



ŠOLSKI CENTER CELJE

Srednja šola za strojništvo, mehatroniko in medije

RAZISKOVALNA NALOGA

ULTIMATIVNI ROBOT

Avtorji:

Nik Čurin, M-2. f

Denis Fale, M-2. f

Miha Krumpačnik, M-2. f

Mentorja:

mag. Andro Glamnik, univ. dipl. inž.

Matej Veber, univ. dipl. inž.

Mestna občina Celje, Mladi za Celje

Celje, marec 2016

## **POVZETEK**

Za raziskovalno nalogo smo želeli izdelati mobilnega robota na kolesih, ki ga bomo vozili na daljinsko vodenje ter da bo v živo prenašal sliko na pametni telefon. Osnovo smo sestavili iz vseh potrebnih delov, ki smo si jih zamislili. Ker smo inovativni in želimo pridobiti novo znanje, smo se odločili, da bo našega robota poganjal mikroračunalnik Raspberry Pi 2 b+. Ker z njim ni imel izkušenj še nihče od nas, smo vedeli, da nas čaka trdo in težavno, vendar zabavno delo. Za vodenje samega robota smo razmišljali med dvema standardnima igralnima ploščkoma, in sicer med Xboxom ali Playstationom, ki delujeta na osnovi Bluetooth. Na koncu smo se odločili za Playstation igralni plošček. Ko smo dobili dele, smo pričeli s sestavljanjem robota, saj še nismo imeli ustreznega kabla za zagon Raspberryja. Ko smo platformo sestavili, smo vse moči usmerili v samo srce robota, in sicer v delo z Raspberry Pi. Čeprav so bile to za nas povsem nove dimenzije programiranja, smo se hitro naučili osnovne tehnike upravljanja. Ves čas pa smo se posvečali tudi samemu programu za vodenje in prenosu slike. Ko smo na Raspberry Pi namestili vso potrebno programsko opremo, smo pričeli s povezovanjem senzorjev in motorjev. Ker je v grobem vse delovalo pravilno, smo pričeli izpopolnjevati podrobnosti, dokler nismo prišli do zaključnega izdelka.

## Kazalo

HIPOTEZE .....	7
1) STROJNA OPREMA.....	2
1.1 Raspberry Pi 2 B+.....	2
1.2 MIŠKA .....	3
1.4 TIPKOVNICA.....	3
1.5 ZASLON .....	4
1.6 SPOMINSKA KARTICA .....	5
1.7 PIROCON v2.0 .....	5
1.8 ŠTEVILSKÉ OZNAKE NA KRMILNIKU PiRoCon 2.0 .....	6
1.9 Wi-Fi vmesnik .....	7
1.10 Bluetooth vmesnik.....	7
1.11 PS3-BLUETOOTH IGRALNI PLOŠČEK .....	8
2) PROGRAMSKA OPREMA.....	9
2.1 NAMESTITEV OPERACIJSKEGA SISTEMA RASPBIAN .....	9
2.2 POTEK NAMESTITVE .....	10
2.3 PROGRAM PYTHON .....	11
2.4 NAMESTITEV WI-FI VMESNIKA .....	12
2.5 NAMESTITEV EKRANA .....	13
2.6 NAMESTITEV BLUETOOTH VMESNIKA.....	14
2.7 NAMESTITEV KAMERE .....	17
2.8 NAMESTITEV PS3-BLUETOOTH VMESNIKA .....	20
2.8 KAMERA.....	21
3.2 OHIŠJE ZA KAMERO .....	21
3) MEHANSKI SKLOP .....	22
4.1 Pogonski sistem .....	22
4.2 Montažni plošči .....	22
4) ELEKTRONSKI SKLOP.....	23
5.1 IR-SENZORJA .....	23
4.2 LINIJSKA SENZORJA.....	24
4.3 ULTRAZVOČNI SENZOR.....	24
4.4 DC-MOTORJA.....	25
4.5 SERVOMOTORJI ZA NAGIBANJE IN VRTENJE .....	25
4.6 iBoost64.....	26
5) RASPBERRY PI Z OPREMO .....	26
5.1 PRIKLOP OPREME NA KRMILNIK.....	27

6) OSTALI DELI .....	27
6.1 DISTANČNIKI .....	27
6.2 VIJAKI .....	28
7) MONTAŽA ROBOTA .....	28
7.1 NAMEŠČANJE IR-SENZORJEV .....	29
7.2 MONTAŽA LINIJSKIH SENZORJEV .....	29
7.3 MONTAŽA POGONKEGA SISTEMA.....	30
7.4 MONTAŽA ULTRASONIČNEGA SENZORJA .....	30
7.5 MONTAŽA RASPBERRY PI Z OPREMO .....	31
7.6 MONTAŽA ZGORNJE PLOSKVE.....	31
7.7 MONTAŽA PAN/TILT-SISTEMA IN KAMERE .....	32
7.8 MONTAŽA POLNILNE BATERIJE .....	32
7.9 POSTOPKI.....	33
8) PROGRAM ROBOTA.....	35
8.1 KOMENTAR PROGRAMA .....	35
8.2 AVTOMATSKI ZAGON PROGRAMA .....	41
9) 3D-MODELIRANJE IN 3D-TISKANJE.....	42
9.1 MODELIRANJE V PROGRAMU SOLIDWORKS .....	42
9.2 3D-TISKANJE .....	42
10) KONČNI IZDELEK.....	43
10.1 KONČNI IZDELEK .....	43
11) UGOTOVITVE IN REZULTATI .....	44
12) ZAKLJUČEK.....	45
13) VIRI IN LITERATURA.....	46

Slika 1: Raspberry Pi 2 B+ .....	2
Slika 2: Miška.....	3
Slika 3: Tipkovnica .....	3
Slika 4: Priklop ekrana preko GPIO-pinov .....	4
Slika 5: Ekran, na katerega priklopimo Raspberry Pi .....	4
Slika 6: Mikro SD-spominska kartica .....	5
Slika 7 :Prikaz priklopa na Raspberry Pi.....	5
Slika 8: Prikaz številske oznake .....	6
Slika 9: Prikaz Wi-Fi dongle vmesnika.....	7
Slika 10: Prikaz Bluetooth dongle vmesnika.....	7
Slika 11: PS3-Bluetooth igralni plošček.....	8
Slika 12: Namestitev operacijskega sistema.....	9
Slika 13: Namizje .....	10
Slika 14: Primer programiranja v programu Python .....	11
Slika 15: Ukaz ifconfig.....	12
Slika 16: Prikazani IP-ji .....	12
Slika 17: Ukaz lsusb (nameščene naprave na USB-portih).....	12
Slika 18: Prenos datoteke .....	13
Slika 19: Namestitev datoteke .....	13
Slika 20: Uspešno nameščeno .....	13
Slika 21: Namestitev programske knjižice .....	13
Slika 22: Ukaz za namestitev vmesnika .....	14
Slika 23: Naslednji ukaz.....	14
Slika 24: Ukaz .....	15
Slika 25: Ukaz .....	15
Slika 26: Ukaz .....	15
Slika 27: Ukaz .....	15
Slika 29: Ukaz .....	16
Slika 30: Uspešno nameščen Bluetooth v pravilni vrstici .....	16
Slika 31: Prikaz posodobitve vse programske opreme.....	17
Slika 32 : Prikaz namestitve vseh posodobljenih datotek.....	17
Slika 33: V ukaznem oknu smo omogočili kamero.....	18
Slika 34: Prikaz, kako smo omogočili podporo kameri.....	18
Slika 35: Prikaz ukaza za ponovni zagon .....	19
Slika 36: Kamera.....	21
Slika 37: Ohišje za kamero.....	21
Slika 38: Številsko označevanje pogonskega sistema .....	22

Slika 39: Montažna plošča .....	22
Slika 40: Vijak za občutljivost .....	23
Slika 41: IR-senzor .....	23
Slika 42: Linijska senzorja .....	24
Slika 43: Ultrazvočni senzor .....	24
Slika 44: DC-motor .....	25
Slika 45: Sistem za nagib in vrtenje .....	25
Slika 46: iBoost64 .....	26
Slika 47: Raspberry Pi z opremo .....	26
Slika 48: Distančniki .....	27
Slika 49: Sestavljene ploskve z distančniki in vijaki.....	27
Slika 50: Uporabljeni vijaki .....	28
Slika 51: Nameščanje nosilca.....	29
Slika 52: Nameščen IR-senzor na nosilcu.....	29
Slika 53: Namestitev linijskih senzorjev .....	29
Slika 54: Namestitev pogonskega sistema .....	30
Slika 55: Ultrasonični senzor.....	30
Slika 56: Nameščena oprema .....	31
Slika 57: Nameščanje zgornje ploskve.....	31
Slika 58: Nameščanje PAN/TILT in kamere.....	32
Slika 59: Nosilec za polnilno postajo .....	33
Slika 60: Vrtanje lukenj na nosilcu .....	33
Slika 61: Vijachenje nosilca na platformo.....	33
Slika 62: Urejanje zadnjih malenkosti.....	34
Slika 63: Končni izdelek .....	34
Slika 64: Ukaz .....	41
Slika 65: Ukaz .....	41
Slika 66: Prikaz modeliranja ohišja za kamero v programu SolidWorks.....	42
Slika 67: Prikaz 3D-tiskanja ohišja za kamero.....	42
Slika 68: Končni izdelek .....	43
Slika 69: Končni izdelek .....	43

## UVOD

Za ime ultimativni avto smo se odločili zato, ker smo na tem področju želeli razviti nekaj novega. Želeli smo izdelati avto, ki ga bomo vodili brezžično preko Playstation 3 ploščka, ter da bo v živo prenašal sliko preko Wi-Fi vmesnika na pametni mobilni telefon. Pogon avtomobila deluje na vsa štiri kolesa, en motor za vsako stran. Uporabili smo linijske senzorje, IR-senzorje in servomotorja za obračanje kamere. Vse skupaj pa poganja zelo zmogljiv mikroračunalnik Raspberry Pi 2 B+.

## HIPOTEZE

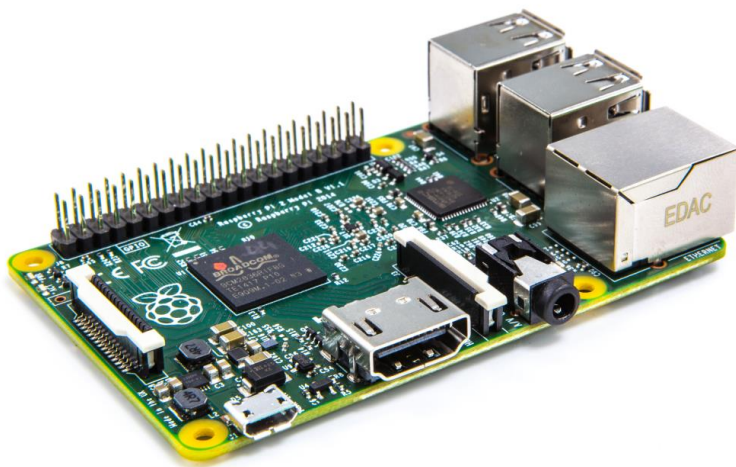
**V okviru raziskovalne naloge smo si zastavili sledeče hipoteze:**

- Izdelati bomo mobilnega robota z upravljanjem preko PS3 Bluetooth igralnega ploščka.
- Narediti prenos slike s kamere na telefon.
- Spoznati mikroračunalnik Raspberry Pi.
- Spoznati programsko okolje Python.
- Narediti estetsko ohišje.
- Ustvariti napajanje tudi s pomočjo fotocelic.
- Izdelati mobilnega avtomatiziranega robota.

