



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

MODEL ROBOTICKÉHO STOLNÍHO FOTBALU

MODEL ROBOTIC TABLE SOCCER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Ľubomír Bubeník

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. Petr Pivoňka, CSc.

BRNO 2017

Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Ľubomír Bubeník

ID: 164840

Ročník: 3

Akademický rok: 2016/17

NÁZEV TÉMATU:

Model robotického stolního fotbalu

POKYNY PRO VYPRACOVÁNÍ:

- 1) Zpracujte rešerši týkající se robotického stolního fotbalu a algoritmy zpracování scény.
- 2) Seznamte se se softwarem Automation Studio V4 a vizualizace od společnosti B&R.
- 3) Seznamte se s inteligentní kamerou pro detekci polohy míčku.
- 4) Navrhněte řídicí algoritmus pro optimální nasvětlení scény pro vyhodnocení polohy míčku inteligentní kamerou.
- 5) Ověřte optimální nasvětlení scény pro inteligentní kameru na reálném modelu robotického stolního fotbalu.

DOPORUČENÁ LITERATURA:

[1] Firemní literatura B&R: Automation Studio V4

[2] Firemní literatura B&R: Produkty pro řízení pohybu a vizualizace společnosti B&R

Termín zadání: 6.2.2017

Termín odevzdání: 29.5.2017

Vedoucí práce: prof. Ing. Petr Pivoňka, CSc.

Konzultant: Ing. Roman Mužík, B&R Brno

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Táto práca sa zaoberá návrhom osvetlenia a scény pre model stolného robotického futbalu. Pozostáva hlavne z rozboru, aké sú možnosti osvetlenia a výberom najvhodnejšieho. Taktiež sa zaoberá návrhom správneho algoritmu, ktorý dokáže rozpoznať polohu loptičky na hracej ploche stolného futbalu s veľkou spoľahlivosťou s priblížením sa k reálnemu času. Súčasťou je aj spracovanie dát z kamery v systéme PLC.

Kľúčové slová

Robotický stolný futbal, kamera, In-Sight, scéna, osvetlenie, PLC, B&R, komunikácia

Abstract

This thesis deals with design of lighting scenes for the robotic table football. It consists mainly of analysis of what are the lighting options and selecting the most appropriate. It also deals with creating algorithm that can detect the position of the ball at the board of table football with great reliability and with approaching to real time. Thesis also includes data processing from the camera in the system PLC.

Key words

Robotic table football, camera, In-Sight, scene, lighting, PLC, B&R, communication

Bibliografická citácia:

BUBENÍK, L. *Model robotického stolního fotbalu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 54 s. Vedoucí bakalářské práce prof. Ing. Petr Pivoňka, CSc.

Prehlásenie

„Prohlašuji, že svou bakalářskou práci na téma Model robotického stolního fotbalu jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **24. května 2017**

.....
Podpis autora

Pod'akovanie

Ďakujem vedúcemu mojej bakalárskej práce prof. Ing. Petrovi Pivoňkovi, CSc. za jeho pedagogickú a odbornú pomoc a ďalšie cenné rady pri spracovaní bakalárskej práce. Moje pod'akovanie patrí aj konzultantovi Ing. Romanovi Mužíkovi za odbornú pomoc. Tiež chcem pod'akovať firme B+R automatizace s.r.o. ktorá poskytla prostriedky pre túto bakalársku prácu. Ďalej chcem pod'akovať firme Smart View za zapožičanie osvetlenia a cenné rady.

V Brne dňa: **24. Mája 2017**

.....
Podpis autora

Obsah

1	Úvod	9
2	Stolný futbal.....	10
3	Existujúce riešenia robotického stolného futbalu	11
	3.1 Ecole polytechnique f'ed'erale de Lausanne.....	11
	3.2 Technical University of Denmark	11
	3.3 PILZ	12
4	Použitý hardware a scéna.....	14
	4.1 Hracia plocha	14
	4.2 Kamera	15
	4.2.1 Kamera umiestnená nad hracou plochou	17
	4.2.2 Kamera umiestnená pod hracou plochou.....	18
	4.3 Osvetlenie.....	19
	4.3.1 Poloha osvetlenia	19
	4.3.2 Výber osvetlenia	21
	4.4 Detekcia gólov	23
	4.5 Hardwarová konfigurácia.....	23
	4.5.1 Použité PLC a zobrazovací panel	24
	4.5.2 Vstupné a výstupné moduly.....	25
	4.5.3 Menič a lineárne motory	26
	4.5.4 Servo zosilňovač a rotačné krokové motory	26
	4.6 Komunikácia	27
5	Použitý Software.....	29
	5.1 In-Sight Explorer.....	29
	5.1.1 Parametre snímania a vyhľadávanie	29
	5.1.2 Optimalizácia času vyhľadávania	31
	5.1.3 Komunikácia s kamerou v prostredí In-Sight Explorer	35
	5.2 Automation Studio	35
	5.2.1 Programovacie jazyky	36
	5.2.2 Funkcia pre spracovanie dát z kamery.....	36
	5.2.3 Funkcia pre ovládanie osvetlenia.....	43
	5.2.4 Funkcia pre ovládanie brán.....	45
	5.2.5 Vizualizácia	46
6	Záver.....	48

Zoznam obrázkov	49
Zoznam tabuliek	50
Literatúra	51
Zoznam symbolov a skratiek	53
Zoznam príloh.....	54

1 ÚVOD

V dnešnej dobe je veľkým trendom automatizovať takmer všetky zariadenia a procesy. Tomuto trendu sa nevyhla ani oblasť športu a zábavy. Ako jeden z najväčších zameraní, čo sa týka športu ako zábavy, môžeme zaradiť aj stolný futbal, ktorý patrí medzi stolné hry. Veľa ľudí si myslí, že je to len zábava, ale možno vás prekvapí, že stolný futbal je taktiež klasifikovaný aj ako profesionálna športová disciplína. Myšlienka automatizácie stolného futbalu plynie z otázky, či môže byť stroj lepší ako človek. Príkladom je legendárna šachová hra Garriho Kasparova proti počítaču Deep Blue od firmy IBM, ktorá sa odohrala pred viac ako 19 rokmi. V tej dobe bol Kasparov majstrom sveta v šachu, prvú hru vyhral Kasparov, ale druhú vyhral počítač. Vtedy počítač vyhral vďaka veľkému výpočtovému výkonu, dnes je taktika omnoho sofistikovanejšia a založená aj na schopnosti učiť sa.

Táto bakalárska práca sa zaoberá hlavne detekciou polohy loptičky na hracej ploche stolného futbalu a následným spracovaním údajov. Tieto údaje budú kľúčové pre pohyb motorov, ktoré riadia hráčov na hracom poli. Poloha bude detegovaná pomocou inteligentnej kamery In-Sight. Pre správne rozpoznanie loptičky v optimálnom čase je nutné vytvoriť dostatočné osvetlenie plochy, ktorá zaistí vhodné podmienky pre detekciu tak, aby malo okolité osvetlenie čo najmenší vplyv na rozpoznávanie.

Jedným z hlavných cieľov práce je zoznámiť sa s programovacími prostrediami Automation studio a In-Sight Explorer. Ďalej je práca zameraná na zoznámenie sa s použitou kamerou a hardwarom použitým pre spracovanie a interpretáciu údajov z kamery. Cieľom práce je tiež výber najvhodnejšieho osvetlenia a scenérie. V poslednom rade bude na reálnom modeli stolného futbalu testované vyhľadávanie loptičky na hracej ploche.

2 STOLNÝ FUTBAL

Stolný futbal je medzinárodne najčastejšie nazývaný aj ako table soccer alebo foosball. Jeho história začína v roku 1921 vo Veľkej Británii, kde ho vynášiel Harold Searles Thornton.

Typické rozmery stolného futbalu sú 120 x 61 cm. Hracia plocha sa skladá z 8 tyčí s hráčmi, kde každý tím má k dispozícii štyri tyče s hráčmi v rovnakej farbe. Každá z tyčí má pritom špecifickú funkciu. Na prvej tyči od bránky sa nachádza brankár, na druhej sú dvaja obrancovia, na tretej je záloha, kde je najväčší počet hráčov, a to päť. Na poslednej sa nachádzajú traja útočníci.



Obr. 1 Stolný futbal od firmy B+R pred úpravou [1]

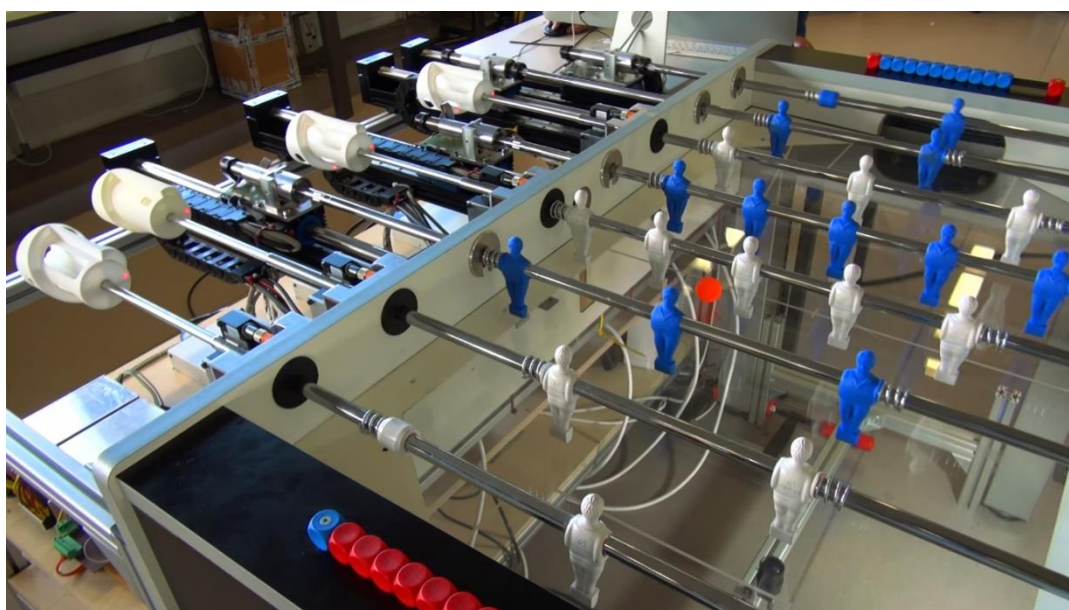
Stolný futbal je hra určená najčastejšie pre štyroch hráčov, ale nie je výnimkou ani hra dvoch hráčov proti sebe. V prípade štyroch hráčov sú hráči rozdelení do dvoch tímov po dvoch. V tíme si hráči rozdelia funkcie, kde jeden hráč obsluhuje brankára s obrancami a druhý obsluhuje záložníkov s útočníkmi. Cieľom hry je trafiť loptičku do súperovej bránky. Pri streľbe je povolené hráča pretočiť maximálne o 360 stupňov, ale v niektorých pravidlách sa uvádza maximálne 2 x 360 stupňov. Hra sa končí, pokiaľ jeden z tímov strelí dopredu dohodnutý počet gólov, štandardne od 5 do 11. [2] [3]

3 EXISTUJÚCE RIEŠENIA ROBOTICKÉHO STOLNÉHO FUTBALU

Vo svete, ale aj doma, existujú rôzne úpravy stolného futbalu tak, aby robot dokázal nahradiť celý tím na jednej strane hracieho stola. Tým vzniká pomenovanie robotický stolný futbal. Úpravou sa zaoberajú najmä univerzity, ale aj firmy. Motiváciou je vedecký výskum, zábavný priemysel, ale aj propagačné účely.

3.1 Ecole polytechnique f'ed'erale de Lausanne

V roku 2015 predstavila svoje riešenie robotického stolného futbalu švajčiarska technická univerzita v Lausanne. Hráči sú ovládaní pomocou rotačných a lineárnych motorov, ktoré zabezpečujú rýchle premiestnenie hráčov s veľkou presnosťou. Polohu hráčov, s ktorými hrá človek, určuje séria vždy dvoch optických snímačov, ktoré získavajú polohu z dômyselného prípravku, ktorý je pripevnený k tyči. Hracia plocha je z priehľadného materiálu, vďaka čomu môže byť kamera umiestnená pod hracou plochou. Táto kamera dokáže zaznamenať až 300 farebných snímkov za sekundu. Snímky sú prenášané do počítača, ktorý vyhodnocuje polohu loptičky. Plocha nedisponovala externým osvetlením. [4] [5]



Obr. 2 Pohľad na technické vybavenie robotického futbalu z univerzity Ecole polytechnique f'ed'erale de Lausanne [4]

3.2 Technical University of Denmark

Projekt z Dánskej technickej univerzity bol ukončený v roku 2007. V tom období bol projekt už plne funkčný, schopný konkurovať ľudskému protivníkovi. Pohyb jednotlivých hráčov zabezpečovala séria dvoch rotačných motorov, kde jeden sa staral priamo o rotačný

pohyb hráčov a druhý pomocou prevodov zabezpečoval lineárny pohyb. Táto konštrukcia bola relatívne masívna. Loptička bola rozpoznávaná pomocou kamery umiestnenej nad hracou plochou. Snímky z kamery boli odosielané do počítača, ktorý z nich vyhodnocoval pozíciu loptičky. Kamera bola čiernobiela s frekvenciou 25 snímkov za sekundu. [6]



Obr. 3 Pohľad na technické vybavenie robotického futbalu z Technickej Univerzity v Dánsku [5]

3.3 PILZ

Tento robotický stolný futbal od firmy PILZ bol prezentovaný na medzinárodnom veľtrhu Ampér v roku 2016. Pohyb vykonával rotačný motor, ktorý pomocou skrutkovice prenášal pohyb na lineárny. Natočenie hráčov sa menilo prostredníctvom rotačného motora spojeného s tyčou pomocou remeňa. Hracia plocha je vyrobená z matného priehľadného materiálu. Kamera aj osvetlenie je umiestnené pod hracou plochou.



Obr. 4 Pohľad na robotický futbal od firmy PILZ [1]

4 POUŽITÝ HARDWARE A SCÉNA

Táto časť sa bude zaoberať najmä hardwarovými časťami, s ktorými prideme priamo do styku. Bude sa teda jednať o samotný stolný futbal, použitú kameru a jej súčasti, ktoré budú popísané v kapitole 4.2. Ďalšou časťou bude výber optimálnej kombinácie osvetlenia a jeho polohy, ktoré bude zaisťovať potrebné osvetlenie hracej plochy. Ďalej sa zameriame na prípravok určený k detekcii gólov. V poslednej časti sa zameriame na použitý PLC systém, pohony a komunikáciu s kamerou.

