



ŠOLSKI CENTER CELJE

Srednja šola za strojništvo, mehatroniko in medije

RAZISKOVALNA NALOGA

SAMODEJNO PREPOZNAVANJE STEKLENIC

Avtorja:

Simon SOVIČ, M-4. c

Jan RUTAR, M-4. c

Mentor:

Miloš BEVC, univ. dipl. inž.

Celje, marec 2022

Mentor Miloš Bevc v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavlja, da je v raziskovalni nalogi z naslovom Samodejno prepoznavanje steklenic, katere avtorja sta Simon Sovič in Jan Rutar:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, _____

žig šole

Podpis mentorja

Podpis odgovorne osebe

*

POJASNILO

V skladu z 20. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja (-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja (-ice) fotografskega gradiva, katerega ni avtor (-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.

POVZETEK

Cilj raziskovalne naloge je bil rešiti problem strojnega ločevanja steklenic različnih oblik s fotografij, ki bi potreboval čim manjšo človeško vmešavanje. Naredili smo sistem pametne kamere Omron FQ2, krmilnika Siemens S7-1200 in tekočega traku, ki je bil tega zmožen. S funkcijo detekcije robov na pametni kameri smo pridobili podatke o njihovi obliki na fotografijah, v krmilniku pa smo jih poskusili obdelati na več načinov in tako doseči čim boljše kategoriziranje. Ob delu smo si pomagali z različnimi metodami raziskovanja, največ z metodo testiranja, saj je omogočala najboljši vpogled v delovanje in funkcionalnost našega izdelka.

Ključne besede: FQ2, pametna kamera, ločevanje predmetov

ABSTRACT

The goal of our research was to solve the problem of automatically distinguishing between glass bottles of different shapes and sizes using a smart camera and a PLC. We built a system with an Omron FQ2 smart camera, a Siemens S7-1200 PLC, and a conveyor belt capable of that. We used the FQ2's edge detection feature to measure the dimensions of the glass bottles on the photographs. That information got sent to the PLC, where we tried to implement multiple different algorithms to analyse it and find out which bottle it most likely depicts. During our research, we used various research methods, but primarily experimentation, since it enabled us to evaluate the reliability of our system.

Keywords: FQ2, smart camera, object sorting

KAZALO VSEBINE

1	Uvod.....	1
1.1	Opredelitev problema	1
1.2	Hipoteze	4
1.3	Raziskovalne metode	4
2	Osnovna ideja.....	5
3	Teorija	7
3.1	Strojni vid	7
3.2	Industrijski Ethernet.....	9
4	Izbira komponent.....	11
4.1	Pametna kamera FQ2.....	11
4.1.1	Programiranje kamere.....	12
4.1.1.1	Zavihek »Image«.....	13
4.1.1.2	Zavihek »Inspect«	14
4.1.1.3	Zavihek »In/Out«	16
4.1.2	Nastavitev protokola za komuniciranje z drugimi napravami.....	17
4.1.3	Ozadje predmeta	18
4.2	Krmilnik.....	19
4.2.1	Siemens S7-300	20
4.2.2	Siemens S7-1200	21
4.3	Motor/tekoči trak	22
4.4	Frekvenčnik	23
4.5	HMI panel	24
4.6	Ethernet stikalo	25
4.7	Napajalnik	26
4.8	Programska oprema	26

4.8.1	TIA Portal.....	26
5	Program kamere	27
6	Povezava komponent.....	28
6.1	IP naslovi	28
6.2	Protokoli povezav	28
6.2.1	Nastavitve povezave kamere in krmilnika	29
7	Program krmilnika.....	32
7.1	Pristopi k računanju odstopanja dveh steklenic	32
7.1.1	Izračun odstopanja steklenice glede na največjo vertikalno mero	33
7.1.2	Izračun odstopanja steklenice glede na vsoto odstopanja posameznih vertikalnih in horizontalnih mer	34
7.1.3	Izračun odstopanja steklenice glede na vsoto vertikalnih mer	35
7.1.4	Izračun odstopanja steklenice glede na vsoto odstopanja n-tih največjih vertikalnih mer	36
7.2	Določanje steklenice z glasovanjem različnih algoritmov	37
7.3	Podatkovni bloki	38
7.4	Glavni program	39
7.4.1	Seznam funkcij	39
7.5	Program HMI	43
7.7	Upravljanje motorja	48
7.8	Optimizacija programa	50
7.9	Diagnoza programa	52
8	Ovrednotenje hipotez	53
9	Zaključek.....	54
10	Viri.....	55

KAZALO SLIK

Slika 1: Osnovni problem.....	1
Slika 2: Valovita steklenica.....	2
Slika 3: Okrogla steklenica.....	2
Slika 4: Široka steklenica.....	3
Slika 5: Osnovni problem.....	5
Slika 6: Osnovna ideja.....	6
Slika 7: Strojni vid.....	7
Slika 8: Osvetljava pri strojnem vidu.....	8
Slika 9: Komunikacijske plasti po ISO/OSI.....	9
Slika 10: Zgradba PROFINET-a.....	10
Slika 11: Žice, povezane s kamero.....	11
Slika 12: Prenos programa FQ Touchfinder Simulator.....	12
Slika 13: Zavihek »Image«.....	13
Slika 14: Zavihek »Trigger«.....	13
Slika 15: Zavihek »Image adjustment«.....	14
Slika 16: Zavihek »Inspect«.....	14
Slika 17: Dodajanje meritev.....	15
Slika 18: Možnosti zaznavanja.....	15
Slika 19: Zaznavanje robov.....	16
Slika 20: Zavihek »In/Out«.....	17
Slika 21: »Output« format.....	17
Slika 22: Primer pretemnega ozadja.....	18
Slika 23: Primer presvetlega ozadja.....	18
Slika 24: Krmilnik S7-300.....	20
Slika 25: Krmilnik S7-1200.....	21
Slika 26: Motor.....	22
Slika 27: Tekoči trak.....	22
Slika 28: Frekvenčnik.....	23
Slika 29: HMI zaslon.....	24
Slika 30: Ethernet stikalo.....	25
Slika 31: Napajalnik.....	26

Slika 32: Vertikalne in horizontalne meritve.....	27
Slika 33: TRCV_C	29
Slika 34: TRC_V parametri.....	30
Slika 35: Nastavitev komunikacije na kameri	31
Slika 36: Primer fotografije s preslabim ozadjem, z obkroženo napačno določeno mero ..	33
Slika 37: Pravilni in napačni meritvi najvišjih točk steklenice	36
Slika 38: »Simboli v array«.....	39
Slika 39: »Vsota absolutnih vrednosti«.....	40
Slika 40: »Razlika največjih višin«	40
Slika 41: »Seštevek vseh višin«	41
Slika 42: »Vsota n-tih višin«	41
Slika 43: »Poišči najmanjše odstopanje«.....	42
Slika 44: »Štetje steklenic z glasovi«	42
Slika 45: »Move«	42
Slika 46: Začetni zaslon.....	43
Slika 47: Spreminjanje steklenic	44
Slika 48: Spreminjanje mer steklenice	44
Slika 49: Spreminjanje mer s podatki iz kamere	45
Slika 50: Štetje steklenic	46
Slika 51: Izbira steklenice na liniji	46
Slika 52: Nastavljanje frekvenčnika.....	47
Slika 53: »NORM_X«.....	48
Slika 54: »SCALE_X«	48
Slika 55: Upravljanje motorja.....	49
Slika 56: Podatkovni tip »Steklenica«.....	50
Slika 57: Podatkovni tip »Glas«	50
Slika 58: Konstante.....	51
Slika 59: Diagnoza »error« stanja	52
Slika 60: Diagnoza dolžine vhoda.....	52

KAZALO GRAFOV

Graf 1: Odstopanje steklenice, zamaknjene levo ali desno od iste nezamaknjene steklenice	35
---	----

ZAHVALA

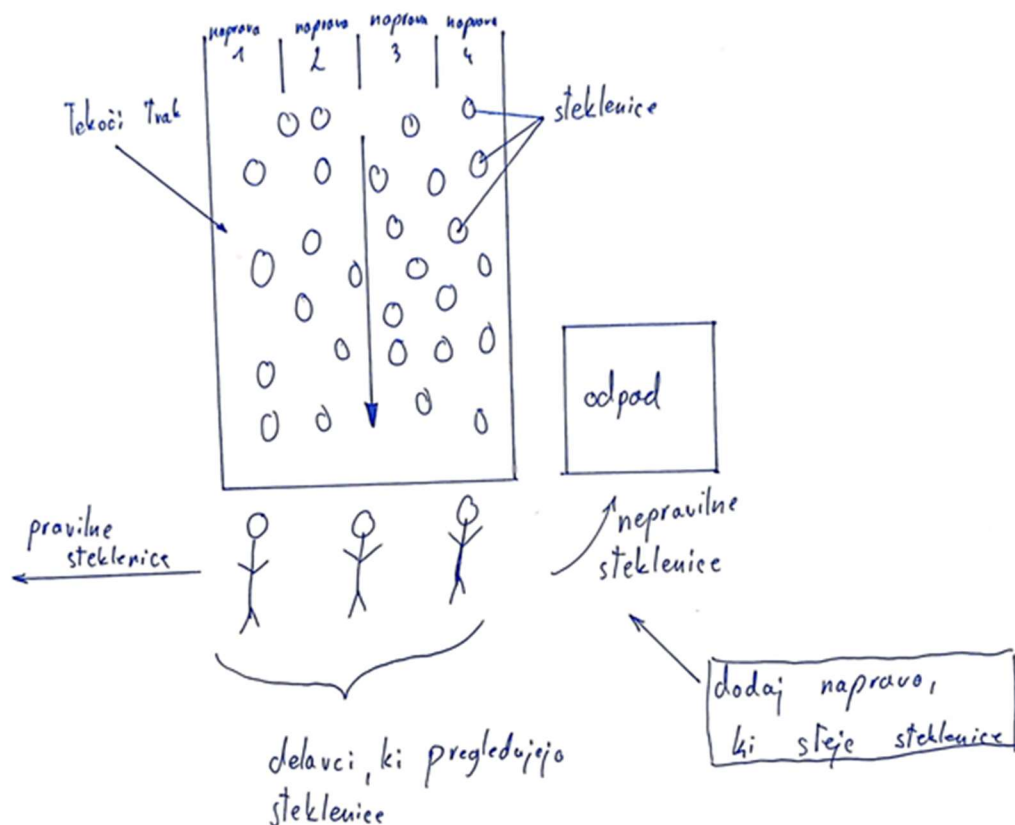
Iskreno se zahvaljujemo mentorju Milošu Bevcu, univ. dipl. inž., za koordinacijo in pomoč pri raziskovalni nalogi.

Prav tako se zahvaljujemo Tadeji Kolman, prof., za pregled in lektoriranje naloge, ter Simoni Tadeji Ribič, prof., za pregled povzetka v angleščini.

1 UVOD

1.1 OPREDELITEV PROBLEMA

V raziskovalni nalogi smo obravnavali pridobljeni problem Steklarne Hrastnik, kjer izdelujejo različne oblike steklenic. Največ štiri linije različnih steklenic pridejo na velik tekoči trak, kjer se med sabo pomešajo. Na koncu traku delavci izločujejo tiste, ki so nepravilne oblike ali poškodovane, in jih odvržejo v odpad. V steklarni želijo vedeti, koliko steklenic vsake oblike pristane na odpadu, saj lahko s tem podatkom najprej popravijo naprave, ki proizvajajo največ napačnih steklenic. Želimo izdelati napravo, s katero bi lahko samodejno prepoznali obliko steklenic in beležili njihovo število ter te podatke prikazali na zaslonu.



Slika 1: Osnovni problem

(Vir: osebni arhiv)

Iz steklarne so nam poslali naslednje steklenice, ki jih proizvajajo:

- Valovita steklenica – najvišja izmed dobljenih steklenic. Zaradi neokrogle oblike lahko pri merjenju širine pride do napak.



Slika 2: Valovita steklenica

(Vir: osebni arhiv)

- Okrogla steklenica – najnižja izmed steklenic in tudi najlažja za prepoznavo, saj njena orientacija ne vpliva na meritve.



Slika 3: Okrogla steklenica

(Vir: osebni arhiv)

- Široka steklenica – srednje višine, vendar najširša izmed vseh steklenic. Poleg tega njena oblika ni okrogla, zato pride do velikih razlik meritev pri njeni rotaciji.



Slika 4: Široka steklenica

(Vir: osebni arhiv)

1.2 HIPOTEZE

Cilj raziskovalne naloge je integracija strojnega vida v praktično aplikacijo. Pri tem želimo, da bi naprava delovala brez pomoči človeka in bi bila dovolj zanesljiva.

Za raziskovanje našega cilja smo si zastavili naslednje hipoteze.

1. Z ustrezno kamero in krmilnikom je možno prepoznati steklenico.
2. Sistem bo deloval z vsaj 80% zanesljivostjo.
3. Možno bo hitro dodajanje novih steklenic in spreminjanje podatkov že obstoječih.

