
Projekti za kamere

39 projekata za Arduino i Raspberry Pi



Dogan Ibrahim

Agencija Eho
www.infoelektronika.net

- Sva prava zadržana. Nijedan deo ove knjige ne sme biti reprodukovan u bilo kom materijalnom obliku, uključujući fotokopiranje ili slučajno ili nenamerno smeštanje na bilo koji elektronski medijum sa ili uz pomoć bilo kog elektronskog sredstva, bez pismenog odobrenja nosioca autorskih prava osim u skladu sa odredbama zakona o autorskim pravima, dizajnu i patentima iz 1988. godine ili pod uslovima izdatim od Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London, England W1P 9HE. Prijave za pismene dozvole radi štampanja bilo kog dela ove publikacije upućuje se izdavaču ove knjige.
- Izjava: Autor i izdavač su uložili najveće napore da bi se obezbedila tačnost informacija sadržanih u ovoj knjizi. Autor i izdavač ne mogu da pretpostave neprijatnosti i ovom izjavom isključuju bilo kakvu odgovornost za bilo koju stranku koja bi imala gubitke ili štetu uzrokovanu greškama ili propustima u ovoj knjizi, bez obzira da li su greške ili propusti nastali usled nemara, nezgode ili bilo kog drugog razloga.
- Zahvalnost: Autori bi želeli da izraze svoju zahvalnost kompaniji „Espressif Systems“ za davanje dozvole za uključivanje ESP-EYE slika i podataka u ovoj knjizi. Autori se takođe zahvaljuju na pomoći i dragocenim sugestijama Volkera Bombiena tokom čitavog trajanja pripreme ove knjige. Autor želi da se zahvali i dr Adrianu Rosebrocku iz „PyImageSearch“ tima za davanje dozvole za upotrebu instrukcija za instaliranje biblioteke OpenCV za Raspberry Pi u ovoj knjizi.

ISBN 978-86-80134-25-3

Projekti za kamere

Naslov originala: Camera Projects

Autor: Dogan Ibrahim

Prevod: Biljana Tešić

Izdaje i štampa: Agencija Eho, Niš

e-mail: redakcija@infoelektronika.net

Tiraž: 300

Godina izdanja: 2019

<p>CIP - Каталогизација у публикацији Народна библиотека Србије, Београд 004.383/.384 681.772.2:004.383.3 ДОГАН, Ибрахим, 1954 Projekti za kamere : 39 projekata za Arduino i Raspberry Pi / Dogan Ibrahim ; [prevod Biljana Tešić]. - Niš : Agencija Eho, 2019 (Niš : Agencija Eho). - 251 str. : ilustr. ; 25 cm Prevod dela: Camera Projects. - Tiraž 300. ISBN 978-86-80134-25-3 а) Микроконтролери -- Примена б) Дигиталне камере -- Примена COBISS.SR-ID 278432268</p>

Predgovor	11
Poglavlje 1 • Pikseli, slike, datoteka slika i terminologija kamere.....	12
1.1 Pregled.....	12
1.2 Pikseli.....	12
1.3 Boje piksela	12
1.4 Datoteke slika	13
1.4.1 JPEG	13
1.4.2 TIFF	14
1.4.3 GIF.....	15
1.4.4 BMP	15
1.4.5 PNG.....	16
1.6 Rezime	18
Poglavlje 2 • Raspberry Pi i Arduino kompatibilne kamere	19
2.1 Pregled.....	19
2.2 Raspberry Pi kamere	19
2.2.1 Standardne Raspberry Pi dnevne kamere.....	19
2.2.2 Standardne Raspberry Pi infracrvene kamere.....	21
2.2.3 Unistorm noćna Raspberry Pi kamera.....	22
2.2.4 Dodatna oprema Raspberry Pi kamere za kamere zasnovane na CSI magistrali	22
2.2.5 Veb kamere	24
2.3 Arduino Uno kamera	25
2.3.1 Kamera ArduCAM OV2640.....	25
2.3.2 Kamera ArduCAM OV5642.....	25
2.3.3 Arduino JPEG modul za kameru.....	26
2.3.4 Kamera ArduCAM MT9D111	26
2.3.5 TTL serijska kamera	27
2.4 Rezime	27
Poglavlje 3 • Projekti za Raspberry Pi kamere	28
3.1 Pregled.....	28
3.2 Instaliranje kamere	28
3.3 Projekat 1 – Komande za statične slike	30
3.3.1 raspistill	30

3.4 Projekat 2 – Kreiranje kamere sa snimanjem u vremenskim razmacima	36
3.5 Projekat 3 – Komande video kamere	42
3.6 Korišćenje USB kamere sa Raspberry Pi-jem.....	43
3.7 Projekat 4 – Ko zvonči na vratima? – Bluetooth rešenje.....	51
3.8 Projekat 5 – Uživo strimovanje videa na platformi YouTube	55
3.9 Rezime	60
Poglavlje 4 - Jednostavni Raspberry Pi projekti pomoću Pythona	61
4.1 Pregled.....	61
4.2 Python picamera biblioteka	61
4.3 Korišćenje kamere u Python programima	61
4.4 Projekat 1 – Snimanje više slika.....	61
4.5 Postavke kamere.....	63
4.5.1 Dodavanje teksta	63
4.5.2 Promena svetline i kontrasta slika	64
4.5.3 Efekti slike	65
4.5.4 Režimi ekspozicije	65
4.5.5 Ostale postavke	65
4.6 Uzastopno snimanje slika.....	65
4.7 Kontinuirano snimanje slika.....	66
4.8 Projekat 2 – Snimanje slika u vremenskim razmacima	67
4.9 Projekat 3 – Snimanje slika pomoću dugmeta – snimanje selfija	68
4.10 Projekat 4 – Snimanje selfija pomoću zujalice	72
4.11 Projekat 4 – Snimanje selfija: fotografisanje u vremenskim razmacima pomoću zujalice	75
4.12 Projekat 6 – Snimanje selfija pomoću displeja	77
4.13 Projekat 7 – Snimanje selfija: dodavanje softverske tastature na desktopu	84
4.14 Projekat 8 – Snimanje selfija pomoću softverske tastature – snimanje više slika	86
4.15 Projekat 9 – Dizajniranje photobox-a	88
4.17 Projekat 11 – Snimanje videa: korišćenje dugmeta za pokretanje/zaustavljanje snimanja.....	93
4.18 Projekat 12 – Dizajniranje digitalnog okvira za fotografije	95
4.19 Projekat 13 – Slanje snimljene slike putem e-pošte.....	96
4.20 Projekat 14 – Ko zvonči na mojim vratima? Rešenje u kome se koristi e-pošta.....	99
4.21 Projekat 15 – Snimanje videa biljnog i životinjskog sveta pomoću PIR senzora	102

4.22 Projekat 16 – Objavljivanje slike na Twitter-u.....	107
4.32 Rezime	113
Poglavlje 5 • Kreiranje špijunske kamere koja se aktivira na pokret	114
5.1 Pregled.....	114
5.2 Detekcija pokreta.....	114
5.3 Projekat 1 – Špijunska kamera koja se aktivira na pokret	114
5.4 Projekat 2 – Špijunska kamera koja se aktivira na pokret sa LED-om	118
5.5 Rezime.....	120
Poglavlje 6 • Blic za Raspberry Pi kameru	121
6.1 Pregled.....	121
6.3 Projekat 1 – Nesinhronizovani blic.....	121
6.3 Projekat 2 – Sinhronizovani blic.....	124
6.4 Rezime	127
Poglavlje 7 • Dizajniranje Raspberry Pi sistema kamera za nadzor	128
7.1 Pregled.....	128
7.2.1 Motion biblioteka.....	128
7.2.2 Konfigurisanje motion softvera	128
7.2.3 Pokretanje motion servera	130
7.2.4 Prilagođavanje Motion servera.....	131
7.2.5 Slanje e-pošte kada se detektuje pokret.....	132
7.2.6 Opšte napomene	134
7.3 Rezime	134
Poglavlje 8 • motionEyeOS softver za nadzor	135
8.1 Pregled	135
8.2 Instaliranje softvera motionEyeOS	135
8.3 Početna konfiguracija.....	138
8.4 Napredna konfiguracija.....	139
8.5 Konfigurisanje detekcije pokreta i automatskog slanja e-pošte	140
8.6 Rezime	142
Poglavlje 9 • OpenCV softverski paket.....	143
9.1 Pregled.....	143

9.2 Kopiranje Raspberry Pi microSD kartice	143
9.3 Instaliranje OpenCV softverskog paketa na Raspberry Pi-ju	145
9.4 Upotreba OpenCV softverskog paketa na Raspberry Pi-ju	149
9.4.1 Povezivanje pomoću 7-inčnog Raspberry Pi displeja	149
9.4.2 Povezivanje pomoću Putty-a i vncserver-a	149
9.4.3 Povezivanje pomoću Putty-a i Xming-a	150
9.5 Osnovne OpenCV operacije za slike	152
9.5.1 imread(), imshow(), waitKey(), destroyAllWindows(), imwrite().....	153
9.5.2 Crtanje oblika	155
9.5.3 Transformacije slika	159
9.6 Detekcija ivica – Canny Edge Detection.....	164
9.7 Detekcija kruga – Hough Circle Transformation	165
9.8 Detekcija linije – Hough Line Transformation	167
9.9 Aritmetičke operacije na slikama	169
9.9.1 Dodavanje slika	169
9.9.2 Oduzimanje slika	171
9.10 Morfološke operacije na slikama	173
9.11 Jednostavno prepoznavanje geometrijskih oblika.....	174
9.12 Zamućivanje/izgladivanje slike	176
9.13 Detektovanje boja na slikama	178
9.14 Projekat 1 – Detekcija lica i očiju.....	180
9.15 Rezime	182
Poglavlje 10 • Snimanje slika sa kamere pomoću OpenCV-a	183
10.1 Pregled.....	183
10.2 Snimanje slika sa Raspberry Pi kamere pomoću OpenCV-a	183
10.2.1 Projekat 1 • Snimanje nekoliko slika.....	185
10.2.2 Projekat 2 – Snimanje slika u vremenskim razmacima	186
10.2.3 Projekat 3 – Snimanje slika u određenom vremenu	187
10.3 Snimanje videa pomoću Raspberry Pi kamere.....	188
10.4 Upotreba USB kamere sa OpenCV-om	189
10.4.1 Projekat 4 – Snimanje više slika.....	190
10.4.2 Snimanje video kadrova u OpenCV-u pomoću USB kamere	191

