

Univerza v Ljubljani

Fakulteta za elektrotehniko

Klemen Mravlje

Računalnik Raspberry Pi v vlogi PLC-ja

Diplomsko delo visokošolskega strokovnega študija

Mentor: doc. dr. Gorazd Karer

Ljubljana, 2016

Zahvala

Iskreno se zahvaljujem mentorju doc. dr. Gorazdu Karerju za vse nasvete, podporo in pomoč pri izdelavi diplomskega dela.

Posebna zahvala pa gre moji družini in puncu Nuši, ki so mi tekom študija vedno stali ob strani. Hvala, ker ste verjeli vame!

Vsebina

1 Uvod	13
2 Programirljivi logični krmilnik	15
2.1 Standard IEC 61131-3	16
2.2 Programski jeziki IL, ST, FBD, SFC in LD	16
2.2.1 IL – seznam ukazov	16
2.2.2 ST – strukturirani tekst	17
2.2.3 FBD – funkcijski bločni diagram.....	17
2.2.4 SFC – sekvenčni funkcijski diagram	17
2.2.5 LD – lestvični diagram	17
2.3 Kontakti in tuljave v lestvičnem diagramu.....	18
2.4 Povezave v lestvičnem diagramu	20
3 Računalnik Raspberry Pi	21
3.1 Specifikacije	22
3.2 Prvi koraki z računalnikom Raspberry Pi.....	23
3.2.1 Operacijski sistem.....	24
3.2.2 Tipkovnica in miška.....	24
3.2.3 Zaslon.....	25
3.2.4 Napajalnik.....	25
3.2.5 Zagon	25
4 Razširitveni modul PiFace Digital 2	27
4.1 Strojna oprema	28
4.2 Montaža razširitvenega modula PiFace Digital 2.....	33

5 Programsko okolje CODESYS	35
5.1 Priprava programskega okolja	36
5.2 Priprava računalnika Raspberry Pi	36
5.2.1 Aktiviranje vmesnikov I ² C, SPI, 1-wire	37
5.2.2 Namestitev knjižnic na računalnik Raspberry Pi	38
5.2.3 Statični naslov IP	39
6 Aplikacija v programskem okolju CODESYS	41
6.1 Nastavljanje novega projekta	41
6.2 Pisanje lestvičnega diagrama.....	43
6.3 Nalaganje aplikacije	44
7 Upravljanje zapornice z računalnikom Raspberry Pi	47
7.1 Izdelava zapornice	47
7.2 Vezava zapornice.....	48
7.3 Izvedba aplikacije za delovanje zapornice	49
7.4 Simulacija aplikacije.....	51
7.5 Zagon aplikacije	53
8 Zaključek	55
Dodatek	57
Literatura	59

Seznam slik

Slika 1: Kontakti v lestvičnem diagramu.....	19
Slika 2: Tuljave v lestvičnem diagramu	19
Slika 3: Zaporedna vezava v lestvičnem diagramu	20
Slika 4: Vzporedna vezava v lestvičnem diagramu	20
Slika 5: Računalnik Raspberry Pi 2 Model B	21
Slika 6: Razširitveni modul PiFace Digital 2	27
Slika 7: Strojna oprema razširitvenega modula PiFace Digital 2 [11]	28
Slika 8: Vhodi na razširitvenem modulu PiFace Digital 2	29
Slika 9: Primer vezave tipke ali stikala.....	30
Slika 10: Izhodi na razširitvenem modulu PiFace Digital 2	30
Slika 11: Primer vezave LED diode	31
Slika 12: Delovanje releja.....	32
Slika 13: Primer vezave elektromotorja na rele.....	32
Slika 14: Montiran razširitveni modul PiFace Digital 2	33
Slika 15: Orodje raspi-config.....	37
Slika 16: Izbira naprave in jezika	42
Slika 17: Pravilno nastavljen projekt.....	42
Slika 18: Input Assistant	43
Slika 19: Aplikacija	44
Slika 20: Elementi zapornice	48
Slika 21: Sestavljena zapornica	49
Slika 22: Diagram poteka	50
Slika 24: Virtualna tipka "Push switch"	52
Slika 25: Lestvični diagram z virtualno tipko.....	53

Povzetek

V modernem svetu, kjer vse stremi k avtomatizaciji, se je skoraj vsak izmed nas že srečal s procesom, ki ga vodi programirljivi logični krmilnik (PLK). Z avtomatizacijo in programiranjem se vse bolj srečujemo v šolstvu in celo doma. PLK-ji so večinoma zelo dragi, zato je pred nakupom potrebno temeljito premisliti, ali se bo tak nakup izplačal. V šolstvu se tako redko srečamo s PLK-ji, saj se zaradi njihove visoke cene za nakup odloči le peščica šol. Posledično nima vsak možnosti, da bi se preizkusil v razvijanju aplikacij za PLK-je in tako marsikatera potencialna inovativna ideja ostane nerealizirana. Cenovno dostopni računalnik Raspberry Pi z razširitvenim modulom PiFace Digital 2 pa približuje svet avtomatizacije ne samo v šolstvu, ampak tudi v domači rabi.

V diplomskem delu sem preizkusil in opisal računalnik Raspberry Pi in razširitveni modul PiFace Digital 2, ki ju skupaj uporabimo v vlogi PLK-ja. Opisal sem tudi vse nastavitve in korake za aktivacijo, ki so potrebni za zagon in pravilno delovanje. Za preizkus računalnika Raspberry Pi v vlogi PLK-ja sem s programskim okoljem CODESYS razvil aplikacijo za dviganje in spuščanje modela zapornice in uspešno preizkusil njeno delovanje.

Ključne besede: računalnik Raspberry Pi, razširitveni modul PiFace Digital 2, programsko okolje CODESYS, PLK, avtomatizacija

Abstract

Due to worldwide modernization, many people have come across a process, which is controlled by a programmable logic controller (PLC). Automation and programming have already become a part of everyday life and they are also involved in our educational system. But since PLC is a rather expensive tool, there is a clear need to carefully evaluate its purchase decision and define whether it will bring the expected results. PLC's high price is unfortunately still an unbridgeable obstacle for many schools. Consequentially, not many students have the opportunity to test their skills in developing applications for PLC's. Thus, many potentially innovative ideas are still left unrealized. However, the arrival of the cheap Raspberry Pi computer and its PiFace Digital 2 expansion module has made the world of automatization more accessible not only for schools, but for home projects as well.

In my thesis I have described the Raspberry Pi computer and the PiFace Digital 2 expansion module, which together form a PLC, and tested both in practice. Furthermore, I have described all settings and activation steps that are required for their successful boot and operation. For the purpose of testing how the Raspberry Pi computer functions as a PLC, I have developed and also successfully tested an application in CODESYS environment, which controls a model barrier.

Keywords: Raspberry Pi computer, PiFace Digital 2 expansion module, CODESYS environment, PLC, automation

1 Uvod

Namen moje diplomske naloge je predstaviti z računalnik Raspberry Pi in osnovno rokovanje ter kasneje programiranje razširitvenega modula PiFace Digital 2 v računalniškem okolju CODESYS z uporabo lestvičnih diagramov. Razširitveni modul PiFace Digital 2 nam preko vhodov in izhodov omogoča programiranje in povezovanje različnih stikal, gumbov, senzorjev, enostavnih vezij, elektromotorjev in še marsičesa. Z uporabo programskega okolja CODESYS lahko razvijemo aplikacijo v enem izmed jezikov za programiranje krmilnikov iz standarda IEC 61131-3. Izbral sem si lestvični diagram, ki sledi neki določeni sekvenci vhodnih signalov in nam želeno prikaže na izhodu. S tem lahko računalnik Raspberry Pi z razširitvenim modulom PiFace Digital 2 uporabljamo kot programirljivi logični krmilnik (v nadaljevanju PLK).

V poplavi PLK-jev je težko najti ustrezen krmilnik, ki bo imel relativno nizko ceno in vseeno omogočal vse ali pa skoraj večino operacij kot dražji industrijski krmilniki. Na Fakulteti za elektrotehniko za potrebe laboratorijskih vaj uporabljamo Siemensove PLK-je s katerimi pri predmetu Računalniško vodenje procesov vodimo pomanjšano industrijsko linijo. Z lestvičnimi diagrami se naučimo krmiliti in nadzorovati potek oziroma delovanje stikal in elektromotorjev.

V nadaljevanju bom predstavil računalnik Raspberry Pi in razširitveni modul PiFace Digital 2, ki skupaj lahko tvorita PLK. Delovanje bom preizkusil z aplikacijo, ki bo upravljala z vhodnimi in izhodnimi elementi ter elektromotorjem in ocenil, če bi s tako izvedbo lahko nadomestili mnogo dražje krmilnike.

V drugem poglavju sem na kratko opisal nastanek in razvoj PLK-jev in standard IEC 61131-3, pri čemer sem se posvetil programskim jezikom, ki jih

uporabljamo za programiranje PLK-jev. Večjo pozornost sem posvetil lestvičnemu diagramu, saj je to programski jezik, ki sem ga uporabil za diplomsko delo.

Tretje poglavje opisuje računalnik Raspberry Pi in rokovanje z njim. Na kratko sem opisal specifikacije računalnika Raspberry Pi, strojno opremo in po vrsti naštel korake, ki so potrebni za uspešen zagon.

V četrtem poglavju sem na kratko predstavil razširitveni modul PiFace Digital 2 in podrobneje opisal njegovo strojno opremo.

Peto poglavje opisuje programsko okolje CODESYS in opisal korake, ki so potrebni za namestitev in inicializacijo komunikacije z računalnikom Raspberry Pi. V tem poglavju so točno predstavljeni vsi koraki, ki so potrebni za uspešno razvijanje aplikacij za računalnik Raspberry Pi.

V šestem poglavju je predstavljena prva aplikacija, ki sem jo razvil za računalnik Raspberry Pi. Prav tako so podrobno opisani vsi koraki, ki jih moramo izvesti pri vsakem zagonu novega projekta v programskem okolju CODESYS, če želimo razviti aplikacijo za računalnik Raspberry Pi in razširitveni modul PiFace Digital 2.

Sedmo poglavje opisuje zamisel in realizacijo aplikacije za dvig zapornice, ki bi jo poganjal računalnik Raspberry Pi. Opisal sem celoten postopek od ideje, do razvite aplikacije, ki dviga in spušča zapornico s pomočjo elektromotorja. V zadnjem poglavju so podane še sklepne ugotovitve.

2 Programirljivi logični krmilnik

Programirljivi logični krmilnik (PLK) je posebne vrste krmilnik z mikroprocesorjem, ki uporablja programirljiv pomnilnik za shranjevanje navodil, logičnih funkcij, časovnikov, števec in ostalih funkcij, ki so potrebne za vodenje strojev in procesov. Izdelan je tako, da lahko z njim delajo tudi inženirji, ki niso vešči programiranja, saj uporabljajo enostavnejše programske jezike [1].

Industrijske krmilnike so izumili zaradi potreb v avtomobilski industriji v Združenih državah Amerike. Vsakih nekaj let je bilo potrebno spremeniti krmiljenje strojev zaradi prihoda novih modelov avtomobilov. Pred prihodom krmilnikov je nalogo krmiljenja strojev opravljala relejska tehnika, kar je pomenilo kopico relejskih in kontaktorskih omar, časovnikov in kamer za nadzor procesov. Vse skupaj je bilo potrebno še ožičiti, kar je pomenilo kilometre povezovalnih kablov. Če je bila zahtevana sprememba ali dopolnitev logične funkcije, je bilo potrebno ponovno ožičiti in povezati celotno fizično enoto. Poleg tega vezje ni omogočalo testiranja ali simulacije, torej je bilo testiranje potrebno opravljati na že postavljenem vezju. Če se je takrat izkazalo, da je prišlo do napake na shemi ali pa pri povezovanju kablov, je to lahko pomenilo ogromno težavo [2].

Leta 1968 je General Motors podal zahtevo za zamenjavo klasičnega relejskega krmilja z elektronskim. Prvi krmilnik, ki je šel v prodajo se je imenoval MODICON 084, razvilo pa ga je podjetje Bedford Associates. Uspeh krmilnika je temeljil na programskem jeziku, saj je uporabljal lestvični diagram, ki temelji na relejski shemi, ki je bila znana električarjem. Prvi krmilniki so bili zelo enostavni, saj je njihova naloga bila le nadomestiti takratno relejsko vezje. S časom so se logični krmilniki nadgrajevali na tak način, da so začeli ponujati tudi števec in časovnike, kasneje pa so dobili še analogne vhode in izhode ter aritmetične funkcije. Dandanes so PLK-je

razvili do te mere, da ponujajo vrsto razvijalnih orodij, logičnih funkcij, nadzor servomotorjev in komunikacijo z drugimi PLK-ji [2].

