42 000 lidí ročně. Zaměřuje se na průmyslovou a procesní automatizaci => zefektivnění procesů v údržbě, výrobě a samotné konstrukci.

2. Teoretická část

2. 1. Modularita

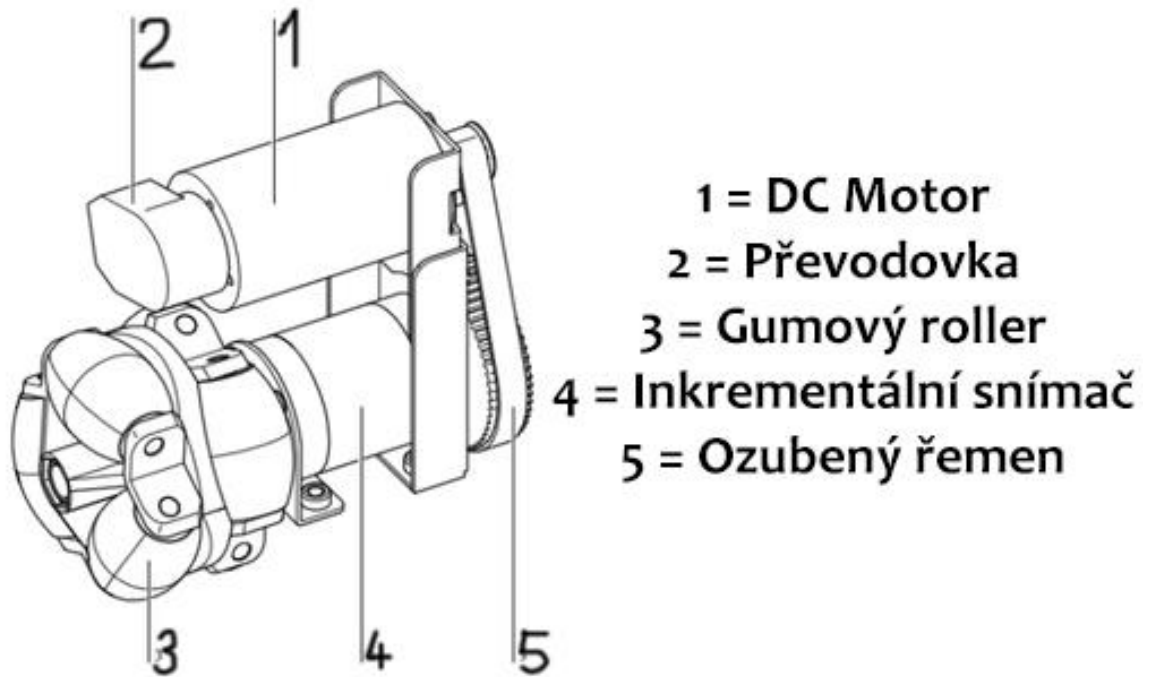
FESTO™ Robotino® je tedy komplexně řešený mobilní robotický systém založený na principu modularity a v rámci možností jednoduché manipulace v případech potřeby jak k údržbě z důvodů potencionální mechanické únavy, tak k modifikaci určitých částí daného robotu jakožto fungujícího celku v rámci zlepšení specifických funkcí individuálně vyžadovaných daným uživatelem. Díky těmto charakteristickým vlastnostem je tento robot v rámci svého vybavení značně variabilní a umožňuje studentům využívat této skutečnosti k výukovým účelům. Společnost zprostředkovává klientům přímý způsob využití této modularity v rámci možnosti individuálního výběru dvou verzí výbavy tohoto robota, lišících se ve výpočetním výkonu procesorů, kapacitě interní paměti určené k archivaci případných programů pro robot a fyzicky „hmatatelným“ aspektem, montážní věží určenou pro instalaci dalších požadovaných částí, jež je k dispozici pochopitelně pouze v té vybavenější ze dvou variant. Jde o verze Basic Edition a Premium Edition, lišících se samozřejmě značně i v ceně.

2. 2. Pohon

Robot je vybaven tzv. všesměrovým pohonem, jakožto jeho pohonnou a pohybovou jednotkou, která mu zajišťuje plynulý pohyb ve všech směrech. Je vybaven třemi gumové rollery pohánějícími na sobě nezávislými pohonnými moduly, díky čemuž se může pohybovat směrem dopředu, dozadu, do obou stran a také rotovat na místě. Tyto moduly jsou umístěny v přesném 120° širokém úhlu. Smysl a rychlost separátních motorů má na starost výpočetní jednotka robotu, která vše složitými matematickými výpočty v reálném čase vyhodnocuje.

2.2.1. Konstrukce pohonného modulu

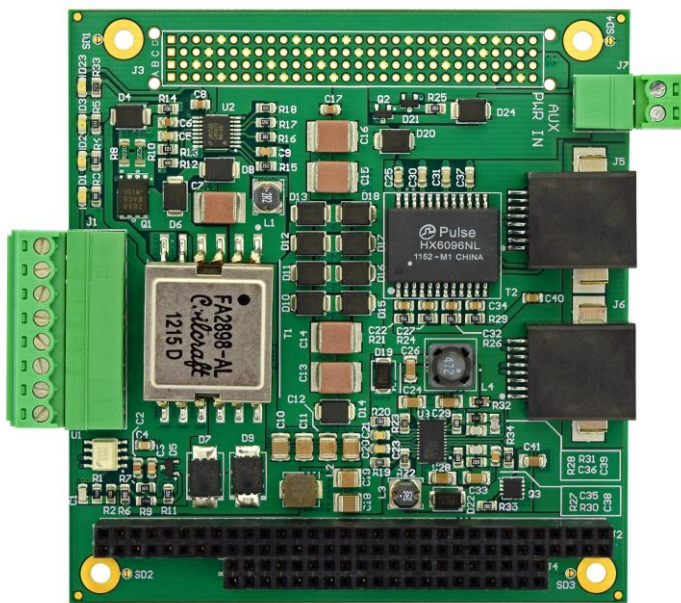
Každý pohonný modul je vybaven jedním stejnosměrným motorem, převodovkou s převodovým stupněm 16:1, ozubeným řemenem, inkrementálním snímačem a samotným gumovým rollerem.



2.3 Řídicí jednotka

Základem řídicího systému tohoto robotu je průmyslový počítač typu PC104, běžícím na speciálně upraveném operačním systému Linux. Jeho mozkem je, za předpokladu verze Basic edition, procesor Intel Atom pracující na frekvenci 1,8 GHz doplněný o operační paměť RAM s kapacitou 4 GB. V případě dražší varianty robotu Premium Edition je dodáván s CPU Intel Core i5 schopného práce na 2,4 GHz a operační paměti RAM 8 GB. Veškerá sensorika robota včetně pohonů je připojena na jednočipový (32bitový) modul microcontrolleru AVR ATmega - Atmel. Robot je osazen nárazníkem vybaveným snímačem pro detekci kolizí, devíti IR triangulačními senzory Sharp

s pracovním rozsahem 4 až 30cm, dva limitní optické snímače a jeden analogový indukční snímač polohy. Přímou k řídicímu počítači je také připojena USB kamera pro získání a práci s vizuální informací. Robot jako takový může být ovládán jak bezdrátově využitím své vlastní Wi-Fi sítě z počítače či jiného média vybaveného Wi-Fi konektivitou, či interně zasunutím USB paměti obsahující hotový program pro robota. Ty, včetně operačního systému robota, robot uchovává ve svém SSD (solid-state drive) disku o kapacitě 32 GB / 64 GB, samozřejmě v závislosti na variantě zakoupeného modelu robota.

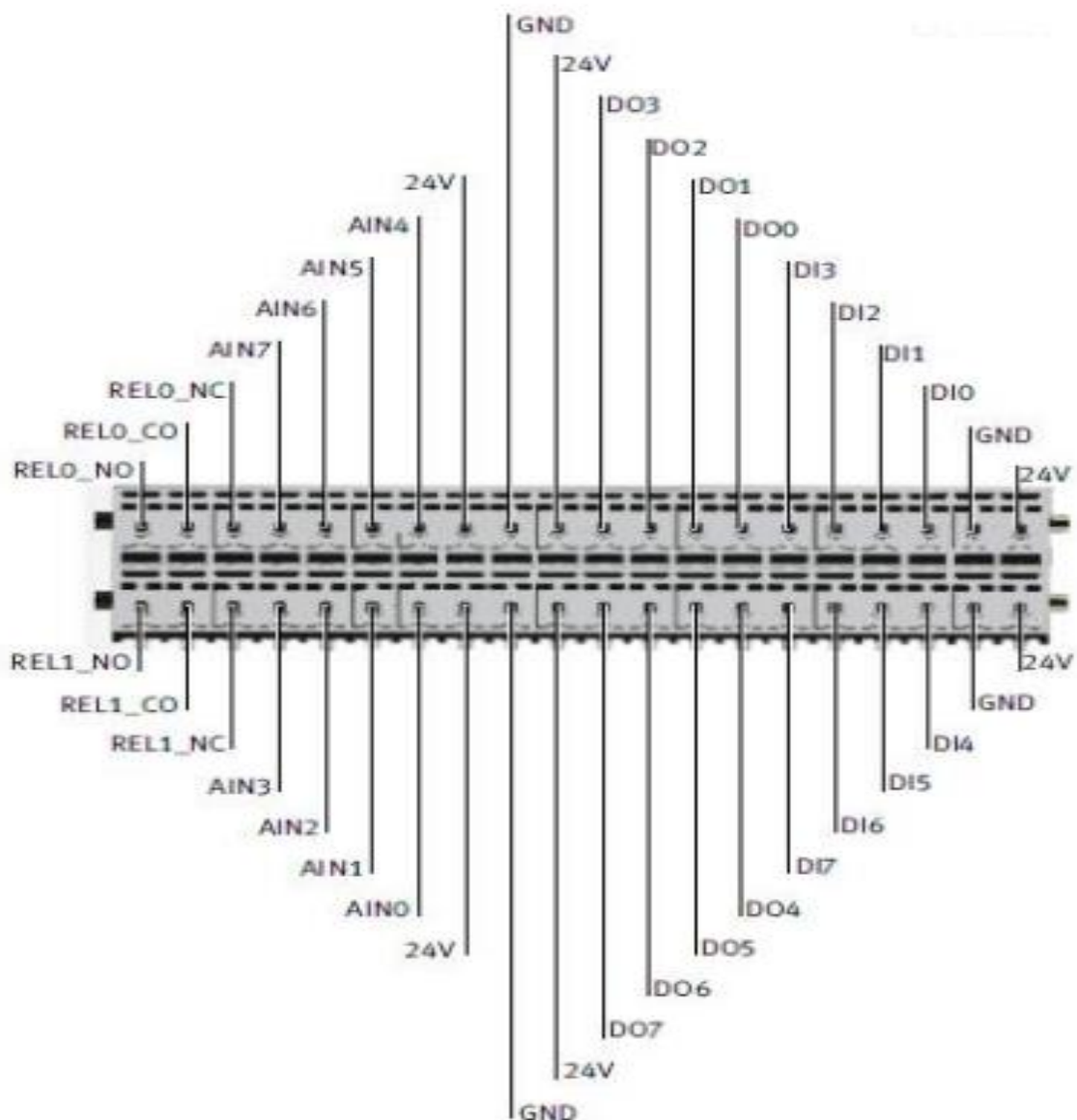


Ukázka jednodeskového průmyslového počítače typu PC104

2.4 Rozšiřující deska

Robot je vybaven velkým množstvím různých výstupů a vstupů. Navíc je dodáván i se speciální rozšiřující deskou, jež je připojená k řídicí elektronice a umožňuje uživateli připojovat modulární senzory na specifické analogové či digitální vstupy.