# 1) STROJNA OPREMA

## 1.1 Raspberry Pi 2 B+

- Je mikroračunalnik , ki ima 1 GB delovnega pomnilnika.
- Ima 4 USB-vhode, HDMI-izhod, mikro USB-vhod, 3.5 mm avdio izhod, izhod za ekran, vhod za mrežo in kamero.
- Napajamo ga s 5 V.
- Preko GPIO PIN-ov računalnik komunicira z zunanjim svetom. Pini uporabljajo digitalni signal ter so lahko vhodi ali izhodi.
- 4-jedrni ARM Cortex-A7 procesor 900 MHz.



Slika 1: Raspberry Pi 2 B+



## 1.2 MIŠKA

Miška je eden glavnih delov strojne opreme, če sploh želimo delati z Raspberry Pi, saj z njo dostopamo do programov ter datotek v operacijskem sistemu Raspbiana. Za njeno delovanje ne potrebujemo dodatne programske opreme.



Slika 2: Miška

## 1.4 TIPKOVNICA

Je eden glavnih delov strojne opreme, če sploh želimo delati z Raspberry Pi, saj z njo vnašamo simbole v najrazličnejša okna. Za njeno delovanje ne potrebujemo dodatne programske opreme.



Slika 3: Tipkovnica

## 1.5 ZASLON

Monitor je eden glavnih delov strojne opreme, če sploh želimo delati z Raspberry Pi. Preko njega dostopamo do grafičnega vmesnika operacijskega sistema. Poznamo tri načine priklopa zaslona.

1. Priključitev na HDMI-izhod, pri čemer za njegovo delovanje ne potrebujemo dodatne programske opreme. Monitor mora podpirati digitalni oz. HDMI-signal, sicer potrebujemo pretvornik iz HDMI na VGA.
2. Priključitev preko GPIO-pinov in adapterja. Če se odločimo za to opcijo, potrebujemo dodatno programsko opremo, ki jo bomo podrobneje opisali v poglavju Programska oprema. Slabost tega načina je, da nam adapter zavzame veliko število GPIO-pinov, zato v našem primeru ne bi mogli priključiti še PiRoCon-a.
3. Priključitev na namenski izhod za ekran. Pri tej opciji prav tako potrebujemo dodatno programsko opremo. Slednjega postopka nismo prakticirali.



Slika 4: Priklop ekrana preko GPIO-pinov



Slika 5: Ekran, na katerega priklopimo Raspberry Pi

## 1.6 SPOMINSKA KARTICA

Spominska kartica je eden od glavnih delov strojne opreme, ki jo potrebujemo za delovanje Raspberry Pi. Na njo naložimo celoten operacijski sistem ter vse ostale podatke, ki jih imamo na Raspberry Pi. Njena kapaciteta mora biti vsaj 8 Gb. Naš model podpira mikro SD-kartice.



Slika 6: Mikro SD-spominska kartica

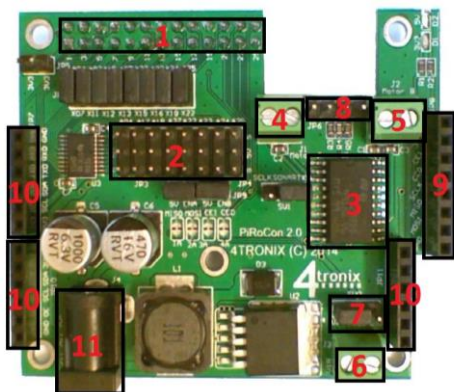
## 1.7 PIROCON v2.0

Pirocon v2.0 je krmilnik, s katerim Raspberry Pi lažje komunicira z zunanji komponentami. Za našega robota so ga izdelali pri podjetju 4tronix. Na njega lahko priključimo 8 digitalnih vhodnih ali izhodnih komponent, krmilimo lahko dva DC- motorja, ima poseben priključek za ultrazvočni senzor, 3 I2C-konektorje ter SPI-konektor. Pirocon ima veliko porabo električne energije. Če zanj nimamo dodatnega napajanja in ga priključimo na Raspberry Pi se le-ta izklopi in ponovno zažene zaradi prevelikega odvzema toka naenkrat. Proizvajalec je naredil tudi nekaj napak pri oznakah ter povezavah na samem vezju. Ena od pomanjkljivosti je tudi zaščita pred menjavo polov na H-mostiču. Zaradi tega smo H-mostič tudi uničili in smo ga bili prisiljeni nadomestiti z novim. To nam je vzelo precej časa, saj čipov še nikoli nismo menjali ter smo se vse veščine menjavanja morali naučiti.



Slika 7 :Prikaz priklopa na Raspberry Pi

## 1.8 ŠTEVILSKÉ OZNAKE NA KRMILNIKU PiRoCon 2.0



Slika 8: Prikaz številske oznake

### Številске oznake s slike

1. GPIO-konektor za priklop na Rpi
2. 8-vhodno-izhodnih pinov z GND in 5 V
3. H-mostič za krmiljenje motorjev
4. Motor A
5. Motor B
6. Priključek za ločeno napajanje motorjev (5-15 V DC)
7. Izbira zunanjega napajana oziroma napajanje z Raspberry Pi za motorje
8. 4-pinski priključek za ultrazvočni senzor
9. SPI-konektor
10. I2C-konektorji
11. Priključek za dodatno napajanje Pirocona v2.0

## 1.9 Wi-Fi vmesnik

Wi-Fi vmesnik nam omogoča povezovanje na Raspberry Pi z Wi-Fi omrežij oziroma komunikacijo prek Wi-Fi signala. Pri našem projektu smo uporabili USB Wi-Fi dongle vmesnik, ki deluje na standardni frekvenci 50 Mhz ter valovni dolžini 802.11 n. Če želimo, da vmesnik deluje pravilno, moramo na mikroračunalnik naložiti ustrezno programsko opremo, kar bomo podrobneje opisali v poglavju Programska oprema.



Slika 9: Prikaz Wi-Fi dongle vmesnika

## 1.10 Bluetooth vmesnik

Bluetooth vmesnik nam omogoča povezovanje Rpi z ostalimi napravami preko tehnologije Bluetooth. Pri našem projektu smo uporabili standardni USB-Bluetooth dongle vmesnik, ki podpira Bluetooth 3.0. Če želimo, da vmesnik deluje pravilno, moramo imeti nameščeno ustrezno različico programske opreme, kar bomo podrobneje opisali v poglavju Programska oprema. Z Bluetooth vmesnikom nismo imeli nikakršnih težav.



Slika 10: Prikaz Bluetooth dongle vmesnika

### 1.11 PS3-BLUEETOOTH IGRALNI PLOŠČEK

Odločili smo se, da bomo za vodenje robota uporabili Sonyjev igralni plošček Sonyjeve igralne konzole Playstation 3. Z njim je upravljanje precej preprosto, obenem pa nam ponuja veliko število (12) gumbov in kar 10 različnih PWM-osi. Za njegovo pravilno delovanje smo potrebovali dodatno programsko opremo, ki jo bomo podrobneje opisali v poglavju Programska oprema.



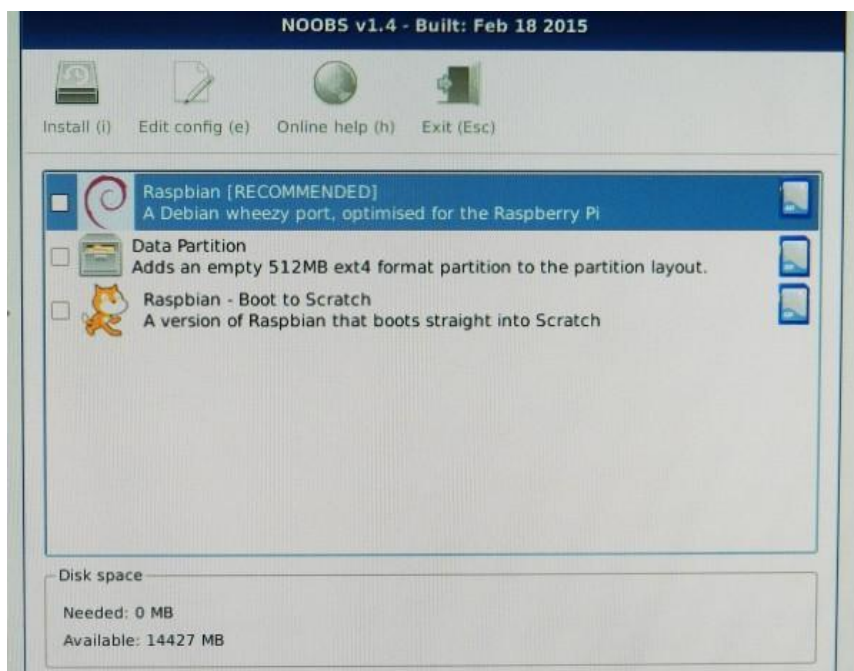
Slika 11: PS3-Bluetooth igralni plošček

## 2) PROGRAMSKA OPREMA

### 2.1 NAMESTITEV OPERACIJSKEGA SISTEMA RASPBIAN

Raspbian je eden od najpogostejših operacijskih sistemov za Raspberry Pi. Temelji na osnovah Linuxa, zato je njuna uporaba precej podobna. Raspbian operacijski sistem najprej z uradne strani Raspberry Pi shranimo na osebni računalnik. Nato vse datoteke, ki jih shranjena datoteka vsebuje, preprosto prenesemo na pomnilniško kartico. Ko pomnilniško kartico vstavimo v Raspberry Pi ter ga priključimo na vse potrebne komponente (miška, ekran, tipkovnica), se nam prikaže pričetek inštalacijskega postopka operacijskega sistema Raspbian.

Raspbian je potrebno vsakič, ko ga zaženemo, posodobiti.



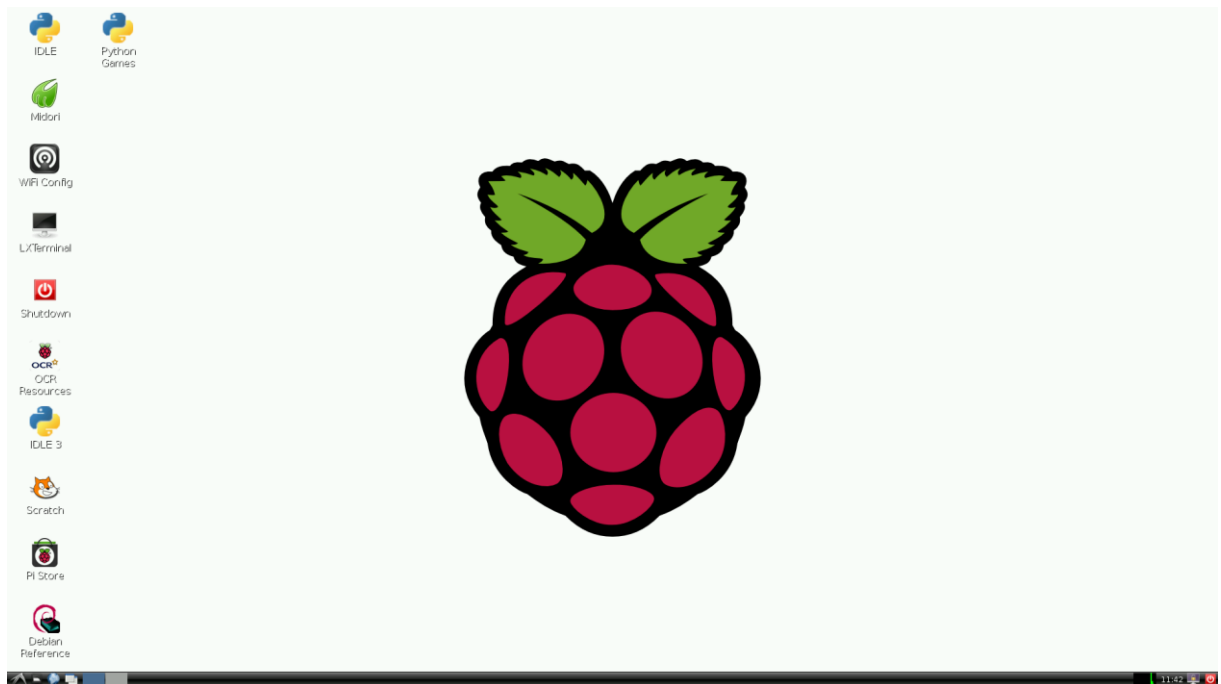
Slika 12: Namestitev operacijskega sistema

## 2.2 POTEK NAMESTITVE

Obkljukamo prvo opcijo ter kliknemo install. Vse ostalo se izvede avtomatsko, mi le počakamo, da se namestitveni postopek zaključi.