4.1 Hracia plocha

Použitý stolný futbal je od firmy Garlando model G-500 EVOLUTION. Rozmery stola súhlasia so štandardmi, s akými sa hralo na ITSF World Championship. Špecifikácia stola je:

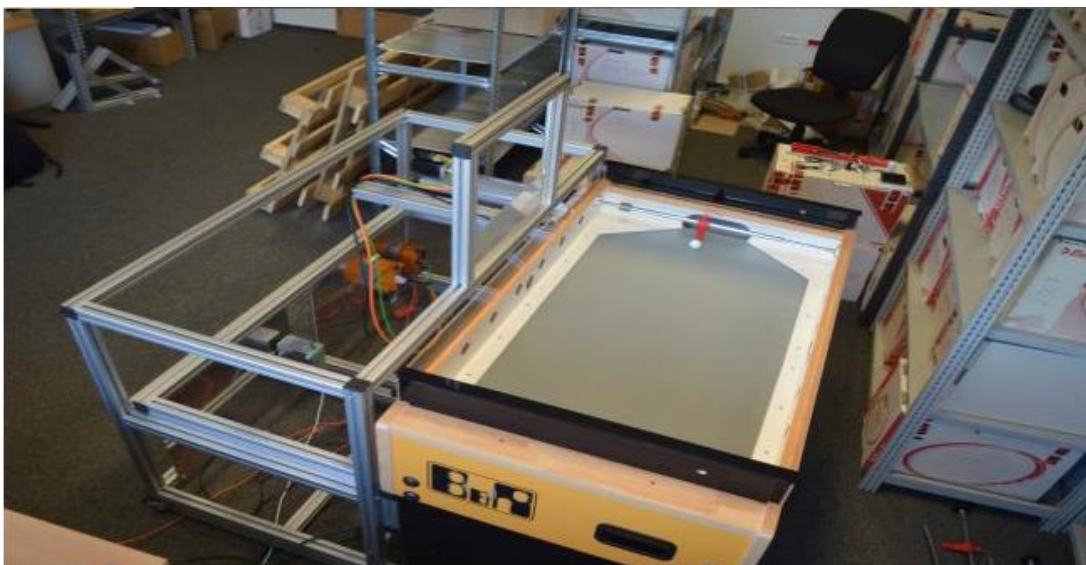
Tab. 1 Špecifikácia hracieho stola [7]

Model	G-500
Dĺžka	143 cm
Šírka	76 cm
Výška	88 cm
Váha	75 kg

Pôvodný povrch hracej plochy bol odstránený a vymenený za dve vrstvy materiálov, z ktorých každá má iný účel a inú priehľadnosť.

Prvá vrstva je vyrobená zo skla, ktoré má matný povrch. Táto vrstva slúži priamo ako hracia plocha, a preto je matný povrch účelný. Matný povrch zaisťuje, že loptička položená na ploche je dobre viditeľná pre kameru, ktorá sa nachádza pod hracou plochou. Taktiež je matná plocha odolnejšia proti škrabancom ako číre sklo. Pre ľudské oko je plocha ideálna, pretože objekty za matným sklom nie sú viditeľné, a preto nie je hráč rozptyľovaný pozadím. Pri samotnej hre na matnej sklenenej ploche sa loptička nadmerne nešmýka po povrchu a ani nie je príliš brzdená, čo sú žiadané vlastnosti.

Druhý povrch je vyrobený z priehľadného číreho plexiskla. Tento povrch sa nachádza pod skleneným materiálom a slúži ako spevnenie pre prvú vrstvu, ktorá je krehká a pri manipulácii so stolom alebo popri hre by mohla prasknúť.



Obr. 5 Pohľad na matnú hraciu plochu [1]

4.2 Kamera

Kameru som obdržal k bakalárskej práci, preto nebudem rozoberať výber kamery. Pre kameru som zadal vyrobiť puzdro, do ktorého som ju vložil. Táto forma je na pohyblivom profile, teda je možné nastaviť jej polohu a taktiež po rozdelení hracieho stola od konštrukcie je možné ju zasunúť do konštrukcie pre lepšiu manipuláciu. Použitá kamera je značky Cognex rady IS 7402. Jedná sa o rýchlu inteligentnú kameru vďaka výkonnému procesoru. Kamera je predovšetkým určená pre priemyselné použitie napríklad na čítanie čiarových kódov, kontrolu kvality výrobkov a podobne. V robotickom futbale je použitá pre rozpoznávanie polohy loptičky na hracej ploche. Princíp vyhľadávania je programovaný v prostredí In-Sight Explorer.

Kamera obsahuje 512 MB flash pamäte pre ukladanie programov, s ktorými bude kamera pracovať. Pre spracovanie obrázkov slúži 256 MB pamäte typu SDRAM. Typ senzoru je 1/1,8-palca CMOS. Rozlíšenie je 1280 x 1024 s veľkosťou pixelu 5.3μm x 5.3 μm. Elektronická uzávierka pracuje v rozsahu od 16 μs do 950 ms. Obsahuje 255 odtieňov šedej farby, teda kamera je čiernobiela. Maximálne snímkovanie je 60 snímkov za sekundu v plnom rozlíšení s minimálnou expozíciou. Komunikácia prebieha pomocou Ethernetu a taktiež je kompatibilná s POWERLINKom. Kamera je napájaná jednosmerným napätím 24 V ± 10%. [8]



Obr. 6 Kamera Cognex s formou a ramenom

Na kamere je použitý objektív Kowa LM3NCM. Tento objektív je určený pre rozmer čipu 1/1.8 palca. Ohnisková vzdialenosť je 3.5 mm. Objektív je vybavený irisovou clonou, pomocou ktorej sa dá meniť množstvo prepusteného svetla do objektívu. Uhol viditeľnosti je pre horizontálny smer 89° a pre vertikálny 73.8°. Keďže poznám všetky podstatné údaje pre výpočet, viem pomocou jednoduchej trigonometrie vypočítať do akej vzdialenosti má byť kamera umiestnená od hracej plochy. [9] [10]

$$h = \frac{d}{2 \cdot \tan\left(\frac{\alpha}{2}\right)} \text{ [mm]} \quad (1)$$

$$h_h \geq \frac{1210}{2 \cdot \tan\left(\frac{89}{2}\right)} \quad h_h \geq 616 \text{ mm} \quad (1.1)$$

$$h_v \geq \frac{703}{2 \cdot \tan\left(\frac{73,8}{2}\right)} \quad h_v \geq 468 \text{ mm} \quad (1.2)$$

Kde:

α = Uhol pozorovania [°]

d = Rozmer hracej plochy [mm]

h_h = Vzdialenosť objektívu pre horizontálny smer [mm]

h_v = Vzdialenosť objektívu pre vertikálny smer [mm]



Obr. 7 Použitý objektív Kowa LM3NCM [11]

Z rovníc 1.1 a 1.2 vyplýva, že kamera musí byť minimálne 616 mm od hracej plochy, aby bola v zornom poli celá hracia plocha.

Pri umiestnení kamery sa ponúkajú dva varianty. Obidva varianty vyžadujú špecifickú úpravu či už hracej plochy, konštrukcie alebo kombinácie oboch variant.

4.2.1 Kamera umiestnená nad hracou plochou

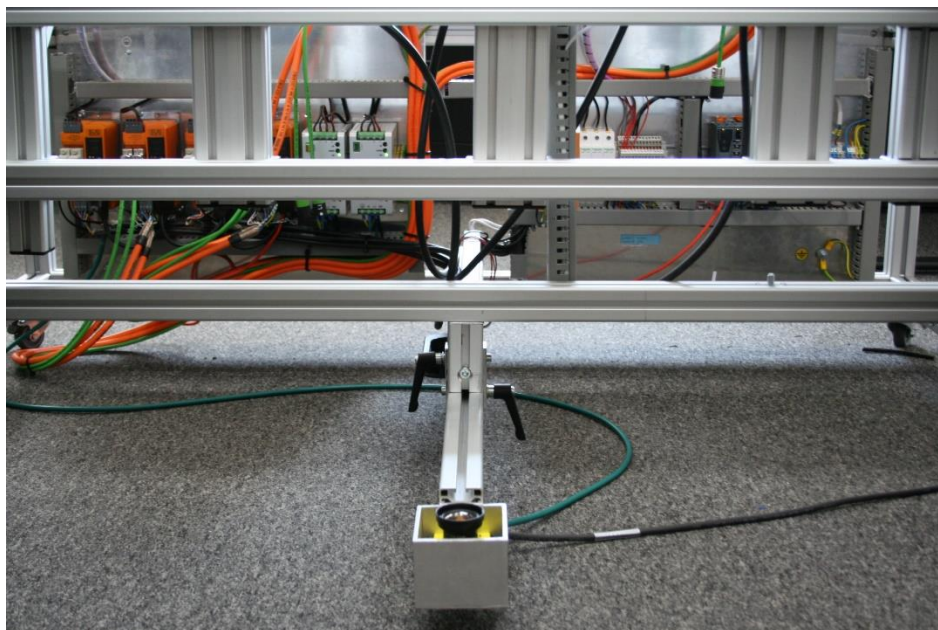
Toto umiestnenie by bolo nevyhnutné, pokiaľ by uhol viditeľnosti bol pre horizontálnu stranu menší ako $87,7^\circ$ a pre vertikálnu stranu $58,3^\circ$ podľa vzorca číslo 1. Vo výpočte som použil maximálnu vzdialenosť 630 mm kamery od hracej plochy. Veľkou výhodou pri tomto umiestnení je, že jediná nutná úprava by bola pridanie konštrukcie pre nastavenie kamery na správnu polohu. Ďalšou výhodou je, že okolité osvetlenie by len napomáhalo kamere detegovať loptičku. No veľkou nevýhodou je, že medzi hraciu plochu a kameru sa môže jednoducho dostať objekt, ktorý znemožní detekciu. Za najväčší problém pokladám situáciu, kedy by bola loptička v zákryte s hráčom. Pokiaľ by takáto situácia nastala, kamera nedokáže detegovať loptičku, a tým pádom počítač nevie reagovať na hru súpera. Taktiež popri hraní sa môže dostať hlava ľudského protivníka alebo pozorovateľa do zorného poľa kamery a opäť počítač nenájde polohu loptičky, pokiaľ bude v danom mieste zákrytu. A v neposlednom rade môže nastať situácia, kedy by ľudskí protivníci spozorovali túto slabinu a začali by podvádzať zámerným zakrývaním zorného poľa, preto som túto možnosť vylúčil.



Obr. 8 Loptička v zákryte s hráčom [12]

4.2.2 Kamera umiestnená pod hracou plochou

Druhá možnosť je kameru umiestniť pod hraciu plochu. Táto možnosť vyžaduje prídavnú konštrukciu pre kameru a taktiež úpravu hracej plochy tak, aby bolo možné detegovať objekt skrz povrch. Výhodou je, že medzi kameru a hraciu plochu sa nedostane objekt, ktorý by znemožnil detekciu. Tomu, aby sa žiadny objekt nedostal na uvedené miesto, zabráni plastová konštrukcia okolo stola. Ďalšia výhoda je, že kamera bude chránená pred vonkajším poškodením, keďže bude uzavretá pod hracím stolom a ľudský protivník nebude mať k nej prístup. Pre detekciu loptičky som zvolil umiestnenie kamery pod hraciu plochu.



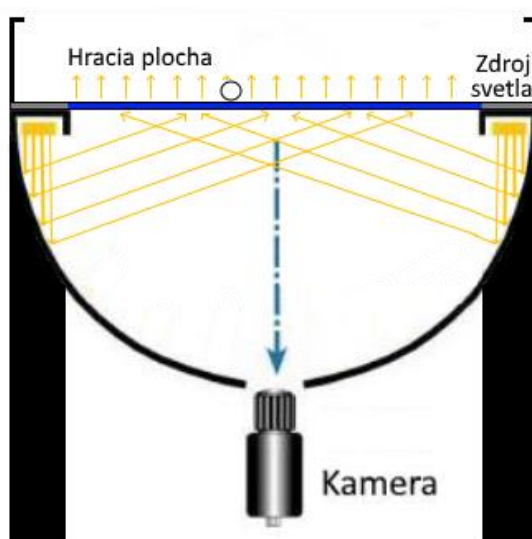
Obr. 9 Umiestnenie kamery pod hracou plochou

4.3 Osvetlenie

Správny výber polohy osvetlenia je veľmi dôležitý v systéme pre detekciu polohy loptičky. Určuje, akým spôsobom bude kamera vyhľadávať a taktiež ako budú okolité podmienky vplývať na detekciu. Podľa polohy osvetlenia sa odvíja aj samotný typ svietidiel.

4.3.1 Poloha osvetlenia

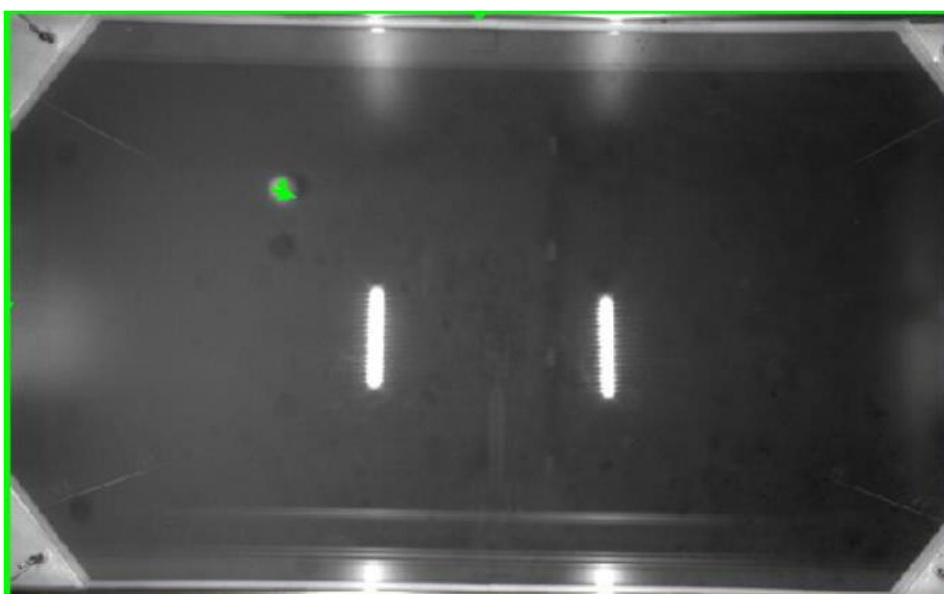
Najjednoduchším spôsobom by bolo umiestniť osvetlenie pod hraciu plochu. Zdroj svetla by bol chránený pred poškodením vďaka plastovej ochrane okolo stola. Uloženie by poskytovalo výhodu, že medzi svetlo a hraciu plochu by sa nedostal cudzí objekt, ktorý by mohla kamera detegovať ako loptičku. Pri umiestnení kamery a osvetlenia pod hraciu plochu je objekt na ploche detegovaný ako biely bod na čiernom pozadí. V analyzovanom obraze neboli hráči príliš viditeľní, a preto by nezasahovali do detekcie. Ale táto poloha mala zásadné nevýhody. Medzi najpodstatnejšie patril problém s výkonom osvetlenia. Za ideálnych podmienok by stačila dvojica 11 wattových svietidiel, no prostredie, v akom bol stôl uložený, veľmi rozhodovalo o viditeľnosti loptičky. Pokiaľ by na plochu svietilo slnečné svetlo poprípade reflektor nad hracou plochou, vznikali by biele miesta, v ktorých by nebolo možné detegovať polohu loptičky. Navyše, vďaka plexisklu vznikol odraz svetla od hracej plochy, a tým aj permanentné miesta, na ktorých bola loptička neviditeľná. Problém s odrazom som sa snažil vyriešiť scénou v tvare kopule. Profesionálne sa nazýva táto metóda „Dome“ a hlavným princípom je odraz svetla od vnútorného profilu ako je na Obr. 10.



Obr. 10 Princíp scény typu "Dome"

V reálnej praxi sa používa tento spôsob pre homogénne osvetlenie plochy. Po vytvorení a otestovaní provizórnej kopule pre stolný futbal som zhodnotil, že na dostatočné osvetlenie tak veľkej plochy by bol nutný veľký počet svetiel. Taktiež tvorba stabilnej verzie kopule by bola veľmi ťažko realizovateľná.

Odraz od plexiskla som sa ďalej pokúšal vyriešiť zmenou polohy osvetlenia, no neexistoval kompromis medzi dostatočným osvetlením plochy a mierou odrazu. Preto som a pokúsil odstrániť odraz fóliami, ktoré som nalepil na plexisklo. Zameral som sa najmä na matné, ktoré dokážu rozptýliť svetlo. Skúšal som niekoľko druhov, ale žiadna neposkytla požadovaný efekt, dokonca niektoré druhy fólie vytvárali ešte väčší odraz. Preto som sa rozhodol úplne zmeniť riešenie a umiestniť osvetlenie nad hraciu plochu.



