



LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**RASPBERRY PI MIKRODATORA
IZMANTOŠANA AUTOMAŠĪNU NUMURU
ATPAZĪŠANAI UN PLŪSMAS REĢISTRĒŠANAI**

MAGISTRA DARBS

Autors: **Haralds Plass**

Stud.apl. Nr.HP13006

Darba vadītājs: profesors Dr. sc. comp. Guntis Bārzdīņš

RĪGA 2017

ANOTĀCIJA

Lai automatizētu – noprogrammētu – kādu procesu, darbinātu to, izmantojot mākoņtehnoloģijas, parasti ir nepieciešams dators – kvadrātiskas formas lieta, ko varētu izmantot programmēšanai un ko nevar nopirkt par viena pārtikas veikala apmeklējuma čeka summu, ātri ielikt kabatā un aiznest uz mājām. Tāpēc problēmjautājums ir, kā uzprogrammēt un darbināt kādu risinājumu maksimāli lēti, izmantojot kādu mobilu, mazu iekārtu, piemēram, mikro datoru?

Maģistra darba mērķis un rezultāts ir uzprogrammēts un attālināti darbināms, lēts un mobils risinājums, kas izmanto kredītkartes lieluma mikro datoru *Raspberry Pi*, kameras moduli un programmēšanas valodu *Python*, veic slēgtā teritorijā iebraucošu automašīnu fiksēšanu, numuru atpazīšanu un reģistra izveidi. Reģistrs ar automašīnu iebraukšanas laiku un numura datiem ir pieejams un klonējams no *Google Drive* ar ne vairāk kā 5 min novēlošanos.

Atslēgvārdi: vienplates dators, mikro dators, *Raspberry Pi*, *Python*, *OpenALPR*, videokamera.

ABSTRACT

The theme of master thesis: The use of microcomputer Raspberry Pi for car license plate recognition and car flow registration.

In order to automate – code – any process, run it by using cloud technologies, usually it has a need for a computer - a square thing that could be used for programming and cannot be bought by one supermarket receipt amount, quickly put in a pocket and carried home. So the challenge is how to code and run any solution the most inexpensive using a mobile, small equipment, such as a microcomputer?

The goal and the result of the Master's thesis is coded and remotely operated, cheap and mobile solution that uses a credit card sized microcomputer Raspberry Pi, camera module and the programming language Python, fixes incoming and out coming cars in a closed area, recognizes the car license plate number and creates the register. The register with the car in or out time and license plate number data is available and downloadable from Google Drive not more than 5 minutes later.

Keywords: single-board computer, microcomputer, *Raspberry Pi*, *Python*, *OpenALPR*, videocamera, *IoT*.

AUTOREFERĀTS

Maģistra darbā apskatītās problēmas (lēts un mobils teritorijas novērošanas risinājums ar automašīnu plūsmas reģistru) svarīgums ir pēc iespējas plašāk izpētīti pasaulē pieejamie mikrodatori, un novitāte – iestādes vajadzībām pielāgots – uzprogrammēts kustības sensors ar automašīnu numuru atpazīšanu un plūsmas reģistru – lietu interneta (*IoT*) sastāvdaļa.

Literatūras izpētes ietvaros ir izmantoti dažādi avoti: grāmatas, žurnāli, zinātniskie raksti, internetā piejamā informācija par mikrodatoriem un atvērtā koda risinājumi. Zinātniskie raksti un atsauces uz tiem meklētas datu bāzēs, piemēram, *ACM Digital Library*, *IEEE Xplore*, *Google Scholar*, izmantots arī atsauču pārvaldības rīks *Mendeley*.

Maģistra darba pamatā (tikai izmantoto avotu sarakstā iekļautie) ir 29% zinātniskie raksti, 19% - grāmatas, 10% - žurnāla „The MagPi” raksti un 43% - dažādi raksti internetā. Internets kā avots ir arī plaši izmantots, apkopojot datus par mikrodatoriem no to ražotāju mājas lapām. Literatūras pārskata rezultātā ir:

- 1) definēts vienplates datora jēdziens un apskatīta to rašanās vēsture, attīstība, novitātes;
- 2) apskatīti pasaulē populārākie mikrodatori, veicot to salīdzināšanu pēc dažādiem kritērijiem (cena, veiktspēja, ievadizvades saskarnes un pieslēgvietas, audiovizuālās saskarnes, izmērs, citi fiziskie un elektriskie parametri);
- 3) analizētas mikrodatoru priekšrocības un trūkumi, lai izdarītu viena mikrodatora izvēli maģistra darba risinājuma vajadzībām;

Maģistra darba praktiskās daļas ietvaros ir izveidots risinājums, kas veic slēgtā teritorijā iebraucošu automašīnu reģistrēšanu: fotografēšanu, automašīnas numura atpazīšanu, attēlu un datu sinhronizāciju ar *Google Drive*, padarot iespējamu teritorijas kontroli attālināti, izmantojot mākoņtehnoloģijas. Tādējādi ir sasniegts galvenais izvirzītais mērķis – maksimāli lēti radīt mobilu tehnoloģisku risinājumu, kas var aizvietot teritorijas sargu un nodrošināt iestādes administratīvās daļas vadītāju ar informāciju, vai teritorijā iebrauc automašīna, kurai ir izsniegta caurlaide vai kāda cita automašīna.

Maģistra darba izstrādei veltīto laiku (vairāk kā 800 stundu) var sadalīt šādi:

- 1) darba mērķa, problēmas definēšana, literatūras un atvērtā koda risinājumu izpēte (tai skaitā, atlase, analīze, tulkošana, rezultātu apkopošana) – 25%;
- 2) pasaulē pieejamo mikrodatoru analīze, risinājumam piemērotākā mikrodatora izvēle, iegāde – 20%;
- 3) tehnisko detaļu iegāde, salikšana un programmatūras instalācija – 5%;
- 4) risinājuma programmēšana, atvērtā koda komponentu pielāgošana, testēšana un ieviešana uzņēmumā – 40%;
- 5) Maģistra darba rakstīšana un noformēšana – 10%.

Darba rezultātam ir ne tikai risinājuma izveidošanas apraksta vērtība, bet arī praktisks raksturs kā lietu interneta (*IoT*) sastāvdaļai. Risinājums darbojas un ir ieviests vienas iestādes slēgtas teritorijas novērošanai, teritorijā iebraucošu un no tās izbraucošu automašīnu reģistrēšanai. Interese par šo risinājumu ir radusies arī uzņēmumiem. No Administratīvās daļas vadītāja kā lietotāja ir saņemti arī ierosinājumi risinājuma uzlabošanai un tālākam darbam.

Uz darbā izmantotiem citu autoru pētījumu rezultātiem un idejām ir norādīta atsauce. Darbs ir pārlasīts un pārbaudīts pret kontrollsarakstu. Darba kvalitāte (ievades un valodas kļūdas) ir pārbaudītas arī programmā *Microsoft Word 2010*, izmantojot *Tildes Birojs 2011* piedāvāto latviešu valodas pareizrakstības rīku.

SATURS

APZĪMĒJUMU SARAKSTS	7
IEVADS	9
1. Vienplates datori	12
1.1. Īss ieskats vēsturē	12
1.2. Pielietojums	13
1.3. Veidi	14
1.4. Priekšrocības un trūkumi	14
1.5. Vienplates datoru apskats	15
1.4.1. Salīdzinājums pēc cenas	15
1.4.2. Salīdzinājums pēc veiktspējas (procesora un atmiņas)	17
1.4.3. Salīdzinājums pēc ievadizvades saskarnēm un pieslēgvietām	18
1.4.4. Salīdzinājums pēc audiovizuālām saskarnēm	20
1.4.5. Fizisko un elektrisko parametru salīdzinājums	22
2. Kāpēc Raspberry Pi?	24
2.1. Īss ieskats vēsturē	24
2.1.1. Pirms izlaišanas	25
2.1.2. Izlaišana	25
2.1.3. Pēc izlaišanas	26
2.2. Raspberry Pi modeļi	29
2.3. Maģistra darba praktiskajā daļā izmantotais Raspberry Pi Zero	30
3. Kāpēc PYTHON?	34
3.1. Īss ieskats Python rašanās vēsturē	34
3.2. Python sintakses un semantikas īpatnības	34
3.3. Python 2 vai Python 3	35
3.4. Python filosofija	36
3.5. Kopsavilkums	37
4. Raspberry Pi programmēšana	38
4.1. Raspberry Pi Zero sagatavošana darbam	38
4.2. Tīmekļa servera un mājas lapas risinājums	42
4.3. Kustības sensors	45
4.3.1. Programmas iestatīšana darbībai fonā pēc Raspberry Pi ieslēgšanas	47
4.3.2. Attēlu sinhronizācija ar Google Drive	48
4.4. Automašīnu numuru atpazīšana	51

4.5. Automašīnu plūsmas reģistrs	52
REZULTĀTI UN SECINĀJUMI	55
IZMANTOTĀ LITERATŪRA UN AVOTI.....	58
PIELIKUMI	60
1.pielikums. Vienplates datoru salīdzinājums pēc izlaišanas gada, cenas un sistēmas mikroshēmā	60
2.pielikums. OpenALPR un saistīto bibliotēku instalācijas skripts (openalpr_install_v3.sh)	68
3.pielikums. Kustības sensora programmas kods (record.py)	71
4.pielikums. Kustības sensora konfigurācijas parametri (config.py).....	79
5.pielikums. Kustības sensora programmas palaišanas skripts (run.py).....	81
6.pielikums. Google Drive sinhronizācijas skripts attēliem (sync_pics.sh)	82
7.pielikums. Google Drive sinhronizācijas skripts datiem (sync_data.sh).....	84

APZĪMĒJUMU SARAKSTS

- AMD – Amerikas kompānija, kas ražo pusvadītājus, datoru procesorus un ar tiem saistītas tehnoloģijas (*Advanced Micro Devices*)
- API – lietojumprogrammu saskarne (*Application Programming Interface*)
- ARM – datoru procesoru instrukciju kopas arhitektūra, kas balstīta uz Lielbritānijas kompānijas ARM Holdings izstrādāto samazināto instrukciju kopuma skaitļošanas arhitektūru (*RISC - Reduced Instruction Set Computing*)
- ASV – Amerikas Savienotās Valstis
- ATX – mātesplates konfigurācijas specifikācija (*Advanced Technology eXtended*)
- CMS – satura vadības sistēma (*content management system*)
- Composite – analoga viena kanāla video pārraide (bez audio), kas pārraida standarta video, parasti, pie 480i vai 576i izšķirtspējas
- CSI – kameras seriālā saskarne (*Camera Serial Interface*)
- CPU – centrālais procesors (*central processing unit*)
- DDR – divkārsa datu ātruma sinhronā dinamiskā brīvpiekluves atmiņa (*Double data rate SDRAM*)
- DSP – digitālā signāla apstrāde (*Digital Signal Processor*)
- EEPROM – energoneatkarīga, elektriski dzēšama programmējama, tikai nolasāma atmiņa (*Electrically Erasable Programmable Read-Only Memory*)
- EPROM – pārprogrammēšana lasāmatmiņa, atmiņas tipa mikroshēma, kas saglabā informāciju, kad strāvas padeve ir izslēgta (*Erasable Programmable Read Only Memory*)
- eMMC – energoneatkarīgs, iegultās atmiņas kartes standarts (*embedded Multi Media Card*)
- Ethernet – datoru tīklu tehnoloģija (piemēram, lokāliem un lielpilsētu tīkliem)
- EUR – eiro, eirozonas oficiālā valūta
- Flash – energoneatkarīgs datu uzglabāšanas līdzeklis
- GbE – tīkls Ethernet ar datu pārraides ātrumu 1'000'000'000 biti sekundē (*Gigabit Ethernet*)
- GB – datu apjoma mērvienība (gigabaits)
- GHz – frekvences mērvienība (gigahercs)
- GPIO – universāla ievadizvade (*General purpose input/output*)
- GPS – globālā pozicionēšanas sistēma – kosmosa satelītu navigācijas sistēma, kas nodrošina atrašanās vietas un laika informāciju visos laika apstākļos, jebkurā vietā vai tuvu Zemei, kur ir brīva redzamības līnija ar četriem vai vairāk GPS pavadoņiem (*Global Positioning System*)
- GPU – grafiskais procesors (*Graphics Processing Unit*)

HDMI – augstas izšķirtspējas multivides interfeiss (*High-Definition Multimedia Interface*)

HTML – hiperteksta iezīmēšanas valoda (*HyperText Markup Language*)

IDE – integrētā izstrādes vide (*Integrated Development Environment*)

I/O – ievadizvade

IT – informācijas tehnoloģijas

IS – informācijas sistēma

KB – datu apjoma mērvienība (kilobaits)

LXDE – bezmaksas, viegla darbvirsma vide (*Lightweight X11 Desktop Environment*)

LPDDR – mazjaudas atmiņa DDR (*low-power DDR*)

LVDS – zemsprieguma diferenciāla signāls (*Low-voltage differential signaling*)

MB – datu apjoma mērvienība (megabaits)

MHz – frekvences mērvienība (megahercs)

N/A – nav attiecināms

NOOBS – jauna neiekasota programmatūra, viegli instalējama operētājsistēma (*New Out Of the Box Software*)

PCB – shēmas plate (*printed circuit board*)

PVN – pievienotās vērtības nodoklis

RAID – neatkarīgu disku redundants kārtējums, datu uzglabāšanas virtualizācijas tehnoloģija (*Redundant Array of Inexpensive Disks*)

RAM – brīvpieejas atmiņa (*random-access memory*)

RISC OS Open Ltd. – IT konsultāciju un programmatūras sabiedrība ar ierobežotu atbildību

SATA – seriālā datoru kopnes saskarne, kas savieno resursdatora kopnes adapterus ar lielapjoma atmiņas ierīcēm, piemēram, ar cietajiem diskiem un optisko disku iekārtām (*Serial ATA*)

SD – droša digitālā karte, energoneatkarīga zibatmiņas karte (Secure Digital)

SRAM – statiskā brīvpieejas atmiņa (*static random-access memory*)

QEMU – bezmaksas un atvērta koda pārraugs, kas izpilda aparatūras virtualizāciju (*Quick Emulator*)

USB – universālā seriālā kopne (*Universal Serial Bus*)

USD – Amerikas Savienoto Valstu dolārs, valūtas vienība (*United States dollar*)

VIA – Taivānas kompānija, kas ražo integrētās shēmas (arī procesorus)

Wi-Fi – lokālā bezvadu tehnoloģija

? – informācija nav pieejama

IEVADS

Lai automatizētu – noprogrammētu – kādu procesu un pēc tam darbinātu to, izmantojot mākoņtehnoloģijas, parasti ir nepieciešams dators – kvadrātiskas formas lieta, ko varētu izmantot programmēšanai un ko nevar nopirkt par viena pārtikas veikala apmeklējuma čeka summu, ātri ielikt kabatā un aiznest uz mājām. Šajā maģistra darbā ir risināts problēmjautājums, kā uzprogrammēt un darbināt kādu risinājumu maksimāli lēti, izmantojot kādu mobilu, mazu iekārtu, piemēram, mikrodatoru.

Mūsdienās datori vairs nav tikai kvadrātiskas formas lietas, ko izmanto darbam. Dators ir viss, kas satur programmējamu datu apstrādes bloku, piemēram, spēļu konsole, viedtālrunis, *GPS* iekārta, planšetdators un cita veida ierīces. Tās ir lietas, ko mēs izmantojam darbā, komunikācijai, relaksācijai un citiem mērķiem. Šīs iekārtas seko vienkāršam komandu sarakstam vai instrukcijai soli pa solim, ko sauc par programmu.

Raspberry Pi ir viena no iekārtām, kas ir pietiekami lēta un vienkārša, lai varētu nodrošināt automašīnas fiksēšanu, attēla atpazīšanu, lai iegūtu datus par automašīnas numuru tā automātiskai ievadei datu bāzē, tādējādi nodrošinot slēgtas teritorijas kontroli un ne tikai. *Raspberry Pi* faktiski ir izejmateriāls – plate, kurā var iestrādāt dažādas programmas un pēc tam tās izmantot citās iekārtās. *Raspberry Pi* lētums izpaužas arī tajā, ka iekārta izmanto *Linux*, kas ir atvērtā koda operētājsistēma, un uz tā ir iespējams izmantot vienkāršu programmēšanas valodu *Python*, lai uzprogrammētu dažnedažādus risinājumus. Tā kā programmatūra (nepieciešamība iegādāties licenci) ir viena no komponentēm, kas sadārdzina programmēšanas prieku un datora izmantošanu vispār, tad bezmaksas atvērtā koda operētājsistēmas *Linux* izmantošana izskaidro *Raspberry Pi* lētumu.

Maģistra darba uzdevums ir, izmantojot vienplates datoru un programmēšanas valodu *Python*, demonstrēt lētas, mazas un vienkāršas iekārtas tehnoloģiskās iespējas, piemēram, izveidot mobilu risinājumu – uzprogrammēt 1) kustības sensoru, kas darbojas, analizējot kameras uzņemtus attēlus, 2) automašīnu numuru atpazīšanas risinājumu un 3) automašīnu plūsmas reģistru, kas glabājas *Google Drive*.

Maģistra darba mērķis ir pierādīt *Raspberry Pi* – lētās un vienkāršās iekārtas – tehnoloģiskās iespējas: uzprogrammēt kustības sensoru, kas analizē attēlus, veic attēlu atpazīšanu, reģistrē teritorijā iebraucšanas automašīnas laiku un atpazītā automašīnas numura datus reģistrā. Izvirzītā mērķa motivācija ir saistīta ar vienas iestādes administratīvās daļas vadītāja pasūtījumu radīt tehnoloģisku aizvietāju teritorijas sargam, kas nodrošinās viņu ar informāciju, vai slēgtajā iestādes teritorijā iebrauc automašīna, kurai ir izsniegta caurlaide, vai kāda cita.

Vienlaicīgi maģistra darbs demonstrē risinājumu problēmai, kā apgūt programmēšanu lēti, ātri un vienkārši, izmantojot kādu mobilu, mazu iekārtu – vienplates datoru – mikrodatu. Pasaulē ir vairāki vienplates datoru ražotāji un arī uzdevumu klāsts, ko varētu risināt ar vienplates datoru, ir plašs. Tāpēc maģistra darbam ir izvirzīti šādi apakšuzdevumi:

- ✓ apskatīt pasaulē pieejamos populārākos vienplates datorus - mikrodatu;
- ✓ iegādāties un sagatavot darbam vienu no vislētākiem mikrodatu (*Raspberry Pi*);
- ✓ izvēlēties vienkāršu programmēšanas valodu tālākam darbam, salīdzinot *Python* ar *Visual Basic*;
- ✓ izmantojot valodu *Python*, uzprogrammēt tīmekļa serveri un vienkāršu mājas lapu;
- ✓ palaist darbībā tīmekļa lapu, testējot no dažādām ierīcēm (datu, planšetdatu, viedtālruni);
- ✓ iegādāties, sagatavot darbam videokameru un darbināt to, izmantojot valodu *Python*;
- ✓ uzprogrammēt kustības sensoru, kas analizē attēlus, ko fiksē *Raspberry Pi* kamera;
- ✓ uzinstalēt automašīnu numuru atpazīšanas bibliotēku *OpenALPR* un ar to saistītās bibliotēkas (*leptonica*, *opencv*, *tessdata*, *tesseract* u.c.), pielāgot tās Latvijas automašīnu numuru atpazīšanai;
- ✓ uzrakstīt *Bash* skriptus, kas veic automašīnas numura atpazīšanu un ierakstīšanu teksta datnē CSV kopā ar laiku, kā arī datu un attēlu sinhronizāciju ar *Google Drive*;
- ✓ testēt risinājumu slēgtā teritorijā, kur automašīnu plūsma ir neliela, kā arī uz ielas, kur ir intensīvāka satiksme.

Izvirzītā maģistra darba mērķa sasniegšanai izmantota izpētes, analīzes, programmēšanas un testēšanas metode.

Šajā maģistra darbā termins ‘mikrodatu’ ir lietots kā sinonīms terminam ‘zemo izmaksu datu’ un ‘vienplates datu’, lai gan terminoloģiski pareizāks nosaukums ir ‘vienplates datu’ saskaņā ar Latvijas Zinātņu akadēmijas Terminoloģijas komisijas veidoto akadēmisko terminu datubāzi AkadTerm¹ [12]. Ārvalstu literatūrā vienplates datu apzīmēšanai bieži tiek lietots termins ‘zemo izmaksu datu’ [13] un ‘vienplates datu’ [17].

Maģistra darbu veido teorētiskā un praktiskā daļa. Teorētiskajā daļā ir apskatīti pasaulē pieejamie populārākie vienplates datu, veicot to salīdzinājumu pēc cenas, veiktspējas, pieslēgvietām, audiovizuālām saskarnēm, fiziskiem un elektriskiem parametriem. Tāpat teorētiskajā daļā ir apskatīta vienplates datu rašanās vēsture, *Raspberry Pi* arhitektūra un tehniskās iespējas, kā arī programmēšanas valoda *Python*, atbildot uz jautājumu, kāpēc maģistra darba ietvaros ir izvēlēta tieši programmēšanas valoda *Python*, nevis cita. Praktiskā daļa ir veltīta maģistra darbā definētā risinājuma programmēšanai, testēšanai un darbināšanai.

¹ <http://termini.lza.lv/term.php?term=single%20board%20computer&list=&lang=EN&h=yes>

Šo risinājumu ir plānots veidot no programmēšanas valodā *Python* programmēta tīmekļa servera un mājas lapas, kas ik pēc 5 sekunžu intervāla attēlo aktuālāko *Raspberry Pi* kameras uzņemtu attēlu, no kustības sensora, kas analizē kameras uzņemto attēlu, ģenerējot jaunus attēlus, ja ir notikusi kustība. Automašīnu numuru atpazīšanas, uzņemto attēlu un datu sinhronizācijas ar *Google Drive* risinājumu ir plānots realizēts, izmantojot *Bash* skriptus. Nobeigums un secinājumi satur darba galvenos rezultātus, secinājumus un priekšlikumus. Darbā izmantotās literatūras un avotu sarakstu satur pēdējā nodaļa.