Díky této desce můžeme použít několik dalších akčních členů: 8 analogových vstupů (0 až 10V, značené AIN0 až AIN7); 8 digitálních vstupů (značených D10 až D17); 8 digitálních výstupů (značených DO0 až DO7); a dvě relé.



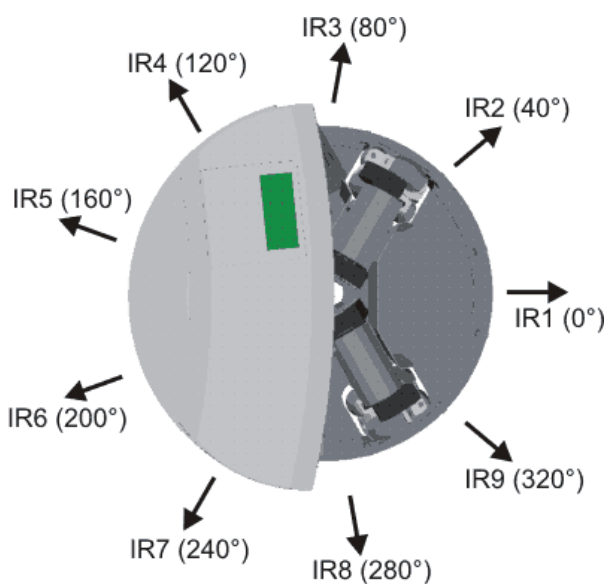
Schematické znázornění situace – Rozšiřující deska

2.5 Senzorika

Robotino® je robot schopný v případě správného naprogramování využívat celou řadu senzorů, kterými je osazen.

2.5.1 IR triangulační dálkoměry

Robotino® je vybaven devíti IR senzory typu SHARP GD2D120 , které slouží k měření vzdálenosti. Princip senzoru je takový, že vysílač vyzařuje paprsek směrem k objektu, od kterého se odráží zpět do přijímače. Jsou namontovány na podvozku pod úhlem 40°. Prostřednictvím těchto snímačů je možné snímat vše kolem robota na vzdálenost 4 až 30 cm. V programovacím prostředí, oficiálně vytvořeném a předurčeném k práci s tímto robotem, Robotino® View 3, je funkční blok „Distance“, ve kterém lze tuto vzdálenost nastavit. Každý ze snímačů může být dotazován individuálně. Díky tomu si Robotino® dokáže udržovat odstup od různých překážek a předejít nárazu.



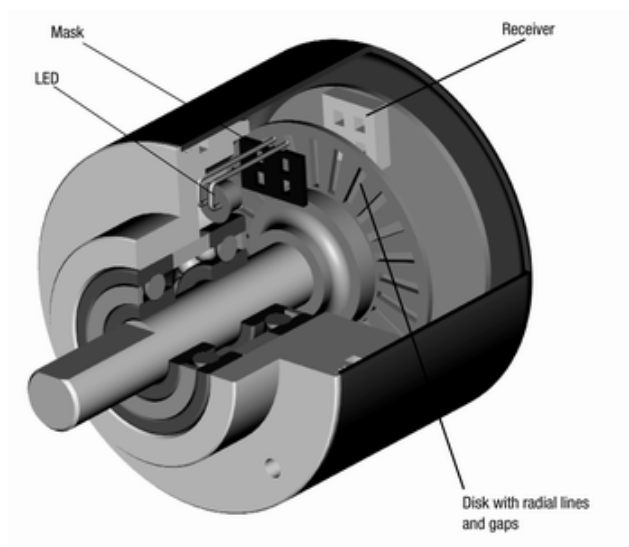
2.5.2 Optické inkrementální snímače úhlu natočení

Jejich základem je takzvaný pulzní disk. Ten je převážně skleněný, vyrábí se ale s různých materiálů. Ten je mechanicky upevněn s hřídelí procházející osou snímače.

Disk obsahuje světlá a tmavá pole. Světlo generované diodami, zpravidla pracující v IR části spektra, prochází membránou a tímto pulzním diskem. Poté je zachyceno fotodetektorem umístěným na druhé straně. Při otáčení hřídele pak disk střídavě světlo propouští a nepropouští – (tzv. cloní fotodetektor). Tyto světelné pulzy jsou dále zpracovány elektronikou snímače a převedeny na výstupní signál zpravidla obdélníkového či harmonického typu.

Počet tmavých – neprůhledných a světlých – průhledných polí odpovídá počtu pulzů na jednu otáčku. To je v podstatě jeden z nejdůležitějších parametrů týkajících se každého snímače, jenž je inkrementální.

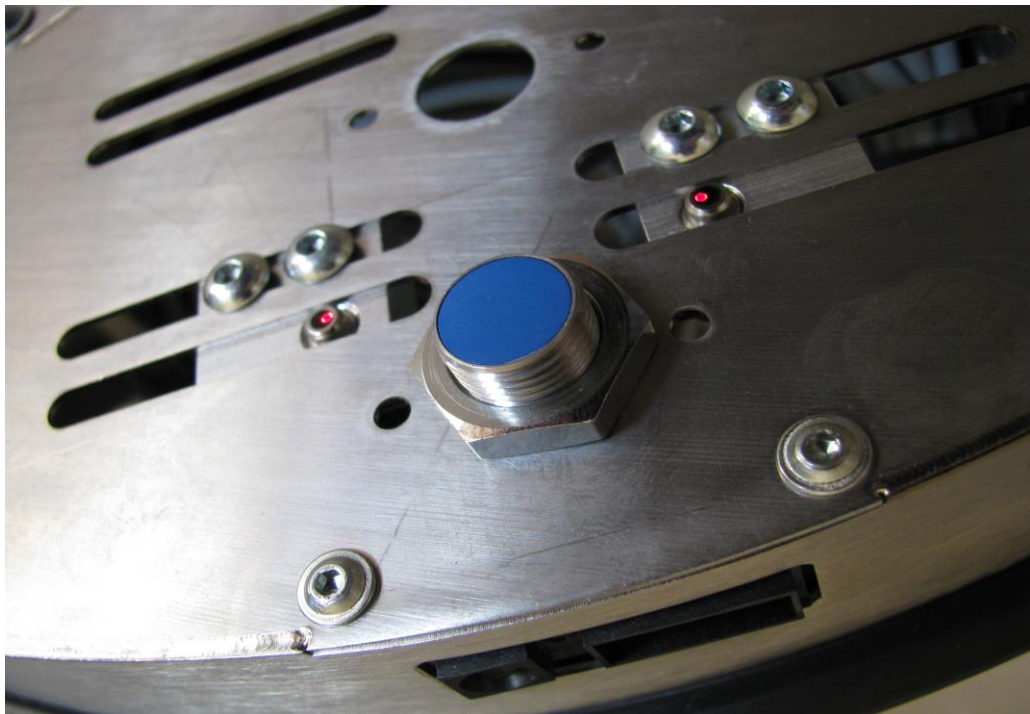
Dnes se vyrábí jakožto zaběhnutý standard takzvané snímače tříkanálové (dřív šlo o typy mající pouze jeden kanál, tedy jednu stopu pulzního disku). Signály dvou kanálů jsou vzájemně posunuty o 90° , což umožňuje identifikaci jejich směru otáčení. Třetí kanál generuje pulz jednou za otáčku a zpravidla se nazývá tzv. „nulovým pulzem“. Z principu činnosti si inkrementální snímače nepamatují narozdíl od snímačů absolutních polohu po vypnutí napájení.



2.5.3 Optické limitní snímače a indukční analogový snímač

Indukčnost vyjadřuje velikost magnetického indukčního toku kolem cívky při jednotkovém elektrickém proudu, jehož velikost je jeden ampér, procházejícím cívkou. Indukčnost cívek závisí na jejich počtu závitů, rozměrech a tvarech, na magnetických vlastnostech specifického prostředí, které cívky obklopuje a na magnetických vlastnostech které cívky obklopují.

Rozlišujeme indukčnost jedné cívky a indukčnosti vlastní, indukčnost dvou na sebe působících cívek nazýváme vzájemnou indukčností. **Indukční snímač, jímž je vybaven robot, detekuje kovové předměty na podlaze.**



Technické specifikace snímače:

Typ: SIEA-M12B-UI-S; M12

Provozní napětí: 15 až 30 V, DC

Výstupní napětí: 0 až 10 V

Rozsah snímání: 0 až 6 mm

Frekvence: 1000 Hz

Provozní teploty: -25°C až +70°C

2.5.4. Ostatní senzory

Robot je osazen i dalšími senzory, jež mu umožňují monitorovat různorodé veličiny, jak informativní, tak bezpečnostní. Jako příklad lze uvést senzor napětí baterie a kvality signálu Wi-Fi. Svou funkční podstatou může být do sensoriky zařazena i kamera, jelikož je využívána pro různé prostorové detekční systémy.

2.6. Navigace v prostoru

Automatická navigace je jednou z dominant problematiky zabývající se komplexním tématem řešícím skutečnost snahy o dosažení speciální takzvané umělé inteligence u specifických autonomních robotických technologických systémů jako takových. Jednou z metod využívaných k řešení takové problematiky jakožto celku je odborně pojmenovaná metoda *Dead Reckoning*. Tato metoda je česky definovaná jako „Matematická procedura pro určování současné pozice objektu pomocí postupného přičítání dráhy díky známému kurzu a rychlosti v čase.“. Nejjednodušší implementace takovéto metody do použití v praxi se nazývá *odometrie*.

Odometrie se zabývá odhadem aktuální pozice pohybujícího se objektu vůči počáteční poloze, ve kterém se nacházel před začátkem takového pohybu, a k stanovení vzdálenosti, kterou objekt urazil od začátku pohybu.

Používá se zejména u mobilních robotických systémů s podvozkem vybaveným koly. Aplikace bývá provedena obvykle pomocí výše zmíněných inkrementálních snímačů. Základním předpokladem zůstává fakt, že otáčky kol mohou být převedeny v lineární posuv vůči podlaze. To nicméně vede také k akumulaci chyb. Odometrie umožňuje spolu s měřením pohybu získat lepší odhad pozice. V některých případech, když nejsou k dispozici externí reference, může být odometrie jedinou možností navigace. Nepřesnost odometrie způsobují následovné chyby: Rozdílné průměry kol, Výchylka kol, jízda po nerovném povrchu, jízda přes neočekávané překážky, prokluz kol.

Navigační systém *NorthStar* je doplňkovým modulem využívajícím optické lokalizace robota při pohybu v místnosti. Jedním z hlavních problémů v oblasti mobilních robotů je právě ona navigace, tedy lokalizace a hledání správné cesty k cílovému bodu – takzvaný *pathfinding*. Nejjednodušší metodou lokalizace je právě ona *odometrie*, jež je však poměrně nepřesná. Tím pádem je vhodné použít další externí snímače.

3. Praktická část

3.1. Využitý software

3.1.1. Robotino® View 3

Robotino® View 3 je oficiálním bezplatně dostupným programovacím softwarem originálně poskytovaným firmou FESTO™ pro programovací účely produktů Robotino®. Jedná se o sofistikovanou aplikaci intuitivně navrženou tak, aby byla po prvním detailnějším seznámení se a porozumění toho, jak funguje, uživatelsky přívětivá. Tento software je dostupný ke stažení na oficiálních internetových stránkách firmy FESTO™. Tento software, vzhledem k faktu, že může být poměrně náročný na matematické i fyzikální výpočty, vyžaduje relativně výkonnější počítač vybavený kvalitním procesorem.