Obr. 11 Odraz svetla od plexiskla [1]

Osvetlenie uložené nad hracou plochou funguje na opačnom princípe. Pri dopade svetla vzniká tieň objektov na hracej ploche, teda obraz snímaný kamerou pozostáva z bieleho pozadia s tmavými objektami. Nevýhodou je, že taktiež vzniká tieň aj od hráčov, ale tento problém som vyriešil vyhľadávacím algoritmom v kapitole 5.1.2. Ďalšou komplikáciou bol problém akým spôsobom zaistiť stabilnú polohu osvetlenia nad plochou. Preto som navrhol funkčnú konštrukciu, ktorú som neskôr zadal do výroby. Konštrukcia je zobrazená na *Obr. 12*.

Výhodou osvetlenia nad hracou plochou je, že prítomnosť cudzieho osvetlenia len napomáha detekcií. Po vyriešení problémov som sa rozhodol použiť toto riešenie ako konečné.



Obr. 12 Nosná konštrukcia pre osvetlenie

4.3.2 Výber osvetlenia

Po kontaktovaní firmy Smart View som dostal možnosť otestovať pruhové svetlá rôznych výkonov a rôznych vlnových dĺžok. Konkrétne sa jednalo o svetlá rady LL-120, LL-245 a LL-490, ich parametre sú uvedené v tabuľke 2.

Radu pruhových svetiel LL-120 som testoval v bielom svetelnom prevedení a infračervenom prevedení o vlnovej dĺžke 850 nm. Keďže kamera sníma svetlo aj o vlnovej dĺžke 850 nm, použitie infračerveného farebného spektra by bolo ideálne. Infračervené svetlo je na hranici viditeľnosti ľudského oka, preto by hráči neboli rušení osvetlením. Pri testovaní najslabšej rady svetiel som zistil, že výkon je nedostatočný a osvetlenie plochy by vyžadovalo veľké množstvo svetiel.

Ďalšiu testovanú radu LL-245 som testoval len v bielom svetelnom prevedení. Testovaná rada mala dostatočný výkon na osvetlenie stredu ihriska, ale okrajové často boli nedostatočne osvetlené.

Preto som použil najsilnejšiu testovanú radu LL-490. Pruho­vé svetlo je len v bielom svetelnom prevedení. Výkon je dostatočne veľký aj pre osvetlenie okrajových častí. Osvetlenie je ovládané signálovým vodičom určeným pre funkciu impulznej šírko­vej modulácie. Prehľad testovaných svetiel je v tabuľke Tab. 2. [13] [14]

Tab. 2 Parametre pruhového osvetlenia [13] [14]

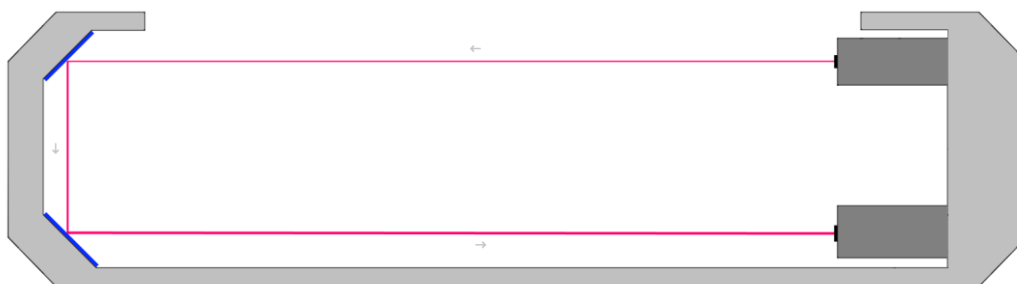
Model	LL-120W120	LL-120IR50	LL-245W	LL-490W
Rozsah napätí	23-28 V	22–28 V	20-28 V	20-28 V
Menovité napätie	24 V	24 V	24 V	24 V
Menovitý prúd	133 mA	150 mA	460 mA	930 mA
Maximálny prúd	150 mA	160 mA	625 mA	1,25 A
Príkon	3,5 W	3.6 W	11 W	22 W
Vlnová dĺžka	-	850 nm	-	-
Farba	Biela	Infračervená	Biela	Biela
Vyžarovací uhol	120°	50°	120°	120°
Dĺžka	134 mm	134 mm	285 mm	530 mm
Šírka	15 mm	15 mm	46 mm	46 mm
Výška	20 mm	20 mm	50 mm	50 mm



Obr. 13 Pruho­vé svetlo LL-245W [14]

4.4 Detekcia gólov

Ako funkčné riešenie som použil návrh Ing. Petra Marečka, ktorý popisuje vo svojej diplomovej práci možnosti detekcie gólu. Do priestoru medzi brámkou a zásobníkom na loptičky som vložil prvok, ktorý dokáže detegovať prechádzajúcu loptičku. Prvok pozostáva z profilu, ktorý je na *Obr. 14*, odrazových zrkadiel a fotoelektrického snímača. Dĺžka profilu je 210 mm a šírka je 75 mm, čo je viac ako dvojnásobok priemeru loptičky. Použitý snímač je typ OESV 19 200 P3K od firmy Di-Sonic a skladá sa z prijímača a vysielača. Výstup je typu PNP a je privedený na digitálny vstup. Elektronika je napájaná z 24 V. [15]



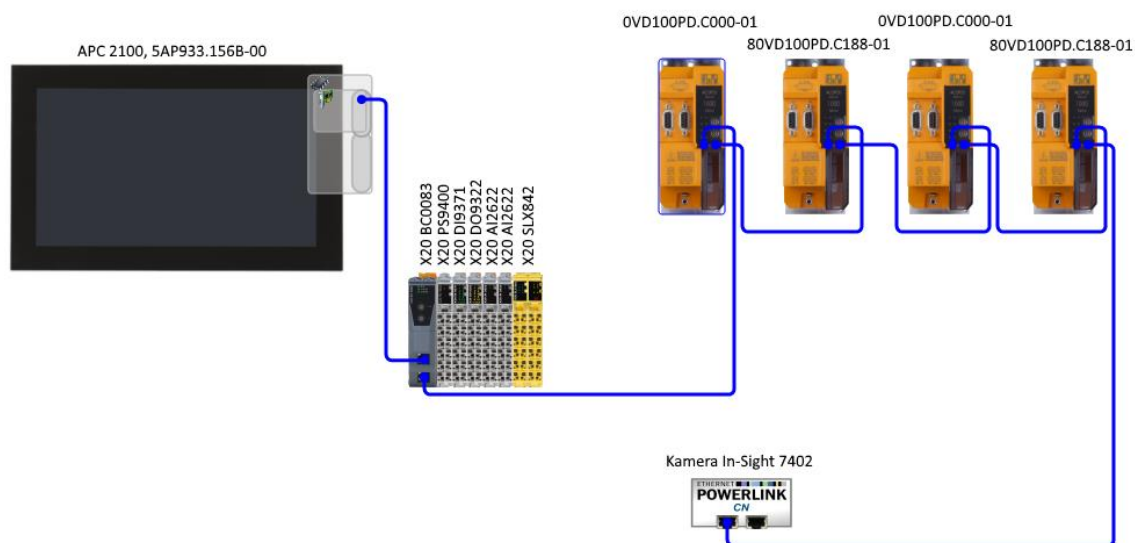
Obr. 14 Prípravok pre snímanie gólov



Obr. 15 Umiestnenie prvku pre snímanie gólov

4.5 Hardwarová konfigurácia

V tejto časti je opísaná hardwarová konfigurácia vytvorená v PLC. Obsahuje použitú výpočtovú jednotku, vstupné, výstupné moduly, bezpečnostné moduly a servo zosilňovače s motormi.



Obr. 16 Hardwarová konfigurácia v PLC

4.5.1 Použité PLC a zobrazovací panel

Ako výpočtová jednotka je použitý Automation PC 2100, ktorý integruje všetky dôležité rozhrania vrátane 2 x Gigabit Ethernet, 1 x USB 2.0 a 1x USB 3.0. Taktiež obsahuje rozhranie pre pridanie modulov. Túto možnosť som využil a pridal som modul pre komunikáciu cez protokol POWERLINK. Pridaný modul má označenie 5ACCIF01.FPSC-000. Samotný procesor v použitom APC je 4-jadrový Intel Atom s označením 5PPC2100_BY44_000. Frekvencia jadra je taktovaná na 1910 Mhz. Použitá pamäť je typu DDR3 SDRAM o veľkosti 4 GB. V systéme je použitá grafická karta Intel HD graphics. K APC je integrované zobrazovacie zariadenie Automation Panel s označením 5AP933.156B-00. Veľkosť panela je 15.6'' s rozlíšením 1366x768 pixelov (16:9). Systém obsahuje operačné systémy Windows 7 a AR Win. Použitá sústava požaduje napájanie 24 VDC $\pm 25\%$. [16]

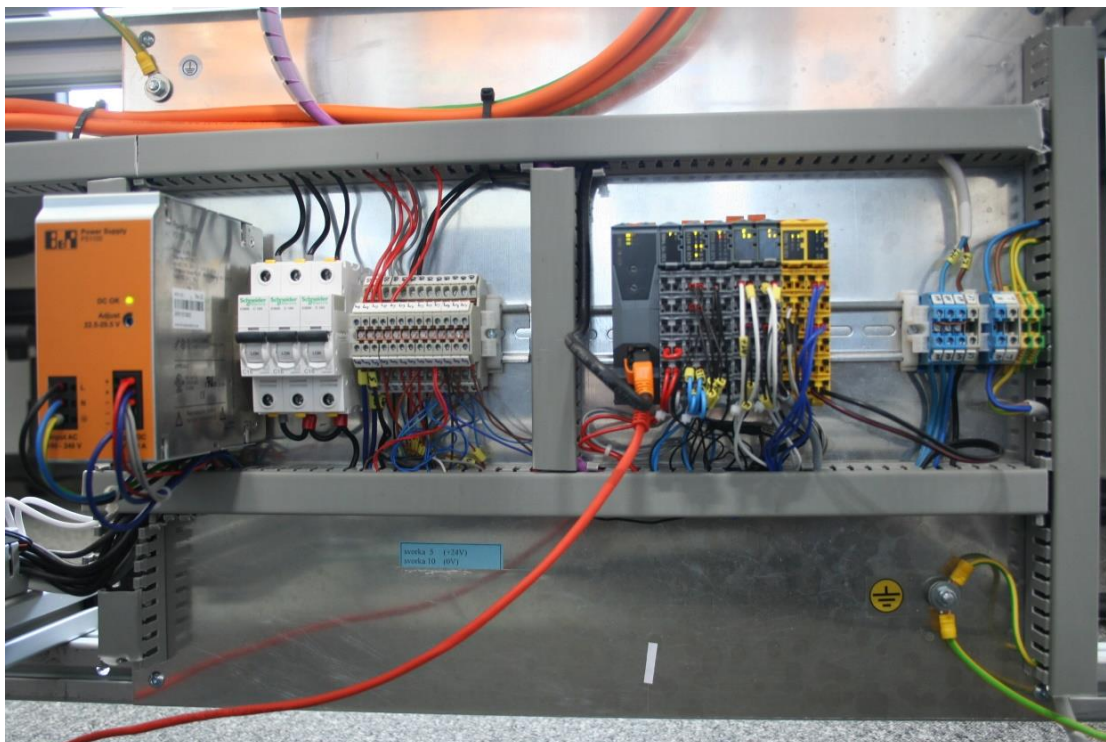


Obr. 17 APC2100 s použitým panelom [16]

4.5.2 Vstupné a výstupné moduly

Keďže APC nemá zabudovanú zbernicu pre komunikáciu pomocou X2X Linku, musel som použiť modul X20BB80. Tento modul je priamo určený pre komunikáciu s modulmi pripojenými cez X2X Link a zároveň s ďalšími systémami pripojenými cez POWERLINK. Vďaka takejto komunikácii môžu byť vstupné a výstupné karty lepšie umiestnené a tým odpadá problém s vedením množstva dlhých vodičov. K X20BB80 sú pripojené nasledujúce moduly:

- X20DI9371 – Jedná sa o digitálny vstup typu „sink“. Teda pri logickej úrovni TRUE je na vstupe napätie 24 V. Vstupný odpor je 6,4 k Ω , čomu odpovedá prúd 3.75 mA. Použitý modul má až 12 digitálnych vstupov. Z toho sú 4 vstupy použité pre snímanie natočenia hráčov pomocou optických snímačov. Ďalšie dva vstupy sú použité pre detekciu gólov pomocou fotoelektrických snímačov, ktorá je popísaná v kapitolách 4.4 a 5.2.4. [17]
- X20DO9322 – Jedná sa o digitálny výstup typu „source“ s úrovňami 0V pri FALSE a +24V pri logickej úrovni TRUE. Výstup môže byť zaťažený prúdom 0.5 A. Modul disponuje dvanástimi výstupmi. V zapojení je použitý jeden výstup ovládanie osvetlenia pomocou PWM. Funkcia ovládania osvetlenia je popísaná v kapitole 5.2.3. [17]
- X20AI2622 – V zapojení sú použité aj dva moduly s analógovými vstupmi, kde každý obsahuje dva vstupy. Vstup funguje pri rozsahu napätí 0-10V alebo v rozsahu prúdu 0-20 mA. V možnostiach karty sa dá zvoliť aj rozsah 4-20 mA. Výsledná hodnota je vyjadrená v dátovom formáte UINT v rozsahu od 0 do 65535. Moduly sú využité pre detekciu polohy hráčov pomocou optických snímačov polohy. [17]
- X20SLX842 – Bezpečnostný modul. Stolný futbal je vybavený svetelnou závorou tak, že keď sa dostane cudzí objekt pod úroveň steny ihriska, okamžite sa zastavia všetky motory. [17]



Obr. 18 Vstupné a výstupné moduly

4.5.3 Menič a lineárne motory

Použitý menič je výhradne určený pre ovládanie lineárnych motorov značky LinMot. Označenie meniča je 80VD100PD.C188-01. Riadiaca elektronika je napájaná z 24 VDC a silová časť je napájaná z 80 VDC. [17]

LinMot je od firmy NTI AG, použitý typ má označenie PS01-23x80/210x270. Lineárny pohyb zabezpečuje iba elektromagnetické pole. Keďže pohyb nie je vykonávaný mechanickými členmi, je veľmi efektívny. Motor sa skladá zo statora, ktorý ma funkciu vinutia a z jadra, ktoré je tvorené neodymovými magnetmi, ktoré sú použité v tyči. [18]

Tab. 3 hodnoty lineárneho motoru [18]

Maximálna sila	44 N
Maximálny prúd	4 A
Maximálna rýchlosť	6 m/s
Priemer hriadeľa	12 mm

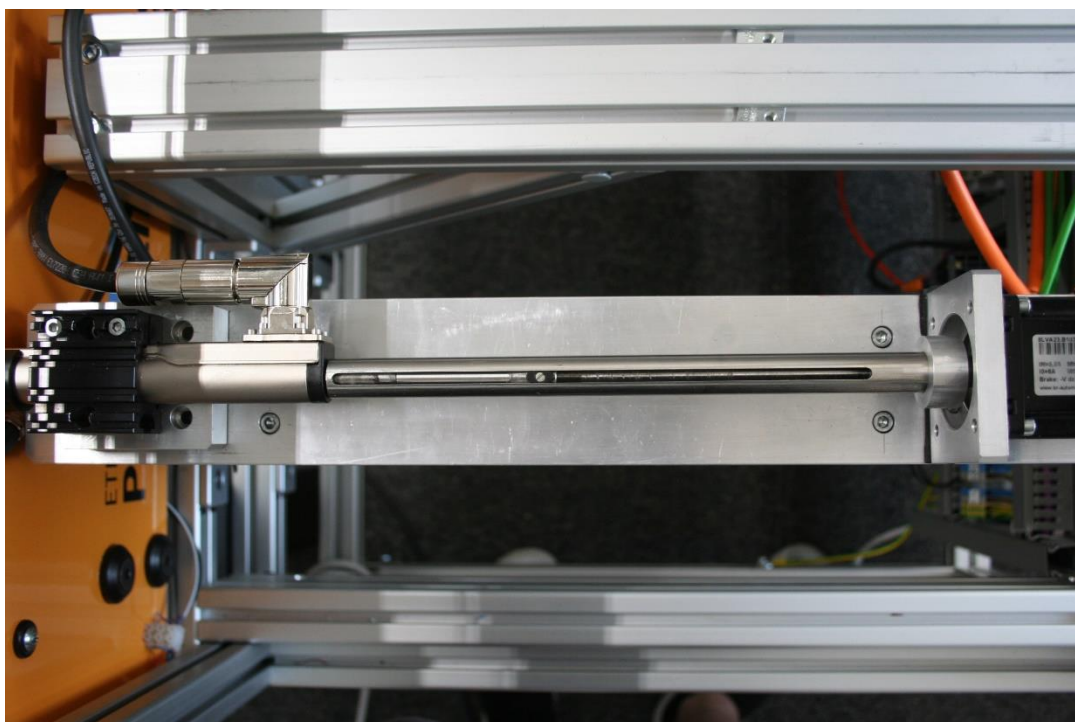
4.5.4 Servo zosilňovač a rotačné krokové motory