1. Vienplates datori

Vienplates dators ir vienkāršs dators, kas ir izstrādāts uz vienas shēmas plates. Tas ir pilnvērtīgs dators tādā nozīmē, ka ietver visas funkcijas, kuras parasti ir raksturīgas pilnīgam datoram: ievadizvades (I/O), datu apstrādes iespējas, mikroprocesoru(rus), atmiņu un spēju uzturēt programmas izpildi [6]. Vienplates datori tika radīti demonstrēšanas, iegulto sistēmu, izglītības sistēmu vajadzībām vai izmantošanai kā iestrādātiem datoru kontrolieriem. Dažāda veida mājas datoros vai portatīvajos datoros visas funkcijas ir integrētas vienā iespiedshēmas platē. Atšķirībā no personāliem datoriem, vienplates datori visbiežāk nepiedāvā paplašināšanas slotus perifērijas funkcijām vai paplašināšanas iespējas. Daži vienplates datori ir veidoti tā, lai tos varētu iespraust aizmugures platē sistēmas paplašināšanas nolūkā. Vienplates datori ir veidoti, izmantojot plašu klāstu mikroprocesoru. Vienkāršāki modeļi, piemēram, izstrādāti datoru hobijiem, bieži izmanto statisko *RAM* un zemu izmaksu astoņu vai 16 bitu procesorus. Citi veidi, piemēram, asmens serveri (*blade servers*), ietver visu servera datora atmiņas un procesora veiktspēju kompaktā formātā [17]. Vienplates datorus dēvē arī par mikrodatoriem – ļoti maziem vienplates datoriem, kas izmēros salīdzināmi ar kredītkartes lielumu un patērē relatīvi maz enerģijas. Tiem parasti nav arī ventilatora un citas sastāvdaļas. Tie var darbināt jebkuru normālu programmatūru, kas ir uzrakstīta atbilstoši to arhitektūrai (parasti *ARM*), un tiem nav arī augstu prasību attiecībā pret resursiem. Tie visbiežāk ir konstruēti kā "sistēma mikroshēmā", un tiem parasti ir daudz ievadizvades savienotāju, kuri atbalsta arī *GPIO* tapas, lai varētu pievienot pielāgotu aparatūru [1].

1.1. Īss ieskats vēsturē

Amerikāņu žurnāla „Radio-Electronics” 1976.gada maija numurā pieminēts pirmais, īstais vienplates dators "*dyna-micro*", kas bija balstīts uz *Intel* ražoto un 1974.gada aprīlī izlaisto 8 bitu mikroprocesoru *C8080A*, izmantoja arī *Intel* pirmo *EPROM* (*erasable programmable read only memory*) *C1702A* – atmiņas mikroshēmu, kas saglabāja datus, ja strāvas padeve bija atslēgta. 1976.gadā „*Dyna-micro*” atkārtoti pārstrādāja un nosauca par "*MMD-1*" (*Mini-Micro Designer 1*), kas bija plašāk pazīstams kā parauga mikrodators populārajās 8080 "*BugBook*" laika sērijās [17].

Vienplates datori ir pieminēti arī mājas datoru agrīnā vēsturē, piemēram, *BBC Micro* un *Acorn Electron*, kas ir kompānijas „*Arcon Computers*” radīta mājas datora *BBC Micro* budžeta versija ar 32 kilobaitiem RAM un ROM ar *BBC BASIC V2* programmēšanas valodu un operētājsistēmu. Citi tipiskie senie vienplates datori, piemēram, *KIM-1*, bieži tika piegādāti bez korpusa, ko savukārt pievienoja saimnieks. Citi piemēri ir *Ferguson Big Board* un *Nascom* [17].

Līdz ar datoru attīstību notika strauja pāreja no vienplates datoriem uz datoriem, konstruētiem no mātesplates ar tādām meitasplates nodrošinātām funkcijām kā, piemēram, seriālie porti, diskdziņa kontrolieris un grafika. Nesenā modernāku mikroshēmu komplektu pieejamība nodrošina lielāko daļu ievadizvades funkciju kā iegultās komponentes, kas mātesplates ražotājiem ļauj piedāvāt mātesplates ar jau iebūvētu ievadizvades funkcionalitāti, kas tradicionāli bija meitas plašu pārziņā. Lielākā daļa personālo datoru mātesplašu tagad piedāvā uz plates iebūvētu diskdziņu atbalstu, tostarp *IDE* un *SATA* ar *RAID*, grafiku, *Ethernet* un tradicionālo ievadizvades funkcionalitāti, piemēram, seriālos un paralēlos portus, USB un tastatūras / peles atbalstu. Iespraužamās kartes tagad biežāk ir augstas veiktspējas grafikas kartes (īsti grafikas līdzprocesori), augstas klases RAID kontrolieri un specializētās ievadizvades kartes, piemēram, datu ieguves un DSP (*Digital Signal Processor*) plates [17].

1.2. Pielietojums

Vienplates datoru ražošana kļuva iespējama, palielinot integrēto shēmu blīvumu. Vienplates konfigurācija samazina sistēmas kopējās izmaksas, samazinot nepieciešamo shēmas plašu skaitu, kā arī izslēdzot savienotājus un kopnes dziņu shēmas, kas citādi tiktu izmantotas. Saliekot visas funkcijas uz vienas plates, var iegūt mazāku vispārējo sistēmu, piemēram, līdzīgi kā piezīmjdatoros. Savienotāji bieži ir drošības problēmu avots, tāpēc vienplates sistēma novērš šīs problēmas [17].

Vienplates datorus vispārīgi definē pēc divām noteiktām arhitektūrām: bez izvēršanas slotiem un ar slotu atbalstu. Iegultie vienplates datori ir vienības, kas sniedz visu nepieciešamo ievadizvadi un neparedz spraudņu kartes. Tos parasti izmanto spēlēm (spēļu automāti, video pokeri), kioskos un mašīnu kontrolei. Iegultie vienplates datori ir daudz mazāki nekā datoros atrodamās *ATX* tipa mātesplates, un tie nodrošina ievadizvades sajaukumu industriālām lietotnēm, piemēram, iebūvētai digitālai un analogai ievadizvadei, iebūvētai sāknējamai zibatmiņai (novēršot nepieciešamību pēc diska dziņa), ne video, u.c. Termins "vienplates dators" vispārīgi tiek attiecināts arī uz arhitektūru, kur vienplates dators ir pieslēgts aizmugures platei, lai nodrošinātu ievadizvades kartes. Attiecībā uz *PC104*, kopne nav aizmugures plate tradicionālajā nozīmē, bet gan virkne kontakta tapu, kas ļauj pieslēgt ievadizvades plates [17].

Vienplates datorus visbiežāk izmanto industriālās situācijās, kad tos izmanto statnes (*rackmount*) formātā procesa kontrolei vai iestrādā citās ierīcēs, lai nodrošinātu kontroli un saskarnes. Pateicoties ļoti augstam integrācijas, samazinātam sastāvdaļu skaitam un samazinātam savienotāju skaitam līmenim, vienplates dators ir mazāks, vieglāks, enerģijas efektīvāks un uzticamāks nekā vairāk-plašu dators. Vienplates datora galvenā priekšrocība, salīdzinot ar *ATX* mātesplates datoriem, ir izmaksas. Mātesplates ražo miljoniem attiecībā uz

patērētāju un biroja tirgiem. Savukārt vienplates datori ir specializēta tirgus niša. Tie tiek ražoti daudz mazāku skaitu ar izrietoši lielākām izmaksām. Mātesplates un vienplates datori piedāvā līdzīgu līmeni funkciju integrācijas nozīmē. Mātesplates atteice vai standarta maiņa izraisīs līdzvērtīgu aizstājēju [17].

Vienplates datorus literatūrā dēvē arī par zemo izmaksu datoriem [13]. To funkcionālās iespējas svārstās no tīmekļa pārlūkošanas un biroja programmu atbalsta līdz pilntiesīga servera funkcionalitātei. Jāatzīmē, ka, pateicoties tehnoloģiju straujajai attīstībai, zemo izmaksu datori šodien spēj daudz vairāk nekā jaudīgi serveri vakardien (pirms vairākiem gadiem). No otras puses, zemo izmaksu datora pamatā ir operētājsistēma *Linux* [15], lai pieņemtu aparatūru un nodrošinātu pamata funkcionalitāti par minimālām izmaksām [13].

Zemo izmaksu datoru izmantošanu pamatā var iedalīt šādās kategorijās:

- ✓ izglītībai;
- ✓ lietošanai birojā (neintelektisko termināļu aizvietotājs);
- ✓ plāno klientu aizvietotājs;
- ✓ zemo cenu serveri;
- ✓ ātri izvietošanas eksrompta sistēmas [13].

Šo uzskaitījumu, protams, var papildināt arī ar citiem mērķiem (eksperimentāliem, hakera darbībām, mājas automatizācijas vajadzībām u.c.) atkarībā no lietotāja nolūkiem.

1.3. Veidi

Zemo izmaksu datoru veidus var klasificēt, balstoties uz procesoru (CPU) un formas faktoru. Ņemot vērā tehnoloģiju attīstību un piesātināto tirgu, skaidra un izsmeļoša vienplates datoru definīcija un iedalījums nav iespējama. Toties aptuvenus, tuvināts iedalījums var viest skaidrību un noderēt, vadoties pēc darba veida, kam to var izmantot, un sistēmas veiktspējas.

Vienplates datorus var iedalīt pēc procesora, sistēmas, kas darbojas uz:

- ✓ ARM bāzes CPU
- ✓ Intel bāzes CPU
- ✓ VIA bāzes CPU
- ✓ AMD bāzes CPU [13]

1.4. Priekšrocības un trūkumi

Zemo izmaksu datoriem ir vairākas priekšrocības un trūkumi. Lai sniegtu īsu pārskatu un aptuvenu skaidrojumu, ko var sagaidīt un ar ko jāreķinās, investējot zemo izmaksu datorā, galvenie desmit kritēriji ir apkopoti priekšrocību un trūkumu analīzes matricā (sk. 1.1. tab.).

Vienplates datoru priekšrocības un trūkumi [13]

Priekšrocības	Trūkumi
Izmaksu faktori [21, 11, 19]	Atjaunināšana un paplašināšana ir jautājums
Formas faktors parasti ir mazs [19]	Ierobežots ārējo perifērijas ierīču un paplašināšanas karšu atbalsts
Piemērots teksta apstrādes lietojumprogrammai	Tas nav modernu lietojumprogrammu dzinējspēks
Parasti <i>Linux</i> bāzēts [15]	Ir jautājums cilvēkiem, kuri izmanto <i>Windows</i>
Lielisks resurss hobijs	Nav piemērots apjomīgām lietojumprogrammām
Zems enerģijas patēriņš [4, 19]	Veiktspēja ir kritiska
Jaunu biznesu attīstība	Joprojām neskaidrs tirgus
Parasti iztiek bez dzesēšanas	Ierobežota veiktspēja
Pārdod tikai kā plati	Ir nepieciešami papildus piederumi (aksesuāri)
Plašas iespējas	Nav piemērots grafiskām/resursus prasošām lietojumprogrammām

1.5. Vienplates datoru apskats

Pasaulē ir pieejami daudz un dažādi vienplates datori (sk. 1.pielikumā). Tos var atšķirt pēc cenas, izmēra un tehniskiem parametriem. Daži no tiem piedāvā labāku veiktspēju un vairāk atmiņas nekā citi. Dažiem no tiem var būt ļoti dažādi savienotāji, kamēr citi piedāvā tikai nepieciešamo minimumu. Daļai no šīm ierīcēm var pievienot pielāgotu aparatūru, izmantojot universālas kontaktplāksnes, kamēr citas ir vairāk integrētas un mazāk pielāgojamas. Lielākā daļa no vienplates datoriem ir balstīti uz *ARM* arhitektūru, kas ierobežo to lietošanu operētājsistēmām, piemēram, aprobežojoties ar *Linux*, *Android*, kaut gan daži mikrodatori spēj pārsteigt arī ar *x86* un uz tiem izdodas pat palaist *Windows* [5]. Lai gan parasti tie ir mazi, starp tiem joprojām ir nozīmīgas atšķirības izmēra ziņā. Daži no tiem ir paredzēti lietotājiem mājās, kamēr citi ir projektēti hakeriem un datoru ekspertiem. Visbeidzot, bet ne mazsvarīgāk, cena mikrodatoriem var atšķirties ievērojami. Tāpēc, piemēram, pirms pirkuma izvēles izdarīšanas, nozīmīgs ir tirgus apskats un populārāko mikrodatoru salīdzinājums dažādās kategorijās, protams, pievēršot uzmanību arī tehniskiem parametriem, vadoties pēc tā, kādam nolūkam vienplates dators tiks iegādāts un kādu mērķi ar to plānots sasniegt.

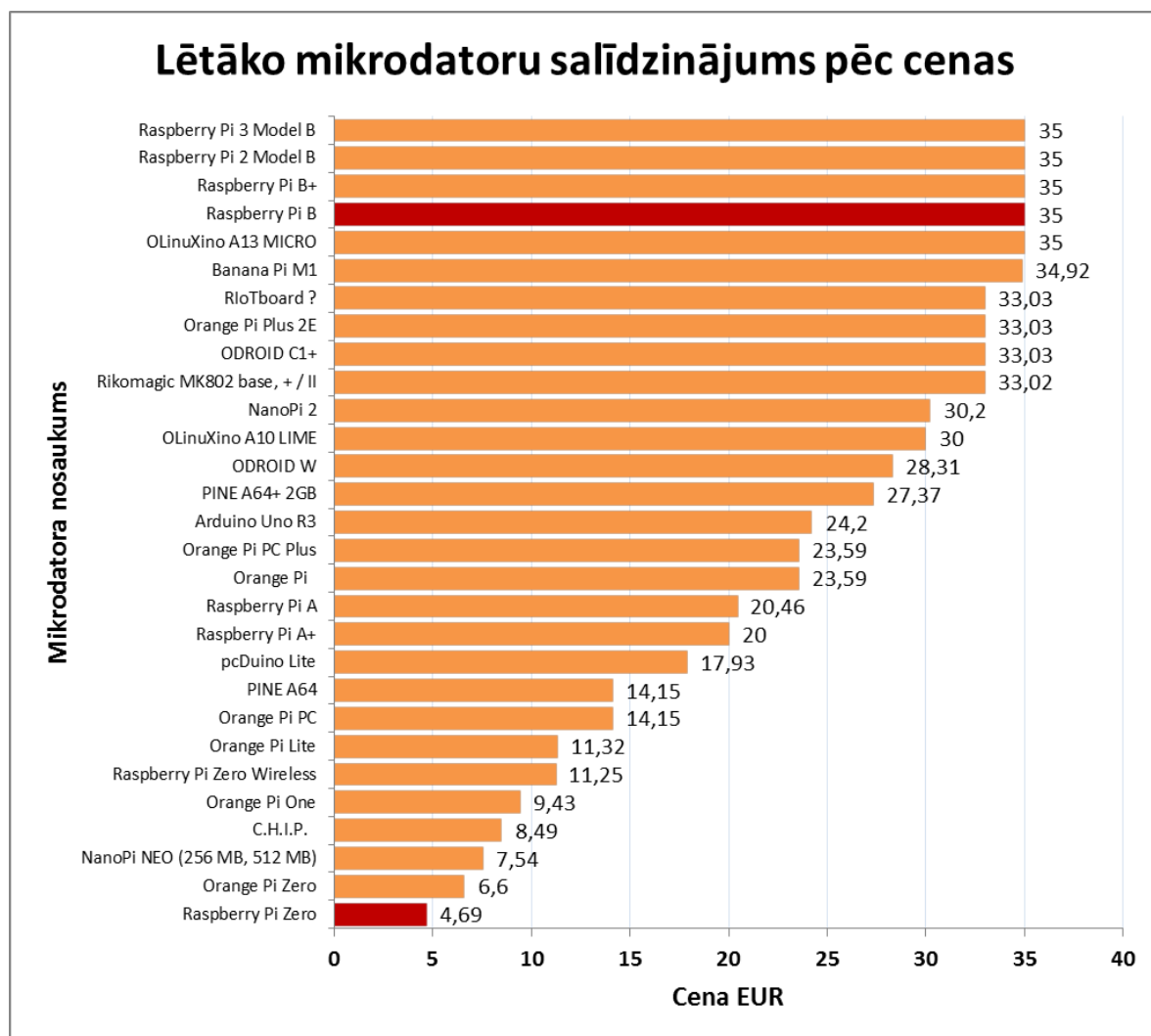
1.4.1. Salīdzinājums pēc cenas

Ņemot vērā, ka maģistra darba mērķis ir demonstrēt automašīnu plūsmas reģistra risinājumu, izmantojot maksimāli lētu vienplates datoru, tad, nepretendējot uz pilnīgi izsmeļošu pasaulē pieejamo mikrodatoru uzskaitījumu, kā pirmais salīdzinājuma kritērijs ir izvirzīta cena, ieviešot piecas kategorijas šādiem cenu diapazoniem:

1. kategorija – viszemākā cena (0-35 EUR) - XS
2. kategorija – zema cena (36-50 EUR) - S
3. kategorija – vidēja cena (51-100 EUR) - M
4. kategorija – augsta cena (101-150 EUR) - L

5. kategorija – visaugstākā cena (151 EUR un vairāk) - XL

Cenu diapazona kategorija ir noteikta un norādīta 1.pielikumā iekļautās tabulas 4. ailē (sk. 1.pielikumā), kur apzīmējums „?” nozīmē, ka attiecīgo informāciju neizdodas noskaidrot. Cenu avots ir ražotāju mājas lapas tīmeklī un interneta veikali, kas apskatītas laika posmā no 2016.gada decembra līdz 2017.gada februārim. Tātad, pirmkārt, tās atsevišķos gadījumos ir cenas bez PVN, un, otrkārt, 1. kategorijā klasificēto vienplates datoru cena, veicot to iegādi kādā interneta veikalā, ar PVN var pārsniegt 35 EUR robežu. Arī tad, ja mikrodatu izdodas iegādāties eBay vai kādā ASV interneta veikalā, saņemot precīzi Latvijā, jāreķinās ar muitas nodokli (PVN). Ņemot vērā, ka lielākā daļa mikrodatu cenu ražotāju mājas lapās ir pieejamas USD valūtā, taču valūtas kurss USD/EUR pēc Latvijas Bankas noteiktā valūtas kursa 28.02.2017 ir 1,05970², proti, ASV dolāra valūtas kurss ir gandrīz līdzvērtīgs eiro (vai drīz tāds būs), tad cenu diapazonos norādītās cenas nosacīti var uzskatīt kā eiro cenas (1:1), nepilnos 6 centus norakstot uz vienplates datora piegādes un pieejamības nodrošināšanas Eiropā izmaksām.



1.1. att. Lētāko vienplates datoru salīdzinājums pēc cenas (~EUR)

² www.bank.lv/statistika/datu-telpa/galvenie-raditaji/valutu-kursi#Cits datums

Detalizētāk apskatot lētākos, pirmās cenu diapazona kategorijas mikroprocesorus, atrodot jau konkrētu vienplātes datora iegādes vietu (interneta veikalu) un pieskaitot PVN, ir uzskatāmāk redzams līderis – lētākais mikroprocesors – *Raspberry Pi Zero* (sk. 1.1.att.). Tomēr lētākais vēl nenozīmē labākais un tālākie salīdzinājumi tiek veikti, lai noskaidrotu tieši maģistra darbā izvirzīto uzdevumu izpildei piemērotāko mikroprocesoru, maksimāli ievērojot nosacījumu „vislētākais risinājums”.

1.4.2. Salīdzinājums pēc veikspējas (procesora un atmiņas)

Ievērojot maģistra darba uzdevumu demonstrēt noderīgu risinājumu – automašīnu kustības sensoru ar numuru atpazīšanu – uz lēta vienplātes datora, tālāk tiek apskatīti tikai 1.pielikumā iekļautajā tabulā redzamie pirmā cenu diapazona (*līdz 35 EUR*) un 1.1. attēlā redzamie lētākie vienplātes datori, veicot to salīdzinājumu pēc procesora, grafiskā procesora un atmiņas (sk. 1.2. tab.).

1.2. tabula

Lētāko vienplātes datoru salīdzinājums pēc procesora un atmiņas

Nosaukums	CPU			GPU	RAM	
	Arhitektūra	Ko doli	Frekvence		Apjoms	Veids
Raspberry Pi Zero v1.3	ARM11	1	1 GHz	Broadcom VideoCore IV @ 250 MHz	512 MB	LPDDR2 SDRAM
Orange Pi Zero	H2 Quad-core Cortex-A7	4	1 GHz	Mali400MP2 GPU	256 MB	DDR3 SDRAM
NanoPi NEO	Allwinner H3, Quad-core Cortex-A7	4	1 GHz	Mali-400MP GPU	256 MB	DDR3 SDRAM
C.H.I.P.	ARMv7 Allwinner R8	1	1 GHz	Mali-400MP GPU	512 MB	SDRAM
Orange Pi One	H2 Quad-core Cortex-A7	4	1 GHz	Mali400MP2 GPU	512 MB	DDR3 SDRAM
Raspberry Pi Zero Wireless	ARM11	1	1 GHz	Broadcom VideoCore IV @ 250 MHz	512 MB	LPDDR2 SDRAM
Orange Pi Lite	H3 Quad-core Cortex-A7	4	1 GHz	Mali400MP2 GPU	256 MB	DDR3 SDRAM
Orange Pi PC	H3 Quad-core Cortex-A7	4	1 GHz	Mali400MP2 GPU	1 GB	DDR3 SDRAM
PINE A64	ARM Cortex A53	4	1.2 GHz	Dual Core Mali 400 MP2	512 MB	DDR3 SDRAM
pcDuino Lite	ARM Cortex-A8	1	1 GHz	Mali-400	512 MB	DRAM
Raspberry Pi A+	ARM11	1	700 MHz	Broadcom VideoCore IV @ 250 MHz	512 MB	LPDDR2 SDRAM
Raspberry Pi A	ARM11	1	700 MHz	Broadcom VideoCore IV @ 250 MHz	256 MB	LPDDR2 SDRAM
Orange Pi 2	H2 Quad-core Cortex-A7	4	1 GHz	Mali400MP2 GPU	1 GB	DDR3 SDRAM
Orange Pi PC Plus	H3 Quad-core Cortex-A7	4	1 GHz	Mali400MP2 GPU	1 GB	DDR3 SDRAM
Arduino Uno R3	ATmega328P	4	16 MHz	?	2 KB	SRAM
PINE A64+	ARM Cortex A53	4	1.2 GHz	Mali400MP2 GPU	2 GB	DDR3 SDRAM
ODROID W	ARM11	1	700 MHz	Broadcom VideoCore IV	512 MB	DRAM LPDDR2
OLinuXino A10 LIME	ARM Cortex-A8	1	1 GHz	Mali-400	512 MB	DDR3

NanoPi 2	Quad Core Cortex-A9	4	1.4 GHz	?	1 GB	DDR3
Rikomagic MK802 base, + / II	ARM Cortex-A8	1	1 GHz	MALI-400	1 GB	DDR3
ODROID C1+	ARM Cortex-A5	4	1.5 GHz	Dual Core Mali™-450 GPU	1 GB	DDR3
Orange Pi Plus 2E	H3 Quad-core Cortex-A7	4	1 GHz	Mali400MP2 GPU	2 GB	DDR3 SDRAM
RIoTboard	ARM Cortex-A9	1	1 GHz	Vivante GC880 + GC320	1 GB	DDR3 @ 800MHz
Banana Pi M1	ARM Cortex-A7	2	1 GHz	Mali-400MP2	1 GB	DDR3
OLinuXino A13 MICRO	ARM Cortex-A8	1	1 GHz	3D Mali400 GPU	256 MB	DDR3
Raspberry Pi B	ARM11	1	700 MHz	Broadcom VideoCore IV @ 250 MHz	512 MB	LPDDR2 SDRAM
Raspberry Pi B+	ARM11	1	700 MHz	Broadcom VideoCore IV @ 250 MHz	512 MB	LPDDR2 SDRAM
Raspberry Pi 2 Model B	ARM Cortex-A7	4	900 MHz	Broadcom VideoCore IV @ 250 MHz	1 GB	LPDDR2 SDRAM
Raspberry Pi 3 Model B	ARM Cortex-A53	4	1.2GHz	Broadcom VideoCore IV @ 250 MHz	1 GB 900 MHz	LPDDR2 SDRAM

Kā redzams no lētāko vienplates datoru salīdzinājuma pēc procesora un atmiņas (sk. 1.2. tab.), vislabākā veiktspēja ir vienplates datoram *ODROID-C1+*, kuru raksturo tādi tehniskie parametri kā 4 kodolu procesors ar takts frekvenci 1,5 Ghz un 1 GB DDR3 SDRAM atmiņa³. Ražotāja mājaslapā šī vienplates datora cena ir norādīta USD 35, kas ir līdzvērtīga *Raspberry Pi 3 Model B* cenai⁴. Jāatzīmē, ka *Raspberry Pi 3 Model B* no *ODROID-C1+* atpaliek tikai mazliet (procesora takts frekvences ziņā). Savukārt no operatīvās atmiņas (RAM) viedokļa ar 2GB lielu atmiņu starp visiem lētajiem (cena zem 35 EUR) vienplates datoriem izceļas divi modeļi: *PINE A64+* un *Orange Pi Plus 2E* (sk. 1.2. tab.), kuriem abiem ir tikai mazliet mazāka procesora takts frekvence, attiecīgi 1.2 GHz un 1 GHz.