Po zaključeni namestitvi se na zaslonu pokaže namizje, na katerem je koš ter opravilna vrstica.

Če želimo na namizju dodatne ikone, jih moramo prenesti iz menija.

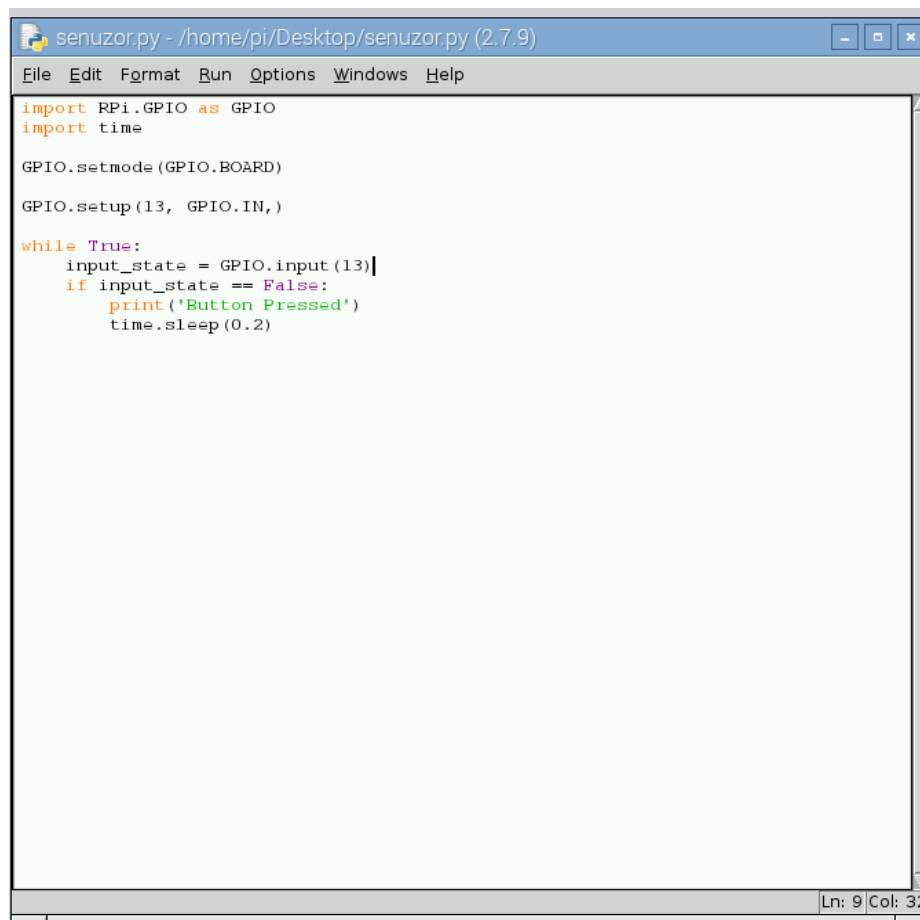


Slika 13: Namizje



## 2.3 PROGRAM PYTHON

Python je programsko okolje za pisanje uporabniških programov. V njem smo v celoti napisali program za našega robota in je že vnaprej nameščen na operacijskem sistemu, zato ga ni potrebno dodatno nalagati. Spada med naprednejša programska okolja v razredu Python.



```
senuzor.py - /home/pi/Desktop/senuzor.py (2.7.9)
File Edit Format Run Options Windows Help

import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)

GPIO.setup(13, GPIO.IN,)

while True:
    input_state = GPIO.input(13)
    if input_state == False:
        print('Button Pressed')
        time.sleep(0.2)

Ln: 9 Col: 32
```

Slika 14: Primer programiranja v programu Python

## 2.4 NAMESTITEV WI-FI VMESNIKA

```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:5c:5f:0b
          inet addr:169.254.189.61  Bcast:169.254.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:252 errors:0 dropped:0 overruns:0 frame:0
          TX packets:118 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:37506 (36.6 KiB)  TX bytes:23024 (22.4 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:68 errors:0 dropped:0 overruns:0 frame:0
          TX packets:68 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6484 (6.3 KiB)  TX bytes:6484 (6.3 KiB)

wlan0     Link encap:Ethernet  HWaddr 00:17:7c:22:81:f1
          inet addr:192.168.1.123  Bcast:255.255.255.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

pi@raspberrypi ~ $
```

Slika 15: Ukaz ifconfig

```
wlan0     Link encap:Ethernet  HWaddr 00:17:7c:22:81:f1
          inet addr:192.168.2.10  Bcast:192.168.2.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:222 errors:0 dropped:6 overruns:0 frame:0
          TX packets:81 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:27392 (26.7 KiB)  TX bytes:9901 (9.6 KiB)

pi@raspberrypi ~ $
```

Slika 16: Prikazani IP-ji

```
pi@raspberrypi ~ $ lsusb
Bus 001 Device 002: ID 0424:9512 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 004: ID 05e3:0608 Genesys Logic, Inc. USB-2.0 4-Port HUB
Bus 001 Device 005: ID 148f:5370 Ralink Technology, Corp. RT5370 Wireless Adapter
Bus 001 Device 006: ID 046d:c404 Logitech, Inc. TrackMan Wheel
Bus 001 Device 007: ID 045e:00dd Microsoft Corp. Comfort Curve Keyboard 2000 V1.0
```

Slika 17: Ukaz lsusb (nameščene naprave na USB-portih)

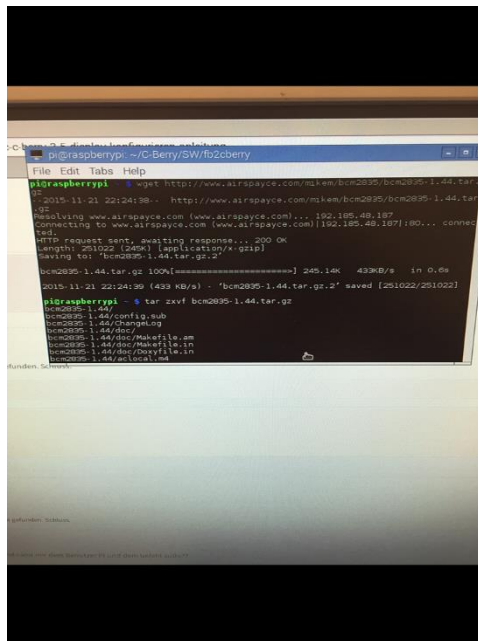
**Za namestitev Wi-Fi vmesnika smo uporabili naslednje ukaze v LX-Terminalu:**

- sudo apt-get update
- sudo apt-get upgrade
- ifconfig
- lsusb

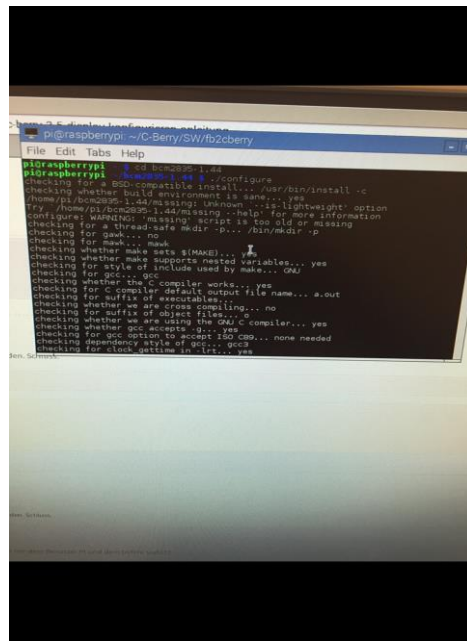
Nato se nam je v opravilni vrstici prikazal terminal za brezžično povezavo in nato smo se lahko povezali na omrežje, ki nam bo v nadaljevanju služilo za prenos slike pri kameri.

## 2.5 NAMESTITEV EKRANA

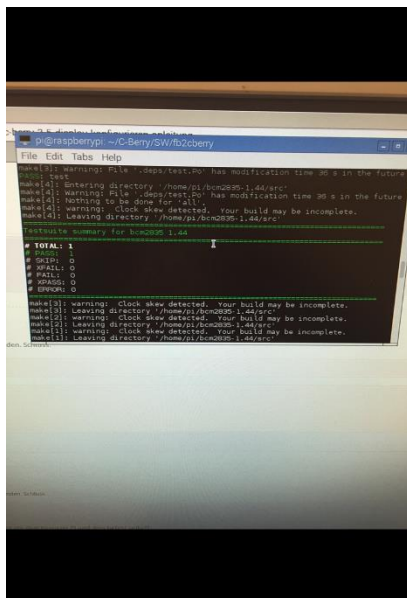
## Inštalacijski postopek za Display AdmaTec C-Berry 3.5"