10.4.3 Projekat 5 – Snimanje videa	192
10.5 Rezime	194
Poglavlje 11 • Upotreba SimpleCV-a u projektima za kameru	195
11.1 Pregled.....	195
11.2 Bar-kodovi	195
11.2.1 Code 39	195
11.2.2 Code 128	195
11.2.3 Universal Product Code	195
11.3 Projekat 1 – Šifra bar-koda	196
11.4 Projekat 2 – Detekcija pokreta pomoću zujalice	200
11.5 Rezime	202
Poglavlje 12 • Upotreba kamere pomoću Arduina	203
12.1 Pregled.....	203
12.2 Projekat 1 – Interfejs Arduino Uno SD kartice: Upisivanje na karticu	203
12.3 Projekat 2 – Interfejs Arduino Uno SD kartice: Čitanje/upisivanje u karticu.....	208
12.4 Arducam kamera.....	213
12.4.1 Arducam interfejs.....	215
12.4.2 Interfejs za više kamera	215
12.4.3 Projekat 3 – Snimanje slika pomoću Arducam-a i Arduina	216
12.4.5 Projekat 5 – Snimanje slike pomoću spoljnog dugmeta zatvarača	231
12.5 Rezime	237
Poglavlje 13 • Automatsko prepoznavanje broja registarske tablice	238
13.1 Pregled.....	238
13.2 Brojevi registarskih tablica vozila	238
13.3 Biblioteka OpenALPR.....	238
13.4 Upotreba sa komandnom linijom.....	238
13.4.1 Projekat 1 – Prepoznavanje broja registarska tablice iz datoteke	238
13.4.2 Projekat 2 – Prepoznavanje sa kamere	240
13.5 Upotreba biblioteke OpenALPR sa Python-om	241
13.5.1 Projekat 3 – Prepoznavanje broja registarska tablica pomoću Python-a.....	242
13.5.2 Projekat 4 - Tata je stigao kući, stavite čajnik!	244
13.6 Rezime	247

Dodatak A • Raspored pinova za Raspberry Pi 3	248
Dodatak B • ESP-EYE	249
B.1 Pregled	249
B.2 Detekcija i prepoznavanje lica	250
B.3 Dodavanje novog lica	251
B.4 Brisanje postojećeg lica	251

Predgovor

U ovoj knjizi su opisani projekti za Raspberry Pi 3 i Arduino kamere. U knjizi je jednostavnim rečima i testiranim i radnim primerima projekata objašnjeno kako se konfiguriše i koristi Raspberry Pi kamera i veb kamera zasnovana na USB-u projektima za kamere koje koriste Raspberry Pi. Dati su primeri projekata za snimanje slika, fotografisanje u određenim vremenskim razmacima, snimanje video zapisa, upotrebu kamere i Raspberry Pi-ja u aplikacijama za bezbednost i nadzor, slanje slika na Twitter, snimanje biljnog i životinjskog sveta, emitovanje uživo na YouTube-u, korišćenje noćne kamere, slanje slika na pametne telefone, detekciju lica i očiju, prepoznavanje oblika i boja, prepoznavanje registarskih tablica, prepoznavanje bar-koda itd. Instaliranje i upotreba popularnih biblioteka za obradu slika i softvera, uključujući OpenCV, SimpleCV i OpenALPR, detaljno su objašnjeni pomoću Raspberry Pi-ja.

U knjizi je takođe detaljno objašnjeno kako se koristi kamera na Arduino razvojnoj ploči za snimanje slika i njihovo čuvanje na microSD karticu.

Svi projekti navedeni u knjizi su u potpunosti testirani i funkcionalni. Sledeći podnaslovi se koriste u projektima u kojima je to primenljivo:

- Naziv projekta
- Opis projekta
- Cilj projekta
- Blok dijagram
- Električna šema
- Listing programa

Popis programa za sve Raspberry Pi i Arduino projekte koji se koriste u knjizi dostupni su na Elektorovom veb sajtu i sajtu InfoElektronike.

Nadam se da će čitaocima ova knjiga biti korisna i da će uživati u njenom čitanju.

Prof. dr Dogan Ibrahim
London

Poglavlje 1 • Pikseli, slike, datoteka slika i terminologija kamere

1.1 Pregled

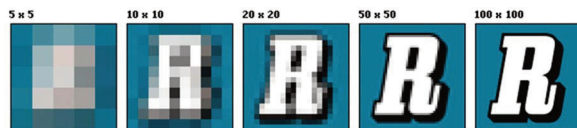
Pre nego što pogledate detalje o kamerama i njihovoj upotrebi u projektima za Raspberry Pi i Arduino, vredi nakratko pogledati opšte koncepte slika. U ovom poglavlju će biti reči o pikselima, slikama, datoteka slika i bojama.

1.2 Pikseli

Možemo položiti sliku na mrežu sa kvadratima. Pikseli su tada osnovni gradivni blokovi slike u kojima piksel zauzima jedan kvadrat mreže. Ova mreža se takođe naziva bit mapa. Slika sa 1024 x 768 piksela je mreža sa 1024 kolone i 768 redova, tako da sadrži $1024 \times 768 = 786,432$ piksela. Važno je shvatiti da poznavanje broja piksela na slici ne ukazuje na fizičke dimenzije te slike, jer moramo znati veličinu piksela pre nego što saznamo fizičku veličinu čitave slike. Fizička veličina slike može se proceniti iz poznavanja piksela po inču (PPI, Pixels-Per-Inch), odnosno broja piksela u jednom inču slike. Ako je na primer PPI 100, a slika ima 1024 x 768 piksela, onda je fizička veličina slike 10,24 inča x 7,68 inča. Broj piksela koji čine sliku je takođe poznat kao rezolucija te slike. Rezolucije velikih slika obično se označavaju megapikselima.

U tehnologiji štampe rezolucija se označava tačkama po inču (DPI, Dots-Per-Inch) koje prikazuju broj tačaka u jednom inču štampanog materijala. tj. to je fizička gustina tačke slike kada se reprodukuje na papiru. Različiti štampači postižu različite rezolucije. Na primer, rezolucija inkdžet štampača je 300-700 DPI. Rezolucija laserskih štampača je oko 600-2500 DPI. Vrednost DPI štampača treba da bude znatno veća od PPI vrednosti slike da bi se dobio sličan kvalitet ispisa.

Slika postaje jasnija kako se broj PPI (ili rezolucija) povećava. Na slici 1.1 prikazana je slika sa različitim rezolucijama. U ovom primeru, iako je slika veoma loša u rezoluciji 10 x 10, mnogo je jasnija u rezoluciji 100 x 100.



Slika 1.1 Slika u različitim rezolucijama (www.wikipedia.org/wiki/Image_resolution)

1.3 Boje piksela

Boja koju predstavlja piksel zavisi od broja bitova koji se koriste za informacije o boji. Monohromatske slike se sastoje od crno-belih slika i koriste samo dva bita. Pomoću 8 bitova možemo predstaviti do 256 boja, pomoću 16 bitova 65.536 boja (Highcolor), a sa 24 bita možemo predstaviti 16.777.216 boja (Truecolor). Boja piksela je predstavljena kombinacijom crvene, zelene i plave boje (RGB, Red, Green and Blue), ti za predstavljanje cr-

vene boje, 8 bitova za predstavljanje zelene i 8 bitova za prikazivanje plave boje. Vrednosti boja mogu biti od 0 do 255. Neki primeri kombinacija boja su dati u nastavku:

	Crvena	Zelena	Plava
Bela boja:	255	255	255
Crna boja:	0	0	0
Crvena boja:	255	0	0
Žuta boja:	255	0	0
Ružičasta boja:	255	255	255

U štampanju u boji koriste se suptraktivne boje cijan, magenta i žuta (CMY, Cyan, Magenta and Yellow). Na primer, mešanjem cijan i žute dobija se zelena boja, mešanjem magente i žute dobija se crvena boja, mešanjem sve tri boje dobija se crna boja i tako dalje.

1.4 Datoteke slika

Danas se koristi mnogo datoteka slika. U ovom odeljku ćemo se osvrnuti na neke od najčešće korišćenih i poznatih svojstava datoteka slika. Kolorne datoteke slika su veoma velike datoteke. Na primer, datoteka slike 640 x 480 piksela sa 24 bitnim informacijama o boji zauzima $640 \times 480 \times 24 = 7.372.800$ bitova, odnosno 921.600 bajtova, što je skoro jedan megabajt. Čuvanje takvih velikih datoteka može zauzeti previše prostora za skladištenje. Takođe, obrada ili slanje velikih datoteka putem Interneta može potrajati dugo.