1.4.3. Salīdzinājums pēc ievadizvades saskarnēm un pieslēgvietām

Ņemot vērā maģistra darba uzdevumu izstrādāt mobilu automašīnu kustības sensoru ar numuru atpazīšanas risinājumu un automašīnas plūsmas reģistru, kas izmantots automašīnu fotogrāfiju augšupielādi mākonī, svarīgi vienplates datora tehniskie parametri ir ievadizvades iespējas un pieslēgvietas, piemēram, lai mikrodatoram būtu maģistra darba uzdevuma izpildei nepieciešamā tīkla *Ethernet* pieslēgvietā un *WiFi* mobilitātes nodrošināšanai, *USB* pieslēgvietas klaviatūras un peles pieslēgšanai, *CSI* kameras pieslēgšanai un atmiņa, vēlams, ar paplašināšanas iespēju. Tāpēc izvēles izdarīšanai ne mazāk svarīgs ir lētāko vienplates datoru salīdzinājums pēc ievadizvades saskarnēm un pieslēgvietām (sk. 1.3. tab.).

³ http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143703355573&tab_idx=2

⁴ <https://www.raspberrypi.org/blog/raspberry-pi-3-on-sale/>

Lētāko vienplates datoru salīdzinājums pēc ievadizvades saskarnēm un pieslēgvietām

Nosaukums	USB 2.0	Atmiņa		Tīkls		GPIO galvene
		Iebūvēta	Zibatmiņas slots	Ethernet Mbps	Wi-Fi	
Raspberry Pi Zero v1.3	1 MicroUSB	-	microSD	-	+ 4EUR caur microUSB	40-pin
Orange Pi Zero	1	-	TF card (Max. 64GB)	10/100	IEEE 802.11 b/g/n	26-pin
NanoPi NEO	1 + 1 MicroUSB	-	microSD	10/100	-	36-pin
C.H.I.P.	1	4 GB NAND	-	-	802.11b/g/n Bluetooth 4.0 LE	80-pin
Orange Pi One	1	-	TF card (Max. 64GB) / MMC	10/100	-	40-pin
Raspberry Pi Zero Wireless	1 MicroUSB	-	microSD	-	802.11n & Bluetooth 4.0	40-pin
Orange Pi Lite	2	-	TF card (Max. 64GB) / MMC	-	Wifi Antenna	40-pin
Orange Pi PC	3	-	TF card (Max. 64GB) / MMC	10/100	-	40-pin
PINE A64	2	-	microSD	10/100	-	46-pin
pcDuino Lite	2	-	microSD (TF) līdz 32GB	10/100	-	22-pin
Raspberry Pi A+	1	-	microSD	-	-	17-pin
Raspberry Pi A	1	-	SD, MMC	-	-	8-pin
Orange Pi 2	4	-	TF card (Max. 64GB) / MMC	10/100	Realtek RTL8189ETV, IEEE 802.11 b/g/n	40-pin
Orange Pi PC Plus	3	8GB EMMC Flash	TF card (Max. 64GB) / MMC	10/100	-	40-pin
Arduino Uno R3	-	32 KB Flash + 1 KB EPROM	-	-	-	26-pin
PINE A64+	2	-	microSD	10/100/1000	-	46-pin
ODROID W	Nav, ir tikai bloks	eMMC moduļa iespēja	microSD	-	-	32-pin
OLinuXino A10 LIME	2	-	microSD	100	-	160-pin
NanoPi 2	1	-	2x microSD	NanoPi 2	AP6212 Wireless 802.11 b/g/n un Bluetooth	40-pin
Rikomagic MK802 base, + / II	1 + 2 x MicroUSB	4 GB Flash	microSD (max 32 GB)	-	b/g/n, WAPI(Ralink8188)	-
ODROID C1+	4	eMMC moduļa iespēja	microSD	10/100/1000 Mbps	-	40-pin
Orange Pi Plus 2E	3	16GB EMMC Flash	TF card (Max. 64GB) / MMC / līdz 2T ar 2.5 SATA disku	10/100/1000 M Ethernet RJ45	Realtek RTL8189ETV, IEEE 802.11 b/g/n	40-pin
RIoTboard	4	4GB eMMC	microSD un SD	10/100/1000	-	10-pin

Banana Pi M1	2	-	SD (max.64 GB)	10/100/1000	-	26-pin
OLinuXino A13 MICRO	1	-	microSD	-	-	142-pin
Raspberry Pi B	2	-	SD, MMC	10/100	-	8-pin
Raspberry Pi B+	4	-	microSD	10/100	-	17-pin
Raspberry Pi 2 Mod.B	4	-	microSD	10/100	-	17-pin
Raspberry Pi 3 Mod.B	4	-	microSD	10/100	2.4GHz 802.11n & Bluetooth 4.1	40-pin

Kā redzams no lētāko vienplates datoru salīdzinājuma pēc ievadizvades saskarnēm un pieslēgvietām (sk. 1.3. tab.), vismaz 2 *USB* pieslēgvietas (klaviatūrai un pelei), zibatmiņas (*SD* vai *microSD*) slots un bezvadu tīkla (*WiFi*) iespēja ir deviņiem no apskatītajiem mikroordinatoriem: *Orange Pi Zero*, *C.H.I.P.*, *Raspberry Pi Zero Wireless*, *Orange Pi Lite*, *Orange Pi 2*, *NanoPi 2*, *Rikomagic MK802 base + / II*, *Orange Pi Plus 2E* un *Raspberry Pi 3 Model B* mikroordinatoram. Tātad līderis pēc veiktspējas *ODROID-C1+* un līderi pēc operatīvās atmiņas *PINE A64+* un *Orange Pi Plus 2E* (sk. 1.2. tab.) zaudē konkurentam *Raspberry Pi 3 Mod.B* galvenokārt vienā prasībā (bezvadu tīkla iespēja - *WiFi*).

1.4.4. Salīdzinājums pēc audiovizuālām saskarnēm

Lai nodarbotos ar programmēšanu mikroordinatorā, ir nepieciešama vizuāla saskarne, piemēram, HDMI monitora pieslēgšanai. Tāpēc svarīgs ir arī mikroordinatoru salīdzinājums pēc vizuālām saskarnēm (sk. 1.4. tab.). Tā kā maģistra darba uzdevumos nav audio risinājumu izstrāde, piemēram, izstrādāt mazu diktofonu, tad no audio iespējām ir apskatīta tikai audio izeja (*audio out*). Toties maģistra darbā definētā uzdevuma – uzprogrammēt kustības sensoru, kas darbojas, analizējot kameras uzņemtus attēlus – realizācijai ir svarīga kameras pieslēgvietā, piemēram, *CSI*.

1.4. tabula

Lētāko vienplates datoru salīdzinājums pēc audiovizuālām saskarnēm

Nosaukums	Audio izeja	HDMI	Kameras pieslēgum vieta	Cita video izeja
Raspberry Pi Zero v1.3	1) Mini-HDMI, 2) stereo audio, 3) caur PWM uz GPIO	1 Mini-HDMI ligzda 1080p60 video izejai*	1 CSI	Composite video
Orange Pi Zero	-	-	-	Atbalsta ārējo plati caur 13pin
NanoPi NEO	-	-	-	-
C.H.I.P.	Izmantojot GPIO	-	-	-
Orange Pi One	-	1	1 CSI	-
Raspberry Pi Zero Wireless	1) Mini-HDMI, 2) stereo audio, 3) caur PWM uz GPIO	1 Mini-HDMI ligzda 1080p60 video izejai*	1 CSI	Composite video
Orange Pi Lite	HDMI	1	1 CSI	-
Orange Pi PC	3.5 mm ligzda un caur HDMI	1	1 CSI	-
PINE A64	3.5 mm, HDMI, S/PDIF	1 (DVI nesaderīgs)	DSI	LVDS
pcDuino Lite	-	1	-	-
Raspberry Pi A+	Analogais caur 3.5 mm	1*	1 CSI, 1	Composite video

	ligzdu, digitālais caur HDMI		DSI	
Raspberry Pi A	Analogais caur 3.5 mm ligzdu, digitālais caur HDMI	1*	1 CSI, 1 DSI	Composite video
Orange Pi 2	Analogais caur 3.5 mm ligzdu, digitālais caur HDMI	1	1 CSI	-
Orange Pi PC Plus	Analogais caur 3.5 mm ligzdu, digitālais caur HDMI	1	1 CSI	-
Arduino Uno R3	-	-	-	-
PINE A64+	3.5 mm, HDMI, S/PDIF	1 x HDMI 1.4a	DSI	LVDS
ODROID W	-	1	-	Composite video
OLinuXino A10 LIME	-	1	-	LCD galvene
NanoPi 2	-	1 x HDMI 1.4A	0.5mm 24 pin DVP interface	LVDS 0.5 mm 45 pin interfeiss
Rikomagic MK802 base, + / II	HDMI	1** (DVI nesaderīgs)	-	-
ODROID C1+	-	1	-	-
Orange Pi Plus 2E	3.5 mm ligzda un caur HDMI	1	1 CSI	-
RIoTboard	3.5 mm ligzda un caur HDMI	1	? Kameron interfeiss (atbalsta CCD vai CMOS kameru)	Paralēlais RGB interfeiss, LVDS
Banana Pi M1	3.5 mm ligzda un caur HDMI	1	1 CSI	Composite, CVBS, LVDS/RG B
OLinuXino A13 MICRO	3.5 mm	-	-	VGA, LCD galvene
Raspberry Pi B	Analogais caur 3.5 mm ligzdu, digitālais caur HDMI	1*	1 CSI, 1 DSI	Composite video
Raspberry Pi B+	Analogais caur 3.5 mm ligzdu, digitālais caur HDMI	1*	1 CSI, 1 DSI	Composite video
Raspberry Pi 2 Mod.B	Analogais caur 3.5 mm ligzdu, digitālais caur HDMI	1*	1 CSI, 1 DSI	Composite video
Raspberry Pi 3 Mod.B	Analogais caur 3.5 mm ligzdu, digitālais caur HDMI	1*	1 CSI, 1 DSI	Composite video

Jāatzīmē, ka *Raspberry Pi* gadījumā „*” pie *HDMI* nozīmē *DVI* saderīgu savienojumu, proti, *HDMI* signālu var pārveidot uz *DVI*, izmantojot pasīvo adapteri. Turpretim *Rikomagic MK802* gadījumā „**” *HDMI* nozīmē *DVI* nesaderīgu savienojumu, proti *HDMI* signālu nevar pārveidot *DVI*, izmantojot pasīvo adapteri. Tāpēc ir jābūt piesardzīgiem, izvēloties *HDMI* ekrānus, kuriem ir nepieciešams *DVI* signāls.

Ja lētāko vienplates datoru salīdzinājumā pēc ievadizvades saskarnēm un pieslēgvietām (sk. 1.3. tab.) maģistra darba uzdevumu izpildei nepieciešamajiem parametriem atbilst deviņi vienplates datori, tad to salīdzinājums pēc audiovizuālām saskarnēm, proti, vai mikrodatoram ir iespēja pieslēgt *HDMI* monitoru un kameru, izvēles iespējas sašaurinās līdz pieciem potenciālajiem pretendentiem: *Raspberry Pi Zero Wireless*, *Orange Pi Lite*, *Orange Pi 2*, *Orange Pi Plus 2E* un *Raspberry Pi 3 Model B*.

1.4.5. Fizisko un elektrisko parametru salīdzinājums

Tā kā maģistra darba mērķis ir mobila programmējama ierīce, svarīgs ir salīdzinājums arī pēc izmēra, svara un enerģijas patēriņa, lai ierīci varētu darbināt no baterijas pēc iespējas ilgāku laiku (sk. 1.5. tab.).

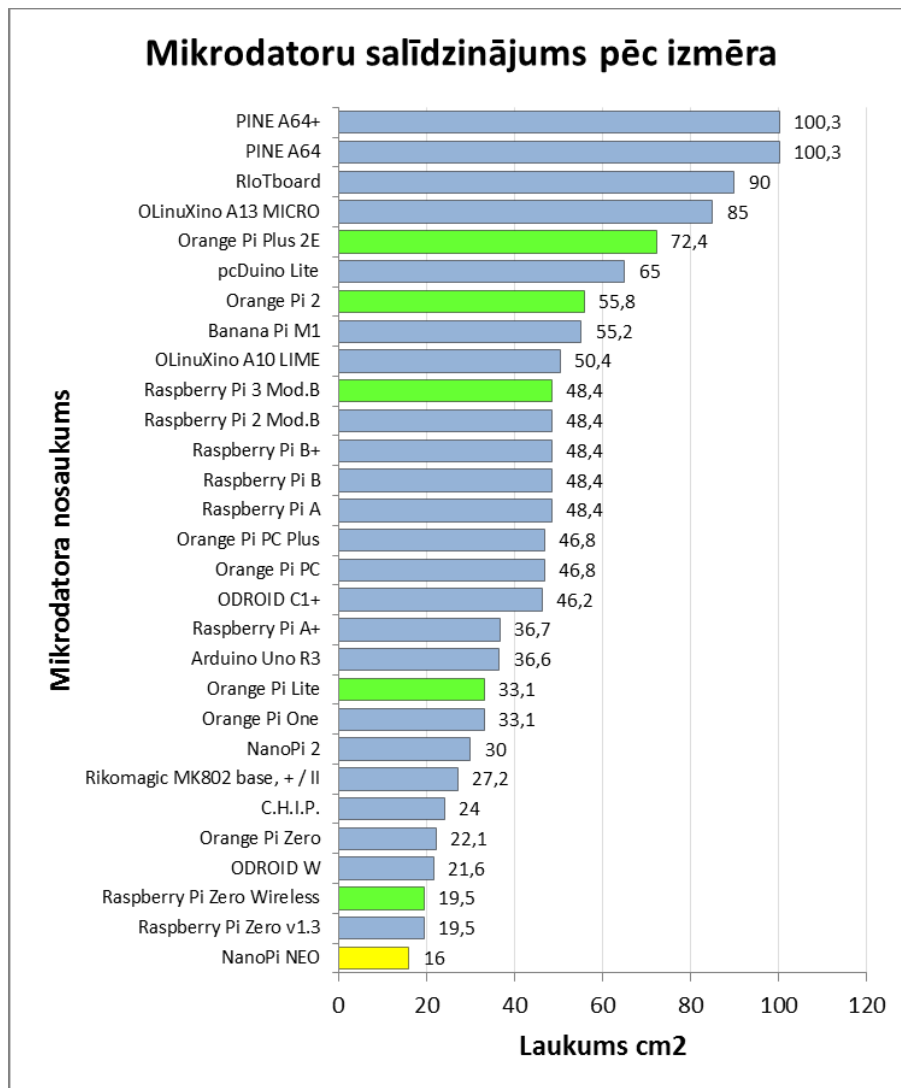
1.5. tabula

Lētāko vienplātes datoru salīdzinājums pēc izmēra un elektrības patēriņa

Nosaukums	Izmērs (mm) G x P x A	Svars (g)	Ieejas spriegums	Enerģijas patēriņš gaidīšanas režīmā	Maksimālais enerģijas patēriņš	Barošanas avots
Raspberry Pi Zero v1.3	65 x 30 x 5	9	5V	0,4 W	0,8 W	Micro USB vai GPIO galvne
Orange Pi Zero	48 x 46 x 5	26	5V	~3,3 W	5 W	USB OTG
NanoPi NEO	40 x 40	-	5V	?	?	Micro USB
C.H.I.P.	40 x 60	-	USB 5V @ 300mA	?	?	USB, microUSB, LiPo battery
Orange Pi One	69 x 48	36	5V	~3,3 W	5 W	DC ieeja
Raspberry Pi Zero Wireless	65 x 30 x 5	9	5V	0,4 W	0,8 W	Micro USB vai GPIO galvne
Orange Pi Lite	69 x 48	60	5V	~3,3 W	5 W	DC ieeja
Orange Pi PC	85 x 55	38	5V	~3,3 W	5 W	DC ieeja
PINE A64	127 x 79 x 21	46	5V	?	?	Micro USB or GPIO header
pcDuino Lite	125 x 52	?	5V	?	10 W	Micro USB
Raspberry Pi A+	65 x 56.5 x 10	23	5V	0,4 W	1.0 W (5 W piegāde)	Micro USB vai GPIO galvne
Raspberry Pi A	85.6 x 56.5 x 21	31	5V	0,4 W	1.5 W (5 W piegāde)	Micro USB vai GPIO galvne
Orange Pi 2	93 x 60	46	5V	~3,3 W	5 W	DC ieeja
Orange Pi PC Plus	85 x 55	70	5V	~3,3 W	5 W	DC ieeja
Arduino Uno R3	68.6 x 53.4	25	7–12V	0.172 W	0,233 W	DC ieeja
PINE A64+	127 x 79 x 21	46	5V	?	?	Micro USB vai GPIO galvne
ODROID W	60 x 36 x 7	8	5V	?	?	Micro USB, GPIO galvne vai LiPo baterijas
OLinuXino A10 LIME	84 x 60	?	5V	?	1.3 W	DC līgza vai USB OTG ieeja vai LiPo baterijas
NanoPi 2	75 x 40	22	5V	?	?	Micro USB vai GPIO galvne
Rikomagic MK802 base, + / II	97 x 28 x 12	20	?	?	?	?
ODROID C1+	85.6 x 54.0 x 19.5	45	5V	-	-	Micro USB, 2.5mm DC vai GPIO galvne
Orange Pi Plus 2E	108 x 67	83	5V	~3,3 W	5 W	DC ieeja
RIoTboard	120 x 75	?	5V		5 W	DC ieeja
Banana Pi M1	92 x 60	48	5V	1.15 W	10 W	Micro USB un/vai MicroUSB (OTG)
OLinuXino A13 MICRO	100 x 85	?	6-16V			USB OTG vai 5VDC ieeja
Raspberry Pi B	85.6 x 56.5 x 21	45	5V	0,9 W	3.5 W (5 W piegāde)	Micro USB vai GPIO galvne
Raspberry Pi	85.6 x 56.5	45	5V	0,9 W	3.0 W (5 W)	Micro USB vai GPIO

B+	× 21				piegāde)	galvene
Raspberry Pi 2 Mod.B	85.6 × 56.5 × 21	45	5V	1.0 W	4.0 W (5 W piegāde)	Micro USB vai GPIO galvene
Raspberry Pi 3 Mod.B	85.6 × 56.5 × 17	45	5V	1.0 W	4.0 W (5 W piegāde)	Micro USB vai GPIO galvene

Ja vienplates datora izmērus *mm* vai dimensijas reducē uz laukumu *cm²*, tad 1.5. tabulā apkopoto mikrodatoru salīdzinājums skaidri norāda uz līderi (mazāko mikrodatoru) – *NanoPi Neo* (sk. 1.2.att.), taču jāņem vērā, ka šim mikrodatoram nav bezvadu tīkls *WiFi*, *HDMI*, audio izeja, ir tikai 256MB liela operatīvā atmiņa (*RAM*) un nav vissvarīgākais – kameras pieslēgvietā *CSI*.



1.2. att. Lētāko vienplates datoru salīdzinājums pēc izmēra (laukuma)

Ja salīdzina iepriekš 1.4.4. nodaļā veiktajā salīdzinājumā izvirzītos piecus modeļus (*Raspberry Pi Zero Wireless*, *Orange Pi Lite*, *Orange Pi 2*, *Orange Pi Plus 2E* un *Raspberry Pi 3 Model B*), tad *Raspberry Pi Zero Wireless* ir izmēra ziņā vismazākais un maģistra darba uzdevuma realizācijai atbilstošākais mikrodators, ja neskaita veiktspējas parametrus (procesora kodolu skaitu, takts frekvenci un operatīvās atmiņas lielumu), kas foto attēlu uzņemšanai (kustības sensora risinājumam) arī nebūtu tik svarīgi.

2. Kāpēc Raspberry Pi?

Parasti programmēšana asociējas ar darbu pie stacionārā datora un reti kurš iedomājas, ka tas var būt arī kredītkartes lieluma vienplates dators, kura izmēri nepārsniedz 65mm x 30mm x 5mm [8] vai 19cm². *Raspberry Pi Zero* faktiski ir izejmateriāls – plate (sk. 2.1.att.).

Raspberry Pi Zero izvēle šī maģistra darba uzdevumu izpildei ir pamatojama ar tādiem kritērijiem kā lēts, mazs, bagāts ar ievadizvades, audiovizuālām saskarnēm, pieslēgvietām, mazs enerģijas patēriņš [9] un vienlaicīgi pietiekami jaudīgs maģistra darba uzdevuma – kustības sensors – izpildei (sk. pirmajā nodaļā veikto mikrodatoru salīdzinājumus).