1.3 RAZISKOVALNE METODE

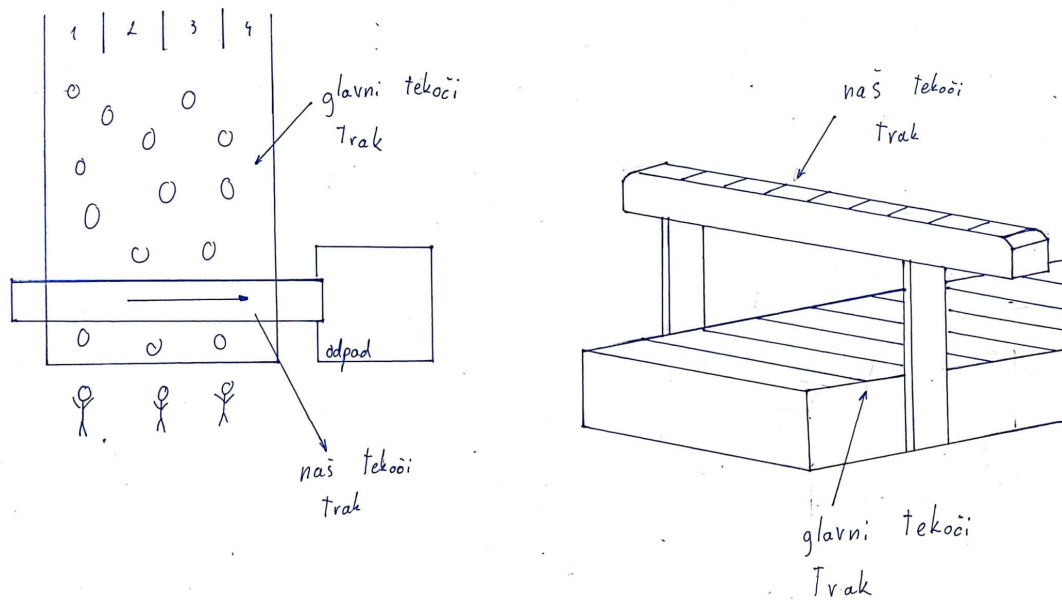
Pri raziskovanju rešitve smo uporabili sledeče metode raziskovanja:

- Metoda analize, pri kateri razdelimo celoten projekt na manjše dele. Projekt smo razdelili na izdelavo električnega vezja, komunikacijske povezave in programiranja.
- Primerjalna metoda, s katero smo si pomagali pri izbiri kamere in krmilnika.
- Testiranje, s katerim smo primerjali in ovrednotili rezultate ter se na podlagi teh odločili za izboljšanje sistema.

2 OSNOVNA IDEJA

V Steklarni Hrastnik že imajo pripravljen prostor, kamor bi namestili tekoči trak, na katerega bi delavci odlagali nepravilne steklenice. Mi bi morali na ta trak dodati našo napravo, sestavljeno iz:

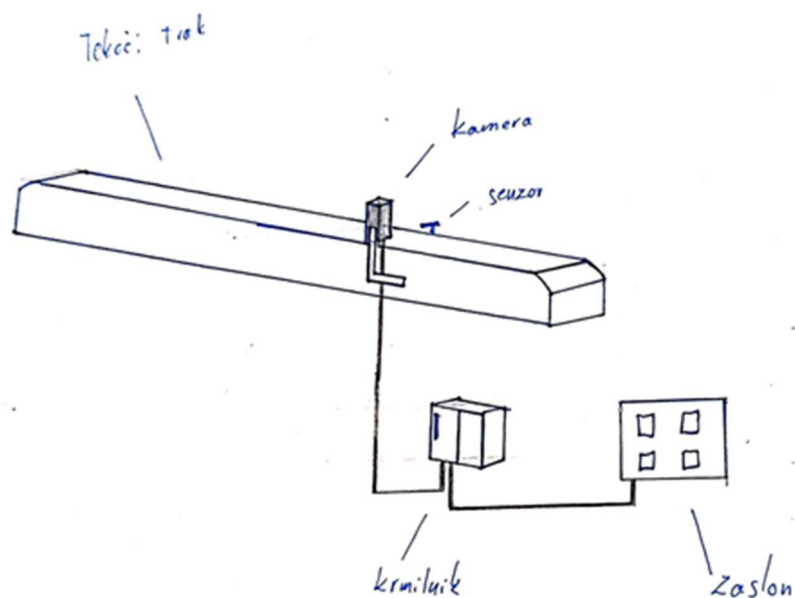
1. senzoričnega dela, ki bi prihod steklenice zaznal, jo izmeril in posredoval podatke;
2. krmilnega dela, ki bi podatke sprejel, jih obdelal in se odločil, za katero vrsto steklenice gre;
3. močnostnega dela, ki bi poganjal tekoči trak.



Slika 5: Osnovni problem

(Vir: osebni arhiv)

Vlogo fotografiranja in merjenja steklenice bi v našem zamišljenem sistemu prevzela pametna kamera, ki bi meritve poslala krmilniku. Krmilnik bi podatke obdelal in informacije prikazal na zaslonu. Steklenico, prihajajočo po tekočem traku, bi zaznal senzor.



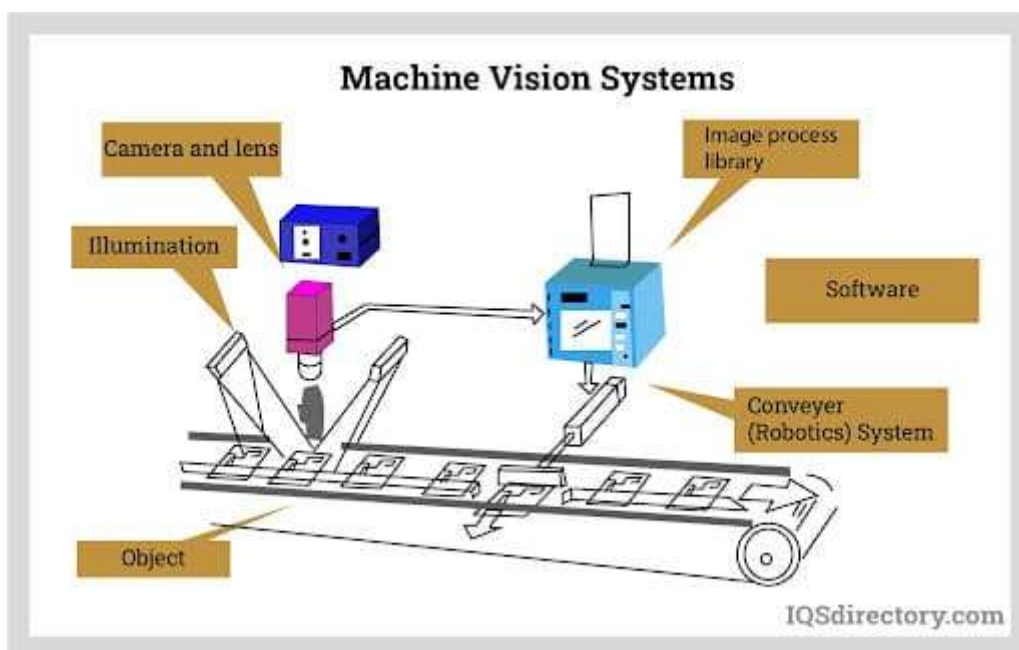
Slika 6: Osnovna ideja

(Vir: osebni arhiv)

3 TEORIJA

3.1 STROJNI VID

Strojni vid zajema tehnologijo in metode, ki se uporabljajo za zajemanje slike in njeno obdelavo. Nanaša se na številne tehnologije, izdelke programske in strojne opreme, integrirane sisteme, akcije, metode ter strokovno znanje. Največkrat se uporablja v industriji za avtomatski pregled izdelkov, nadzor procesov in robotsko usmerjevanje. Strojni vid, kot disciplina systemskega inženiringa, poizkuša uporabiti obstoječe tehnologije na nove načine in jih uporabiti za reševanje resničnih problemov. [3]

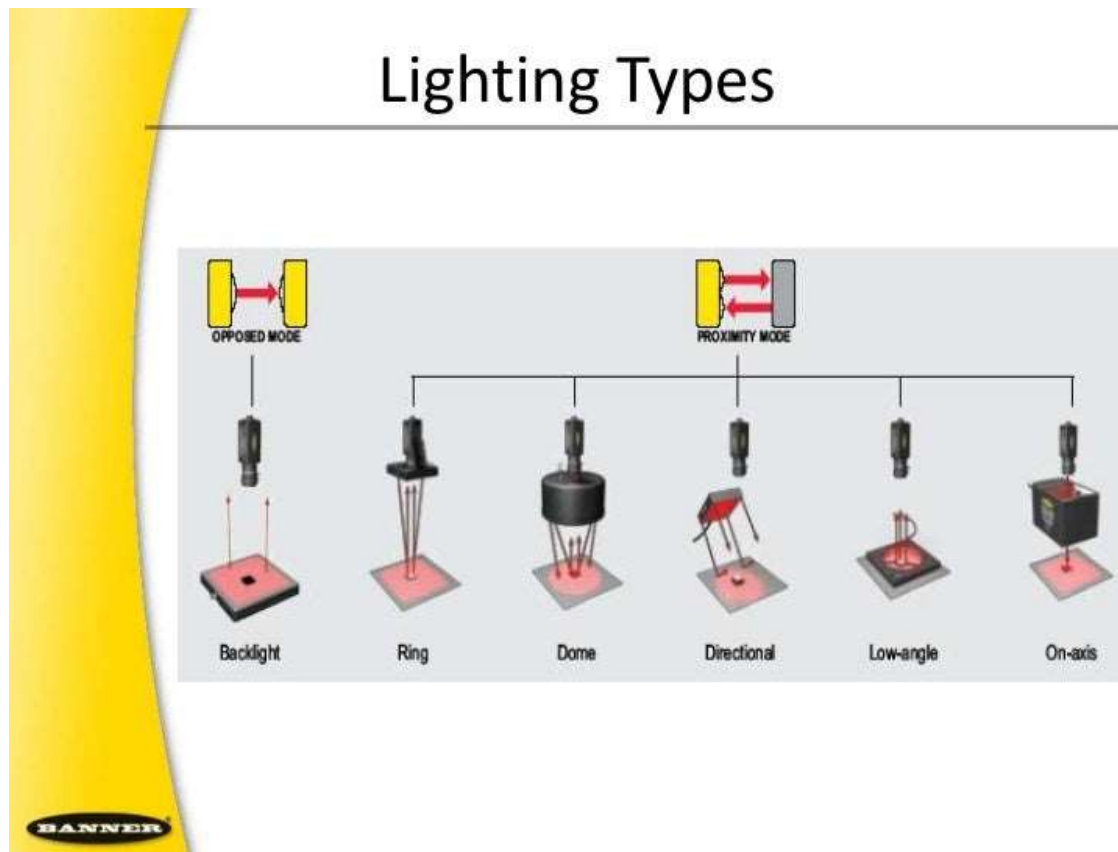


Slika 7: Strojni vid

(Vir: <https://www.iqsdirectory.com/articles/machine-vision-system.html>)

Sistemi strojnega vida so sestavljeni iz številnih komponent, ki vključujejo kamere, osvetlitev, optiko, leče, procesorje, programsko opremo in zaslone. Medtem ko lahko preprosti sistemi strojnega vida prepoznajo 2D- ali 3D-črtne kode, lahko bolj izpopolnjeni sistemi zagotovijo, da pregledani deli ustrezajo določenim tolerancam, so pravilno sestavljeni in brez napak. [14]

Pri strojnem vidu je zelo pomembna osvetlitev predmeta. Kamera ne more pregledovati predmetov, ki jih ne vidi. Zato je treba parametre osvetlitve, kot so oddaljenost svetlobnega vira od kamere in predmeta, kot, intenzivnost, svetlost, oblika, velikost in barva osvetlitve, optimizirati, da se poudarijo značilnosti, ki jih pregledujemo. Pri osvetlitvi je treba upoštevati tudi površinske lastnosti predmeta. [2]



Slika 8: Osvetljava pri strojnem vidu

(Vir: <https://pt.slideshare.net/engineerfriend/machine-vision-lighting>)

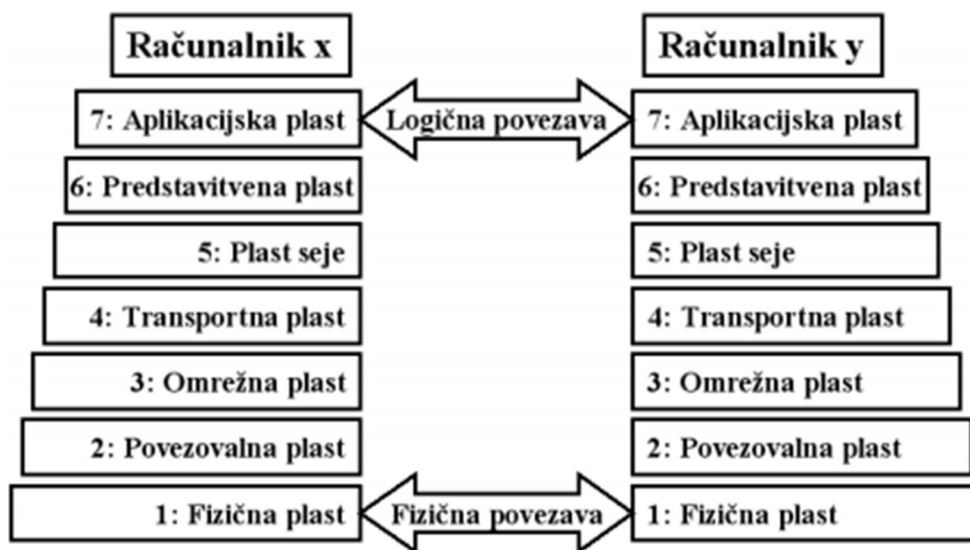
3.2 INDUSTRIJSKI ETHERNET

V industriji se danes srečujemo z velikim povečevanjem količine podatkov. Standardi, ki so bili razviti pred leti, ne zadovoljijo več potrebe po prepustnosti omrežja. Edini komunikacijski standard, ki dosega to prepustnost, je Ethernet. Industrijski Ethernet se v osnovi ne razlikuje od navadnega, dodane ima samo nekatere dodatne funkcije. Narejen je tako, da je primeren za uporabo v industriji, kjer je veliko elektromagnetnih motenj in velikih temperaturnih razlik. [13]

Ethernet je zgrajen po ISO/OSI standardu, kar pomeni, da je zgrajen iz sedem plasti. Vsaka od teh ima svojo nalogo pri komunikaciji.