Pošto su datoteke u boji veoma velike, one se obično komprimuju radi uštede prostora za skladištenje. Danas postoje dva tipa algoritama kompresije: kompresija bez gubitaka i kompresija sa gubicima. Kompresija bez gubitaka čuva savršenu kopiju slike i pokušava smanjiti veličinu datoteke. Kompresija sa gubicima kreira mnogo manje datoteke, ali slika nije savršena kopija originala.

Karakteristike nekih od najčešće korišćenih datoteka slika date su u nastavku.

1.4.1 JPEG

JPEG (Joint Photographic Experts Group) je jedan od najčešće korišćenih formata datoteka slika koji se zasniva na kompresiji sa gubicima. Oznaka tipa tih datoteka je **.JPG** ili **.JPEG**. Većina kamera i drugih sistema za digitalno snimanje podržava JPEG format koji daje 8-bitne slike u sivim tonovima ili slike u 24-bitnim bojama. Iako JPEG smanjuje veličinu datoteke, kvalitet slike se smanjuje zbog kompresije sa gubicima. Step kompresije, a samim tim i kvalitet proizvedene slike se može promeniti pomoću mnogih aplikativnih programa. Na slici 1.2 prikazana je slika u JPEG formatu, gde je veličina datoteke 32 KB.

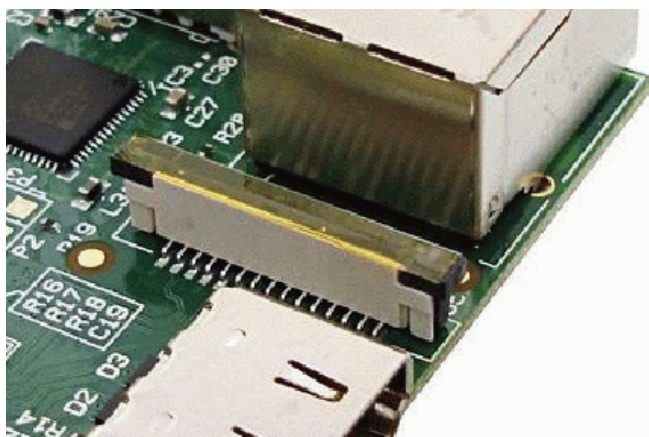
Poglavlje 2 • Raspberry Pi i Arduino kompatibilne kamere

2.1 Pregled

U ovom poglavlju ćemo razmotriti osnovne karakteristike različitih Raspberry Pi i Arduino Uno kompatibilnih kamera.

2.2 Raspberry Pi kamere

Raspberry Pi računari prihvataju kamere koje su kompatibilne sa magistralom CSI (Camera Serial Interface). CSI ima veoma velike brzine prenosa podataka. Priključak CSI magistrale (Slika 2.1) je ZIF 15 utičnica za montažu na površini koja se povezuje sa kamerom preko trakastog kabela. Pored toga, moguće je priključiti i kamere koje su kompatibilne sa USB magistralom (npr. veb kameru) na Raspberry Pi USB port.



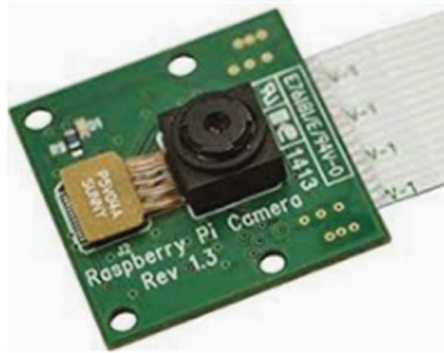
Slika 2.1 Konektor CSI magistrale

U osnovi postoje 4 modela Raspberry Pi kamera koje su zvanično kompatibilne sa CSI magistralom. Karakteristike ovih kamera i one koje su proizvele druge kompanije su ukratko opisane u sledećim odeljcima.

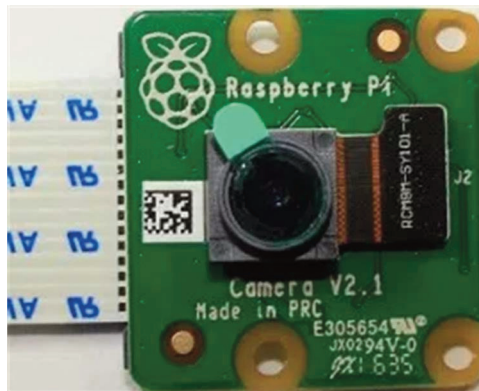
2.2.1 Standardne Raspberry Pi dnevne kamere

Ovo je bila prva verzija kamere Raspberry Pi (v1), koju je 2013. godine objavila kompanija fondacija Raspberry Pi. Ova kamera (slika 2.2) ima OmniVision OV5647 tip senzora od 5 megapiksela, koji daje rezoluciju od 2592 x 1944 piksela i dizajniran je za upotrebe po danu.

Godine 2016. Raspberry Pi fondacija je objavila verziju 2 (v2) CSI kamere (slika 2.3), koja ima 8 megapiksela i nešto šira horizontalna i vertikalna vidna polja, i 3280 x 2464 piksela. Ova nova kamera koristi Sony IMXS219.



Slika 2.2 Standardna Raspberry Pi v1 kamera



Slika 2.3 Standardna Raspberry Pi v2 kamera

U tabeli 2.1 data su poređenja V1 kamere i V2 kamere.

	Camera v1	Camera v2
Resolution (Still)	5 megapixels	8 megapixels
Video modes	1080p30, 720p60, 640x480p60/90	1080p30, 720p60, 640x480p60/90
Sensor	OmniVision OV5647	Sony IMX219
Sensor resolution	2592 x 1944	3280 x 2464
Pixel size	1.4 μ m x 1.4 μ m	1.12 μ m x 1.12 μ m
Sensor image area	3.76mm x 2.74mm	3.68mm x 2.76mm
Focal length	3.60mm	3.04mm
Focus	Fixed focus	Fixed focus
Horizontal field of view	53.50 degrees	62.2 degrees
Vertical field of view	41.41 degrees	48.8 degrees
Shutter	Rolling shutter	Rolling shutter

Poglavlje 3 • Projekti za Raspberry Pi kamere

3.1 Pregled

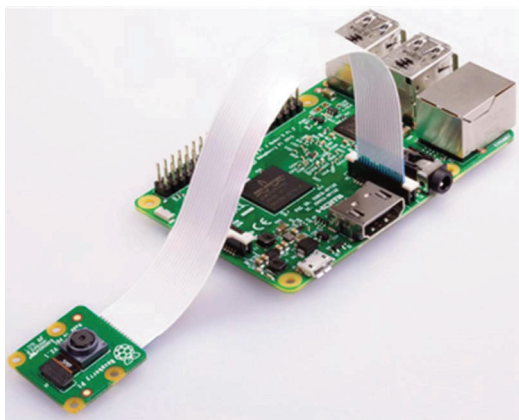
U prethodnim poglavljima videli smo osnovne karakteristike Raspberri Py i Arduino računara. Takođe smo videli osnovnu terminologiju kamere i tipove Raspberry Pi i Arduino kamera koji su dostupni na tržištu.

U ovom poglavlju ćemo razviti različite projekte za kameru koristeći Raspberry Pi. U prvim odeljcima poglavlja opisano je kako se instalira i koristi kamera iz Raspberry Pi komandne linije. U kasnijim odeljcima razvićemo zanimljivije projekte zasnovane na kameri koristeći programski jezik Python.

3.2 Instaliranje kamere

Koraci za instaliranje kamere na Raspberry Pi-ju su sledeći:

- Uverite se da je vaš Raspberry Pi isključen.
- Pronađite port kamere, koji je označen kao CSI ili CAMERA.
- Nežno povucite klizač plastičnog držača porta kamere držeći ga za oba kraja. Klizač držača kabla će se otvoriti.
- Umetnite trakasti kabl kamere (sa stranom konektora okrenutom od uređaja) USB utičnice) u držač kabla.
- Gurnite klizač plastičnog držača kabla tako da se on zaključa i drži kabl.
- Primenite napajanje na Raspberry Pi.



Slika 3.1 Kamera koja je instalirana na Raspberry Pi

Sada bi trebalo da možete da testirate interfejs kamere i da izvršite operacije kamere da biste snimili nepokretne slike i video kadrove. Ovo će biti opisano u narednim odeljcima pomoću Raspberry Pi komandne linije.

Postoje četiri komande kamere koje su dostupne iz komandnog interfejsa. To su: **raspi-still** i **raspiyuv** se koriste za snimanje slika. **raspistill** koristi komponentu Image Encode, a **raspiyuv** ne koristi koder već šalje izlaz direktno iz kamere u datoteku.

raspivid i **raspividuvc** se koriste za snimanje video zapisa. **raspivid** koristi komponentu Video Encode, a **raspividuvc** šalje izlaz direktno u datoteku. Detaljne informacije o komandama kamere dostupne su na sledećim veb sajtovima:

<https://www.raspberrypi.org/app/uploads/2013/07/RaspiCam-Documentation.pdf>

i

<https://www.raspberrypi.org/documentation/raspbian/applications/camera.md>

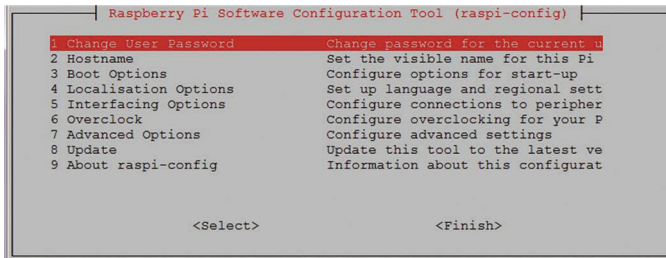
Pre korišćenja komandi kamere iz Raspberry Pi komandne linije, morate pokrenuti softver za emulaciju terminala, kao što je **Putty**, tako da možete da unesete komande. Ako imate direktnu vezu sa Raspberry Pi-jem pomoću monitora, onda nema potrebe da koristite **Putty** i možete da unosite komande direktno pomoću tastature.

