



# Robotický stolní fotbal

Popis implementovaných funkcí a jejich rozhraní

Vojtěch Myslivec, [vojtech@myslivec.net](mailto:vojtech@myslivec.net)

Květen 2015

## Obsah

<b>1</b>	<b>Anotace</b>	<b>2</b>
<b>2</b>	<b>Implementované funkce</b>	<b>2</b>
2.1	funkce <code>attackAction</code> . . . . .	2
2.2	funkce <code>attack2Action</code> . . . . .	2
2.3	funkce <code>axesInitialization</code> . . . . .	2
2.4	funkce <code>coordinateSystemTransfer</code> . . . . .	3
2.5	funkce <code>countDefense</code> . . . . .	3
2.6	funkce <code>countVectorAndEstimateNextPosition</code> . . . . .	3
2.7	funkce <code>defenseAction</code> . . . . .	3
2.8	funkce <code>disableAxes</code> . . . . .	3
2.9	funkce <code>enableAxes</code> . . . . .	4
2.10	funkce <code>initVariables</code> . . . . .	4
2.11	funkce <code>movePlayers</code> . . . . .	4
2.12	funkce <code>poke</code> . . . . .	4
2.13	funkce <code>returnAction</code> . . . . .	4
2.14	funkce <code>rotateAxisWithLimit</code> . . . . .	5
2.15	funkce <code>selectAndDoAttack</code> . . . . .	5
2.16	funkce <code>selectMode</code> . . . . .	5
2.17	funkce <code>setSearchRegion</code> . . . . .	5
2.18	funkce <code>shoot</code> . . . . .	6
<b>3</b>	<b>Implementované programy</b>	<b>6</b>
3.1	program <code>go_to_stop</code> . . . . .	6
3.2	program <code>initialization</code> . . . . .	6
3.3	program <code>main_routine</code> . . . . .	6

# 1 Anotace

Tento dokument popisuje implementované funkce a jejich vstupy a výstupy v prostředí Simotion Scout, použité při řízení jedné osy stolního fotbalu.

## 2 Implementované funkce

Vzhledem k tomu, že prostředí Simotion Scout umožňuje sledování hodnoty za běhu programu pouze pro globální proměnné, je většina metod bez vstupů a výstupů, protože čtou a zapisují z a do těchto globálních proměnných. Přestože je tento způsob nestandardní a náchylnější na chyby, byl zvolen z důvodu snadnějšího hledání a odstraňování chyb v programu.

### 2.1 funkce `attackAction`

Vstupy funkce: místo výkopu míče **target** (INT) a předchozí již provedený posun v rotační ose **shift** (INT)

Výstupy funkce: indikátor provedení výkopu **done** (INT)

Příklad volání: `attackAction(target := 300, shift := 0, done => retDINT)`

Popis funkce: *Funkce provede jednoduchý výkop míče ze zadané pozice.*

### 2.2 funkce `attack2Action`

Vstupy funkce: místo výkopu míče **target** (INT)

Výstupy funkce: nejsou

Příklad volání: `attack2Action(target := 300)`

Popis funkce: *Funkce provede výkop míče s nahrávkou spoluhráči ze zadané pozice.*

### 2.3 funkce `axesInitialization`

Vstupy funkce: nejsou

Výstupy funkce: nejsou

Příklad volání: `axesInitialization()`

Popis funkce: *Funkce provede inicializaci osy v translačním i rotačním posunu pomocí pomalého dojezdu k okraji stolu se sledováním momentu motoru a sledováním inicializační značky na ose pomocí kamery. Funkce je volána při startu systému.*

## 2.4 funkce `coordinateSystemTransfer`

Vstupy funkce: nejsou

Výstupy funkce: nejsou

Příklad volání: `coordinateSystemTransfer()`

Popis funkce: *Funkce přečte aktuální údaje o pozici míče z výstupů kamery v px a přepočte je na údaj o pozici v mm.*

## 2.5 funkce `countDefense`

Vstupy funkce: pozice osy `axisXposition` (INT), pro kterou je funkce volána

Výstupy funkce: nejsou

Příklad volání: `countDefense(axisXposition := 915)`

Popis funkce: *Funkce určí průsečík budoucí trajektorie míče a fotbalové osy, do kterého je potřeba přesunout hráče pro zamezení pohybu míče.*

## 2.6 funkce `countVectorAndEstimateNextPosition`

Vstupy funkce: nejsou

Výstupy funkce: nejsou

Příklad volání: `countVectorAndEstimateNextPosition()`

Popis funkce: *Funkce vypočte z aktuální a předchozí změřené pozice míče vektor rychlosti jeho pohybu a odhadne jeho budoucí pozici.*

## 2.7 funkce `defenseAction`

Vstupy funkce: nejsou

Výstupy funkce: nejsou

Příklad volání: `defenseAction()`

Popis funkce: *Funkce provede obrannou akci, tedy nastavení hráče osy na trajektorii pohybujícího se míče. Rychlost přesunu závisí na rychlosti míče.*