1. Fizična plast
2. Povezovalna plast
3. Omrežna plast
4. Transportna plast
5. Plast seje
6. Predstavitvena plast
7. Aplikacijska plast

Dva sistema, zgrajena po ISO referenčnem sistemu, sta med seboj povezljiva. [13]

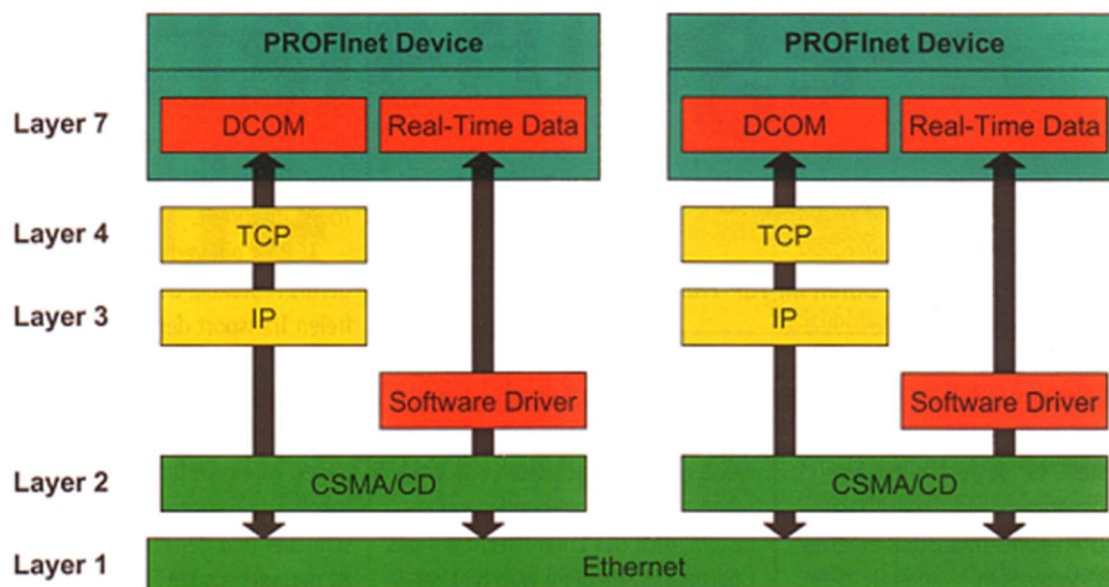


Slika 9: Komunikacijske plasti po ISO/OSI

(Vir: <https://dk.um.si/IzpisGradiva.php?id=1812&lang=eng>)

V industriji so zelo potrebni sistemi realnega časa, še posebej tam, kjer imamo razne pogone in aktuatorje, na primer v robotiki. Ravno v robotiki pa se srečujemo z največjim naraščanjem prenesene količine podatkov, ki jim klasični realnočasovni standardi niso več kos.[13]

Vzporedno z industrijskim Ethernetom že 15 let obstaja najbolj razširjen in največkrat uporaben realnočasovni protokol PROFIBUS, za katerim stoji Siemens. Ker pa PROFIBUS prihaja na skrajno mejo svojih zmogljivosti, je Siemens razvil nov protokol PROFINET, ki združuje PROFIBUS in Ethernet. Velike hitrosti prenosa podatkov pri Ethernetu odpirajo nove možnosti za realnočasovne komunikacije. [13]



Slika 10: Zgradba PROFINET-a

(Vir: <https://dk.um.si/IzpisGradiva.php?id=1812&lang=eng>)

4 IZBIRA KOMPONENT

Za potrebe raziskovalne naloge smo potrebovali kamero, ki bi bila zmožna zaznati robove steklenic, izmeriti njeno višino in širino ter te podatke poslati krmilniku preko Ethernet omrežja. Po pregledu trga smo se odločili za pametno kamero FQ2.

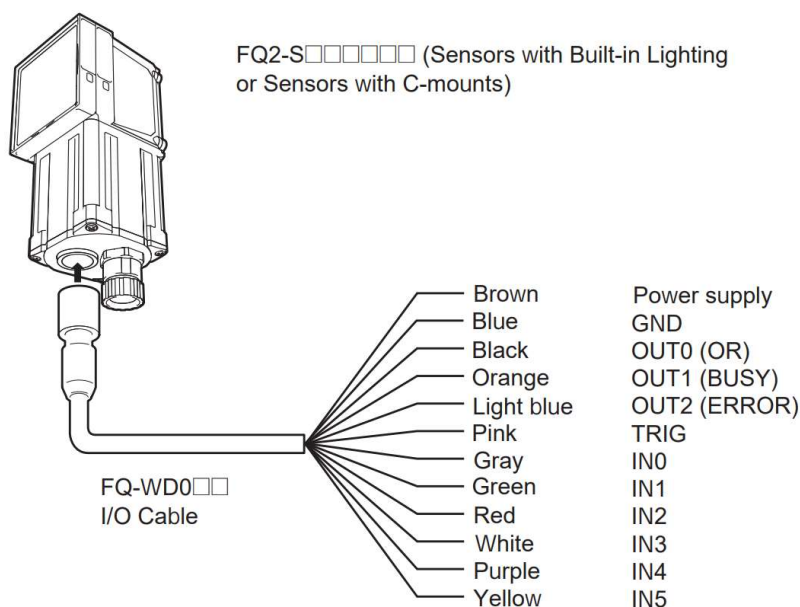
4.1 PAMETNA KAMERA FQ2

Pametne kamere FQ2 podjetja Omron so zanesljive in kompaktno kamere s številnimi funkcijami. Potrebujejo napajanje 24 V enosmernega toka. [4]

Pri raziskovanju smo uporabili model kamere FQ2-S45100N. Ta model ima vgrajeno belo bliskavico in ga lahko fokusiramo na predmete, oddaljene med 32 in 380 mm. [4]

Fokus se nastavlja z vijakom na vrhu kamere.

Na spodnji strani ima kamera priključek za Ethernet in skupek žic. Rjava in temno modra sta namenjeni napajanju kamere, sinal na rozasti žici sproži fotografiranje, črna, oranžna in svetlo modra pa so namenjene signaliziranju stanja kamere in se lahko reprogramirajo. [4]

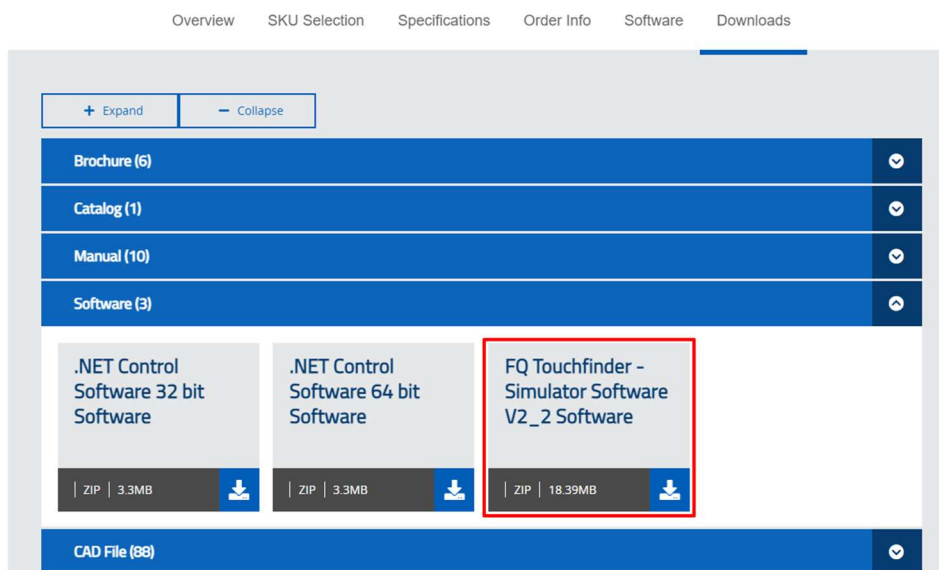


Slika 11: Žice, povezane s kamero

(Vir: Uporabniški priročnik za kamere FQ2, posnetek zaslona)

4.1.1 Programiranje kamere

Za programiranje kamere FQ2 je možno uporabljati zaslon na dotik, ki je v kompletu s kamero, ali simulator tega zaslona, ki je na voljo za prenos na spletni strani <https://automation.omron.com/en/us/products/family/FQ2>, pod zavihkom Downloads. Pri našem delu smo uporabljali oba načina.



Slika 12: Prenos programa FQ Touchfinder Simulator

(Vir: <https://automation.omron.com/en/us/products/family/FQ2>, posnetek zaslona)

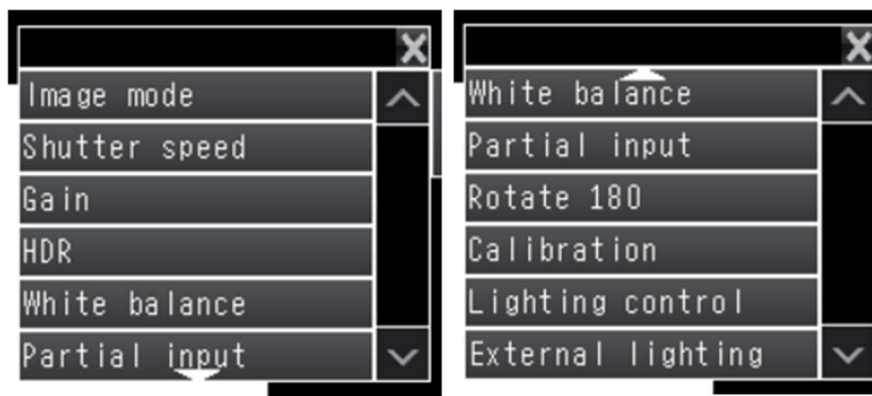
Da se zaslon ali računalnik s simulacijo zaslona lahko poveže s kamero, morata biti na istem omrežju. Ko je povezovanje uspešno končano, se na zaslonu odpre način za urejanje nastavitev kamere in programa (angl. setup način).

4.1.1.1 Zavihek »Image«

S pritiskom na gumb »Camera setup« lahko s predogledom v živo spreminjamo nastavitve fotografiranja, kot so:

- popravljanje razmerja barv;
- fotografiranje samo dela vidnega polja kamere, s čimer zmanjšamo skupen čas fotografiranja;
- orientacija fotografije;
- hitrost zaklopa, s katero spreminjamo svetlost fotografije.

V tem meniju lahko tudi kalibriramo piksele v nam prijaznejše enote, v katerih kamera krmilniku pošlje meritve.



Slika 13: Zavihek »Image«

(Vir: osebni arhiv)

V meniju »Trigger setup« spreminjamo zakasnitev fotografiranja po prejetju signala.



Slika 14: Zavihek »Trigger«

(Vir: osebni arhiv)

V meniju »Image adjustment« lahko fotografijo obdelamo s filtri.



Slika 15: Zavihek »Image adjustment«

(Vir: osebni arhiv)

4.1.1.2 Zavihek »Inspect«

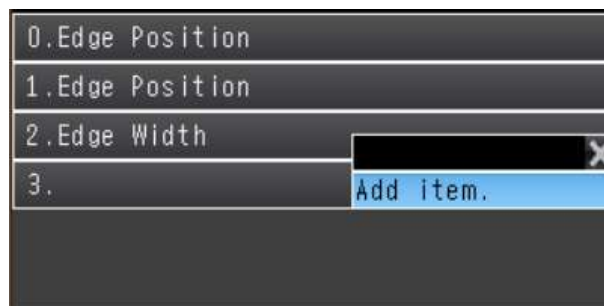
Pod zavihkom »Inspect« najdemo tri gumbе: »Inspection«, »Calculation« in »Retry details«.



Slika 16: Zavihek »Inspect«

(Vir: osebni arhiv)

S klikom na gumb »Inspection« se nam prikaže seznam meritev, ki se bodo izvedle po vsakem fotografiranju. Na tega lahko s klikom na prazno mesto dodamo nove meritve, s klikom na obstoječo meritev pa lahko to spreminjamo, izbrišemo, kopiramo ali preimenujemo.



Slika 17: Dodajanje meritev

(Vir: osebni arhiv)

Kamera je sposobna branja črtnih in QR kod ter iskanja poljubnih oblik, vendar smo pri našem raziskovanju uporabljali le funkcionalnosti za iskanje robov in merjenje širin predmetov.

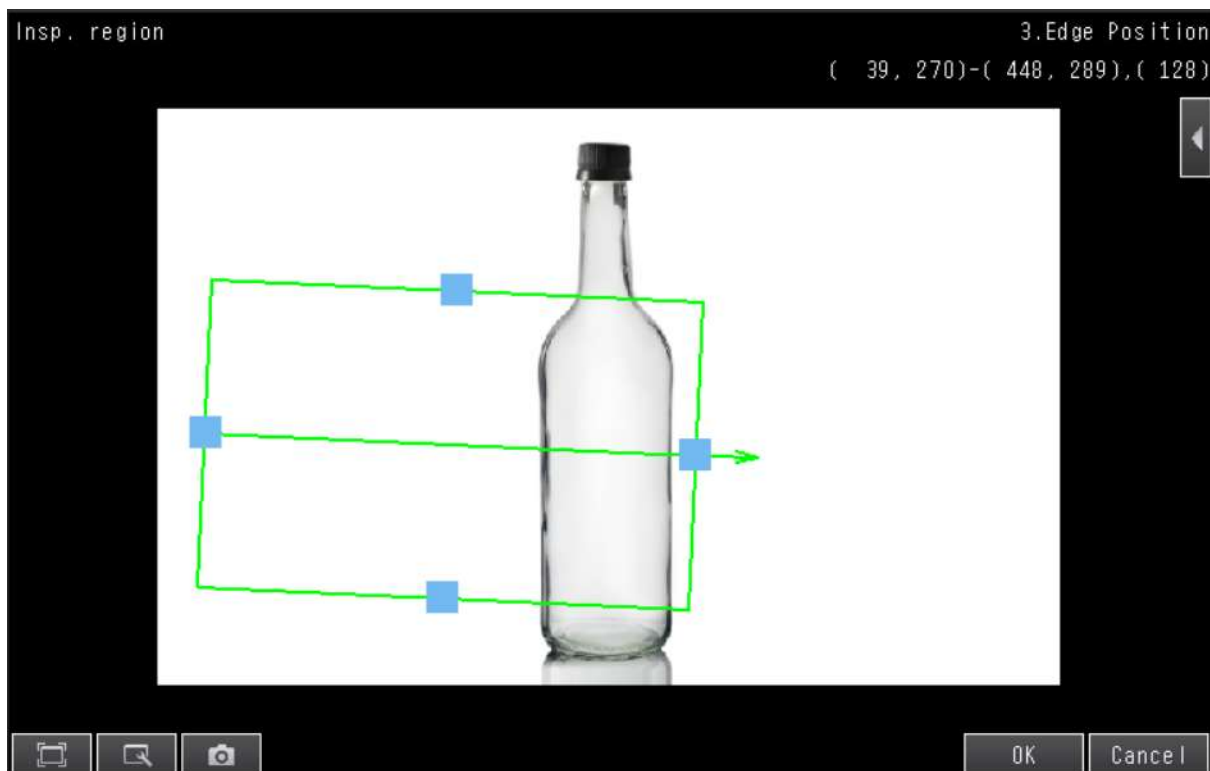


Slika 18: Možnosti zaznavanja

(Vir: osebni arhiv)

Ko izberemo »Edge Position« ali »Edge Width«, se nam prikažeta zavihka »Settings« in »Details«.