2.1 Standard IEC 61131-3

Zaradi vse večje ponudbe različnih PLK-jev je bilo skozi leta predlaganih mnogo programskih standardov s strani nacionalnih in mednarodnih komitejev, ki so želeli razviti skupni vmesnik za programirljive krmilnike. Leta 1993 je bil prvič objavljen standard za programske jezike imenovan IEC 61131-3. Ta del standarda, ki se ga drži večina večjih proizvajalcev PLK-jev, določa skladnjo in semantiko programskih jezikov za programirljive krmilnike [2], [3]. Niz sestavljajo:

- seznam ukazov – IL (ang. instruction list),
- strukturirani tekst – ST (ang. structured text),
- lestvični diagram – LD (ang. ladder diagram),
- funkcijski bločni diagram – FBD (ang. function block diagram),
- sekvenčni funkcijski diagram – SFC (ang. sequential function chart).

LD in FBD sta grafična programska jezika, IL in ST pa sta tekstovna programska jezika. SFC je tekstovni in grafični jezik za strukturiranje in notranjo organizacijo programov in funkcijskih blokov pri programiranju PLK-jev, ki je poleg seznama ukazov, strukturiranega teksta, lestvičnega diagrama in funkcijskega bločnega diagrama eden od osnovnih standardiziranih programskih jezikov.

2.2 Programski jeziki IL, ST, FBD, SFC in LD

2.2.1 IL – seznam ukazov

IL je enostavni programski jezik, kjer je podajanje kode podobno kodiranju v zbirniku. Njegove slabosti se odražajo predvsem v enostavnosti jezika in posledično nezmožnosti uporabe bolj kompleksnih ukazov. Prednosti uporabe jezika IL pa so v nizkih zahtevah po računalniški moči. Razlog, da se ta jezik še vedno uporablja v

programiranju krmilnikov je, da obstaja že dlje kot ostali jeziki, zajeti v standardu. Mnogo starejših krmilnikov lahko uporablja samo IL in LD jezik [2], [3].

2.2.2 ST – strukturirani tekst

ST je programski jezik ki temelji na besedilu. V nasprotju z IL je ST programski jezik na visoki ravni. To pomeni, da se lahko v eni besedilni vrstici izvedejo številna navodila in operacije. Če ga primerjamo z drugimi jeziki na visoki ravni, je najbolj podoben programiranju v programskem jeziku Pascal ali C.

ST je bil posebej razvit za potrebe programiranja kompleksnih aritmetičnih funkcij in delu z besedilnimi objekti in tekstom [2], [3].

2.2.3 FBD – funkcijski bločni diagram

FBD je grafični programski jezik, ki temelji na povezovanju funkcij in funkcijskih blokov. Jezik vključuje standardne logične funkcije kot so AND, OR, NOT, itn., ter funkcije bloke kot so števcji in časovniki. Poleg tega lahko določimo tudi poljubne bloke. Uporaba FBD-ja kot programskega jezika za programiranje krmilnikov nima posebnih prednosti. Vse kar lahko napišemo v FBD-ju lahko napišemo tudi v ST-ju. Edina razlika je, da so dogodki bolj pregledni pri enostavnejših programih [2], [3].

2.2.4 SFC – sekvenčni funkcijski diagram

SFC je grafično orodje primerno za uporabo pri programiranju sekvenčnega vodenja. SFC pravzaprav ni tradicionalen programski jezik, ampak grafični pristop k strukturiranju programske kode. Vsi ostali jeziki v standardu so lahko uporabljeni skupaj s SFC-jem [2], [3].

2.2.5 LD – lestvični diagram

Lestvični diagram (ang. Ladder Diagram – LD) je eden najbolj razširjenih načinov programiranja PLK-jev [2] kljub dejstvu, da je izvedba kompleksnih

algoritmov in iskanje napak lažje z uporabo programskega jezika ST [4]. Poleg tega se ga skoraj vsi novi inženirji in tehniki še vedno morajo naučiti in razumeti. Kot eden od grafičnih programskih jezikov je vključen tudi v mednarodni standard IEC 61131-3 [2], [4].

Lestvični diagram je ime dobil po obliki celotnega vezja, ki spominja na lestev. Zaradi boljše preglednosti električnega načrta se je uveljavil način, pri katerem rišemo napajalne vodnike navpično in usmerjena vezja kontaktov in tuljav vodoravno. Lestvični diagram temelji na relejskih vezalnih shemah, kjer so narisani kontakti, katerih vzporedne in zaporedne vezave predstavljajo testiranje logičnih pogojev, in tuljave, ki se aktivirajo kadar vezje prevaja [4].

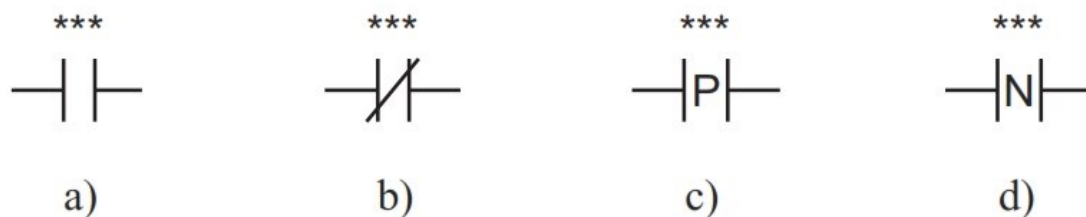
Ker je razvoj PLK-jev prinesel na trg vedno več krmilnikov različnih proizvajalcev, je nastala potreba po čim bolj enostavnem načinu programiranja, zato so obliko lestvičnega diagrama prenesli v programski jezik. Simboli programskih elementov so zamenjali simbole gradnikov električnih vezij, le simbol za kontakt in tuljavo sta ostala enaka. Simbole zlagamo v »prečke«, ki so levo omejene z virom napajanja in desno omejene s ponorom. Poleg napajalnih vodnikov, tuljav in kontaktov sestavljajo lestvični diagram še povezave ter ostali gradniki, kot so funkcijski bloki in časovniki. Vsi simboli lestvičnih diagramov imajo smisel le, če so prirejeni določeni sponki krmilnika ali notranjem stanju [4].

Program se izvaja v ciklih, saj se s tem posnema delovanje električnih vezij, kje se vse veje vezja izvajajo hkrati. Program ob začetku vsakega cikla prebere stanja vhodov krmilnika in jih zapiše v pomnilnik. Te nato obdeluje in rezultate zapiše v del pomnilnika, ki je namenjen izhodom. Tako obdela vse veje diagrama in ob koncu cikla prepiše rezultate iz pomnilnika na izhodne sponke. Na ta način se izhodi, ki so posledica sočasnih sprememb vhodov, spremenijo sočasno, ne glede na vrstni red prečk v lestvičnem diagramu. Ker krmilnik izvede cikel v zelo kratkem času, je nam videti, kot da se vse izvaja hkrati [4].

2.2 Kontakti in tuljave v lestvičnem diagramu

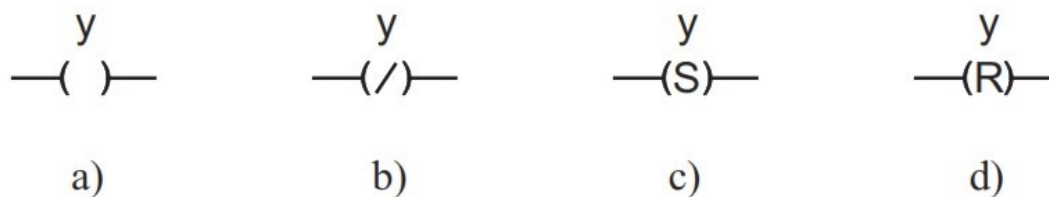
Delovni kontakt je prikazan na sliki 1 a. Kontakt je sklenjen, če je logična spremenljivka v visokem stanju. Mirovni kontakt lahko vidimo na sliki 1 b. Kontakt

je sklenjen, če je logična spremenljivka v nizkem stanju. Detektor pozitivnega prehoda je prikazan na sliki 1 c. Ko detektor zazna prehod stanja pripadajoče logične spremenljivke z 0 na 1, kontakt se sklence za čas enega programskega cikla. Detektor negativnega prehoda, prikazan na sliki 1 d, pa se sklence takrat, ko zazna prehod stanja pripadajoče logične spremenljivke z 1 na 0.



Slika 1: Kontakti v lestvičnem diagramu

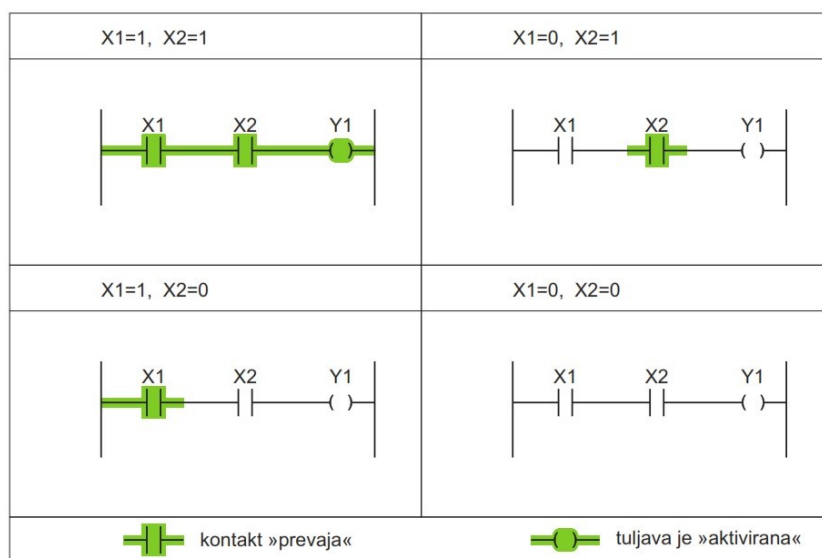
Na sliki 2 a je simbol za trenutno delujočo tuljavo. Ta preslika stanje povezave na levi strani v stanje pripadajoče logične spremenljivke. Inverzno delujoča tuljava, katere simbol lahko vidimo na sliki 2 b, preslika inverzno stanje povezave na levi strani v stanje logične spremenljivke. Tuljava tipa SET (glej sliko 2 c) postavi vrednost logične spremenljivke na 1, ko zazna vrednost 1 na levi strani povezave. Takšno stanje se ohrani dokler ga tuljava tipa RESET (glej sliko 2 d) ne postavi ponovno na 0. Tuljava RESET postavi pripadajočo logično spremenljivko na vrednost 0, če je povezava na levi strani v stanju 1.



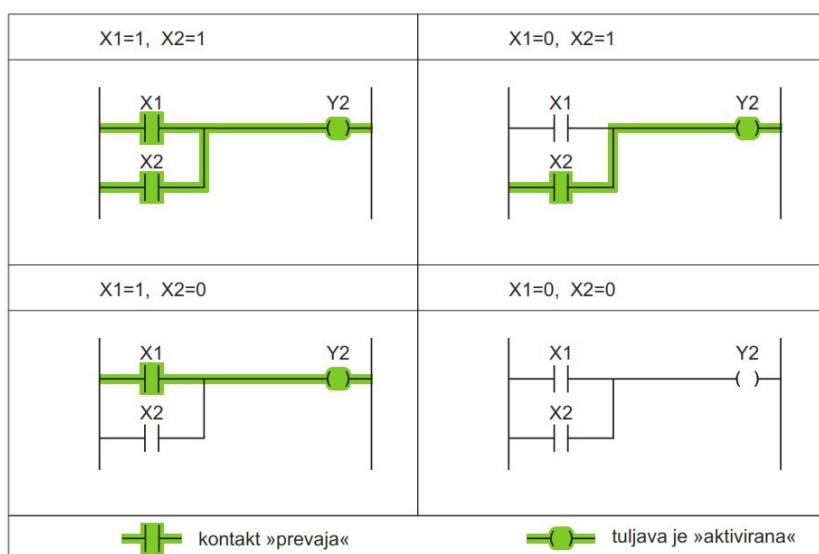
Slika 2: Tuljave v lestvičnem diagramu

2.4 Povezave v lestvičnem diagramu

S povezavami sestavljamo elemente lestvičnega diagrama v logične pogoje. Na sliki 3 je prikazan primer zaporedne vezave, ki predstavlja logični IN. To pomeni, da morata biti v prevajanju oba kontakta, da se stanje povezave prevaja naprej. Na sliki 4 pa je primer vzporedne vezave, ki predstavlja logični ALI. Stanje se prevaja naprej v primeru, da prevajata ali eden ali drugi kontakt.