3.1.2. Robotino® SIM Demo

Robotino® SIM Demo je oficiální bezplatně dostupnou demoverzí simulačního softwaru originálně poskytovaného firmou FESTO™ pro účely simulace hotového programu z programovacího softwaru, Robotino® View 3. Jedná se o propracovaný simulační software, který i ve volně dostupné DEMOverzi plně postačuje k virtualizaci předdefinovaných úloh – programů, a to v přehledném 3D virtuálním prostoru obsahujícím všechny důležité prvky – mechanické překážky (zeď tvořená naskládanými krychlemi, kvádry, válce atd), zdi, pruhy na zemi a další. Veškeré z výše uvedených překážkových prvků podléhají základním fyzikálním zákonům, a to využitím osvědčené prestižní počítačové technologie NVIDIA® PhysX®. Díky tomuto faktu můžeme v simulaci počítat s tím, že pokud ve virtuálním prostoru robot narazí do nějakého z překážkových prvků – například kvádry – tak ten daný prvek bude odražen, posunut či sražen k zemi. Placená verze programu, Robotino® SIM Professional, nabízí kromě těchto funkcí také možnost práce s více roboty najednou a další prémiové prvky pro usnadnění či profesionalizaci a zlepšení preciznosti samotného simulačního procesu. Tento software, vzhledem k faktu, že může být poměrně náročný na fyzikální i vizualizační výpočty, vyžaduje relativně výkonnější počítač vybavený kvalitní grafickou kartou a kvalitním procesorem.

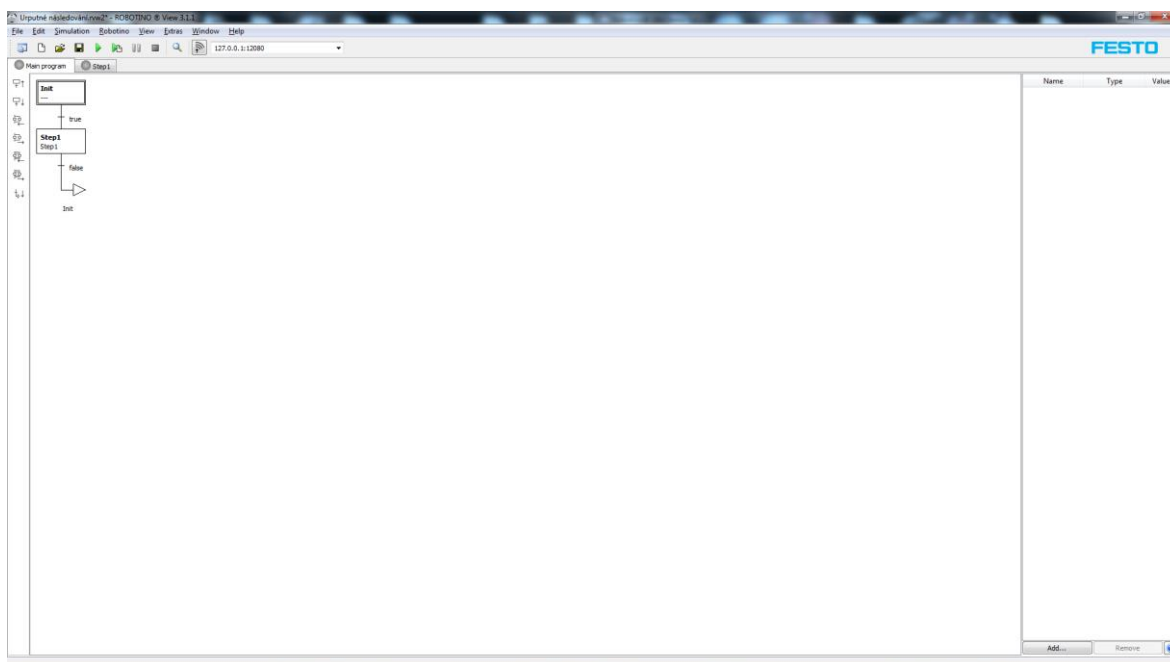
3.2. Základy programování Robotina

Programování Robotina® jako takového může být realizováno několika způsoby, z nichž pravděpodobně nejefektivnějším je právě možnost využití programovacího prostředí softwaru Robotino® View 3. Existují nicméně také jiné způsoby, například přímé psaní programového kódu v programovacím jazyku C++, v němž daný robot funguje implicitně => Robotino® View 3 v podstatě svým principem fungování dělá práci za nás, jelikož z našich příkazů (skládání obvodů a programování v něm) sám automaticky píše kód, kterému Robotino® rozumí a je schopné jej vyhodnotit a exektovat.

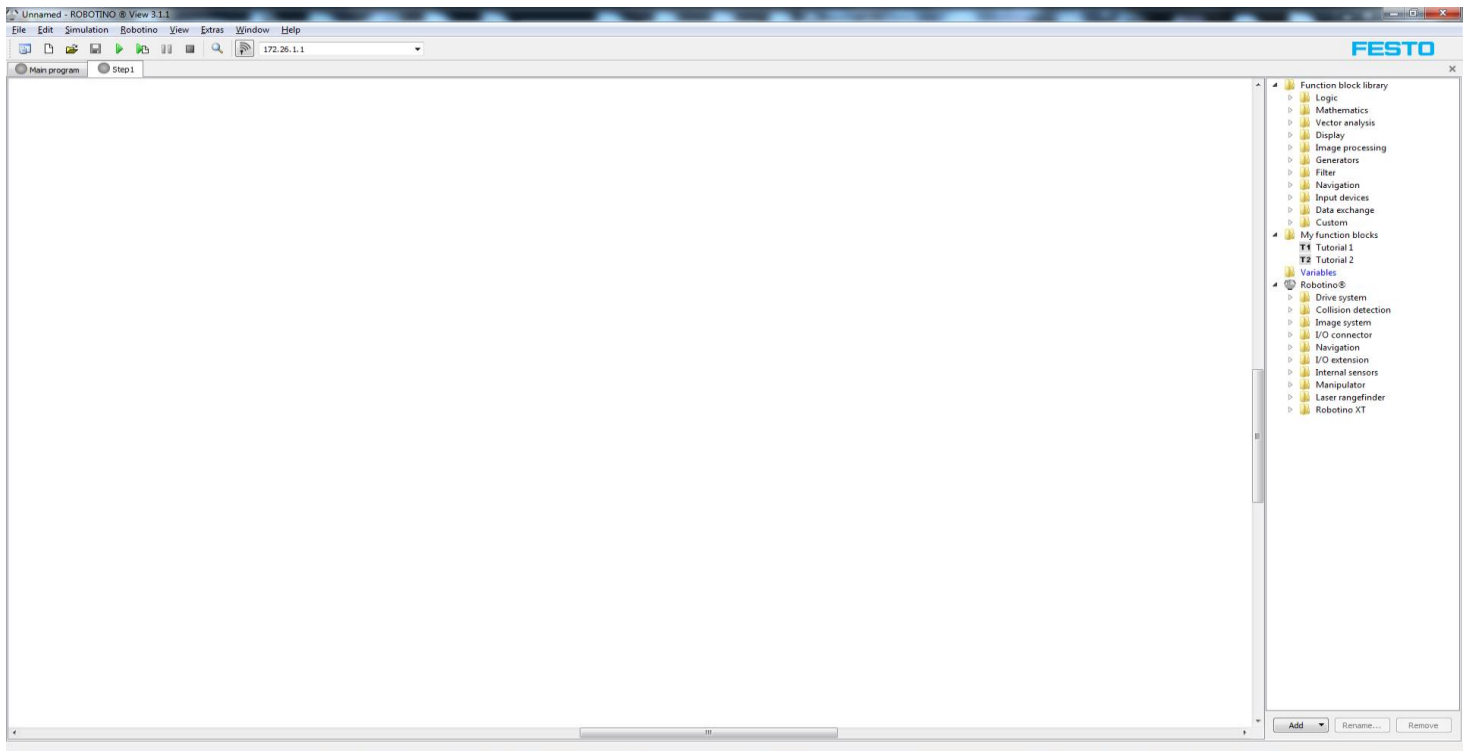
3.2.1. Práce s programovacím prostředím

V této sekci si popíšeme základy práce s programovacím softwarem použitým v této práci. Pro lepší přehlednost, vzhledem k faktu, že v následujících sekcích budou samotné úlohy konkretizovaně popisované i s funkčností jednotlivých segmentů programů, bude stručný popis předcházen užitečným snímkem obrazovky pro danou situaci popisovanou popiskem.

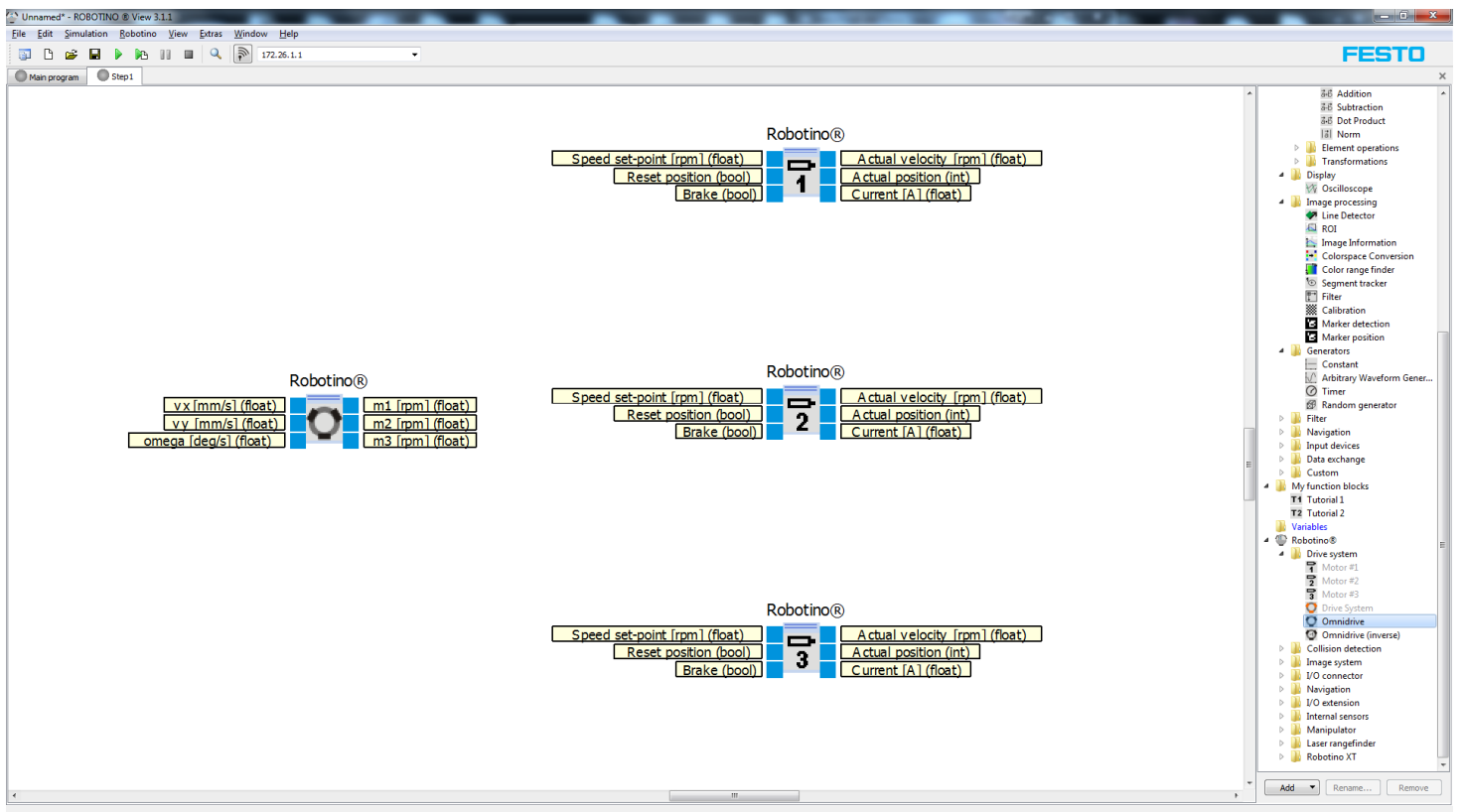
První pohled na program – Strom algoritmu