Z gumbom »Teach« pod zavihkom »Settings« določimo velikost in pozicijo merilnega območja ter smer, v kateri se bo meritev izvajala. Za potrditev postavitve pritisnemo gumb »OK«. Sedaj lahko z gumbom desno spodaj preverimo, ali se meritve na fotografiji, ki je bila nazadnje posneta, opravijo pričakovano.



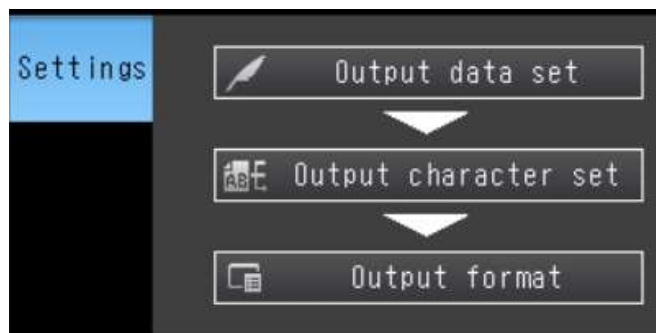
Slika 19: Zaznavanje robov

(Vir: osebni arhiv)

Če algoritmi za iskanje robov teh ne najdejo ali jih ne določijo pravilno, imamo možnost spremeniti nekatere njihove parametre. To lahko storimo na prejšnjem meniju pod zavihkom »Details«.

4.1.1.3 Zavihek »In/Out«

Pod zavihkom »In/Out« nastavljamo podatke, ki jih kamera pošilja drugim napravam. To storimo tako, da najprej pritisnemo »I/O setting«, nato »Output data setting« in izberemo protokol, ki ga nameravamo uporabljati za pošiljanje podatkov. Sedaj se nam prikaže meni s tremi gumbi. S pritiskom na gumb »Output data set« izberemo podatke, ki jih želimo poslati. Z gumbom »Output format« lahko spreminjamo število števk pred in po decimalni piki, simbol pred negativnimi števili, ali se pred številko izpišejo ničle, kateri simbol ločuje številke ipd.



Slika 20: Zavihek »In/Out«

(Vir: osebni arhiv)

Output form	ASCII	^
Digits of integer	6	
Digits of decimal	4	
Negative	-	
0 suppress	No	
Field separator	OFF	v

Slika 21: »Output« format

(Vir: osebni arhiv)

4.1.2 Nastavitev protokola za komuniciranje z drugimi napravami

Protokol za komuniciranje z drugimi napravami je mogoče nastaviti v načinu »Setup« s klikom na gumb desno spodaj, »Sensor settings« in »Data output«. Na voljo so »No-protocol«, »Link« in »Fieldbus« protokoli.

Te nastavitve bodo začele veljati šele po ponovnem zagonu kamere, ampak preden storimo to, jih moramo shraniti tako, da zapustimo meni »Sensor settings«.

4.1.3 Ozadje predmeta

Ozadje predmeta, ki ga fotografiramo, mora za dobre rezultate biti enobarvno in ne sme biti pretemno, da je fotografiranje hitro, ali tako svetlo, da zaklop tudi na najhitrejši nastavitvi zajame preveč svetlobe.



Slika 22: Primer pretemnega ozadja

(Vir: osebni arhiv)



Slika 23: Primer presvetlega ozadja

(Vir: osebni arhiv)

4.2 KRMILNIK

Za sprejemanje, shranjevanje in obdelavo podatkov, ki jih zajame kamera, smo se odločili uporabiti programirljivi logični krmilnik. PLK je krmilni element, ki delovanje stroja v tehnološkem procesu sproži, nadzoruje, nanj vpliva in ga na definiran način tudi zaključí. Ker je PLK mikroprocesorsko krmiljen sistem, ima program sicer shranjen v pomnilniku, vendar ga lahko vedno tudi spremenimo. [1]

Za potrebe naše naloge smo se odločili za Siemensove krmilnike linije S7, saj smo te krmilnike že programirali pri pouku v lanskem šolskem letu. Izbirali smo med dvema krmilnikoma, to sta S7-300 in S7-1200.

Potrebe, ki jih mora izpolnjevati naš krmilnik, so:

- možna povezava s kamero preko Etherneta,
- dovolj velik spomin za shranjevanje vseh izmerjenih podatkov,
- kompaktnost.

Po testiranju obeh tipov smo se odločili za novejši S7-1200. Čeprav oba tipa izpolnjujeta zahtevane pogoje, nam S7-1200 omogoča lažjo vzpostavitev povezave s kamero. Prav tako je tudi manjših dimenzij, kar je še dodatno vplivalo na našo izbiro.

4.2.1 Siemens S7-300

Zaradi modularne zasnove brez ventilatorja, enostavne izvedbe in priročnega rokovanja je SIMATIC S7-300 stroškovno učinkovita in uporabniku prijazna rešitev za najrazličnejše naloge v nizkih in srednjih zmogljivostih. Meri 125 x 120 x 115 mm. Deluje na napetost 24 V DC in vsebuje 24 digitalnih vhodov, 16 digitalnih izhodov, 4 analogne vhode ter 2 analogna izhoda. Le-ti nam omogočajo neposredno povezavo s procesorjem. Krmilnik lahko nadgradimo z največ 31 dodatnimi moduli, ki nam omogočajo dodatne vhode in izhode ali povezavo s krmilniki drugih proizvajalcev. [5]

S centralno procesno enoto (CPE) 313C nam omogoča predelavo 1 bita v 70 ns in 128 KB delovnega spomina. Za shranjevanje programa pa potrebujemo SIMATIC-ovo spominsko kartico (Micro Memmory Card), ki ima do 8MB spomina. [5]

Vmesnik RS 485 nam omogoča MPI povezavo PG/OP, Global data komunikacijo in S7 komunikacijo s hitrostjo prenosa podatkov do 187,5 kb/s. [5]



Slika 24: Krmilnik S7-300

(Vir: <https://mall.industry.siemens.com/mall/en/uk/Catalog/Product/6ES73135BF030AB0>)

Za potrebe našega sistema smo krmilniku dodali komunikacijsko-procesni modul CP 343-1, ki omogoča povezavo z industrijskim Ethernetom preko ISO in TCP/IP protokolov ali pa nastavitve PROFINET I/O kontrolerja ter naprave. To doseže z dvema integriranima Ethernet vmesnikoma ERTEC 200. [6]

4.2.2 Siemens S7-1200

Za vodenje naprav in procesiranje podatkov smo izbrali krmilnik S7-1200 s centralno procesno enoto (CPU) 1212C DC/DC/DC. Z dimenzijami 100 x 90 x 75 mm je zelo kompakten krmilnik, kar nam omogoča enostavno in prostorsko učinkovito inštalacijo. Za delovanje potrebuje napetost 24 V DC. Ker vsebuje 8 digitalnih vhodov, 6 digitalnih izhodov in 2 analogna vhoda, je primeren za našo aplikacijo, saj ne potrebujemo veliko digitalnih vhodov in izhodov. [7]

CPU 1212C je zmožen procesirati bit informacije v 80 ns, s 75 KB delovnega spomina in 2 MB integriranega spomina, ki ga lahko razširimo s SIMATIC-ovo spominsko kartico (Micro Memory Card), ki imajo do 8 MB spomina. [7]

Vsebuje tudi 1 RJ 45 Ethernet vmesnik, ki omogoča naslednje tipe povezave:

1. PROFINET IO kontroler
2. PROFINET IO naprava
3. SIMATIC komunikacija
4. Open IE komunikacija
5. Web server

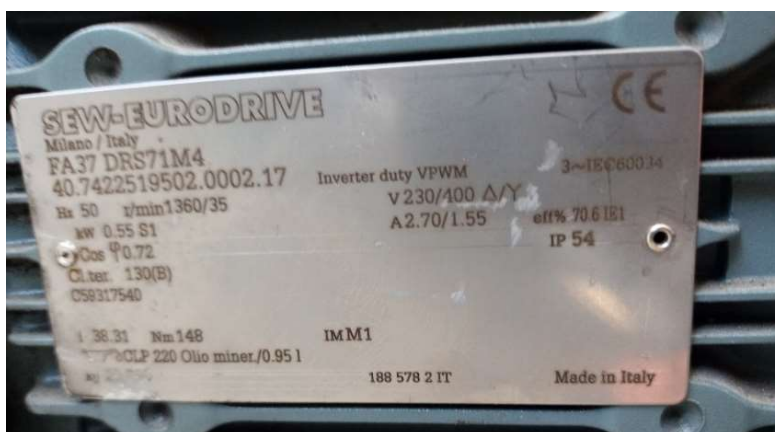


Slika 25: Krmilnik S7-1200

(Vir: <https://support.industry.siemens.com/cs/pd/72709?ptd=td&dl=en&lc=en-DE>)

4.3 MOTOR/TEKOČI TRAK

Tekoči trak smo dobili iz steklarne, saj imajo zanj že pripravljen prostor. Poganja ga trifazni motor proizvajalca Sew-Eurodrive z močjo 550 W. Motor smo priključili v vezavo zvezda. Na tekoči trak je montiran tudi nosilec za kamero in pločevinasta komora, ki nam pomaga pri vzdrževanju konstantne svetlobe in zagotavljanju natančnega ter zanesljivega slikanja kamere.



Slika 26: Motor

(Vir: osebni arhiv)



Slika 27: Tekoči trak

(Vir: osebni arhiv)

4.4 FREKVENČNIK

Za pogon in upravljanje našega motorja smo izbrali frekvenčnik SIEMENS SINAMICS G120C PN. Ta nam omogoča spreminjanje parametrov delovanja motorja, kot sta hitrost in smer vrtenja. Največja dovoljena moč motorja je 750 W, pri čemer tok ne sme presegati 2,2 A. Frekvenčnik lahko krmilimo preko Ethernetne povezave s protokolom PROFINET ali pa uporabimo potenciometer, ki ga priključimo na analogne vhode. [8]



Slika 28: Frekvenčnik

(Vir: <https://support.industry.siemens.com/cs/products/6sl3210-1ke12-3af1/sinamics-g120c-pn-0-75kw-fila?pid=252455&mlfb=6SL3210-1KE12-3AF1&mf=ps&lc=en-DK>)

4.5 HMI PANEL

Za prikazovalnik smo izbrali Simatic HMI KTP 400 Basic zaslon. Ima 4-palčni zaslon, z možnostjo upravljanja na dotik. Uporabljamo lahko tudi štiri tipke, ki se nahajajo pod ekranom. Za delovanje potrebuje 24 V DC napajanje. Vsebuje tudi RJ45 Ethernet vmesnik, ki nam omogoča povezavo preko industrijskega Etherneteta s protokolom PROFINET. [9]



Slika 29: HMI zaslon

(Vir: <https://support.industry.siemens.com/cs/products/6av2123-2db03-0ax0/simatic-hmi-ktp400-basic?pid=379924&ntp=Manual&mlfb=6AV2123-2DB03-0AX0&mfh=ps&lc=en-ID>)

4.6 ETHERNET STIKALO

Za povezavo vseh naših komponent potrebujemo Ethernet stikalo, s katerim lahko povežemo več naprav v eno omrežje. Mi smo uporabili Siemensovo stikalo SIMATIC NET CSM 1277. S štirimi RJ45 vmesniki nam omogoča povezavo krmilnika S7-1200 s še tremi drugimi napravami, s hitrostjo do 100 Mb/s. Za delovanje potrebuje 24 V DC napajanje. [10]



Slika 30: Ethernet stikalo

(Vir: <https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6GK7277-1AA10-0AA0>)

4.7 NAPAVALNIK

Ker vse naše komponente potrebujejo za delovanje 24 V DC, smo se odločili uporabiti Siemensov napajalnik PM 1207. Napajalnik lahko priključimo na 120/230 V AC. Na izhodu pa dobimo 24 V DC z maksimalnim tokom 2.5 A. [11]



Slika 31: Napajalnik

(Vir: <https://mall.industry.siemens.com/mall/en/ww/Catalog/Product/6EP1332-1SH71>)

4.8 PROGRAMSKA OPREMA

4.8.1 TIA Portal

Pri raziskovalni nalogi smo za programiranje PLK-ja uporabljali program TIA Portal V14, ki je namenjen programiranju Siemensovih PLK-jev. Z njim nastavljamo parametre PLK-ja, pišemo programske funkcije in tudi simuliramo program. Omogoča nam programiranje v različnih programskih jezikih, kot so FBD, LAD, STL in SCL. Predvsem najbolj koristna funkcija pa je pregled in prisila parametrov v realnem času, kar nam pomaga pri pregledovanju in zagotavljanju pravilnega delovanja programa.

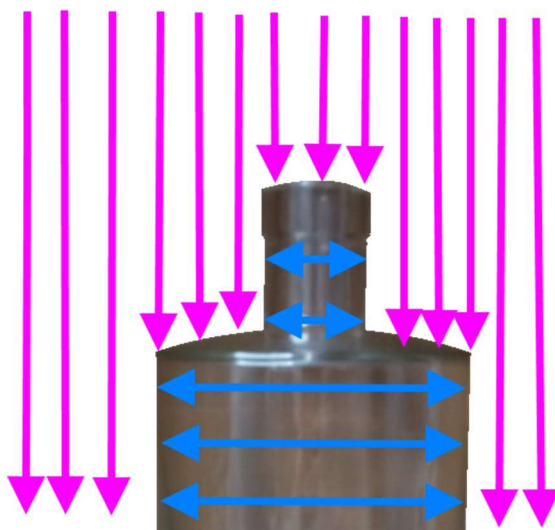
TIA Portalu smo dodali tudi enega izmed mnogih modulov, SINAMICS Drive, ki nam zagotavlja enostavno in hitro programiranje ter nastavljanje našega frekvenčnika.