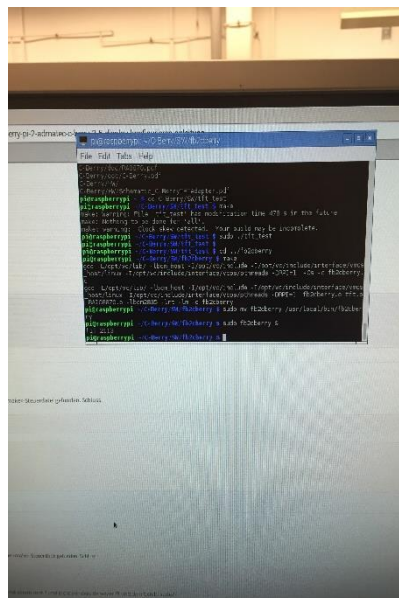
Slika 18: Prenos datoteke



Slika 19: Namestitev datoteke



Slika 20: Uspešno nameščeno

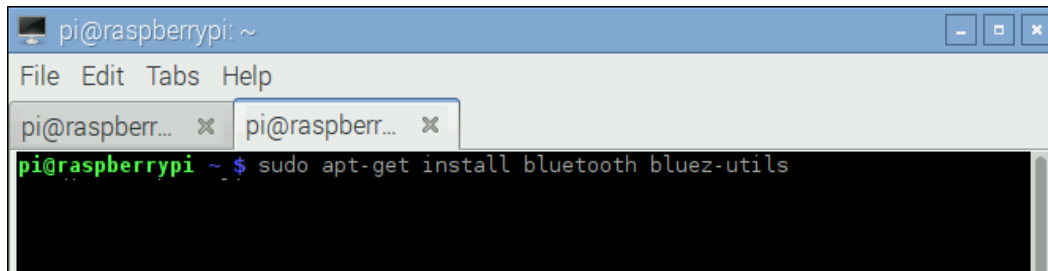


Slika 21: Namestitev programske knjižice

## 2.6 NAMESTITEV BLUETOOTH VMESNIKA

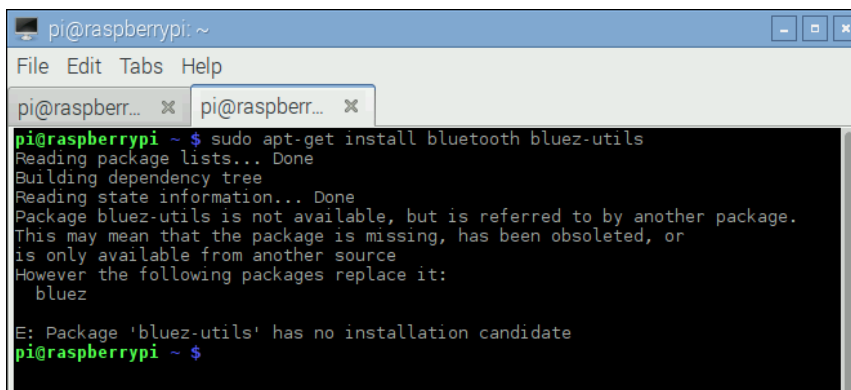
To smo storili v LX-Terminalu na sledeči način:

- Pri namestitvi Bluetootha smo uporabil naslednji postopek:
- Najprej smo v ukazno vrstico v terminalu vnesli naslednji ukaz:  
`sudo apt-get install bluetooth bluez-utils.`



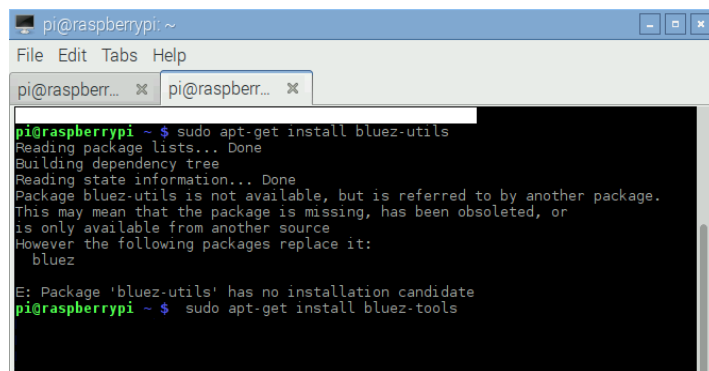
Slika 22: Ukaz za namestitev vmesnika

Po nekaj sekundah smo na terminalu dobili naslednji ukaz:



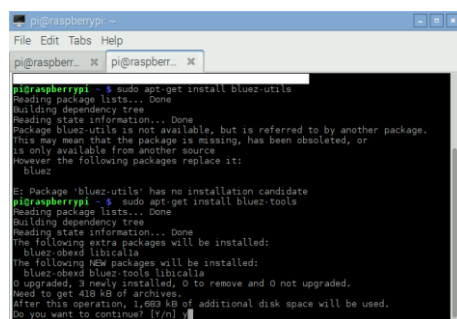
Slika 23: Naslednji ukaz

## Naslednji ukaz:



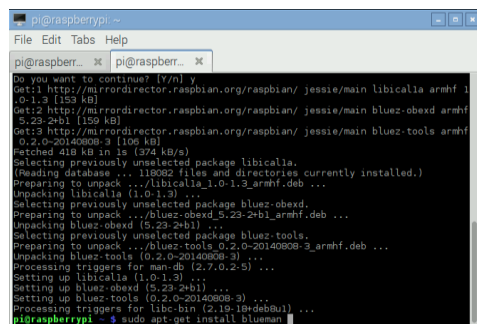
```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberr... x pi@raspberr... x  
pi@raspberrypi ~$ sudo apt-get install bluez-utils  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Package bluez-utils is not available, but is referred to by another package.  
This may mean that the package is missing, has been obsoleted, or  
is only available from another source  
However the following packages replace it:  
  bluez  
E: Package 'bluez-utils' has no installation candidate  
pi@raspberrypi ~$ sudo apt-get install bluez-tools
```

## Slika 24: Ukaz



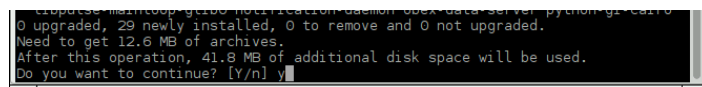
```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberr... x pi@raspberr... x  
pi@raspberrypi ~$ sudo apt-get install bluez-utils  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Package bluez-utils is not available, but is referred to by another package.  
This may mean that the package is missing, has been obsoleted, or  
is only available from another source  
However the following packages replace it:  
  bluez  
E: Package 'bluez-utils' has no installation candidate  
pi@raspberrypi ~$ sudo apt-get install bluez-tools  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  bluez-obsd libical  
The following NEW packages will be installed:  
  bluez-obsd bluez-tools libical  
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.  
Need to get 418 kB of archives.  
After this operation, 1,683 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y
```

## Slika 25: Ukaz



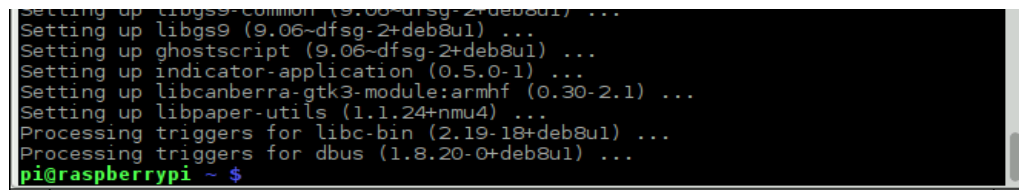
```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberr... x pi@raspberr... x  
pi@raspberrypi ~$ sudo apt-get install bluez-tools  
Do you want to continue? [Y/n] y  
Get:1 http://mirrordirector.raspbian.org/raspbian/ jessie/main libicalia armhf 1.0-1.3 [153 kB]  
Get:2 http://mirrordirector.raspbian.org/raspbian/ jessie/main bluez-obsd armhf 5.23-2+b1 [159 kB]  
Get:3 http://mirrordirector.raspbian.org/raspbian/ jessie/main bluez-tools armhf 0.2.0-20140808-3 [106 kB]  
Fetched 418 kB in 1s (374 kB/s)  
Selecting previously unselected package libicalia.  
(Reading database ... 118982 files and directories currently installed.)  
Preparing to unpack .../libicalia_1.0-1.3_armhf.deb ...  
Unpacking libicalia (1.0-1.3) ...  
Selecting previously unselected package bluez-obsd.  
Preparing to unpack .../bluez-obsd_5.23-2+b1_armhf.deb ...  
Unpacking bluez-obsd (5.23-2+b1) ...  
Selecting previously unselected package bluez-tools.  
Preparing to unpack .../bluez-tools_0.2.0-20140808-3_armhf.deb ...  
Unpacking bluez-tools (0.2.0-20140808-3) ...  
Processing triggers for man-db (2.7.0.2-5) ...  
Setting up libicalia (1.0-1.3) ...  
Setting up bluez-obsd (5.23-2+b1) ...  
Setting up bluez-tools (0.2.0-20140808-3) ...  
Processing triggers for libc-bin (2.19-18+deb8u1) ...  
pi@raspberrypi ~$ sudo apt-get install bluez-tools
```

## Slika 26: Ukaz



```
0 upgraded, 29 newly installed, 0 to remove and 0 not upgraded.  
Need to get 12.6 MB of archives.  
After this operation, 41.8 MB of additional disk space will be used.  
Do you want to continue? [Y/n] y
```

## Slika 27: Ukaz



```
Setting up libgs9-common (9.06-0fsg-2+deb8u1) ...  
Setting up libgs9 (9.06-0fsg-2+deb8u1) ...  
Setting up ghostscript (9.06-0fsg-2+deb8u1) ...  
Setting up indicator-application (0.5.0-1) ...  
Setting up libcanberra-gtk3-module:armhf (0.30-2.1) ...  
Setting up libpaper-utils (1.1.24+nmu4) ...  
Processing triggers for libc-bin (2.19-18+deb8u1) ...  
Processing triggers for dbus (1.8.20-0+deb8u1) ...  
pi@raspberrypi ~$
```

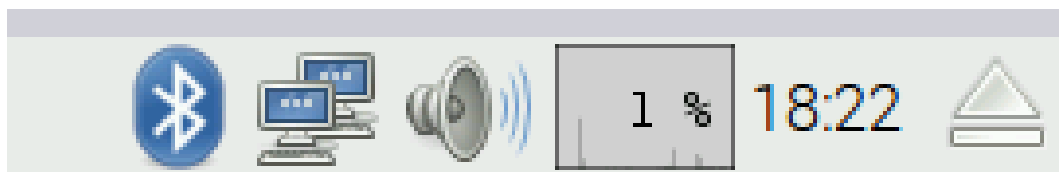
## Slika 28: Ukaz

Naslednji ukaz startx, ki zažene Bluetooth.

```
Setting up libc-bin (2.19-18+deb8u1) ...  
Setting up libpaper-utils (1.1.24+nmu4) ...  
Processing triggers for libc-bin (2.19-18+deb8u1) ...  
Processing triggers for dbus (1.8.20-0+deb8u1) ...  
pi@raspberrypi ~ $ startx
```

Slika 29: Ukaz

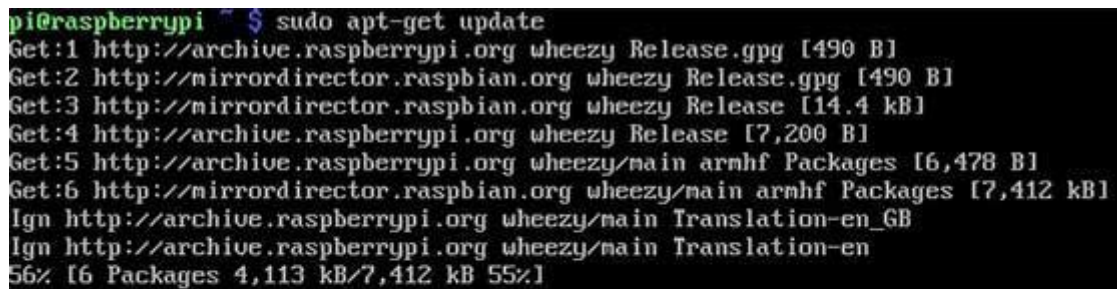
V ukazni vrstici se pojavi naslednja ikona za Bluetooth:



Slika 30: Uspešno nameščen Bluetooth v opravilni vrstici

## 2.7 NAMESTITEV KAMERE

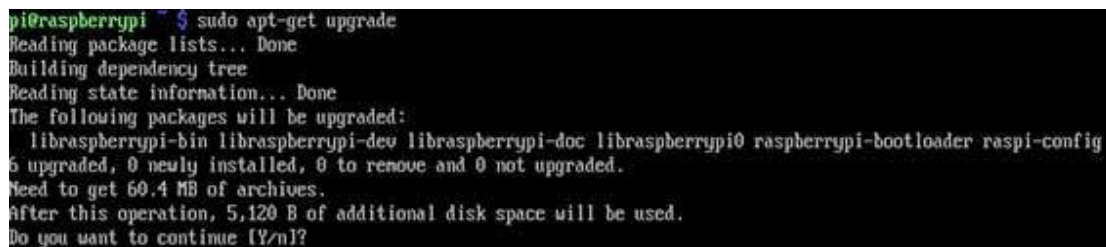
### 1) Ukaz: sudo apt-get update



```
pi@raspberrypi ~ $ sudo apt-get update
Get:1 http://archive.raspberrypi.org wheezy Release.gpg [490 B]
Get:2 http://mirrordirector.raspbian.org wheezy Release.gpg [490 B]
Get:3 http://mirrordirector.raspbian.org wheezy Release [14.4 kB]
Get:4 http://archive.raspberrypi.org wheezy Release [7,200 B]
Get:5 http://archive.raspberrypi.org wheezy/main armhf Packages [6,478 B]
Get:6 http://mirrordirector.raspbian.org wheezy/main armhf Packages [7,412 kB]
Ign http://archive.raspberrypi.org wheezy/main Translation-en_GB
Ign http://archive.raspberrypi.org wheezy/main Translation-en
56% [6 Packages 4,113 kB/7,412 kB 55%]
```

Slika 31: Prikaz posodobitve vse programske opreme

### 2) Ukaz: Sudo apt-get upgrade



```
pi@raspberrypi ~ $ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  libraspberrypi-bin libraspberrypi-dev libraspberrypi-doc libraspberrypi0 raspberrypi-bootloader raspi-config
6 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 60.4 MB of archives.
After this operation, 5,120 B of additional disk space will be used.
Do you want to continue [Y/n]?
```

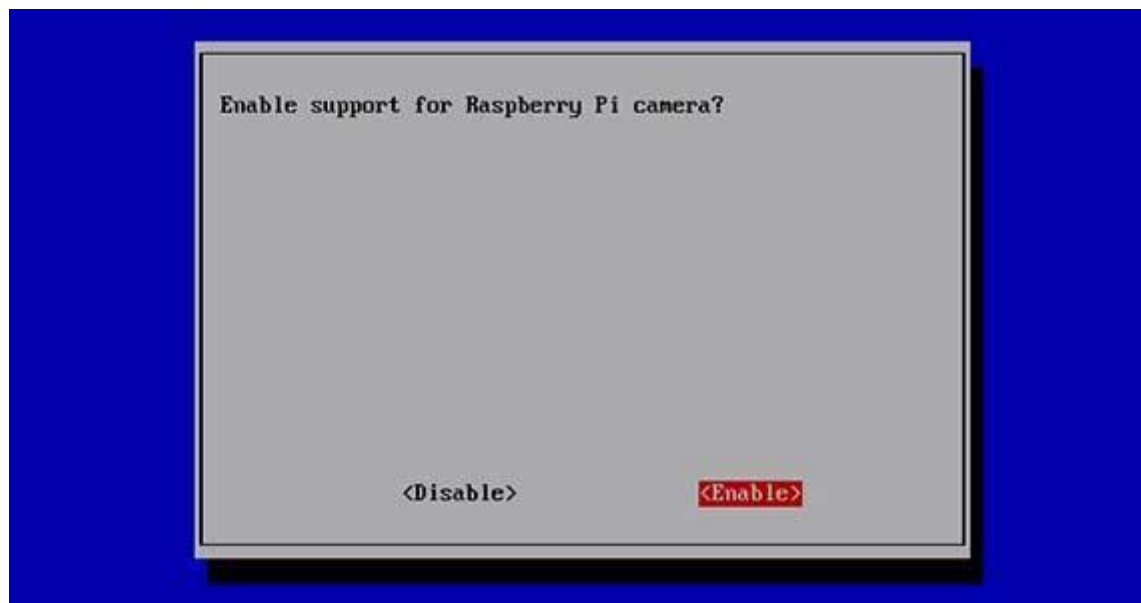
Slika 32 : Prikaz namestitve vseh posodobljenih datotek.

### 3) Ukaz: V terminal smo vnesli raspi-config:



Slika 33: V ukaznem oknu smo omogočili kamero.

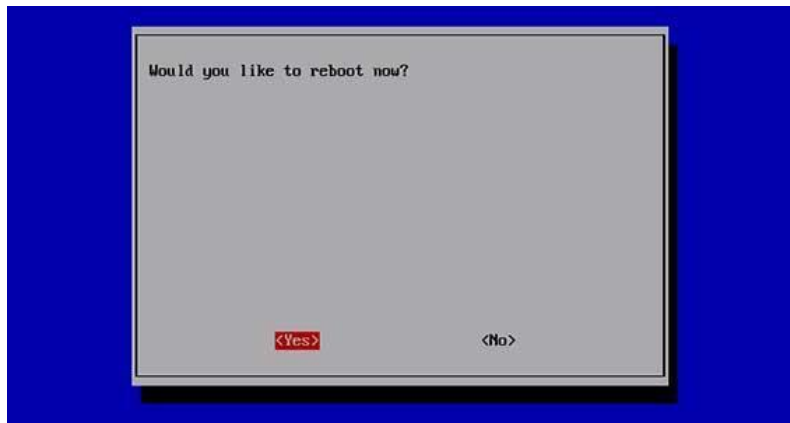
### 4) Ukaz: Enable



Slika 34: Prikaz, kako smo omogočili podporo kameri.



## 5) Ukaz: Yes – potrditev ponovnega zagona



Slika 35: Prikaz ukaza za ponovni zagon

### Ukazi v LX-Terminalu za zagon kamere po inštalacijskem postopku:

- Raspistill -o image.jpg - naredi sliko.
- Raspivid -o video.h264- naredi 5 s video.

Ker smo hoteli sestaviti avto, ki bo povsem avtomatiziran, brez zaganjanja vsakega posameznega programa, smo morali napisati program za zagon kamere v programu Python.

To smo storili na sledeč način:

- V Lx-Terminal smo vnesli naslednji ukaz `sudo apt-get install python camera`.
- Nato smo v programu Python napisali naslednji ukaz, ki je zagnal kamero:
- `import picamera`
- `from time import sleep`
- `camera = picamera.PiCamera ()`
- `camera.capture ('image1.jpg')`
- `sleep (5)`
- `camera.capture ('image2.jpg')`

## 2.8 NAMESTITEV PS3-BLUEETOOTH VMESNIKA

Namestitev smo izvedli v LX terminalu.

### 1) Namestitev potrebnih modulov za Bluetooth

```
sudo apt-get install bluez-utils bluez-compatible bluez-hcidump  
checkinstall libusb-dev libbluetooth-dev joystick
```

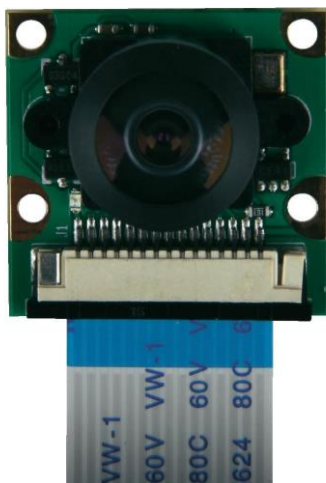
### 2) Namestitev potrebnih modulov za igralni plošček

```
sudo apt-get update  
sudo apt-get install pyqt4-dev-tools  
wget http://www.pabr.org/sixlinux/sixpair.c  
gcc -o sixpair sixpair.c -lusb
```

### 3) Namestitev aplikacije za igralni plošček

```
wget http://sourceforge.net/projects/qtsixa/files/QtSixA%201.5.1/QtSixA-1.5.1-  
src.tar.gz  
tar xfvz QtSixA-1.5.1-src.tar.gz  
cd QtSixA-1.5.1/sixad  
make  
sudo mkdir -p /var/lib/sixad/profiles  
sudo apt-get install checkinstall  
sudo checkinstall
```

## 2.8 KAMERA



Slika 36: Kamera

- To je kamera, ki ima resolucijo 1920 x 1080.
- Služi za zajemanje posnetkov in fotografij.
- Ima možnost razširitve za nočni vid.

## 3.2 OHIŠJE ZA KAMERO

Ohišje smo narisali v programu SolidWorks ter ga natisnili s pomočjo 3D-tiskalnika.

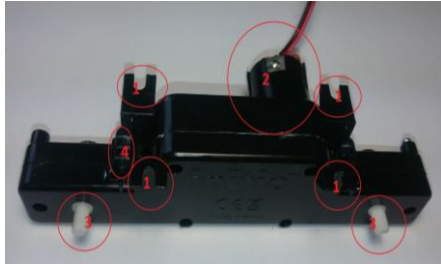


Slika 37: Ohišje za kamero

### 3) MEHANSKI SKLOP

#### 4.1 Pogonski sistem

Pogonski sistem je glavni del robota, če želimo, da se le-ta premika. Na robotu imamo dva pogonska sistema, in sicer za vsako stran enega. V njem najdemo zobniške sisteme.

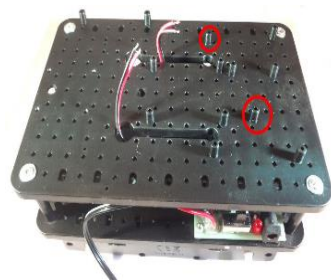


Slika 38: Številsko označevanje pogonskega sistema

1. Nosilci za pritrnitev na prvo nosilno ploščo
2. Nosilec za motor
3. Osi za pritrnitev koles
4. Mehansko stikalo za vklop oz. izklop prenosa na kolo

#### 4.2 Montažni plošči

Montažna plošča nam služi, da nanjo pritrdimos vse potrebne komponente, ki jih želimo imeti, v našem primeru smo uporabili 2 montažni plošči. Kot vezne elemente pa smo uporabili čepe ter vijake.



Slika 39: Montažna plošča

## 4) ELEKTRONSKI SKLOP

### 5.1 IR-SENZORJA

Infrardeča senzorja uporabljamo za zaznavanje ovir pred samim robotom. Imata tri kontakte, in sicer VIN, GND in signal. Če pred senzorjem ni ovire, je na izhodu signal logična 1. Ko pa predenj postavimo predmet, na njem zasveti lučka in signal se postavi na logično 0. Na zadnji strani imata tudi vijak, s katerim nastavimo razdaljo, na kateri senzor zazna oviro. Čez večji del zunanjšega ohišja imata navoj, na katerega privijamo namensko vijaka.



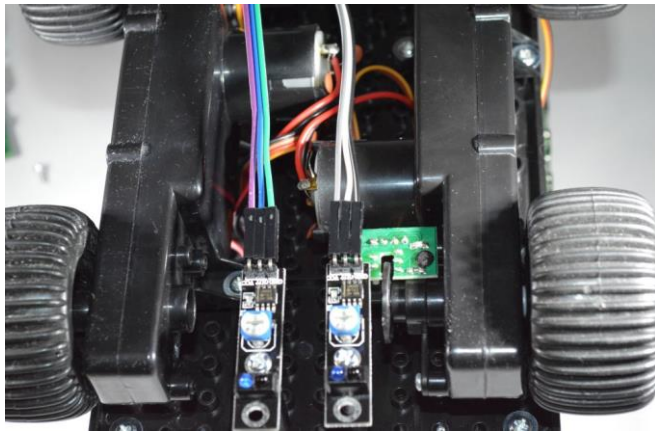
Slika 40: Vijak za občutljivost



Slika 41: IR-senzor

## 4.2 LINIJSKA SENZORJA

Pri robotu smo na spodnjo ploščo namestili tudi linijska senzorja.



