

TEHNIČKO VELEUČILIŠTE U ZAGREBU

SEMINAR

Automatsko sortiranje kutija QR kodovima

Marko Kljajić

Voditelj: *prof. dr. sc. Dario Matika*

Zagreb, ožujak 2023.

SADRŽAJ

1. Uvod	1
2. Raspberry pi	3
2.1. Priprema OS-a	3
2.2. Konfiguriranje mrežnih postavki	4
2.3. Snap7 modul	5
2.4. OpenCV	6
2.5. PyQt5	6
3. TIA Portal	8
3.1. Priprema TIA portala za komunikaciju	9
4. Dodatak	12
4.1. Raspberry pi kod	12
4.2. TIA portal program i podatkovni blokovi	17

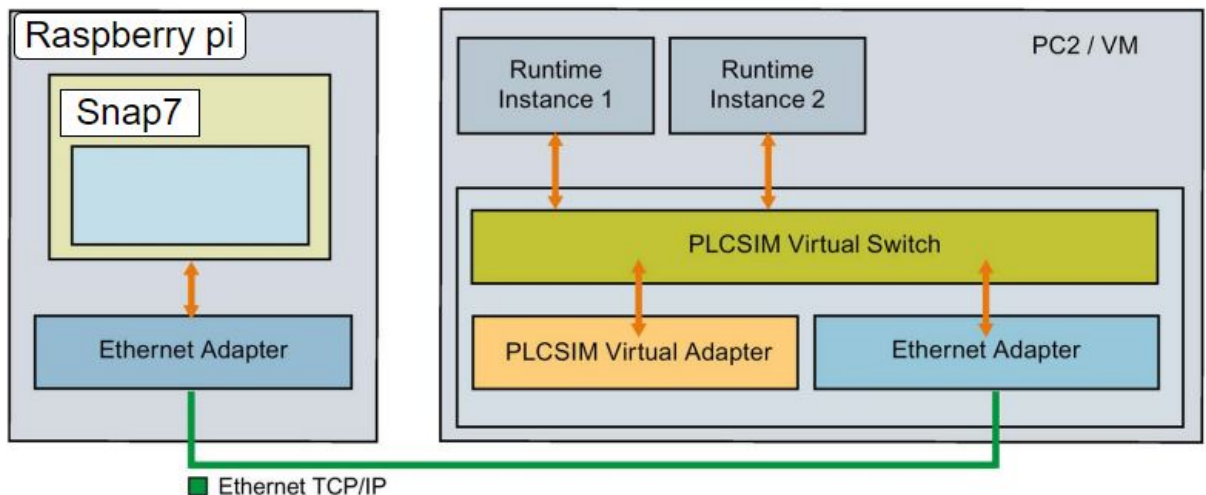
1. Uvod

Rad obuhvaća povezivanje više programskih platformi od kojih se dio simulira u odgovarajućem programskom paketu, a dio odvija u stvarnosti. Zadatak je bio povezati PLC (Simatic S7-1500) i mikroračunalo (Raspberry pi 4, model B) koji zajedno posreduju jednostavnom sortirnom linijom. Linija se sastoji od pokretne trake na kojoj se prenose dva tipa kutija ("Kutija A/ Kutija B"), optičkog prolaznog senzora, pneumatskog cilindra i ormara s tipkalima za pokretanje odnosno zaustavljanje. Difuzni optički senzor u prisutnosti predmeta daje logičku jedinicu, pneumatski cilindar je upravljani elektropneumatskim monostabilnim 5/2 razvodnikom. Tipkala (Start/Stop) su u normalnom stanju otvorena dok je isključak u slučaju nužde po standardu normalno zatvorenog kontakta. Kontrola toka kutija se odvija putem QR kodova putem kamere koja je spojena na sam raspberry pi te je na njemu i HMI. HMI-om se prikazuje brojno stanje kutija po tipu i moguće je pokrenuti ili zaustaviti proces.



Slika 1.1: Izgled sortirne linije u FactoryIO programskom okruženju.

Nakon pokretanja linije kutije na mjestu za skeniranje zaustavlja optički senzor te u slučaju "Kutije A" pneumatski cilindar treba kutiju odvojiti na drugu stranu. Simatic S7-1500 je programiran pomoću TIA portal-u (eng. Totally Integrated Automation Portal) i spaja se na PLCSIM Advanced programski paket za simuliranje rada stvarnog PLC-a. PLCSIM Advanced kreira virtualni PLC s određenom IP adresom na koji možemo interno mrežno spustiti program kao na pravi PLC. Oni su još spojeni s FactoryIO programom koji omogućuje simulirani prikaz same sortirne linije. Raspberry pi spajamo pomoću ethernet adaptera na istu lokalnu mrežu pri čemu moramo voditi računa da su svi uređaji na istom "subnet-u", odnosno ručno podesiti IP adrese kako bi svi bili međusobno dohvatljivi. Raspberry pi je programiran programskim jezikom Python.



Slika 1.2: Shema povezivanja programskih paketa

U nastavku će pojedinačno biti objašnjeni programski kodovi i komunikacije za raspberry pi i PLC.