Tia Portal lahko prenesemo iz Siemensove spletne strani, ali pa naročimo fizično kopijo. Brezplačno licenco lahko uporabljamo do 14 dni, nato jo moramo kupiti. [12]

5 PROGRAM KAMERE

K reševanju tega problema smo pristopili z zamisljivo, da bi steklenico fotografirali in izmerili njeno širino na različnih y-koordinatah ter njeno višino na različnih x-koordinatah. Nato bi primerjali izmerjene vrednosti z dejanskimi merami vseh steklenic in poiskali tisto, katere mere se najbolj ujemajo.

Odločili smo se uporabiti 15 vertikalnih in 5 horizontalnih mer. Za vertikalne smo uporabili funkcijo kamere »Edge Position«, za horizontalne pa »Edge Width«. Po fotografiranju smo najprej poslali vse vertikalne, nato pa še vse horizontalne mere, ločene z vejicami. Če je meritev bila neuspešna, se je poslala številka 0.



Slika 32: Vertikalne in horizontalne meritve

(Vir: osebni arhiv)

Čeprav bi enote, v katerih smo merili dimenzije steklenic, lahko kalibrirali na centimetre ali milimetre, tega nismo naredili, saj to za razlikovanje med oblikami steklenic ni bilo potrebno. Edina slabost tega je bila, da so bile enote meritev v pikslih, kar je bilo nam manj intuitivno.

6 POVEZAVA KOMPONENT

6.1 IP NASLOVI

Ker naprave povezujemo na Ethernetno omrežje, moramo vsaki napravi določiti naslov IP. Te naslove si lahko izbiramo sami. Za naslov našega omrežja smo uporabili 10.5.5.X z masko 255.255.255.0, kjer 255 maskira omrežje, 0 pa napravo v omrežju. Nato smo nastavili tudi IP naslove naslednjih naprav:

1. PLK – 10.5.5.100
2. Prenosni računalnik – 10.5.5.101
3. HMI – 10.5.5.103
4. Frekvenčnik – 10.5.5.104
5. Kamera – 10.5.5.151
6. FQ2 Touch finder – 10.5.5.152

6.2 PROTOKOLI POVEZAV

Komponente smo povezali na dva različna načina:

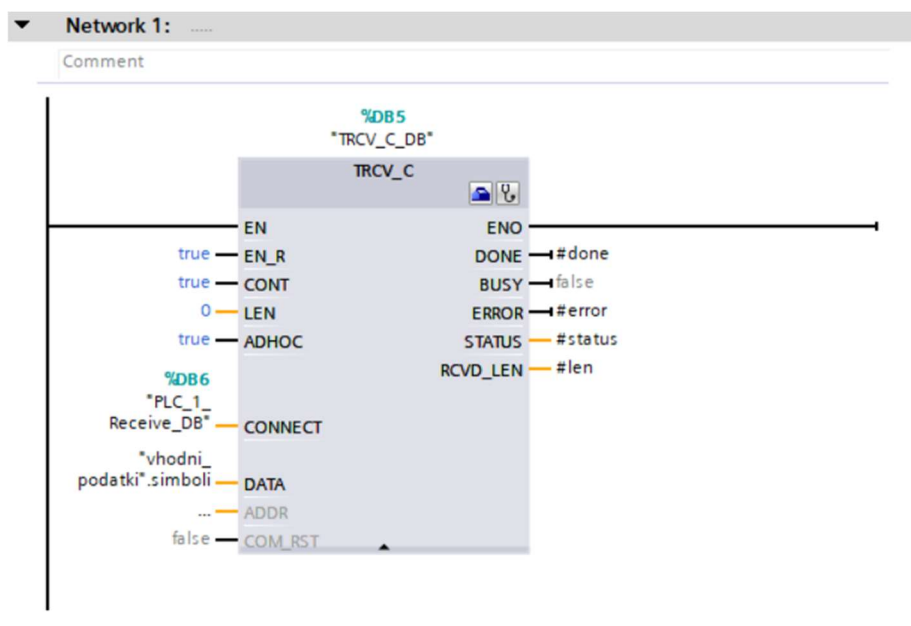
1. Vse Siemensove komponente (PLK, Frekvenčnik, HMI) smo med seboj povezali s protokolom PROFINET, saj potrebujemo sprejemanje in pošiljanje podatkov v realnem času. Ker so vse komponente Siemens, ni potrebno dodatno programiranje in parametriranje povezave. Komponente preprosto dodamo v TIA portal in jim določimo povezavo.
2. Za povezavo krmilnika in kamere smo uporabili no-protocol TCP povezavo, ki je za razliko od PROFINETA le na 4. plasti komunikacije. To povezavo smo izbrali, ker potrebujemo le prejemanje podatkov iz kamere. Nastavljanje takšne povezave je tudi precej lažje od nastavljanja PROFINET povezave.

6.2.1 Nastavitve povezave kamere in krmilnika

Za vzpostavitev no-protocol TCP povezave smo uporabili funkcijski blok TRCV_C, ki združuje funkcionalnost blokov TCON, TRCV in TDISCON.

Pri krmilniku smo nastavljali naslednje parametre.

- EN – kdaj želimo vzpostaviti povezavo.
- EN_R – kdaj želimo prejemanj podatke.
- CONT – ali želimo, da je povezava neprekinjena.
- LEN – največja dolžina prejetih podatkov. Če je 0, funkcijski blok to nastavi samodejno, glede na parameter DATA.
- ADHOC – uporabimo, kadar imamo TCP povezavo.
- CONNECT – podatkovni blok, ki vsebuje parametre povezave.
- DATA – podatkovni blok, v katerega sprejemamo podatke.



Slika 33: TRCV_C

(Vir: osebni arhiv)

Nastavili smo tudi naslednje parametre v podatkovnem bloku za povezavo:

1. PLK, ki ga povezujemo.
2. Partner, ki ga povezujemo.
3. Tip povezave.
4. Vzpostavitev aktivne povezave.
5. Vrata povezave.

Connection parameter

General

Local	Partner
End point: PLC_1 [CPU 1212C DC/DC/DC]	Unspecified
Interface: PLC_1, PROFINET interface_1[X1 : PN(LAN)]	
Subnet: PN/IE_1	
Address: 10.5.5.100	
Connection type: TCP	
Connection ID (dec): 1	
Connection data: PLC_1_Receive_DB	
<input type="radio"/> Active connection establishment	<input type="radio"/> Active connection establishment

Address details

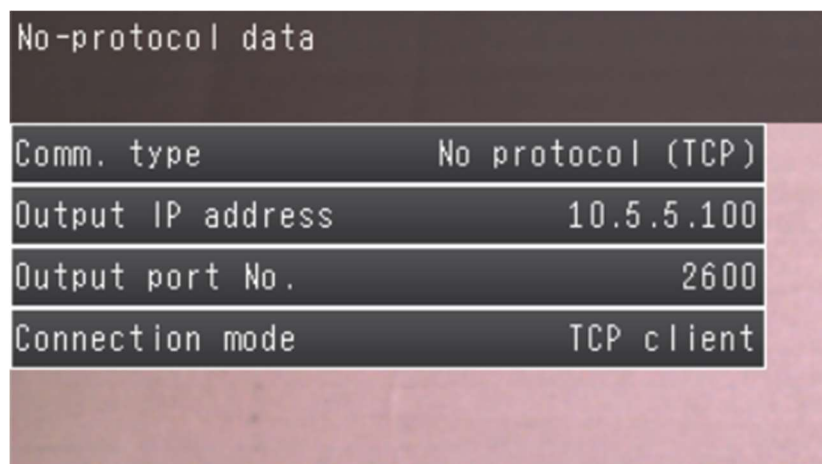
Local Port	Partner Port
Port (decimal): 2600	

Slika 34: TRC_V parametri

(Vir: osebni arhiv)

Nazadnje pa smo še na kameri nastavili naslednje podatke.

1. Tip komunikacije.
2. Izhodni IP naslov.
3. Izhodna vrata povezave.
4. Način povezave.

A screenshot of a camera's configuration menu. The background is dark with white text. At the top, it says "No-protocol data". Below that, there are four rows of settings, each with a label on the left and a value on the right.

No-protocol data	
Comm. type	No protocol (TCP)
Output IP address	10.5.5.100
Output port No.	2600
Connection mode	TCP client

Slika 35: Nastavitev komunikacije na kameri

(Vir: osebni arhiv)

7 PROGRAM KRMILNIKA

Na krmilniku shranjene mere steklenic na vsaki liniji smo se na različne načine lotili primerjati z merami, ki jih krmilnik prejme od kamere.

Med pisanjem programa smo zamenjali programski jezik, ki smo ga najprej uporabljali, FBD (angl. Function Block Diagram), za jezik SCL (angl. Structured Control Language). Z njim lažje ponavljamo ukaze s FOR in WHILE zankami ter razdelimo programe na več vej z IF pogojnimi stavki. Vse to nam je prišlo še posebej prav pri pretvorbi niza znakov, prejetih od kamere, v seznam realnih števil, pri računanju odstopanj steklenic in iskanju steklenice z najmanjšim odstopanjem.

7.1 PRISTOPI K RAČUNANJU Odstopanja DVEH STEKLENIC

Za računanje odstopanja mer pravkar fotografirane steklenice od mer vsake izmed shranjenih steklenic na linijah smo preizkusili štiri različne pristope.

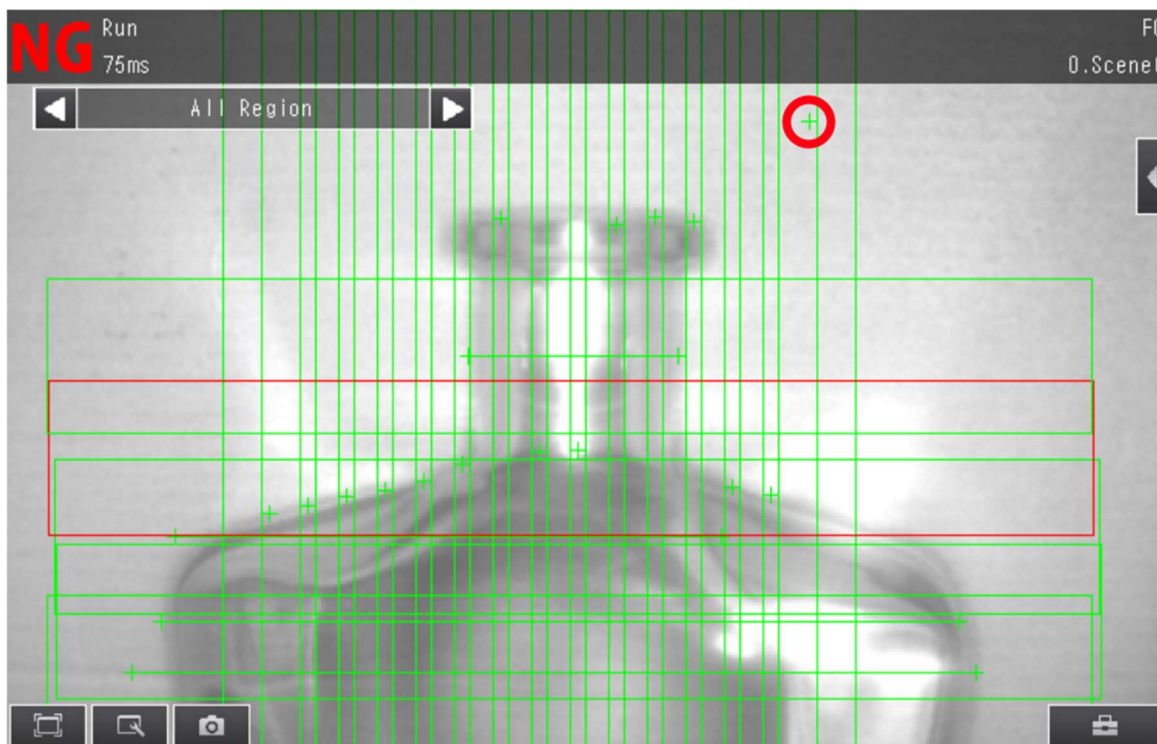
Pri testiranju zanesljivosti pristopov smo uporabljali tri steklenice, ki smo jih prejeli od steklarne, in lasten kozarec, da smo v programu uporabljali vse štiri virtualne linije.

7.1.1 Izračun odstopanja steklenice glede na največjo vertikalno mero

Najprej smo se odločili podatke obdelati na najpreprostejši način, ki smo se ga spomnili. Naš prvi pristop je bil računanje odstopanja na najvišji izmerjeni točki steklenic. Potem ko je krmilnik izmed vseh višinskih mer obeh steklenic poiskal največji, je odstopanje izračunal s spodaj navedeno formulo, kjer je o odstopanje med steklenicama, v in u pa sta najvišji točki steklenic.

$$o = |v - u| \quad (7.1)$$

Kot smo pričakovali, smo ugotovili, da lahko s to metodo zanesljivo prepoznamo prave steklenice različnih višin, ne glede na njihovo orientacijo in položaj. Problemi se pojavijo, če so steklenice podobnih višin, saj tako ne moremo zanesljivo vedeti, za katero steklenico gre. Razmišljali smo, da bi zaradi takšnih primerov dodali poziv delavcu za določitev steklenice, če se dve višini ne razlikujeta za več kot določeno vrednost. Ker smo pri tem načinu primerjali samo eno mero, je bilo treba poskrbeti za kakovostno fotografijo s čim bolj enobarvnim ozadjem. Če ozadje nima enakomerne barve, je večja možnost, da se najmanj ena višinska meritev izvede napačno, kar je prikazano na posnetku zaslona spodaj.



Slika 36: Primer fotografije s preslabim ozadjem, z obkroženo napačno določeno mero

(Vir: osebni arhiv)

Algoritem je pri določanju štirih vrst naključno postavljenih in orientiranih steklenic različnih velikosti pri 191 poskusih pravilno ugotovil vrste steklenic s 187 fotografij, torej je bila njegova uspešnost 98%.