Použitý zosilňovač 80VD100PD.C000-01 je určený pre rotačné motory s enkodérom EnDat 2.2 . Elektronika zosilňovača je napájaná z 24 VDC a silová časť z 80 VDC. [17]

Použitý motor pre rotačný pohyb je od firmy B+R, jeho označenie je 8LVA23.B1015D100-0. Jedná sa o synchronný motor, ktorý vykonáva rotačný pohyb na dutom hriadeľi. Do tohto dutého hriadeľa je umiestnený posuvný hriadeľ z lineárneho motora, v ktorom je pero. Pero zaisťuje rotačný pohyb posuvného hriadeľa. [19]

Tab. 4 hodnoty rotačného motoru [19]

Nominálne otáčky	3000 m ⁻¹
Maximálne otáčky	6600 m ⁻¹
Nominálny prúd	5,8 A
Maximálny prúd	20,7 A
Nominálny výkon	408 W



Obr. 19 Konštrukcia k lineárnemu a rotačnému pohybu zároveň

4.6 Komunikácia

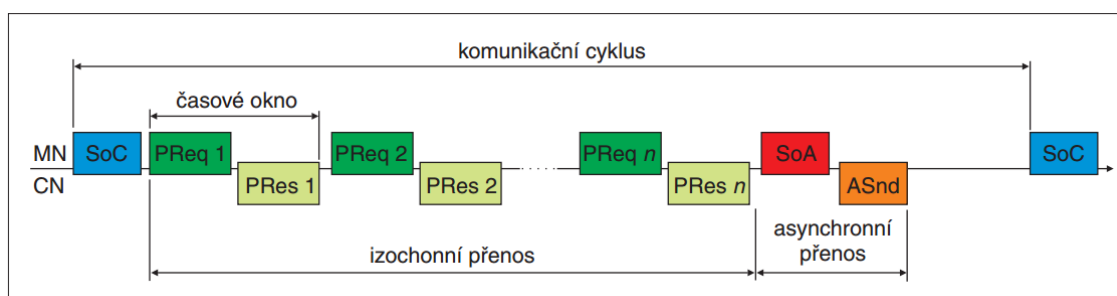
POWERLINK je protokol časovo deterministickej komunikácie, ktorý sa štandardne využíva pri prenose dát v reálnom čase. Táto možnosť sa ponúka vďaka rozdeleniu komunikácie na dve časti:

- Izochrónna komunikácia
- Asynchrónna komunikácia

Každý cyklus v móde POWERLINK začína štartovacou periódou, ktorú vyšle riadiaci uzol s názvom Managing Node. Táto perióda sa nazýva „Start of Cyclic Frame“. Na základe tohto signálu si všetky pripojené zariadenia, nazývané Controlled Nodes, nastaví hodiny reálneho času. Ďalšou fázou je izochrónna komunikácia. V tejto časti funguje komunikácia tak, že MN vyšle požiadavku konkrétnemu CN, pričom túto požiadavku môžu čítať aj ostatné zariadenia. Táto požiadavka sa nazýva Pool Request. Vzápätí na to CN odpovedá vo vyhradenom časovom okne. Odpoveď sa nazýva Pool Response. Potom MN posle požiadavku ďalšiemu zariadeniu. Vďaka tomuto typu komunikácie sa predchádza kolíziám na sieti. Po skončení izochrónnej komunikácie sa ponúka možnosť asynchrónnej komunikácie medzi zariadeniami opäť v obmedzenom časovom okne. Táto fáza sa nazýva „Start of Asynchronous“ a slúži na prenos väčších dát. V sieti môže byť až 240 zariadení, ktoré komunikujú pomocou protokolu POWERLINK. Jednotlivé zariadenia sú navzájom identifikované pomocou „node number“ s označením od 1 do 240. Spravidla má MN číslo 240. [20] [21]

V aktuálnej konfigurácii je Managing Node PLC a CN sú:

- X20 BC0083 s nastaveným node number na 1
- 2x 80VD100PD.C000-01 s node number 2 a 4
- 2x 80VD100PD.C188-01 s node number 3 a 5
- Kamera In-Sight 7402 s node number 6



Obr. 20 Princíp komunikácie POWERLINKu [21]

5 POUŽITÝ SOFTWARE

5.1 In-Sight Explorer

In-Sight Explorer je vytvorený firmou Cognex. Je to univerzálny nástroj pre celú radu kamier In-Sight. Používa sa najmä pre konfiguráciu, ale taktiež sa dá využiť pre diagnostiku alebo jednoduché vyhľadávanie. Tento nástroj ponúka dve vývojové prostredia.

Prvým je EasyBuilder, je to akýsi sprievodca nastavením kamery. Hlavnou prednosťou je prehľadnosť a logické usporiadanie. Pri konfigurácii vás prevedie od pripojenia ku kamere cez konfiguráciu až po samotné testovanie. Postupom času ale začne byť toto vývojové prostredie v niektorých smeroch obmedzujúce, pretože obsahuje len preddefinované okná.

Druhým je SpreadSheet. Toto vývojové prostredie svojou stavbou a funkciou pripomína Microsoft Excel. Všetky nastavenia vykonané v prostredí EasyBuilder sú automaticky generované aj v prostredí SpreadSheet. Toto nemusí byť vždy výhodou, pretože niektoré algoritmy obsahujú funkcie, ktoré sú pre detekciu loptičky nepoužiteľné. Prerobením týchto funkcií sa dá dosiahnuť viac spracovaných obrázkov za sekundu. Rôzne funkcie obsahuje knižnica programu. Tieto funkcie sa dajú využívať v oboch vývojových prostrediach, no v prostredí EasyBuilder je ich oveľa menej.

5.1.1 Parametre snímania a vyhľadávanie

Pri počiatočnom nastavení som využil EasyBuilder. Neskôr som upravil funkcie v prostredí SpreadSheet. Nastavenie je rozdelené do štyroch skupín:

Start obsahuje základné nastavenia kamery. Prvé nastavenie je Get Connected, umožňuje nastaviť IP adresu kamery, masku podsiete a DNS. Po tomto správnom nastavení je možné sa pripojiť na kameru. Druhé je Set Up Image, je jedno z najdôležitejších nastavení, pretože sa ním nastavujú parametre snímania obrazu.

- Trigger – ponúka dve možnosti. Prvou je riadenie spúšte pomocou „Real-time-ethernet“. Celý cyklus získavania polohy by bol riadený cez POWERLINK v cykle Čakať na príkaz->Vyhotovenie a spracovanie snímku->Načítanie do zásobníku->Odoslanie dát. Ďalšou možnosťou je „Continuous“. V tejto možnosti je celá obsluha spúšte a odosielania riadená kamerou, potrebné údaje sa odosielať hneď po spracovaní no automatické spracovanie uberá na výkone kamery. V práci som použil riadenie spúšte druhou spomenutou možnosťou, pretože časový rozdiel nebol badateľný.
- Exposure – je to nastavenie expozície. Hodnota sa môže meniť na základe okolitého osvetlenia, počiatočnú hodnotu som nastavil na 1.3 ms.
- Auto-Exposure – obsahuje možnosť automatického nastavenia expozície. Túto možnosť som nepoužil.

- Start Row – nastavenie bodu, od ktorého sa bude vyhľadávať loptička na osi Y. Nastavil som hodnotu 150 čím som presne nastavil okraj ihriska.
- Number Of Rows – určuje koľko riadkov v pixeloch bude určených pre detekciu. Hodnotu som nastavil na 650.
- Light Control Mode – ovládanie zabudovaného osvetlenia v kamere, keďže osvetlenie je určené skôr na malé vzdialenosti, tak som ho nastavil na hodnotu 0 z rozsahu 0-100.

Set Up Tool je druhá časť konfigurácie kamery. Táto ponuka obsahuje rôzne princípy vyhľadávania objektov a taktiež aj prípadné matematické výpočty. Z ponuky vyhovuje najviac typ „Blob“.

Princíp vyhľadávania „Blob“ spočíva v kontraste čiernych a bielych pixelov. Čiernobiely rozsah kamery taktiež nepredstavuje problém, pretože „Blob“ pracuje iba v tomto rozsahu. Pri vyhľadávaní sú pixely rozdelené do skupiny čiernych a do skupiny bielych bodov podľa parametru threshold. Tento parameter je hranica v rozsahu od 0, čo predstavuje čiernu farbu, do 256, ktorá predstavuje bielu farbu. Pri hodnote napríklad 150 sú všetky hodnoty s vyšším číslom považované za biele body a s nižšou hodnotou za čierne body. Po určení hranice je dôležité stanoviť minimálny a maximálny počet pri sebe ležiacich pixelov v rovnakej farebnej skupine. S tým súvisí aj ďalší parameter, ktorý určuje, či skupina čiernych pixelov leží na bielom pozadí, alebo skupina bielych pixelov leží na čiernom pozadí. Pokiaľ vyhľadávanie nájde viac objektov, výsledkom sú súradnice objektu, ktorý sa najviac blíži k zadaným parametrom. [22]

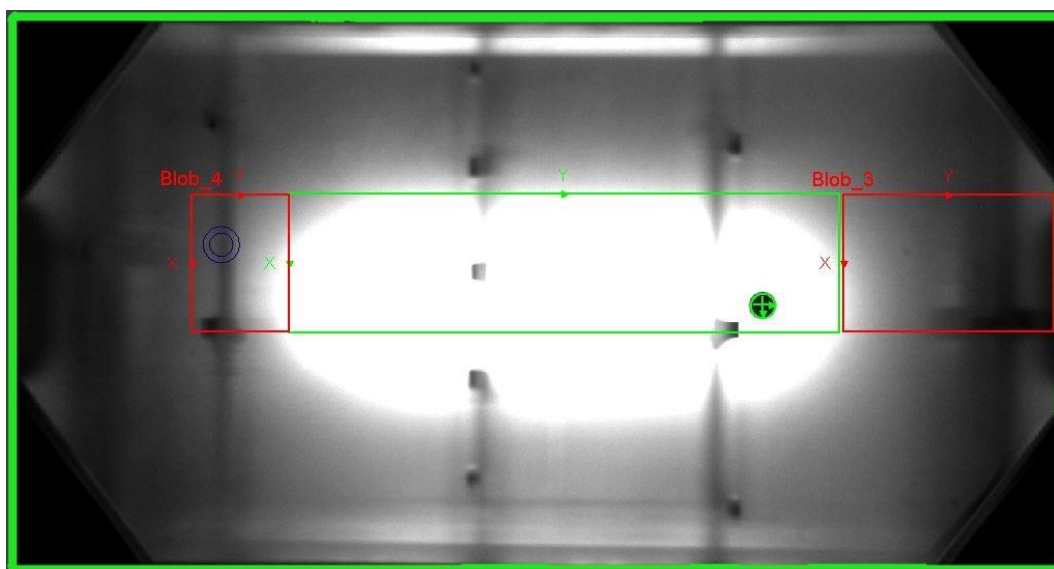
Umiestnením osvetlenia nad hraciu plochu vznikol problém s tieňom hráčov na ihrisku. Pre vyhľadávanie pomocou typu „Blob“ to predstavovalo problém, pretože pri správnom natočení hráčov na určitých miestach tieň splňoval parametre vyhľadávania a tak mohla kamera vyhľadať objekt na ploche, aj keď sa na nej nenachádzala loptička. Dokonca v niektorých prípadoch tieň hráča splňal podmienky lepšie ako loptička, a preto nemohla byť správne vyhladaná. Kvôli výpočtom dráhy letu loptičky je neprípustné, aby bol vyhladaný iný objekt. Preto som musel použiť nástroj pre vyšetrenie nájdeného objektu.

Tu som zvolil možnosť „Presence/absence“, ktorý mi ponúka nástroj pre porovnanie tvaru vyhladaných pixelov, či sa zhodujú s kruhom. Prvým nastavovaním je, v akej veľkej oblasti sa bude porovnávať nájdený objekt. Ďalším je rozdiel kontrastu hrany medzi objektom a pozadím a taktiež, či je objekt čierny na bielom pozadí alebo biely na čiernom pozadí. V ďalšom parametri sa nastavuje, v akom rozsahu sa môže meniť veľkosť skúmaného objektu.

V prípade úspešného vyhľadávania získavam pozíciu objektu. Výsledkom sú súradnice stredu vyhladaného kruhu v pixeloch a taktiež sa navýši celkový počet úspešných hľadání. V prípade neúspešného vyhľadávania je vrátená hodnota „#Err“ a navýšený počet neúspešných hľadání. Kvôli nerovnomernosti osvetlenia a optimalizácií času som musel plochu rozdeliť do viacerých zón, preto sú jednotlivé parametre vyhľadávania uvedené v tabuľke Tab. 6.

5.1.2 Optimalizácia času vyhľadávania

Snahou optimalizovať je dosiahnuť čo najvyšší počet analyzovaných snímkov za sekundu tak, aby pri hľadaní nedochádzalo k chybným výsledkom. Za chybný výsledok sa považuje akýkoľvek iný vyhládaný objekt ako je loptička. Aby som urobil celkový prehľad o rýchlostiach v akých sa pochybuje loptička, vytvoril som testovací algoritmus. K tomuto účelu sa najlepšie hodil test strelbou naprieč celým ihriskom, pretože loptička sa pohybuje veľkou rýchlosťou. Aby som mohol výsledky použiť, vytvoril som testovací algoritmus pre kameru aj pre PLC. Hlavnou funkciou testovacieho algoritmu bolo zaznamenať loptičku minimálne dvakrát tak, aby som vedel určiť rýchlosť pri strele. V PLC som využil mierne modifikovaný funkčný blok popísaný v kapitole 5.2.2. V kamere som vytvoril algoritmus špeciálne určený k testovaniu. Na obrázku *Obr. 21* je grafický náhľad na pásmo určené pre detekciu strely. Pásmo je rozdelené na 3 časti kvôli nerovnomerne osvetlenej ploche. Vyhľadávanie na takejto ploche trvá približne 31 ms.