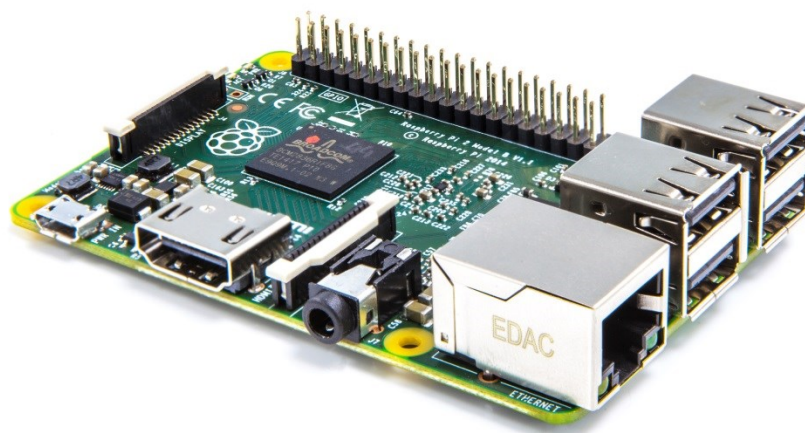
Slika 3: Zaporedna vezava v lestvičnem diagramu



Slika 4: Vzporedna vezava v lestvičnem diagramu

3 Računalnik Raspberry Pi

Raspberry Pi je nizkocenovni mikroračunalnik v velikosti kreditne kartice, ki se postavlja po robu vsem mnogo dražjim osebnim računalnikom. Za razliko od ostalih naprav (pametni telefoni, tablice, igralne konzole), ki so zaprte v ohišje in nam onemogočajo vpogled v »drobovje« naprave, je, kot lahko vidimo na sliki 5, računalnik Raspberry Pi primarno brez ohišja. Že s tem nas pripravi do razmišljanja o zmožnostih take naprave [5]. Ljudje po svetu so tako računalnik Raspberry Pi uporabili za nešteto različnih namenov. Dave Akerman ga je s prijatelji pritrdil na vremenski balon in ga poslal na nadmorsko višino 40 km. Z njim je s spletno kamero posnel fotografije skoraj iz vesolja [6]. Profesor Simon Cox z univerze v Southamptonu je z Lego kockami povezal 64 računalnikov Raspberry Pi v eksperimentalni superračunalnik, ki je konkuriral superračunalnikom vrednosti več milijonov dolarjev [7]. Uporabljajo pa ga tudi v projektu imenovanem »FishPi«, kjer nameravajo računalnik Raspberry Pi uporabiti v brezpilotnem plovilu, ki bo prečkal Atlantik in opravil ogromno meritev [8].



Slika 5: Računalnik Raspberry Pi 2 Model B

Računalnik Raspberry Pi se enostavno priključi na računalniški zaslon ali TV sprejemnik preko kabla HDMI ali RCA. Za delo uporabljamo tipkovnico in miško, omogoča pa tudi povezovanje v svetovni splet preko mrežnega kabla. Operacijski sistem temelji na Linuxu, v prihodnosti pa načrtujejo podporo tudi za Microsoft Windows 10.

Z računalnikom Raspberry Pi lahko opravljamo skoraj vsa opravila, ki jih zmore običajni prenosnik. Na sistemu že dobimo nameščen program LibreOffice, s katerim lahko ustvarjamo in urejamo besedila, podpira tudi .doc dokumente, ki so bolj poznani iz okolja Microsoft Windows. Nameščen je tudi brskalnik, ki omogoča brskanje po spletu. Predvajamo pa lahko tudi filme in glasbo s programi, ki jih (zastonj) dobimo v trgovini z aplikacijami. Vse dodatne aplikacije in programi so dostopni v trgovini, namestimo pa jih enostavno preko terminala z ukaznimi vrsticami.

Računalnik Raspberry Pi omogoča tudi priključitev različnih dodatkov kot so kamera, zaslon na dotik, modul WiFi in še veliko več. S temi priključki pa lahko računalnik Raspberry Pi spremenimo v še bolj vsestransko uporabno napravo. Z razširitvenim modulom PiFace Digital 2 pa ga lahko spremenimo v programirljiv logični krmilnik. To je tudi namen moje diplomske naloge.

3.1 Specifikacije

Trenutno najnovejši in tudi najzmogljivejši računalnik Raspberry Pi je Raspberry Pi 2 Model B. Je druge generacije in je popolnoma združljiv s prejšnjo. Specifikacije so:

- 900 Mhz štirijedrni procesor ARM Cortex-A7,
- 1 GB delovnega pomnilnika,
- 4 priključki USB,
- 40 splošnonamenskih vhodno-izhodnih pinov,
- izhod HDMI,
- 10/100 ethernet,
- kombinirani 3,5-mm avdio in video izhod,

- vmesnik za kamero (CSI),
- vmesnik za zaslon (DSI),
- reža za pomnilniško kartico MicroSD,
- grafični procesor VideoCore IV 3D.

3.2 Prvi koraki z računalnikom Raspberry Pi

Poleg računalnika Raspberry Pi bomo potrebovali tudi pomnilniško kartico velikosti vsaj 8 GB (razreda 4), kamor bomo namestili operacijski sistem, tipkovnico in miško, zaslon oziroma TV sprejemnik in ustrezen napajalnik. Poleg naštetega je priporočljivo imeti tudi mrežni kabel za priklop na splet, preko katerega lahko dobimo vse najnovejše posodobitve in nadgradnje.

Za računalnik Raspberry Pi je trenutno na voljo več različnih operacijskih sistemov. Najbolj pogosti so:

- Noobs: je najbolj primeren za začetnike, na voljo je nakup pomnilniške kartice z že naloženim sistemom,
- Raspbian: je uradno podprt operacijski sistem, na voljo je zastonj prenos z uradne Raspberry Pi spletne strani. Na sistemu dobimo že nameščene vse bistvene programe kot so Python, Scratch, Sonic Pi, Java, Mathematica, LibreOffice itd,
- Ubuntu Mate: je Linux Ubuntu sistem, prilagojen za računalnik Raspberry Pi,
- Windows 10 IOT Core: verzija operacijskega sistema Windows trenutno primerna le za razvijalce,
- OSMC: operacijski sistem, ki računalnik Raspberry Pi pretvori v medijski predvajalnik.

Najbolj primeren operacijski sistem za večino uporabnikov, ki bodo računalnik Raspberry Pi uporabljali tako vsakodnevno kot za programiranje, je Raspbian. To je tudi operacijski sistem, ki sem ga uporabljal za računalnik Raspberry Pi v vlogi PLK-ja.

3.2.1 Operacijski sistem

Na pomnilniško kartico je pred prvim zagonom potrebno namestiti uradni operacijski sistem Raspbian, ki temelji na Linuxu. Trenutno najnovejšo verzijo je dosegljiva na uradni spletni strani [9]. Poleg pomnilniške kartice v velikosti vsaj 8 GB, bomo za namestitev operacijskega sistema potrebovali še računalnik, čitalnik pomnilniških kartic in program za zapis slike pogona. Sam sem uporabil Win32DiskImager, ki je na voljo brezplačno [10]. Za namestitev operacijskega sistema na pomnilniško kartico sledimo naslednjim korakom:

1. pomnilniško kartico vstavimo v čitalnik in preverimo, katero črko pogona nam je dodelil računalnik. Črko pogona lahko preverimo v Raziskovalcu ali Mojem računalniku.
2. Razširimo preneseno datoteko operacijskega sistema Raspbian (z uporabo programa kot je npr. 7z).
3. Ko smo prenesli in namestili Win32DiskImager, ga zaženemo kot administrator in izberemo mesto, kjer se nahaja razširjena datoteka operacijskega sistema.
4. Izberemo črko pogona pomnilniške kartice. Pomembno je, da izberemo pravo črko pogona, saj lahko v nasprotnem primeru izgubimo podatke na izbranem pogonu.
5. Izberemo možnost »Write« in počakamo, da program izvrši vse do konca.
6. Zapremo program in odstranimo pomnilniško kartico.

Tako pripravljeno pomnilniško kartico nato vstavimo v računalnik Raspberry Pi.

3.2.2 Tipkovnica in miška

Za uporabo računalnika Raspberry Pi moramo priključiti tipkovnico in miško. Uporabimo lahko tako navadno USB tipkovnico in miško, kot tudi brezžično. Za lažjo uporabo predlagam brezžično tipkovnico in brezžično miško, ki ju lahko povežemo na skupni manjši sprejemnik.

3.2.3 Zaslon

Priključimo zaslon ali katerikoli TV sprejemnik s kablom HDMI. Če zaslon nima vhoda HDMI, ga lahko povežemo tudi preko HDMI-DVI adapterja ali pa celo HDMI-VGA. Računalnik Raspberry Pi lahko povežemo tudi preko 3,5-mm kompozitnega kabla RCA.

3.2.4 Napajalnik

Nazadnje priključimo še ustrezen napajalnik. Ta mora imeti priključek micro USB, zagotavljati pa mora konstantno napetost 5 V in minimalni tok 700 mA. Priporočena je uporaba močnejšega napajalnika s tokom 1,2 A, saj nam omogoča priključitev tudi modula WiFi brez dodatnega napajanja. Za takojšnjo uporabo je primeren tudi polnilnik za pametne telefone. Takoj ko napajalnik priključimo, se računalnik Raspberry Pi samodejno zažene.

3.2.5 Zagon

Ob prvem zagonu se bo operacijski sistem namestil, nato pa se nam odpre ukazna vrstica za vpis uporabniškega imena in gesla. Privzeto uporabniško ime je »pi« in geslo »raspberry«. Sistem se najprej zažene v ukazno vrstico in če želimo uporabljati vse programe, ki jih računalnik Raspberry Pi omogoča, vpišemo `startx` in potrdimo s pritiskom na tipko »Enter«. S tem ukazom sistem zažene grafični uporabniški vmesnik, ki je podoben bolj poznanemu Linuxu.

4 Razširitveni modul PiFace Digital 2

Na sliki 6 je razširitveni modul PiFace Digital 2 za računalnik Raspberry Pi, ki nam omogoča povezovanje različnih naprav s katerimi lahko nadziramo okolico. Preko relejev, stikal in analognih vhodov in izhodov lahko na razširitveni modul PiFace Digital 2 priključimo različne senzorje, elektromotorje, lučke ipd., ki jih nato nadziramo z računalnikom Raspberry Pi. Razširitveni modul PiFace Digital 2 lahko uporabljamo za izobraževanje, industrijsko avtomatizacijo in nadzorne sisteme [11].



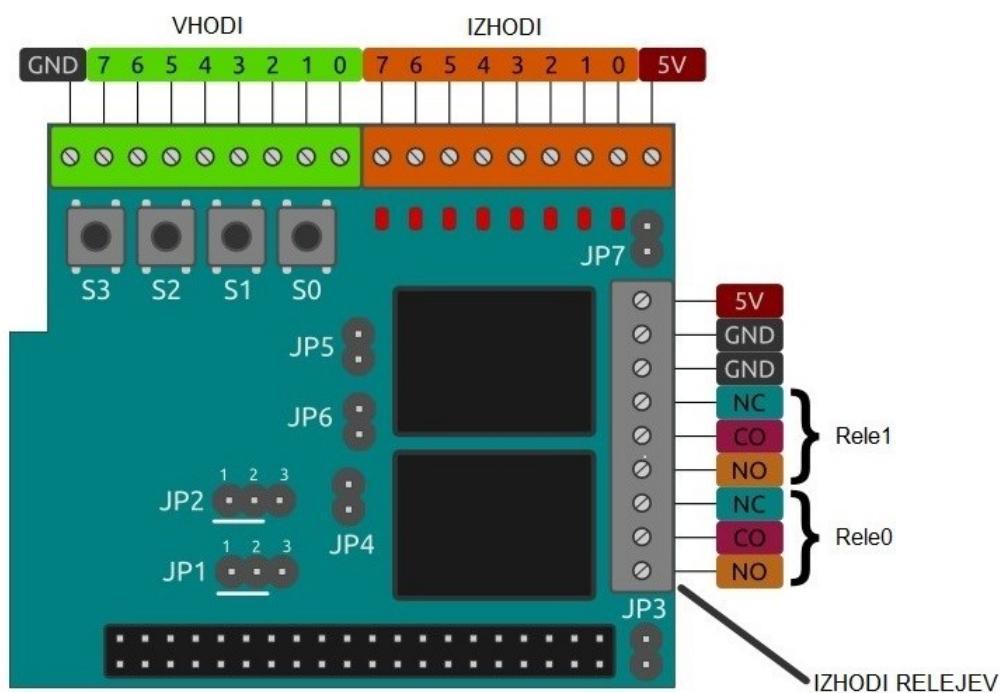
Slika 6: Razširitveni modul PiFace Digital 2

Specifikacije razširitvenega modula:

- 2 preklopna releja (priklop maksimalno 20 V, 5 A),
- 4 stikala,
- 8 digitalnih vhodov,
- 8 izhodov z odprtimi kolektorji,
- 8 indikatorjev s svetlečimi diodami,
- delovna temperatura od $-40\text{ }^{\circ}\text{C}$ do $85\text{ }^{\circ}\text{C}$,
- enostavno programiranje s programskimi jeziki Python, Scratch in C,
- grafični emulator.