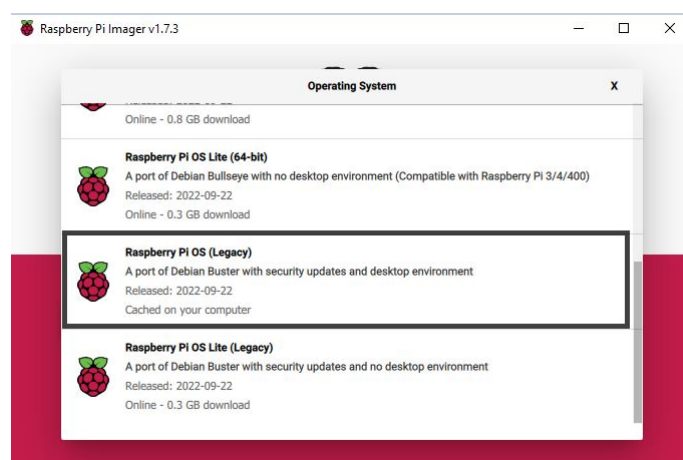
2. Raspberry pi

Prije programiranja potrebno je prilagoditi uređaj za rad instaliranjem python modula i razvojnih okruženja. Prvotno se treba kreirati OS (eng. Operating system) specifičan za raspberry pi te setom (eng. Bash) komandi dodaju moduli i postavljaju konfiguracije za samo programiranje. Potrebni moduli su:

1. Snap7
2. OpenCV
3. PyQt5

2.1. Priprema OS-a

Raspberry pi koristi prilagođenu inačicu linuxa pod imenom "Raspbian", koja je nastala od "Debian" distribucije linuxa. Metodom pokušaja/pogreške sve je radilo sa starijom verzijom OS-a (Debian version: 10 (buster)). Spuštanje OS-a najjednostavnije je napraviti s "Raspberry pi imager" programom.



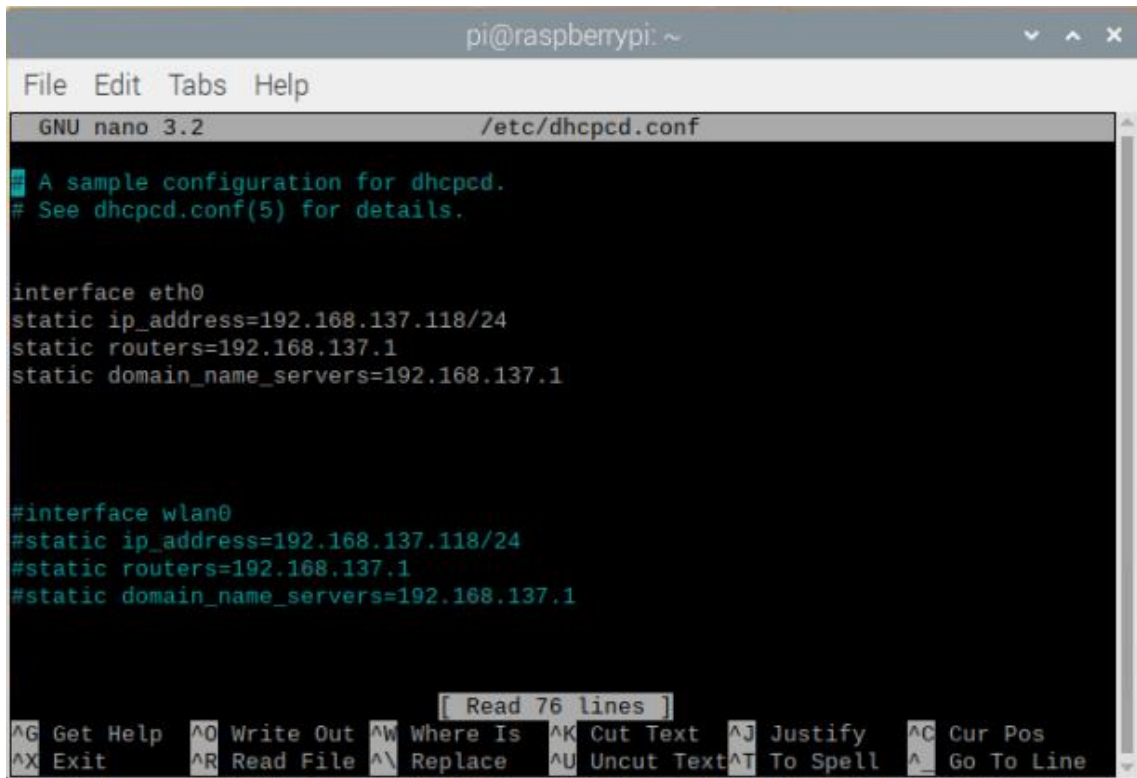
Slika 2.1: Debian version: 10 (buster)

2.2. Konfiguriranje mrežnih postavki

Slanjem slijedećih naredbi u terminal otvara na se prozor. Ovim kodom se Raspberry postavlja na LAN mrežu računala.

```
1 ~$ sudo nano /etc/dhcpd.conf
```

Listing 2.1: bash naredba



```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 3.2 /etc/dhcpd.conf

# A sample configuration for dhcpd.
# See dhcpd.conf(5) for details.

interface eth0
static ip_address=192.168.137.118/24
static routers=192.168.137.1
static domain_name_servers=192.168.137.1

#interface wlan0
#static ip_address=192.168.137.118/24
#static routers=192.168.137.1
#static domain_name_servers=192.168.137.1

[ Read 76 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Slika 2.2: Debian version: 10 (buster)

U projektu je korišten ethernet priključak te shodno tomu ključan dio koda je:

```
1 interface eth0
2 static ip_address=192.168.137.118/24
3 static routers=192.168.137.1
4 static domain_name_servers=192.168.137.1
```

Listing 2.2: dhcpd.conf datoteka

2.3. Snap7 modul

Snap7 je multi-platformski ethernet komunikacijski paket otvorenog koda dizajniran za korištenje s Siemens S7 PLC-ima. Njegovo implementiranje na Raspbian se odrađuje nizom naredbi u terminalu. Njime se mogu dohvaćati memorijski podaci iz PLC-a.