Obr. 21 Testovací algoritmus kamery s úspešne vyhládanou loptičkou

Výsledkom po strele bolo pole 20 štruktúr. Každá štruktúra obsahovala prepočítanú pozíciu loptičky v osi X a Y v jednotkách unitov (10 unitov = 1mm) podľa kapitoly 5.2.2. Ďalšími hodnotami štruktúry bol čas od posledného úspešného vyhľadania loptičky a rýchlosť strely vypočítaná na základe predchádzajúceho nájdeného bodu. Rýchlosť bola počítaná podľa vzorca 2.

$$v = \frac{\sqrt{\left(\frac{X_1 - X_2}{10}\right)^2 + \left(\frac{Y_1 - Y_2}{10}\right)^2}}{\Delta t} \text{ [m/s]} \quad (2)$$

Kde:

v = rýchlosť loptičky [m/s]

X_1 = súradnica X prvého zaznamenaného bodu [unit]

X_2 = súradnica X druhého zaznamenaného bodu [unit]

Y_1 = súradnica Y prvého zaznamenaného bodu [unit]
 Y_2 = súradnica Y druhého zaznamenaného bodu [unit]
 Δt = čas medzi prvým a druhým zaznamenaným bodom [ms]

Z opakovanej streľby som určil, že priemerná rýchlosť loptičky pri streľbe je približne 13.8 m/s, čo je v prepočte asi 50 km/h. Maximálna dosiahnutá rýchlosť bola 21.5 m/s, čo je v prepočte 77.3 km/h. Pre ďalšiu prácu som vytvoril orientačnú tabuľku *Tab. 5* pre odhad doby letu loptičky z rôznych pozícií. Čas je počítaný podľa vzorca 3 pre ideálnu strelu, ktorej sa nemenia hodnoty na osi Y.

$$t = \frac{X_1 - X_2}{v_p} = \frac{X_1 - 80}{13,8} [ms] \quad (3)$$

Kde:

X_1 = pozícia strelca na osi X [mm]
 X_2 = pozícia brankára robotického futbalu na osi X [mm]
 v_p = priemerná rýchlosť loptičky [mm/ms]
 t = doba letu loptičky z pozície X_1 do X_2 [ms]

Tab. 5 Doba letu loptičky podľa pozície strelca

Pozícia	Súradnica na osi X	Doba letu
Brankár	1130 mm	76.1 ms
Obranca	980 mm	65.2 ms
Stred	680 mm	43.5 ms
Útok	380 mm	21.7 ms

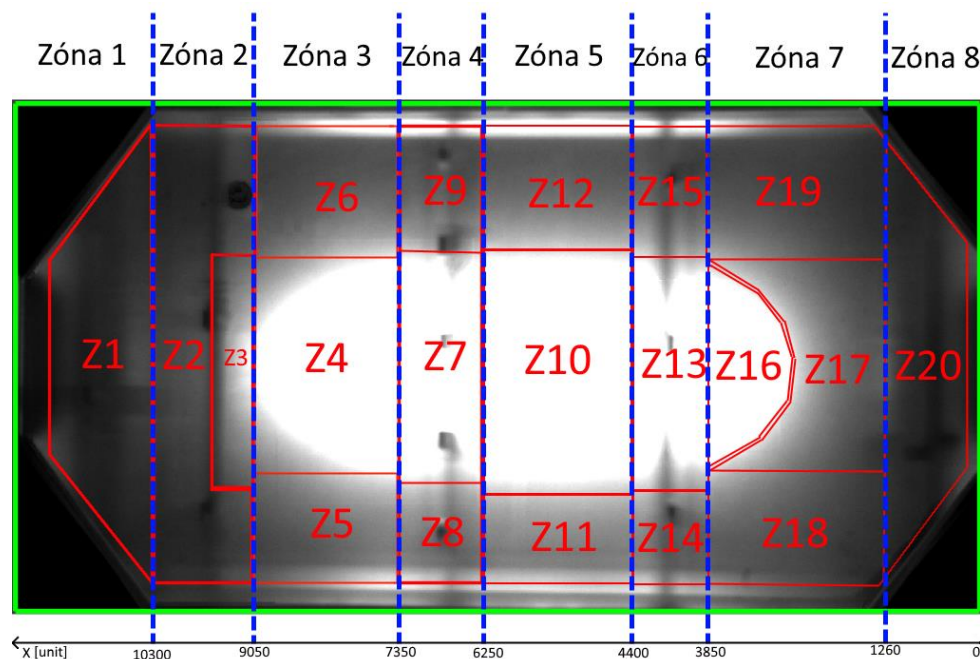
V ďalšom kroku som potreboval zistiť, aký dlhý čas trvá vyhľadať loptičku na celej ploche. Použil som jednu oblasť vyhľadávania typu „Blob“ na celú plochu. Vyhľadávanie trvalo priemerne 70 ms. Podľa *Tab. 5* môžem zhodnotiť, že čas vyhľadávania je príliš dlhý pre situácie, kedy by ľudský protivník strelal z pozície obrancu, stredu alebo z útoku. Taktiež vznikol problém, že je nemožné nastaviť rovnaké parametre vyhľadávania na nerovnomerne osvetlenej ploche. Preto som musel navrhnúť algoritmus tak, aby sa zvýšil počet snímkov za sekundu a zároveň vyriešil problém s parametrami vyhľadávania.

Počet analyzovaných snímkov najviac ovplyvňuje výpočtový výkon kamery, zložitosť vyhľadávacieho algoritmu a veľkosť analyzovanej plochy. Výpočtový výkon kamery nedokážem ovplyvniť, ale zvyšné parametre áno.

Keďže pri použití prostredia EasyBuilder je kód generovaný automaticky a vznikajú aj nepotrebné funkcie, je možné ich upraviť alebo zmazať. Taktiež je možné upraviť alebo zmazať preddefinované systémové funkcie. Samotný cyklus kamery bez vyhľadávania objektu, ale so základnými systémovými funkciami, trvá priemerne 20 ms. Po odstránení

nepotrebných systémom vygenerovaných funkcií som dokázal priemerný čas cyklu znížiť na 18 ms.

Počet analyzovaných obrázkov sa teda dá najviac meniť v závislosti na veľkosti analyzovanej plochy. Keďže nemôžem vynechať nejakú časť plochy pri vyhľadávaní, rozhodol som sa rozložiť analyzovanú plochu na viac častí a jednotlivé časti spúšťať pomocou PLC. Zóny sú zobrazené na Obr. 22.



Obr. 22 Rozloženie zón

Tab. 6 Zóny a parametre vyhľadávania

			Blob			Circle	
	Osa X [unit]	Plocha a	Threshold [-]	Min. plocha [pixel]	Max. plocha [pixel]	Kontrast [-]	Šírka okraja [pixel]
Zóna 1	> 10300	Z1	14	500	800	29	6
Zóna 2	> 9050	Z2	25	500	800	20	4
		Z3	45	500	1000	17	3
Zóna 3	> 7350	Z4	70	500	1000	40	3
		Z5	40	500	800	25	3
		Z6	38	500	800	23	3
Zóna 4	> 6250	Z7	105	500	1000	31	3
		Z8	50	500	800	15	3
		Z9	50	500	800	15	3
Zóna 5	> 4400	Z10	130	500	1000	40	3
		Z11	52	500	800	14	3
		Z12	53	500	800	15	3

Zóna 6	> 3850	Z13	100	500	1000	45	4
		Z14	50	500	800	13	3
		Z15	50	500	800	18	3
Zóna 7	> 1260	Z16	60	500	800	45	3
		Z17	25	500	800	20	3
		Z18	30	500	800	18	4
		Z19	30	500	800	18	4
Zóna 8	> 0	Z20	12	500	800	15	3

Keďže na spínanie zóny stačia logické úrovne 0 a 1, rozhodol som sa použiť pre prenos informácie jednu premennú o veľkosti UINT 32. Táto premenná obsahuje až 32 bitov, takže po vymaskovaní potrebného bitu môžem použiť túto informáciu pre spínanie zóny. Výhodou takejto premennej je, že môžem spúšťať akúkoľvek kombináciu zón.

Tab. 7 Vstupné hodnoty k spúšťaniu zón

Zóna	Spúšťací bit
Zóna 1	00000001
Zóna 2	00000010
Zóna 3	00000100
Zóna 4	00001000
Zóna 5	00010000
Zóna 6	00100000
Zóna 7	01000000
Zóna 8	10000000

Každá plocha označená písmenom „Z“ je v podstate jeden vyhľadávací prvok. Má vlastné nastavenia, vlastné vstupy aj výstupy. V každej ploche je podstatný výstup s informáciou o tom, či bol hľadaný objekt nájdený alebo nie a k tomu príslušné súradnice objektu. Pokiaľ konkrétna plocha našla objekt, výstupom je informácia o úspešnom hľadaní a súradnice objektu v jednotkách pixelov. V inom prípade, keď sa hľadaný objekt nenašiel, výstupom je informácia o neúspešnom hľadaní a na mieste súradníc je hodnota #Err. V poslednom prípade, keď vyhľadávanie v danej ploche nie je povolené, je výstupom na mieste súradníc #Err a v informácií o stave hľadania je permanentne hodnota 0. Algoritmus zjednocuje plochy do zón ako je znázornené na Obr. 22. Pri spájaní do zón vzniká problém s výstupom, ako zjednotiť informácie zo všetkých použitých plôch do jednej premennej, ktorá sa bude odosielať do PLC. Pre zjednotenie informácie o hľadaní som použil súčet informácií o stave hľadania. Pokiaľ sa loptička nenájde v žiadnej ploche, hodnota zjednotenej premennej je 0. Pokiaľ sa loptička nájde práve v jednej zóne, výstupom je 1 a pokiaľ sa objekt nájde vo viacerých zónach, hodnota premennej je vyššia ako 1. Výsledok súčtu sa výborne hodí pre posúdenie, či bola loptička nájdená a či nebolo nájdených viac objektov. Pre súradnice sa nedá použiť rovnaký princíp, pretože pokiaľ by

sa sčítali dve súradnice, stále by z nich mohla vzniknúť jedna nepravdivá, ale zmysluplná hodnota. Preto som využil výsledok súčtu či bola loptička nájdená a použil ho ako podmienku či sú súradnice správne sčítané. Pokiaľ podmienka neplatí, výstupom súradníc je 0; 0. Rozdelením na plochy sa priemerný čas vyhľadávania znížil na 26 ms až 40 ms v závislosti od zóny. Vyhľadávanie na celej ploche trvá priemerne 81 ms.

5.1.3 Komunikácia s kamerou v prostredí In-Sight Explorer

V tejto časti sa nastavuje vstup, výstup a taktiež spôsob komunikácie. K nastaveniu komunikácie som využil prostredie SpreadSheet. Ako prípravu pre odoslanie údajov je potrebné použiť funkciu FormatOutputBuffer. Táto funkcia konvertuje údaje do správneho formátu. Prvý parameter vyžaduje označenie hodnoty, ktorá má byť odoslaná, ďalším parametrom je výstupný formát. Keďže vstup v Automation Studio požaduje dátový typ UDINT, zvolil som jediný kompatibilný formát, ktorý mi ponúkla funkcia, teda 32-bit integer. Po konvertovaní som použil funkciu WriteResultsBuffer, ktorá zaistí odosielanie výstupnej hodnoty. Odosielanie prebieha vždy v okamžiku spracovania obrázku. Jediný parameter, ktorý funkcia vyžaduje, je zásobník, ktorý sa bude odosielať, tento zásobník som vytvoril funkciou FormatOutputBuffer.

K prijatiu dát z PLC je podobne, ako pri odosielaní, potrebné nastaviť vstupný formát, a preto som použil funkciu FormatInputBuffer. Parametrom je vstupný formát, ktorý je opäť UDINT, a preto som nastavil formát pre funkciu 32-bit integer. Pre prijatie som použil funkciu ReadUserDataBuffer, ako zásobník pre funkciu som použil FormatInputBuffer.

Aby kamera dokázala komunikovať cez POWERLINK, bolo nutné nastaviť v prostredí EasyBuilder v sekcii komunikácie POWERLINK a k nemu prislúchajúci node number, ktorý som zhodne nastavil v Automation studio na hodnotu 6.

5.2 Automation Studio

Automation Studio je programovací nástroj vytvorený firmou B+R. Tento nástroj obsahuje tri hlavné časti, ktorými sa programujú, ovládajú a diagnostikujú hardwarové komponenty od firmy B+R.

Prvou časťou je Logical View, v tejto časti je možné vytvárať a editovať zdrojové súbory v rôznych jazykoch, vytvárať vizualizáciu, pridávať a meniť lokálne a globálne premenné. Taktiež je možné vytvárať nové dátové typy.

Druhou časťou je Configuration View, umožňuje nastavovanie hardwarovej konfigurácie, súbory pre softwarovú konfiguráciu, vstupno-výstupné mapovacie tabuľky, nastavenia pohonov a tak ďalej. Veľmi prínosnou funkciou je možnosť prepínať medzi konfiguráciami, no editovať sa dá len aktuálne použitá konfigurácia.

Tretou a poslednou časťou je Physical View, tu sa vytvára a edituje aktuálna hardwarová konfigurácia. Hardware je prehľadne a logicky usporiadaný v štruktúre podľa vstupov a výstupov. Navyše obsahuje Systémového dizajnéra, v ktorom je graficky zobrazená celá konfigurácia, čo veľmi pridáva k prehľadnosti.

5.2.1 Programovacie jazyky

V Automation Studio je možné programovať v grafických a textových jazykoch. Jazyky sú taktiež rozdelené podľa normy IEC61131:

Tab. 8 Zoznam programovacích jazykov v prostredí Automation Studio

Programovací jazyk	IEC61131	Komentár
Ladder Diagram (LD)	Áno	Grafický
Function Block Diagram (FBD)	Áno	Grafický
Continuous Function Chart (CFC)	Nie	Grafický
Sequential Function Chart (SFC)	Áno	Grafický a textový
Instruction List (IL)	Áno	Textový
Structured Text (ST)	Áno	Textový
ANSI C a C++	Nie	Textový

Pri tvorbe projektu je možné jednotlivé jazyky kombinovať v rámci cyklických tried, knižníc, funkcií a podobne. Pri písaní kódu som používal iba jazyk ANSI C.

Pomocou cyklických tried sa nastavuje perióda, s akou sa bude vykonávať daný program, taktiež sa nastavuje tolerancia, aké môže mať cyklická trieda oneskorenie. Napríklad pre prácu s motormi alebo s bezpečnostnými prvkami sa využíva nulová tolerancia. Celkovo je 8 cyklických tried, kde zväčša prvá cyklická trieda má najrýchlejší cyklus a s každou ďalšou sa zväčšuje perióda. Výnimkou je posledná ôsma cyklická trieda, tej perióda býva nastavená porovnateľne s prvou alebo druhou triedou, no tolerancia býva nastavená na rádovo sekundy až desiatky sekúnd. To z toho dôvodu, že táto trieda sa používa predovšetkým pre programy, ktoré nevyžadujú tak prísnu toleranciu, napríklad pomocné programy k vizualizáciám.

5.2.2 Funkcia pre spracovanie dát z kamery

Ako prvé je nutné nastaviť komunikáciu s kamerou na strane Automation Studio. Keďže kamera In-Sight 7402 nie je v hardwarovej databáze B+R, je nutné pridať popisný súbor vo formáte .XDD. Po inštalácii softwaru na spracovanie obrazu In-Sight Explorer je možné tento súbor nájsť priamo v nainštalovanej štruktúre a tento súbor je univerzálny pre celú radu kamier In-Sight 7000. Po načítaní sa kamera už nachádza vo výbere hardwaru. Po pripojení kamery na POWERLINK je potrebné nastaviť node number v projekte Automation Studio aj v nastaveniach kamery na rovnaké číslo. V našom prípade som nastavil node number na 6. Správne nadviazanie komunikácie je možné overiť v I/O mapovanej tabuľke kamery pomocou kanálu „ModuleOk“. V konfigurácii kamery treba nastaviť 4 výstupné kanály s názvom InspectionResultsData_I2011 na hodnotu „Read“ a jeden vstupný s názvom UserData_I2021 na hodnotu „write“.