4.1 Strojna oprema

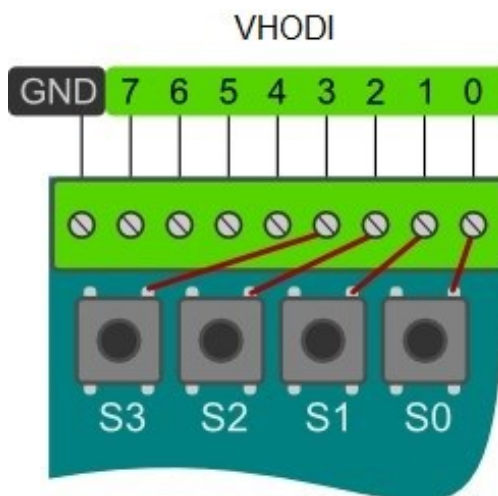
Na sliki 7 je prikazana shema strojne opreme razširitvenega modula PiFace Digital 2.



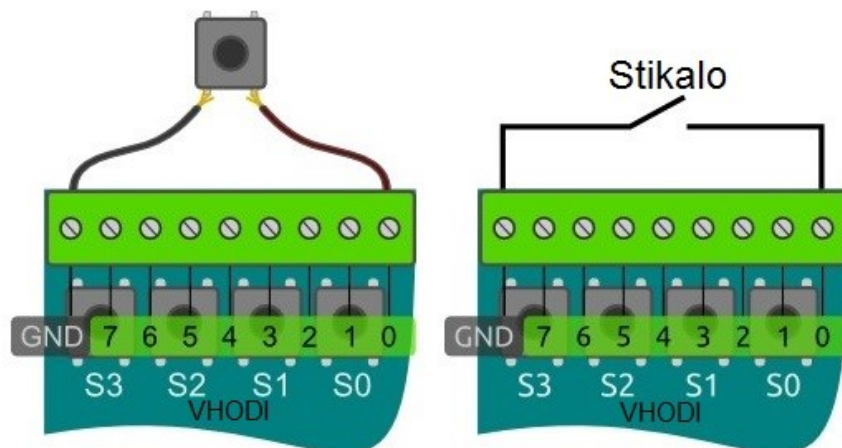
Slika 7: Strojna oprema razširitvenega modula PiFace Digital 2 [11]

- JP1 – naslovni bit 0,
- JP2 – naslovni bit 1,
- JP3 – povezuje razširitveni modul PiFace Digital 2 5 V na računalnik Raspberry Pi 5 V,
- JP4 – povezuje blažilne diode na 5 V,
- JP5 – če je povezan, je rele1 omogočen,
- JP6 – če je povezan, je rele2 omogočen,
- JP7 – povezuje napetost 5 V na diode in releje.

Razširitveni modul PiFace Digital 2 ima 8 vhodov na katere lahko preko vijajčnih sponk povežemo stikala ali kontakte. Vhod zaznava, kadar je povezan na GND. Na sliki 8 vidimo, da so stikala S1, S2, S3 in S4 vzporedno povezana na prve štiri vhode (0, 1, 2, 3). Na sliki 9 pa sta prikazana primera vezave tipke ali stikala na enega izmed vhodov.

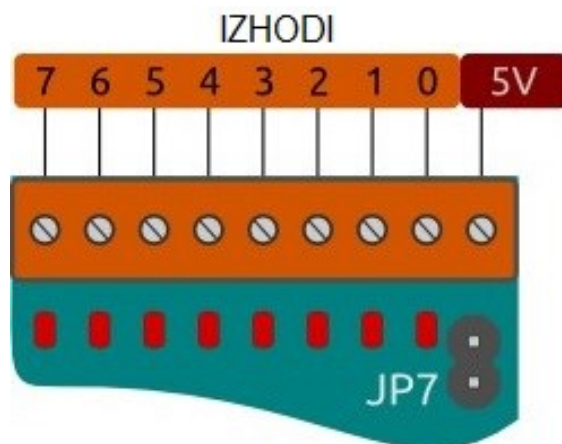


Slika 8: Vhodi na razširitvenem modulu PiFace Digital 2

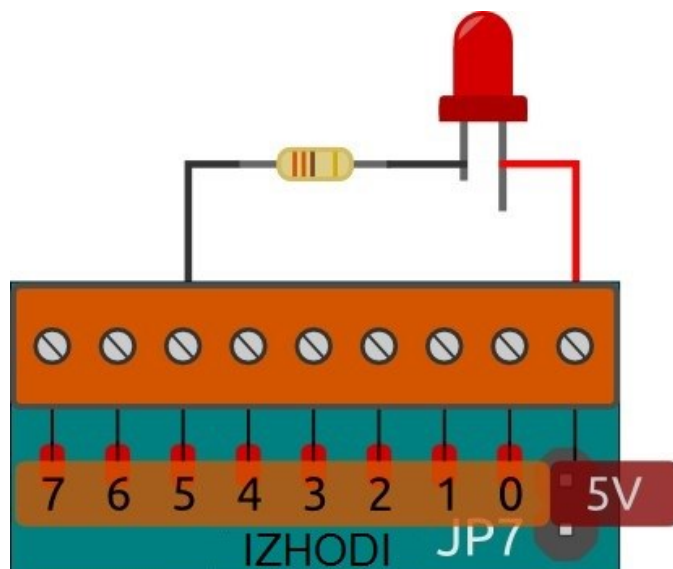


Slika 9: Primer vezave tipke ali stikala

Na sliki 10 je shema osmih izhodnih vijačnih sponk na vrhu razširitvenega modula PiFace Digital 2. To so izhodi z odprtim kolektorjem, ki jih uporabljamo za priključevanje svetlečih diod, lučk ali elektromotorjev. Primer vezave svetleče diode je na sliki 11.



Slika 10: Izhodi na razširitvenem modulu PiFace Digital 2



Slika 11: Primer vezave LED diode

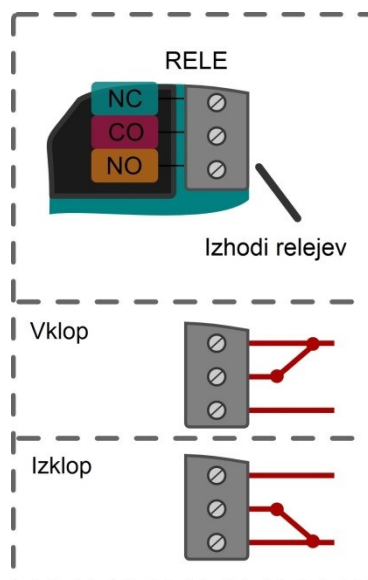
Indikatorje s svetlečimi diodami, ki so vezani na izhodne vijačne sponke, lahko uporabljamo kot prikaz, da neka aplikacija deluje na računalniku Raspberry Pi brez dodatnih zunanjih komponent. Prav tako so zelo uporabni za prikaz napak ob nepravilnem delovanju zelenega izhodnega stanja.

Na razširitvenem modulu PiFace Digital 2 je 8 izhodov z odprtim kolektorjem, preko katerih lahko nadziramo lučke, motorje ali releje. Ker so izhodi z odprtim kolektorjem, ne dajejo na izhod nobene napetosti. Namesto tega omogočajo ali onemogočajo ozemljitev toka. S tako izvedbo lahko na izhode priključimo porabnike z različno napetostjo. Če na izhod priključimo porabnike ki delujejo z napetostjo večjo kot 5 V, je potrebno pravilno nastaviti mostičke JP4, JP5, JP6 in JP7.

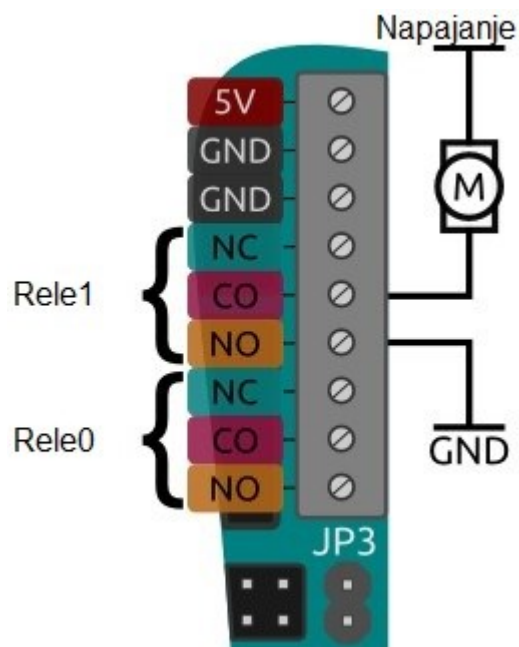
Razširitveni modul PiFace Digital 2 ima tudi dva preklopna releja, ki sta vzporedno vezana na prva dva izhoda. Vsak izmed relejev je priključen na tri črne vijačne sponke. Delovanje sponk je opisano tako:

- NC – mirovni kontakt (ang. **N**ormally **C**losed contact),
- CO – skupni kontakt (ang. **C**OMmon contact),
- NO – delovni kontakt (ang. **N**ormally **O**pen contact).

Mirovni in skupni kontakt bosta delovala dokler se ne aktivira rele, nato pa se skupni kontakt poveže na delovni kontakt. Delovanje releja je prikazano na sliki 12. Na sliki 13 pa lahko vidimo primer vezave elektromotorja na rele.



Slika 12: Delovanje releja



Slika 13: Primer vezave elektromotorja na rele

4.2 Montaža razširitvenega modula PiFace Digital 2

Zelo pomembno je, da pred montažo razširitvenega modula PiFace Digital 2 na računalnik Raspberry Pi le-tega odklopimo iz električnega omrežja.

Razširitveni modul PiFace Digital 2 enostavno montiramo na splošnonamenske vhodno-izhodne pine na računalniku Raspberry Pi, kot je razvidno na sliki 14. Prepričamo se, da so vsi pini poravnani s podnožjem na razširitvenem modulu PiFace Digital 2, sicer lahko pride do okvare. Ko smo ju povezali med seboj, priključimo računalnik Raspberry Pi na električno omrežje.



Slika 14: Montiran razširitveni modul PiFace Digital 2

5 Programsko okolje CODESYS

Za uporabo računalnika Raspberry Pi z razširitvenim modulom PiFace Digital 2 kot PLC bomo potrebovali programsko okolje CODESYS, ki ga uporabljajo v različnih vejah moderne industrije in v avtomatiki.

Programsko okolje CODESYS je orodje za programiranje PLC-jev, ki je povsem neodvisno od naprave, ki jo uporabljamo. Popolnoma sledi standardu IEC 61131-3 in podpira programiranje v vseh petih definiranih jezikih. Sposobno je izvajati vse standardne operatorje, funkcije in funkcijske bloke [2].

Programskega okolja CODESYS niso razvili proizvajalci PLC-jev [2], zato lahko z njim programiramo večino vrst PLC-jev, tudi npr. vsem bolj znane Siemensove PLC-je, ki jih uporabljamo tudi pri vajah pri predmetu Računalniško vodenje procesov. Z računalnikom Raspberry Pi in z razširitvenim modulom PiFace Digital 2 bom poizkusil vsaj delno nadomestiti uporabo teh mnogo dražjih PLC-jev.

V tem poglavju bom opisal, kako pripraviti računalnik Raspberry Pi, da bo programsko okolje CODESYS z njim uspešno komuniciralo. Programsko okolje CODESYS je treba najprej namestiti na računalnik, na katerem bomo razvijali aplikacijo. Nato je treba opraviti še nekaj nastavitev na računalniku Raspberry Pi in aktivirati vmesnike I²C, SPI in 1-wire, ki so potrebni za komunikacijo računalnika Raspberry Pi z razširitvenim modulom PiFace Digital 2. Na koncu bom opisal še postopek namestitve knjižnic programskega okolja CODESYS na računalnik Raspberry Pi.