```
1 #download and compile snap7 for rpi
2
3 wget http://sourceforge.net/projects/snap7/files/1.2.1/
   snap7-full-1.2.1.tar.gz/download
4 tar -zxvf snap7-full-1.2.1.tar.gz
5 cd snap7-full-1.2.1/build/unix
6 sudo make f arm_v7_linux.mk all
7
8 #copy compiled library to your lib directories
9 sudo cp ../bin/arm_v7-linux/libsnap7.so /usr/lib/libsnap7
   .so
10 sudo cp ../bin/arm_v7-linux/libsnap7.so /usr/local/lib/
   libsnap7.so
11
12 #install python pip if you don't have it:
13 sudo apt-get install python-pip
14 sudo pip install python-snap7
15 sudo ldconfig
```

Listing 2.3: Lista naredbi za terminal

2.4. OpenCV

OpenCV (eng. Open Source Computer Vision Library) je biblioteka otvorenog koda za računalni vid i strojno učenje. Pomoću nje je odrađeno prepoznavanje QR kodova. Instaliranje biblioteke se odvija nizom slijedećih naredbi u terminalu:

```
1 sudo apt-get update
2
3 sudo apt-get install python3-opencv
4
5 sudo apt-get install libqt4-test python3-sip python3-
  pyqt5 libqtgui4 libjasper-dev libatlas-base-dev -y
6
7 pip3 install opencv-contrib-python==4.1.0.25
8
9 sudo modprobe bcm2835-v4l2
```

Listing 2.4: Lista naredbi za terminal

2.5. PyQt5

PyQt5 je peto izdanje softverskog alata koji kombinira programski jezik python i Qt alate za izradu grafičkih sučelja. Instaliranje paketa se odvija naredbom u terminalu:

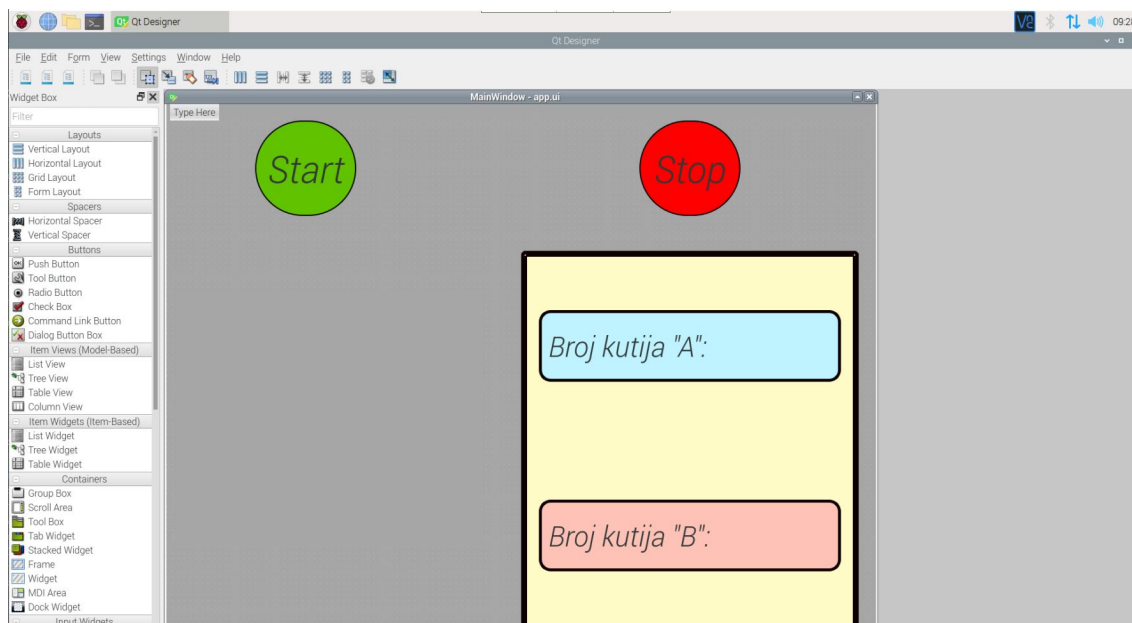
```
1 sudo apt install python3-pyqt5
```

Listing 2.5: Nareba za terminal

Kako bi se olakšala sama izrada sučelja postoji PyQt-Designer alat kojim jednostavnim DD (eng. Drag and drop) se pozicioniraju određeni elementi a njihovo uređivanje kao promjena boje, fonta, oblika, itd., koristi CSS (eng. Cascading Style Sheets). Za instaliranje PyQt-Designer-a je potrebne su slijedeće naredba:

```
1 sudo apt-get install qt5-default
2 sudo apt-get install qtcreator
```

Listing 2.6: Nareba za terminal



Slika 2.3: Izgled PyQt-Designer-a na raspberry pi-u

3. TIA Portal

TIA Portal (Potpuno integrirani automatizacijski portal) softverska je platforma za programiranje, konfiguriranje i vizualizaciju automatizacijskih sustava tvrtke Siemens. To je inženjerski okvir koji podržava automatizacijska i pogonska rješenja, kao i vizualizaciju HMI (čovjek-stroj sučelje) i SCADA (Nadzor i akvizicija podataka) sustava.

TIA Portal je osmišljen kako bi pojednostavio razvojni proces industrijskih automatizacijskih sustava pružajući jedinstvenu softversku platformu koja se može koristiti za sve automatizacijske zadatke. To znači da inženjeri i tehničari mogu koristiti jedan alat za programiranje kontrolera, konfiguriranje uređaja i dizajniranje HMI zaslona.

TIA Portal podržava niz tehnologija automatizacije, uključujući programabilne logičke kontrolere (PLC-ove), ljudsko-strojne sučelja (HMI-je), distribuirane I/O sustave, uređaje za varijabilni napon (VFD-ove) i sustave za upravljanje pokretom. Platforma također podržava industrijske komunikacijske protokole poput PROFINET-a, PROFIBUS-a i AS-Interface-a.

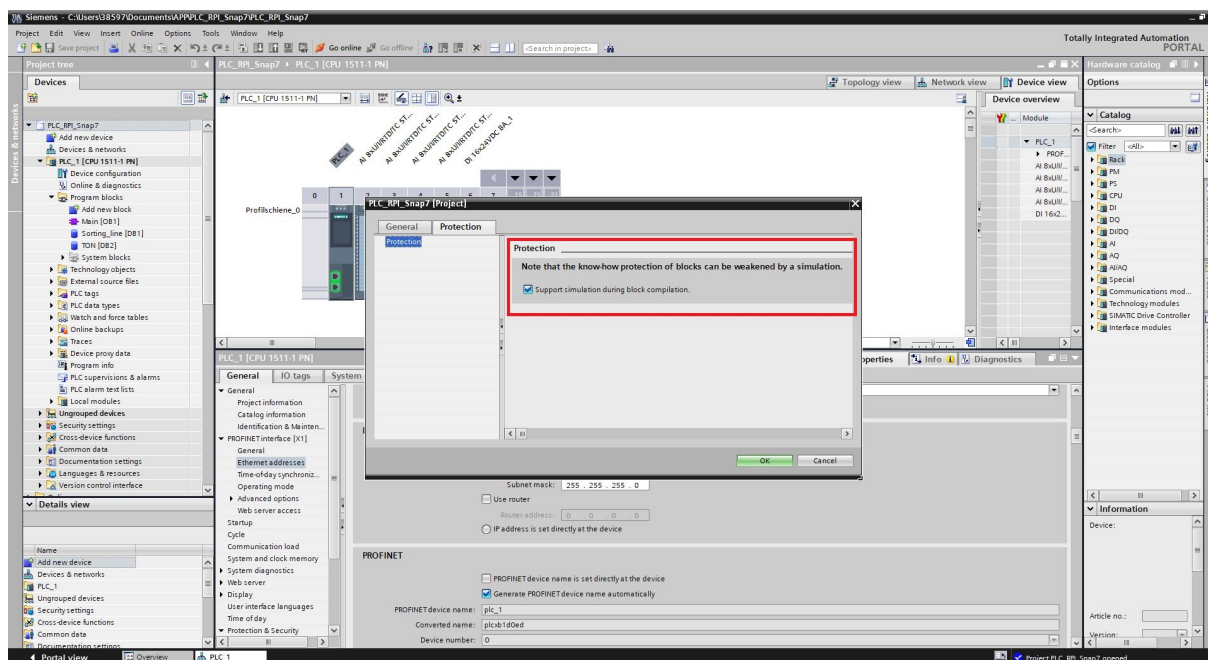