K optimálnemu interpretovaniu výsledkov hľadania som použil funkčný blok so štyrmi vstupmi a siedmymi výstupmi. Vstupy sú:

- Enable – vstup je dátového typu BOOL. Privedením signálu s logickou úrovňou TRUE sa funkčný blok aktivuje.
- InSight – vstup je dátového typu štruktúra s názvom InSight_type. Názov štruktúry je odvodený od názvu kamery pre ľahšie mapovanie k hodnotám z kamery. V štruktúre sa nachádzajú premenné:
 - ModuleOk – indikuje stav kamery logickou úrovňou TRUE alebo FALSE.
 - Control_I2000_S01 – ovládanie kamery bitovým poľom s informáciou uloženou v dátovom type UDINT.
 - Status_I2001_S01 – informuje o stave kamery a taktiež slúži ako spätná väzba pri ovládaní kamery. Informácia je uložená v dátovom poli typu UDINT.
 - InspectionResults_I2011_S01 – výstup hľadania loptičky v súradnici X. Všetky premenné s označením InspectionResults sú v dátovom type UDINT.
 - InspectionResults_I2011_S02 – premenná určená k ukladaniu súradnice Y.
 - InspectionResults_I2011_S03 – do premennej je ukladávaný počet úspešných hľadání kamerou.
 - InspectionResults_I2011_S04 – do premennej je ukladávaný počet neúspešných hľadání kamerou.
 - UserData_I2021_S01 – informácia odosielaná do kamery, ktorou sa určuje aktuálna plocha vyhľadávania.
- ErrorReset – vstupná premenná dátového typu BOOL. Slúži k resetovaniu celého funkčného bloku a taktiež kamery, pokiaľ dôjde ku chybe.
- Search – vstupná premenná dátového typu BOOL. Slúži k spusteniu vyhľadávania.

Výstupom funkčného bloku sú premenné:

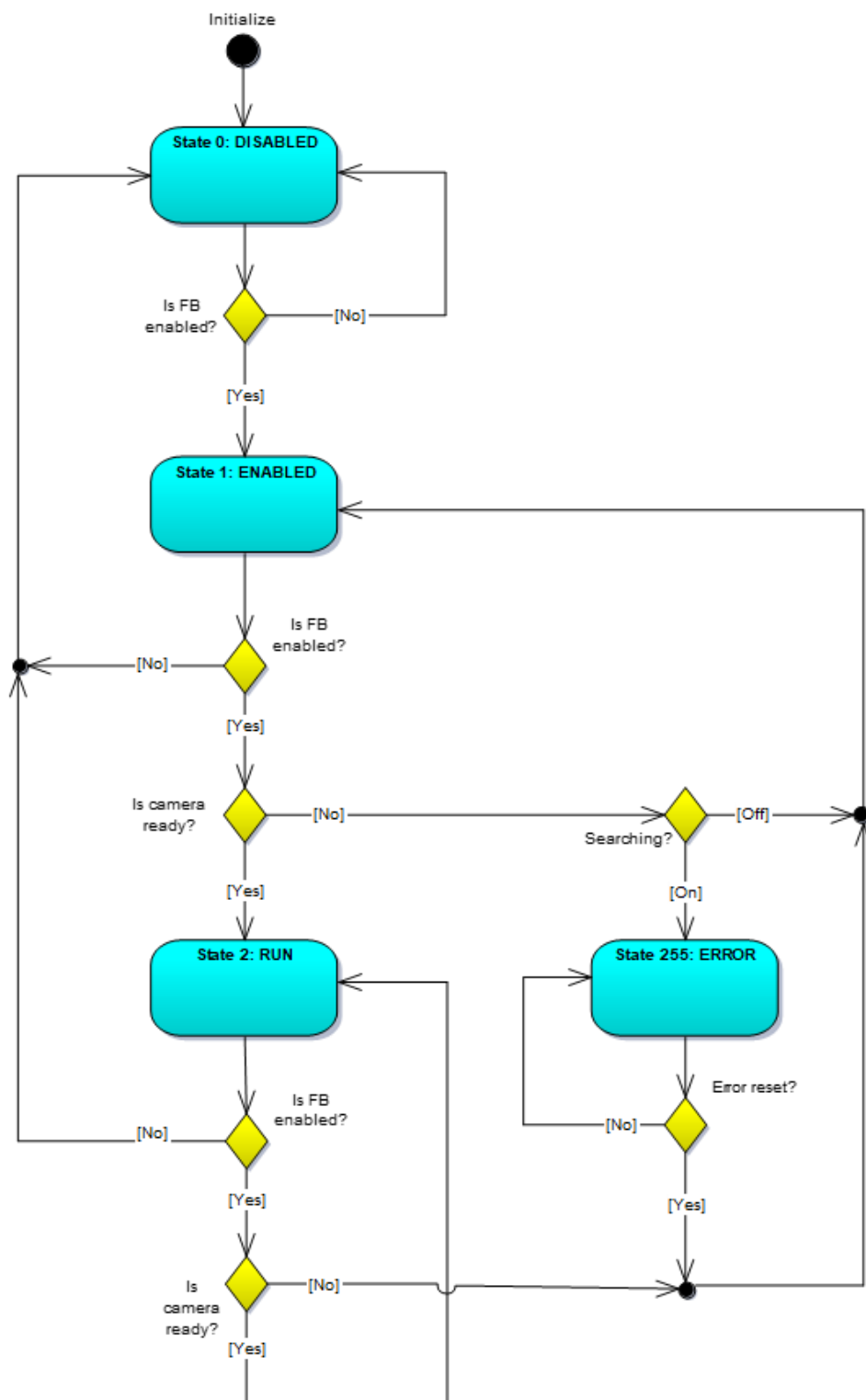
- Active – výstupom je informácia v dátovom type BOOL o tom, či je funkčný blok aktívny.
- Error – výstupom je hodnota typu BOOL. Ak došlo ku chybe vo funkčnom bloku, výstupom je hodnota TRUE.
- Status – výstupom je hodnota typu UDINT. Informuje o prípadnej chybe.
- isCameraReady – výstupom je hodnota typu BOOL. Keďže čas zapnutia kamery je dlhší ako PLC, slúži k indikovaniu, kedy je kamera pripravená k vyhľadávaniu.
- isBallFound – výstupom je hodnota typu BOOL. Po úspešnom vyhľadaní je hodnota TRUE, ale iba jeden cyklus.

- Results – výstupom je štruktúra typu Results_type. Obsahuje spracované výsledky o hľadaní loptičky. V štruktúre sa nachádzajú premenné AxisX, AxisY, ActTime, AxisXOld, AxisYOld, ActTimeOld a TimeDiff_ms.

Funkčný blok sa rozdeľuje na štyri hlavné stavy: DISABLED, ENABLED, RUN a ERROR ktoré sú znázornené v Obr. 23. Ihneď v prvom cykle volania je funkčný blok v stave DISABLE a je volaná inicializačná funkcia `cleanFB(inst)`, ktorá nastaví počiatočné hodnoty. Pokiaľ užívateľ neprivedie logickú úroveň TRUE na vstupnú hodnotu „Enable“, je výstupom Status 65534, čo znamená, že funkčný blok nie je aktívny, ale je volaný. Ak je Premenná „Enable“ nastavená na TRUE, mení sa stav na ENABLED.

V tomto stave je na výstupný bit „Active“ nastavená hodnota TRUE, ktorá značí, že funkčný blok je aktívny. Tento stav je určený k čakaniu, pokiaľ bude kamera schopná detekcie alebo v prípade, keď je kamera neočakávane odpojená. Informácia o stave je vyčítaná z premennej „ModuleOk“. Pokiaľ má premenná hodnotu FALSE, je aktualizovaný výstupný bit „isCameraActive“ na hodnotu FALSE. Ak užívateľ privedie na vstupnú premennú „Search“ logickú úroveň TRUE, zatiaľ čo nie je kamera pripravená, zavolá sa funkcia `setError(inst, ERR_CAM_NOTFOUND)`, ktorá zmení stav na ERROR a nastaví informáciu o chybe 65533. Ak má premenná „ModuleOk“ hodnotu TRUE, mení sa stav na RUN a výstupná premenná „isCameraReady“ taktiež na TRUE.

V stave RUN je vždy kontrolovaná podmienka či je kamera v poriadku. Pokiaľ kamera nie je v poriadku, mení sa stav na ENABLED. Inak je kontrolovaná podmienka, ako je nastavený vstupný bit „Search“. Ak má hodnotu FALSE, je ovládacie bitové pole `Control_I2000_S01` logicky sčítané s decimálnou hodnotou 128. Tento úkon zabezpečí, že kamera nebude vyhľadávať, no bude aktívna. Výstupný bit „isSearching“ je nastavený na hodnotu FALSE, ak je vyhľadávanie neaktívne. Ak je vyhľadávanie aktívne, premenná „isSearching“ je nastavená na hodnotu TRUE. Kamera musí fyzicky vyhľadávať, a preto sa pomocou logického súčinu zjednocuje premenná `Control_I2000_S01` a číslo FFFFFFFF v hexadecimálnej sústave.



Obr. 23 Vývojový diagram hlavných stavov funkčného bloku Camera

Ďalej je volaný stavový automat pre voľbu vyhľadávacej zóny, ktorý obsahuje funkciu getRealAxes. Tá sa stará o spracovanie údajov z kamery. Keďže funkčný blok je použitý v diplomovej práci Bc. Romana Paráka, sú kladené nasledovné požiadavky:

- Nájdené súradnice musia byť v jednotkách unitov (1mm = 10 unitov).
- Ak nebola poloha nájdená, tak premenné AxisX a AxisY obsahujú hodnotu NaN. Premenné AxisXOld a AxisYOld obsahujú poslednú nájdenú pozíciu loptičky a premenná TimeDiff_ms má hodnotu 0.
- Ak sa poloha loptičky nezmenila, AxisX sa bude rovnat' AxisXOld a AxisY sa bude rovnat' AxisYOld. V premennej TimeDiff_ms bude uložený čas hľadania v milisekundách.
- Ak bola poloha loptičky nájdená, v premenných AxisX a AxisY sa uloží aktuálna hodnota. Do premenných AxisXOld a AxisYOld sa uloží posledná predchádzajúca hodnota. V premennej TimeDiff_ms je uložený čas hľadania.

Funkcia je primárne riadená počtom úspešných aj neúspešných hľadání. V prípade že sa zmení premenná InspectionResults_I2011_S03, nájdené súradnice sú aktuálne. Ďalej ich treba prepočítať z jednotiek pixelov na jednotky unitov. K prepočítaniu som potreboval určiť koeficienty k a q z rovnice priamky. Súradnice v unitoch som určil na základe známej pozície hráčov. Súradnice v pixeloch som odčítal priamo z kamery.

Výpočet pre osu X:

$$\begin{aligned} b &= a \cdot k + q \\ b_0 &= a_0 \cdot k + q \end{aligned} \quad (3)$$

$$k = \frac{b - b_0}{a - b_0} = \frac{800 - 11300}{1215 - 80} = -9,251 \quad (3.1)$$

$$q = b - (a \cdot k) = 800 - (1215 \cdot (-9,251)) = 12039,965 \quad (3.2)$$

Výpočet pre osu Y:

$$k = \frac{b - b_0}{a - b_0} = \frac{820 - (-920)}{414 - 604} = -9,158 \quad (3.3)$$

$$q = b - (a \cdot k) = 820 - (414 \cdot (-9,158)) = 4611,412 \quad (3.4)$$

Kde:

b_1 = hodnota prvej súradnice [unit]

b_2 = hodnota druhej súradnice [unit]

a_1 = hodnota prvej súradnice [pixel]

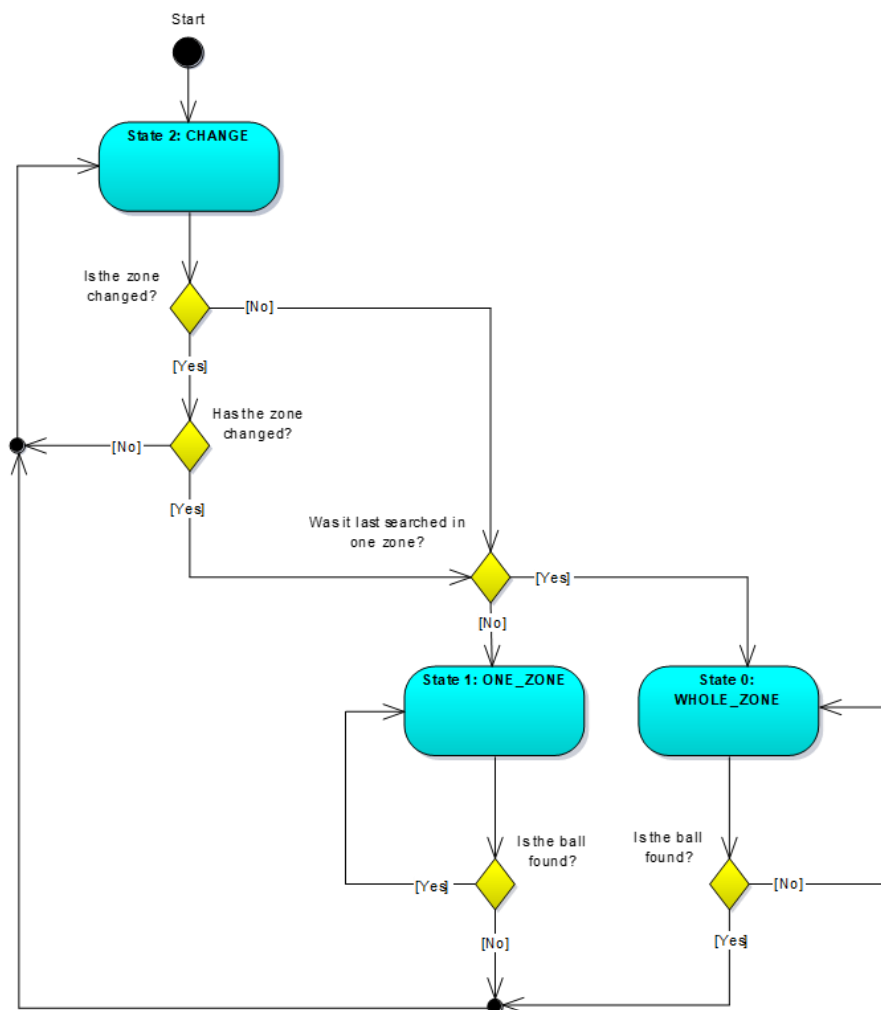
a_2 = hodnota druhej súradnice [pixel]