Slika 42: Linijska senzorja

## 4.3 ULTRAZVOČNI SENZOR

Modul ima štiri priključke: VCC (+5 V), Triger (prožilnik), Echo (odmev) in GND (masa). Zelo enostavno ga je priključiti na katerikoli mikrokontroler. Za proženje moramo dovesti pozitivni pulz, dolžine 10  $\mu$ s ali več. Ta sproži serijo osmih 40 kHz kratkih zvokov na oddajniku (T), katerih odboj od predmeta (njihovo povprečje) sprejemnik (R) sprejme in poda kot pozitivni pulz na nožici Echo. Dolžina tega pulza je odvisna od oddaljenosti predmeta.



Slika 43: Ultrazvočni senzor

## 4.4 DC-MOTORJA

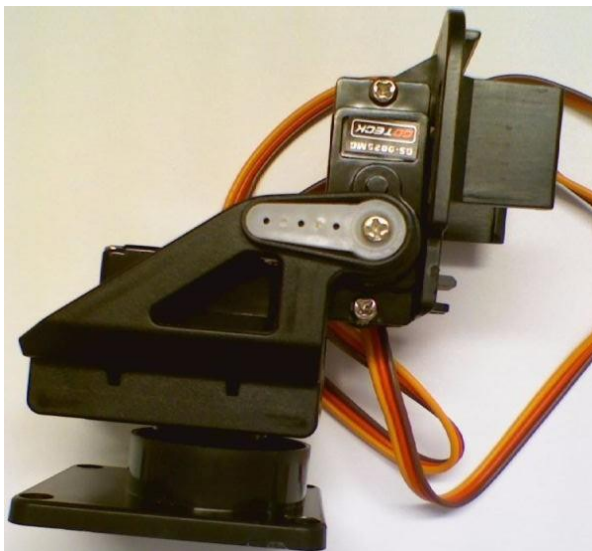
DC-motorja uporabljamo za pogon pogonskega menjalnika ter posredno pogon koles. Na zunaj je motor enosmeren, saj je enosmerna tudi napetost, s katero ga napajamo. Takšna konstrukcija v primerjavi z enosmernimi motorji s ščetkami pomeni boljše odvajanje toplote (hlajenje) in manjši vztrajnostni moment motorja. Pole rotorja s permanentnimi magneti (rdeča barva predstavlja severni, zelena pa južni magnetni pol) privlačijo nasprotni poli statorja, kar povzroči navor.



Slika 44: DC-motor

## 4.5 SERVOMOTORJI ZA NAGIBANJE IN VRTENJE

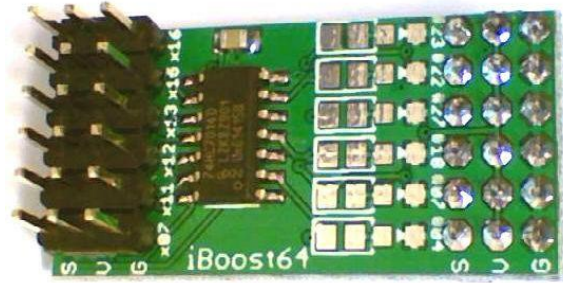
To je sistem, v katerem sta nameščena dva servomotorja. Privijamo ga na montažno ploskev, nato pa nanj lahko namestimo kamero ali ultrazvočni senzor.



Slika 45: Sistem za nagib in vrtenje

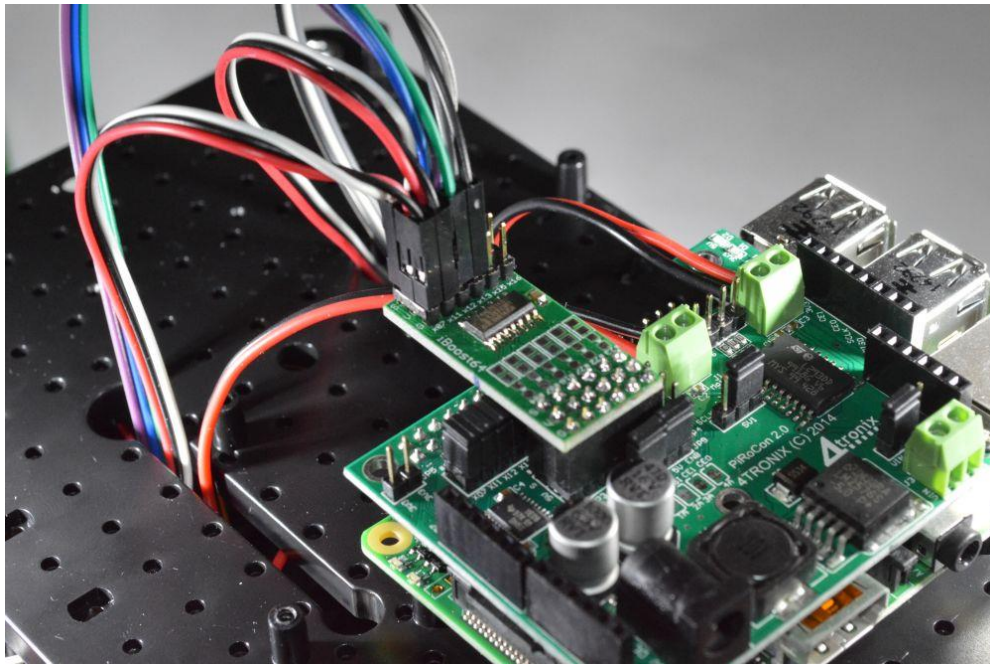
## 4.6 iBoost64

iBoost je del, ki ga namestimo na krmilnik PiRoCon 2.0, služi za povečanje vhodnih signalov in za električno čiščenje signalov.



Slika 46: iBoost64

## 5) RASPBERRY PI Z OPREMO



Slika 47: Raspberry Pi z opremo

- 1) Ko smo napisali ustrezne programe, smo morali elemente tudi priklopiti na ustrezne pine, ki smo jih določili.



## 5.1 PRIKLOP OPREME NA KRMILNIK

- levi IR-senzor na pin 7
- desni IR-senzor na pin 11
- levi linijski senzor na pin 12
- desni linijski senzor na pin 13
- servomotor za vrtenje levo, desno na pin 22
- servomotor za vrtenje gor, dol na pin 18

## 6) OSTALI DELI

### 6.1 DISTANČNIKI

Distančnike smo uporabili, da smo lahko sestavili montažne ploskve. Najprej smo jih sestavili in na koncu z vijaki tudi privijali.



Slika 48: Distančniki



Slika 49: Sestavljene ploskve z distančniki in vijaki

## 6.2 VIJAKI

Vijake smo uporabili za vijačenje vseh elementov, ki smo jih imeli. Privijali smo sestavne dele, montažno ploskev, Raspberry Pi, linijske senzorje, nosilce za IR-senzorje ...



Slika 50: Uporabljeni vijaki

## 7) MONTAŽA ROBOTA

Ko smo dobili potrebne dele, ki smo si jih zamislili, smo morali robota sestaviti. Vendar smo kaj kmalu ugotovili, da bi želeli določene stvari spremeniti in bolj dodelati. Zato smo pričeli postavljati dele po našem načrtu. Tako smo kot primer premaknili Raspberry Pi z opremo na spodnjo ploščo, saj smo na zgornjo pritrdili fotocelice za napajanje samega robota. S premikanjem komponent smo naleteli tudi na nekaj težav, ki smo jih morali rešiti. Ena od teh težav je bila, da so bili konektorji za senzorje previsoko postavljeni in je na njih pritiskala zgornja plošča. Odločili smo se, da bomo malce znižali nosilce, na katere je pritrjen Raspberry Pi. S tem smo pridobili 3 mm višine, kar se je nekoliko poznalo, vendar ne dovolj. Odločili smo se, da bomo podaljšali nosilce za zgornjo ploščo, kar je prineslo dodatnih 5 mm in zato je na konektorje nehala pritiskati zgornja plošča. Naslednji problem, s katerim smo se soočili, so bili nosilci IR-senzorjev, saj so imeli premajhno luknjo. Ta problem smo rešili tako, da smo vzeli večji sveder in luknjo preprosto povrtali. IR-senzorje smo pritrdili s pomočjo namenskih nosilcev. Namenski nosilci imajo na spodnji strani jeziček z luknjo, tako da ga lahko privijamo na osnovno ploščo. Na delovni površini ima nosilec večjo luknjo, skozi katero potisnemo sam senzor in ga z namenskimi vijaki privijamo z obeh strani.

## 7.1 NAMEŠČANJE IR-SENZORJEV

Za namestitev nosilca za IR-senzor smo uporabili vijaka 3 x 8 PB, za vsak senzor enega. Namestili smo jih na spodnjo ploskev samega robota. Služili nam bodo za merjenje razdalje in da se robot ne bo zaletel v kakšno oviro.



Slika 51: Nameščanje nosilca



Slika 52: Nameščen IR-senzor na nosilcu

## 7.2 MONTAŽA LINIJSKIH SENZORJEV

Linijske senzorje smo pritrdili na spodnjo ploskev robota, uporabili pa smo dva vijaka 3 x 8 PB, za vsak senzor enega. Senzorja imata timer, na katerem lahko nastavimo občutljivost zaznavanja.



Slika 53: Namestitev linijskih senzorjev

### 7.3 MONTAŽA POGONSKEGA SISTEMA

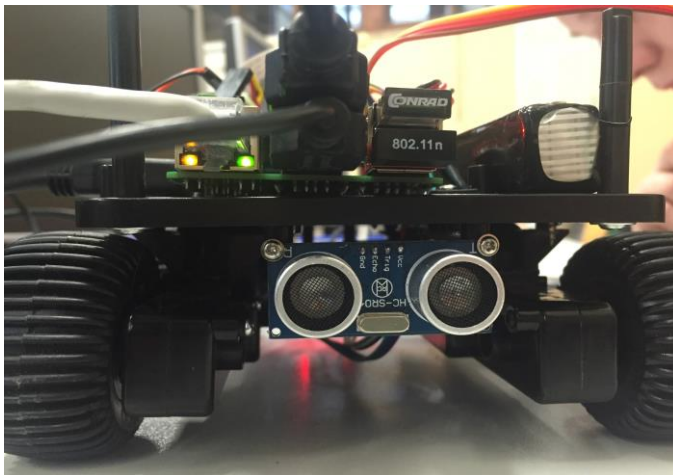
Pogonski sistem s pomočjo 4 vijakov velikosti 2 x 8 PB privijamo na namenske luknje na osnovni plošči.