TIA Portal je organiziran u nekoliko modula, uključujući STEP 7 programsko okruženje, WinCC za dizajn i vizualizaciju HMI-a i Startdrive za konfiguriranje pogona. Ovi se moduli mogu koristiti zajedno ili neovisno, ovisno o specifičnim automatizacijskim zahtjevima projekta.

Ukupno gledano, TIA Portal je sveobuhvatna i integrirana softverska platforma koja pojednostavljuje razvojni proces automatizacijskih sustava, olakšavajući inženjerima i tehničarima dizajniranje, programiranje i održavanje industrijskih automatizacijskih sustava.

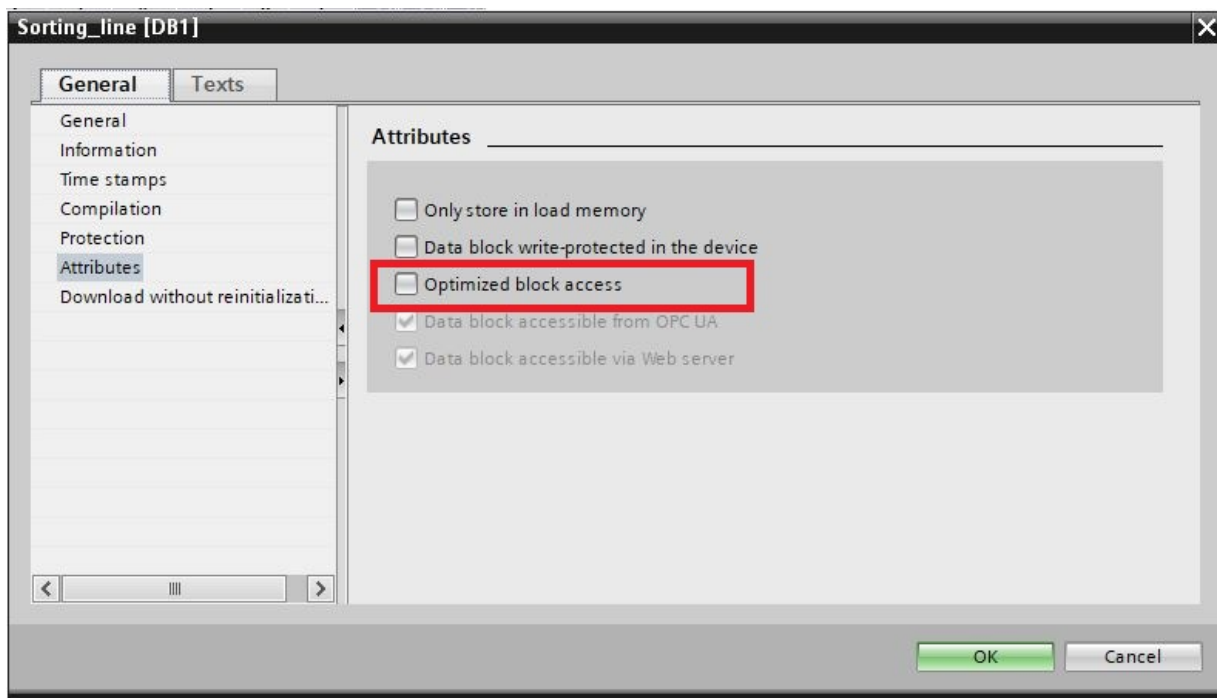
3.1. Priprema TIA portala za komunikaciju

Za uspostavu komunikacije potrebno je prilagoditi postavke u TIA portalu, odnosno:

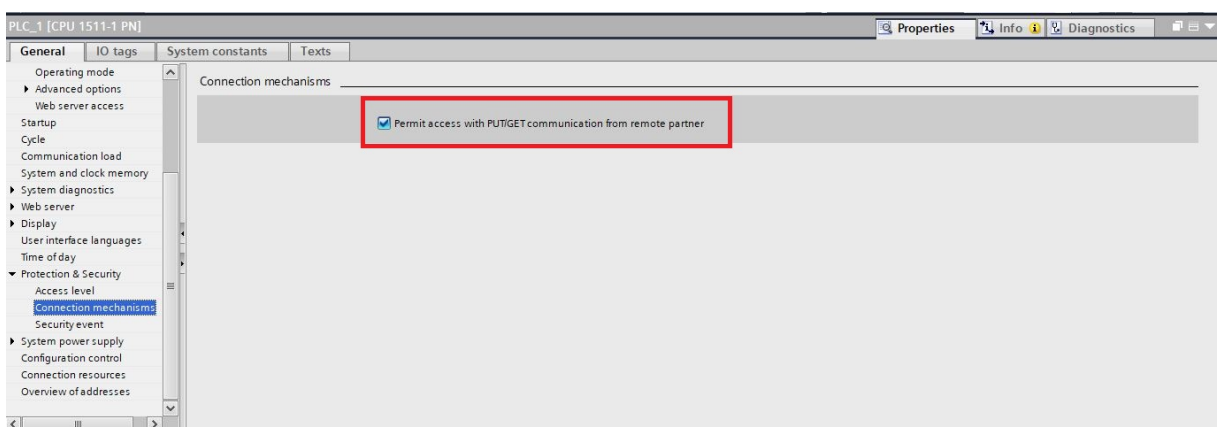
1. dozvoliti simulaciju tijekom kompajliranja blokova (Slika 3.1)
2. ukloniti optimizaciju podatkovnih blokova koji se dohvaćaju (Slika 3.2)
3. PLC konfigurirati da podržava (eng. PUT/GET) od strane povezanog uređaja (Slika 3.3)
4. podesiti IP adresu PLC-a kako bi bio na istom subnetu (Slika 3.4)



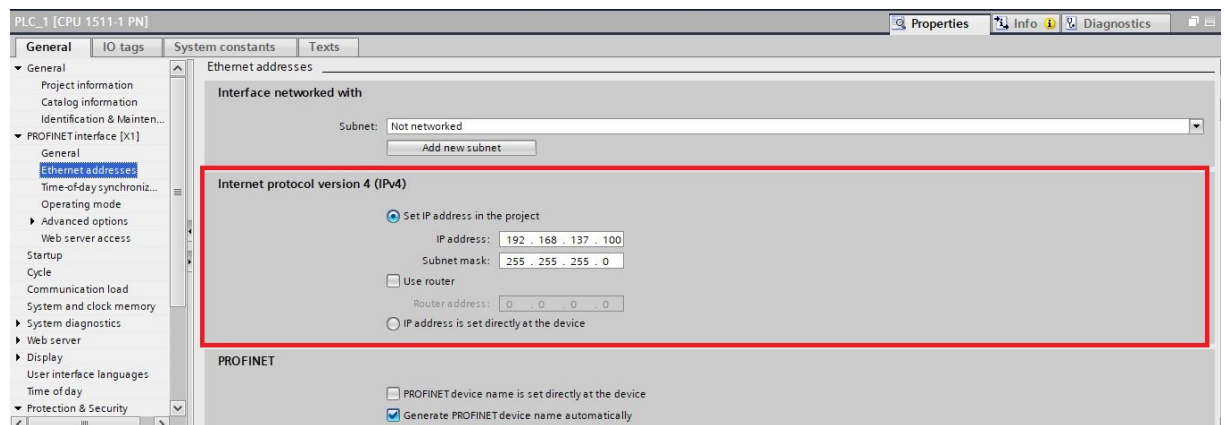
Slika 3.1: TIA portal



Slika 3.2: TIA portal



Slika 3.3: TIA portal



Slika 3.4: TIA portal

4. Dodatak

4.1. Raspberry pi kod

Glavni program:

```
1 from PyQt5.QtWidgets import QApplication, QMainWindow,  
    QLabel, QTextEdit, QPushButton, QDialog, QWidget,  
    QStackedWidget  
2 from PyQt5 import uic  
3 from PyQt5 import QtWidgets  
4 from PyQt5.QtGui import QMovie  
5 import sys  
6 import QrScan  
7 import time  
8 import snap7  
9 import threading  
10  
11  
12  
13 class Snap7():  
14     IP = "192.168.137.100"  
15     RACK = 0  
16     SLOT = 1  
17  
18     DB_NUMBER = 1  
19  
20  
21     data = bytearray(b'\x00')  
22
```

```

23
24 plc = snap7.client.Client()
25 plc.connect(IP, RACK, SLOT)
26     #print(plc.get_cpu_info())
27 plc.db_write(1, 0, data)
28     #db = plc.db_read(DB_NUMBER. 0, 1)
29
30 def ReadDB():
31
32     db = Snap7.plc.db_read(1, 0, 1)
33     #value = int.from_bytes(db[0:1], byteorder = "big
34 ")
35     print (db)
36     #print (value)
37     #data = bytearray(b'\x00')
38     #Snap7.plc.db_write(1, 0, data)
39     #value = int.from_bytes(db[0:1], byteorder = "big
40 ")
41     #print (db)
42
43
44 class Data():
45     a = 0
46     b = 0
47     data = ""
48     prev = 0
49
50
51
52
53
54
55 class MainUi(QMainWindow):
56     def __init__(self):

```

```

57     super(MainUi, self).__init__()
58     uic.loadUi('app.ui', self)
59
60     self.movie = QMovie("box.gif")
61     self.label.setMovie(self.movie)
62     self.movie.start()
63
64     self.label_5.setText(str(Data.a))
65     self.label_6.setText(str(Data.b))
66
67     self.pushButton.clicked.connect(self.Start)
68
69     self.pushButton_2.clicked.connect(self.Stop)
70
71
72     #self.showMaximized()
73
74     #QrScan.Scanner.scan()
75
76     def Start(self):
77         #Snap7.ReadDB()
78         #ScanData()
79         data2 = bytearray(b'\x01')
80         Snap7.plc.db_write(1, 0, data2)
81         time.sleep(0.5)
82         data2 = bytearray(b'\x00')
83         Snap7.plc.db_write(1, 0, data2)
84
85
86
87     def Stop(self):
88         data1 = bytearray(b'\x02')
89         Snap7.plc.db_write(1, 0, data1)
90         time.sleep(0.5)
91         data1 = bytearray(b'\x00')
92         Snap7.plc.db_write(1, 0, data1)

```



```

93
94
95 def ScanData():
96     db = Snap7.plc.db_read(1, 1, 1)
97     value = int.from_bytes(db[0:1], byteorder = "big"
98 )
99     if value == 0:
100         Data.prev = 0
101     print(value)
102     if value == 1 and Data.prev ==0:
103         window.Stop()
104         Data.prev = 1
105         Data.data = QrScan.Scanner.scan()
106         print(Data.data)
107         if Data.data == "A":
108             Data.a = Data.a + 1
109             data2 = bytearray(b'\x05')
110             Snap7.plc.db_write(1, 0, data2)
111             time.sleep(0.5)
112             data1 = bytearray(b'\x00')
113             Snap7.plc.db_write(1, 0, data1)
114             #print(Data.a)
115         if Data.data == 'B':
116             Data.b +=1
117             window.Start()
118             window.label_5.setText(str(Data.a))
119             window.label_6.setText(str(Data.b))
120
121     threading.Timer(0.1, ScanData).start()
122
123 app = QtWidgets.QApplication(sys.argv)
124 window = MainUi()
125 window.show()
126 QrScan.Scanner.scan()
127 Snap7()

```

```

128 threading.Timer(0.1, ScanData).start()
129
130 #t1 = Thread(target = ScanData)
131 #t1.start()
132 #t1.join()
133
134
135 app.exec_()

```

Skripta za pokretanje skeniranja QR koda:

```

1 import cv2
2
3
4 class Scanner:
5
6     def scan():
7
8         close_qr_scanner = ""
9
10        # set up camera object
11        cap = cv2.VideoCapture(0)
12
13        # QR code detection object
14        detector = cv2.QRCodeDetector()
15
16        while True:
17            # get the image
18            _, img = cap.read()
19            # get bounding box coords and data
20            data, bbox, _ = detector.detectAndDecode(img)
21
22            # if there is a bounding box, draw one, along
23            with the data
24            if(bbox is not None):
25                for i in range(len(bbox)):
26                    cv2.line(img, tuple(bbox[i][0]),

```

```

26         tuple(bbox[(i+1) % len(bbox)][0]), color=(255,
27             0, 255), thickness=2)
28         cv2.putText(img, data, (int(bbox
29             [0][0][0]), int(bbox[0][0][1]) - 10), cv2.
30             FONT_HERSHEY_SIMPLEX,
31             0.5, (0, 255, 0), 2)
32         if data:
33             #print("data found: ", data)
34             close_qr_scanner = "1"
35             #print(data)
36             # display the image preview
37             #cv2.namedWindow("code detector")
38
39             cv2.namedWindow("code detector")
40             #cv2.resizeWindow("code detector", 2500,
41             2500)
42             #cv2.resizeWindow("code detector")
43
44             cv2.moveWindow("code detector", 1250, 300 )
45             cv2.imshow("code detector", img)
46
47             if(cv2.waitKey(1) == ord("q")):
48                 break
49             if(close_qr_scanner == "1"):
50                 break
51             # free camera object and exit
52             cap.release()
53             cv2.destroyAllWindows()
54             return data

```

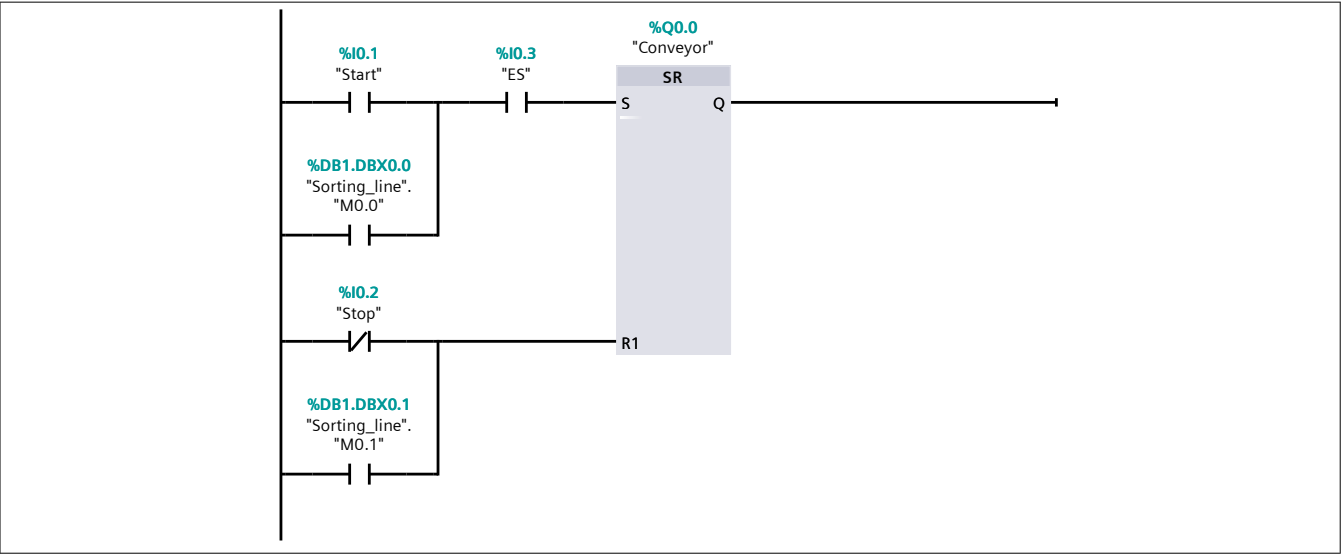
4.2. TIA portal program i podatkovni blokovi