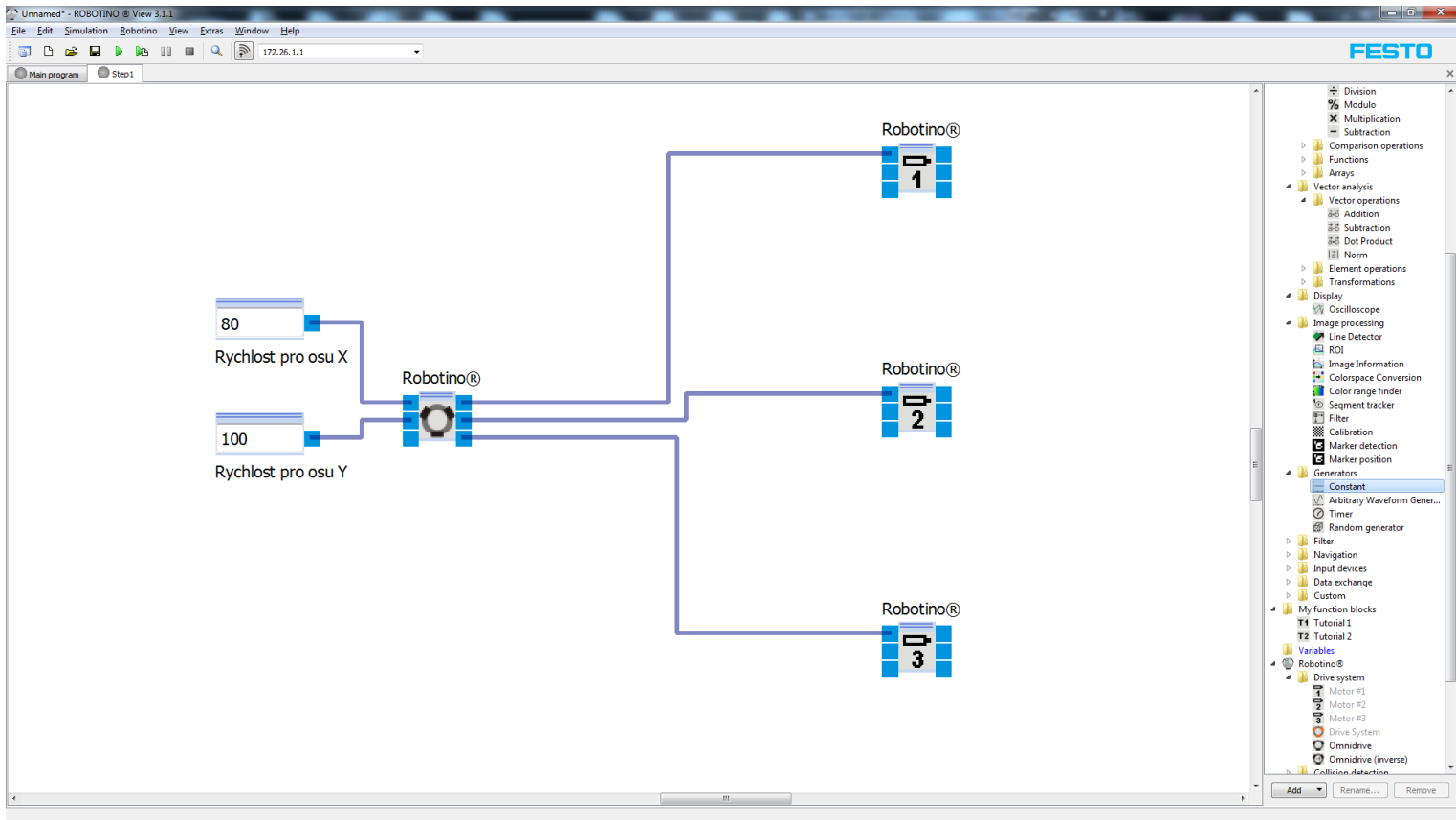
Okno pro práci s funkčními bloky a potažmo tvorbu samotného programu



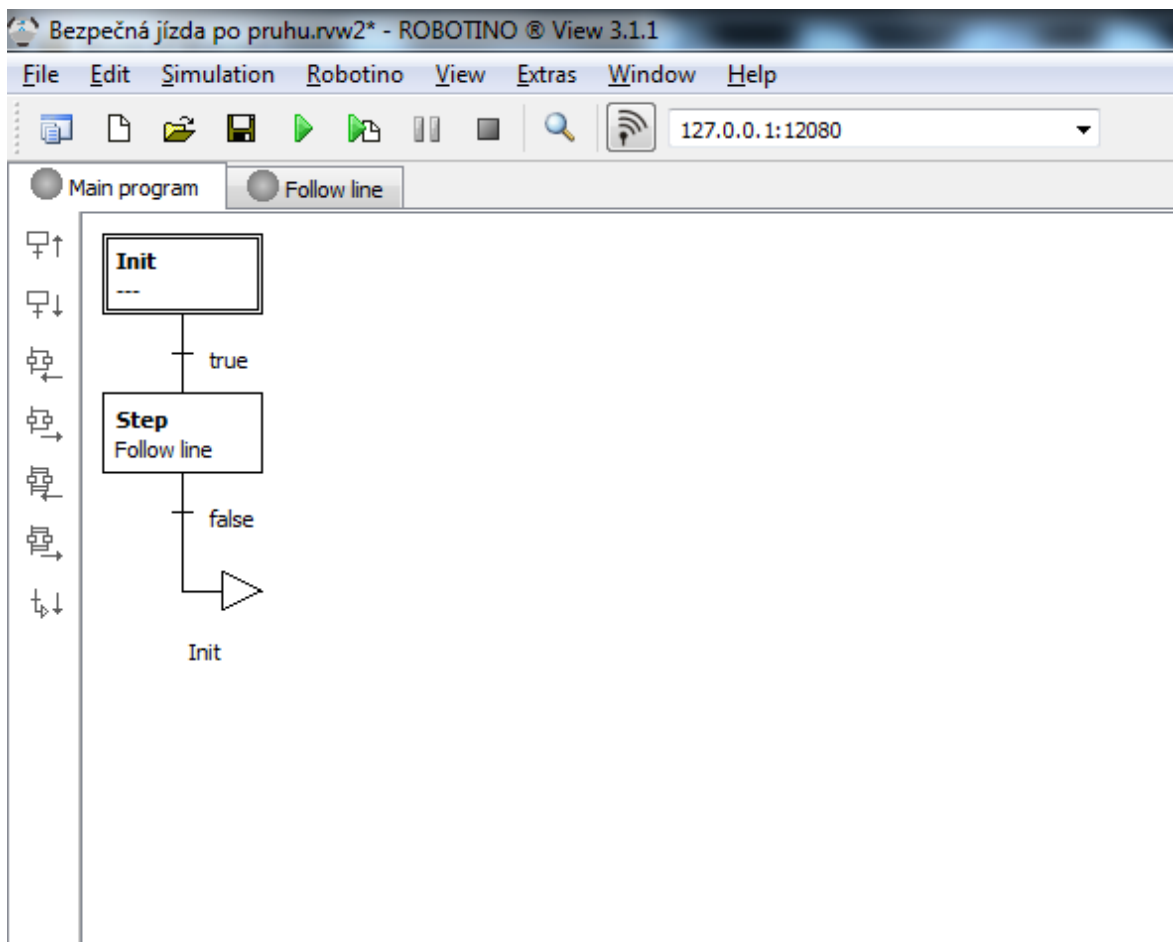
Základní výkonové bloky



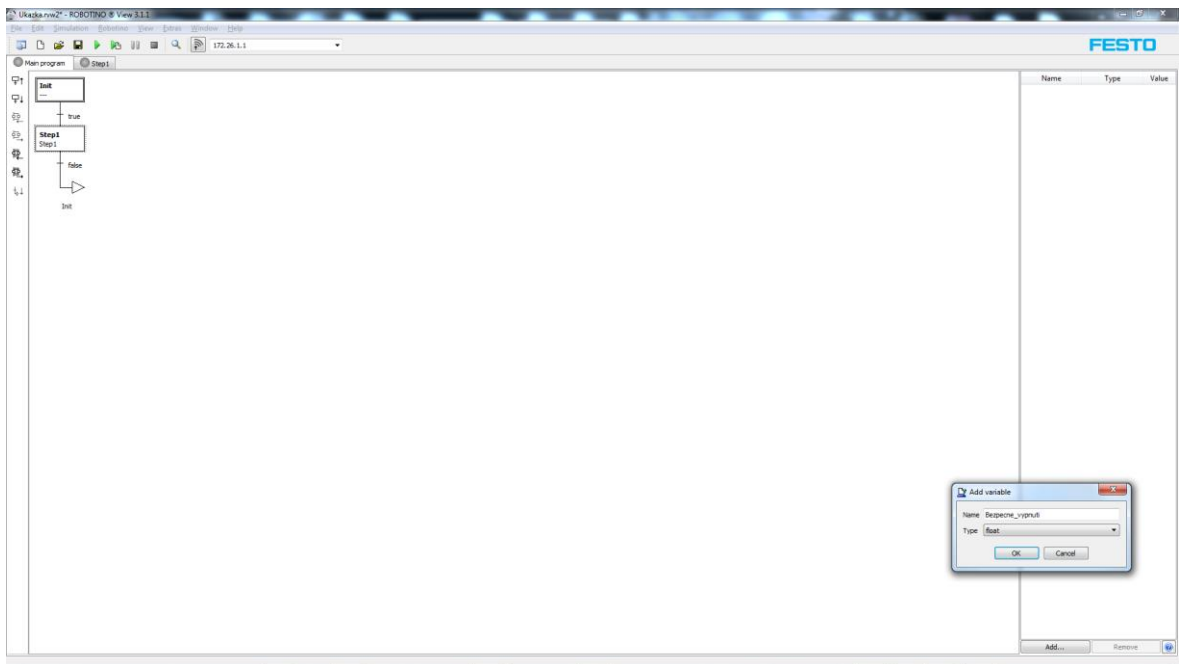
Na tomto snímku obrazovky vidíme základní zapojení schématu pro nejprostší rozběhnutí motoru pomocí dvou konstant přivedených na systém OmniDrive za účelem pohybu v ose X a v ose Y.



Na tomto snímku vidíme základní algoritmický strom. Ten je složen z několika bloků, mezi nimiž jsou pravdivostní (podmínkové) mezikroky. Prvním z bloků je blok Init, tedy Inicializace – Start celého obvodu. Pod ním vidíme blok jménem Step, s dodatkovým popiskem hlásajícím „Follow line“. Tento blok zastupuje exekuci samotného programu, jež byl naprogramován. Mezi každým z kroků (bloků) algoritmu je tzv. podmínkový stupeň, jenž má buď hodnotu „true“ nebo „false“. Může mít také podobu určité podmínky, jejíž splnění zajistí výsledek „true“ a nesplnění výsledek „false“. Pokud je vyhodnocení kladné, dojde k postoupení programu do dalšího bloku algoritmu.



Ve stejném okně máme také seznam proměnných – hodnot, jež můžeme libovolně vytvářet. Jejich vytvořením vytváříme také takzvaný blok „Variable“, tedy blok Proměnná. Ten můžeme využít buď pro zápis, či čtení hodnoty v něm obsažené. Z toho vyplývá, že nejdřív musíme pro každou proměnnou deklarovat její číselnou hodnotu (zpravidla blokem generátoru konstanty) a následně ji můžeme číst, tedy používat (například můžeme udat rychlost systému OmniDrive pomocí proměnné místo pevného „hrubého“ vepsání hodnoty konstantou).