Slika 54: Namestitev pogonskega sistema

### 7.4 MONTAŽA ULTRASONIČNEGA SENZORJA

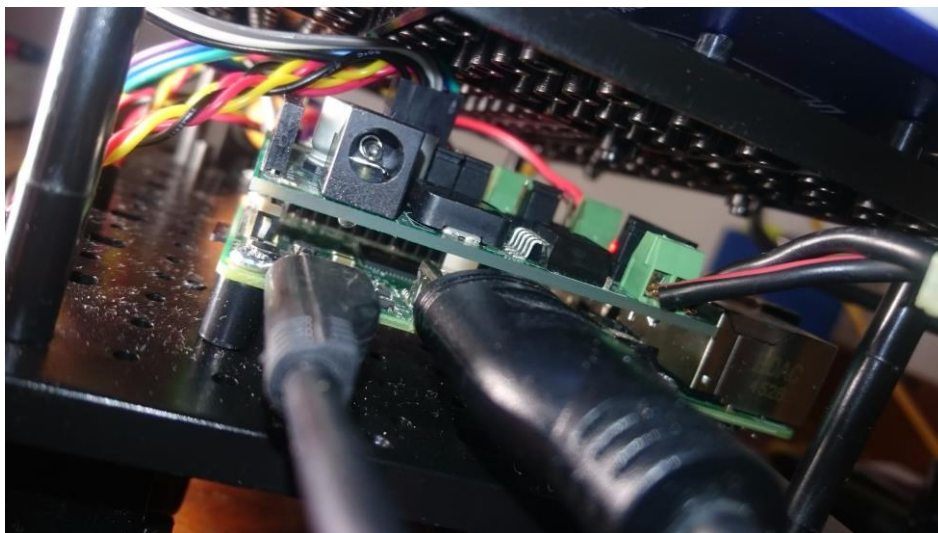
Senzor smo namestili na zadnjo stran mobilnega robota.



Slika 55: Ultrasonični senzor

## 7.5 MONTAŽA RASPBERRY PI Z OPREMO

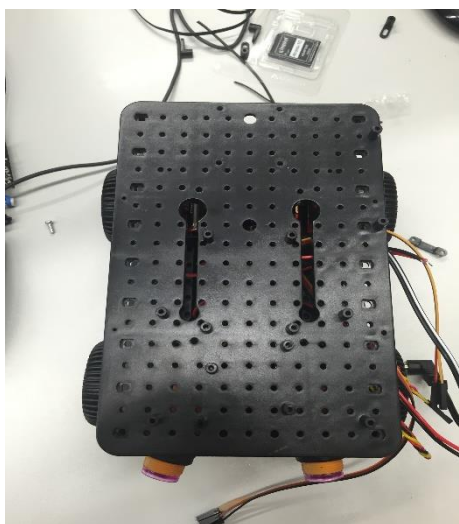
Za montažo smo uporabili štiri vijake 2.6 x 8P B. Opremo smo pritrdili na spodnjo ploskev.



Slika 56: Nameščena oprema

## 7.6 MONTAŽA ZGORNJE PLOSKVE

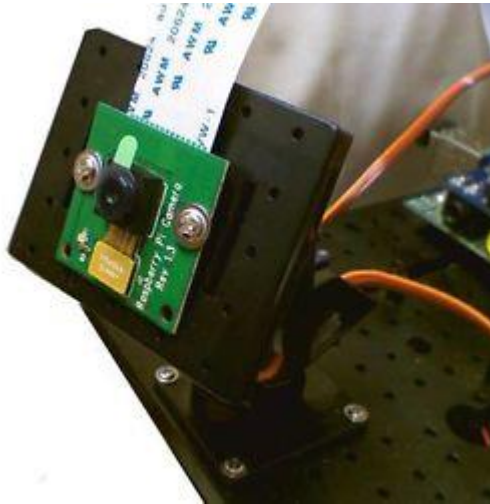
Za namestitev zgornje ploskve smo potrebovali 4 distančnike in 4 vijake velikosti 3 x 12 PB. Na zgornjo ploskev smo pritrdili sistem servomotorjev.



Slika 57: Nameščanje zgornje ploskve

## 7.7 MONTAŽA PAN/TILT-SISTEMA IN KAMERE

Za montažo PAN/TILT-sistema smo uporabili 4 vijake velikosti 3 x 12 PB. Sistem smo namestili na zgornjo ploskev. Za namestitev kamere pa smo uporabili 2 vijaka velikosti 3 x 8 PB.



Slika 58: Nameščanje PAN/TILT in kamere

## 7.8 MONTAŽA POLNILNE BATERIJE

Pri nameščanju polnilne postaje smo se odločili, da bomo spodnji pokrov preprosto privijačili v luknje, ki so na plošči avtomobila. Nosilec za polnilno postajo smo izdelali iz 2 mm pločevine.



## 7.9 POSTOPKI

1. Najprej smo s pomičnim merilom izmerili razdaljo med luknjama ter jo prenesli na pokrov.

Pri tem smo morali paziti, da so vijaki prišli med baterije, ki so v polnilni postaji, sicer se na koncu ne bi dala sestaviti.



Slika 59: Nosilec za polnilno postajo

2. Nato smo luknje ustrezno zarisali ter zatočkali, na koncu smo jih izvrtali s svedrom M3 GT.



Slika 60: Vrtanje lukenj na nosilcu

3. Na koncu smo preverili, če se luknje ujemajo z luknjami na platformi. Pokrov smo na avtomobil privijačili s samoreznimi vijaki M3.



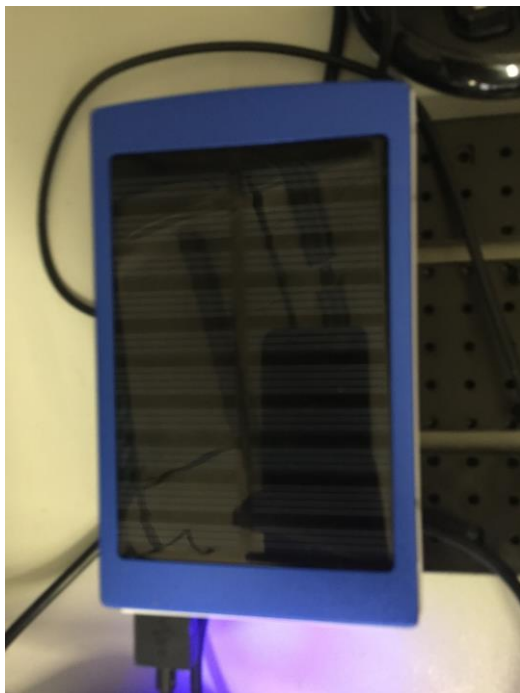
Slika 61: Vijačenje nosilca na platformo

4. Ko smo se prepričali, da je pokrov trdno privijačen, smo nanj postavili preostale dele postaje ter jih privijali s štirimi vijaki, po dva na vsaki strani.



Slika 62: Urejanje zadnjih malenkosti

5. Končni izdelek: Polnilna postaja s sončnimi celicami



Slika 63: Končni izdelek



## 8) PROGRAM ROBOTA

### 8.1 KOMENTAR PROGRAMA

**# import knjižnic**

import pygame

import time

import RPi.GPIO as GPIO

**# nastavitev GPIO pinov na način BOARD**

GPIO.setmode(GPIO.BOARD)

GPIO.setwarnings(False)

**#nastavitev vhodov in izhodov**

m1a = 19

m1b = 21

m2a = 24

m2b = 26

SL = 7

SR = 11

LR = 13

LL = 12

led1 = 40

**# določanje vhodov in izhodov**

GPIO.setup(m1a,GPIO.OUT)

GPIO.setup(m1b,GPIO.OUT)

GPIO.setup(m2a,GPIO.OUT)

GPIO.setup(m2b,GPIO.OUT)

GPIO.setup(SL,[GPIO.IN](#))

GPIO.setup(SR,[GPIO.IN](#))

GPIO.setup(LR,[GPIO.IN](#))

GPIO.setup(LL,[GPIO.IN](#))

GPIO.setup(led1,GPIO.OUT)

### **# določanje vrednosti**

ForBac = 0

LefRig = 0

### **# nastavitve PWM**

p1 = GPIO.PWM(m1a,250)

p2 = GPIO.PWM(m1b,250)

p3 = GPIO.PWM(m2a,250)

p4 = GPIO.PWM(m2b,250)

p1.start(0)

p2.start(0)

p3.start(0)

p4.start(0)

### **# Inicializacija modula pygame**

pygame.init()

j = pygame.joystick.Joystick(0)

j.init()

print 'Initialized Joystick : %s' % j.get\_name()

### **# čakanje na joystick**

while pygame.joystick.get\_count() == 0:

    pygame.joystick.get\_count()

    time.sleep(10)

    pygame.joystick.quit()

    pygame.joystick.init()

**# potreben procent premika joystika za izvajanje operacij (0.25 = 25%)**

threshold = 0.20

**# poimenovanje ps3 osi**

PS3\_AXIS\_RIGHT\_HORIZONTAL = 2

PS3\_AXIS\_RIGHT\_VERTICAL = 3

try:

while True:

**# zasveti led če se program izvaja**

GPIO.output(led1,1)

events = pygame.event.get()

for event in events:

UpdateMotors = 0

**# preveri če se joystick premika**

if event.type == pygame.JOYAXISMOTION:

if event.axis == 3: #če se os 3 premika vrne njeno vrednost pod spremenljivko ForBac

ForBac = event.value

UpdateMotors = 1

elif event.axis == 2: #če se os 2 premika vrne njeno vrednost pod spremenljivko LefRig

LefRig = event.value

UpdateMotors = 1

## # glavni program za motorje

```
if (UpdateMotors):
```

```
    if (ForBac > threshold and GPIO.input(LL)==0 ): #nazaj
```

```
        GPIO.output(m1a,0)
```

```
        GPIO.output(m1b,1)
```

```
        GPIO.output(m2a,1)
```

```
        GPIO.output(m2b,0)
```

```
        speed = int(ForBac*50)
```

```
        p2.ChangeDutyCycle(speed)
```

```
        p3.ChangeDutyCycle(speed)
```

```
        print 'nazaj'
```

```
    elif (ForBac < -threshold and GPIO.input(SL)==1 and GPIO.input(SR)==1 and  
GPIO.input(LL)==0 ):#naprej
```

```
        GPIO.output(m1a,1)
```

```
        GPIO.output(m1b,0)
```

```
        GPIO.output(m2a,0)
```

```
        GPIO.output(m2b,1)
```

```
        speed = -1*(int(ForBac*50))
```

```
        p1.ChangeDutyCycle(speed)
```

```
        p4.ChangeDutyCycle(speed)
```

```
        print 'naprej'
```

### **#obračanje desno**

```
elif (LefRig > threshold and ForBac ==0 and GPIO.input(SR)==1 and GPIO.input(LL)==0):  
    GPIO.output(m1a,0)  
    GPIO.output(m1b,1)  
    GPIO.output(m2a,0)  
    GPIO.output(m2b,1)  
    speed = int(LefRig*50)  
    p2.ChangeDutyCycle(speed)  
    p4.ChangeDutyCycle(speed)  
    print 'desno'
```

### **# obračanje levo**

```
elif (LefRig < -threshold and GPIO.input(SL)==1 and GPIO.input(LL)==0 ):  
    GPIO.output(m1a,1)  
    GPIO.output(m1b,0)  
    GPIO.output(m2a,1)  
    GPIO.output(m2b,0)  
    speed = -1*(int(LefRig*50))  
    p1.ChangeDutyCycle(speed)  
    p3.ChangeDutyCycle(speed)  
    print 'levo'
```

**# izklop motorjev**

else:

GPIO.output(m1a,0)

GPIO.output(m1b,0)

GPIO.output(m2a,0)

GPIO.output(m2b,0)

p1.ChangeDutyCycle(0)

p2.ChangeDutyCycle(0)

p3.ChangeDutyCycle(0)

p4.ChangeDutyCycle(0)

print 'off'

except KeyboardInterrupt:

j.quit()

p1.stop(0)

p2.stop(0)

p3.stop(0)

p4.stop(0)