Takođe, moramo omogućiti interfejs kamere na našem Raspberry Pi-ju. Koraci su sledeći:

Prijavite se na Raspberry Pi daljinskim putem (npr. koristeći **Putty**)

Ukucajte komandu **sudo raspi-config** da biste prikazali **Configuration Tool** (slika 3.2)

```
pi@raspberrypi:~ $ sudo raspi-config
```



Slika Raspberry Pi Configuration Tool

Pristupite meniju **Interfacing Option** i pritisnite taster **Enter**.

Poglavlje 4 • Jednostavni Raspberry Pi projekti pomoću Pythona

4.1 Pregled

U posljednjem poglavlju smo videli kako se kontroliše kamera iz Raspberry Pi komandne linije. U ovom poglavlju ćemo razvijati projekte za kameru koristeći Raspberry Pi hardver sa programskim jezikom Python.

4.2 Python picamera biblioteka

picamera je Python paket koji obezbeđuje programski interfejs za modul kamere Raspberry Pi. **picamera** je podrazumevano unapred instaliran na vašem Raspberry Pi-ju, ali njegova najnovija verzija se može instalirati na sledeći način:

```
pi@raspberrypi:~ $ sudo apt-get install python-picamera
```

4.3 Korišćenje kamere u Python programima

Korišćenje Raspberry Pi kamere u Python programima je jednostavno. Funkcija **start_preview** se koristi za pregled slike. Funkcija **capture** se zatim koristi za snimanje slike u kojoj je naziv slike priložen kao argument funkcije. Funkcija **stop_preview** zaustavlja pregled. Primitite da **start_preview** funkcioniše samo ako je monitor direktno povezan sa Raspberry Pi-jem. Neće funkcionisati u komandnom režimu ili u režimu daljinskog pristupa (SSH).

U primeru interaktivnog Python koda koji je prikazan ispod, slika je snimljena kamerom i pohranjena je u datoteci **myfile.jpg** u okviru fascikle **/home/pi/mypics**. Slika je podrazumevano snimljena u rezoluciji 640 x 480 (zapamtite da ukucate **python** da biste pristupili Python interaktivnom režimu). Trebalo bi da koristite iskaz **camera.close()** na kraju da biste oslobodili sve resurse koji su dodeljeni kameri:

```
>>> from picamera import PiCamera
>>> camera = PiCamera()
>>> camera.start_preview()
>>> camera.capture("/home/pi/mypics/myfile.jpg")
>>> camera.stop_preview()
>>> camera.close()
```

Na primer, možemo da promenimo rezoluciju u 1024 x 768 pomoću komande:

```
camera.resolution = (1024, 768)
```

4.4 Projekat 1 – Snimanje više slika

Opis: U ovom projektu ćemo snimati 5 slika u vremenskim intervalima od 5 sekundi i sačuvati sve datoteke slika u fascikli **/home/pi/mypics**. Slike će imati nazive **Pic1.jpg**, **Pic2.jpg**, **Pic3.jpg**, **Pic4.jpg** i **Pic5.jpg**.

Cilj: Cilj ovog projekta je da pokaže kako se više slika može snimiti i sačuvati u fasciklu.

Listing programa: Na slici 4.1 prikazana je lista programa (program: **multipics.py**). Na početku programa uključeni su moduli **time** i **picamera**. Zatim se formira petlja **for** koja se ponavlja 5 puta. Unutar ove petlje se prikazuju poruke i slike se snimaju i čuvaju u datotekama na svakih 5 sekundi. Na slici su prikazane poruke sa ekrana. Program treba da pokrenete unosom sledeće komande:

```
pi@raspberrypi:~ $ python multipics.py

#-----
#
#           CAPTURING MULTIPLE PICTURES
#           =====
#
# In this program 5 pictures are captured with 10 seconds intervals
# and these pictures are stored in folder /home/pi/mypics with the
# names Myimg1.jpg, Myimg2.jpg, Myimg3.jpg, Myimg4.jpg, Myimg5.jpg.
#
# Author: Dogan Ibrahim
# File  : multipics.py
# Date  : January, 2019
#-----

import time
from picamera import PiCamera
camera = PiCamera()

camera.start_preview()

for c in range(5):
    time.sleep(5)
    print("Capturing image: %d of %d" %(c+1,5))
    camera.capture('/home/pi/mypics/Pic%s.jpg' %(c+1))

camera.stop_preview()
camera.close()
```

Slika 4.1 Listing programa

```
pi@raspberrypi:~ $ python multipics.py
Capturing image: 1 of 5
Capturing image: 2 of 5
Capturing image: 3 of 5
Capturing image: 4 of 5
Capturing image: 5 of 5
pi@raspberrypi:~ $
```

Slika 4.2 Poruke koje su prikazane na ekranu

Poglavlje 5 • Kreiranje špijunske kamere koja se aktivira na pokret

5.1 Pregled

U posljednjem poglavlju smo videli dizajniranje nekoliko projekata u kojima se koristi standardna Raspberry Pi kamera. U ovom poglavlju ćemo dizajnirati špijunsku kameru koja se aktivira na pokret, a koja slika automatski kada se otkrije kretanje unutar prostorije.

5.2 Detekcija pokreta

Pokret ili prisustvo osobe se obično detektuje pomoću PIR (Passive Infrared) senzorskih modula (pogledajte sliku 5.1), koji se u osnovi sastoje od piroelektričnog senzora (pogledajte i odeljak 4.21). Rad senzora se zasniva na detekciji toplote koju emituje telo. Prema tome, kada čovek ili životinja uđe u domet senzora, toplota koju emituju tela u vidu infracrvene energije aktivira senzor. Modul senzora se sastoji od posebno dizajniranog poklopca nazvanog Fresnelov objektiv koja fokusira infracrvene signale na senzor. PIR senzorski moduli su 3 terminalski uređaji sa pinovima: + 5V, GND i OUT. Dva mala potenciometra su obezbeđena na većini modula (npr. model HC-SR501) za podešavanje osetljivosti (obično do 7 metara) i vremena kašnjenja (obično 0,3 sekunde do 5 minuta). Vreme kašnjenja postavlja vreme za koje izlaz ostaje visok nakon što je detektovan pokret. Neki moduli sadrže kratkospojnik za podešavanje režima okidanja modula. U neponovljivom režimu okidanja (ili u režimu pojedinačnog okidanja) izlaz se smanjuje nakon vremena kašnjenja. U ponovljivom režimu aktiviranja izlaz ostaje visok dok god je objekat u dometu.



Slika 5.1 PIR senzorski modul

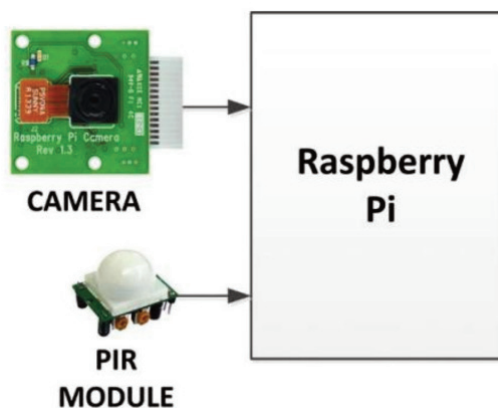
5.3 Projekat 1 – Špijunska kamera koja se aktivira na pokret

Opis: U ovom projektu dizajnirana je špijunska kamera koja se aktivira na pokret. Projekat detektuje kretanje u prostoriji i automatski snima slike svakih 15 sekundi. Program se može konfigurisati tako da se automatski pokreće nakon ponovnog pokretanja Raspberry Pi-ja. Napajanje sistema se može obezbediti iz prenosne USB baterije, jer nema potrebe za povezivanjem sistema sa računarom, monitorom ili ekranom tokom uobičajenog rada.

Cilj: Cilj ovog projekta je da dizajnira špijunsku kameru koja se aktivira na pokret pomoću PIR senzorskih modula.

Blok dijagram: Blok dijagram projekta prikazan je na slici 5.2. U ovom projektu koristi se HC-SR501 PIR senzorski modul. Ovaj modul ima sledeće specifikacije:

- Rad na 4,5 V do 20 V
- 50 μ A struje u mirovanju
- 3,3 V TTL izlaz (kada je na logičkom nivou HIGH)
- Vreme kašnjenja od 3 do 5 minuta
- Opseg osetljivosti od 3 m do 7 m
- Dva režima okidanja
- Opseg osetljivosti manji od 120 stepeni unutar 7 metara
- Rad na -15° do 70° Celzijusovih stepeni
- Vreme inicijalizacije od 1 minuta (aplikacioni sistem treba da sačeka jedan minut pre čitanja izlaza senzora da bi se izbegla lažna očitavanja)



Slika 5.2 Blok dijagram projekta

Električna šema: Na slikama 5.3 je prikazan raspored PCB kola modula senzora. Kratkospojnik na levoj strani podešava režim okidanja, potencijometar u donjem levom delu podešava osetljivost, dok potencijometar u donjem desnom delu podešava vreme kašnjenja (pogledajte sliku 5.1). Profil detekcije senzorskog modula je konus od 110 stepeni sa opsegom od 3 m do 7 m. Okretanjem potencijometra osetljivosti u smeru kretanja kazaljke na satu smanjuje se osetljivost. Kada je potpuno pravo, osetljivost je 3 metra. Slično tome, okretanje potencijometra za osetljivost u smeru suprotnom od kretanja kazaljke na satu povećava osetljivost. Kada je potpuno ulevo, osetljivost je 7 metara.