## 2.8 funkce `disableAxes`

Vstupy funkce: nejsou

Výstupy funkce: nejsou

Příklad volání: `disableAxes()`

Popis funkce: *Funkce vypne napájení motorů a uzamkne ovládání rotačního i translačního pohybu, pokud tomu tak ještě není.*

## 2.9 funkce enableAxes

Vstupy funkce: nejsou

Výstupy funkce: nejsou

Příklad volání: *enableAxes()*

Popis funkce: *Funkce zapne napájení motorů a umožní ovládání rotačního i translačního pohybu, pokud tomu tak ještě není.*

## 2.10 funkce initVariables

Vstupy funkce: nejsou

Výstupy funkce: nejsou

Příklad volání: *initVariables()*

Popis funkce: *Funkce zajistí inicializaci základních proměnných, jako jsou například transformační konstanty pro přepočítání souřadnic míče mezi hodnotou v px a hodnotou v mm. Funkce je volána při startu systému.*

## 2.11 funkce movePlayers

Vstupy funkce: cílová pozice pro hráče **position** (INT)

Výstupy funkce: indikátor úspěšného nebo neúspěšného přesunu **moved** (BOOL)

Příklad volání: *movePlayers(position := 300, moved => retBOOL)*

Popis funkce: *Funkce přesune nejbližšího možného hráče na zadanou pozici, pokud je to možné (zadaná pozice je smysluplná).*

## 2.12 funkce poke

Vstupy funkce: vzdálenost míče od osy **distance** (INT) a směr nahrávky **direction** (INT), s možnostmi +1 (vlevo ve směru od vlastní brány) nebo -1 (vpravo)

Výstupy funkce: nejsou

Příklad volání: *poke(distance := 20, direction := +1)*

Popis funkce: *Funkce provede nahrávku míče do strany z aktuální pozice osy.*

## 2.13 funkce returnAction

Vstupy funkce: nejsou

Výstupy funkce: nejsou

Příklad volání: *returnAction()*

Popis funkce: *Funkce zajistí přesun osy do výchozí pozice na středu hřiště.*

## 2.14 funkce rotateAxisWithLimit

Vstupy funkce: rychlost otáčení **velocity** (INT), akcelerace při rozběhu a doběhu **acceleration** (DNT), relativní pozice otočení **position** (INT), z jejího znaménka je určen směr rotace (+ positive, - negative)

Výstupy funkce: proměnná indikující úspěšné dokončení pohybu **done** (DINT) v případě, že je nulová, pokud je nenulová, pohyb nebyl dokončen a proměnná obsahuje údaj o uražené vzdálenosti ve stupních rotace motoru

Příklad volání: `rotateAxisWithLimit1(velocity := 8000, acceleration := 100000, position := 400, done => retDINT)`

Popis funkce: *Funkce rotuje osou s vnitřně nastaveným limitem na moment síly motoru, v případě jeho překročení (zaseknutí míče) se pohyb přerušuje.*

## 2.15 funkce selectAndDoAttack

Vstupy funkce: nejsou

Výstupy funkce: nejsou

Příklad volání: `selectAndDoAttack()`

Popis funkce: *Funkce zvolí metodu výkopu míče podle jeho pozice a zavolá příslušnou funkci.*

## 2.16 funkce selectMode

Vstupy funkce: pozice osy **axisXposition** (INT), pro kterou je funkce volána

Výstupy funkce: nejsou

Příklad volání: `selectMode(axisXposition := 915)`

Popis funkce: *Funkce podle pohybu míče zvolí herní mód osy, obranu, útok (výkop), návrat do výchozí pozice nebo žádnou akci.*

## 2.17 funkce setSearchRegion

Vstupy funkce: nejsou

Výstupy funkce: nejsou

Příklad volání: `setSearchRegion()`

Popis funkce: *Funkce určí oblast vyhledávání míče na hrací ploše podle jeho odhadované příští pozice a rozměry nové oblasti zapíše do příslušných proměnných reprezentujících vstupní proměnné kamery.*

## 2.18 funkce shoot

Vstupy funkce: předchozí, již provedený posun v rotační ose **shift** (INT)

Výstupy funkce: nejsou

Příklad volání: *shoot(shift := 200)*

Popis funkce: *Funkce provede výkop míče na aktuální poloze osy a poté návrat do původní pozice změněné o hodnotu vstupní proměnné shift.*

## 3 Implementované programy

### 3.1 program go\_to\_stop

Příklad volání: Program je nutné přiřadit k některé z úloh v *Execution System*

Popis programu: *Provádí operace potřebné při vypnutí programu.*

### 3.2 program initialization

Příklad volání: Program je nutné přiřadit k některé z úloh v *Execution System*

Popis programu: *Program provede inicializaci osy v translačním i rotačním posunu pomocí pomalého dojezdu k okraji stolu se sledováním momentu motoru a sledováním inicializační značky na ose pomocí kamery.*

### 3.3 program main\_routine

Příklad volání: Program je nutné přiřadit k některé z úloh v *Execution System*

Popis programu: *Hlavní řídicí smyčka, načítá údaje o změřené poloze z kamery, vybírá a provádí adekvátní reakci osy.*