5.1 Priprava programskega okolja

Programsko okolje CODESYS je brezplačno na voljo za prenos na njihovi uradni spletni strani [12]. Ko smo program namestili na računalnik, na katerem bomo pisali program za krmiljenje, moramo dodati še paket knjižnic in nastavitev CODESYS Control for Raspberry Pi. Knjižnice dobimo na isti spletni strani kot programsko okolje CODESYS. Proizvajalci nam ponujajo brezplačen dvourni preizkus knjižnic, za daljšo neprekinjeno uporabo pa je potrebno kupiti licenco za uporabo. To pomeni, da bo aplikacija, ki jo razvijemo v programskem okolju CODESYS, na računalniku Raspberry Pi neprekinjeno delovala dve uri. Nato se aplikacija zaustavi in potreben je ponovni zagon. Za razvoj aplikacije, s katero sem preizkusil uporabo računalnika Raspberry Pi kot PLK, je bil dvourni preizkus dovolj, zato nisem potreboval licence.

Knjižnice namestimo tako, da zaženemo programsko okolje CODESYS in v meniju »Tools« izberemo »Package Manager«. Tu izberemo možnost »Install«, se pomaknemo v mapo kamor smo shranili prenesen paket knjižnic in izberemo »CODESYS Control for Raspberry Pi«. Za namestitev knjižnic na računalnik Raspberry Pi, moramo na njemu opraviti še nekaj namestitev.

5.2 Priprava računalnika Raspberry Pi

Najprej je potrebno zagotoviti internetno povezavo na računalniku Raspberry Pi. Priključimo ga v isto omrežje kamor je priključen računalnik, na katerem bomo delali s programskim okoljem CODESYS. Če imamo modul WiFi, se v omrežje lahko povežemo brezžično, če ne, ga povežemo s kablom. Nato v terminalu vpišemo ukaz za posodobitev operacijskega sistema.

```
sudo apt-get update  
sudo apt-get upgrade
```

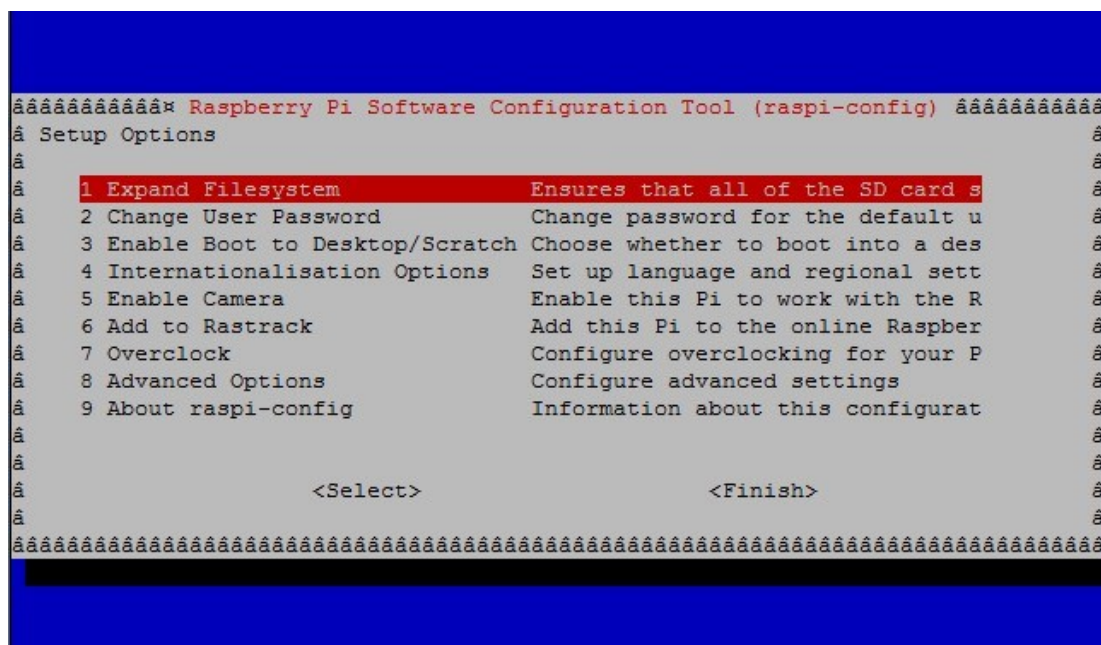
Ukaze potrdimo tako, da v terminal vnesemo `y`.

Prepričamo se tudi, da uporabljamo zadnjo verzijo orodja `raspi-config` z naslednjim ukazom.

```
sudo apt-get install raspi-config
```

Na sliki 15 je orodje za urejanje nastavitev na računalniku Raspberry Pi imenovano raspi-config. Zaženemo ga z naslednjim ukazom.

```
sudo raspi-config
```



Slika 15: Orodje raspi-config

S tipkovnico se pomaknemo v meni »Advanced Options« in izberemo »SPI«. To opcijo nastavimo na »Yes« in izberemo »Ok«.

Razširiti je potrebno tudi datotečni sistem. To storimo tako, da izberemo opcijo »Expand Filesystem« v orodju raspi-config.

Ko smo izvršili vse ukaze zapremo orodje raspi-config in ponovno zaženemo računalnik Raspberry Pi. To storimo tako, da v terminal vpišemo spodnji ukaz.

```
sudo reboot
```

5.2.1 Aktiviranje vmesnikov I²C, SPI, 1-wire

Računalnik Raspberry Pi za komunikacijo z razširitvenim modulom PiFace Digital 2 uporablja protokole I²C, SPI in 1-wire. 1-wire je vodilo za komunikacijo med napravami, ki omogoča počasni prenos podatkov, signalov in napajanja preko ene žice. Uporablja se predvsem za komunikacijo z enostavnimi merilniki kot na

primer digitalni termometer [13]. I²C je vodilo, ki se ga uporablja za povezavo manj zahtevnih integriranih vezij in mikrokontrolerjev [14]. Vodilo SPI se uporablja za komunikacijo v vgrajenih sistemih, kot je na primer komunikacija s pomnilniško kartico ali pa z zaslonom LCD [15]. Računalnik Raspberry Pi bo z razširitvenim modulom PiFace Digital 2 komuniciral z uporabo vodila SPI.

Za aktivacijo vmesnikov I²C, SPI in 1-wire se moramo z računalnikom na daljavo prijaviti na računalnik Raspberry Pi. Priporočam uporabo programa PuTTY, ki uporablja varni protokol SSH ali Secure Shell. Na voljo je brezplačno za prenos na njihovi uradni spletni strani [16]. Ko smo program Putty namestili na računalnik, ga zaženemo kot skrbnik. V meniju, ki se nam odpre, v »Host Name« vpišemo naslov IP računalnika Raspberry Pi. Naslov IP dobimo tako, da v terminalu v računalniku Raspberry Pi vpišemo spodnji ukaz.

```
ifconfig
```

Ko smo se s programom Putty uspešno povezali z računalnikom Raspberry Pi, nas terminal vpraša po uporabniškem imenu in geslu. Privzeto uporabniško ime je »pi« in geslo »raspberry«. V terminal nato vpišemo ukaz

```
sudo nano /boot/config.txt
```

To nam omogoči urejanje različnih nastavitev in dodatkov. Za potrebe programskega okolja CODESYS moramo aktivirati i²c, SPI in 1-wire. Poiskati in urediti moramo naslednje tri vrstice.

- `dtoverlay=i2c_arm=on,`
- `dtoverlay=spi=on,`
- `dtoverlay=w1-gpio-pullup,pullup=1.`

Prepričamo se, da te tri vrstice niso onemogočene z znakom #.

5.2.2 Namestitev knjižnic na računalnik Raspberry Pi

V programskem okolju CODESYS izberemo možnost »Update RaspberryPI« v meniju »Tools«. Vnesemo privzeto uporabniško ime »pi« in geslo za prijavo »raspberry« ter naslov IP računalnika Raspberry Pi. S potrditvijo na »OK« bo

programsko okolje CODESYS namestilo knjižnice in potrebne pakete. Po končanem postopku je računalnik Raspberry Pi potrebno še ponovno zagnati.

5.2.3 Statični naslov IP

Priporočljivo je, da na našem računalniku Raspberry Pi nastavimo tudi statični naslov IP, saj si bomo s tem olajšali delo v programskem okolju CODESYS. Sporazumevanje med programskim okoljem CODESYS in računalnikom Raspberry Pi namreč poteka preko mrežne povezave. V primeru, da statičnega IP-ja nismo nastavili, se je pred nalaganjem kode vedno potrebno prepričati, da je v programskem okolju CODESYS nastavljen pravilni naslov IP za računalnik Raspberry Pi.

Nastavljanje statičnega IP-ja je različno za vsak usmerjevalnik. Statični IP se nastavlja v nastavitvah DHCP strežnika.

6 Aplikacija v programskem okolju CODESYS

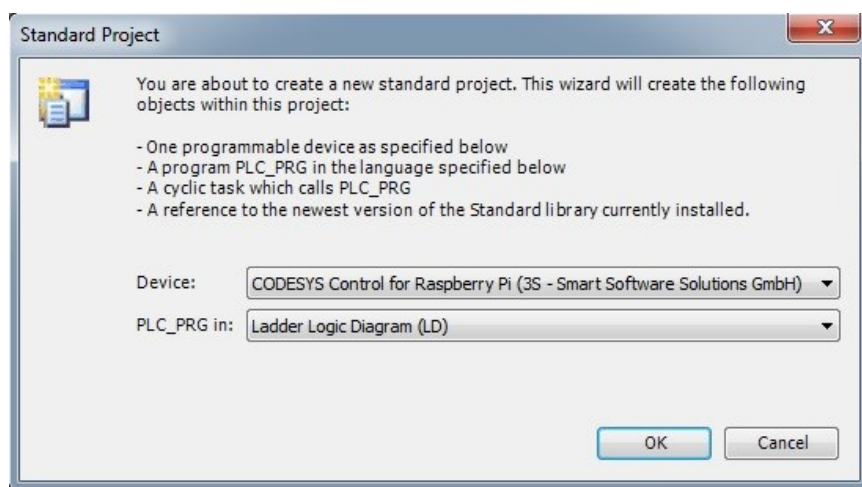
Ko smo opravili z vsemi namestitvami, lahko delovanje preizkusimo z enostavno aplikacijo, ki jo napišemo v lestvičnem diagramu z uporabo programskega okolja CODESYS. Za prvi preizkus sem razvil aplikacijo z uporabo lestvičnega diagrama, ki ob pritisku določene tipke na razširitvenem modulu PiFace Digital 2 prižge določeno LED diodo. S to enostavno aplikacijo sem preizkusil, če razširitveni modul deluje pravilno ter če komunikacija med programskim okoljem CODESYS in računalnikom Raspberry Pi poteka pravilno.

6.1 Nastavljanje novega projekta

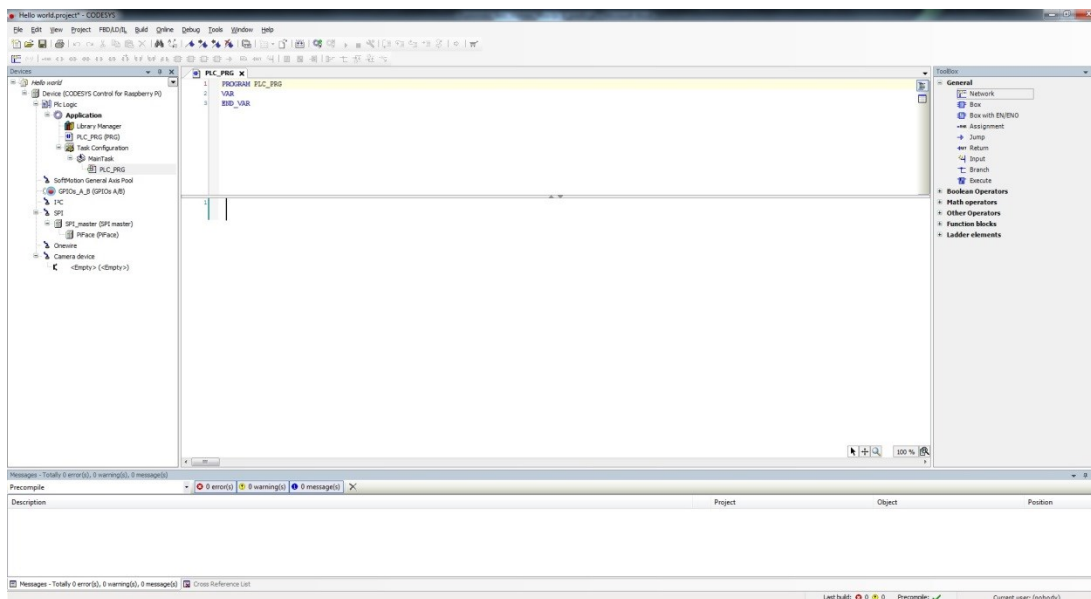
Pred vsakim razvijanjem kode je potrebno pravilno nastaviti projekt v programskem okolju CODESYS. Za pravilno delovanje vsak nov projekt zaženemo po naslednjih korakih.