Okretanje potencijometra za kašnjenje u smeru kretanja kazaljke na satu povećava vreme kašnjenja. Kada je potpuno pravo, vreme kašnjenja je približno 5 minuta. Isto tako, okretanje potencijometra za kašnjenje u suprotnom smeru smanjuje vreme kašnjenja. Kada je potpuno ulevo, vreme kašnjenja je približno 3 sekunde.

Poglavlje 6 • Blic za Raspberry Pi kameru

6.1 Pregled

Iako je kamera Raspberry Pi kamera visoke rezolucije, teško je fotografisati u uslovima slabog osvetljenja, jer nema blica. Većina pametnih telefona ima ugrađeni blic u obliku LED-ova. Ovi LED-ovi su sinhronizovani sa zatvaračem kamere tako da bi mogli da se uključe samo kada se okidač otpusti. U ovom poglavlju ćemo dizajnirati i uobičajeni i sinhronizovani blic za našu Raspberry Pi kameru. Zainteresovani čitaoci mogu pronaći na internetu nekoliko modela blica koji su komercijalno dostupni. Jedan takav proizvod dostupan je na sledećem veb sajtu:

<https://uk.pi-supply.com/products/bright-pi-bright-white-ir-camera-light-raspberry-pi>

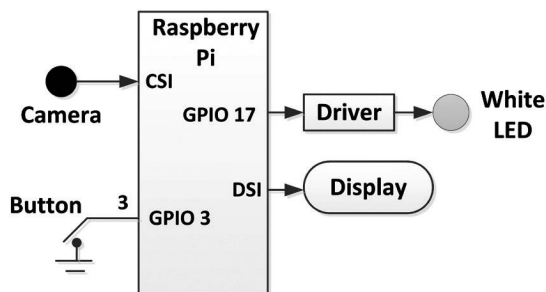
6.3 Projekat 1 – Nesinhronizovani blic

Opis: Možda je najjednostavniji blic za vašu Raspberry Pi kameru beli LED visoke osvetljenosti i uključivanje LED-a neposredno pre snimanja slike. Za primer projekta izuzetno svetli beli LED kompanije SparkFun Electronics (model br: **YSLR547V2C-A13** ili sličan) može se koristiti kao blic. Ovaj LED ima sledeće specifikacije (imajte na umu da je radi poređenja intenzitet svetlosti standardnog 5 mm crvenog LED-a 100 mcd):

- Propusna struja: 20 mA
- Obrnuta struja: 10 μ A
- Rasipanje energije: 105 mW
- Radni napon: 3,2 V
- Intenzitet svetlosti: 8000–10000 mcd

Cilj: Cilj ovog projekta je dizajniranje nesinhronizovanog blica za Raspberry Pi kameru.

Blok dijagram: Blok dijagram projekta je prikazan na slici 6.1. Električna šema se sastoji od tranzistorskog prekidača, kao što je prikazano na električnoj šemi.



Slika 6.1 Blok dijagram projekta

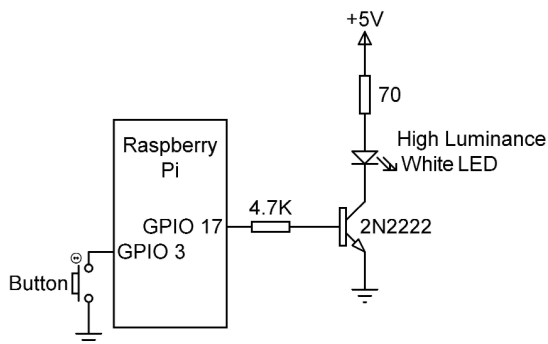
Električna šema: Budući da je napon LED napajanja 3,2 V, ima smisla pokrenuti LED od + 5 V putem, na primer, 2N2222 NPN tranzistorskog kola, kao što je prikazano na slici 6.2, gde se pretpostavlja da je LED povezan sa portom pina GPIO 17. Pod pretpostavkom da je napon napajanja od 3,2 V i tranzistor $V_{CEsat} = 0,4$ V (proverite listu podataka tranzistora), otpor opterećenja je dat formulom:

$$R_L = (5 \text{ V} - 3,2 - 0,4 \text{ V}) / 20 \text{ mA} = 70 \text{ oma}$$

Pod pretpostavkom da je $h_{FE} = 75$, $V_{BEsat} = 0,6$ V, osnovna struja je:

$I_B = I_C / h_{FE} = 20 / 75 = 0,26$ mA. Možemo uzeti $I_B = 0,5$ mA da bismo se pobrinuli da tranzistor bude u punom zasićenju. Osnovno opterećenje je dato formulom:

$R_B = (3,3 - 0,6 \text{ V}) / 0,5 \text{ mA} = 5,4$ K, možemo uzeti 4,7 K kao najbliži fizički otpornik:



Slika 6.2 Električna šema projekta

Sada možemo napisati program za uključivanje našeg LED-a neposredno pre snimanja slike. Potreban Python program (program: **camflash.py**) prikazan je na slici 6.3. Slika će biti snimljena kada se pritisne dugme. U ovom programu LED se uključuje samo jednu sekundu pre i posle snimanja slike.

```
#-----
#
#           CAMERA WITH LED FLASH LIGHT
#           =====
#
# In this program the camera is connected to the Raspberry Pi.
# In addition, a high brightness white LED is connected to GPIO
# 17 through a transistor switch. Also, a button is connected to
# GPIO 3. A picture is taken when the button is pressed.The LED
# flash light turns ON for 1 second just before and after taking
# a picture.
#
# Author: Dogan Ibrahim
```

Poglavlje 7 • Dizajniranje Raspberry Pi sistema kamera za nadzor

7.1 Pregled

U posljednjem poglavlju smo videli kako se dizajnira blic za naš modul kamere. U ovom poglavlju ćemo dizajnirati sistem kamera za nadzor. Ovaj sistem detektuje promene u snimljenim kadrovima i obaveštava ako postoji promena. Svako kretanje ili ometanje između uzastopnih kadrova je indikacija postojanja mogućeg uljeza u području prikaza.

7.2.1 Motion biblioteka

U ovom poglavlju koristićemo **motion library** (biblioteku softvera za nadzor) za naš sistem nadzora. Ova biblioteka obezbeđuje sledeće funkcije:

Gledanje videa uživo u veb pregledaču (potrebna je IP adresa Raspberry Pi-ja).

Snimanje i čuvanje videa svaki put kada se detektuje pokret u području prikaza - kreiranje videa u vremenskim razmacima.

Snimanje fotografija i njihovo čuvanje na disku kada je detektovano kretanje.

Koraci za instaliranje i upotrebu ove biblioteke su dati u nastavku.

Instalirajte motion biblioteku.

```
pi@raspberrypi:~ $ sudo apt-get install motion
```

Omogućite sistemsk uslugu (dimon) tako da se uvek pokrene. Uredite datoteku le **/etc/default/motion** pomoću **nano** uređivača i unesite liniju: **start_motion_daemon=yes**. Unesite **Ctrl+X**, a zatim **Y** da biste sačuvali i izašli iz nano uređivača.

```
pi@raspberrypi:~ $ sudo nano /etc/default/motion
```

7.2.2 Konfigurisanje motion softvera

Sada moramo da konfiguriramo motion softver pre upotrebe. Koraci su sledeći:

Kreirajte rezervnu kopiju motion konfiguracione datoteke za svaki slučaj:

```
pi@raspberrypi:~ $ sudo cp /etc/motion/motion.conf /etc/motion/motion.bak
```

Uredite konfiguracionu datoteku pomoću **nano** uređivača:

```
pi@raspberrypi:~ $ sudo nano /etc/motion/motion.conf
```

Pronađite opcije koje su prikazane u nastavku i uredite ih na način koji je prikazan (proverite dokumentaciju motion softvera za dodatne informacije o ovim i drugim opcijama):

Poništite komentar za **target_dir** i promenite u **/home/pi/motion**
Definišite datoteku evidencije: **/home/pi/motion/mypics/logfile.log**

Postavite **daemon** na **on**

Postavite **output_pictures** na **best**

Postavite **quality** na **80**

Postavite **text_changes** na **on**

Postavite **ffmpeg_output_movies** na **off**

Postavite **webcontrol_port** na **8082**

Postavite **webcontrol_localhost** na **off**

Postavite **width** i **height** na **640** i **480**, tim redom

Postavite **stream_localhost** na **off**

Postavite **framerate** na **20**

Postavite **locate_motion_mode** na **preview**

Postavite **event_gap** na **10**

Postavite **pre_capture** na **1**

Postavite **post_capture** na **1**

- Kreirajte motion fasciklu:

```
pi@raspberrypi:~ $ mkdir /home/pi/motion
```

- Aktivirajte zvanični drajver kamere (primetićete da je malo slovo “l” ispod, a ne broj “1”):

```
pi@raspberrypi:~ $ sudo modprobe bcm2835-v4l2
```

- Možete da umetnete sledeću liniju na kraju datoteke **/etc/rc.local** tako da se kamera učita automatski nakon ponovnog pokretanja:

```
pi@raspberrypi:~ $ sudo nano /etc/rc.local
```

i umetnite **sudo modprobe bcm2835-v4l2** na kraju datoteke. Unesite **Ctrl+X**, a zatim **Y** da biste izašli iz **nano** uređivača.