2.1. Īss ieskats vēsturē

Raspberry Pi ir nesena vēsture, kuras īss apskats palīdz labāk izprast šīs iekārtas konceptu un mikrodatoru pielietojumu kopumā.

Viss aizsākās 2006. gadā ar pirmo *Raspberry Pi* konceptu, balstītu uz *Atmel ATmega644* mikrokontrolieri. Tā shēmas un PCB izkārtojums ir publiski pieejams⁵. Fonda aizgādnieks Ebens Uptons sapulcināja grupu skolotāju, akadēmiķu un datoru entuziastu izstrādāt datoru, lai iedvesmotu bērnus⁶. Mikrodators ir iedvesmots no 1981.gada *Acorn BBC Micro* datora^{7,8}. Modelim A, Modelim B un Modelim B+ ir atsauces uz sākotnējiem britu izglītības BBC Micro datoru modeļiem, ko izstrādāja kompānija *Acorn Computers*⁹. Pirmā datora ARM prototipa versija ir USB atmiņas kartes izmērā¹⁰. Tās vienā galā ir USB ports, bet otrā – HDMI ports. Fonda mērķis sākotnēji ir piedāvāt divas zemo izmaksu mikrodatora versijas, kuru cena ir USD 25 un USD 35, t.i., modeli A un modeli B. 2012. gada 29.februārī ražotāji sāka pieņemt pasūtījumus augstākas cenas modelim B¹¹, bet 2013. gada 4.februārī – zemākas cenas modelim A¹². 2014.gada 10.novembrī tika izlaists vēl zemākas cenas (USD 20) modelis A+¹³, bet 2015.gada 26.novembrī tika izlaists vislētākais un mazākais *Raspberry Pi* modelis *Zero*, kura ražotāja noteiktā cena ir USD 5 vai GBP 4¹⁴.

⁵ <http://www.ubergizmo.com/2011/10/build-raspberry-pi-minicomputer/>

⁶ <http://www.guardian.co.uk/education/2012/jan/09/raspberry-pi-computer-revolutionise-computing-schools?newsfeed=true>

⁷ <http://www.raspberrypi.org/phpBB3/viewtopic.php?f=2&t=5118>

⁸ <http://www.businessweekly.co.uk/blog/cambridge-today-tony-quested/13664-raspberry-blown-at-cambridge-software-detractors>

⁹ http://www.theregister.co.uk/2011/11/28/raspberry_pi?page=3

¹⁰ <http://www.tomshardware.com/news/Raspberry-Pi-David-Braben-Ubuntu-9-OLPC-Railroad-Tycoon,12709.html>

¹¹ <http://www.engadget.com/2012/02/29/raspberry-pi-credit-card-sized-linux-pcs-are-on-sale-now-25-mo/>

¹² <http://www.raspberrypi.org/archives/3215>

¹³ <http://www.raspberrypi.org/blog/#raspberry-pi-model-a-plus-on-sale>

¹⁴ <http://swag.raspberrypi.org/collections/pi-zero/products/pi-zero>

2.1.1. Pirms izlaišanas

2011.gada jūlijā pārstāvis Ebens Aptone publiski uzsāka sarunas ar RISC OS Open Ltd. kopienu par palīdzību pieslēgvietu jautājumā. Adrians Lee, kurš pārstāvēja kompāniju *Broadcom*, ar pieslēgvietām ir strādājis kopš kompānijas izveidošanas dienas. Viņa darbi ir citēti diskusijās par grafikas draiveriem. Šī pieslēgvietas pašlaik ir iekļauta NOOBS.

2011.gada augustā tika saražotas pirmās 50 alfa plates, kas funkcionāli ir identiskas plānotajam B modelim, tikai ar fiziski lielāku uzturēto atklūdošanas galveni. Plates demonstrācijas parāda, ka uz tās darbojas *LXDE* darbvirsma, balstīta uz *Debian*, *Quake 3* pie 1080p un *Full HD MPEG-4* video, izmantojot *HDMI*.

2011.gada oktobrī tika publiski demonstrēta kompānijas „Acorn Computers Ltd”, Anglijā, Kembridžā izstrādātā operētājsistēma RISC OS 5, kas pēc gada tika attīstīta kā ports un 2012.gada novembrī izlaista vispārējai lietošanai.

2011.gada decembrī no simts shēmas platēm tika izlaistas un notestētas 25 modeļa B beta versijas plates¹⁵. Beta platēm bija tāds pats komponentu izkārtojums kā ražošanas platēm. Plates dizainā tika atklāta viena kļūda, kur dažas tapas procesorā nedarbojas augstu. Tas tika izlabots pirmajā ražošanas izlaidumā. Beta platēs darbojās *Linux*, spēlējot 1080p filmas reklāmkadrus un *Rightware Samurai OpenGL ES* etalonuzdevumu.

2012.gada sākumā eBay izsolē tika piedāvātas pirmās 10 plates. Viena no tām tika nopirkta anonīmi un ziedota Anglijas Sufolkas Skaitļotāju Vēstures centram (*The Centre for Computing History in Suffolk, England*). Desmit plates (ar kopējo mazumtirdzniecības cenu GBP 220) kopā vēlāk pacēla ieņēmumus virs GBP 16000¹⁶.

2.1.2. Izlaišana

2012.gada 19.februārī tiek izlaista pirmā pārbaudītā SD kartes koncepta kopija, ko var ielādēt SD kartē, lai izveidotu sākotnējo operētājsistēmu. Kopija ir balstīta uz *Debian 6.0 (Squeeze)*, ar *LXDE* darbvirsmu un Midori pārlūku, kā arī papildus pievienotiem dažādiem programmēšanas rīkiem. Kopija darbojas arī emulatorā *QEMU*, ļaujot *Raspberry Pi* emulēt dažādās citās platformās^{17 18}.

2012.gada 29.februārī plkst.06:00 UTC tiek uzsākta sākotnējā pārdošana¹⁹. Tajā pašā laikā tiek paziņots, ka modelim A ar sākotnēji projektētiem *128 MB RAM* ir jābūt atjauninātam un uzlabotam līdz *256 MB* pirms izlaišanas²⁰.

¹⁵ <http://www.raspberrypi.org/we-have-pcbs/>

¹⁶ http://www.ebay.co.uk/itm/Raspberry-Pi-Model-B-beta-board-01-limited-series-10-/180786868894?pt=LH_DefaultDomain_3&hash=item2a17bcb29e

¹⁷ <https://web.archive.org/web/20120402160328/http://www.linuxnewshere.com/index.php/raspberrypi-releases-1st-sd-card-image-debian>

¹⁸ <https://web.archive.org/web/20120220040851/http://www.raspberrypi.org/archives/645>

¹⁹ <http://www.bbc.co.uk/news/technology-17190918>

2012.gada martā nepareiza tīkla *Ethernet* pieslēgvietas instalācijas rezultātā notika pirmās partijas piegādes kavēšanās, taču Nodibinājums sagaidīja, ka nākamo partiju ražošanas daudzumu varētu palielināt bez grūtībām pēc vajadzības. Uptons teica, ka „mēs bijām pārliecināti, ka mums izdosies *Ethernet* savienotājus ar magnētismu dabūt lielā skaitā. Pamatā divi izplatītāji *Premier Farnell* un *RS Components* sniedza fantastisku palīdzību nepieciešamo komponentu piegādē”. Pirmā partija (10'000 plates) tika saražota Taivānā un Ķīnā²¹.

2012.gada 8.martā tika izlaista Kanādas Senekas koledžā (*Seneca College*) izstrādātā *Raspberry Pi Fedora Remix* rekomendējamā *Linux* distribūcija²².

2012.gada martā bijušais kompānijas Atomz CTO Mike Thompson ieviesa Debian²³.

2.1.3. Pēc izlaišanas

2012.gada 16.aprīlī parādās pirmie ziņojumi no pircējiem, kas saņēmuši *Raspberry Pi*²⁴.

2012.gada 20.aprīlī tika izlaisti Modeļa A un B shēmu paraugi²⁵.

2012.gada 18.maijā fonds savā tīmekļa dienasgrāmatā (blog) ziņo par testēto kameras moduļa prototipu. Prototips izmanto 14-megapikseļu moduli²⁶.

2012.gada 22.maijā tiek piegādātas vairāk kā 20'000 vienības²⁷.

2012.gada 16.jūlijā tiek paziņots, ka dienā tiek saražots 4000 vienību, ļaujot *Raspberry Pi* pārdot lielākos apjomos (partijās)²⁸.

2012.gada 24.augustā kļūst pieejama aparatūras paātrināta video (H.264) kodēšana pēc tam, kad kļūst zināms, ka esošās licences ietver arī kodēšanu. Iepriekš tika uzskatīts, ka kodēšana tiks pievienota ar izziņotā fotokameras moduļa izlaišanu²⁹. Tomēr stabila nemaz nepastāv stabila programmatūra aparatūras H.264 kodējumam³⁰.

2012.gada jūlijā tiek izlaista operētājsistēma *Raspbian* [14].

2012.gada 5.septembrī tika paziņota otrā pārskatītā *Raspberry Pi* modeļa B versija³¹. Plates versija 2.0 tika izlaista ar vairākiem nelieliem labojumiem un uzlabojumiem³².

²⁰ <http://www.engadget.com/2012/02/29/raspberry-pi-credit-card-sized-linux-pcs-are-on-sale-now-25-mo/>

²¹

http://www.civilsociety.co.uk/finance/news/content/11318/uk_computing_charity_manufacturing_product_a_broad

²² <http://www.raspberrypi.org/raspberry-pi-fedora-remix-our-recommended-distro-is-ready-for-download/>

²³ <http://arstechnica.com/information-technology/2013/03/how-two-volunteers-built-the-raspberry-pis-operating-system/>

²⁴ <http://www.raspberrypi.org/forum/general-discussion/delivery>

²⁵ <http://www.raspberrypi.org/archives/1090>

²⁶ <http://www.raspberrypi.org/camera-module-first-pictures/>

²⁷ <http://www.raspberrypi.org/archives/1298>

²⁸ <http://www.raspberrypi.org/archives/1588>

²⁹ <http://www.raspberrypi.org/phpBB3/viewtopic.php?f=7&t=2886>

³⁰ <http://raspberrypi.stackexchange.com/questions/3936/what-speed-can-i-expect-from-the-hardware-h264-encoding>

³¹ <http://www.raspberrypi.org/archives/1929>

³² <http://www.raspberrypi.org/archives/1929>

2012.gada 6.septembrī tika paziņots, ka nākotnē lielākā daļa *Raspberry Pi* vienību ražos Sony iekārtās Lielbritānijā, Velsā. Fonda aplēse ir, ka uzņēmums radītu 30 000 vienību mēnesī un aptuveni 30 jaunas darbavietas³³.

2012.gada 15.oktobrī tiek paziņots, ka *Raspberry Pi* modelis B ir jāaprīko ar 512 MB RAM iepriekš projektēto 256MB RAM vietā³⁴.

2012.gada 24.oktobrī fonds paziņo, ka viss VideoCore draivera kods, kas darbojas uz ARM procesoru bāzes, tiks izlaists kā bezmaksas programmatūra saskaņā ar BSD licenci³⁵.

2012.gada oktobrī tiek ziņots, ka daži klienti savus pasūtījumus gaida vairāk nekā sešus mēnešus pie viena no diviem galvenajiem izplatītājiem. Tas tika saistīts ar grūtībām saņemt CPU piegādes un ar konservatīvo pārdošanas prognozēšanu ar konkrēto izplatītāju³⁶.

2012.gada 17.decembrī sadarbībā ar *IndieCity* un *Velocix*, fonds atver *Pi* veikalu kā "vienas pieturas aģentūru visām *Raspberry Pi* (programmatūras) vajadzībām"³⁷.

2013.gada 3.jūnijā tiek ieviesta programmatūra *NOOBS*. Tas padara *Raspberry Pi* vieglāk lietojamu, vienkāršojot operētājsistēmas instalāciju. Speciālas programmatūras lietošanas vietā, lai sagatavotu *SD* karti, faili tiek atarhivēti no *ZIP* datnes un viss saturs iekopēts *FAT* formatējuma 4 GB vai lielākā *SD* kartē. Ar šo karti tad var iesāknēt *Raspberry Pi* un tālāk tiek piedāvāta 6 operētājsistēmu izvēle. Sistēma satur arī diska nodalījumu atjaunošanai, kas ļauj veikt ātru instalēto OS rīku atjaunošanu, lai mainītu *config.txt*, un piedāvā tiešsaistes palīdzības pogu ar tīmekļa pārlūku, kas pāradresē uz *Raspberry Pi* Forumu³⁸.

2013.gada oktobrī tiek paziņots, ka Lielbritānijā ir saražots viens miljons *Pi*³⁹.

2013.gada novembrī tiek paziņots, ka laikā no 24.līdz 31.oktobrim ir nosūtīti 2 miljoni *Pi*⁴⁰.

2014.gada 28.februārī, *Raspberry Pi* divu gadu jubilejas dienā, *Broadcom* kopā ar *Raspberry Pi* fondu paziņo par pilnas *VideoCore IV* grafikas kodola dokumentācijas izdošanu un pilnu grafikas izejas tekstu zem 3.klauzulas *BSD* licences^{41 42}.

³³ <http://www.raspberrypi.org/archives/1925>

³⁴ <http://www.raspberrypi.org/archives/2180>

³⁵ <http://arstechnica.com/information-technology/2012/10/all-code-on-raspberry-pis-arm-chip-now-open-source/>

³⁶ <http://www.zdnet.com/raspberry-pi-delivery-delays-leave-buyers-hungry-and-angry-7000005919/>

³⁷ <http://www.raspberrypi.org/archives/2768>

³⁸ <http://www.raspberrypi.org/archives/4100>

³⁹ <http://www.bbc.co.uk/news/technology-24435809>

⁴⁰ <http://www.raspberrypi.org/archives/5265>

⁴¹ <http://arstechnica.com/information-technology/2014/02/raspberry-pi-marks-2nd-birthday-with-plan-for-open-source-graphics-driver/>

⁴² <http://www.raspberrypi.org/a-birthday-present-from-broadcom/>

2014.gada 7.aprīlī *Raspberry Pi* oficiālajā blogā tiek izziņots *Raspberry Pi Compute modulis*, ierīce faktorā ar 200-tapiņu DDR2 SO-DIMM atmiņas moduli, kas gan nekādā veidā nav saderīgs ar šādiem RAM un ir paredzēta plaša patēriņa elektronikas dizaineriem, lai to izmantotu savu produktu pamatā⁴³.

2014.gada jūnijā *Raspberry Pi* oficiālajā blogā minēts, ka 2014.gada maijā ir nosūtīta trīs miljonā *Pi* iekārta⁴⁴.

2014.gada 14.jūlijā *Raspberry Pi* oficiālajā blogā tiek paziņots par *Raspberry Pi* modeli B+, "sākotnējā *Raspberry Pi* evolūcijas gala produktu. Tas tiek piedāvāts par to pašu cenu kā sākotnējais *Raspberry Pi* modelis B, taču iekļauj vairākus patērētāju lūgtos nelielos uzlabojumus"⁴⁵.

2014.gada 10.novembrī *Raspberry Pi* oficiālajā blogā tiek izziņots *Raspberry Pi* modelis A+, kas ir līdz šim mazākais un lētākais (\$20!) *Raspberry Pi* modelis un satur to pašu procesoru un *RAM* kā Modelis A, vienīgi tam nav tīkla *Ethernet* pieslēgvietas un ir tikai viens *USB* ports, toties ir citi Modeļa B+ jauninājumi, piemēram, mazāks enerģijas patēriņš, *micro-SD* kartes slots un 40 tapu galvenes saderīgs *GPIO*⁴⁶.

2015.gada 2.februārī *Raspberry Pi* oficiālajā blogā tiek izziņots *Raspberry Pi 2*. Līdzīgi kā modelim B+, tam ir 900 MHz četru kodolu *ARMv7 Cortex-A7 CPU*, divas reizes lielāka atmiņa (kopā 1 GB) un pilnīga savietojamība ar sākotnējās paaudzes *Raspberry Pi*⁴⁷.

2015.gada 14.maijā tiek samazināta modeļa B cena no USD 35 līdz 25, šķietami kā "ražošanas optimizācijas blakusparādība" *Pi 2* attīstības rezultātā⁴⁸. Nozares novērotāji to atzīmē diezgan skeptiski, lai gan izrādās, ka cenas kritums ir tieša atbildes reakcija C.H.I.P., vēl zemākas cenas konkurentam⁴⁹.

2015.gada 26.novembrī *Raspberry Pi* Nodibinājums izdod *Raspberry Pi Zero*, līdz šim mazāko un lētāko *Raspberry Pi* ģimenes locekli, vien 65mm × 30 mm lielu un USD 5 dārgu. *Zero* modelis ir līdzīgs modelim A+ bez kameras un LCD savienotājiem, toties mazāks un patērē mazāk enerģijas. Tas tika dots kā dāvana kopā ar *Raspberry Pi* žurnālu *MagPi* # 40, kas tika izplatīts Lielbritānijā un ASV, tā izdošanas dienā. *MagPi* tika izpārdots pilnībā gandrīz katram starptautiskam mazumtirgotājam, pateicoties šim bezmaksas bonusam⁵⁰.

⁴³ <http://www.raspberrypi.org/raspberry-pi-compute-module-new-product/>

⁴⁴ <http://www.raspberrypi.org/raspberry-pi-at-buckingham-palace-3-million-sold/>

⁴⁵ <http://www.raspberrypi.org/introducing-raspberry-pi-model-b-plus/>

⁴⁶ <http://www.raspberrypi.org/blog/#raspberrypi-model-a-plus-on-sale>

⁴⁷ <http://www.raspberrypi.org/raspberry-pi-2-on-sale/>

⁴⁸ <https://www.raspberrypi.org/price-cut-raspberry-pi-model-b-now-only-25/>

⁴⁹ <http://www.itpro.co.uk/desktop-hardware/24587/raspberry-pi-slashes-price-after-rival-launches-on-kickstarter>

⁵⁰ <https://www.raspberrypi.org/blog/raspberry-pi-zero/>

2016.gada 29.februārī tiek izziņots *Raspberry Pi 3*. Šis modelis izceļas ar *BCM2837* 1,2 GHz 64-bitu 4-kodolu procesoru, balstītu uz *ARMv8 Cortex A53*, iebūvētu *Wi-Fi BCM43438 802.11n* 2,4 GHz un *Bluetooth 4.1 Low Energy (ble)*. Šajā pašā paziņojumā ir teikts, ka dažus mēnešus vēlāk ir sagaidāms jauns uz *BCM2837* balstīts Skaitļošanas Modulis (*Compute Module*)⁵¹.

2016.gada februārī *Raspberry Pi* nodibinājums paziņo, ka ir pārdots astoņi miljoni ierīču (skaitot visus modeļus kopā), padarot to par vislabāk pārdoto UK personālo datoru, apsteidzot *Amstrad PCW* ⁵². 2016.gada septembrī pārdošanas apjomi jau sasniedza desmit miljonus⁵³.

2016.gada 25.aprīlī tiek izdota *Raspberry Pi Camera v2.1* ar 8 megapikseliem normālajai un *NOIR* (var saņemt *IR*) versijai. Kamera izmanto *Sony IMX219* mikroshēmu ar izšķirtspēju 3280 × 2464. Lai izmantotu jauno rezolūciju, ir jāatjaunina programmatūra⁵⁴.

2016.gada 10.oktobrī *NEC Display Solutions* paziņo, ka atsevišķiem modeļiem tiks izlaisti komerciālie displeji 2017.gada sākumā, iekļaujot *Raspberry Pi 3 Compute Module*⁵⁵.

2016.gada 14.oktobrī *Raspberry Pi* nodibinājums paziņo par savu sadarbību ar *NEC Display Solutions*. Viņi sagaida, ka *Raspberry Pi 3 Compute Module* būs pieejams plašākai sabiedrībai līdz 2016.gada beigām⁵⁶.

2016.gada 25.novembris izceļas ar 11 miljoniem pārdoto vienību⁵⁷.

2017.gada 16.janvārī tiek izlaists *Compute Module 3* un *Compute Module 3 Lite*⁵⁸.

2017.gada 28.februārī tiek izlaists *Raspberry Pi Zero W* ar *WiFi* un *Bluetooth*, izmantojot mikroshēmā iestrādātu antenu [7, 10].

2.2. Raspberry Pi modeļi

Šobrīd ir pieejami vairāki *Raspberry Pi* modeļi, no kuriem pēdējais modelis, *Raspberry Pi Zero W*, nācis klajā 2017.gada 28.februārī, izceļas ar saskarni kamerai *CSI*, *WiFi* un *Bluetooth*. Tālāk salīdzinošā tabulā (sk. 2.1. tab.) ir apskatīti maģistra darba rakstīšanas brīdī (2017.gada martā) pārdošanā esošie *Raspberry Pi* modeļi.

⁵¹ <https://www.raspberrypi.org/blog/raspberry-pi-3-on-sale/>

⁵² <https://www.theguardian.com/technology/2016/feb/29/raspberry-pi-3-launch-computer-uk-best-selling>

⁵³ <https://www.raspberrypi.org/blog/ten-millionth-raspberry-pi-new-kit/>

⁵⁴ <https://www.raspberrypi.org/blog/new-8-megapixel-camera-board-sale-25/>

⁵⁵ <https://www.nec-display-solutions.com/p/hq/en/news/dp/Products/Shared/News/2016/PressReleases/Company/RaspberryPi/Raspberrypi.html>

⁵⁶ <https://www.raspberrypi.org/blog/compute-module-nec-display-near-you/>

⁵⁷ <https://www.raspberrypi.org/magpi/issues/53/>

⁵⁸ <https://www.raspberrypi.org/blog/compute-module-3-launch/>

Raspberry Pi modeļu salīdzinājums

Nosaukums	Raspberry Pi Model A+ ⁵⁹	Raspberry Pi 2 Model B ⁶⁰	Raspberry Pi 3 Model B ⁶¹	Raspberry Pi Zero ⁶²	Raspberry Pi Zero W ⁶³
Izlaiduma gads/mēnesis	2014/11	2015/02	2016/02	2015/11	2017/02
Ražotāja noteiktā cena	\$20	\$35	\$35	\$5	\$10
CPU	700 MHz single-core ARM11	900 MHz quad-core ARM Cortex-A7	1.2GHz 64-bit quad-core ARMv8	700 MHz single-core ARM11	700 MHz single-core ARM11
GPU	Broadcom VideoCore IV @ 250 MHz OpenGL ES 2.0 (24 GFLOPS)				
GPIO galvenes tapīņu skaits	40	40	40	40	40
Atmiņa (SDRAM)	256 MB	1 GB	1 GB	512 MB	512 MB
Zibatmiņas slots	microSD	microSD	microSD	microSD	microSD
USB 2.0 portu skaits	1	4	4	1 Micro-USB	1 Micro-USB
Tīkls Ethernet 10/100	-	1	1	-	-
Bezvadu tīkls	-	-	802.11n	-	802.11n
HDMI izeja	1	1	1	1 Mini-HDMI ligzda 1080p60 video izejai	1 Mini-HDMI ligzda 1080p60 video izejai
Bluetooth	-	-	4.1	-	Bluetooth 4.0
Composite video izeja	1	1	1	-	-
Kameras saskarnes ports	CSI	CSI	CSI & DSI	1 CSI	1 CSI
Audio izeja 3.5 mm	1	1	1	-	-
Operētājsistēmas atbalsts	Linux	Linux Windows 10	Linux Windows 10	Linux	Linux
Barošanas avots	5 V MicroUSB vai GPIO galvene				
Enerģijas patēriņš (W) ⁶⁴	0,5 – 1 ⁶⁵	1 – 5	1 – 5	0,4 – 0,8	0,4 – 0,8

Raspberry Pi oficiālā žurnāla *MagPi* 53.izdevumā publicētā ziņa, ka 2016.gada 25.novembrī ir pārdoti 11 miljoni *Raspberry Pi* [16], liecina par šī mikrodatora strauji pieaugošo popularitāti, bezmaksas apmācību resursa, žurnāla *MagPi* un foruma nozīmi, atbalstu un lielisko risinājumu kopumā. Īpaši jāakcentē *Raspberry Pi 2* un *Raspberry Pi 3* modeļi, kuram ir Windows 10 atbalsts⁶⁶. To apstiprina arī oficiāli Microsoft paziņojumi⁶⁷.