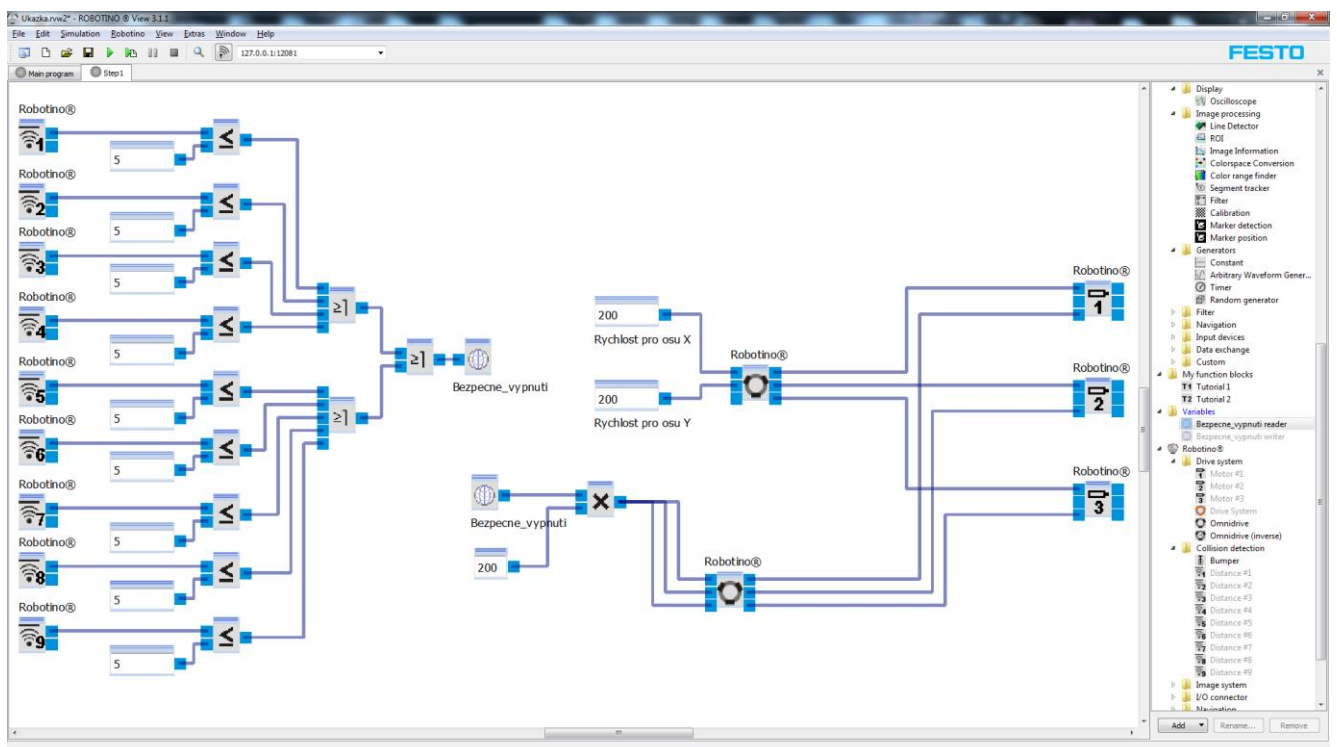


Na tomto snímku můžeme vidět již hotový pokročilejší program. Zajišťuje nám, že pokud bude robot ve vzdálenosti menší než 5cm od nějaké překážky, dojde k aktivaci proměnné Bezpecne_vypnuti a robot dostane signál pro zbrždění motorů, tedy zastavení.

Jsou zde využity matematické komparativní funkční bloky „menší nebo rovno“ v kombinaci s logickými funkčními bloky „OR“.

Funguje to tak, že pokud jakýkoli ze senzorů vzdálenosti vykáže hodnotu menší nebo rovnou 5ti, dojde ke splnění jedné z podmínek hradel OR, čímž se jejich výstup změní na hodnotu „1“ z hodnoty „0“. Tím pádem dojde k zápisu do proměnné Bezpecne_vypnuti.

Následně použijeme blok pro čtení proměnné Bezpecne_vypnuti vynásobený konstantou 200. To z toho důvodu, že pokud je proměnná rovna nule, nic se nestane, nicméně pokud bude rovna 1, robot bude okamžitě brzděn hodnotou 200 vyslanou do brzdícího vstupu.

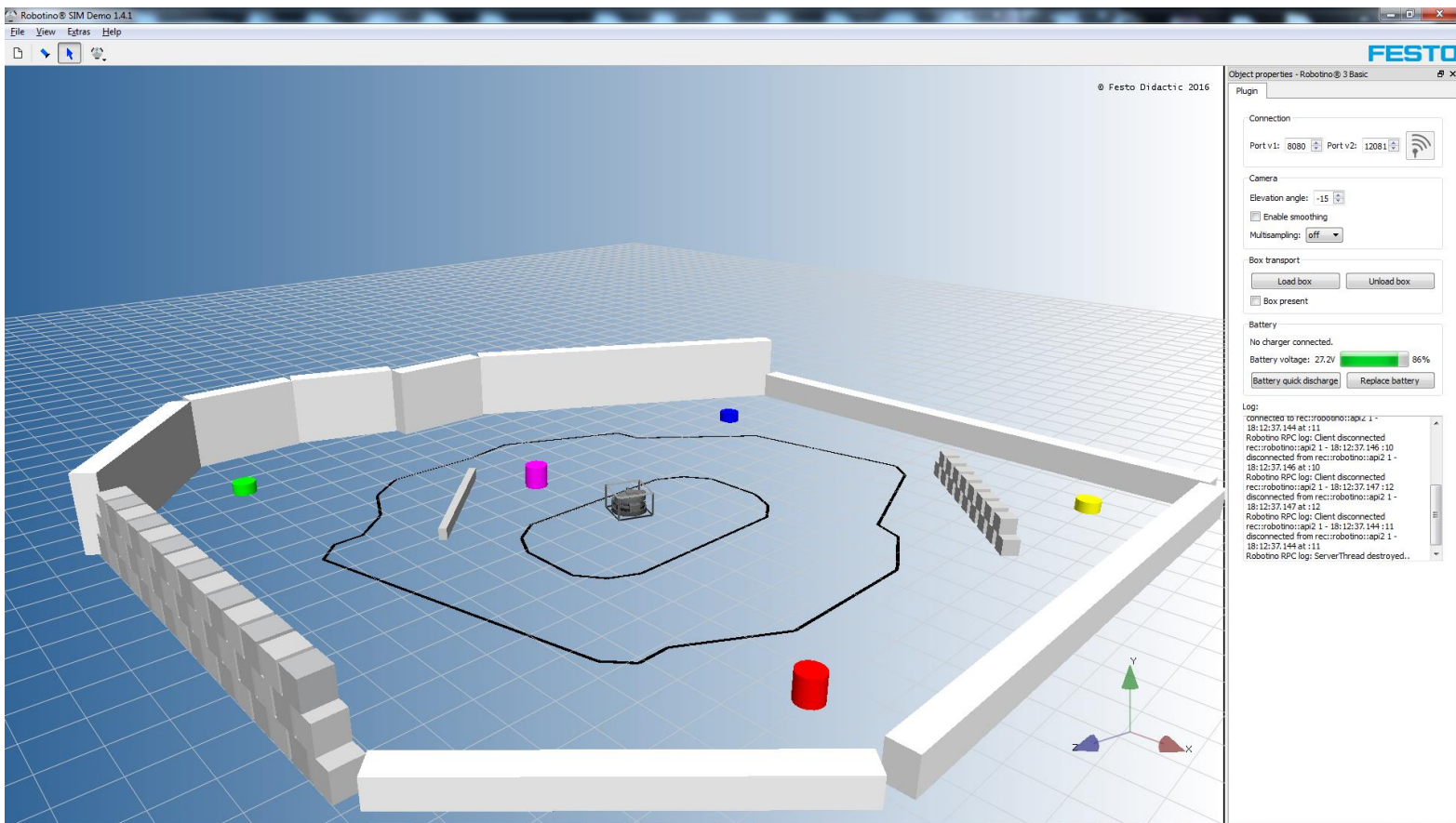


3.2.2. Simulace

Pro správný postup programování čehokoliv je vždy dobré mít určitou zpětnou odezvu své práce, ideálně pro libovolný krok své práce, díky níž můžeme předejít zbytečným komplikacím a chybám a zefektivnit samotné programování.

Využijeme k tomu zmíněný program Robotino SIM Demo. Ten nám nabízí propracované 3D prostředí pro vizualizaci a testování programů vytvořených v předcházejícím programu.

Pro správnou funkčnost musíme propojit programovací a simulační program. To provedeme zkopírováním hodnoty „Port v2“ (implicitně 12080) zobrazené v simulačním programu a vložíme s dvojtečkou za lokální virtuální IP adresu Robotina v programovací části. Stejně adresové pole pro IP využíváme pro propojení počítače s Robotinem.



3.3. První úloha - "Bezpečná jízda po pruhu"

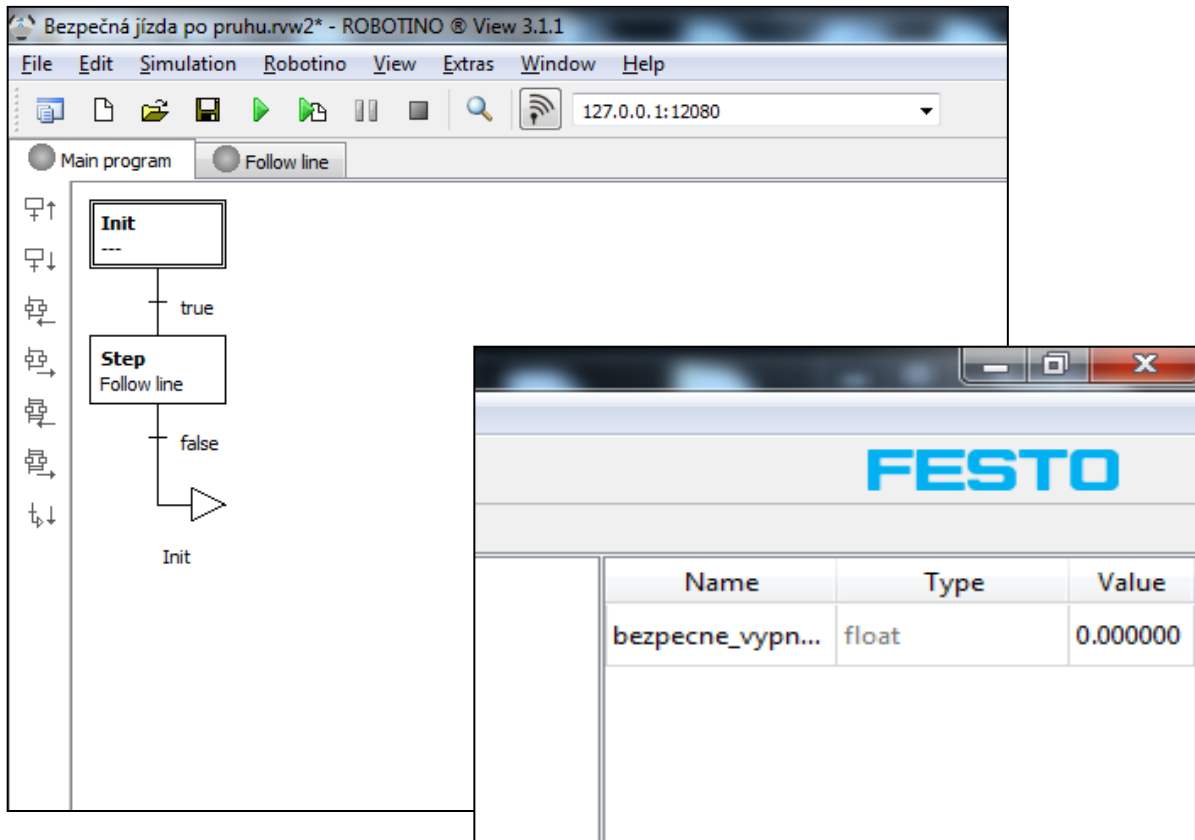
3.3.1. Cíl úlohy

Cílem této, první úlohy, je, aby Robotino® pomocí svých infračervených senzorů umístěných ve spodní části robotu, snímajících povrch země, detekoval a jel po vyznačeném pruhu na povrchu. V případě, že dojde ke snížení napětí baterie pod 24,5 V, či dojde-li k poklesu procentuální intenzity Wi-Fi signálu pod hodnotu 65, robot se sám zastaví.