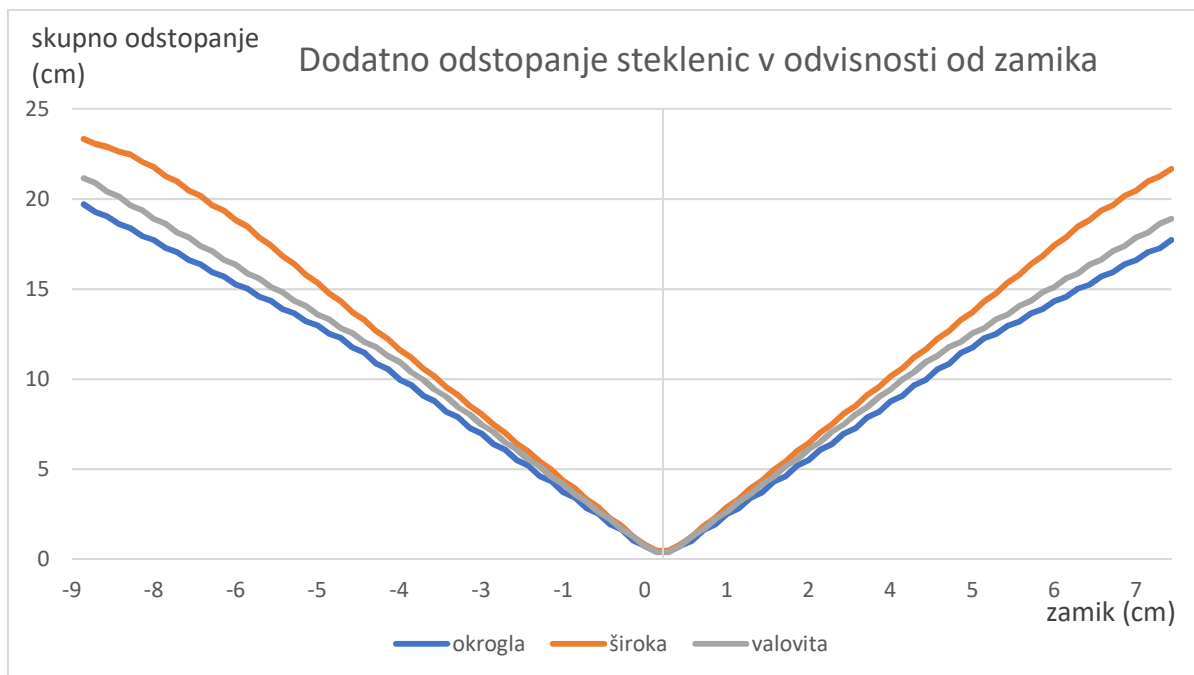
7.1.2 Izračun odstopanja steklenice glede na vsoto odstopanja posameznih vertikalnih in horizontalnih mer

Naš drugi pristop je bil računanje odstopanja vsake mere prve steklenice od mere z istim indeksom druge steklenice in seštetje vseh teh odstopanj. Odstopanja mer so se računala z enako formulo kot pri prejšnjem algoritmu (7.1).

$$o = \sum_{i=1}^n |v_i - u_i| \quad (7.2)$$

Pri zgornji formuli je o odstopanje, n število vertikalnih mer, v in u pa vertikalni meri prve in druge steklenice.

Pričakovali smo, da bo ta postopek zelo slab, saj se že, če steklenico malo premaknemo levo ali desno, odstopanja posameznih mer od shranjenih močno povečajo. Poleg tega se izmerjene horizontalne in vertikalne mere, ko steklenico vrtimo, vedno manj ujemajo s shranjenimi. V praksi se je, vsaj z našimi primeri steklenic, izkazalo, da njihovo premikanje levo in desno nima bistvenega vpliva na končno odločitev. Mislimo, da je do tega prišlo, ker se je pri teh steklenicah odstopanje pri vertikalnih merah, ki je nastalo zaradi horizontalnega premika, od vseh možnih izbir povečevalo skoraj enakomerno. To je razvidno s spodnjega grafa, ki smo ga kasneje ustvarili s simulacijo izvajanja meritev na modelih steklenic, ki smo jih uporabljali.



Graf 1: Odstopanje steklenice, zamaknjene levo ali desno od iste nezamaknjene steklenice

Težave nastopijo, če tloris steklenice ni krog. Izmerjene horizontalne in vertikalne mere se, ko steklenico vrtimo, vedno manj ujemajo s shranjenimi. Uspešnost tega algoritma omejujejo steklenice, ki pridejo pred kamero drugače obrnjene, kot so bile pri vnašanju podatkov.

Algoritem je pri določanju štirih vrst naključno postavljenih in orientiranih steklenic pri 60 poskusih pravilno ugotovil vrste steklenic s 47 fotografij, torej je bila njegova uspešnost 78%.

7.1.3 Izračun odstopanja steklenice glede na vsoto vertikalnih mer

Za tretji način računanja smo preizkusili, kakšne rezultate dobimo, če za odstopanje namesto vsote odstopanj posamičnih mer vzamemo odstopanje vsote posamičnih vertikalnih mer.

$$o = |(v_1 + v_2 + \dots + v_n) - (u_1 + u_2 + \dots + u_n)| \quad (7.3)$$

V tej formuli je o odstopanje steklenic, v predstavlja vertikalno mero prve steklenice, u vertikalno mero druge steklenice in n število vseh vertikalnih mer.

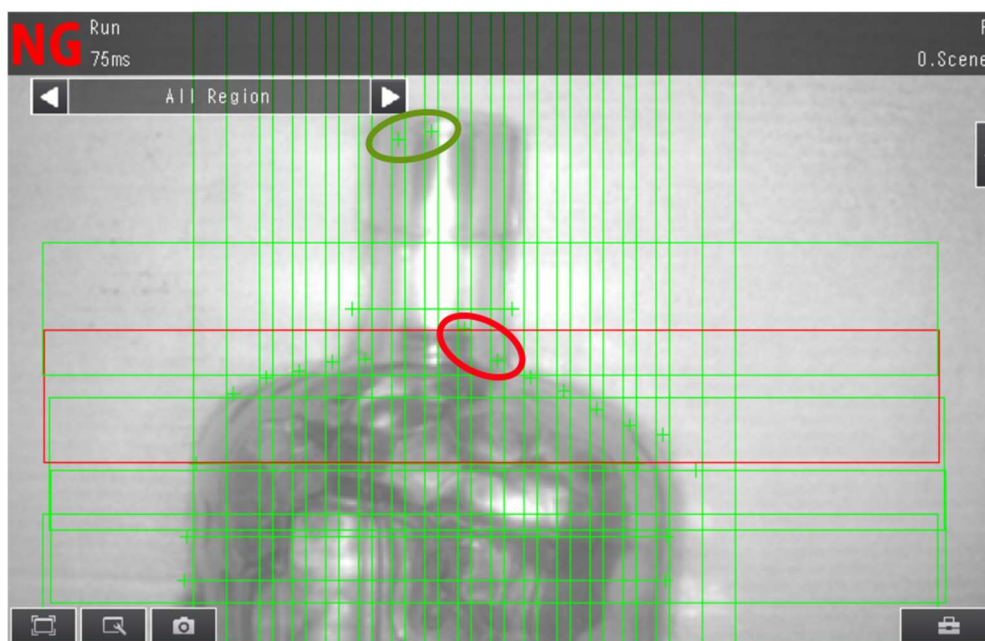
Ugotovili smo, da ta način ni zanesljiv v situacijah, kjer steklenice niso okrogle in vedno enako obrnjene, saj se z rotacijo neokrogle steklenice vsota višin drastično spreminja.

Pri določanju štirih vrst naključno postavljenih in orientiranih steklenic je algoritem pri 55 poskusih pravilno ugotovil vrste steklenic s samo 31 fotografij, torej je bila njegova uspešnost 56%.

7.1.4 Izračun odstopanja steklenice glede na vsoto odstopanja n-tih največjih vertikalnih mer

Za naslednji pristop smo se vrnili k našemu prvemu algoritmu, s katerim smo steklenice primerjali po največji vertikalni meri. Slabost tistega algoritma, da je bila lahko izbrana napačna steklenica zaradi samo enega previsoko izmerjenega roba, smo poskusili odpraviti z upoštevanjem več kot ene vertikalne mere. Odstopanje steklenic smo pri tem pristopu računali kot absolutno vrednost povprečja n-tih največjih vertikalnih mer.

Kot smo pričakovali, orientacija steklenice ni vplivala na rezultate. Včasih pa so se pri steklenicah z ozkimi vratovi in v posebnih primerih bleščanja stekla zgodile nepravilne meritve več največjih mer, kar je posledično pripeljalo do napačne kategorizacije steklenice. Spodnji posnetek zaslona prikazuje napačne meritve dveh od štirih vertikalnih mer zaradi bleščanja, ko je bil n nastavljen na 4.



Slika 37: Pravilni in napačni meritvi najvišjih točk steklenice

(Vir: osebni arhiv)

Tako je algoritem pri $n = 4$ pri določanju štirih vrst naključno postavljenih in orientiranih steklenic različnih velikosti pri 80 poskusih pravilno ugotovil vrste steklenic s 65 fotografij, torej je bila njegova uspešnost 81%.

Testiranje smo ponovili še z $n = 2$. Takrat je algoritem pri 80 poskusih uspešno ugotovil steklenice z 69 fotografij, z uspešnostjo 86 %.

7.2 DOLOČANJE STEKLENICE Z GLASOVANJEM RAZLIČNIH ALGORITMOV

Po testiranju vseh načinov določanja steklenic z odstopanjem smo se odločili, da še preizkusimo, kako bi deloval sistem, pri katerem bi več naših že napisanih algoritmov glasovalo za tisto steklenico, ki ima po njihovem najmanjše odstopanje od izmerjenih mer. Steklenica z največ glasovi bi bila izbrana za zmagovalko. Da bi lahko rešili situacije, v katerih bi več steklenic dobilo enako največje število glasov, smo vsakemu glasgu glede na znano uspešnost algoritma dodali težo, ki je v takšnih primerih odločala.

Uporabili smo algoritme, opisane v podpoglavjih 7.1.1, 7.1.2 in 7.1.4, s težami glasov 1, 0,8 in 0,9.

Upali smo, da bomo s takšnim pristopom uspeli kombinirati slabosti in prednosti teh algoritmov. Na koncu se je to zgodilo, vendar se je s takšnimi težami glasov nekajkrat pripetilo, da sta dva algoritma storila podobno napako in bila odgovorna za napačno določitev steklenice.

Pri določanju štirih vrst naključno postavljenih in orientiranih steklenic je algoritem pri 166 poskusih pravilno ugotovil vrste steklenic s 160 fotografij, torej je bila njegova uspešnost 96%.

7.3 PODATKOVNI BLOKI

V programu se srečamo z veliko količino podatkov, zato smo naš program uredili v posamezne podatkovne bloke. S tem smo dosegli boljšo organizacijo in lažje ter hitrejše pisanje programa.

V programu so naslednji podatkovni bloki:

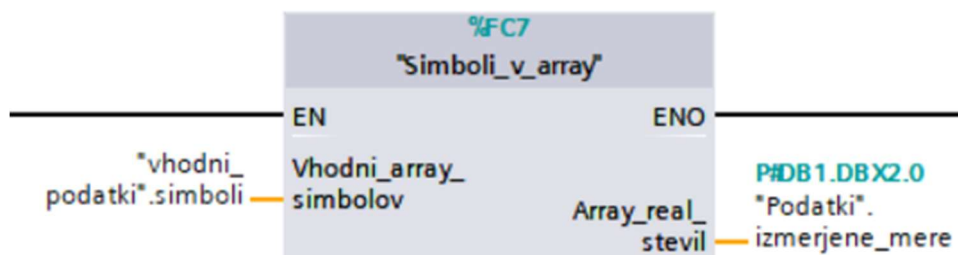
1. »Steklenice« – v tem bloku so shranjeni vsi podatki steklenic. Sestavljeni so iz dveh tabel. Prva se imenuje »Steklenice« in vsebuje množico podatkovnega tipa steklenice, ki smo ga ustvarili sami. Druga množica se imenuje »indeksi_steklenic_na_linijah« in je sestavljena iz podatkovnega tipa celih števil (angl. integers).
2. »Podatki« – v tem bloku so shranjeni nekateri splošni podatki. Tu so shranjene izmerjene mere, indeks steklenice z najmanjšim odstopanjem, glasovi, število oddanih glasov ter logična spremenljivka, namenjena nastavljanju mer.
3. »Vhodni_podatki« – v ta blok se shranjujejo vsi sprejeti podatki iz kamere. Shranijo se v tabelo (angl. array) karakterjev.
4. »Zaslon« – v to podatkovno bazo smo shranili vse pomembne spremenljivke, ki jih potrebujemo za nadzorovanje zaslona.

7.4 GLAVNI PROGRAM

Glavni program je sestavljen iz takoimenovanih funkcij. Te funkcije lahko napišemo sami, s pomočjo ukazov, ali pa uporabimo že narejene funkcije. Večino funkcij, ki smo jih uporabili v našem programu, smo napisali sami. Pri vseh funkcijah, ki implementirajo algoritme za prepoznavo prave steklenice, upoštevamo samo steklenice, ki so trenutno na linijah.

7.4.1 Seznam funkcij

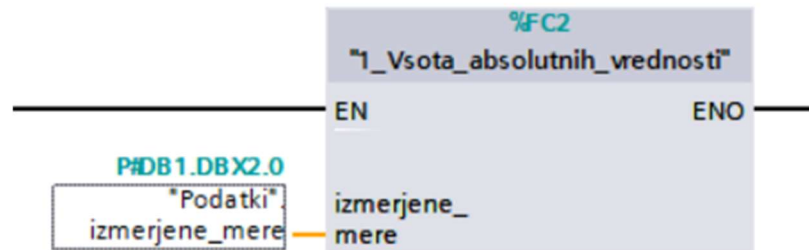
1. »Simboli_v_array« – s to funkcijo združimo vse podatke, ki jih pridobimo iz kamere in iz njih pridobimo potrebne podatke ter jih uredimo. Ker podatki v krmilnik pridejo kot tabela (angl. array) znakov, moramo te znake najprej združiti v nize znakov (angl. strings), nato pa vsak niz spremeniti v število in ga shraniti v drugo tabelo. Zaradi že vnaprej določenega števila podatkov je to veliko lažje.