2.3. Maģistra darba praktiskajā daļā izmantotais Raspberry Pi Zero

Maģistra darba praktiskajā daļā ir izmantots *Raspberry Pi Zero* modelis (sk. 2.1.att.), jo tas ir 1) vislētākais mikrodators (sk. 1.1.att.), tam ir maģistra darba uzdevumam nepieciešamā 2) kameras pieslēgvietā CSI un 3) iespēja pieslēgt bezvadu WiFi moduli par papildus EUR 4, 4) ieslēgta pieslēgt HDMI monitoru un 5) micro-USB ports, kuru ar adaptera palīdzību var pārveidot uz 3 USB 2.0 portiem, lai pieslēgtu klaviatūru un peli un sāktu programmēt. Tas ir arī vismazākais – mobilākais – mikrodators.

⁵⁹ <http://www.i-programmer.info/news/91-hardware/7956-new-raspberry-pi-a.html>

⁶⁰ <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

⁶¹ <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

⁶² <https://www.raspberrypi.org/products/pi-zero/>

⁶³ <https://www.raspberrypi.org/blog/#raspberrypi-zero-w-joins-family>

⁶⁴ <https://www.raspberrypi.org/help/faqs/#powerReqs>

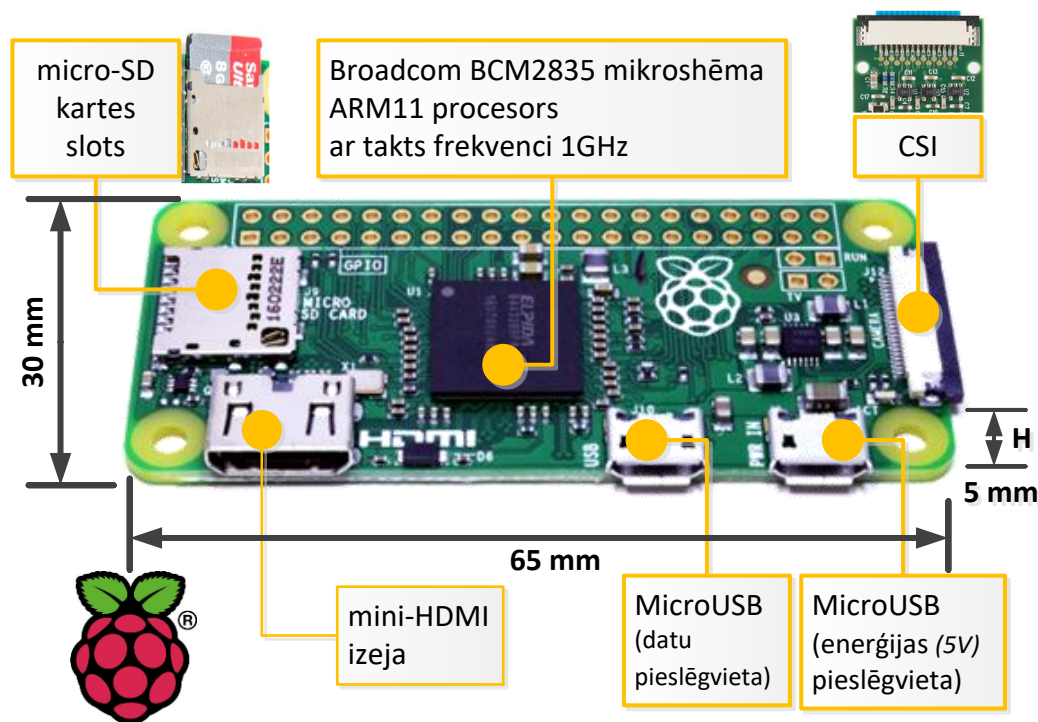
⁶⁵ <https://www.raspberrypi.org/products/model-a-plus/>

⁶⁶ <http://www.raspberrypi.org/raspberry-pi-2-on-sale/>

⁶⁷ <http://dev.windows.com/en-us/featured/raspberrypi2support>

Jāatzīmē, ka *Raspberry Pi Zero* modeli Latvijā neizdodas iegādāties, tāpēc šī maģistra darba vajadzībām tas ir pirktis 2016.gada oktobrī Lielbritānijā, Šefildā, kompānijas *Pimoroni* ražotnē par 4 GBP. *Raspberry Pi Zero* modeli raksturo šādas tehniskās īpašības:

- ✓ Procesors: 1 kodols x 1 GHz
- ✓ Operatīvā atmiņa: 512 MB SDRAM
- ✓ USB 2.0: 1 x micro-USB, ko var paplašināt ar papildus adaptera palīdzību
- ✓ Video izeja: 1x mini-HDMI
- ✓ Kameras pieslēgvietā: 1 CSI
- ✓ Atmiņas ierīce: microSD kartes slots
- ✓ Tīkls: bezvadu *WiFi* iespēja ar papildus moduli, pievienotu micro-USB portam
- ✓ Bluetooth 4.0
- ✓ Operētājsistēma: Linux



2.1. att. *Raspberry Pi Zero* arhitektūra

Lai gan pasaulē var atrast interneta veikalus, kuros *Raspberry Pi Zero* tirgo par ražotāja noteikto mērķa cenu USD 5 vai GBP 4, ja pieskaita piegādes izdevumus (aptuveni vēl GBP 4), tad Latvijā tā iegādes izmaksas ir vismaz 2 reizes lielākas, turklāt parasti *Raspberry Pi Zero* modeli interneta veikali ļauj iegādāties ierobežotā daudzumā (tikai 1 mikrodatoru vienam patērētājam)⁶⁸. Šī maģistra darba vajadzībām *Raspberry Pi Zero* ir pirktis 2016.gada oktobrī Lielbritānijā, Šefildā, kompānijas *Pimoroni* ražotnē par 4 GBP, lai ekonomētu piegādes izdevumus. Jārēķinās, ka paša mikrodatora iegādes izmaksas nav vienīgās, lai sāktu

⁶⁸ <https://shop.pimoroni.com/products/raspberry-pi-zero>

programmēt un veidot maģistra darba uzdevumos paredzēto risinājumu, jo vienplates dators nav datora komplekts, kurā ir ietverta arī klaviatūra, pele, monitors. *Raspberry Pi* netiek piedāvāts komplektā pat strāvas padeves bloku, kas ir jāiegādājas atsevišķi, ja vien par enerģijas avotu nav plānots izmantot citus risinājumus (piemēram, baterijas vai *GPIO* galveni). Strāvas padeves bloks ar *Raspberry Pi* darbināšanai nepieciešamo spriegumu 5V un strāvas stiprumu max 2A maksā papildus USD 3,07⁶⁹. Ņemot vērā, ka *Raspberry Pi* mikrodatordi izmanto *microUSB* pieslēgvietu enerģijas saņemšanai, mobilitātes nodrošināšanas nolūkos to par papildus maksu var aizstāt arī ar ārēju enerģijas avotu, piemēram, bateriju, kas nodrošinātu mikrodatorda darbību vismaz diennakti, piemēram, *YooBao M10 10000mAh 2A Dual USB External Battery Power Bank* par USD 18⁷⁰. Lētākā klaviatūra izmaksā vēl papildus 3 eiro⁷¹, datorpele – 1,42 eiro⁷², *WiFi* adapteris – USD 3,99⁷³, bet HDMI monitors – 93 eiro⁷⁴. Tāpat ir nepieciešama arī *microSD* karte, kurā varētu ielādēt operētājsistēmu. Lētākā 4GB *microSD* karte maksā vēl papildus 2,85 eiro⁷⁵. Kā vēlāk noskaidrojas, tad ar 4GB ir par maz, lai realizētu visus maģistra darba uzdevumus, t.i. atvērtā koda automašīnu numura zīmju atpazīšanas bibliotēka *OpenALPR* un ar to saistītās bibliotēkas (*leptonica*, *opencv*, *tessdata*, *tesseract* u.c.) vien aizņem vairāk kā 3GB, neizdodas pabeigt to instalāciju pat uz 8GB *microSD* kartes, tāpēc šī maģistra darba vajadzībām ir jāatvēr vismaz 16 GB *Class 10 microSD* karte, kuras cena jau ir 6,28 eiro⁷⁶. Tāpat maģistra darba uzdevuma – kustības sensora uzprogrammēšanai – ir nepieciešams arī *Raspberry Pi* kameras modulis. Oficiālā versija ir salīdzinoši ļoti dārgs priekš, tā cena ir GBP 29⁷⁷. Ķīnā ražotu kameras moduļa variantu izdodas iegādāties par USD 9,93⁷⁸. Līdz ar to pilna programmēšanai nepieciešamā komplekta sākotnējās izmaksas sastāda kopējo summu 130,54 eiro, tā skaitā daži papildus

⁶⁹ <http://www.banggood.com/5V-2A-EU-Power-Supply-Micro-USB-AC-Adapter-Charger-For-Raspberry-Pi-p-949349.html>

⁷⁰ <http://www.banggood.com/YooBao-M10-10000mAh-2A-Dual-USB-External-Battery-Power-Bank-p-993520.html>

⁷¹ <https://zcena.lv/lv/KEYBOARDS/msonic-computer-keyboard-usb-cyrylic-ru-layout-mk151uk-black>

⁷² <https://www.e-abc.lv/datortehnika/periferijas-ierices-izejmateriali-ups/datorpeles/extreme-xm102k-wired-usb-optic-38997>

⁷³ <http://www.banggood.com/Realtek-8188-150M-USB-Wi-Fi-Wireless-Adapter-Realtek-RTL8188-Chip-For-Windows-Mac-Linux-p-983419.html>

⁷⁴

http://www.1a.lv/datortehnika_preces_birojam/personalie_datori_monitori_ups/led_monitori/philips_223v5l_hsb200

⁷⁵ <https://www.zauers.lv/computer-hardware/accessories/komponentes/kingston-microsd-hccard-4gb-class-4-190037>

⁷⁶ <http://www.v24.lv/datortehnika/datu-neseji/atminas-kartes/microsd-sandisk-sdhc-16gb-class-10-sdsqunb-016g-gn3mn?p=183721>

⁷⁷ <https://shop.pimoroni.com/products/raspberry-pi-camera-module-v2-1-with-mount>

⁷⁸ <https://ru.aliexpress.com/item/Raspberry-Pi-zero-camera-Module-Board-5MP-Webcam-also-support-PI3-PI2-B/32784837169.html>

aksesuāri, piemēram, OTG adapteris USB konvertēšanai uz *microUSB*, kas viss pārskatāmā veidā ir apkopots tabulā (sk. 2.2. tab.).

2.2. tabula

Risinājuma izmaksas maģistra darba uzdevuma realizācijai

NPK	Nosaukums	Cena	Valūta	Cena EUR
1	Raspberry Pi Zero	4	GBP	4,69
2	USB to microUSB OTG Converter Shim	2	GBP	2,34
3	Realtek RTL8188 150M USB WiFi Adapter	3,99	USD	3,77
4	MicroSD SanDisk SDHC 16GB Class 10	6,28	EUR	6,28
5	5V 2A EU Power Supply Micro USB AC Adapter Charger	3,07	USD	2,9
6	Raspberry Pi zero camera Module Board 5MP	9,93	USD	9,37
Risinājuma izmaksas KOPĀ				29,35
7	OTG 3 in1 USB 3.1 Type-C to 3 USB Ports 2.0 Power Charging Hub Data Cable Adapter	3,99	USD	3,77
8	MSONIC Computer Keyboard	3	EUR	3
9	EXTREME XM102K Wired USB Optical Mouse	1,42	EUR	1,42
10	Philips 223V5LHSB2/00	93	EUR	93
Netiešās izmaksas				101,19
Pilna programmēšanas komplekta sākotnējās izmaksas				130,54
11	YooBao M10 10000mAh 2A Dual USB External Battery Power Bank	17,99	USD	16,98
Papildus izmaksas mobilam risinājumam				16,98
Mobila risinājuma izmaksas KOPĀ				46,33

Ņemot vērā, ka monitors, klaviatūra un pele ir tikai palīgierīces, kas nepieciešamas darbam (programmēšanai) un vēlāk pēc *Raspberry Pi Zero* sagatavošanas kādam uzdevumam vairs nav nepieciešamas (tās var tikt izmantotas citām vajadzībām, kā arī uz programmēšanas laiku tās var aizņemties no datora komplekta, iznomāt vai iegādāties lietotas, lai samazinātu izmaksas), tad reālās mikrodatora darbināšanas izmaksas sastāda vien nepilnus 30 eiro.

3. Kāpēc PYTHON?

Pasaulē ir maz efektīvu un vienlaicīgi vienkāršu, nesarežģītu lietu. Programmēšanas valodas ir viena no jomām, kuru nevar nosaukt par vienkāršu. Programmēšanas valodu izvēle *Basic* līmenī nav pārāk plaša. *Python* faktiski ir skriptu valoda, bet pietiekami vienkārša, lai būtu viegli apgūstama un varētu paveikt arī smagnējākām programmēšanas valodām paredzētus uzdevumus [3]. *Python* atbalsta plašu klāstu datu tipu (sk. 2.1. tab.).

2.1. tabula

Python un Visual Basic sintakses salīdzinājums

Datu tips	Visual Basic	Python
String	Dim nosaukums as String	nosaukums = „Kamera”
Integer number	Dim skaits As Integer	skaits = 1
Boolean	Dim darbojas as Boolean	darbojas = False
Floating point number	Dim summa As Double	summa = 525,20

Ja *Visual Basic* prasa mainīgos definēt kopā ar to veidu, valodā *Python* pietiek tikai ar mainīgā nosaukumu, vienādības zīmi un vērtību (sk. 2.1. tab.).

Programmēšanas valodas *Python* izvēle šī maģistra darba ietvaros ir saistīta arī ar iekārtas *Raspberry Pi* izvēli. *Raspberry Pi* izmantošana nav ierobežota tikai ar *Python* programmēšanas valodas lietošanu. *Raspberry Pi* pēc noklusējuma piedāvā arī *C*, *C++*, *Java*, *Scratch* un *Ruby* programmēšanas valodu. Uz *Raspberry Pi Zero* var izmantot jebkuru programmēšanas valodu, kura kompilējas uz ARMv6. Tomēr *Raspberry Pi* oficiālajā tīmekļa vietnē kā oficiālā programmēšanas valoda ir minēta *Python*⁷⁹.

3.1. Īss ieskats Python rašanās vēsturē

Ideja par programmēšanas valodu *Python* radās 1980.gadu beigās, taču tās ieviešanu uzsāka Gvido van Rosums Nīderlandes nacionālajā matemātikas un informātikas pētniecības institūtā kā pēcteci ABC valodai, kas spēj apstrādāt izņēmumus un mijiedarboties ar Andrew Stuart Tanenbauma un citu autoru izstrādāto operētājsistēmu *Amoeba* [3].

3.2. Python sintakses un semantikas īpatnības

Python ir paredzēta kā ļoti viegli lasāma valoda. Tai ir izstrādāts nepārblīvēts vizuālais izkārtojums, bieži izmantojot angļu atslēgvārdus kur citās valodās izmanto pieturzīmes. Programmēšanas valodā *Python* ir mazāks skaits sintaktisko izņēmumu un īpašo gadījumu, salīdzinot, piemēram, ar *C* vai *Pascal*.

Python izmanto atkāpes nevis figūriekavas vai atslēgvārdus, lai atdalītu blokus. Šo īpatnību sauc arī par ārpusmalas (*off-side*) likumu. Atkāpes palielināšana seko pēc noteiktiem apgalvojumiem. Atkāpes samazināšana nozīmē pašreizējā bloka beigas.

Python apgalvojumi ir šādi [3]:

⁷⁹ <http://www.raspberrypi.org/help/faqs/#softwareLanguages>

- `if` apgalvojums nosacīti izpilda koda bloku kopā ar `else` un `elif` (kontrācija `else-if`)
- `for` apgalvojums atkārtoti atkārtojamo objektu, uztverot lokālajā mainīgajā katru elementu lietošanai pievienotajā blokā
- `while` apgalvojums
- `try` apgalvojums
- `class` apgalvojums
- `def` apgalvojums
- `with` apgalvojums
- `pass` apgalvojums
- `assert` apgalvojums
- `yield` apgalvojums
- `import` apgalvojums
- `print()` *Python 3* ir mainīta uz funkciju.

3.3. Python 2 vai Python 3

Šobrīd pastāv uzskats, ka *Python 2* ir mantojums, bet *Python 3* ir tagadne un valodas nākotne⁸⁰. *Python 2* tika izlaists 2000.gadā, savukārt *Python 3* - 2008.gadā. Ieteicamāka ir *Python 3*, lai gan joprojām uz *Python 3* vēl nav pārnestas dažas bibliotēkas, kas ir iemesls tam, kāpēc *Python 2* joprojām ir diezgan plaši izmantota *Python* versija.

Galvenās *Python 2* un *Python 3* atšķirības ir šādas:

- `Print`. Versijā *Python 2* `print` ir apgalvojums, kas neprasa iekavas, piemēram, `print „Sveiki“`. Versijā *Python 3* `print` ir funkcija, kurai ir jāpadod drukāšanai paredzētie parametri, piemēram, `print("Sveiki")` vai `print("Mani sauc", vards)`. Lai arī *Python 2* `print` strādā arī ar iekavām, vairāku objektu drukāšana ar vienu `print` komandu tomēr darbojas atšķirīgi. Ja *Python 3* drukā katru vienību, kas ir atdalīta ar atstarpī, tad *Python 2* vienību kolekciju drukā kā vektoru (*tuple*), piemēram, („Mana kaķa vecums ir”, 9)
- Ievade / izejas datu ievade. Versijā *Python 2* lietotāja ievadītos datus paņem funkcija `raw_input`, kas *Python 3* ir nosaukta vienkārši `input`.
- Veselu skaitļu (*Integer*) dalīšana. Versijā *Python 2* dalīšanas zīmi `„/”` izmanto tiešai veselu skaitļu dalīšanai, vienmēr atgriežot veselu skaitli. Tas nozīmē, ka

⁸⁰ <http://www.raspberrypi.org/documentation/usage/python/more.md>

dalīšanas darbība atgriež vērtību, kas ir veselas reizes, cik viens skaitlis dalās ar citu, ignorējot atlikumu. Piemēram, $1/2$ atgriež 0, $2/2$ atgriež 1, $3/2$ atgriež 1. Versijā *Python 3* dalīšanas darbība dod pareizu dalījuma rezultātu, atgriežot decimāldaļskaitli. Piemēram, $1/2$ atgriež 0.5. Lai decimāldaļskaitli dabūtu *Python 2*, pirms dalīšanas izpildes uz decimāldaļskaitli ir jākonvertē viens vai abi skaitļi, piemēram, $1.0/2$ atgriezīs 0.5. Savukārt, lai *Python 3*, dalot skaitļus, dabūtu precīzi veselu skaitli, dalīšanā ir jāizmanto divas dalīšanas zīmes `//`. Piemēram, `1//2` atgriezīs 0 [20].

Kopš versijas *Python 2.7.6* izlaišanas 2013.gadā, versiju zaram 2.x vairs nav bijuši jauni, nozīmīgi izlaidumi.

3.4. Python filosofija

Python izvēli programmēšanai ietekmē arī *Python* filosofija, ko visspilgtāk raksturo ilggadējā *Python* entuziasta Tima Petera uzrakstītie 20 aforismi, no kuriem publiski pieejami ir tikai 19 [18]:

- 1) Skaists ir labāk nekā neglīts (*Beautiful is better than ugly*);
- 2) Skaidri izteikts ir labāk nekā netieši izteikts (*Explicit is better than implicit*);
- 3) Vienkāršs ir labāk nekā salikts (*Simple is better than complex*);
- 4) Salikts ir labāk nekā sarežģīts. (*Complex is better than complicated*);
- 5) Vienots ir labāk nekā iekļauts (*Flat is better than nested*);
- 6) Izretināts ir labāk nekā blīvs (*Sparse is better than dense*);
- 7) Lasāmība skaitās (*Readability counts*);
- 8) Specifiski gadījumi nav pietiekami īpaši, lai pārkāptu noteikumus (*Special cases aren't special enough to break the rules*);
- 9) Lai gan praktiskums pārspēj skaidrību (*Although practicality beats purity*);
- 10) Kļūdas nekad nedrīkst noklusēt (*Errors should never pass silently*);
- 11) Ja vien skaidri izteiktais ir noklusēts (*Unless explicitly silenced*);
- 12) Saskaroties ar neskaidrību, atteikties no vilinājuma minēt (*In the face of ambiguity, refuse the temptation to guess*);
- 13) Vajadzētu būt tikai vienam – un vislabāk tikai vienam – nepārprotami skaidram veidam to darīt (*There should be one-- and preferably only one --obvious way to do it*);
- 14) Lai gan tas var nebūt acīmredzams sākumā, ja vien tu esi holandietis (*Although that way may not be obvious at first unless you're Dutch*);
- 15) Tagad ir labāk nekā nekad (*Now is better than never*);

- 16) Kaut arī nekad biežāk ir labāk nekā **tieši** tagad (*Although never is often better than *right* now*);
- 17) Ja ieviešanu ir grūti izskaidrot, tā ir slikta ideja (*If the implementation is hard to explain, it's a bad idea*);
- 18) Ja ieviešanu ir viegli izskaidrot, tā var būt laba ideja (*If the implementation is easy to explain, it may be a good idea*);
- 19) Nosaukumvietas ir viena lieliska ideja – izdariet vairāk! (*Namespaces are one honking great idea -- let's do more of those!*).