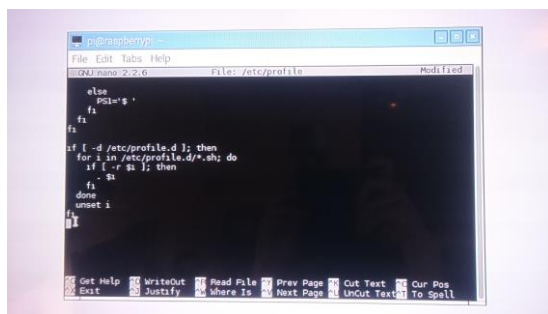
GPIO.cleanup()

## 8.2 AVTOMATSKI ZAGON PROGRAMA

Če želimo, da se nam program, ki smo ga napisali v programu python, prične izvajati avtomatsko, ne da bi ga bilo potrebno zagnati s pomočjo uporabniškega vmesnika in komponent, ga moramo dati med zagonska opravila operacijskega sistema. To storimo tako, da v terminal zapišemo:

### 1. `sudo nano /etc/profile`

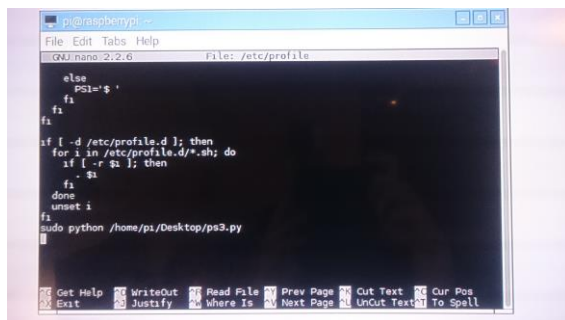
Prikaže se nam etc profil v GNU (Global Null Unit) meniju.



Slika 64: Ukaz

S kolesčkom miške drsamo do dna in dodamo vrstico, ki programu pove, kje v sistemu se nahaja python skript, ki ga želimo zagnati.

```
sudo python /home/pi/desktop/ps3.py
```



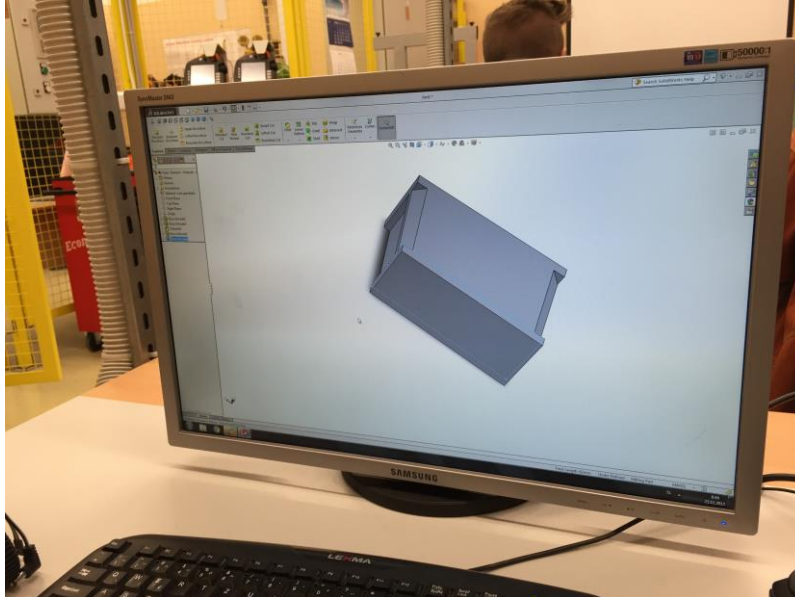
Slika 65: Ukaz

`sudo python` + lokacija naše datoteke.

Če ne vemo, kje se naša datoteka nahaja, jo poiščemo v File Managerju. Nato kliknemo Properties in prepišemo vrstico pod imenom Location. Za zadnji besedi dodamo poševnico ter ime datoteke s končnico .py

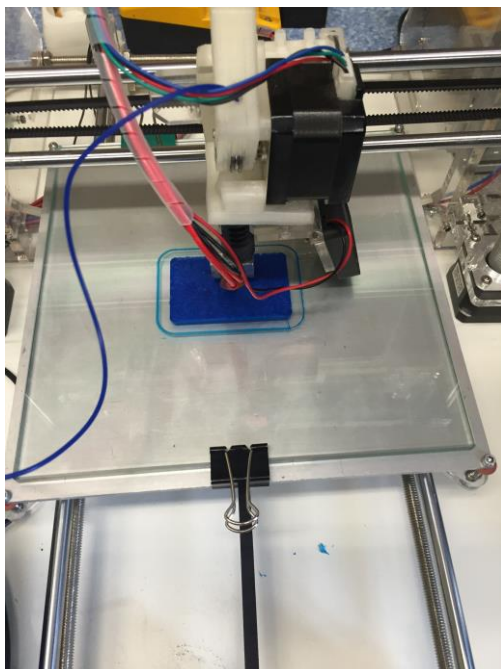
## 9) 3D-MODELIRANJE IN 3D-TISKANJE

### 9.1 MODELIRANJE V PROGRAMU SOLIDWORKS



Slika 66: Prikaz modeliranja ohišja za kamero v programu SolidWorks

### 9.2 3D-TISKANJE

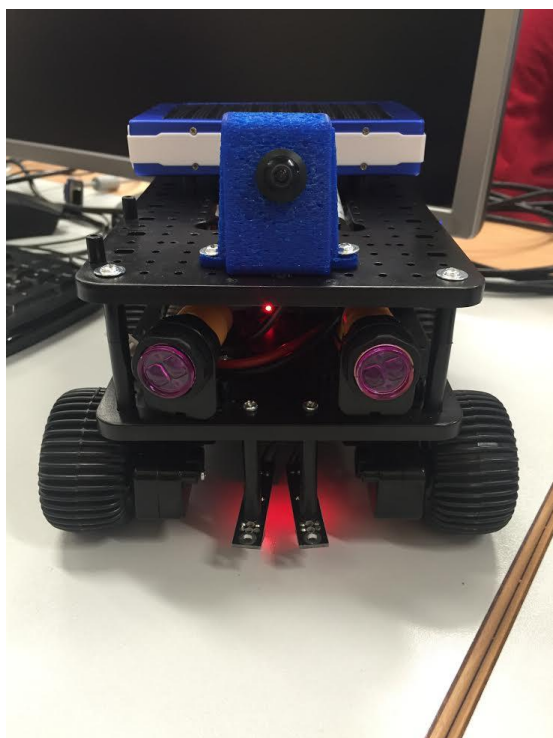


Slika 67: Prikaz 3D-tiskanja ohišja za kamero

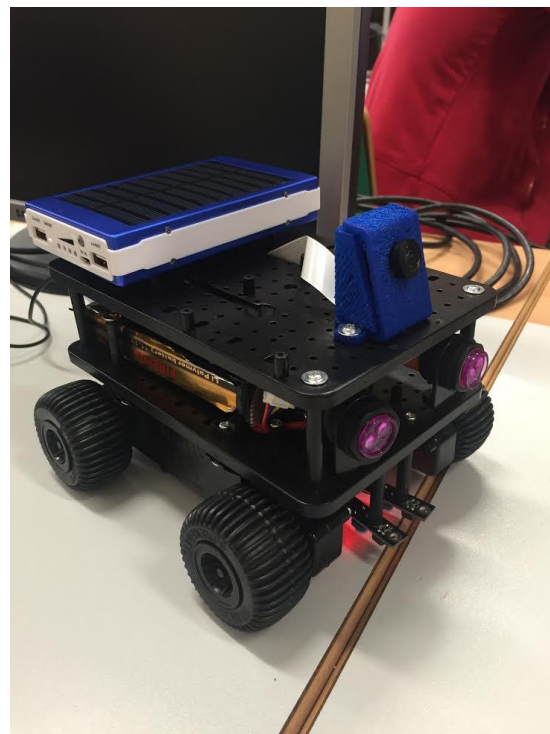


## 10) KONČNI IZDELEK

### 10.1 KONČNI IZDELEK










Slika 68: Končni izdelek



Slika 69: Končni izdelek

## 11) UGOTOVITVE IN REZULTATI

Na začetku raziskovalne naloge smo si zastavili naslednje hipoteze. Hipoteze so se izkazale za uspešne, vendar smo za to porabili veliko časa in znanja. Pri tem smo imeli tudi veliko problemov, ki smo jih uspešno realizirali.

Izdelati mobilnega robota z opravljanjem preko PS3 Bluetooth igralnega ploščka.	
Prenos slike v živo s kamere na telefon.	
Spoznati mikroračunalnik Raspberry Pi.	
Spoznati programsko okolje Python.	
Narediti estetsko ohišje.	
Ustvariti napajanje tudi s pomočjo fotocelic.	
Izdelati mobilnega avtomatiziranega robota.	

## 12) ZAKLJUČEK

Pri raziskovalni nalogi smo uporabili veliko znanja, ki smo ga pridobili v prejšnjih letih šolanja, vendar pa to ni bilo dovolj, saj smo se podali na neznana pota mikroračunalnika Raspberry Pi, s katerim smo se srečali prvič. Na novo smo morali usvojiti vsa osnovna znanja za delo z njim. Osnove smo dokaj hitro usvojili, nekaj težav pa smo imeli z nadaljnjimi postopki dela, kot sta prenos slike ter obdelava signalov senzorjev. Probleme smo uspeli rešiti, pa čeprav smo za njih porabili veliko časa. Kljub temu smo s pomočjo vodičev na spletu rešili večino problemov in smo z zaključnim izdelkom zadovoljni. Ker smo mladi in polni novih zamisli oziroma idej, vedno najdemo še kakšno pomanjkljivost, ki bi jo lahko odpravili. Celoten projekt je bil zanimiv, saj smo pridobili veliko novega znanja, s katerim bomo lahko v nadaljevanju delali še na večjih projektih.

## 13) VIRI IN LITERATURA

- (1) 3D-TISKANJE (spletni vir). 2015. (citirano 9. 3. 2016) Dostopno na naslovu:  
<http://www.mattercontrol.com/>
- (2) BLUETOOTH (spletni vir). 2016. (citirano 9. 3. 2016) Dostopno na naslovu:  
<https://www.raspberrypi.org/learning/robo-butler/bluetooth-setup/>
- (3) EL. VEZJA (spletni vir). 2016. (citirano 9. 3. 2016) Dostopno na naslovu:  
<http://www.eleccircuit.com/>
- (4) GITHUB (spletni vir). 2013. (citirano 9. 3. 2016) Dostopno na naslovu:  
<https://github.com/chrisalexander/initio-python>
- (5) KAMERA (spletni vir). 2016. (citirano 9. 3. 2016) Dostopno na naslovu:  
<http://raspberrypi.stackexchange.com/questions/23182/how-to-stream-video-from-raspberry-pi-camera-and-watch-it-live>
- (6) RASPBERRY PI (spletni vir). 2015. (citirano 9. 3. 2016) Dostopno na naslovu:  
<https://www.raspberrypi.org/>
- (7) SENZORJI (spletni vir). 2015. (citirano 9. 3. 2016) Dostopno na naslovu:  
<http://www.dexterindustries.com/GrovePi/supported-sensors/>
- (8) WI-FI (spletni vir). 2016. (citirano 9. 3. 2016) Dostopno na naslovu:  
<https://www.raspberrypi.org/documentation/configuration/wireless/>