k = smernica priamky

q = priesečník priamky

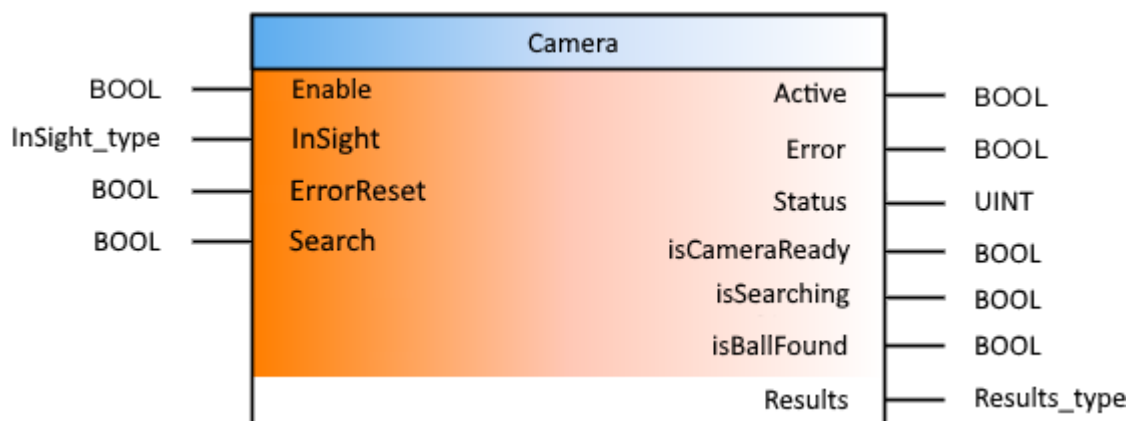
Po prepočítaní súradníc z pixelov na unity sú výsledky kontrolované, či nepresiahli maximálne a minimálne rozsahy hracej plochy. Pokiaľ je aspoň jedna súradnica mimo rozsahu, výsledky nie sú interpretované a je vyhlásený Status 61125. Ak sú súradnice v rozsahu je vždy výsledkom čas hľadania v milisekundách a premenná „isBallFound“ má hodnotu TRUE. Ďalej program pokračuje podmienkou že, pokiaľ je na mieste aspoň jednej aktuálnej súradnice hodnota NaN, je premenná AxisX a AxisY naplnená aktuálnymi hodnotami. V opačnom prípade sú aktuálne súradnice kontrolované, či sa nezhodujú predchádzajúcimi. Samozrejme, súradnica sa môže líšiť v jednotkách unitov aj pokiaľ sa poloha nezmenila, a preto som zo vzorcov 3.1 a 3.3 vypočítal, že najmenšia možná zmena je 9.251 unitov. Z výsledku som teda určil, že pokiaľ je rozdiel starej aktuálnej hodnoty a najnovšej hodnoty menší ako dvojnásobok minimálneho rozdielu, loptička je nehybná. Podmienka musí platiť pre obe súradnice zároveň. Tento rozdiel však môže vznikať aj v prípade, že loptička je na rozmedzí dvoch hodnôt. Preto som zaviedol aproximáciu priemerom na základe poslednej a aktuálnej hodnoty. Potom sú výsledné súradnice AxisX, AxisXOld interpretované s rovnakou hodnotou. Taktiež aj súradnice AxisY a AxisYOld majú rovnakú hodnotu. Ak súradnice nespĺňajú podmienku o tom či sú nehybné, sú výsledky z AxisX a AxisY kopírované do AxisXOld a AxisYOld. Potom sú najaktuálnejšie výsledky zapisované do AxisX a AxisY. V prípade, že sa zmení hodnota premennej InspectionResults_I2011_S04, znamená to, že loptička nebola nájdená. V tom prípade sa testuje, či premenné AxisX alebo AxisY neobsahujú hodnotu NaN. Ak neobsahujú, tak posledná nájdená poloha je presunutá do premenných AxisXOld a AxisYOld. Ďalej sú vždy premenné AxisX a AxisY naplnené hodnotou NaN, výstupná premenná „isBallFound“ má logickú úroveň FALSE a je vypočítaný časový rozdiel od posledného hľadania. Ak sa nezmenil počet úspešných a neúspešných hľadání, má výstupná premenná „isBallFound“ hodnotu FALSE.

Použitý stavový automat pre voľbu v akej zóne sa bude vyhľadávať loptička, má tri základné stavy: CHANGE, ONE_ZONE, WHOLE_ZONE. Ak sa mení zóna stavový automat, musí vždy prejsť cez stav CHANGE. V tomto stave sa čaká na spätnú väzbu o tom, či bola zmena zón vykonaná. Tento údaj je uložený v premennej „Status_I2001_S01“ v bite číslo 16. Ďalší stav je potom volený na základe prechádzajúceho stavu. Napríklad, ak bol prechádzajúci stav ONE_ZONE, stav sa zmení na WHOLE_ZONE. V stave WHOLE_ZONE sa vyhľadáva loptička na celej ploche. Stav sa mení, pokiaľ je hľadanie loptičky úspešné a premenná „UserData_I2021_S01“ je vymaskovaná na základe zóny, v ktorej sa našla podľa Tab. 7. Zóna číslo 7 je vždy volaná, lebo sa očakáva, že tu bude zaznamenaná loptička po strele. Potom je nutné vymaskovať premennú „Control_I2000_S01“ s 3. a 16. bitom pre potvrdenie užívateľských údajov. V stave jednej zóny je jedinou podmienkou, či hodnoty súradníc nie sú NaN. V prípade, ak majú hodnotu NaN, mení sa stav rovnakým spôsobom ako pri stave WHOLE_ZONE. Rozdielom je iba, že premenná UserData_I2021_S01 je maskovaná hexadecimálnym číslom FFFF FFFF, aby boli všetky zóny aktívne.



Obr. 24 Vývojový diagram funkčného bloku Camera popisujúci zmenu zón

Ak je hlavný stav zmenený na ERROR, výstupná premenná „Error“ ma hodnotu TRUE a premenná „isCmeraSeraching“ je zmenená na FALSE. K zmene stavu je určená vstupná premenná „ErrorReset“. Ak je hodnota tejto premennej v stave TRUE, je volaná funkcia pre resetovanie celého funkčného bloku a hlavný stav sa mení na ENABLED.



Obr. 25 Funkčný blok pre detekciu polohy loptičky

5.2.3 Funkcia pre ovládanie osvetlenia

K ovládaniu osvetlenia som vytvoril funkčný blok, ktorý som pomenoval „Bar_Light“. Keďže použité pruhové svetlá sú riadené predovšetkým pomocou riadiaceho kontaktu, ktorým je vedený signál z digitálneho výstupu, je možné celú funkciu osvetlenia ovládať z PLC, a to vrátane nastavenia jasu.

Funkčný blok obsahuje celkovo päť vstupov z toho dva sú štruktúry:

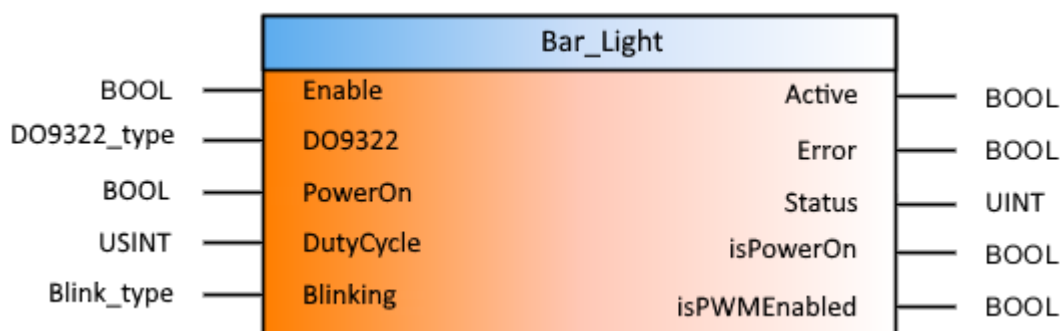
- Enable – premenná je typu BOOL. Vo funkčnom bloku slúži pre aktiváciu ovládania osvetlenia. V deaktivovanom stave má blok minimálne nároky na výkon.
- DO9322 – vstup je pomenovaný podľa použitej výstupnej karty pre jednoduchšie mapovanie štruktúry k výstupným premenným karty pre nezainteresované osoby. Vstup je dátového typu DO9322_type, ktorá obsahuje premenné ModuleOk a DigitalOutput.
- PowerOn – premenná typu BOOL. Hodnotou TRUE alebo FALSE je spúšťané osvetlenie.
- DutyCycle – premenná typu USINT. Týmto vstupom je riadená intenzita osvetlenia. Zadávaná je hodnota v percentách v rozsahu od 0 do 100.
- Blink – vstupom je štruktúra typu Blink_type. Štruktúra obsahuje premennú StartBlinking, ktorou sa aktivuje blikanie. Ďalej obsahuje premenné BlinkPeriod_ms, ktorou sa v milisekundách zadáva perióda blikania a premenná BlinkingTime_ms, ktorou sa nastavuje celkový čas blikania.

Výstupom funkčného bloku sú premenné:

- Active – premenná typu BOOL. Indikuje, že funkčný blok je aktívny.
- Error – premenná typu BOOL. V prípade, že nastane chyba vo funkčnom bloku alebo na použítom hardvéri, je bit nastavený na hodnotu TRUE inak je výstupom FALSE.
- Status – premenná typu UINT. Výstupom je číslo chyby. Pokiaľ nie je chyba prítomná, výstupom je 0.
- isPowerOn – premenná typu BOOL. Výstupom je informácia, či je osvetlenie zapnuté.
- isPWMEabled – premenná typu BOOL. Výstupom je informácia o tom, či sa vo funkčnom bloku využíva PWM pre riadenie intenzity osvetlenia.

V prvom cykle po volaní funkčného bloku z hlavného programu je hlavný riadiaci switch nastavený na stav INIT_BARLIGHT. Inicializačná časť slúži predovšetkým k zisteniu, v akej cyklickej triede a s akým časom cyklu sa funkčný blok pre riadenie osvetlenia nachádza. Aktuálny čas cyklu dokážem zistiť pomocou funkčného bloku RTInfo, ktorý sa nachádza v knižnici BRSystem. Po volaní funkčného bloku dostávam štruktúru s informáciami o runtime. Pre zistenie cyklického času je podstatná premenná s názvom cycle_time, kde hodnota času je uvádzaná v mikrosekundách. Vďaka tejto

informácii môžem rozhodnúť, či bude možné ovládať intenzitu osvetlenia pomocou pulznej šírkovej modulácie v priebehu programu. Funkcia PWM je vytvorená iba za účelom budúceho využitia, keďže k snímaniu obrazu sa nehodí. Základnou podmienkou pre riadenie intenzity pomocou PWM je, aby perióda cyklu bola minimálne 20 ms. Pokiaľ by bol čas cyklu väčší, bolo by možné ľudským okom zachytiť blikanie svetla. Ako ďalšiu podmienku som stanovil, že musí byť možné nastaviť aspoň 10 úrovní jasu, aby malo riadenie intenzity zmysel. Tým sa minimálny čas cyklu pre riadenie intenzity osvetlenia znížil na 2 ms. Pokiaľ bude funkčný blok umiestnený do cyklickej triedy vyššej ako 2 ms, bude automaticky blokována možnosť zmeny jasu. Po vykonaní inicializačnej časti prechádza stav funkčného bloku do stavu `DISABLED_BARLIGHT`. V tomto stave sa čaká na zopnutie vstupnej premennej `Enable`. Pokiaľ má premenná hodnotu `FALSE`, má funkčný blok minimálne nároky na výkon. Pri nastavení bitu na hodnotu `TRUE` prechádza program do podmienky, či je použitý výstupný modul v poriadku. Pokiaľ modul nie je funkčný, je stavový automat nastavený do stavu `ERROR_BARLIGHT`. Ak je modul v poriadku, stavový automat prechádza do stavu `ENABLED_BARLIGHT`. V tomto stave je funkčný blok schopný ovládať osvetlenie, ale len na základe podmienky v inicializačnej časti. Po vstupe do stavu je automaticky zopnutý výstupný bit `Active` na hodnotu `TRUE`. Keďže osvetlenie je spínané vstupnou premennou `PowerOn`, je každým cyklom kontrolovaná hodnota tohto vstupu. Ak je bit nastavený na `FALSE`, je táto hodnota ďalej predaná na kartu digitálneho výstupu a teda osvetlenie nesvieti. Ak je hodnota bitu nastavená `TRUE`, sú 3 možnosti čo sa môže stať podľa priority. Najvyššiu prioritu má Blikanie. Pokiaľ je zopnutá aj premenná `StartBlinking`, je volaná funkcia `BlinkControl`. Funkcia je volaná cyklicky a automaticky spína a vypína osvetlenie s nastavenou periódou podľa premennej `BlinkPeriod_ms`. Po uplynutí stanoveného času blikania premennou `BlinkingTime_ms` sa spúšťa bit hodnoty `StartBlinking` nastaví na `FALSE`. Druhú prioritu má PWM. Ak je splnená počiatočná podmienka pre PWM, tak je vždy možné riadiť osvetlenie pomocou vstupu `DutyCycle`. Pri poslednej a najnižšej priorite je osvetlenie možné iba zapnúť alebo vypnúť. Pokiaľ nastane chyba, je stav prehodený na `ERROR_BARLIGHT`. Funkčný blok ostáva v tomto stave, pokiaľ nie je chyba vyriešená alebo blok nie je vypnutý pomocou vstupu `Enable`. Výstupný bit `Error` je nastavený na `TRUE` a `Status` je nastavený na základe chyby.



Obr. 26 Funkčný blok pre ovládanie osvetlenia

5.2.4 Funkcia pre ovládanie brán

Výstup fotoelektrického snímača, ktorý je pripojený k digitálnemu vstupu, musí byť taktiež spracovaný. Aby bol kód prehľadný, opäť som vytvoril funkčný blok, ktorým je možné spúšťať funkciu, zisťovať chyby na karte, ale hlavne detegovať gól a vo vhodnej forme informáciu predať ďalej. Vstupy do funkčného bloku sú:

- Enable – premenná typu BOOL. Logickou hodnotou TRUE je funkčný blok aktivovaný. Hodnotou FALSE je neaktívny.
- DI9371 – štruktúra typu DI9371_type. Vstup je pomenovaný podľa vstupnej karty pre jednoduchšie mapovanie premenných pre nezainteresovanú osobu. Štruktúra obsahuje premennú ModuleOk pre kontrolu funkčnosti vstupnej karty. Ďalej štruktúra obsahuje premenné DigitalInput_1 a DigitalInput_2, ktoré sú mapované na vstupy zo snímačov.

Funkčný blok obsahuje tieto výstupy:

- Status – premenná je dátového typu UINT. Výstupom je informácia o stave funkčného bloku.
- isGoal1 – premenná dátového typu BOOL. V prípade, keď snímač deteguje prechádzajúcu loptičku, je na výstupe hodnota TRUE inak je výstupom hodnota FALSE. Premenná je mapovaná k vstupu snímača pod bránkou robotického protihráča.
- isGoal2 – premenná je taktiež dátového typu BOOL. Výstup má rovnaké vlastnosti ako isGoal1. Rozdiel je, že premenná je mapovaná k snímaču použitom pod bránkou ľudských hráčov.

V prvom cykle po volaní funkčného bloku je počiatočná hodnota „Enable“ nastavená na FALSE. Pokiaľ užívateľ nezmení túto hodnotu výstupom Statusu, je hodnota 66534, ktorá znamená, že funkčný blok nie je spustený, ale je volaný. Keď užívateľ aktivuje funkčný blok, program vykoná inicializačnú časť, v ktorej sú nastavené hodnoty časovačov na 100 milisekúnd. Inicializácia sa vykonáva vždy iba raz, a to hneď po aktivovaní funkčného bloku. Potom je každý cyklus kontrolovaná podmienka, či je použitá vstupná karta v poriadku. Pokiaľ nie je karta v poriadku, je výstupom funkcie Status s hodnotou 45813, čo značí nefunkčný modul a zvyšok programu sa nevykonáva. Pre potreby ďalšej práce s funkčným blokom je nutné, aby po páde gólu bola na výstupe isGoal1 alebo isGoal2 hodnota TRUE iba jeden cyklus. Bez použitia časovačov by nebolo možné tento stav dosiahnuť, pretože funkčný blok je umiestnený v cyklickej triede 1 milisekunda a loptička môže byť zaznamenaná aj desaťkrát za sebou, kým celá prejde cez snímač. Taktiež môže nastať situácia, že loptička sa odrazí od zásobníka a opäť sa na malú chvíľu ocitne v poli snímača. Preto po každom zaznamenanom góle je výstup zo snímačov ignorovaný.