3.5. Kopsavilkums

Python pārākumu pār citām programmēšanas valodām raksturo šādi argumenti:

- ✓ universāla programmēšanas valoda (piemēram, ar efektīvu grafisko lietotāja saskarni (GUI), ievadizvadi, internetu, datubāzes bibliotēkām)
- ✓ bezmaksas un atvērta koda;
- ✓ pilnībā objektorientēta;
- ✓ ļoti portatīva;
- ✓ paplašināma un iegulstama.

Šo iemeslu dēļ programmēšanas valoda *Python* ir izvēlēta maģistra darbā noteikto uzdevumu izpildei – tālākajās nodaļās aprakstītā risinājuma programmēšanai.

4. Raspberry Pi programmēšana

Pasaulē ir izstrādāti dažādi risinājumi, kas darbojas uz *Raspberry Pi*, sākot no vienkāršākiem, piemēram, vienkārša datorspēle, līdz sarežģītākiem robotikas risinājumiem, kuri prasa *Raspberry Pi* aprīkot ar papildus iekārtām (piemēram, sensoriem, ievadizvades ierīcēm, kamerām utt.). Šī maģistra darba ietvaros ir plānots, izmantojot vislētāko mikroprocesoru *Raspberry Pi Zero*, programmēšanas valodu *Python* un atvērtā koda automašīnu numura zīmju atpazīšanas bibliotēku *OpenALPR*, kā arī ar to saistītās bibliotēkas (*leptonica*, *opencv*, *tessdata*, *tesseract* u.c.), izstrādāt un palaist darbībā:

- 1) tīmekļa serveri un mājas lapu, kura atjaunojas 5 sekunžu intervālā un parāda aktuālāko kameras uzņemto attēlu;
- 2) kustības sensoru, kas darbojas, analizējot izmaiņas kameras uzņemtajos attēlos, izmantojot *Raspberry Pi Camera* moduli;
- 3) automašīnu numuru atpazīšanas risinājumu ar automatisku automašīnu plūsmas reģistru.

4.1. Raspberry Pi Zero sagatavošana darbam

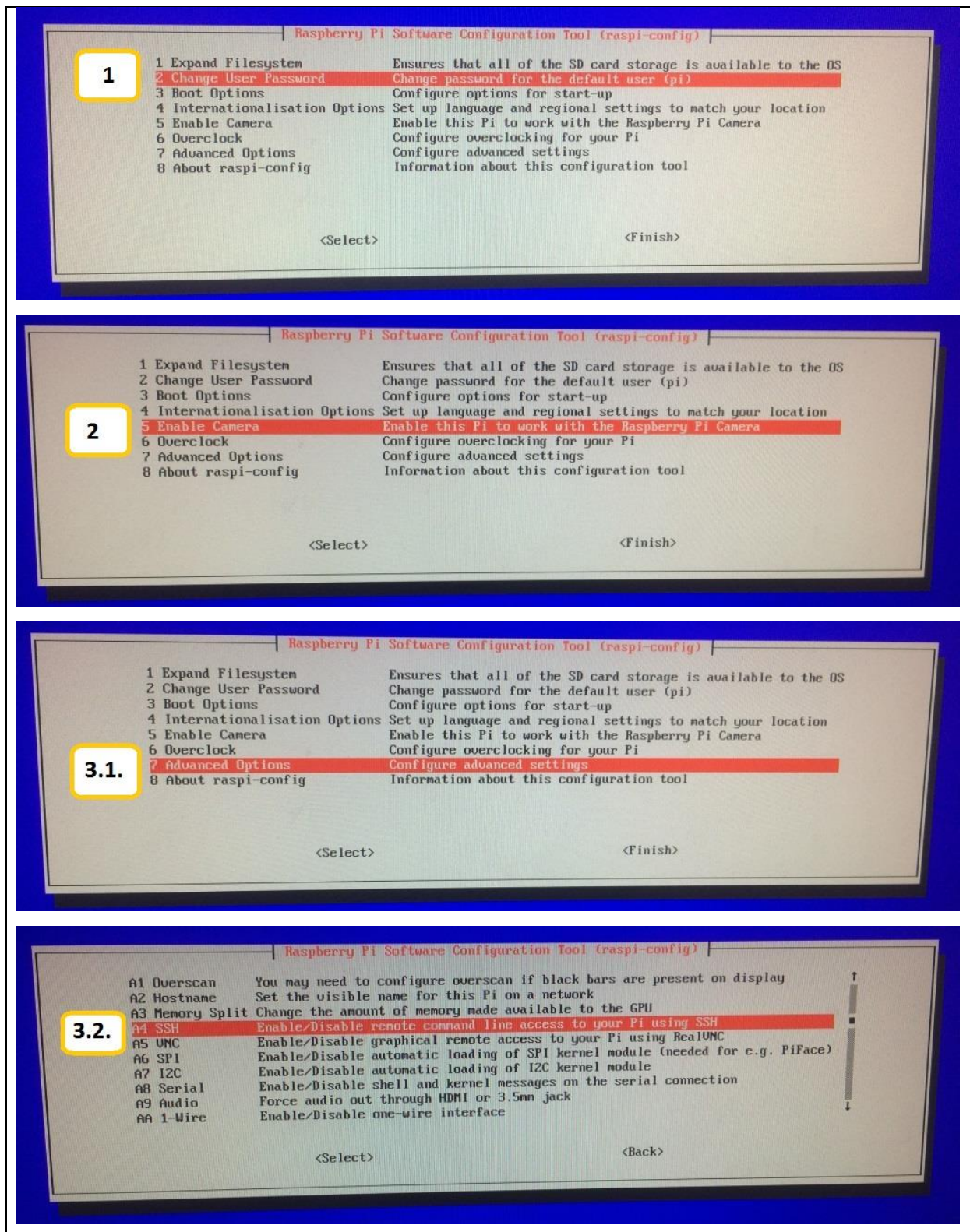
Pirms sākt kaut ko programmēt uz *Raspberry Pi*, tas vispirms ir jāgatavo darbam, t.i.

- 1) jālejupielādē operētājsistēma un jāuzkopē tā uz *MicroSD* kartes. Maģistra darbā izmantotā operētājsistēma ir *Raspbian Jessie Lite*, kas ir *Raspberry Pi* nodibinājuma oficiāli atbalstītā operētājsistēma⁸¹. Maģistra darba ietvaros izmantotā un lejupielādētā operētājsistēmas datne ir *2017-03-02-raspbian-jessie-lite.img*, kas tālāk, izmantojot programmu *Win32DiskImager*⁸² ir ierakstīta uz 16 GB *MicroSD* kartes⁸³;
- 2) jāievieto *MicroSD* karte attiecīgajā ligzdā uz *Raspberry Pi Zero* plates (sk. 2.1.att.), jāpieslēdz monitors, klaviatūra, pele, *WiFi* adapteris, strāvas padeves avots un jāieslēdz *Raspberry Pi Zero*;
- 3) pēc aptuveni 1 min ilgas iesāknēšanās jāpiesakas sistēmā ar sākotnējo lietotāja vārdu „*pi*” un paroli „*raspberrypi*”;
- 4) jānokonfigurē *Raspberry Pi Zero* attālinātam darbam (piemēram, izmantojot *SSH*), jāiespējo kameras atbalsts, jānomaina sākotnējā parole utt. konfigurācijas rīkā, ko var atvērt, izmantojot komandu „`sudo raspi-config`” (sk. 4.1.att.);

⁸¹ <https://www.raspberrypi.org/downloads/raspbian/>

⁸² <http://sourceforge.net/projects/win32diskimager/>

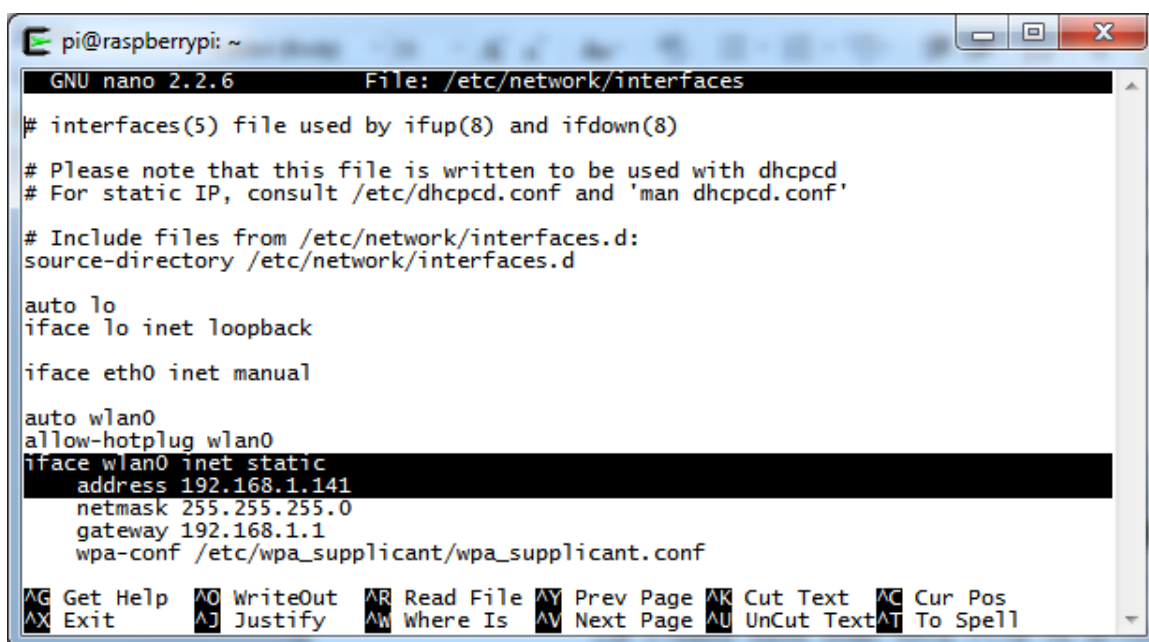
⁸³ <https://www.raspberrypi.org/documentation/installation/installing-images/windows.md>



4.1. att. 3 sākotnējās darbības Raspberry Pi Zero konfigurēšanas rīkā

5) pēc „Finish” (sk. 4.1.att.-1) nospiešanas *Raspberry Pi* konfigurācijas rīkā (*raspi-config*) un iekārtas restarta, *Raspberry Pi Zero* ir jānokonfigurē darbībai tīklā, izmantojot statisku IP adresi. To var izdarīt attālināti, izmantojot *SSH* un dinamiski piešķirto IP adresi, ko var uzzināt ar komandu *ifconfig*, vai turpat konsolē.

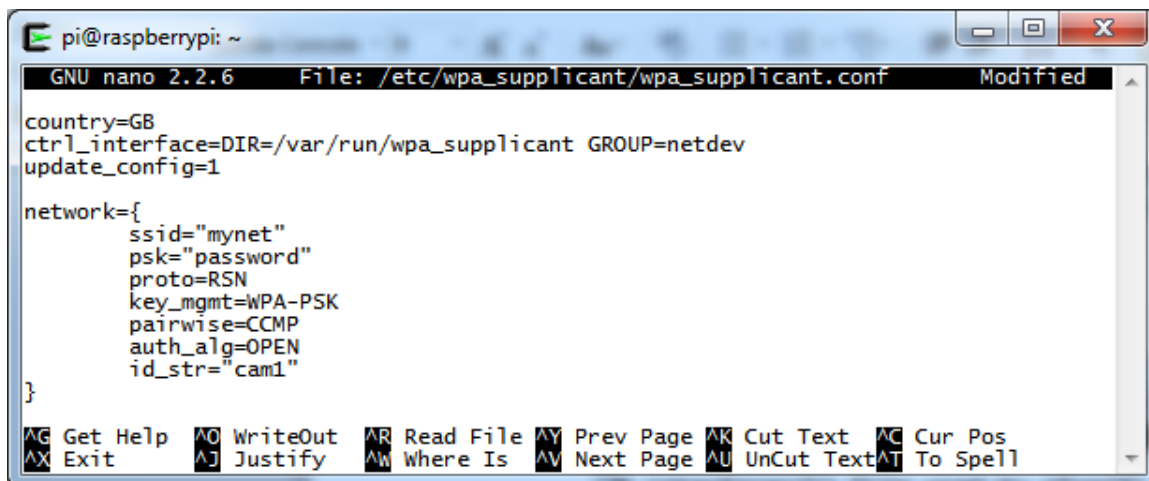
Vispirms, izmantojot komandu "sudo nano /etc/network/interfaces", nokonfigurē statisku adresi tīkla interfeisos (sk. 4.2.att.).



```
pi@raspberrypi: ~
GNU nano 2.2.6 File: /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d
auto lo
iface lo inet loopback
iface eth0 inet manual
auto wlan0
allow-hotplug wlan0
iface wlan0 inet static
address 192.168.1.141
netmask 255.255.255.0
gateway 192.168.1.1
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

4.2. att. Tīkla interfeisa konfigurācija statiskas IP adreses izmantošanai

Pēc tam, izmantojot komandu "sudo nano /etc/wpa_supplicant/wpa_supplicant.conf", nokonfigurē WiFi tīkla parametrus datnē wpa_supplicant.conf, piemēram, sk. 4.3.att.).



```
pi@raspberrypi: ~
GNU nano 2.2.6 File: /etc/wpa_supplicant/wpa_supplicant.conf Modified
country=GB
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="mynet"
    psk="password"
    proto=RSN
    key_mgmt=WPA-PSK
    pairwise=CCMP
    auth_alg=OPEN
    id_str="cam1"
}
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

4.3. att. WiFi tīkla piekļuves parametru konfigurācija

6) jāuzinstalē atvērtā koda automašīnu numura zīmju atpazīšanas bibliotēka *OpenALPR*, kā arī ar to saistītās bibliotēkas (*leptonica*, *opencv*, *tessdata*, *tesseract* u.c.), jāizpilda 'sudo apt-get update && sudo apt-get upgrade', kā arī citas komandas, lai sagatavotu *Raspberry Pi Zero* kustības sensora programmas vajadzībām. Visas nepieciešamās komandas ir apkopotas un ierakstītas vienā *bash* skriptā, kura pilns teksts ir lasāms 2.pielikumā un ko var arī lejupielādēt ar vienu komandu „wget www.haralds.id.lv/openalpr_install_v3.sh”. Pēc skripta

veiksmīgas lejupielādes, to ir nepieciešams nokonfigurēt kā izpildāmu datni, izpildot komandu „`sudo chmod +x openlpr_install_v3.sh`” un palaist darbībā (ar komandu „`./openlpr_install_v3.sh`”).

Visa *OpenALPR* un ar to saistīto bibliotēku uzstādīšanas skripta (sk. `openlpr_install_v3.sh` 2.pielikumā) izpilde (t.sk.bibliotēku *leptonica*, *opencv*, *tessdata*, *tesseract*, *OpenALPR* kompilācija, kas ir vislaikietilpīgākais process) uz *Raspberry Pi Zero* aizņem 19 stundas 10 minūtes, t.sk.nepieciešamo bibliotēku ~1,2GB apjomā lejupielāde no interneta, izmantojot *WiFi* (vidējais lejupielādes ātrums ~200 KB/s), atarhivēšana, kompilācija un instalācija.

Pēc *OpenALPR* instalācijas lietotājs var pārbaudīt automašīnas numura zīmes atpazīšanu, izmantojot paraugattēlu, ko var iegūt no *OpenALPR* mājas lapas ar komandu „`wget http://plates.openalpr.com/ea7the.jpg`” (sk. 4.4.att.).



4.4. att. *OpenALPR* piedāvātais paraugattēls (`ea7the.jpg`) automašīnas numura atpazīšanai

Kā redzams 4.4.att., *OpenALPR* bibliotēka pamatā atbalsta ASV automašīnu numuru atpazīšanu. Rezultātu var iegūt, ievadot komandu „`alpr -c us ea7the.jpg`” (sk. 4.5.att.).

```
pi@raspberrypi: ~/ocr/in
pi@raspberrypi:~/ocr/in $ alpr -c us ea7the.jpg
Warning: You are running an unsupported version of Tesseract.
Expecting at least 3.03, your version is: f566a45
Info in bmfCreate: Generating pixa of bitmap fonts from string
plate0: 10 results
- EA7THE confidence: 91.0578
- EA7TBE confidence: 84.133
- EA7T8E confidence: 83.0083
- EA7TRE confidence: 82.7869
- EA7TE confidence: 82.5961
- EA7TME confidence: 80.2908
- EA7TH6 confidence: 77.0045
- EA7THB confidence: 75.5779
- EA7TH confidence: 74.6576
- EA7TB6 confidence: 70.0797
pi@raspberrypi:~/ocr/in $
```

4.5. att. OpenALPR automašīnas numura atpazīšanas rezultāts, izmantojot attēlu 'ea7the.jpg'

Kā redzams 4.5.att., *OpenALPR* bibliotēka atpazīst ASV reģistrētas automašīnas numuru ar augstu ticamību (91.0578%). Ja šo rezultātu ierakstītu *LOG* datnē, tad, izmantojot papildus rīkus, no visa teksta vēl būtu jāizvelk tikai automašīnas numurs, kuram ir visaugstākā ticamība (pirmā rinda), un jāieraksta tas datu bāzē vai teksta datnē, piemēram, *CSV*, ko vēlāk varētu importēt jebkurā datu bāzē.

4.2. Tīmekļa servera un mājas lapas risinājums

Tīmekļa servera palaišana programmēšanas valodā *Python* ir ļoti vienkārša. *Python 3* pēc noklusējuma piedāvā *SimpleHTTPServer* moduli, kurā ir definēta viena klase *SimpleHTTPRequestHandler*, kas ir saderīga ar *BaseHTTPServer.BaseHTTPRequestHandler* [2]. Piemēram, tīmekļa servera palaišanu darbībā, izmantojot saknes direktoriju [/home/pi/web](#) un portu 8080, var paveikt ar vienkāršu 4 rindu kodu (sk. 4.6.att.).

```
import http.server, os
os.chdir („/home/pi/web“)
httpd = http.server.HTTPServer(('127.0.0.1', 8080), http.server.SimpleHTTPRequestHandler)
httpd.serve_forever()
```

4.6. att. Vienkāršs Python kods tīmekļa servera palaišanai

Ja saknes direktorijā [/home/pi/web](#) neatrodas datne *index.html*, tad tīmekļa lapā tiek attēlots šīs saknes direktorijas saturs. Uzrakstot vienkāršu *index.html* (sk. 4.7.att.), tīmekļa pārlūkā pēc adreses <http://localhost:8080/> var pārlicināties par tā attēlojumu (sk. 4.8.att.).

```

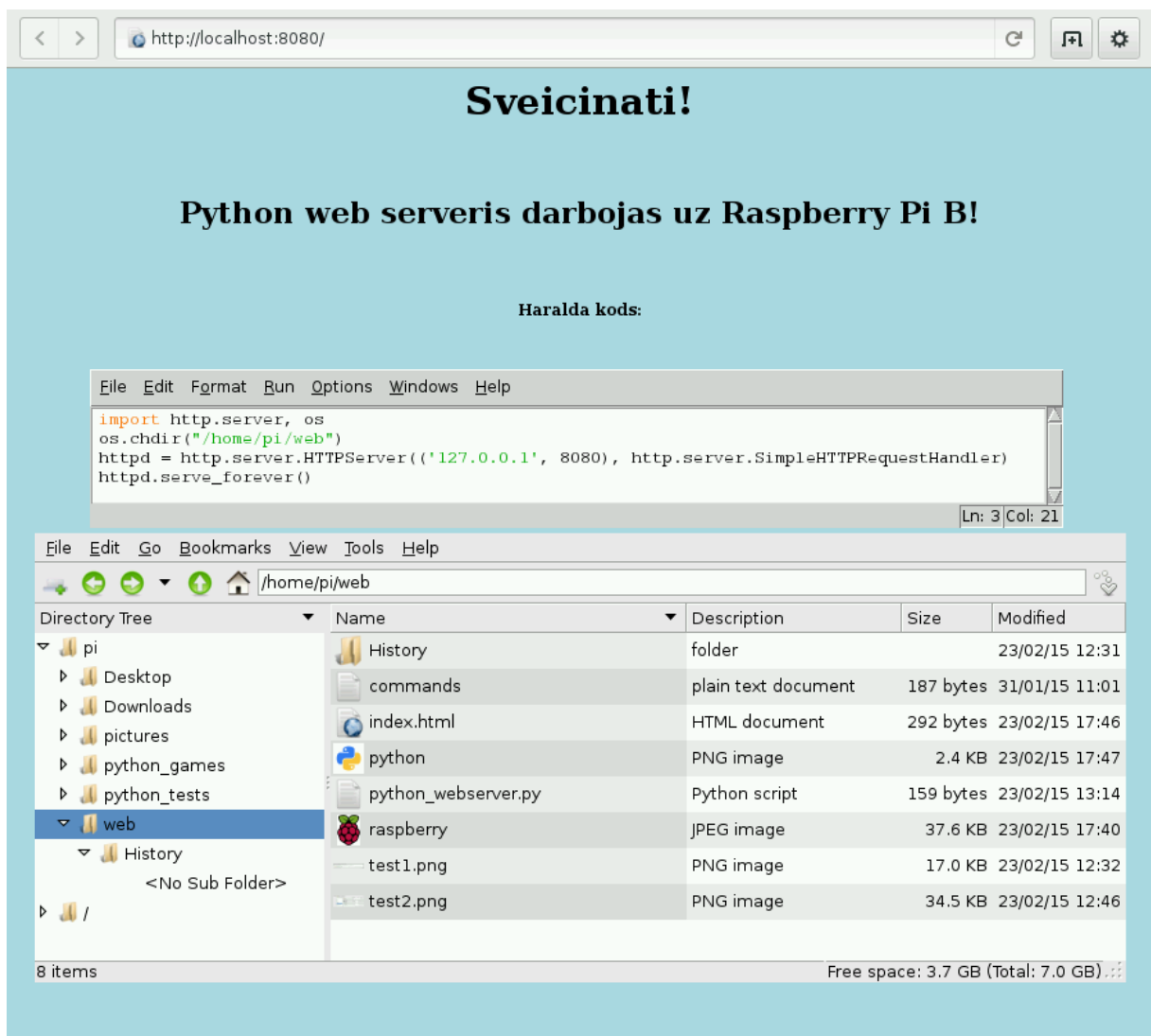
File Edit Search Options Help
<html>
<head>
<title>Sveicinati!</title>
</head>
<body bgcolor="lightblue" text="black">
<center><h1>Sveicinati!</h1>
<br>
<h2>Python web serveris darbojas uz Raspberry Pi B!</h2>
<br>
<b>
<h5>Haralda kods:</h5>|
<br>

<br>

</center>
</body>
</html>

```

4.7. att. Vienkāršs index.html datnes kods



4.8. att. Vienkārša index.html attēlojums, izmantojot Python tīmekļa serveri

Ņemot vērā, ka maģistra darba uzdevums ir saistīts ar kameras izmantošanu (skat.risinājuma realizāciju nākamajā apakšnodaļā), tad, lai izstrādātu tīmekļa serveri un mājas lapu, kura atjaunojas 2 sekunžu intervālā un parāda aktuālāko kameras uzņemto attēlu, var izmantot *Python* web ietvaru *Tornado*⁸⁴, kas ir sākotnēji *FriendFeed* izstrādātais web ietvars un asinhronā tīkla bibliotēka. Izmantojot ne-bloķējošu tīkla I/O (ievad-izvades) iespēju, *Tornado* var izmantot līdz desmitiem tūkstošu atvērto savienojumu, padarot to gandrīz ideālu ilgam pieprasījumu procesam, *WebSockets* un citām programmām, kas prasa ilgu aktīvu savienojumu katram lietotājam⁸⁵.