- Neka motion bude vlasnik fascikle u kojoj će biti uskladištene snimljene datoteke:

```
pi@raspberrypi:~ $ sudo chown motion:motion /home/pi/motion
pi@raspberrypi:~ $ sudo chown motion:motion/home/pi/motion/logfile.log
```

- Ponovo pokrenite Raspberry Pi.

Poglavlje 8 • motionEyeOS softver za nadzor

8.1 Pregled

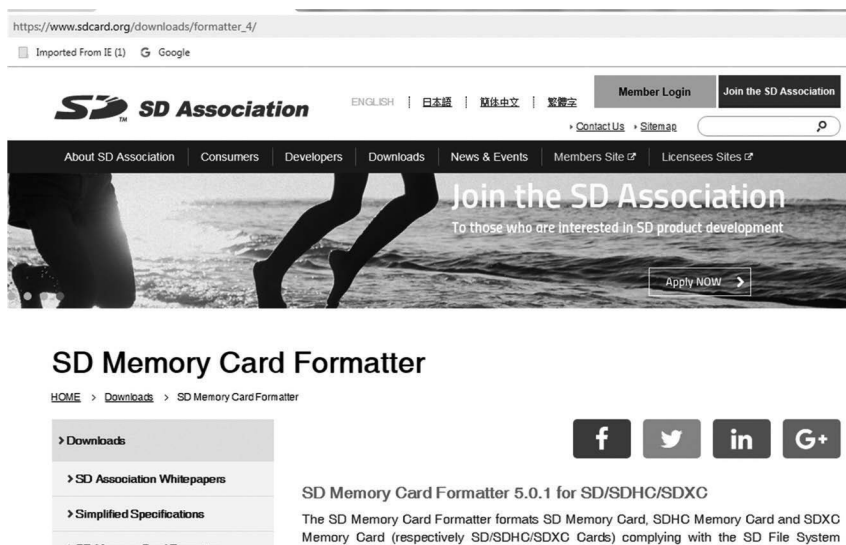
U ovom poglavlju ćemo instalirati i koristiti motionEyeOS na našem Raspberry Pi-ju 3. Neke od motionEyeOS funkcija su sledeće:

- Podrška za više kamera, uključujući i USB kamere
- Snimanje statičnih slika
- Podedešavanje rasporeda rada
- Detekcija pokreta sa obaveštenjem putem e-pošte
- Otpremanje datoteke na Google Drive ili Dropbox
- Nekoliko formata za filmove
- Podrška za snimanje filmova u vremenskim razmacima
- Podrška za Ethernet ili WiFi
- Strimovanje videa
- Datotekama se može pristupiti preko ugrađenog FTP servera (ili SFTP servera)

8.2 Instaliranje softvera motionEyeOS

motionEyeOS se instalira na novu formatiranu microSD karticu i onda se ova kartica “ubacuje” u vašu malu Pi ploču. U ovom poglavlju koristićemo microSD karticu od 16 GB, mada možete koristiti i karticu većeg kapaciteta ako želite. Koraci za instaliranje softver motionEyeOS su dati u nastavku:

- Instalirajte softver za formatiranje SD kartice sa sledećeg sajta (pogledajte sliku 8.1):
https://www.sdcard.org/downloads/formatter_4/



Slika 8.1 Instalirajte softver za formatiranje SD kartice

- Umetnite microSD karticu u računar i pokrenite aplikaciju za formatiranje SD kartice (možda ćete morati da koristite adapter da biste umetnuli microSD karticu u otvor za SD karticu standardne veličine na računaru). Izaberite karticu i formatirajte je pomoću opcije **Overwrite format** (pogledajte sliku 8.2).



Slika 8.2 Formatirajte SD karticu

- Izaberite odgovarajuću datoteku slike za vaš uređaj sa stranice softvera MotionEyeOS izdanja i sačuvajte datoteku u fasciklu:
<https://github.com/ccrisan/motioneyeos/releases>
- Ako koristite Raspberry Pi 3, u vreme pisanja ove knjige izabrana je datoteka slike:
motioneyeos-raspberrypi3-20190119.img.xz
- Sada morate ekstrahovati i učitati datoteku slike na microSD karticu. Primetićete da ne možete jednostavno ekstrahovati i kopirati datoteku slike na SD karticu, jer operativni sistem treba da kreira blok za ponovno pokretanje sistema na kartici. Za ovaj zadatak se preporučuje da koristite program **Etcher**. Program Etcher možete da instalirate na računaru sa sledećeg sajta:
<https://www.balena.io/etcher/>
- Pokrenite program Etcher (slika 8.3). Kliknite na Select image i pronađite datoteku slike koju ste ranije sačuvali.

Poglavlje 9 • OpenCV softverski paket

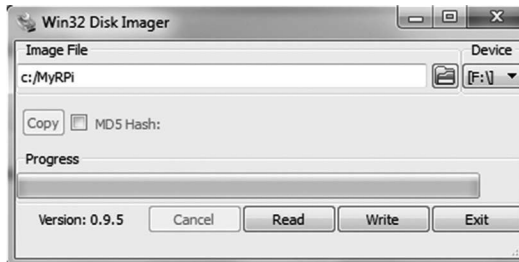
9.1 Pregled

U prethodnom poglavlju videli smo kako se instalira i koristi motionEyeOS na našem Raspberry Pi-ju 3. U ovom poglavlju ćemo videti kako se instalira OpenCV softverski paket na naš Raspberry Pi 3. OpenCV je veoma popularan softver za obradu slika koji se može koristiti za snimanje i manipulisanje slikama na našem Raspberry Pi-ju.

9.2 Kopiranje Raspberry Pi microSD kartice

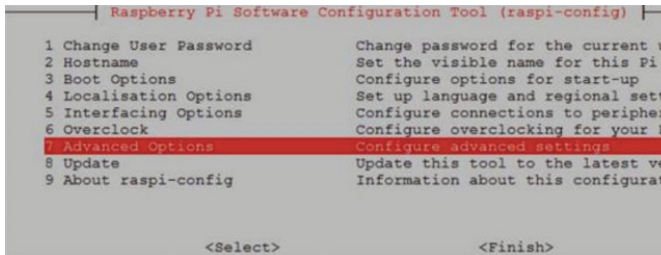
Ako je vaš Raspberry Pi operativni sistem na microSD kartici od 8 GB, možda ćete želeći da ga kopirate na veću karticu pre instaliranja i korišćenja softverskog paketa OpenCV, jer ovaj paket zahteva mnogo prostora. microSD kartica koju je autor koristio bila je samo 8 GB. Koraci za kopiranje svih datoteka i operativnog sistema na microSD karticu većeg kapaciteta su dati u nastavku. Ovde se datoteke kopiraju na microSD karticu od 16 GB:

- Uklonite napajanje iz vašeg Raspberri Py i microSD karticu.
- Umetnite microSD karticu u računar (možda ćete morati da koristite adapter, jer većina računara prihvata SD kartice standardne veličine).
- Pokrenite program **Win32DiskImager** na računaru.
- Uverite se da je microSD kartica izabrana u gornjem desnom uglu (**F:** u ovom primeru. Dodelite naziv datoteci slike koja će biti sačuvana na računaru. U ovom primeru se datoteka slike naziva **C:/MyRPinm**, kao što je prikazano na slici 9.1.
- Kliknite na **Read** da biste kopirali datoteke u **C:/MyRPI**. Sačekajte da se postupak kopiranja završi i zatvorite program **Win32DiskImager**.
- Izvadite microSD karticu iz računara i umetnite novu microSD karticu od 16 GB u računar.
- Ponovo pokrenite program **Win32DiskImager**, unesite naziv datoteke slike kao **C:/MyRPI** i kliknite na **Write** da biste kopirali sliku na novu microSD karticu. Sačekajte da se postupak kopiranja završi.
- Izvadite novu microSD karticu iz računara i umetnite je u svoj Raspberry Pi. Pri-
menite napajanje na Pi i trebalo bi da se pokrene kao i ranije.



Slika 9.1 Upotrebite program Win32DiskImager

- Otkrićete da samo kopiranje datoteke na veću microSD karticu ne povećava automatski iskoristivi prostor na disku. Ako je vaš operativni sistem instaliran bez korišćenja NOOBS-a, uradite sledeće da biste povećali iskoristivi prostor na vašoj microSD kartici:
 - Pokrenite **raspi-config** za prikaz menija za konfiguraciju i izaberite **Advanced Options** (slika 9.2).



Slika 9.2 Konfiguracioni meni

- Izaberite **Expand Filesystem** za proširenje sistema datoteka na microSD karticu:
 - Izadite iz konfiguracionog menija.
- Ako je vaš operativni sistem instaliran pomoću NOOBS-a, tada je mnogo složenije proširiti raspoloživi prostor na novoj microSD kartici. Na sledećem linku su data uputstva korak po korak za proširenje sistema datoteka na microSD kartici (upozoreni ste da su to operacije na disku niskog nivoa i ako nešto uradite pogrešno, to može učiniti vašu microSD karticu neupotrebljivom):
 - <http://dracoy.com/2016/06/26/expanding-partition-ondsd-card-for-raspberry-pi-with-noobs-pre-installed/>

Poglavlje 10 • Snimanje slika sa kamere pomoću OpenCV-a

10.1 Pregled

U poslednjem poglavlju videli smo kako se instalira softverski paket za obradu slika OpenCV na naš Raspberry Pi. U ovom poglavlju ćete naučiti kako se snimaju statične slike i video filmovi sa kamere pomoću OpenCV-a.