Main [OB1]

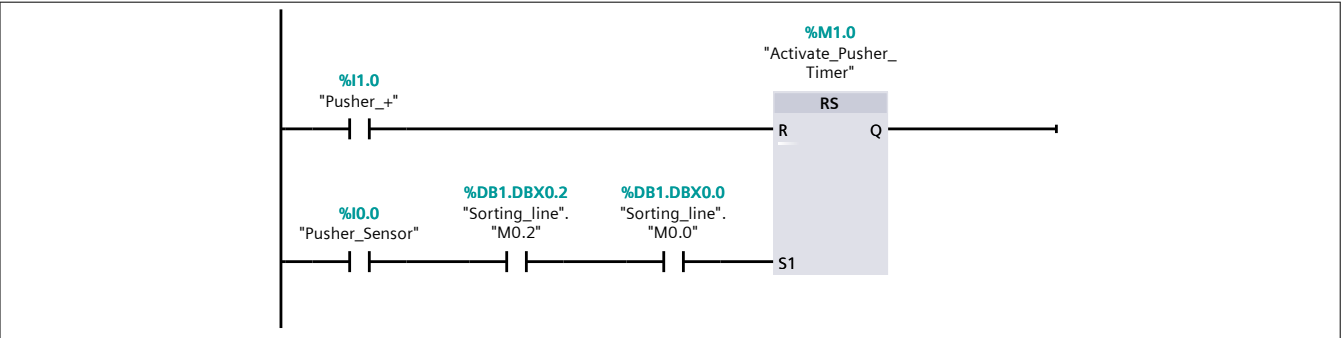
Main Properties					
General					
Name	Main	Number	1	Type	OB
Language	LAD	Numbering	Automatic		
Information					
Title	"Main Program Sweep (Cycle)"	Author		Comment	
Family		Version	0.1	User-defined ID	

Name	Data type	Default value	Comment
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

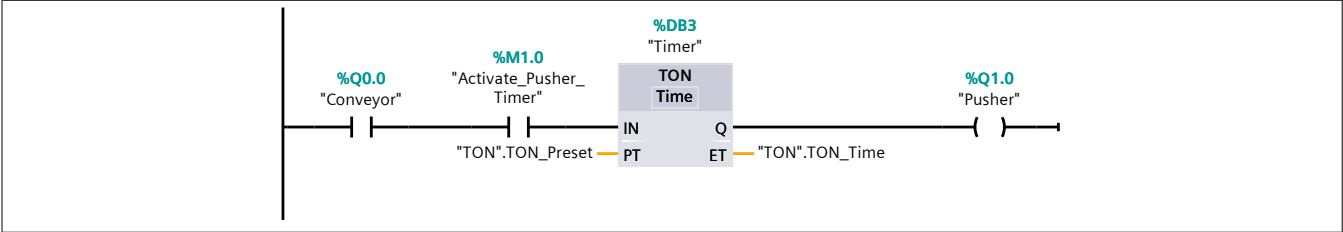
Network 1:



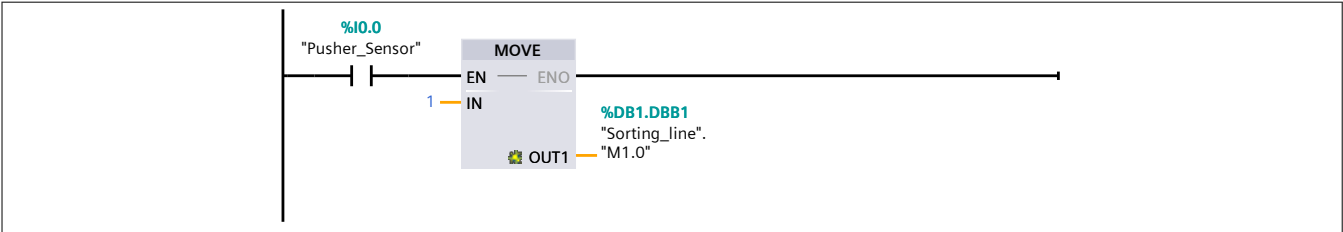
Network 2:



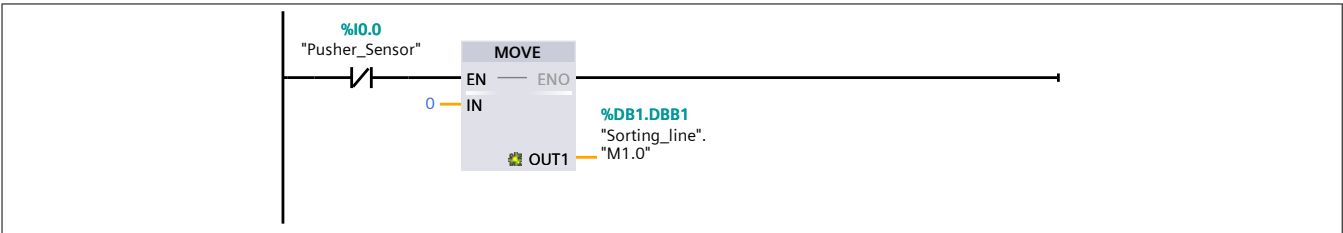
Network 3:



Network 4:



Network 5:



Totally Integrated Automation Portal

Sorting_line [DB1]

Sorting_line Properties

General

Name	Sorting_line	Number	1	Type	DB
Language	DB	Numbering	Automatic		

Information

Title		Author		Comment	
Family		Version	0.1	User-defined ID	

Name	Data type	Offset	Start value	Retain	Access- sible from HMI/O PC UA/W eb API	Wri- ta- ble from HMI eng- neer- ing	Visi- ble in HMI eng- neer- ing	Set- point	Super- vision	Comment
▼ Static										
M0.0	Bool	0.0	false	False	True	True	True	False		HMI_Start
M0.1	Bool	0.1	false	False	True	True	True	False		HMI_Stop
M0.2	Bool	0.2	false	False	True	True	True	False		Box_Type
M1.0	Byte	1.0	16#0	False	True	True	True	False		

Totally Integrated Automation Portal

TON [DB2]

TON Properties

General

Name	TON	Number	2	Type	DB
Language	DB	Numbering	Automatic		

Information

Title		Author		Comment	
Family		Version	0.1	User-defined ID	

Name	Data type	Start value	Retain	Access- sible from HMI/O PC UA/We b API	Wri- ta- ble fro m HM I/O PC UA/ We b API	Visible in HMI engi- neer- ing	Set- point	Super- vision	Comment
▼ Static									
TON_Start	Bool	false	False	True	True	True	False		
TON_Time	Time	T#0ms	False	True	True	True	False		
TON_Q	Bool	false	False	True	True	True	False		
TON_Preset	Time	T#1S_300ms	False	True	True	True	False		