Obr. 27 Funkčný blok pre detekciu gólov

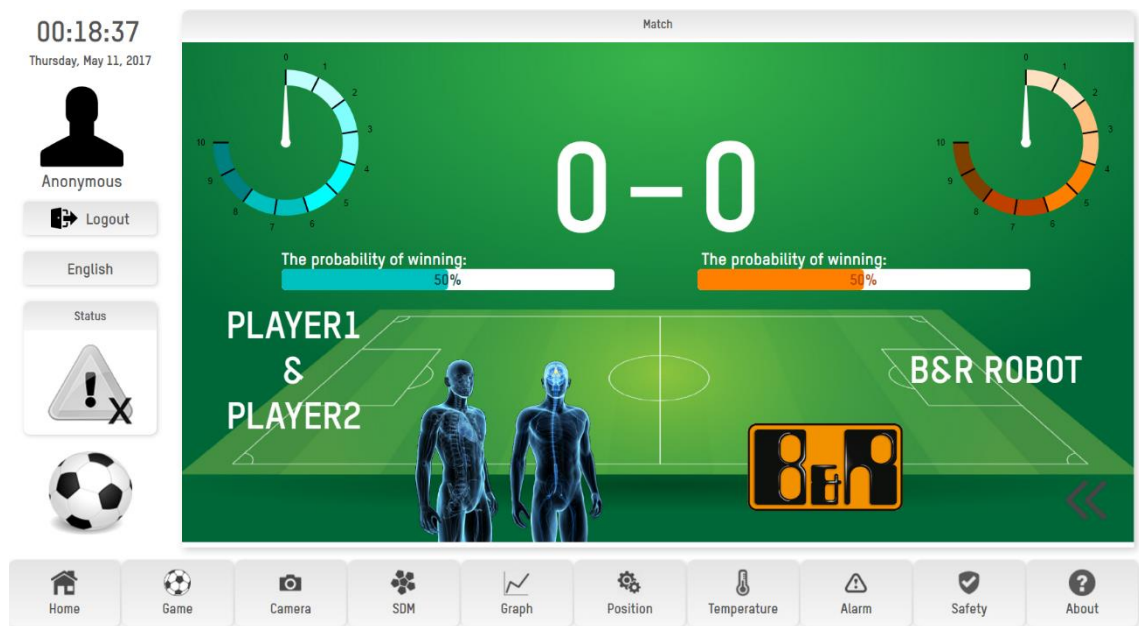
5.2.5 Vizualizácia

Automation Studio ponúka 3 typy vizualizácii. Najstaršou je vizualizácia s označením VC3. VC3 sa využívalo najmä pri paneloch s textovým displejom. Keďže tento typ je skôr historickou záležitosťou, preto sa jej nebudeme venovať.

Aktuálne najpoužívanější vizualizácia nesie označenie VC4. Tento typ vizualizácie prináša obrovské množstvo možností ako interpretovať požadovaný obsah. Aktuálny obraz sa prenáša pomocou VNC servera vytvorený PLC zariadením. V nastaveniach sa nastavuje perióda, s akou sa obnovuje obraz. [23]

Najnovší typ vizualizácie sa nazýva „mapp View“. Tento typ vizualizácie využíva HTML5, CSS3 a JavaScript, no veľkou výhodou je, že programátor nemusí tieto jazyky poznať. Vizualizáciu je možné vytvárať priamo v prostredí Automation Studio s predpripraveným grafickým prostredím. Jednotlivé vizualizačné komponenty sa nazývajú „widgety“. Sú to grafické prvky, ktoré fungujú na princípe „drag-and-drop“, teda chyť a pusti. Jedným z takýchto widgetov je napríklad tlačidlo alebo zmena jazyku. Po pridaní takého prvku sa kód generuje automaticky na základe toho ako programátor vyplňa hodnoty v predpripravených tabuľkách. Výsledná obrazovka sa môže skladať z viacerých častí, ktoré sa nazývajú „Layout“. Pomocou týchto layout-ov je možné dynamicky meniť len časti výslednej obrazovky. Napríklad v jednej časti obrazovky sa bude nachádzať panel na prepínanie stránok, ktorý bude nemenný a v inej časti obrazovky bude hlavná stránka, ktorá sa bude dynamicky meniť na základe aktuálne zvolenej stránky. Keďže vizualizácia sa vykonáva priamo na zobrazovacom zariadení, je nutná komunikácia s výpočtovým zariadením. K tomuto účelu sa používa server OPC-UA. [23] [24]

OPC je štandard, ktorý sa využíva na bezpečný a spoľahlivý prenos dát v sektore priemyselnej automatizácie. Pomocou tohto štandardu je možné prenášať dáta v reálnom čase, informáciu o upozorneniach, prístup k historickým dátam a podobne. Veľkou výhodou tejto komunikácie je univerzálnosť, keďže OPC využíva veľké množstvo firiem a teda je možné kombinovať aj rozdielne systémy. OPC-UA je rozšírením klasického OPC napríklad o hľadanie OPC serverov na sieti. [25] [26]



Obr. 28 Príklad vizualizácie pomocou Mapp View

6 ZÁVER

Výsledkom bakalárskej práce je reálny funkčný systém, ktorý dokáže detegovať loptičku na hracej ploche robotického stolného futbalu v primeranom čase, rádovo v desiatkach milisekúnd. V prvom rade bola vyberaná pozícia kamery. Po zvážení možností bola kamera umiestnená pod hraciu plochu, kde je chránená pred poškodením a do snímaného priestoru sa nedostane cudzí objekt vďaka plastovej ochrane okolo stola. Priehľadná hracia plocha bola použitá z prechádzajúcej diplomovej práce [1]. Spoľahlivá detekcia loptičky vyžadovala osvetlenie hracej plochy. Po testovaní osvetlenia z rôznych smerov bola vybraná pozícia nad hracou plochou. Ako zdroj svetla bola vybraná séria lineárnych priemyselných svietidiel s bielym svetlom. Použité osvetlenie malo dostatočný výkon pre osvetlenie všetkých častí ihriska. Osvetlenie bolo umiestnené do správnej polohy vďaka konštrukcii, ktorá bola zadaná do výroby.

Ďalšia časť bakalárskej práce bola zameraná najmä na softwarovú časť. Použitá kamera bola programovaná v prostredí In-Sight Explorer. Vyhľadávací algoritmus kamery dokázal vyhľadať pozíciu loptičky aj napriek vzniknutým tieňom od hráčov. Períodu vyhľadávania sa skrátila viac ako trojnásobne vďaka rozdeleniu plochy na vyhľadávacie zóny. Dáta z kamery sú ďalej spracované systémom PLC. V bakalárskej práci boli vytvorené tri knižnice:

- Camera – slúži k ovládaniu kamery, spracovanie získaných údajov a interpretáciu výsledkov
- Bar_Light – slúži k ovládaniu osvetlenia
- Goalkeeper – spracúva informácie zo snímačov určených pre detekciu gólov

Pri testovaní na reálnom modeli bolo overené, že vytvorený algoritmus dokáže zaznamenať pri priemernej strele z pozície brankára, obrancov a zálohy dostatočný počet bodov na efektívnu obranu. Pri streľbe od útočníka kamera často nestihne zaznamenať viac ako jeden bod. Tento problém by mohlo riešiť použitie inej kamery, ktorá dokáže zaznamenať viac snímok za sekundu a následne by boli externe spracované.

ZOZNAM OBRÁZKOV

Obr. 1 Stolný futbal od firmy B+R pred úpravou [1]	10
Obr. 2 Pohľad na technické vybavenie robotického futbalu z univerzity Ecole polytechnique fédérale de Lausanne [4]	11
Obr. 3 Pohľad na technické vybavenie robotického futbalu z Technickej Univerzity v Dánsku [5]	12
Obr. 4 Pohľad na robotický futbal od firmy PILZ [1]	13
Obr. 5 Pohľad na matnú hraciu plochu [1]	15
Obr. 6 Kamera Cognex s formou a ramenom	16
Obr. 7 Použitý objektív Kowa LM3NCM [11]	17
Obr. 8 Loptička v zákryte s hráčom [12]	18
Obr. 9 Umiestnenie kamery pod hracou plochou	19
Obr. 10 Princíp scény typu "Dome"	20
Obr. 11 Odraz svetla od plexiskla [1]	20
Obr. 12 Nosná konštrukcia pre osvetlenie	21
Obr. 13 Pružové svetlo LL-245W [14]	22
Obr. 14 Prípravok pre snímanie gólov	23
Obr. 15 Umiestnenie prvku pre snímanie gólov	23
Obr. 16 Hardwarová konfigurácia v PLC	24
Obr. 17 APC2100 s použitým panelom [16]	24
Obr. 18 Vstupné a výstupné moduly	26
Obr. 19 Konštrukcia k lineárnemu a rotačnému pohybu zároveň	27
Obr. 20 Princíp komunikácie POWERLINKu [21]	28
Obr. 21 Testovací algoritmus kamery s úspešne vyhladanou loptičkou	31
Obr. 22 Rozloženie zón	33
Obr. 23 Vývojový diagram hlavných stavov funkčného bloku Camera	39
Obr. 24 Vývojový diagram funkčného bloku Camera popisujúci zmenu zón	42
Obr. 25 Funkčný blok pre detekciu polohy loptičky	42
Obr. 26 Funkčný blok pre ovládanie osvetlenia	44
Obr. 27 Funkčný blok pre detekciu gólov	46
Obr. 28 Príklad vizualizácie pomocou Mapp View	47

ZOZNAM TABULIEK

Tab. 1 Špecifikácia hracieho stola [7]	14
Tab. 2 Parametre pruhového osvetlenia [13] [14]	22
Tab. 3 hodnoty lineárneho motoru [18]	26
Tab. 4 hodnoty rotačného motoru [19]	27
Tab. 5 Doba letu loptičky podľa pozície strelca	32
Tab. 6 Zóny a parametre vyhľadávania	33
Tab. 7 Vstupné hodnoty k spúšťaniu zón	34
Tab. 8 Zoznam programovacích jazykov v prostredí Automation Studio	36

LITERATÚRA

- [1] L. JEDINÝ, Robotický stolní fotbal, [online]. 2016. [Cit. 27-12-2016].
- [2] Wikipedia, Stolný fotbal, [online]. [Cit. 17-11-2016]. Dostupné z: <https://sk.wikipedia.org/wiki/StolnC3%BD_futbal>.
- [3] ITFS, ITFS Rule Book, [online]. [Cit. 17-11-2016]. Dostupné z: <<http://www.table-soccer.org/rules/documents/ITSFRulesSlovakian.pdf>>.
- [4] EPFL 2016, EPFL PRESENTATION & INTRODUCTION, [online]. [Cit. 11-12-2016]. Dostupné z: <<http://information.epfl.ch/introduction>>.
- [5] 2015 EPFL, All brawn, little brains: EPFL students' table-football robot, [online]. [Cit. 11-12-2016]. Dostupné z: <<http://actu.epfl.ch/news/all-brawn-little-brains-epfl-students-table-football/>>.
- [6] Automated foosball table, Automated foosball table, [online]. [Cit. 11-12-2016]. Dostupné z: <<http://foospmp.myl.dk/about>>.
- [7] 2016 Garlando Sports & Leisure, G-500, [online]. [Cit. 17-12-2016]. Dostupné z: <<http://www.garlando.com/products/the-classics/home-football-tables/g-500/>>.
- [8] P3 - I7000_SpecSheet, Specifications, [online]. [Cit. 18-12-2016]. Dostupné z: <https://partner.cognex.com/FileLibrary/77d129094c634939987bd8fcc75b6d99/P3%20-%20I7000_SpecSheet.pdf>.
- [9] 2015 Kowa, LM3NCM, [online]. [Cit. 24-12-2016]. Dostupné z: <<http://www.kowa.eu/lenses/en/LM3NCM.935.php>>.
- [10] Wikipedia, Angle of view, [online]. [Cit. 25-12-2016]. Dostupné z: <https://en.wikipedia.org/wiki/Angle_of_view>.
- [11] Kowa Optical Products Co, WIDE MEGAPIXEL, [online]. [Cit. 25-12-2016]. Dostupné z: <http://www.kowa-optical.co.jp/fa/e/products/wide_megapixel.html>.
- [12] DocteurCosmos, File:Table football ball close-up 1.jpg, [online]. [Cit. 28-12-2016]. Dostupné z: <https://commons.wikimedia.org/wiki/File:Table_football_ball_close-up_1.jpg>.
- [13] Smart View, PRUHOVÁ SVĚTLA LL120 A LL60, [online]. [Cit. 29-12-2016]. Dostupné z: <http://www.smartview.cz/sites/default/files/lights/data_sheet/smartview_datasheet_bar_cz.pdf>.

- [14] Smart View, VÝKONNÁ PRUHOVÁ SVĚTLA, [online]. [Cit. 29-12-2016]. Dostupné z: http://www.smartview.cz/sites/default/files/lights/data_sheet/smartview_datasheet_strong_cz.pdf.
- [15] M. Petr, Robotický stolní fotbal, [online]. 2016. [Cit. 7-5-2017].
- [16] 2017 B&R, Automation PC 2100, [online]. [Cit. 6-5-2017]. Dostupné z: <https://www.br-automation.com/en/products/industrial-pcs/automation-pc-2100/>.
- [17] B&R, *Automatio studio - B&R Help Explorer 4.2.6.23*, 2016.
- [18] B&R, ACOPOSmicro user's manual, [online]. 12 2016. [Cit. 12-5-2017]. Dostupné z: http://www.br-automation.com/downloads_br_productcatalogue/BRP44400000000000000489410/MAACPMICRO1-ENG_V1.20.1.pdf.
- [19] NTI AG LinMot, Linear Motors, [online]. 2015. [Cit. 12-5-2017]. Dostupné z: <http://www.linmot.com/products/linear-motors/>.
- [20] B&R, 8LVA23.B1030D100-0, [online]. [Cit. 5-20-2017]. Dostupné z: <https://www.br-automation.com/en-us/products/undefined/8lva23b1030d100-0/#techdata>.
- [21] F. ZEŽULKA a O. HYNČICA, Průmyslový Ethernet VIII: Ethernet Powerlink, Profinet, [online]. [Cit. 19-12-2016]. Dostupné z: <http://www.odbornecasopisy.cz/res/pdf/37288.pdf>.
- [22] 2016 openPOWERLINK, POWERLINK Mechanism, [online]. [Cit. 19-12-2016]. Dostupné z: <http://openpowerlink.sourceforge.net/web/POWERLINK/Mechanism.html>.
- [23] In-Sight, *In-Sight Explorer Help 5.4.0*, 2017.
- [24] 2017 B&R, Web meets automation, [online]. [Cit. 2-1-2017]. Dostupné z: <https://www.br-automation.com/en/products/innovations-2016/web-meets-automation/>.
- [25] OPC Foundation, What is OPC?, [online]. [Cit. 2-1-2017]. Dostupné z: <https://opcfoundation.org/about/what-is-opc/>.
- [26] OPC Foundation,, Unified Architecture, [online]. [Cit. 2-1-2017]. Dostupné z: <https://opcfoundation.org/about/opc-technologies/opc-ua/>.
- [27] sensotronik, OESV 19 K 2000 P3K, [online]. [Cit. 7-5-2017]. Dostupné z: http://www.sensotronik.se/pdf/03foto/DIS/201173_en.pdf.

ZOZNAM SYMBOLOV A SKRATIEK

APC – Automation PC
BOOL – boolean
CN – Controlled Node
CSS3 - Cascading Style Sheets 3
FB – Function Block
GB – Gigabyte
HD – Hight Definition
HTML5 - HyperText Markup Language 5
Hz – Hertz
I/O – Input/Output
IP – Internet Protocol
IS – In-Sight
mA – mili Ampér
mm – milimeter
MN – Managing Node
ms – milisekunda
NaN – Not a Number
OPC – Open Connectivity
PLC - Programmable Logic Controller
SoA – Start of Asynchronous
SoC – Start of Cyclic Frame
PRes – Pool Response
PReq – Pool Request
UA - Unified Architecture
UDNIT – Unsigned Double Integer
UINT – Unsigned Integer
USB – Universal Serial Bus
V - Volt
VC3 – Visual Components 3
VC4 – Visual Components 4
VDC - Volts of direct current
VNC – Virtual Network Computing
W - Watt
 Ω - Ohm

ZOZNAM PRÍLOH

Príloha 1: Bloková schéma elektrického zapojenia robotického stolného futbalu

Príloha 2: Multimédia

Príloha 3: Projekty z prostredia In-Sight Explorer a Automation Studio

Príloha 4: Bakalárska práca v elektronickej podobe