Kompletne detalje o biblioteci **picamera** možete dobiti na sledećem linku:

<https://media.readthedocs.org/pdf/picamera/latest/picamera.pdf>

10.2 Snimanje slika sa Raspberry Pi kamere pomoću OpenCV-a

U ovom poglavlju koristićemo biblioteku **picamera** za snimanje slika sa naše Raspberry Pi kamere. Prvo, moramo da instaliramo paket **picamera**:

```
(cv) pi @ raspberrypi: ~ $ pip install "picamera [arrai]"
```

Na slici 10.1 je prikazan program (program: **cam.py**) koji se koristi za snimanje slike sa kamere. Slika se skladišti u fasciklu **/home/pi/mypics** sa nazivom **img.jpg**.

Primitićete da se u programu koriste dve sekunde za "zagrevanje" kamere:

```
#          CAPTURE AN IMAGE
#          =====
#
# This program captures an image using the Raspberry Pi
# camera and OpenCV. The image is stored in folder
# /home/pi/mypics withh filename img.jpg
#
# File:    cam.py
# Date:    February, 2019
# Author:  Dogan Ibrahim
#-----
import cv2
from picamera import PiCamera
import time

camera = PiCamera()
camera.resolution = (1024, 768)
camera.start_preview()
print("Ready to capture an image")
time.sleep(2)
camera.capture('/home/pi/mypics/img.jpg')
print("Image has been captured...")
camera.stop_preview()
camera.close()
```

Slika 10.1 Listing programa

Možemo ponovo promeniti veličinu tokom pomoću opcije **resize**, npr.:
`camera.capture('/home/pi/mypics/img.jpg, resize=(320,240))`

Neke važne postavke kamere koje podržava OpenCV su date u nastavku:

shutter_speed: postavlja ili učitava brzinu zatvarača u mikrosekundama

exposure_speed: učitava brzinu ekspozicije kamere

exposure_mode: postavlja ili učitava režim ekspozicije kamere

Neke važeće vrednosti su sledeće:

off, auto, night, nightpreview, backlight, spotlight, sports, snow, beach etc (podrazumevano je auto)

iso: postavlja ili učitava postavku ISO za kameru

awb_mode: postavlja ili učitava režim za automatski balans boje za kameru.

Važeće vrednosti su sledeće:

off, auto, sunlight, cloudy, shade, tungsten, fluorescent, incandescent, flash, horizon (podrazumevano je auto)

awb_gains: postavlja ili učitava dobitak automatskog balansa bele boje za kameru

annotate_text: umeće tekst u gornji srednji deo slike

annotate_text_size: postavlja veličinu teksta. Podrazumevano je 32, važeće vrednosti su od 6 do 160.

brightness: postavlja ili učitava svetlinu za kameru. Važeće vrednosti su od 0 do 100: (podrazumevano je 50)

contrast: postavlja ili učitava kontrast kamere. Važeće vrednosti su od -100 do 100 (podrazumevano je 0)

flash_mode: postavlja ili učitava režim blica. Važeće vrednosti su: off, auto, on, redevy, fillin, torch (podrazumevano je off)

hflip: postavlja ili učitava da li je izlaz kamere horizontalno isečen

image_effect: postavlja ili učitava efekat slike koji je primenjen na sliku. Neke od važećih vrednosti su: none, negative, solarize, sketch, denoise, emboss itd. (podrazumevano je none)

rotation: postavlja ili učitava rotaciju kamere. Važeće vrednosti su: 0, 90, 180, 270 (podrazumevano je 0)

sharpness: postavlja ili učitava postavku oštine. Važeće vrednosti su od -100 do 100. (podrazumevano je 0)

vflip: postavlja ili učitava da li je izlaz kamere okrenut vertikalno

Poglavlje 11 • Upotreba SimpleCV-a u projektima za kameru

11.1 Pregled

U ovom poglavlju ćemo razvijati projekte zasnovane na kameri koristeći Raspberry Pi sa kamerom, koristeći popularni softver za obradu slika SimpleCV. SimpleCV je sličan OpenCV-u, ali je lakši za upotrebu i takođe nije tako moćan kao OpenCV. Prvi projekat u ovom poglavlju je čitač bar-koda. U ovom projektu se bar-kod koristi kao lozinka za aktiviranje releja - na primer, za otvaranje vrata. U drugom projektu se aktivira zujalica ako se u prostoriji otkrije kretanje.

Pre nego što razmotrimo detalje projekta za bark-kod, vredi se osvrnuti na osnovne principe različitih vrsta bar-kodova.

11.2 Bar-kodovi

Bar-kod su osmislili Joseph Woodland i Bernard Silver pedesetih godina u SAD. Bar-kodovi se trenutno koriste u supermarketima i prodavnicama kao nalepnice koje su pričvršćene ili otisnute na proizvodima da bi se lako i brzo identifikovali. U upotrebi je mnogo vrsta bar-kodova. U nastavku su ukratko opisani neki od popularnih vrsta bar-kodova:

11.2.1 Code 39

Ovo je jedan od prvih kodova razvijen 1974. godine. Ovaj kod ima mogućnost kodiranja svih 128 ASCII skupova znakova. Širina koda se može proširiti i ograničiti samo veličinom nalepnice. Nedostatak ovog koda je što nije kompaktan i samim tim može da zauzme više prostora od ostalih kodova.

11.2.2 Code 128

Ovaj kod je kompaktan i obično se koristi u aplikacijama za identifikaciju proizvoda. S obzirom da je vrlo kompaktan, Code 128 u većini aplikacija zamenjuje Code 39. Ovo je kod promenljive dužine.

11.2.3 Universal Product Code

Universal Product Code (ili skraćeno UPC) nalazi se skoro na svim prehrambenim proizvodima kao način identifikacije proizvoda. UPC je višesmerni kod od 12 znakova koji je dodelio Globalni standard za standardizaciju (GS1). Primer UPC-a je prikazan na slici 11.1. Ovi bar-kodovi se lako štampaju na standardnom visokokvalitetnom štampaču. Na slici 11.1 (kod = 725272730706), prvih 6 cifara (725272) je šifra proizvođača koja omogućava da jedinstveno kodira do milion predmeta, sledećih 5 cifara (73070) su brojevi predmeta, poslednja cifra (6) kontrolna cifra. Kontrolna cifra se izračunava na sledeći način (na primer, kao na slici 11.1):

- Sve cifre saberite u neparnim pozicijama (tj. cifre 1,3,5,7,9,11). U ovom primeru imamo: $7 + 5 + 7 + 7 + 0 + 0 = 26$

- Pomnožite broj sa 3. tj. $3 \times 26 = 78$
- Saberite sve cifre na parnim položajima (tj. cifre 2,4,6,8,10). U ovom primeru imamo: $2 + 2 + 2 + 3 + 7 = 16$
- Dodajte rezultat vrednosti koja je izračunata ranije. tj. $78 + 16 = 94$
- Uzmimo u obzir broj koji se nalazi iznad, tj. 94, tako da kada se ovom broju doda broj dobijemo množinu od 10. Najbliži broj od 10 do 90 je 100, stoga je kontrolna cifra $100 - 90 = 10$, tj. poslednja cifra na slici 11.1.



Slika 11.1 Primer UPC bar-koda

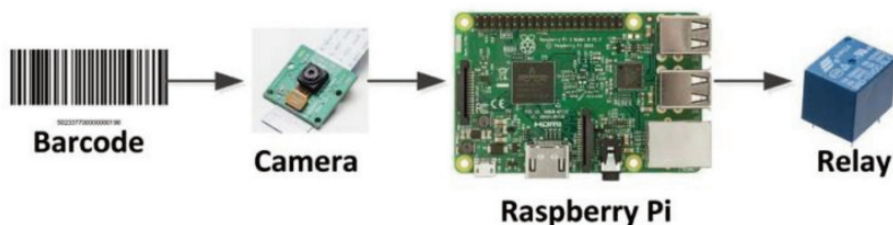
Važno je osigurati da svaki artikal u prodavnici ima jedinstveni UPC broj. Čak i različite veličine istog artikla moraju imati različit UPC kod.

11.3 Projekat 1 – Šifra bar-koda

Opis: Ovo je projekat za bar-koda koji se zasniva na šifri. Projekt čuva tajni broj bar-koda. Kad se odgovarajući bar-kod stavi ispred kamere, sistem prepoznaje ovaj kod kao ispravan i aktivira relej. Zatim, relej, na primer, otvara vrata ili izvršava neke druge radnje.

Cilj: Cilj ovog projekta je pokazati kako se bar-kodovi mogu čitati sa Raspberry Pi-ja i kamere.

Blok dijagram: Na slici 11.2 prikazan je blok dijagram projekta.



Slika 11.2 Blok dijagram projekta

Električna šema: Električna šema projekta je prikazana na slici 11.3. U ovom projektu koristi se standardna kamera Raspberry Pi-ja i povezana je sa CSI magistralom. Relej je povezan sa GPIO pinom 2 Raspberry Pi-ja.

Poglavlje 12 • Upotreba kamere pomoću Arduina

12.1 Pregled

U ovom poglavlju ćemo naučiti kako se snimaju slike pomoću kamere i razvojne ploče mikrokontrolera Arduino Uno. U stvari, Arduino Uno nije pogodan procesor za obradu slika zbog sledećih razloga:

Memorija ploče Arduino Uno je veoma ograničena.