Pieņemot, ka nākamajā nodaļā izstrādātā risinājuma kods pēc kustības noteikšanas aktuālāko attēlu **.jpg* formātā ieraksta direktoriājā `"/home/pi/record/mol/"` datnē `live.jpg`, tad realizēt mājas lapu, kas atjaunojas 2 sekunžu intervālā un parāda aktuālāko kameras uzņemto attēlu, var ar pavisam vienkāršu *Python* kodu (sk. 4.9.att.).

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import tornado.ioloop
import tornado.web
import subprocess

class MainHandler(tornado.web.RequestHandler):
    def get(self):
        self.write('<!DOCTYPE html><head><title>' +
            'Kameras attēls</title>' +
            '<META HTTP-EQUIV="refresh"' +
            'CONTENT="5"></head><body>' +
            '</body>')

class ImageHandler(tornado.web.StaticFileHandler):
    def set_extra_headers(self, path):
        self.set_header('Cache-Control',
            'no-store, no-cache, must-revalidate' +
            ' max-age=0')

application = tornado.web.Application([
    (r"/", MainHandler),
    (r"/home/pi/record/mol/(.*)", ImageHandler,
    {"path": "/home/pi/record/mol/"})]

if __name__ == "__main__":
    # subprocess.Popen(["python3", "record.py"])
    application.listen(8888)
    tornado.ioloop.IOLoop.instance().start()
```

4.9. att. *Python* kods (web.py) aktuālākā attēla parādīšanai, izmantojot *Python Tornado* ietvaru
Web serveri iedarbināt un palaist kodu darbībā var ar vienkāršu komandu (sk. 4.10.att.).

```
pi@raspberrypi:~/record $ python web.py
```

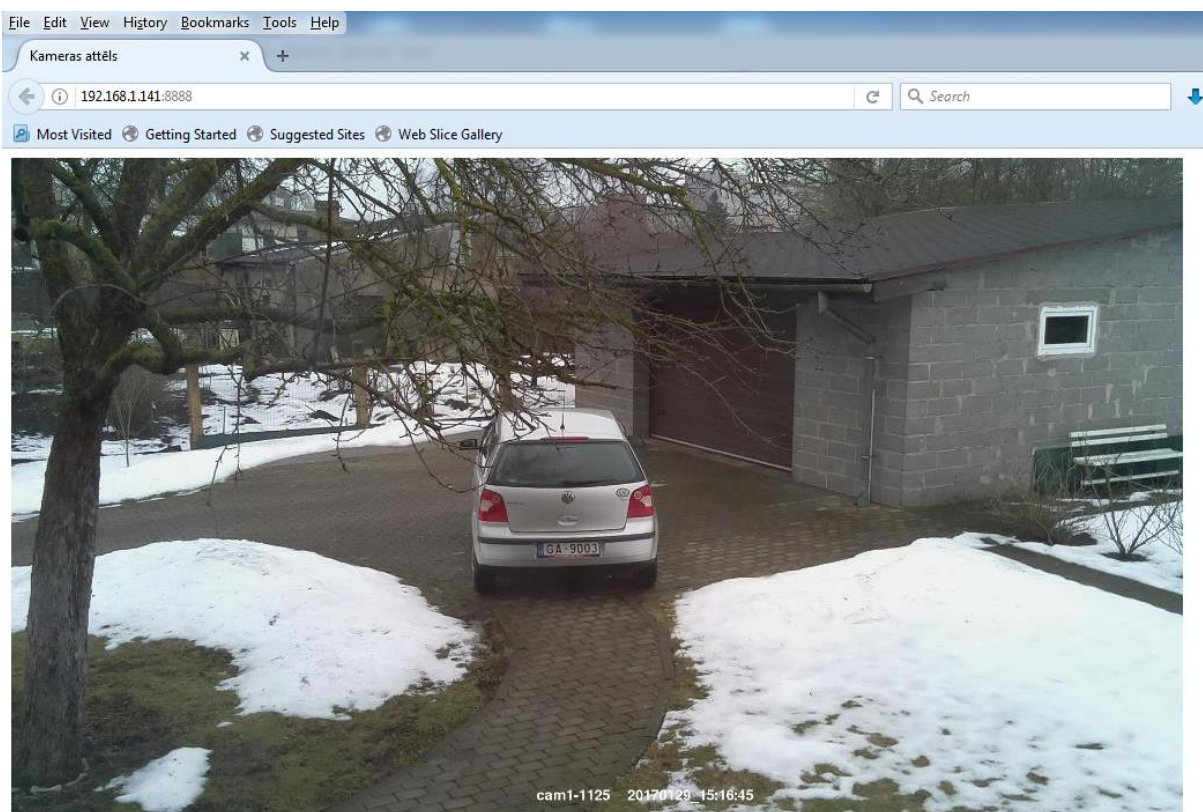
4.10. att. Konsoles komanda mājas lapas (web.py) palaišanai darbībā

Rezultātā, ņemot vērā, ka *Raspberry Pi Zero* ir uzkonfigurēts uz statiskas IP adreses (192.168.1.141), mājas lapu ar rezultātu – aktuālāko kameras uzņemto attēlu – var paskatīt no jebkura cita iekšējā

⁸⁴ <https://pypi.python.org/pypi/tornado/3.2.1>

⁸⁵ <https://pypi.python.org/pypi/tornado/3.2.1>

tīklā esoša datora, planšetes vai citas ierīces ar interneta pārlūku, ievadot tajā adresi <http://192.168.1.141:8888/> (sk. 4.11.att.).



4.11. att. Mājas lapa ar aktuālāko kameras uzņemto attēlu

Atbilstoši *Python* kodam (*web.py*) mājas lapa atjaunojas ik pēc 5 sekundēm, parādot jaunāko kameras uzņemto attēlu, kas ierakstīts datnē *live.jpg* (pilns atrašanās ceļš: `"/home/pi/record/mol/live.jpg"`). Līdz ar to nākamais solis ir uzprogrammēt kustības sensoru – programmu, kas iegūst aktuālāko kameras uzņemto attēlu un ieraksta to datnē *live.jpg* tikai tad, kad kadrā ir notikusi kustība.

4.3. Kustības sensors

Lai iegūtu aktuālāko kameras uzņemto attēlu un ierakstītu to attiecīgajā datnē tikai tad, kad kadrā ir notikusi kustība (sk. *live.jpg* iepriekšējā apakšnodaļā), ir jāveic mazliet komplicētāki programmēšanas darbi. Lai uzprogrammētu kustības sensoru, kas darbojas, analizējot kameras uzņemtos attēlus, vispirms ir attiecīgi jā sagatavo aparātūra. *Raspberry Pi Zero* ir nepieciešams aprīkot ar papildus iekārtu – video kameru, kas ir papildus izmaksas aptuveni 10 eiro apmērā (sk. 2.2. tab.), ja iegādājas Ķīnā ražotu moduli, kas ir līdzvērtīgs oriģinālajam, vismaz 25 eiro vērtajam modulim. Var izvēlēties arī USB video kameru, taču šī maģistra darba ietvaros ir izvēlēts *Raspberry Pi Camera* modulis (sk. 4.12.att.), jo:

- ✓ tas piedāvā augstu izšķirtspēju (5 megapikseļi), nodrošinot attēla kvalitāti 2592 x 1944 pikseļi un video kvalitāti 1080p30, 720p60 un 640x480p60/90;
- ✓ mazāku enerģijas patēriņu;

- ✓ to var ērti pievienot *Raspberry Pi*, izmantojot CSI savienojumu;
- ✓ tas ir maza izmēra modulis (tikai 25mm x 20mm x 9mm);
- ✓ tā svars ir tikai 3g;
- ✓ to kopā ar *Raspberry Pi* var darbināt, izmantojot baterijas (mobilitātes kritērijs).

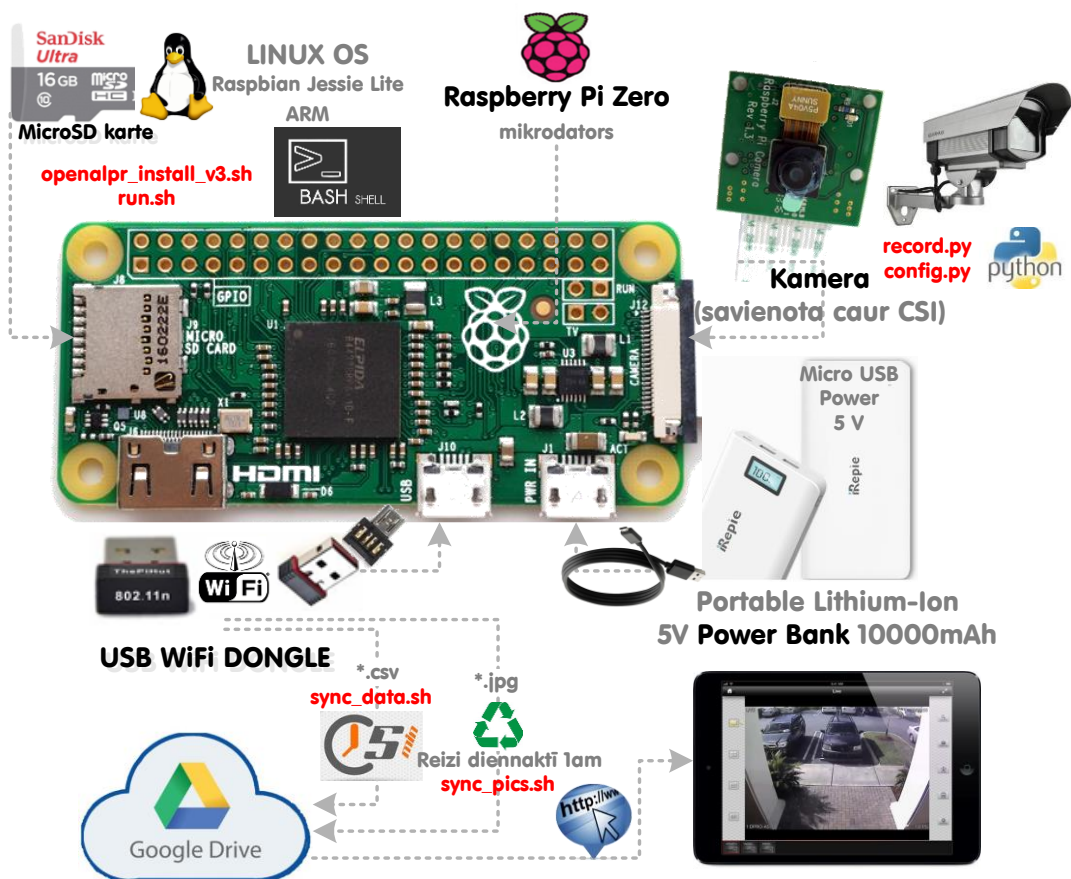


4.12. att. **Raspberry Pi Camera modulis**

Pirms sākt programmēt kaut ko pilnīgi no jauna noder papētīt, kādi līdzīgi risinājumi pasaulē jau ir pieejami, un viens no lieliskiem avotiem ir *GitHub*⁸⁶, kas ir tīmeklī balstīts *Git* repositorijs, kurā bieži var atrast dažādus, labus atvērtā koda risinājumus, t.sk. arī priekš *Raspberry Pi* iekārtām. Viens no tādiem risinājumiem ar tuvu līdzību maģistra darbā noteikto uzdevumu realizācijai ir *pi-timolo*⁸⁷. Mazliet modificējot kodu un izdzēšot lieko, nevajadzīgo koda daļu, ievērojot *pi-timolo* risinājuma atvērtā koda licences (*MIT License*) nosacījumus, var sagatavot maģistra darbā noteiktam risinājumam nepieciešamo programmu – kodu *record.py* (sk. 3.pielikumā), kas izmanto datnē *config.py* ierakstītus konfigurācijas parametrus (sk. 4.pielikumā). Šī programma gan nav vienīgā, lai realizētu visu risinājumu pilnībā (sk. 4.13.att.). Lai programma darbotos uzreiz pēc *Raspberry Pi Zero* ieslēgšanas, ir nepieciešams tās automātiskas palaišanas skripts no komandrindas vai ar attiecīgu ierakstu `/etc/rc.local` datnē (sk. *run.sh* 4.13.att. un 5.pielikumā). Lai notiktu kustības sensora uzņemto attēlu sinhronizācija ar *Google Drive*, ir nepieciešams attēlu sinhronizācijas skripts (sk. *sync_pics.sh* 4.13.att. un 6.pielikumā), kā arī attiecīgs sinhronizācijas laika intervāla uzstādījums `crontab`. Lai notiktu katra kustības sensora uzņemtā attēla apstrāde, atpazīšana, vai tajā ir redzama automašīnas numura zīme, un, ja ir, tad automašīnas numura atpazīšana, ierakstīšana teksta datnē `*.csv` formātā un sinhronizācija ar *Google Drive*, ir nepieciešams attiecīgs skripts, kas veic visas šīs darbības (sk. *sync_data.sh* 4.13.att. un 7.pielikumā).

⁸⁶ <https://github.com/>

⁸⁷ <https://github.com/pageauc/pi-timolo/wiki/Introduction>



4.13. att. Kustības sensora risinājums ar automašīnu plūsmas reģistru

Lai realizētu 4.13.att. redzamo risinājumu, proti, reizi diennaktī notiktu attēla datņu *JPG* sinhronizācija ar *Google Drive*, bet ik pēc 5 minūtēm notiktu datu *CSV* formātā sinhronizācija ar *Google Drive*, ir nepieciešams ieplānot šo noteikto uzdevumu, realizētu *Bash* skripta veidā, izpildi noteiktā laikā vai ar noteiktu laika intervālu, kam lieliski noder *Linux* sistēmas dēmons *Cron* – fona programma, kas vajadzības gadījumā tiek izmantota datoros, kuri darbojas *UNIX* vai kādas citas operētājsistēmas vidē⁸⁸ [12]. Par *crontab* uzstādījumu attēlu sinhronizācijas skriptam un datu sinhronizācijas skriptam var attiecīgi lasīt šī maģistra darba 4.2.2.apakšnodaļas un 4.5.nodaļas beigās.

4.3.1. Programmas iestatīšana darbībai fonā pēc *Raspberry Pi* ieslēgšanas

Lai kustības sensora (aktuālā attēla iegūšanas, ja notikusi kustība) programma *record.py* darbotos fonā un uzreiz pēc iekārtas *Raspberry Pi Zero* ieslēgšanas, ir nepieciešams to automātiski palaist līdz ar iekārtas iesāknēšanos – ieslēgšanos. Ņemot vērā, ka *Raspberry Pi Zero* darbināšanai ir izvēlēta *Linux* operētājsistēmas distribūcija *Raspbian Jessie Lite* (versija 2017-03-02), tad to var izdarīt, izmantojot direktoriju */etc/init.d*, kurā ir jāiekopē programmas *record.py* palaišanas skripts *run.sh* (sk. 5.pielikumā) un jāveic tā uzstādīšana, izmantojot 5 komandas (sk. 4.14.att.).

⁸⁸ <http://termini.lza.lv/term.php?term=daemon&lang=EN>

```
sudo cp run.sh /etc/init.d
cd /etc/init.d
chmod +x run.sh
sudo nano run.sh
# izmainīt ierakstu, kas norāda uz programmas record.py atrašanās vietu un saglabāt datni
# Inicializē /etc/init.d/run.sh skriptu tā, lai tas izpildītos pēc iekārtas ieslēgšanas.
sudo update-rc.d pi-timolo.sh defaults
```

4.14. att. Komandas kustības sensora iedarbināšanai uzreiz pēc iekārtas ieslēgšanas

Tagad kustības sensors darbosies uzreiz pēc iekārtas iesāknēšanās, pārstartēšanas, pārlādēšanas, arī enerģijas padeves pārtraukšanās un atjaunošanās gadījumos.

4.2.2. Attēlu sinhronizācija ar Google Drive

Kāpēc *Google Drive*? Galvenais iemesls *Google Drive* izmantošanai šī maģistra darba ietvaros ir bezmaksas serviss, ko piedāvā kompānija *Google*.

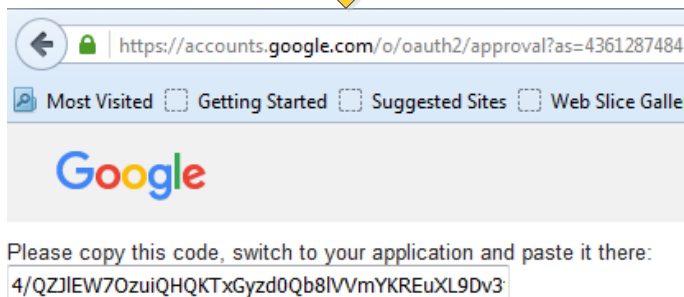
Lai realizētu kustības sensora – programmas *record.py* – uzņemto attēlu sinhronizāciju ar *Google Drive* (sk. 4.16.att.), ir nepieciešams:

- 1) izveidot *Gmail* kontu vai izmantot jau esošu kontu, kas ir bezmaksas;
- 2) uzstādīt *gdrive sync* (programmas binārais kods arī ir pieejams bez maksas);
- 3) izstrādāt un pievienot *crontab* attēlu sinhronizācijas skriptu (sk. *sync_pics.sh* 4.13.att. un 6.pielikumā), izvēloties noteiktu attēlu sinhronizācijas laika intervālu.

Pieņemot, ka lietotājam jau ir *gmail* konts, var sākt ar otro soli. Programmas *gdrive* binārais kods ir pieejams *GitHub*⁸⁹ un to ir nepieciešams uzkompilēt vai arī paļauties un izmantot jau gatavu *gdrive*, ko arī var atrast *GitHub*. Šis fails ir jāiekopē uz *Raspberry Pi* (var jau iepriekš izveidotā direktoriijā *record*), jāpadara izpildāms (komanda `chmod +x gdrive`) un jāizpilda komanda ‘`sudo gdrive init`’, kas inicializē servisu un uzģenerē *URL*, kas ir jāiekopē interneta pārlūkā ar autorizētu *gmail* kontu. Tālāk *Google* parādītais kods ir jāiekopē atpakaļ *Raspberry Pi SSH* sesijā, *gdrive* autorizācijas koda rindā (sk. 4.15.att.). To visu vieglāk un vienkāršāk ir izdarīt no *Windows*, izmantojot *SSH* un rīku *Cygwin* vai *PuTTY*, lai pieslēgtos pie *Raspberry Pi*.

⁸⁹ <https://github.com/odeke-em/drive/releases/tag/v0.2.8-arm-binary>

```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ sudo gdrive init  
Visit this URL to get an authorization code  
https://accounts.google.com/o/oauth2/auth?access_type=offline&client_id=3547909620747rrlnuanmamggi4feed12dpuq871bvd.apps.googleusercontent.com&redirect_uri=urn%3Aietf%Awg%3Aoauth%3A2.0%3Aaob&response_type=code&scope=https%3A%2F%2Fwww.googleapis.com%2Futh%2Fdrive&state=2017-03-30+12%3A32%3A25.202747918+%2B0300+EEST2596996162  
Paste the authorization code:
```



```
pi@raspberrypi ~/installs $ gdrive init  
Visit this URL to get an authorization code  
https://accounts.google.com/o/oauth2/auth?access_type=offline&client_id=3547909620747rrlnuanmamggi4feed12dpuq871bvd.apps.googleusercontent.com&redirect_uri=urn%3Aietf%Awg%3Aoauth%3A2.0%3Aaob&response_type=code&scope=https%3A%2F%2Fwww.googleapis.com%2Futh%2Fdrive&state=2017-03-30+12%3A32%3A25.202747918+%2B0300+EEST2596996162  
Paste the authorization code: 4/QZJIEW7OzuiQHKTxGyzd0Qb8lVvmYKREuXL9Dv3  
pi@raspberrypi ~/installs $
```

4.15. att. Gdrive uzstādīšana

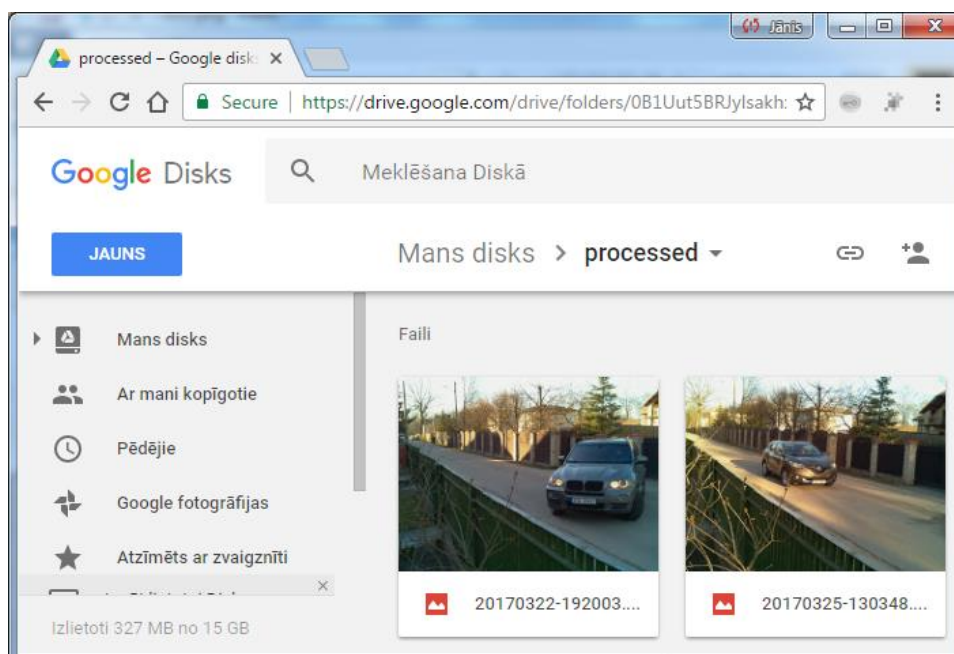
Par *Google Drive* veiksmīgu uzstādīšanu var pārliecināties ne tikai pēc neesošiem kļūdu paziņojumiem pēc enter/ok nospiešanas terminālī, bet arī pēc izveidotās slēptās direktorijas `‘.gd’`, kas satur drošības datnes un ko var redzēt, izpildot komandu `‘ls -al’`. *Gdrive* palīdzība ir pieejama, vienkārši ievadot terminālī komandu `‘gdrive’` bez parametriem. Savukārt komandu `‘sudo gdrive ls’` var izmantot, lai paskatītu attālināto saturu (*Google Drive* saknes direktorijas saturu). Ja kustības sensora programmas konfigurācijas datnē `config.py` ir uzstādījums `‘createLockFile = True’`, tad `record.sync` datne tiks izveidota tiklīdz būs notikusi kustība un uzģenerēti attēli direktorijā `‘motionDir’` (skat.šo parametru datnē `config.py`). Attēlu sinhronizācija ar *Google Drive* nenotiks, ja kustības sensora programmas direktorijā nebūs atrodama datne `record.sync`. Turklāt attēlu datnēm `*.jpg` formātā ir jau jābūt apstrādātām, atpazītām ar *OpenALPR* algoritmu, ja tajās ir redzama automašīnas numura zīme, un pārvietotām uz direktoriju `‘processed’` (sk.4.4.nodaļu), kas saskaņā ar 4.13. att.notiek ik pēc 5 minūtēm.