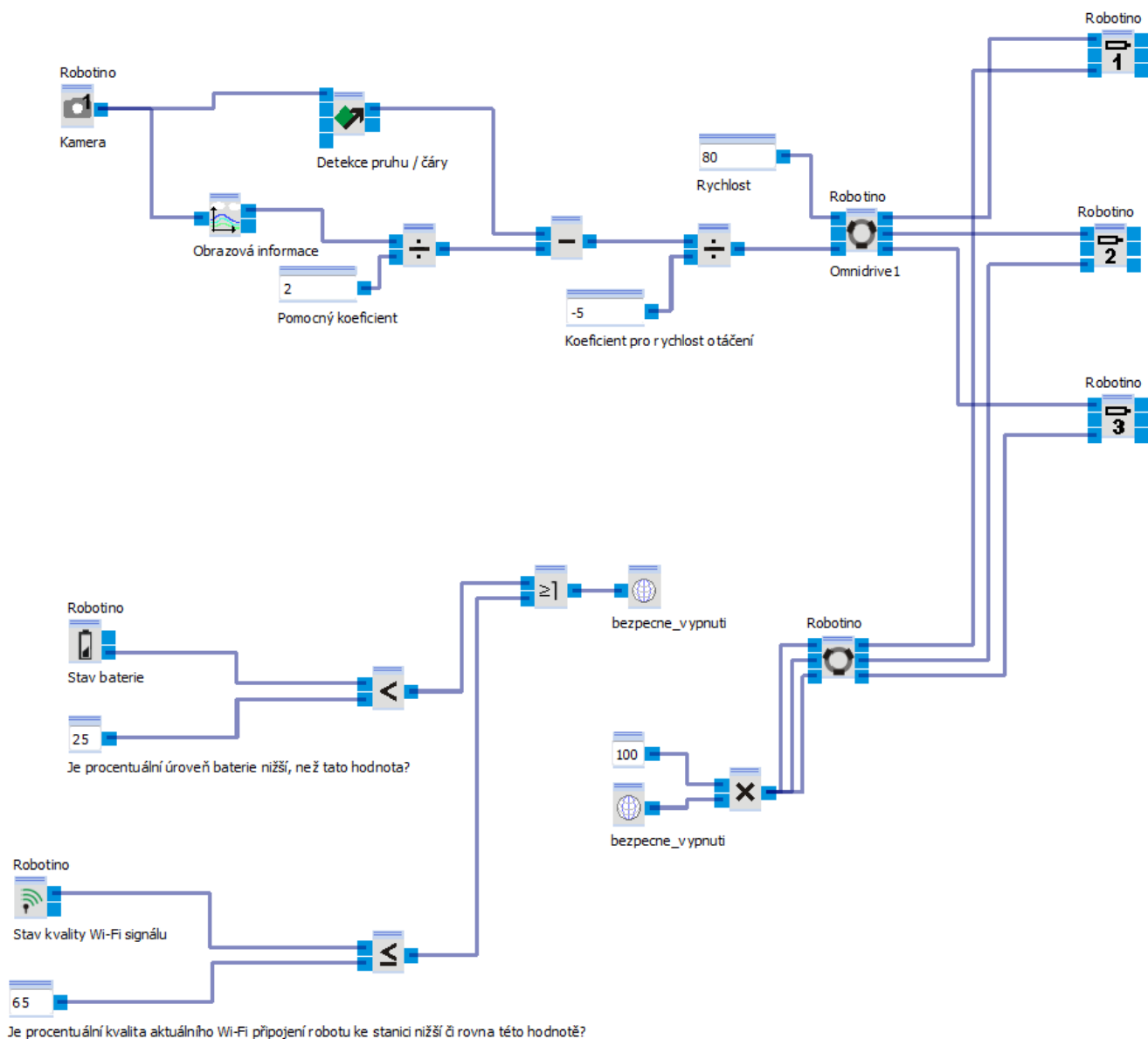
3.3.2. Použité programovací bloky

V této úloze byly využity funkce jak interních, tak externích senzorů – Interními myslíme snímač napětí na baterii a snímač intenzity signálu WiFi, externí jsou zastoupeny několika senzory, jež snímají vizuální podněty okolí - obrazovým senzorem využívajícím výpočty z reálného obrazu z kamery a infračerveným senzorem pro detekci čáry na zemi. Dále byly z několika důvodů využity bloky základních matematických operací, jež nám v případě využití komparativních matematických vyhodnocovacích bloků umožňují například realizaci bezpečnostního obvodu zajišťujícího přidání dalšího stupně sofistikovanosti a propracovanosti ve smyslu kontinuálního monitorování stavu napětí baterie a stavu procentuální kvality signálu Wi-Fi. Samozřejmým základem každé úlohy mající za cíl jakýmkoliv způsobem pohybovat s robotem je nutnost přítomnosti takzvaného systému OmniDrive, jenž nám umožňuje rozpohybování jednotlivých (případně i separátních) motorů, jejichž bloky musí být samozřejmě v obvodu implementovány taktéž.

3.3.3. Ukázka programu



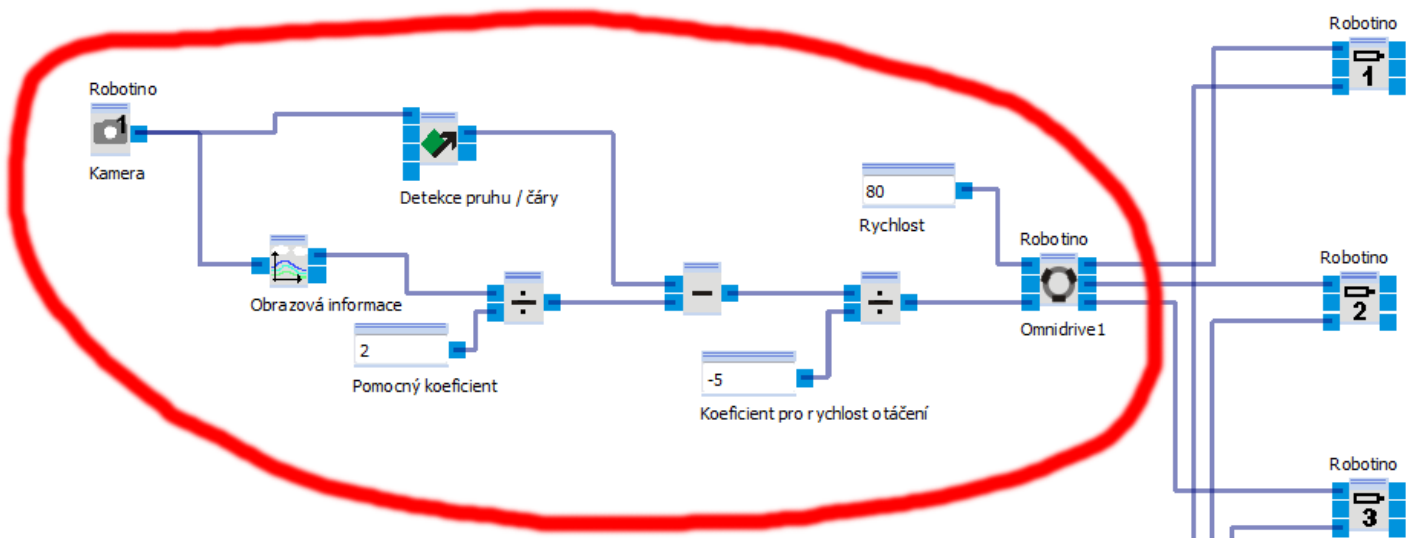
Zde si můžeme prohlédnout první krok programování první úlohy. Na pravém snímku obrazovky vidíme první vytvořenou proměnnou hodnotu pod názvem „bezpecne_vypnuti“. Tato proměnná hodnota nám zastupuje logickou 1 či 0 v závislosti na pravdivosti určitých monitorovaných jevů v obvodu, který si dále popíšeme. Stručně řečeno slouží k zastavení robotu, dojde-li ke splnění jednoho z rizikových stavů. Na levém snímku obrazovky si můžeme povšimnout algoritmického stromu o dvou blocích. Prvním z nich je blok Init, tedy Inicializace – Start celého obvodu. Pod ním vidíme blok jménem Step, s dodatkovým popisem hlásajícím „Follow line“. Tento blok zastupuje exekuci samotného programu, jež byl naprogramován. Mezi každým z kroků (bloků) algoritmu je tzv. podmínkový stupeň, jenž má buď hodnotu „true“ nebo „false“. Může mít také podobu určité podmínky, jejíž splnění zajistí výsledek „true“ a nesplnění výsledek „false“. Pokud je vyhodnocení kladné, dojde k postoupení programu do dalšího bloku algoritmu.



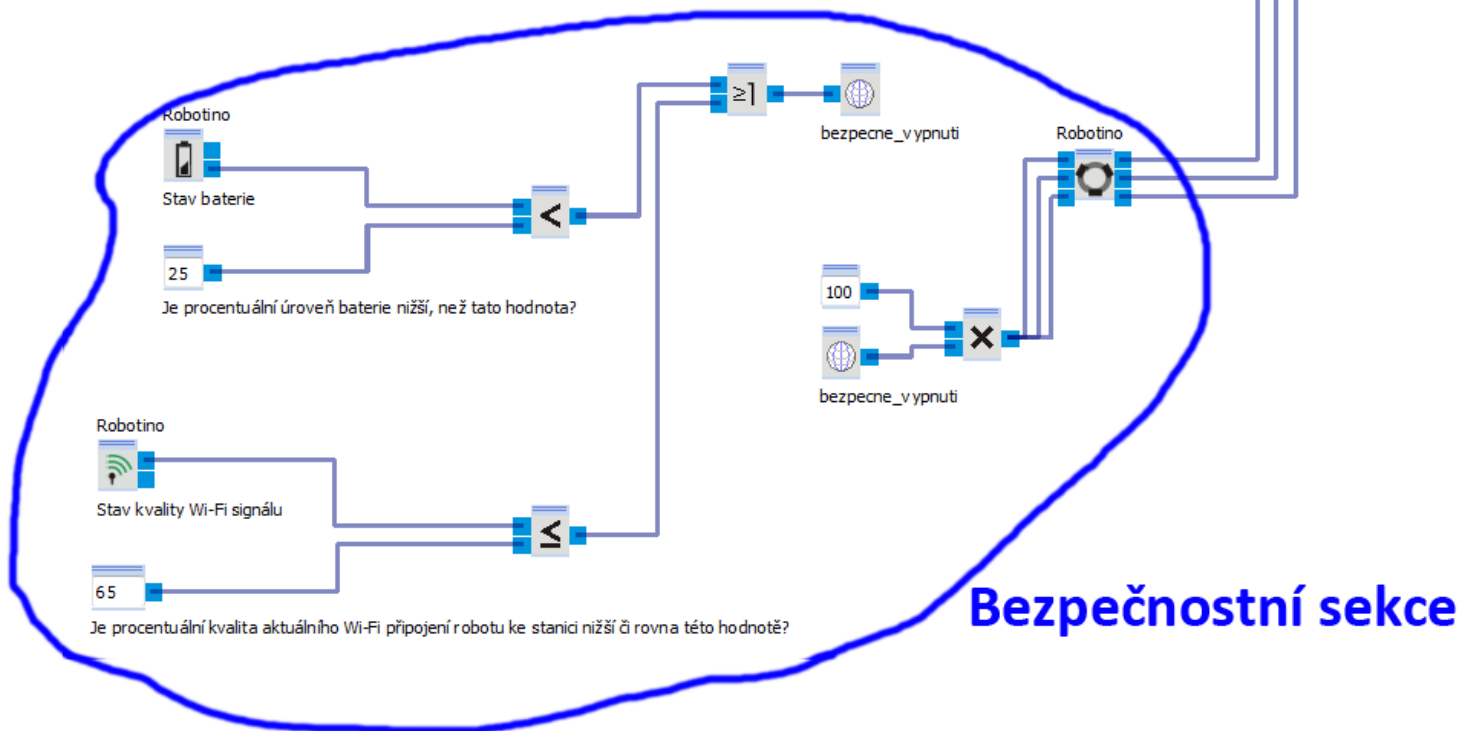
Tento obvod si můžeme pro přehlednost rozdělit do dvou segmentů, které dohromady tvoří jeden funkční celek. V prvním z nich, v detekčním segmentu, se nacházejí sensorické bloky zajišťující detekci pruhu na povrchu země, doprovázené bloky matematických operací nezbytných pro správnou funkčnost a využití surových dat získaných ze sensorických bloků pro použití pro samotný výkon pohybu.