Slika 38: »Simboli v array«

(Vir: osebni arhiv)

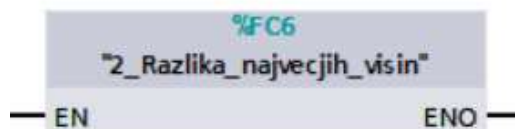
2. »1_Vsota_absolutnih_vrednosti« – s to funkcijo vse pridobljene podatke primerjamo s podatki steklenic na linijah, shranjenimi v podatkovnem bloku. Vsako vhodno mero primerjamo z ustrežno mero iz podatkovnega bloka in jih odštejemo. Nato absolutne vrednosti teh števil seštejemo in dobimo odstopanje, ki ga začasno shranimo v spremenljivko.



Slika 39: »Vsota absolutnih vrednosti«

(Vir: osebni arhiv)

3. »2_Razlika_najvecjih_visin« – s to funkcijo iz pridobljenih podatkov poiščemo največjo višino. Enako storimo za že shranjene podatke o steklenicah na linijah. Nato poiskano največjo višinsko mero pravkar fotografirane steklenice primerjamo z največjo višino vsake steklenice na linijah. Ta odstopanja ponovno shranimo v spremenljivko.



Slika 40: »Razlika največjih višin«

(Vir: osebni arhiv)

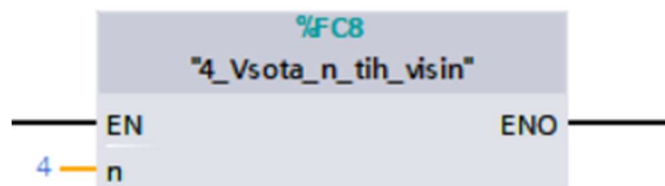
4. »3_Seštevek_vseh_visin« – s to funkcijo vse pridobljene podatke višin seštejemo. Ta rezultat nato primerjamo s seštevkom višinskih mer steklenic na linijah. Odstopanja ponovno shranimo v spremenljivko.



Slika 41: »Seštevek vseh višin«

(Vir: osebni arhiv)

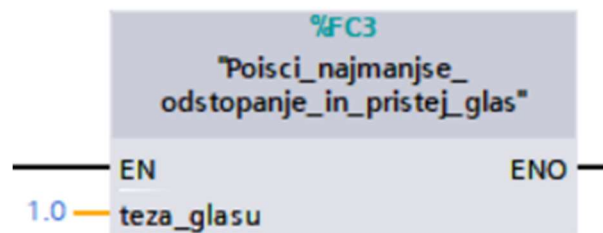
5. »4_Vsota_n_tih_visin« – s to funkcijo poiščemo n največjih višin. To storimo tako, da najprej tabelo vhodnih podatkov razvrstimo po velikosti. Nato izberemo prvih n podatkov. Izračunamo njihovo povprečno vrednost in enako storimo za podatke steklenic na linijah. Povprečne vrednosti nato primerjamo med seboj in odstopanje prav tako shranimo v spremenljivko. Pri tej funkciji lahko poljubno spreminjamo število n, v spodnjem primeru smo izbrali število 4.



Slika 42: »Vsota n-tih višin«

(Vir: osebni arhiv)

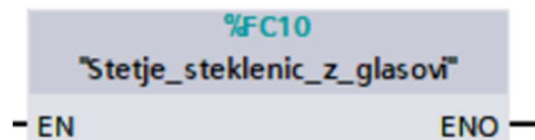
6. »Poišči_najmanjše_odstopanje_in_prištej_glas« – s to funkcijo najprej poiščemo steklenico na linijah z najmanjšim odstopanjem, iz spremenljivk, v katere smo shranili podatke prejšnjih funkcij. Tej steklenici prištejemo glas. To funkcijo uporabimo po vsaki funkciji, ki primerja mere. Določimo lahko tudi pomembnost glasu, tako da imajo nekatere funkcije v primeru izenačenja glasov prednost pri določanju prave steklenice.



Slika 43: »Poišči najmanjše odstopanje«

(Vir: osebni vir)

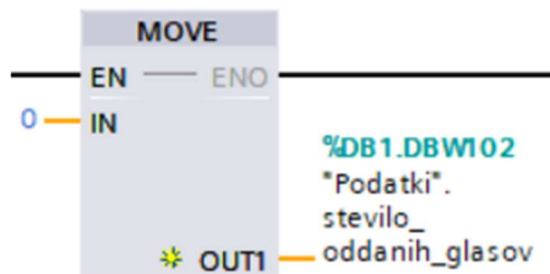
7. »Stetje_steklenic_z_glasovi« – ta funkcija poišče pravo steklenico s pomočjo glasov. Steklenica, ki ima največ glasov, je določena kot prava steklenice. Tej steklenici prištejemo 1.



Slika 44: »Štetje steklenic z glasovi«

(Vir: osebni vir)

8. »Move« – ta funkcija je že narejena. Omogoča nam prestavljanje vrednosti iz ene v drugo spremenljivko. Mi smo jo uporabili za ponastavljanje glasov.



Slika 45: »Move«

(Vir: osebni arhiv)

7.5 PROGRAM HMI

Program našega HMI-ja je sestavljen iz več zaslonov. Na vsakem zaslonu so različne tipke, s katerimi opravljamo različne funkcije.

Ko prižgemo naš sistem, se HMI zažene z začetnim zaslonom. Izbiramo lahko med spreminjanjem steklenic, štetjem steklenic in nastavitvami frekvenčnika. Imamo tudi možnost prisilno sprožiti kamero da slika, kar je bilo še posebej pomembno pri testiranju sistema.



Slika 46: Začetni zaslon

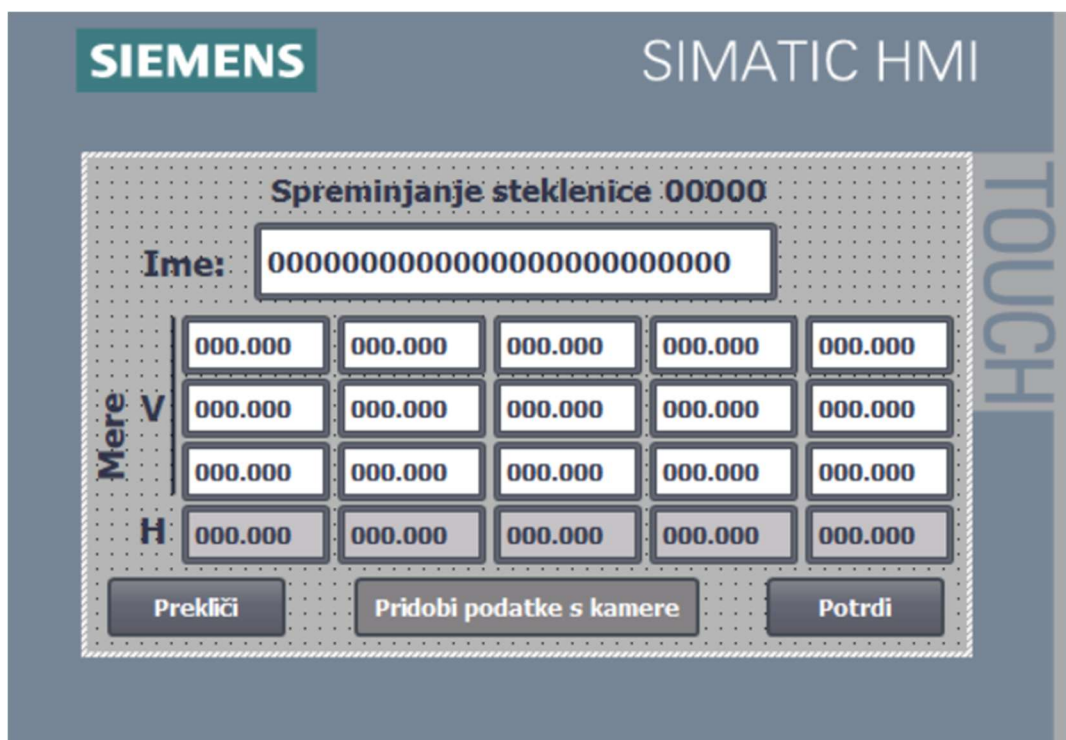
(Vir: osebni arhiv)

Če izberemo spreminjanje steklenic, pridemo do novega zaslona, kjer lahko izbiramo, katero steklenico želimo spremeniti. S pritiskom na gumb »potrdi« lahko izbiramo ime steklenice in njene podatke. Te podatke lahko vnašamo ročno ali pa jih pridobimo iz kamere. Podatke iz kamere lahko pridobimo na dva načina: tako, da zamenjamo vse že obstoječe mere ali tako, da zamenjamo samo mere, ki so nič, torej jih kamera v prejšnjem poskusu ni uspela izmeriti. S pritiskom na tipko »potrdi« shranimo vse mere.



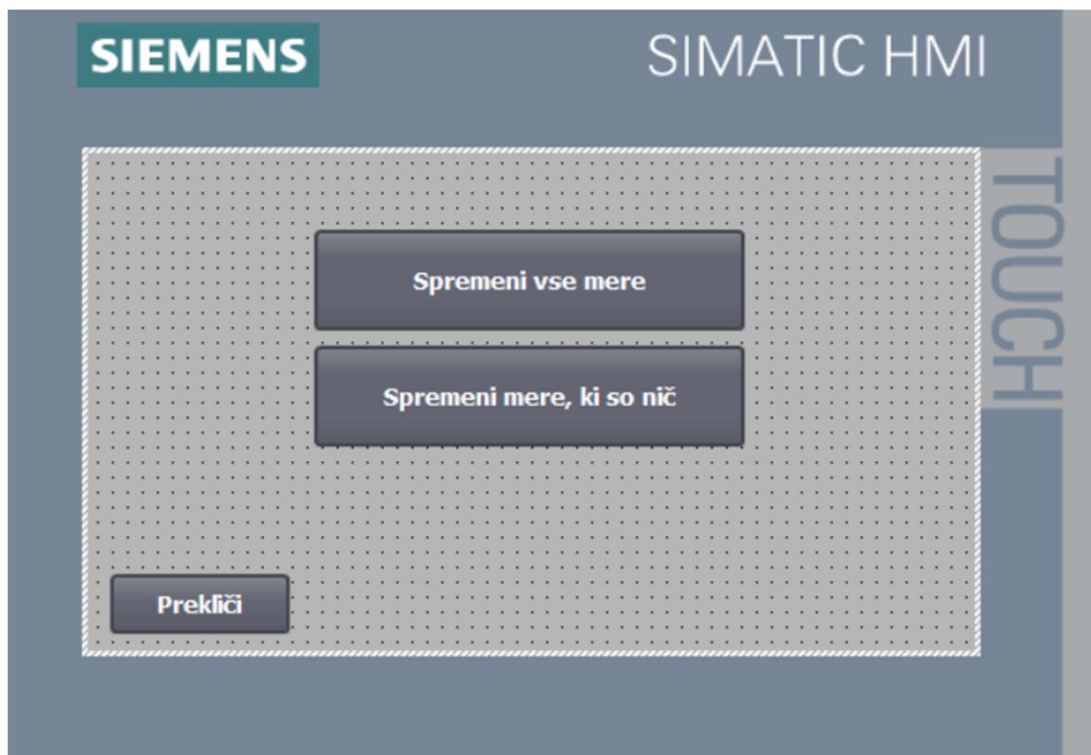
Slika 47: Spreminjanje steklenic

(Vir: osebni arhiv)



Slika 48: Spreminjanje mer steklenice

(Vir: osebni arhiv)



Slika 49: Spreminjanje mer s podatki iz kamere

(Vir: osebni arhiv)

Na začetnem zaslonu lahko izberemo tudi štetje steklenic. Na tem zaslonu imamo prikazane štiri linije. Ob vsakem števcu je tudi možnost za izbiro steklenice na tej liniji. Ta tipka nas privede do podobnega zaslona kot tipka za spreminjanje steklenic. Zraven števca je tudi možnost ponastavljanja števca. Na dnu ekrana ste še dve tipki, ki nam omogočata ponastavljanje vseh števcov ali vrnitev na začetni zaslon.



Slika 50: Štetje steklenic

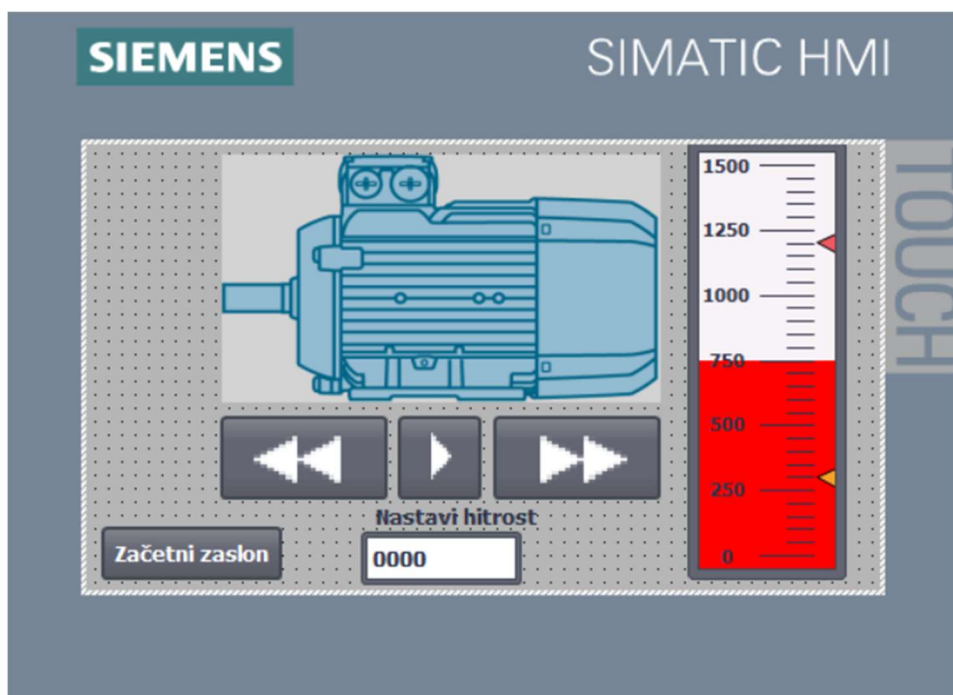
(Vir: osebni vir)



Slika 51: Izbira steklenice na liniji

(Vir: osebni vir)

Zadnja možnost na začetnem zaslonu je nastavljanje frekvenčnika. Tu lahko vklopimo ali izklopimo motor, spremenimo njegovo hitrost ali smer vrtenja. Na desnem delu zaslona je tudi števec, ki nam prikazuje trenutno nastavljeno hitrost.