1. Zaženemo programsko okolje CODESYS in v meniju »File« izberemo »New Project«.
2. Za predlogo vzamemo »Standard Project«.
3. V naslednjem meniju, ki je prikazan na sliki 16, nas program vpraša s katero napravo želimo delati in v katerem programskem jeziku želimo pisati. Za delo z računalnikom Raspberry Pi izberemo napravo »CODESYS Control for Raspberry Pi« in programski jezik »Ladder Logic Diagram (LD)«.
4. Ko program opravi prve nastavitve, moramo določiti še napravo SPI. Z desnim klikom na seznam »SPI« izberemo »Add Device« in enkrat kliknemo na »SPI Master«.

5. Za tem z desnim klikom na »SPI_Master (SPI master)« enako kot pri prejšnjem koraku dodamo še napravo »PiFace«. Na sliki 17 je pravilno nastavljen projekt z dodano napravo »PiFace«.
6. V meniju »Device (CODESYS Control for Raspberry Pi)« nastavimo naslov IP računalnika Raspberry Pi.
7. Na seznamu dvakrat kliknemo na »PLC_PRG« in v oknu, ki se nam odpre, lahko začnemo s pisanjem lestvičnega diagrama.



Slika 16: Izbira naprave in jezika

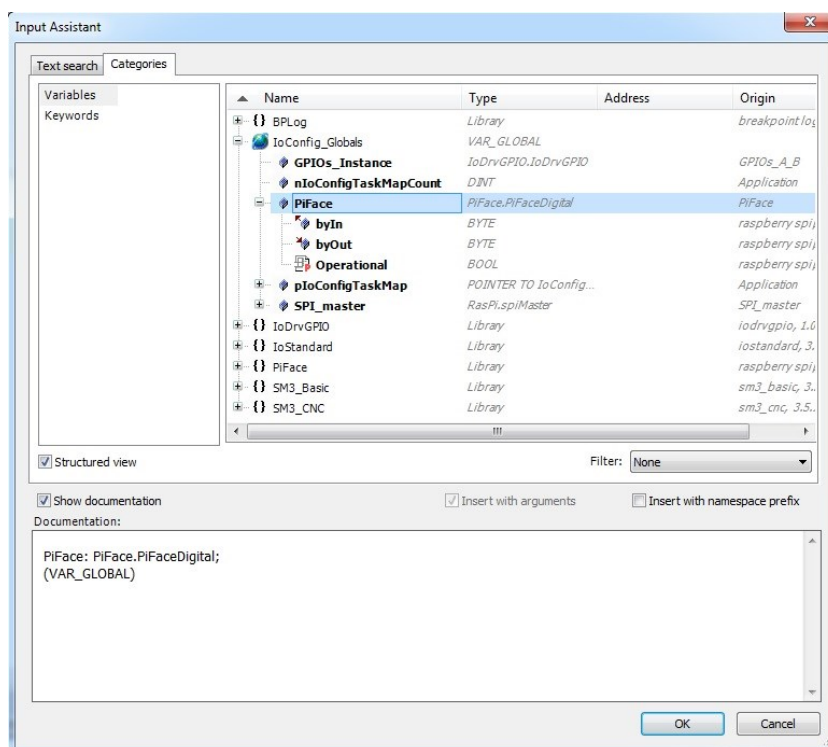


Slika 17: Pravilno nastavljen projekt

6.2 Pisanje lestvičnega diagrama

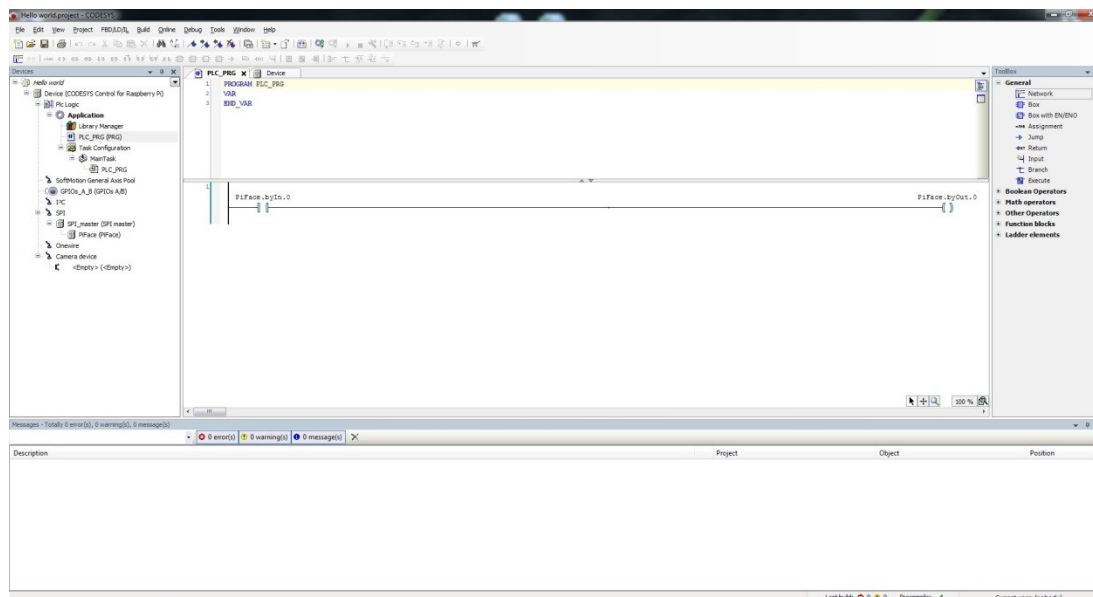
Za aplikacijo sem napisal lestvični diagram, s katerim se ob pritisku tipke S0 na razširitvenem modulu PiFace Digital 2 prižge svetleča dioda LED0. Vhodi razširitvenega modula PiFace Digital 2 so v knjižnici programskega okolja CODESYS definirani kot PiFace.byIn in oštevilčeni s številkami od 0 do 7. Ker so tipke vezane na prve štiri vhode, jih upravljamo s sklicem na PiFace.byIn in številko od 0 do 3. Izhodi pa so v knjižnici definirani kot PiFace.byOut in oštevilčeni s številkami od 0 do 7. S sklicevanjem na izhode upravljamo tudi z diodami, saj so le-te vezane na izhode razširitvenega modula PiFace Digital 2.

Lestvični diagram za aplikacijo napišemo tako, da v omrežje diagrama dodamo kontakt, s katerim bomo upravljali s tipko S0, in ga povežemo s tuljavo, s katero bomo upravljali svetlečo diodo LED0. Najprej dodamo kontakt in vpišemo sklic PiFace.byIn.0. Sklic lahko opravimo tudi s pritiskom na »...« zraven dodanega kontakta in nato s seznama, ki nam ga ponudi »Input Assistant« na sliki 17, izberemo pravi sklic.



Slika 18: Input Assistant

Na kontakt PiFace.byIn.0 nato povežemo tuljavo, s katero bomo upravljali svetlečo diodo LED0. To naredimo s sklicem na PiFace.byOut.0. S tem smo napisali lestvični diagram, ki je prikazan na sliki 19.



Slika 19: Aplikacija

6.3 Nalaganje aplikacije

Aplikacijo je nato potrebno še naložiti na računalnik Raspberry Pi in jo zagnati. To storimo po naslednjih korakih:

1. prepričamo se, da je računalnik Raspberry Pi zagnan in da je nastavljen pravilen naslov IP. Če nimamo več priključenega zaslona, lahko v programskem okolju CODESYS v meniju »Device (CODESYS Control for Raspberry Pi)« preverimo, če je naš računalnik Raspberry Pi dosegljiv.
2. Aplikacijo najprej prevedemo s pritiskom na gumb »Build«, ali pa s pritiskom tipke »F11«. S tem programsko okolje CODESYS preveri tudi vse sklice in povezave. Če je prišlo do napake, nas na to opozori.
3. Nalaganje aplikacije na računalnik Raspberry Pi izvršimo s pritiskom na tipko »Login«, ali pa s pritiskom kombinacije tipk »Alt+F8«.

4. Zagon aplikacije opravimo s pritiskom na gumb »Start« ali s tipko »F5«.

Če smo vse storili pravilno in nam programsko okolje CODESYS ni vrnilo nobenih napak, lahko na računalniku Raspberry Pi preizkusimo ali aplikacija deluje. S pritiskom na tipko S0 bi na razširitvenem modulu PiFace Digital 2 morala zasvetiti svetleča dioda LED0. S tem smo razvili prvo enostavno aplikacijo in hkrati preizkusili, če razširitveni modul PiFace Digital 2 deluje.

7 Upravljanje zapornice z računalnikom Raspberry Pi

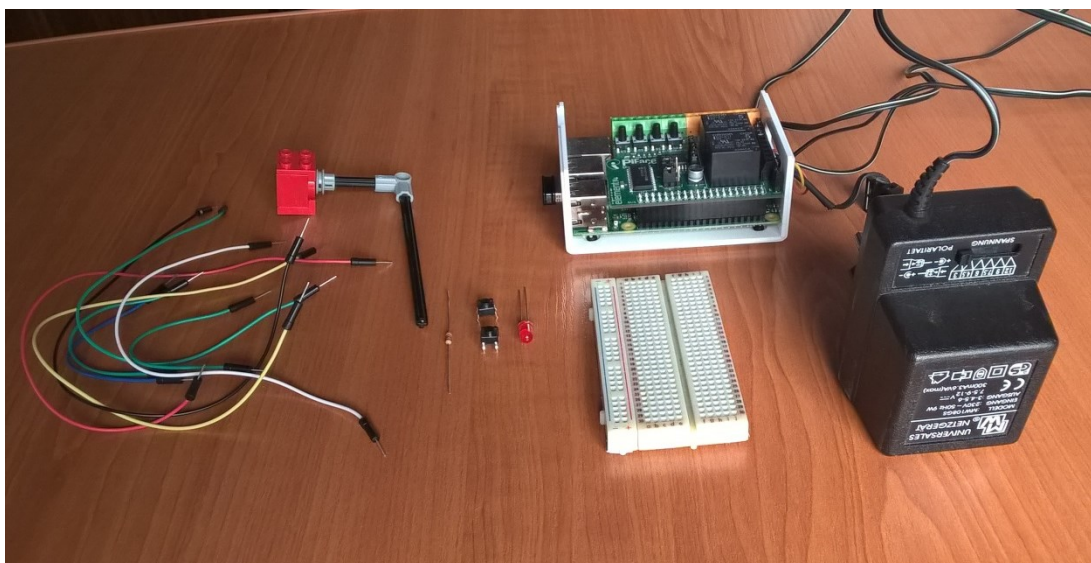
Da bi računalnik Raspberry Pi lahko primerjal s PLK-jem, moramo v aplikacijo vključiti tudi z zunanje vhodne in izhodne elemente. Razširitveni modul PiFace Digital 2 ima na voljo 8 izhodov, ki jih lahko priključimo na 5-V vijačno sponko. Na izhode bi torej lahko priključil koračni motor, ki bi ga vodil s klicem na zelen izhod. Odločil sem se, da bom izhode uporabil le za vezavo svetleče diode, saj se zaradi nizke napetosti zelo omeji nabor elektromotorjev. Poleg tega lahko z zunanjim napajalnikom, ki ga priključimo na releje, nastavljamo poljubno napetost. Na voljo imamo tudi 8 vhodov, na katere je mogoče priključiti tipke, senzorje in stikala, ki sprožijo ali prekinejo proces v aplikaciji.

Za preizkus, kako se računalnik Raspberry Pi obnese kot PLK, sem se odločil, da uporabim lestvičnega diagrama razvijem aplikacijo za dviganje in spuščanje zapornice. Tako bo računalnik Raspberry Pi deloval z zunanjim vhodnim elementom, ki bo sprožil delovanje zapornice, in elektromotorjem, ki bo zapornico dvigoval in spuščal.

7.1 Izdelava zapornice

Zapornico sem izdelal v pomanjšani verziji z uporabo kock Lego. Celotna zapornica je enostavne izdelave, saj je bolj kot videz pomembna njena funkcionalnost. Vse elemente, ki sem jih uporabil za izdelavo, vidimo na sliki 20. Za pogon zapornice sem se po dolgem razmisleku odločil za majhen 9-V elektromotor, prav tako iz programa Lego. Za tak elektromotor sem se odločil zato, ker ima zelo majhno vrtilno hitrost. Drugi manjši nizkonapetostni motorji imajo v večini primerov veliko vrtilno hitrost in bi bilo zato dvigovanje in spuščanje zapornice izredno hitro.