V sekci bezpečnostní zase vidíme bloky monitorující dvě hodnoty, stav napětí baterie a spotřeby energie, z nichž nás zajímá první zmíněná: Komparativními funkčními bloky zajistíme, že v případě poklesu napětí baterie pod libovolnou hodnotu V (v našem případě jde o hodnotu 24,5 V), dojde k zbrždění motorů robotu a jeho zastavení.

Na této stránce si můžeme prohlédnout zřehledněné rozdělení tohoto obvodu.



Detekční sekce



Bezpečnostní sekce

3.4. Druhá úloha - "Kopírování zdi"

3.4.1. Cíl úlohy

Cílem této, druhé úlohy, je, aby Robotino® lineárním pohybem našlo zeď a využitím svých vzdálenostních senzorů a řady funkčních bloků udržovalo ve svém následovném kontinuálním pohybu plynule vzdálenost 10cm od překážky.

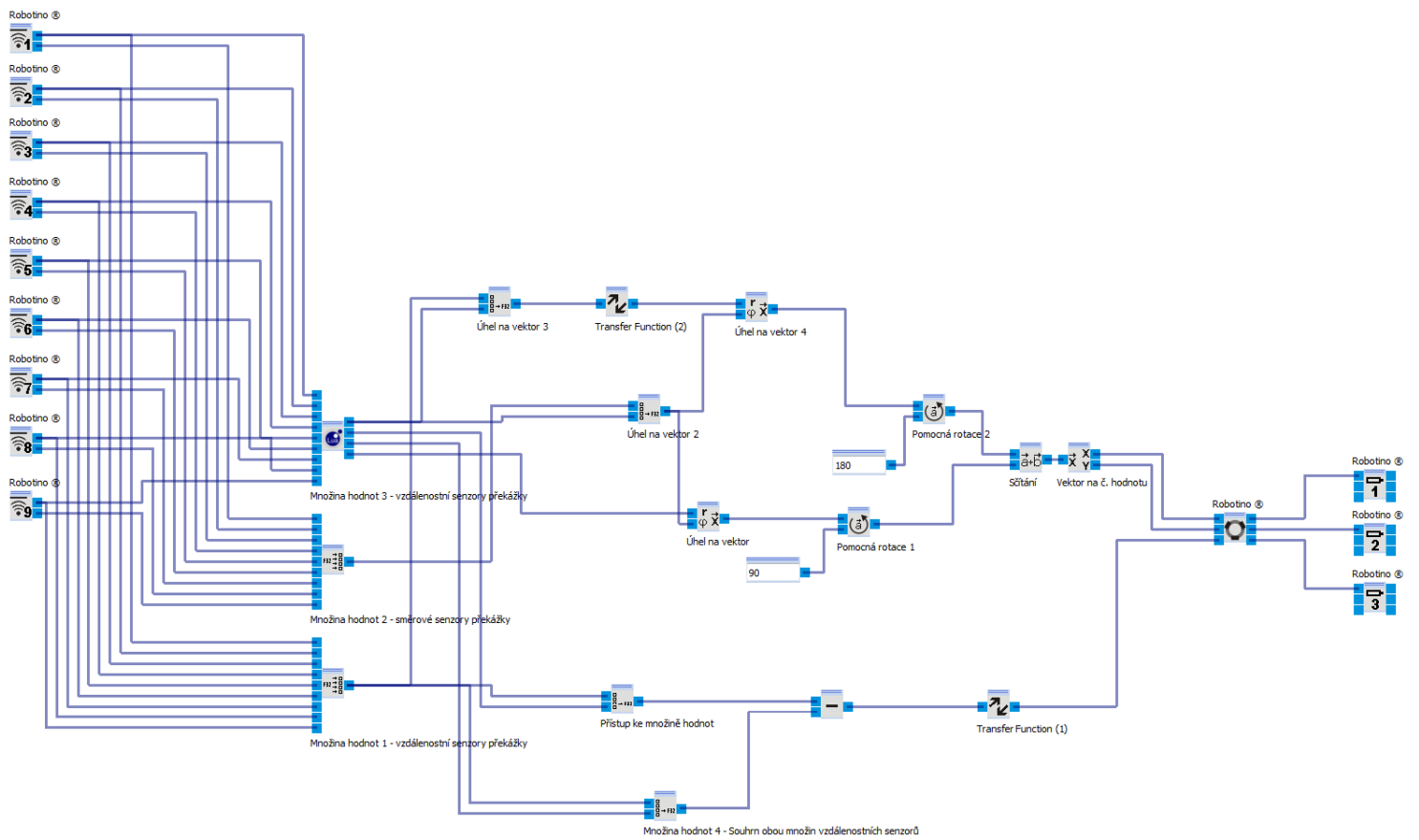
3.4.2. Použité programovací bloky

V této programovací úloze je hlavním senzorkým blokem optický snímač vzdálenosti od překážky. Samozřejmě je však potřebné využít všech 9ti těchto senzorů. Dále je potřeba využít bloky převodů úhlových hodnot na vektory a převodních bloků z vektoru na číselnou hodnotu, vysvětleno níže. Dále je použit systém OmniDrive a bloky jednotlivých motorů.

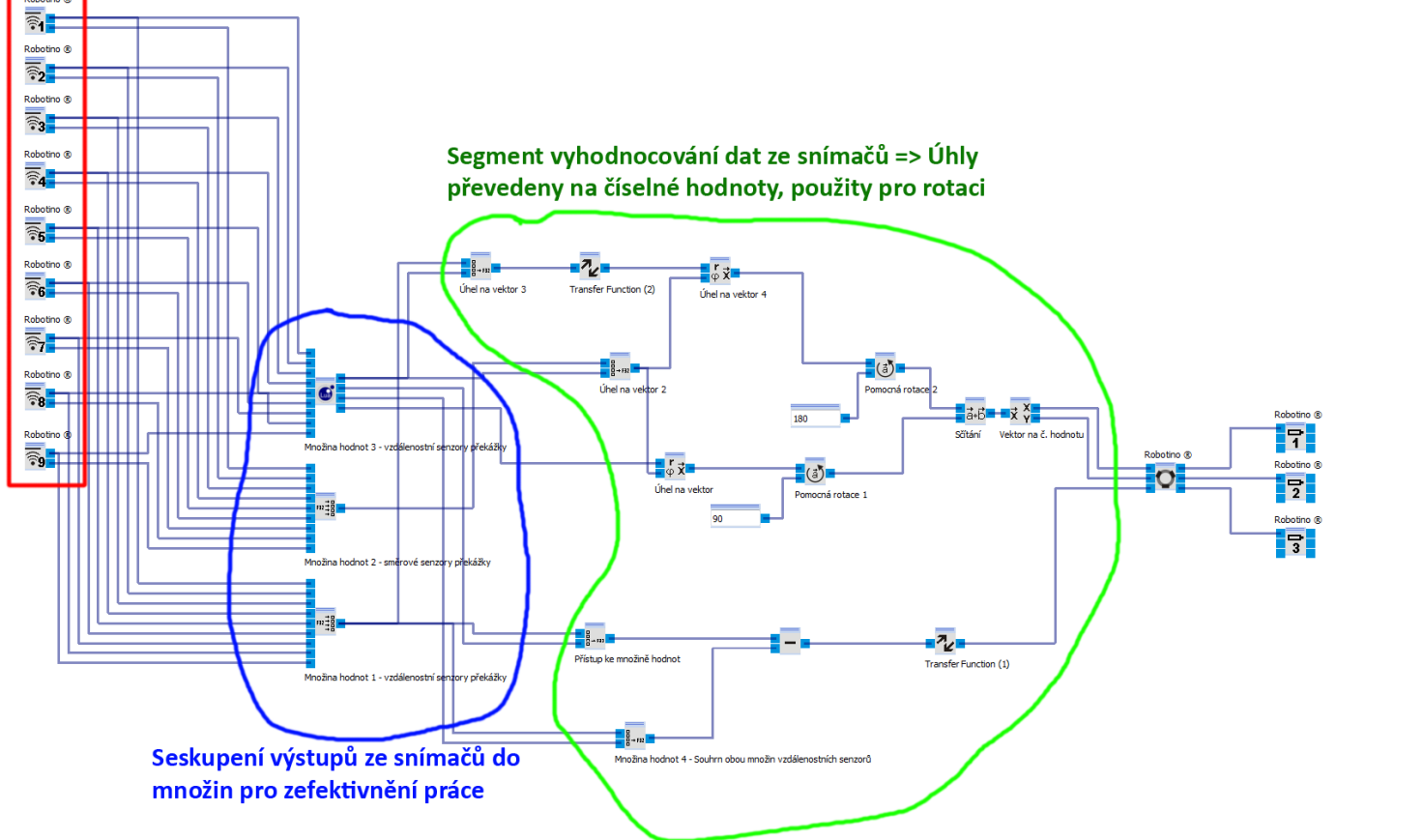
3.4.3. Ukázka programu

Jelikož z každého z optických snímačů vzdálenosti jsou výstupem 2 druhy dat, kde prvním z nich je samotná vzdálenost a druhým směr, kterým se překážka nachází v relaci k samotnému konkrétnímu senzoru, je nutné v případě využití obou těchto dat převést úhlovou hodnotu směru převést na vektor pro následovné použití ve vyhodnocovacích a funkčních blocích. To je realizováno speciálními funkčními bloky trefně nazvanými „Úhel na vektor“. Tím zajistíme správnost obvodu pro pokračování do bloků zajišťujících rotační pohyb robotu jakožto celku. Součtem hodnot bloků pomocných rotací dosáhneme další části segmentu vyhodnocování dat (viz. obrázek níže), kde již potřebujeme vektor převést na číselnou hodnotu pro použití v samotných vstupech systému OmniDrive, který vyžaduje zadání přímé rychlosti pro osu X a osu Y.

Opět na prvním snímku vidíme surový snímek obrazovky, na druhém zpřehledněný.



9 vzdálenostních senzorů



3.5. Třetí úloha - "Provokace"

3.5.1. Cíl úlohy

Cílem této úlohy je zajistit pohyb robotu k, za předpokladu, že je taková reálná (že existuje), nejbližší překážce, a to kontinuálně i ve směru jejího pohybu.