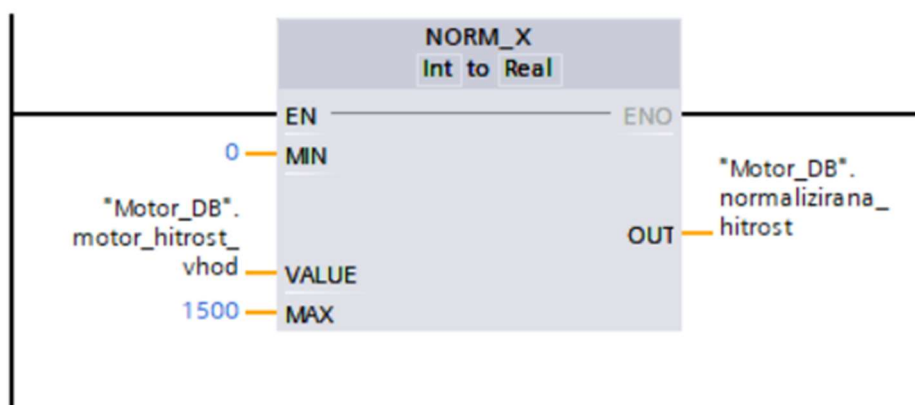
Slika 52: Nastavljanje frekvenčnika

(Vir: osebni arhiv)

7.7 UPRAVLJANJE MOTORJA

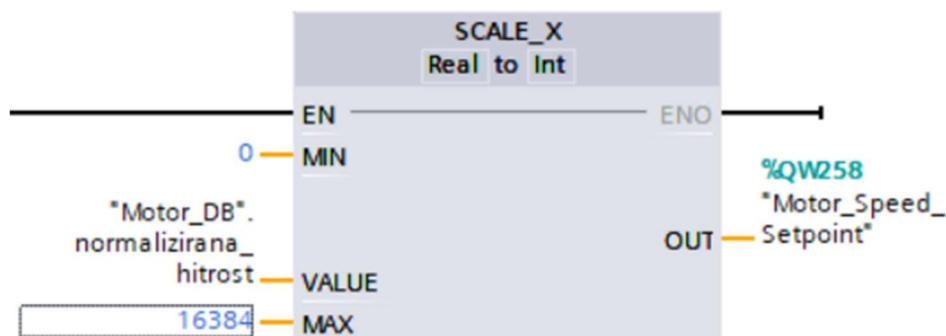
Naš motor upravljamo preko frekvenčnika. Programiranje tega frekvenčnika je precej enostavno. Pod zavihkom »zagon« (angl. commissioning) izberemo opcijo »commissioning wizzard«. Ta nam pomaga pri programiranju frekvenčnika. Vnesti moramo le podatke motorja, način upravljanja motorja in način komunikacije s krmilnikom.

Ko imamo nastavljene parametre frekvenčnika, moramo napisati le še preprost program. Uporabiti moramo funkciji »NORM_X« in »SCALE_X«. S temi funkcijami spremenimo dejansko hitrost motorja v odstotek maksimalne hitrosti motorja, ki jo nato posredujemo frekvenčniku. Uporabimo tudi dve preprosti stikalni funkciji za vklop in izklop motorja ter zamenjavo smeri.



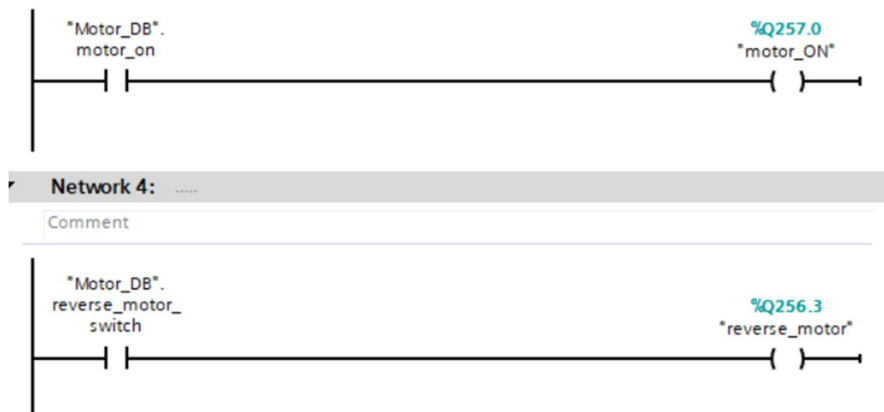
Slika 53: »NORM_X«

(Vir: osebni vir)



Slika 54: »SCALE_X«

(Vir: osebni vir)



Slika 55: Upravljanje motorja

(Vir: osebni vir)

7.8 OPTIMIZACIJA PROGRAMA

Za lažje in hitrejšje programiranje smo naš program optimizirali. To smo naredili s pomočjo podatkovnih tipov, ki smo jih ustvarili sami.

Prvi podatkovni tip, ki smo ga ustvarili, je bil »Steklenica«. Sestavljen je iz tabele mer, odstopanja, imena in števila steklenic. S pomočjo tega podatkovnega tipa lažje dodajamo in spreminjamo steklenice, hkrati pa je naše beleženje podatkov preglednejše in bolj urejeno.

Steklenica							
	Name	Data type	Default value	Accessible f...	Writa...	Visible in ...	Setpoint
1	mere	Array[1..*STE...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	mere[1]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	mere[2]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	mere[3]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	mere[4]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	mere[5]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	mere[6]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	mere[7]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	mere[8]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	mere[9]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	mere[10]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	mere[11]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	mere[12]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	mere[13]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	mere[14]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	mere[15]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	mere[16]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18	mere[17]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19	mere[18]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
20	mere[19]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
21	mere[20]	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
22	odstopanje	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
23	ime	String[*MAX_DOLZI...	"	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
24	stevilo	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Slika 56: Podatkovni tip »Steklenica«

(Vir: osebni arhiv)








Drugi podatkovni tip je »Glas«. Ta je sestavljen iz indeksa linije, na kateri je steklenica, ki ji je bil oddan glas, in teže glasu. S tem podatkovnim tipom si pomagamo pri beleženju različnih glasov.

Glas			
	Name	Data type	Default value
1	indeks_linije	UInt	0
2	teza_glasu	Real	0.0
3	<Add new>		

Slika 57: Podatkovni tip »Glas«

(Vir: osebni arhiv)

Za lažje spreminjanje programa smo ustvarili tudi nekaj konstant. Te konstante lahko uporabimo kjerkoli v programu. Pomagajo nam pri hitrejšem spreminjanju kode. Namesto da bi neko številko pisali večkrat v programu, jo lahko zamenjamo s konstanto. Če želimo to število zamenjati, lahko spremenimo samo vrednost konstante, namesto da popravljamo celoten program.

Konstante			
	Name	Data type	Value
1	 STEVILO_MER	UInt	20
2	 STEVILO_STEKLENIC	UInt	10
3	 MAX_DOLZINA_IMENA_STEKLENICE	UInt	25
4	 STEVILO_NAVPICNIH_MER	UInt	15
5	 STEVILO_POLOZNIH_MER	UInt	5
6	 STEVILO_GLASOV	UInt	3
7	<input type="text" value="<Add new>"/>		

Slika 58: Konstante

(Vir: osebni vir)

7.9 DIAGNOZA PROGRAMA

Za diagnozo programa smo uporabili dve funkciji »MOVE«. Obe funkciji nam pomagata pri spremljanju funkcije »TRCV_C«. Ker so nekateri podatki vidni samo en cikel, jih ne moremo videti pri nadzoruvanju programa v živo. Prva se sproži, ko je »error« izhod funkcije »TRCV_C« enak ena. V tem primeru je prišlo do napake pri povezavi ali pri prejemanju podatkov. Zato nam funkcija »MOVE« shrani status v novo spremenljivko, da jo lahko vidimo. Druga se sproži, ko je status »Done« enak ena. V tem primeru je »TRCV_C« sprejel podatke, mi pa s funkcijo »MOVE« shranimo dolžino prejetih podatkov.



Slika 59: Diagnoza »error« stanja

(Vir: osebni vir)



Slika 60: Diagnoza dolžine vhoda

(Vir: osebni vir)

8 OVREDNOTENJE HIPOTEZ

- Hipoteza 1 – Z ustrezno kamero in krmilnikom je možno prepoznati steklenico.
POTRJENO – S kamero FQ2 in krmilnikom Siemens S7-1200 je možno izmeriti mere steklenice in jo prepoznati.
- Hipoteza 2 – Sistem bo deloval z vsaj 80% zanesljivostjo.
POTRJENO – Vrste steklenic, s katerimi smo sistem testirali, je ta prepoznal z več kot 80% zanesljivostjo.
- Hipoteza 3 – Možno bo hitro dodajanje novih in spreminjanje podatkov že obstoječih steklenic.
POTRJENO – S pomočjo HMI-ja je možno dodati novo steklenico ali jo urediti, ne da bi posegali v kodo krmilnika.

9 ZAKLJUČEK

Vsak pristop, ki smo ga imeli pri reševanju problema ločevanja med steklenicami s fotografij, je imel, tudi ob primernem ozadju in konstantni osvetlitvi, slabosti, ki bi postale še posebej izrazite pri določenih skupinah možnih pravilnih odgovorov. Menimo, da bi za popolnoma zanesljiv sistem bile potrebne dve kameri, ena za fotografiranje steklenice od strani in ena za fotografiranje steklenice od zgoraj. Na žalost tega nismo mogli preizkusiti.

Ugotovili smo, da je za zanesljivo prepoznavanje steklenic dovolj že zaznavanje robov na fotografijah s pravilno osvetljenim ozadjem in bi zato takšne projekte lahko izpeljali tudi s cenejšimi kamerami, če to podpirajo ali če bi bili takšne algoritme pripravljene napisati sami.

Precejšen izziv nam je predstavljalo povezovanje kamere in krmilnika ter prejemanje podatkov na krmilniku, saj smo takšno stvar delali prvič. Dodatne skrbi smo imeli tudi zato, ker smo na krmilnik, zaslon in napajalnik, čakali precej dolgo, ampak ker smo o možnih algoritmih za prepoznavo steklenic razmišljali že prej, nam je uspelo naš sistem usposobiti relativno hitro po tem, ko smo od kamere prejeli prve podatke. Kljub vsem izzivom nam je zastavljene cilje uspelo doseči.

10 VIRI

- [1] *Mehatronika*. 2. izdaja. Ljubljana: Pasadena, 2017.
- [2] *iqsdirectory*. (online). (citirano 5. 2. 2022). Dostopno na naslovu: <https://www.iqsdirectory.com/articles/machine-vision-system.html>.
- [3] *Machine vision*. (online). (citirano 5. 2. 2022). Dostopno na naslovu: https://en.wikipedia.org/wiki/Machine_vision.
- [4] OMRON. *Automation Omron*. (online). (citirano 6. 3. 2022). Dostopno na naslovu: <https://automation.omron.com/en/us/products/family/FQ2>.
- [5] SIEMENS. *Siemens Industry online support*. (online). (citirano 5. 2. 2022). Dostopno na naslovu: <https://mall.industry.siemens.com/mall/en/uk/Catalog/Product/6ES73135BF030AB0>.
- [6] SIEMENS. *Siemens Industry online support*. (online). (citirano 5. 2. 2022). Dostopno na naslovu: <https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6GK7343-1EX30-0XE0>.
- [7] SIEMENS. *Siemens Industry online support*. (online). (citirano 5. 2. 2022). Dostopno na naslovu: <https://support.industry.siemens.com/cs/pd/72709?ptdi=td&dl=en&lc=en-DE>.
- [8] SIEMENS. *Siemens Industry online support*. (online). (citirano 5. 2. 2022). Dostopno na naslovu: SINAMICS G120C PN 0,75KW FILA - 6SL3210-1KE12-3AF1 - Industry Support Siemens.
- [9] SIEMENS. *Siemens Industry online support*. (online). (citirano 5. 3. 2022). Dostopno na naslovu: <https://support.industry.siemens.com/cs/products/6av2123-2db03-0ax0/simatic-hmi-ktp400-basic?pid=379924&ntp=Manual&mlfb=6AV2123-2DB03-0AX0&mfn=ps&lc=en-ID>.
- [10] SIEMENS. *Siemens Industry online support*. (online). (citirano 5. 3. 2022). Dostopno na naslovu: <https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6GK7277-1AA10-0AA0>.

[11] SIEMENS. *Siemens Industry online support*. (online). (citirano 5. 3. 2022). Dostopno na naslovu: <https://mall.industry.siemens.com/mall/en/ww/Catalog/Product/6EP1332-1SH71>.

[12] SIEMENS. *Siemens Industry online support*. (online). (citirano 5. 2. 2022). Dostopno na naslovu: <https://support.industry.siemens.com/cs/products?mf=ps&lc=en-WW>.

[13] VALENTAN, B. *Industrijski Ethernet : diplomska naloga visokošolskega strokovnega študijskega programa*. (online). 2003. (citirano 19. 2. 2022). Dostopna na naslovu: <https://dk.um.si/IzpisGradiva.php?lang=eng&id=1812>.

[14] *Vision systems*. (online). (citirano 5. 2. 2022). Dostopno na naslovu: <https://www.vision-systems.com/cameras-accessories/article/16736053/explore-the-fundamentals-of-machine-vision-part-i>.