Da bi vrtljaje zmanjšal, bi poleg elektromotorja moral uporabiti tudi reduktor. Za proženje dviga zapornice sem izbral manjšo tipko, za indikacijo obratovanja zapornice pa sem priključil še zunanjo svetlečo diodo, ki med obratovanjem zapornice utripa in nas tako opozori na nevarnost nenadnega spusta. Kot varnostno stikalo, ki prekine spuščanje zapornice, sem prav tako uporabil zunanjo tipko. S tem sem želel simulirati oviro, ki se nahaja v delovnem območju zapornice. Vse skupaj sem na razširitveni modul PiFace Digital 2 povezal s pomočjo testne ploščice.

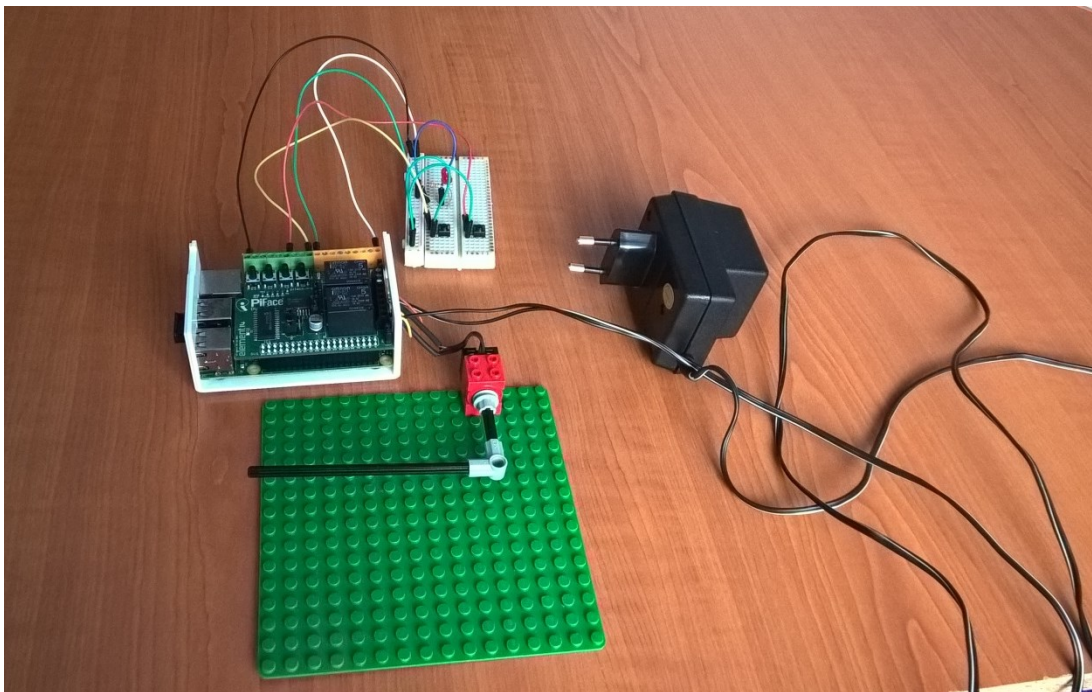


Slika 20: Elementi zapornice

7.2 Vezava zapornice

Da bi elektromotor lahko vrtel v obe smeri, sem moral pravilno povezati zunanji napajalnik na releje. V primeru, kjer bi zadostovalo vrtenje motorja v eno smer, bi elektromotor lahko povezal na 5-V izhodno vijačno sponko na razširitvenem modulu PiFace Digital 2. Ker pa sem želel simulirati delovanje zapornice, sem moral zagotoviti pravilno vezavo zunanjega napajalnika na oba releja. Na skupna kontakta sem povezal motor, preko mirovnih in delovnih kontaktov pa sem ustvaril H-mostič. Z izmeničnim vključevanjem relejev sem tako lahko motor poganjal v obe smeri. Na eno vhodno sponko na razširitvenem modulu PiFace Digital 2 sem povezal tipko, ki požene aplikacijo. Na drugo vhodno sponko sem povezal drugo tipko, ki prekine

spuščanje zapornice in tako deluje kot varnostno stikalo. Na izhodno sponko pa sem povezal še svetlečo diodo. Vse pravilne vezave so vidne na sliki 21.

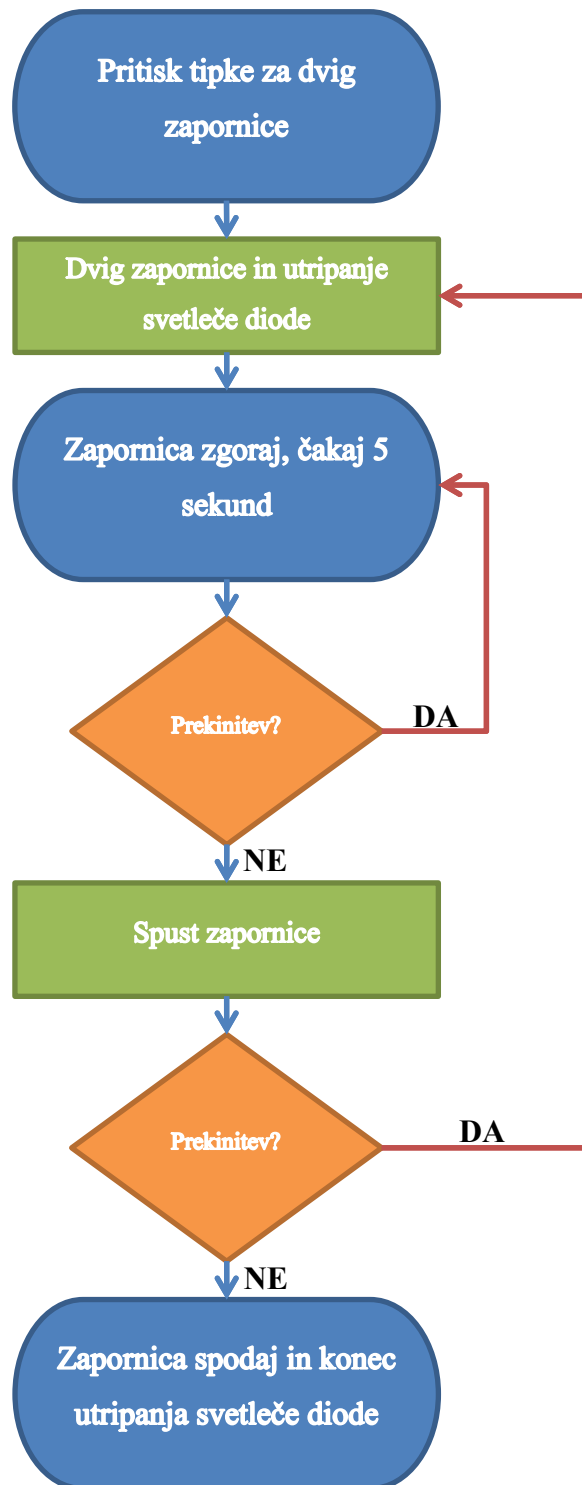


Slika 21: Sestavljena zapornica

7.3 Izvedba aplikacije za delovanje zapornice

Z aplikacijo sem se želel kar najbolj približati delovanju zapornic, kot jih vidimo na parkiriščih. Ko se na parkirišču približamo zapornici, največkrat pritisnemo tipko, vzamemo parkirno kartico, ali pa pokličemo na dodeljeno telefonsko številko. S tem sprožimo dvig zapornice. Elektromotor v zapornici deluje toliko časa, dokler zapornica ne doseže vertikalnega položaja. Ko je enkrat v takem položaju, zapornica počaka določeno število sekund, nato pa, ob pogoju, da je pot prosta, elektromotor požene v obratno smer in zapornico zopet postavi v horizontalni položaj. V primeru, ko pot ni prosta, torej se pod zapornico nahaja avtomobil ali oseba, se zapornica ne spusti. Prav tako sem želel doseči, da se zapornica dvigne nazaj v vertikalni položaj, če pride do prekinitve med spuščanjem. Prekinitve v resničnem primeru povzroči nek objekt, ki se je znašel v delovnem območju

zapornice. Takšno prekinitev sem simuliral z uporabo druge tipke. Za lažjo predstavo kako aplikacija deluje, sem najprej naredil diagram poteka, ki je na sliki 22.



Slika 22: Diagram poteka

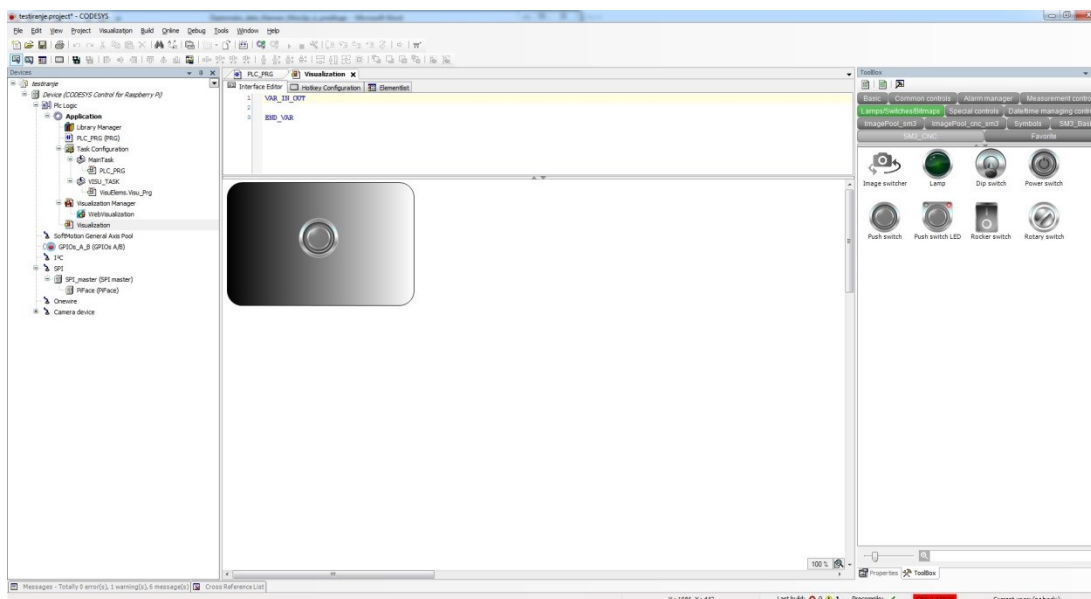
Ko sem določil, kako si bodo dogodki sledili ob zagonu aplikacije, je bilo potrebno napisati še lestvični diagram. V lestvičnem diagramu sem najprej s kontaktom realiziral branje tipke za dvig zapornice. Da prvi kontakt prevaja, je sploh predpogoj za zagon aplikacije. Čas delovanja motorja, ki je potreben za doseg zelenega položaja sem določil na podlagi preizkusov. S časovniki sem prekinil delovanje elektromotorja, ko je zapornica dosegla vertikalni položaj. Enako sem storil tudi za vračanje zapornice nazaj v horizontalni položaj. Namesto časovnikov bi delovanje elektromotorja lahko prekinil z uporabo bližinskih merilnikov in bi se tako zapornica vedno ustavila v istem položaju. Ker sem se odločil za časovnike, sem zapornico načrtoval tako, da sem ustvaril fizično oviro za končni in za začetni položaj. V primeru, da bi uporabil napačen čas delovnega cikla motorja, bi se zapornica lahko znašla izven delovnega območja. Kombinacijo časovnikov sem uporabil tudi za realizacijo utripanja svetleče diode, s katero zapornica opozarja na delovanje. Dva časovnika izmenično prevajata povezavo v razmiku ene sekunde in s tem ustvarita utripanje. V lestvični diagram sem moral vključiti tudi pogoj, ki dovoljuje spust zapornice. Zadnji korak aplikacije, torej spuščanje zapornice, se izvrši šele, ko v delovnem območju dlje časa ni ovire, ki bi prekinila kontakt. Poleg tega se zapornica dvigne nazaj v vertikalni položaj, če se kontakt prekine med spuščanjem. Celoten lestvični diagram aplikacije je v Dodatku.

7.4 Simulacija aplikacije

Programsko okolje CODESYS nam ponuja tudi orodje za vizualizacijo in simulacijo aplikacije. To orodje je zelo priročno, saj nam pred zagonom aplikacije na PLC-ju omogoča, da pregledamo, če smo kje napravili napake oziroma kako potuje signal po lestvičnem diagramu.