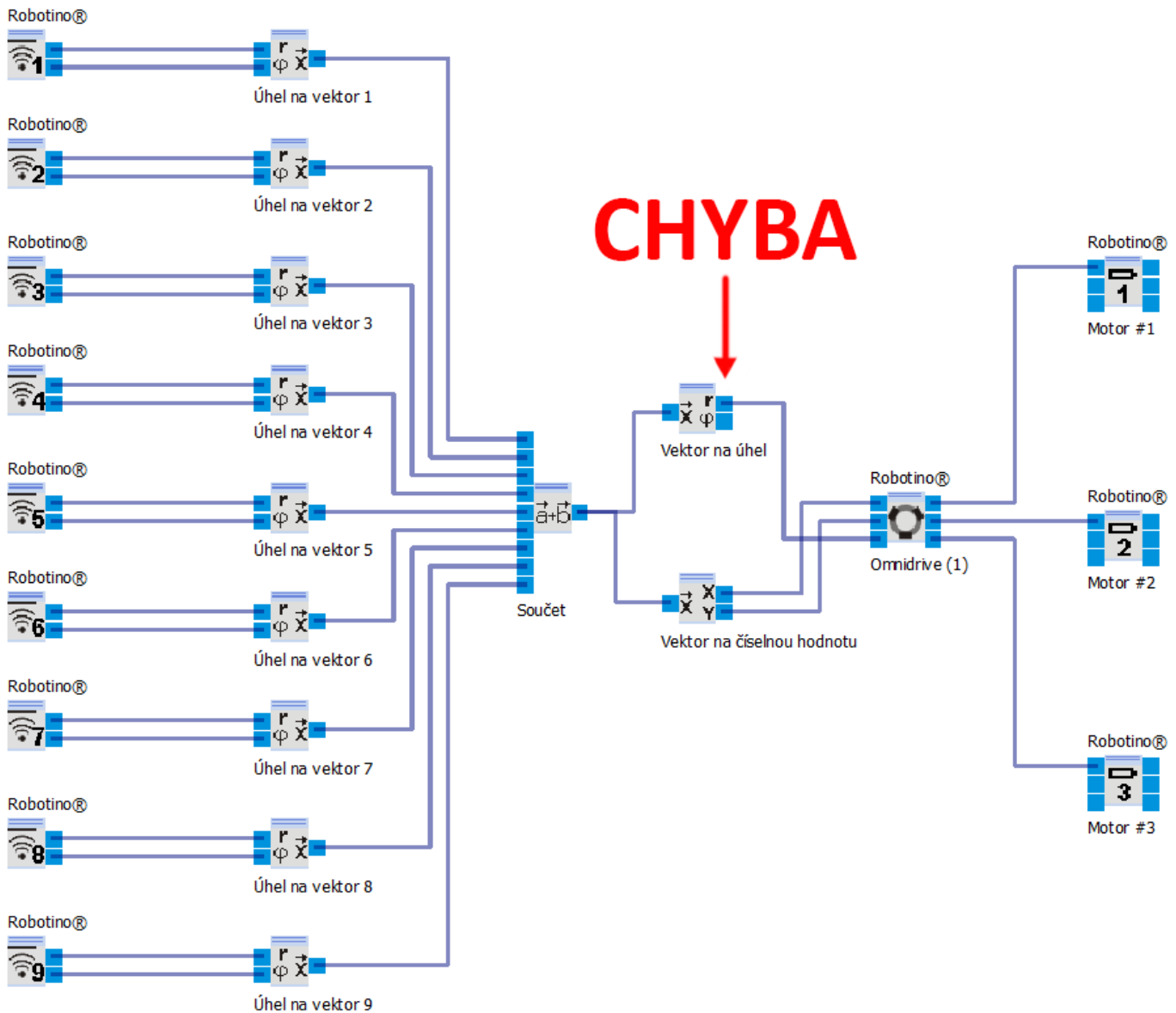
3.5.2. Použité programovací bloky

V této úloze je opět hlavní devízou blok optického snímače vzdálenosti, dále je použit převodní blok úhel – vektor, vektor – číslo, systém OmniDrive a jednotlivé motory.

3.5.3. Ukázka programu

Vzhledem k tomu, že během programování této, paradoxně jednodušejší vypadající úlohy, došlo k fatálnímu problému způsobujícímu havárii simulačního softwaru, zvolím pro ukázkou této unikátní metodu, kde nejprve ukáži vadný obvod, ve kterém musela být vada lokalizována a odstraněna, a ve druhém obrázku zřehledněný správný obvod s objasněním závady. Opět využíváme oba výstupy bloků optických vzdálenostních snímačů osazených v počtu 9 po obvodu podstavy robotu, jejichž úhlová hodnota musí být pro správnou funkčnost programu nejprve převedena na vektor a následně spolu se surovou hodnotou vzdálenosti získanou z prvního datového typu výstupu bloku optického vzdálenostního snímače sečtena sčítacím blokem pro další použití v obvodu. Jelikož potřebujeme následování robotu pohybující se překážky, je nezbytné využít jak převodních bloků vektor-úhel, tak i vektor-číslo.

Na následující stránce se dozvíme, jak vypadá tento program, včetně chyby, jež musela být eliminována pro úspěšnou funkčnost programu.

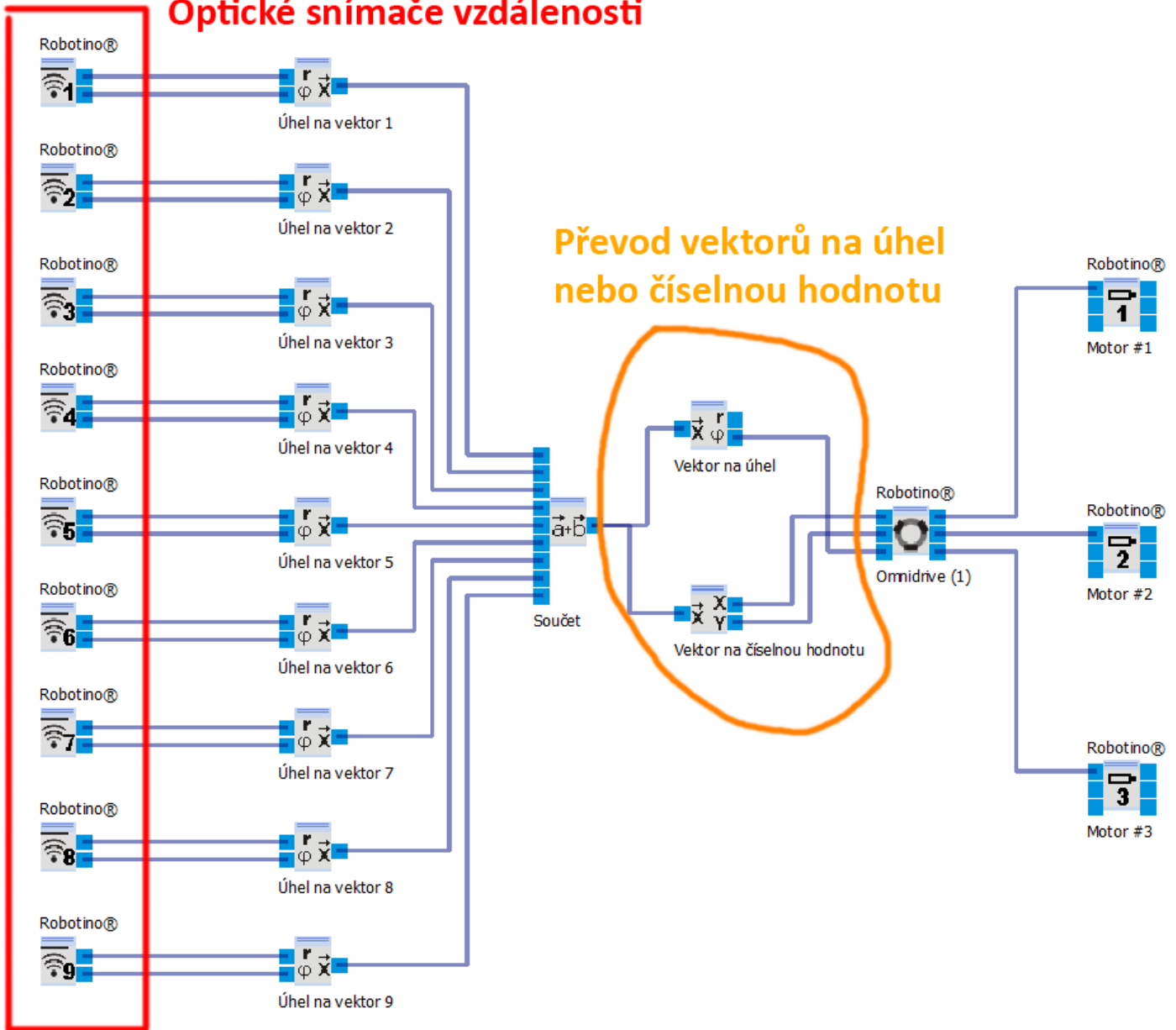


Chyba v tomto obvodu spočívala ve faktu, že došlo k použití špatného výstupu převodního bloku vektor-úhel, kde místo samotného úhlu pro použití v systému OmniDrive byl využit výstup s číselnou hodnotou poloměru otáčení.

Kvůli tomuto pochybení celý obvod nefungoval správně, jelikož jedna ze dvou hlavních hodnot => směr pohybu nebyla správně nastavena.

Opětovným přepojením do výstupu s úhlovou hodnotou program úspěšně začal fungovat a tím došlo k eliminaci chyby v obvodu

Optické snímače vzdálenosti



4. Závěr

Tato práce je zaměřena na programování Robotina pod zadáním Využití senzorů Robotina a je rozdělena na dvě části.

První částí, teoretická, je zaměřena na rozbor dané tematiky ohledně samotné konstrukce a obecného popisu robotu a firmy robot vyrábějící a v druhé praktické části, která je nosná, je zpracována samotná problematika programování.

Vzhledem k tomu, že tematika programování Robotina je, obzvláště v České Republice, relativně novým aspektem vědy pro veřejnost, nebylo prvotní seznámení s programovacím prostředím tohoto produktu nikterak jednoduché. Nicméně po vložení potřebného úsilí do studia tohoto programovacího prostředí a využití notné dávky trpělivosti jsem začínal chápat základy a později i pokročilé techniky a metody programování tohoto robota. Proto si myslím, že tato práce jakožto celek může být považována za můj úspěch.

5. Seznam použité literatury a zdrojů informací

5. 1. Další zdroje informací

- [1] Intenet
<<https://cs.wikipedia.org/wiki/Festo/>>
- [2] Internet
<<https://cs.wikipedia.org/wiki/Robotino/>>
- [3] Internet
<<http://www.festo-didactic.com/int-en/learning-systems/education-and-research-robots-robotino/robotino-for-research-and-education-premium-edition-and-basic-edition.htm/>>
- [4] Odborná konzultace s vedoucím práce

5. 2. Literatura

Vzhledem k faktu, že tematika programování Robotina, obzvláště v České Republice, je relativně mladým odvětvím veřejné vědy, není pro tuto problematiku téměř žádná dostupná literatura v podobě knižní, česky psané.

6. Seznam použitého software

- Microsoft Word 2010
- Adobe Photoshop CS6 Extended
- Robotino View 3
- Robotino SIM Demo
- Evernote

7. Seznam příloh

Veškeré přílohy jsou přiloženy na příslušném CD, které se nachází na zadní straně této práce. CD obsahuje: Příslušné programy, Vypracovanou úlohu, Maturitní práci.

8. Licenční ujednání

Ve smyslu zákona č. 121/2000 Sb., O právu autorském, o právech souvisejících s právem autorským, ve znění pozdějších předpisů (dále jen autorský zákon) jsou práva k maturitním nebo ročníkovým pracím následující:

Zadavatel má výhradní práva k využití práce, a to včetně komerčních účelů.

Autor práce bez svolení zadavatele nesmí využít práci ke komerčním účelům.

Škola má právo využít práci k nekomerčním a výukovým účelům i bez svolení zadavatele a autora práce.