Procesor ploče Arduino Uno nema dovoljno procesorske snage za upravljanje slikama.

Arduino Uno nema operativni sistem kao što je Linux, Windows, MAC itd, pa nije moguće instalirati paket za obradu slika OpenCV na Arduino.

Zbog pomenutih ograničenja Arduino Uno-a, snimaćemo datoteke koristeći samo odgovarajući kompatibilni modul kamere. Za čuvanje snimljenih slika koristiće se spoljna SD kartica. Slike se mogu prikazati ili obraditi umetanjem SD kartice u moćni računar poput Windows sistema, Raspberry Pi-ja ili bilo kojeg drugog procesora.

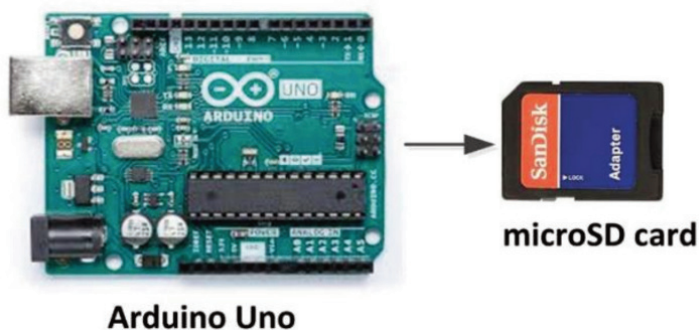
Pošto ćemo SD karticu koristiti za skladištenje snimljenih slika, korisno je naučiti kako povezati SD karticu sa Arduino Uno-m i kako kreirati datoteke na SD kartici za čitanje i upisivanje podataka. To je objašnjeno u sledećem projektu.

12.2 Projekat 1 – Interfejs Arduino Uno SD kartice: Upisivanje na karticu

Opis: U ovom projektu ćemo povezati microSD karticu sa našim Arduino Uno-m i videti kako se može kreirati nova datoteka na microSD kartici i naučiti kako se upisuje tekst u ovu datoteku. U ovom projektu tekst **My Computer** će biti sačuvan u datoteku pod nazivom **MyFile.txt** na SD kartici.

Cilj: Cilj ovog projekta je pokazati kako microSD kartica može biti povezana sa Arduino Uno-m i kako se može kreirati datoteka, a tekst sačuvati u ovu datoteku.

Blok dijagram: Na slici 12.2 prikazan je blok dijagram projekta.



Slika 12.1 Blok dijagram projekta

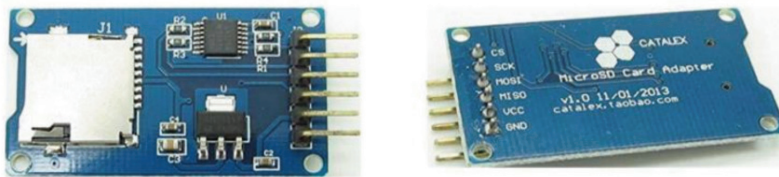
Električna šema: U ovom projektu koristi se microSD kartica, kao što je prikazano na slici 12.2 (Ako želite da koristite odgovarajuću standardnu ili mini SD karticu, električna šema i program su isti. Jedino na šta treba da obratite pažnju je da naponski nivoi adapterske kartice treba da budu kompatibilni sa Arduino nivoima signala).



Slika 12.2 microSD kartica

SD kartice koriste SPI interfejs i mogu se lako povezati sa mikrokontrolerom pomoću adapterskih čitača. Adapterski čitač koji se koristi u ovom projektu je prikazan na slici 12.3. Električna šema projekta je prikazana na slici 12.4. Sledeći pinovi se koriste za povezivanje adapter SD kartice za Arduino Uno (primetićete da su Arduino SPI pinovi porta MOSI (11), MISO (12), SCK (13)):

Pin microSD kartice	Arduino Uno pin
Vcc	5V
CS	4
MOSI	11
MISO	12
SCK	13
GND	GND



Slika 12.3 Adapterski čitač microSD kartice koji se koristi u projektu

Adapterski čitač microSD kartice koji je korišćen u ovom projektu ima sledeće specifikacije:

- Napajanje je 4,5 V ~ 5,5 V sa regulatorom napona od 3,3 V na ploči.
- Standardni SPI interfejs
- 4 M2 rupica zavrtnja za pozicioniranje radi lakše instalacije.

Poglavlje 13 • Automatsko prepoznavanje broja registarske tablice

13.1 Pregled

U ovom poglavlju ćemo videti kako prepoznati broj registarske tablice vozila (ANPR) pomoću biblioteke softvera OpenALPR sa programom Raspberry Pi. Primeri su dati i za interfejs komandne linije, ali i za upotrebu Python jezika. Iako je moguće izvršiti i prepoznavanje registarskih tablica pomoću OpenCV-a, ova metoda je visoko teorijska i složena i nećemo je razmatrati u ovoj knjizi.

13.2 Brojevi registarskih tablica vozila

Generalno, brojevi registarskih tablica vozila u različitim zemljama imaju različite oblike. Na slici 13.1 su prikazani primeri brojeva registarskih tablica u nekim zemljama. “Le-pota” biblioteke OpenALPR je u tome što se može konfigurisati za prepoznavanje registarskih tablica različitih zemalja.



Slika 13.1 Brojevi nekih registarskih tablica vozila

13.3 Biblioteka OpenALPR

Biblioteka OpenALPR je biblioteka softvera za prepoznavanje brojeva registarskih tablica otvorenog koda koja se može koristiti u kombinaciji sa nekoliko kompajlera, kao što su C #, C ++, Java, Python itd. za prepoznavanje brojeva registarskih tablica vozila. Umesto da koristi algoritme neuronske mreže, biblioteka OpenALPR koristi tehnike računarskog “vida” i zasniva se na OpenCV i Tesseract OCR bibliotekama.

13.4 Upotreba sa komandnom linijom

Biblioteka OpenALPR mora biti instalirana da bi se mogla koristiti. Sledeća komanda instalira biblioteku:

```
pi@raspberrypi:~ $ sudo apt install openalpr openalpr-daemon openalpr-  
utils libopenalpr-dev
```

13.4.1 Projekat 1 – Prepoznavanje broja registarska tablice iz datoteke

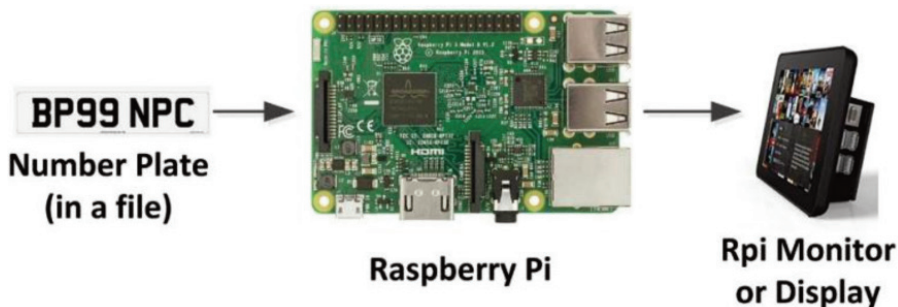
Opis: U ovom projektu ćemo videti kako se koristi biblioteka OpenALPR prepoznajući broj registarske tablice prikazana u formatu jpeg slike na slici 13.2, pod nazivom **Plate.jpg**, i kopira je u naš podrazumevani direktorijum. Ovo je primer registarske tablice za vozilo u Nemačkoj.



Slika 13.2 Primer broja registarske tablice (plate.jpg)

Cilj: Cilj ovog projekta je pokazati kako se broj registarske tablice može prepoznati sa slike.

Blok dijagram: Blok dijagram projekta je prikazan na slici 13.3.



Slika 13.3 Blok dijagram projekta

Komande: Unesite sledeću komandu da biste prepoznali registarsku tablicu. U primeru u nastavku opcija **-c** bira zemlju, gde **eu** označava Evropu:

```
pi@raspberrypi:~ $ alpr -c eu Plate.jpg
```

Koji prikazuje tekst prikazan na slici 13.4. Imajte na umu da se podrazumevano daju 10 izlaza sa različitim nivoima poverljivosti. Rezultat na vrhu liste daje se sa preko 88% poverljivosti i moramo prihvatiti ovu vrednost koja je tačan broj registarske tablice:

```
pi@raspberrypi:~ $ alpr -c eu Plate.jpg
plate0: 10 results
- HD79Z      confidence: 88.8041
- IHD79Z     confidence: 87.3512
- HD79ZI     confidence: 85.9533
- IHD79ZI    confidence: 84.5003
- 0HD79Z     confidence: 83.4587
- BHD79Z     confidence: 83.2794
- HD79ZE     confidence: 83.2735
- DHD79Z     confidence: 83.2735
- RHD79Z     confidence: 83.2013
- EHD79Z     confidence: 83.1869
pi@raspberrypi:~ $
```

Slika 13.4 Izlaz prepoznavanja broja registarske tablice

OpenALPR ima nekoliko opcija. Najčešće korišćene opcije su:

- c navodi šifru zemlje (npr. **us** za SAD, **gb** za Veliku Britaniju, **eu** za Evropu itd.)
- n maksimalni broj mogućih registarskih tablica koje se vraćaju (podrazumevano je 10)
- p pokušaj da se broj registarske tablice podudara sa obrascem registarske tablice (npr. ca za Kaliforniju)
- j rezultat prepoznavanja izlaza u JSON formatu (podrazumevano je off)
- h prikazuje informacije o korišćenju