Z desnim klikom na »Application« iz seznama izberemo možnost »Add object« in »Visualization«. V seznam smo tako dodali možnost »Visualization«, kjer lahko ustvarimo virtualno tipko za zagon aplikacije. Poleg tipke lahko dodamo tudi napise, merilnike, kazalnike statusa itn. Za potrebe simulacije je dovolj le virtualna tipka, ki bo namesto tipke na razširitvenem modulu PiFace Digital 2 zagnala aplikacijo.

Na seznamu se pomaknemo v okno »Visualization«, kjer lahko v meniju »ToolBox« dodamo ozadje in tipko. Na sliki 23 je virtualna tipka »Push switch«, ki bo pognala simulacijo aplikacije. Tipki je potrebno še določiti ime, na katerega se bomo sklicevali v lestvičnem diagramu. To storimo tako, da na desni strani v vizualizaciji v meniju »Properties« napišemo ime v možnosti »Variable«.



Slika 23: Virtualna tipka "Push switch"

V lestvičnem diagramu je nato treba nov kontakt postaviti vzporedno s kontaktom »PiFace.byIn.0« in se sklicevati na virtualno tipko. V mojem primeru sem vpisal sklic »xSwitch«, kot je vidno na sliki 24.

Ob pritisku na tipko se aplikacija zažene in pravilno sledi dogodkom, ki sem jih realiziral z uporabo lestvičnega diagrama.

8 Zaključek

Cilj mojega diplomskega dela je bil preizkusiti računalnik Raspberry Pi z razširitvenem modulom PiFace Digital 2 v vlogi PLK-ja in hkrati bralca naučiti rokovanja z njim. V besedilu sem predstavil zmogljivost računalnika Raspberry Pi in opisal, kako sem ga preizkusil v vlogi PLK-ja. Besedilo sem poskusil podajati na način, ki ga razume vsak, tudi nekdo, ki se še nikoli ni srečal z računalnikom Raspberry Pi ali s programskim okoljem CODESYS.

Sama zmogljivost računalnika Raspberry Pi me je pozitivno presenetila, saj je izredno odziven. Za ceno okoli 40 € dobimo računalnik, ki z lahkoto konkurira veliko dražjim prenosnikom. Uporabil sem ga za različne namene, s katerimi se sreča povprečen uporabnik. Od predvajanja filmov, do brskanja po spletu, urejanja dokumentov in poslušanja glasbe. Zmogljivost gre pripisati tudi operacijskemu sistemu, ki temelji na sistemu Linux, ki ni tako zahteven kot na primer operacijski sistem Windows.

Z razširitvenim modulom PiFace Digital 2 pa računalnik Raspberry Pi pretvorimo v PLK, s katerim lahko upravljamo z zunanjimi napravami. Tak PLK ne bo ravno nadomestil industrijskih krmilnikov, vendar pa je za domačo uporabo in za potrebe v šolstvu po mojem mnenju zadovoljiv. Kot sem prikazal z aplikacijo, omogoča programsko okolje CODESYS programiranje z lestvičnim diagramom, ki je za neizkušene uporabnike najbolj primeren. Z lestvičnimi diagrami smo se srečali že pri predmetu Računalniško vodenje procesov, torej bi računalnik Raspberry Pi lahko vsaj v začetni fazi nadomestil PLK-je, ki se pri vajah trenutno uporabljajo.

Če primerjamo računalnik Raspberry Pi z industrijskimi PLK-ji, je hitro opaziti nekaj bistvenih prednosti in slabosti. Ena največjih prednosti računalnika Raspberry Pi je njegova nizka cena v primerjavi z na primer krmilniki Siemens. V primeru, kjer

potrebujemo le malo število vhodnih in izhodnih elementov, je računalnik Raspberry Pi povsem enakovreden ostalim PLK-jem. Prav tako je prednost računalnika Raspberry Pi v njegovi velikosti in mobilnosti, saj ga lahko priključimo na baterijsko napajanje, kjer konstantna povezava na električno omrežje ni izvedljiva.

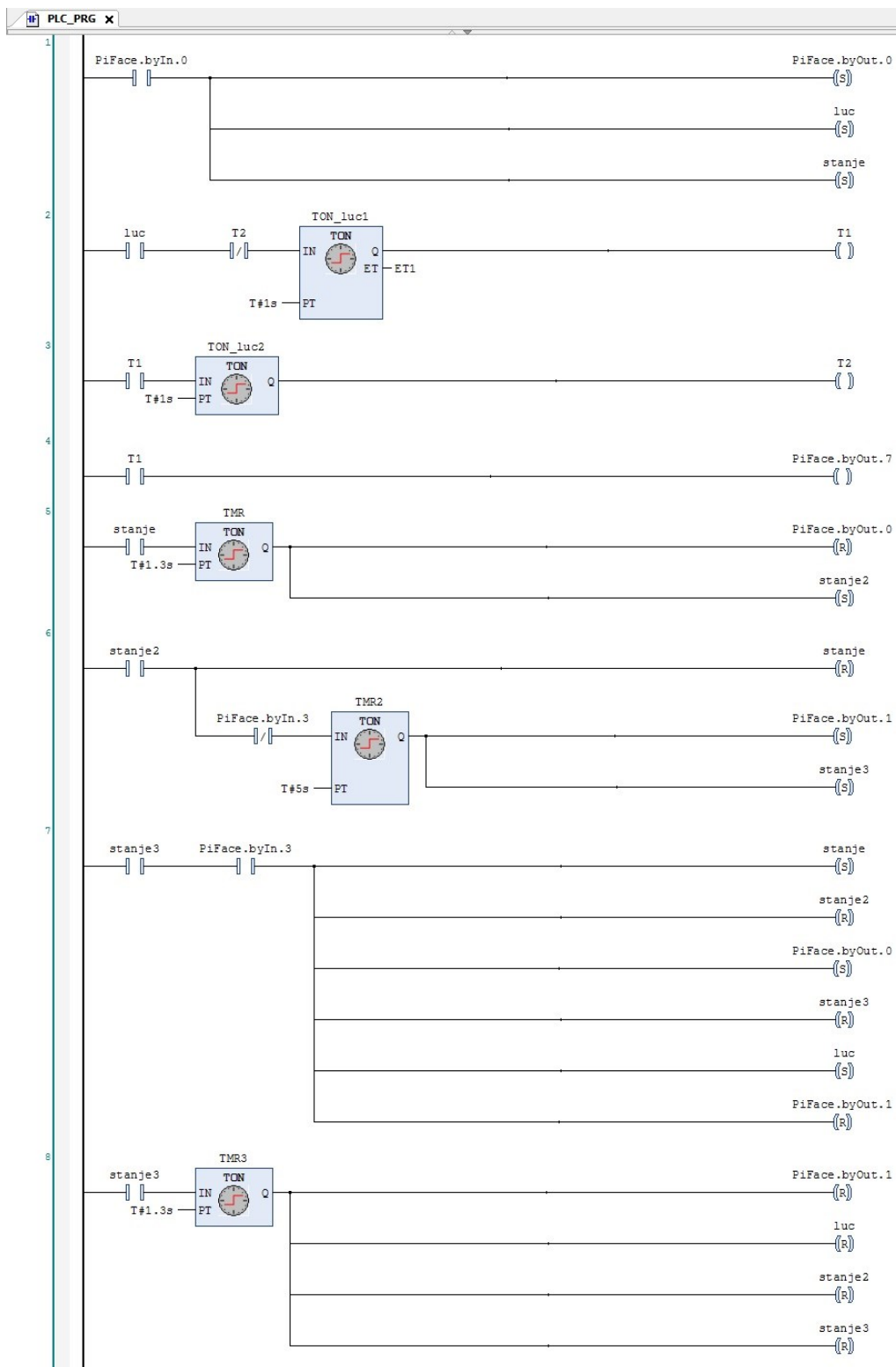
Slabosti računalnika Raspberry Pi v vlogi PLK-ja pa se pokaže predvsem v omejenem številu vhodov in izhodov, ter v (ne)robustnosti izdelave. V industrijskem okolju bi bil računalnik Raspberry Pi kot PLK neprimeren, saj ne vzdrži tako ekstremnih temperaturnih razlik, prahu in konstantne obremenjenosti. Računalnik Raspberry Pi ni izdelan namensko za uporabo kot PLK in je zato tudi manj zmogljiv kot PLK-ji proizvajalcev, kot so Siemens, Hitachi, Omron, Mitsubishi in ostali.

V ostalih aspektih je težko primerjati računalnik Raspberry Pi s PLK-ji, saj se moramo odločiti, kakšno aplikacijo želimo razviti in kaj želimo, da nam PLK ponuja. Če nam je dovolj 8 vhodov in izhodov, je računalnik Raspberry Pi po mojem mnenju dobra izbira, če aplikacija ni razvita za industrijski namen. Programsko okolje CODESYS nam ponuja tudi možnost simulacije in vizualizacije, kot lahko simuliramo delovanje večino PLK-jev.

Vsestranska uporabnost računalnika Raspberry Pi bi se po mojem mnenju najbolje obnesla v avtomatizaciji doma. Programsko okolje CODESYS ponuja tudi orodje za vizualizacijo aplikacije, s katerim lahko izdelamo grafični uporabniški vmesnik. Če temu dodamo še povezljivost računalnika Raspberry Pi preko modula WiFi in možnost nadgradnje z zaslonom na dotik, dobimo napravo, ki jo lahko uporabimo za nadzorni sistem celotnega stanovanja. Tu predvsem mislim na avtomatizacijo ogrevanja oziroma hlajenja posameznih prostorov, preverjanja vključenosti naprav in avtomatizacijo senčil. V prihodnosti bom poskusil realizirati vsaj en tak projekt. Z nadgradnjo aplikacije, ki sem jo razvil v okviru diplomskega dela, bi lahko avtomatiziral dviganje in spuščanje okenskih rolet. Možnosti uporabe računalnika Raspberry Pi kot PLK so zelo obetavne. Uporabniki po vsem svetu vsakodnevno objavljajo svoje ideje in nove načine uporabe računalnika Raspberry Pi.

Dodatek

Na sliki je prikazan lestvični diagram aplikacije za zapornico, ki sem ga napisal v programskem okolju CODESYS.



Literatura

- [1] W. Bolton, Programmable logic controllers, Fifth edition, Amsterdam: Elsevier Newnes, 2010.
- [2] Dag H. Hanssen, Programmable logic controllers: A practical approach to IEC 61131-3 using CoDeSys, Chichester: Wiley, 2015.
- [3] J. Stenerson, Fundamentals of programmable logic controllers, sensors, and communications, Third edition, New Jersey: Pearson Prentice Hall, 2004.
- [4] V. Logar in G. Mušič. (2015). Računalniško vodenje procesov: praktikum [Elektronski vir]. Dosegljivo: http://msc.fe.uni-lj.si/Download/Logar/RVP_skripta_2015.pdf. [Dostopano 12. 12. 2015].
- [5] S. McManus in M. Cook, Raspberry Pi for dummies, New Jersey: Wiley, 2013.
- [6] Raspberry Eye In The Sky [Online]. Dosegljivo: <http://www.daveakerman.com/?p=1154>. [Dostopano: 15. 1. 2016].
- [7] University of Southampton, Raspberry Pi at Southampton [Online]. Dosegljivo: <http://www.southampton.ac.uk/~sjc/raspberrypi/>. [Dostopano: 15. 1. 2016].
- [8] FishPi, An autonomous drop in the ocean [Online]. Dosegljivo: <http://fishpi.org/>. [Dostopano: 15. 1. 2016].
- [9] Raspberry Pi, Raspbian [Online]. Dosegljivo: <https://www.raspberrypi.org/downloads/raspbian/>. [Dostopano: 6. 12. 2015].
- [10] Win32 Disk Imager [Online]. Dosegljivo: <https://sourceforge.net/projects/win32diskimager/>. [Dostopano: 6. 12. 2015].

- [11] PiFace Digital 2 for Raspberry Pi – Getting started [Online]. Dosegljivo: <http://www.farnell.com/datasheets/1881551.pdf>. [Dostopano: 25. 1. 2016].
- [12] CODESYS Development System [Online]. Dosegljivo: <http://store.codesys.com/codesys.html>. [Dostopano: 6. 12. 2015].
- [13] 1-Wire [Online]. Dosegljivo: <https://en.wikipedia.org/wiki/1-Wire>. [Dostopano: 27. 3. 2016].
- [14] I²C [Online]. Dosegljivo: <https://en.wikipedia.org/wiki/I%C2%B2C>. [Dostopano: 27. 3. 2016].
- [15] Serial Peripheral Interface Bus [Online]. Dosegljivo: https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus. [Dostopano: 27. 3. 2016].
- [16] PuTTY [Online]. Dosegljivo: <http://www.putty.org/>. [Dostopano: 6. 12. 2015].