```

pi@raspberrypi:~
pi@raspberrypi:~ $ ls -lah ./record/mol
total 6.8M
drwxr-xr-x 2 pi pi 4.0K Mar 30 11:39 .
drwxr-xr-x 5 pi pi 4.0K Mar 28 16:28 ..
-rw-r--r-- 1 pi pi 725K Mar 22 17:25 20170322-172502.jpg
-rw-r--r-- 1 pi pi 720K Mar 22 17:34 20170322-173421.jpg
-rw-r--r-- 1 pi pi 677K Mar 22 17:59 20170322-175955.jpg
-rw-r--r-- 1 pi pi 700K Mar 22 18:11 20170322-181144.jpg
-rw-r--r-- 1 pi pi 655K Mar 22 19:10 20170322-191050.jpg
-rw-r--r-- 1 pi pi 672K Mar 22 19:20 20170322-192003.jpg
-rw-r--r-- 1 pi pi 625K Mar 22 20:06 20170322-200646.jpg
-rw-r--r-- 1 pi pi 625K Mar 22 20:07 20170322-200715.jpg
-rw-r--r-- 1 pi pi 787K Mar 25 13:25 20170325-132545.jpg
-rw-r--r-- 1 pi pi 723K Mar 27 10:41 20170327-104111.jpg
-rw-r--r-- 1 root root 51 Mar 28 16:28 2017-03-28_log.csv
-rw-r--r-- 1 root root 2.4K Mar 27 10:28 .csv
pi@raspberrypi:~ $

```

sync_pics.sh



4.16. att. Attēlu sinhronizācija ar Google Drive

Lai nepārslogotu *Raspberry Pi Zero* mikrodatoru, attēlu datņu **.jpg* sinhronizāciju ar *Google Drive* ir paredzēts veikt reizi diennaktī. Manuāli attēlu datņu sinhronizācijas skriptu var izsaukt ar komandu ‘`sudo ./sync_pics.sh`’, atrodoties šī skripta direktorijā. Rezultātā notiks visu **.jpg* datņu, kas atrodas direktorijā ‘*processed*’ augšupielāde – sinhronizācija ar – *Google Drive* (sk. 4.16.att.). Aptuveni 6,75 MB vai 10 *JPG* formāta datnes var augšupielādēt *Google Drive* 1 minūtes laikā, izmantojot *WiFi* (skat. skripta `sync_pics.sh` izvaddatus 4.17.att.).

```

pi@raspberrypi:~ $ sudo ./record/sync_pics.sh
-----
STATUS - Script Variable Setting CHECK_FOR_SYNC_FILE=true
STATUS - Found File /home/pi/record/record.sync
STATUS - Check if Remote Folder /processed Exists
-----
"/processed" does not exist remotely "/processed" not matched
-----
STATUS - Creating Remote Folder /processed
-----
/processed 0B1Uut5BRJy1sakhxUEdFRV1LS1U

```

```

FileId                                     Relative Path
"OB1Uut5BRJylsakhUEDFRVlLSlU"          "/processed"
-----
STATUS - Successfully Created Remote Folder /processed
-----
STATUS - Start gdrive Sync ....
STATUS - Local Source Folder - /home/pi/record/processed
STATUS - Remote Destn Folder - /processed
STATUS - Files *jpg
STATUS - Running This May Take Some Time .....
STATUS - sudo /usr/local/bin/gdrive push -no-prompt -ignore-conflict
processed/*jpg
-----
Thu 30 Mar 11:59:15 EEST 2017
Resolving...
+ /processed/20170322-192003.jpg
+ /processed/20170325-130348.jpg
+ /processed/20170325-132535.jpg
+ /processed/20170325-132545.jpg
+ /processed/20170327-100326.jpg
+ /processed/20170327-101125.jpg
+ /processed/20170327-103634.jpg
+ /processed/20170327-104111.jpg
+ /processed/20170327-170042.jpg
+ /processed/20170328-162701.jpg
Addition count 10 src: 6.95MB
7284574 / 7284574
[=====
Thu 30 Mar 12:00:14 EEST 2017
-----
STATUS - ./record/sync_pics.sh Completed Successfully
STATUS - Processing Took 59 seconds
STATUS - Deleting Sync Lock File /home/pi/record/record.sync
Done ...

```

4.17. att. Google Drive sinhronizācijas skripta izvad dati

Lai attēlu sinhronizācijas skripts `sync_pics.sh` darbotos regulāri, piemēram, katru diennakti plkst.1am, ir nepieciešams attiecīgs tā palaišanas ieraksts un laika intervāla uzstādījums `crontab`, piemēram, `'00 01 * * * /home/pi/record/sync_pics.sh >/dev/nul'`, kas jāieraksta kā rinda teksta redaktorā *nano*, kas atveras pēc komandas `'sudo crontab -e'` izpildes.

4.4. Automašīnu numuru atpazīšana

Pirms attēlu datņu `*.jpg` sinhronizācijas ar *Google Drive*, ir nepieciešams uzinstalēt atvērtā koda automašīnu numura zīmju atpazīšanas bibliotēku *OpenALPR*, kā arī ar to saistītās bibliotēkas (*leptonica*, *opencv*, *tessdata*, *tesseract* u.c.), jāizpilda `'sudo apt-get update && sudo apt-get upgrade'`, kā arī citas komandas, lai sagatavotu vidi automašīnas numura teksta atpazīšanai, kas ir diezgan darbietilpīgs process. Lielākoties tas jau ir aprakstīts 4.1.nodaļas 6.solī. Visas nepieciešamās komandas ir apkopotas un ierakstītas vienā *bash* skriptā, kura pilns teksts ir lasāms 2.pielikumā un ko var arī lejupielādēt ar vienu komandu `'wget www.haralds.id.lv/openalpr_install_v3.sh'`. Pēc skripta veiksmīgas lejupielādes, to ir nepieciešams nokonfigurēt kā izpildāmu datni, izpildot komandu `'sudo chmod +x openalpr_install_v3.sh'` un palaist darbībā (ar komandu `'./openalpr_install_v3.sh'`).

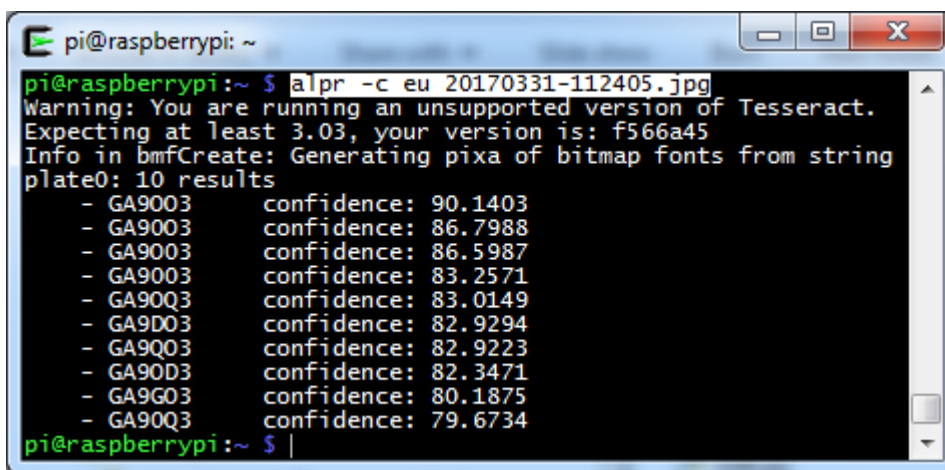
Kad *OpenALPR* un visas ar to saistītās bibliotēkas ir uzinstalētas un automašīnas numura zīmes atpazīšana darbojas ar *OpenALPR* piedāvāto izmēģinājuma datni (sk. 4.5.att.), var sākt risinājuma pielāgošanu Latvijas automašīnu numura zīmju atpazīšanai.

Pēc *OpenALPR* instalācijas direktorijas izpētes var secināt, ka pastāv *OpenALPR* konfigurācijas datne Eiropas numura zīmēm *eu.conf*, ko pēc *OpenALPR* sekmīgas uzinstalēšanas var atrast, izpildot komandu ‘`ls /usr/local/share/openalpr/runtime_data/config/`’ (sk. 4.18.att.).

```
pi@raspberrypi:~$ ls /usr/local/share/openalpr/runtime_data/config/
au.conf  auwide.conf  br.conf  eu.conf  fr.conf  gb.conf  kr2.conf  kr.conf  mx.conf  sg.conf  us.conf  vn2.conf
pi@raspberrypi:~$
```

4.18. att. *OpenALPR* atbalstīto valstu auto numura zīmju konfigurācijas datnes

Līdz ar to, komandrindā ‘`alpr`’ nomainot parametru ‘`us`’ (Amerikas numura zīme) uz ‘`eu`’, kas apzīmē Eiropas numura zīmju atpazīšanas parametrus, un kā attēla datni norādot *JPG* ar redzamu Latvijas automašīnu (piemēram, sk. 4.11.att.), var iegūt ticamu rezultātu (sk. 4.19.att.).



```
pi@raspberrypi:~$ alpr -c eu 20170331-112405.jpg
Warning: You are running an unsupported version of Tesseract.
Expecting at least 3.03, your version is: f566a45
Info in bmfCreate: Generating pixa of bitmap fonts from string
plate0: 10 results
- GA9003 confidence: 90.1403
- GA9003 confidence: 86.7988
- GA9003 confidence: 86.5987
- GA9003 confidence: 83.2571
- GA9003 confidence: 83.0149
- GA9D03 confidence: 82.9294
- GA9Q03 confidence: 82.9223
- GA90D3 confidence: 82.3471
- GA9G03 confidence: 80.1875
- GA90Q3 confidence: 79.6734
pi@raspberrypi:~$
```

4.19. att. *OpenALPR* izvaddati (automašīnas numurs) attēlam ar Latvijas automašīnu

Kā redzams 4.19.att., *OpenALPR* bibliotēka atpazīst Latvijā reģistrētas automašīnas numuru ar augstu ticamību (90.1403%). Šo rezultātu atliek vien ierakstīt *LOG* datnē, apstrādāt un kopā ar datnes nosaukumu, kas norāda uz attēla uzņemšanas datumu un laiku, ierakstīt datu bāzē vai teksta datnē, piemēram, *CSV*, ko vēlāk varētu importēt jebkurā datu bāzē.

4.5. Automašīnu plūsmas reģistrs

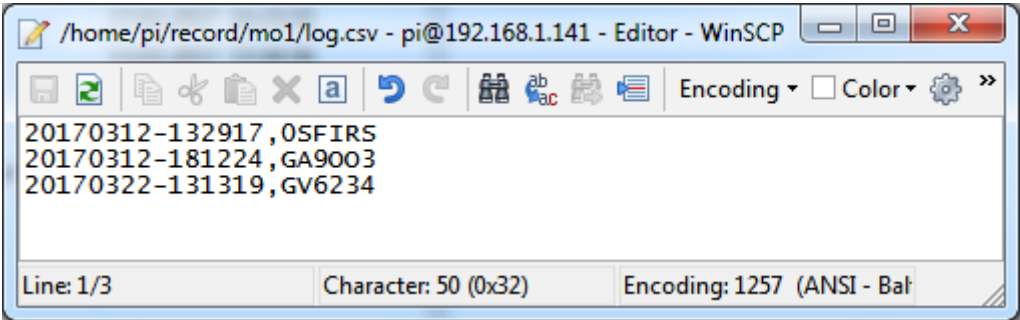
Kā viena veida automašīnu plūsmas reģistrs saskaņā ar risinājuma arhitektūru (sk. 4.13.att.) jau kalpo *Google Drive* redzamie attēli, kuru nosaukums glabā attēla uzņemšanas laiku, t.i., veidots formātā ‘*yyyymmdd-hhmmss*’ (sk. 4.16.att.). Tomēr tie ir tikai attēli un informācija par attēla uzņemšanas laiku, pieņemot, ka *Raspberry Pi* kameras tehnoloģiskās iespējas nenodrošina vairāku attēlu uzņemšanu sekundē un teritorijas novērošanas nolūkos biežāks uzņemto kadru intervāls kā sekunde nav nepieciešams, jo automašīnu plūsma pie barjeras, kur vienmēr notiek apstāšanās, lai iebrauktu teritorijā, nemaz nav intensīvāka, lai veidotos vairāku automašīnu kadri vienā sekundē. *Google Drive* redzamie

attēli nav strukturēta informācija, kas satur tekstu par automašīnas numuru. Līdz ar to ir nepieciešams uzrakstīt skriptu, kas apstrādā 4.19.att. redzamos izvaddatus, piemēram, saglabā izvaddatus *TXT* datnē, apstrādā tos, izgriežot automašīnas numura datus no otrās rindas, un kopā ar datnes nosaukumu, kas norāda uz attēla uzņemšanas datumu un laiku, ieraksta tos *CSV* datnē, ko vēlāk varētu importēt jebkurā datu bāzē. Šāda skripta pilns teksts, kas veic arī *CSV* datnes augšupielādi *Google Drive*, ir lasāms 7.pielikumā. Galvenā koda daļa, kas, izmantojot visas *JPG* datnes, atrodamas mainīgjos (*DIR* un *SYNC_DIR*) nodefinētajā direktoriijā, atpazīst attēlos redzamo automašīnu numurus, ieraksta izvaddatus *TXT* datnē, veic *TXT* datnes apstrādi, izgriežot informāciju par auto numuru, un ieraksta to kopā ar *JPG* datnes nosaukumu ‘yyyymmdd-hhmmss’ formātā jaunizveidotajā *CSV* datnē, kuras nosaukums sastāv no attiecīgās dienas datuma formātā ‘yyyy-mm-dd_log.csv’, ir lasāma 4.20.att.

```
for file in "$DIR/$SYNC_DIR/"*.jpg; do
# iegūst datnes nosaukumu bez paplašinājuma un ieraksta mainīgajā full_fn
full_fn=${file##*/}
# iegūst tikai datnes nosaukumu bez paplašinājuma un ceļa uz datni, ieraksta mainīgajā fn
fn=${full_fn%.*}
# atpazīst automašīnas numuru attēlā un ieraksta rezultātu *.txt datnē
alpr -c eu $file > $DIR/$SYNC_DIR/$fn.txt
# izgriež datus par auto numuru no OpenALPR izvaddatnes *.txt 2 rindas
car_num=$(sed -r 's|.*- (.*)confidence.*|\1|' $DIR/$SYNC_DIR/$fn.txt | sed '2q;d')
# izveido datni log.csv, izmantojot šodienas datumu kā prefiksu formātā Y-M-D
# (>, ja tāda neeksistē) un ievada/pievieno (>>, ja eksistē) datus par auto numuru
nowdt=$(date +%Y-%m-%d_%H-%M-%S")
now=$(date +%Y-%m-%d")
log_file="$now"_log.csv
if [ ! -e $DIR/$SYNC_DIR/$log_file ]; then
echo $fn,$car_num > $DIR/$SYNC_DIR/$log_file
else
echo $fn,$car_num >> $DIR/$SYNC_DIR/$log_file
fi
# pārvieto apstrādāto attēla failu un izvaddatus *.txt uz direktoriju $PROC_DIR
mv $file $DIR/$PROC_DIR/
mv $DIR/$SYNC_DIR/$fn.txt $DIR/$PROC_DIR/
done
```

4.20. att. Koda daļa *OpenALPR* izvaddatu apstrādei - automašīnas numura datu ieguvei

4.20. att. redzamās koda daļas izpildes rezultātā, ja mainīgjos (*DIR* un *SYNC_DIR*) nodefinētajā direktoriijā atrodas *JPG* datnes, kas satur automašīnas numuru, izveidojas *CSV* datne, kas satur informāciju par 1) automašīnas iebraukšanas laiku formātā ‘yyyymmdd-hhmmss’ un 2) automašīnas numuru, kas iegūts, atpazīstot to no kustības sensora fiksēta attēla datnes *JPG* formātā, izmantojot *OpenALPR*, kā arī ar to saistītās bibliotēkas (*leptonica*, *opencv*, *tessdata*, *tesseract* u.c.), kas nepieciešamas komandas ‘alpr’ darbībai. Izveidotās *CSV* datnes piemērs ar minēto informāciju redzams 4.21.att.



```
/home/pi/record/mo1/log.csv - pi@192.168.1.141 - Editor - WinSCP
20170312-132917,0SFIRS
20170312-181224,GA9003
20170322-131319,GV6234
Line: 1/3 Character: 50 (0x32) Encoding: 1257 (ANSI - Bal)
```

4.21. att. CSV datne ar tajā ierakstītu automašīnas iebraukšanas laika un numura informāciju
Pilns skripta teksts, kas lasāms 7.pielikumā, veic ne tikai attēlu atpazīšanu, CSV datnes izveidi, bet arī tās sinhronizāciju ar *Google Drive*.

Līdzīgi kā attēlu singronizācijas skripta gadījumā (sk. 4.16.att.un 4.17.att.) atliek tikai uzstādīt 7.pielikumā lasāmā datu skripta (`sync_data.sh`) izpildi regulāri, piemēram, ik pēc 5 minūtēm, veicot attiecīgu ierakstu ar laika intervāla uzstādījumu `crontab`, piemēram, `*/5 * * * * /home/pi/record/sync_data.sh >/dev/nul`, kas jāieraksta kā rinda teksta redaktorā *nano*, kas atveras pēc komandas `'sudo crontab -e'` izpildes. Līdz ar to dati par teritorijā iebraukušās automašīnas numuru būs pieejami *Google Drive* ne vēlāk kā pēc 5 minūtēm, jo teksta informācija CSV formātā datu apjoma ziņā sinhronizācijas vajadzībām ir niecīga, bet paši attēli ar redzamo automašīnu būs pieejami *Google Drive* ne vēlāk kā pēc 24 stundām, ņemot vērā `'crontab'` uzstādīto laiku 1am attēlu datņu sinhronizācijai (sk. 4.2.2.apakšnodaļu).

REZULTĀTI UN SECINĀJUMI

Pasaulē pieejamo vienplates datoru klāsts ir ļoti plašs. Līdz ar to mikrodatora izvēle ir atkarīga no tā izmantošanas mērķa un, piemēram, tādiem kritērijiem kā veiktspēja, izmērs, cena, ievadizvades un audiovizuālās saskarnes, pieslēgvietas, enerģijas patēriņš. *Raspberry Pi* no visiem šajā maģistra darbā apskatītajiem mikrodatoriem izceļas kā lētākā ierīce attiecībā uz veiktspēju un izmēru. Savukārt programmēšanas valoda *Python* ir pietiekami vienkārša un vienlaicīgi arī jaudīga, lai veiktu komplicētus uzdevumus. *Raspberry Pi* piedāvā šo programmēšanas valodu komplektā ar operētājsistēmu *NOOBS* instalācijas pakotnē.

Maģistra darba teorētiskās daļas ietvaros ir apskatīta vienplates datoru definīcija un rašanās vēsture, programmēšanas valodas *Python* vēsture un tās izvēles pamatojums, apskatīti pasaulē pieejamie mikrodatori, izmantojot salīdzināšanas metodi pēc dažādiem kritērijiem (cena, veiktspēja, ievadizvades saskarnes un pieslēgvietas, audiovizuālās saskarnes, izmērs, citi fiziskie un elektriskie parametri).

Maģistra darba praktiskās daļas ietvaros ir izdevies izveidot darba vajadzībām nepieciešamo risinājumu, kas veic slēgtā teritorijā iebraucošu automašīnu reģistrēšanu: fotografēšanu, automašīnas numura atpazīšanu, attēlu un datu sinhronizāciju ar *Google Drive*, padarot iespējamu teritorijas kontroli attālināti, izmantojot mākoņ tehnoloģijas. Tādējādi ir sasniegts galvenais šī risinājuma mērķis – aizvietot teritorijas sargu un nodrošināt iestādes administratīvās daļas vadītāju ar informāciju, vai teritorijā iebrauc automašīna, kurai ir izsniegta caurlaide vai kāda cita. Ja cita, tad risinājums ērti piedāvā noskaidrot iebraukšanas laiku un automašīnas numuru. Papildus maģistra darba ietvaros, izmantojot programmēšanas valodu *Python*, ir izveidots tīmekļa serveris un mājas lapa, kura atjaunojas 5 sekunžu intervālā, parādot aktuālāko kameras uzņemto attēlu. Teritorijā iebraucošo automašīnu fotografēšana arī ir realizēta, izmantojot kustības sensoru, kas darbojas programmēšanas valodā *Python*, analizējot izmaiņas kameras uzņemtajos attēlos, izmantojot *Raspberry Pi Camera* moduli. Automašīnu numuru atpazīšanas risinājuma pamatā ir *Bash* skripts, kas izmanto apjomīgo atvērtā koda automašīnu numuru atpazīšanas bibliotēku *OpenALPR*, kā arī ar to saistītās bibliotēkas (*leptonica*, *opencv*, *tessdata*, *tesseract* u.c.). Ar *Bash* skripta palīdzību tiek veidots arī automašīnu plūsmas reģistrs teksta datu *CSV* formātā, kuru tiek ievadīts automašīnas iebraukšanas laiks formātā ‘*yyyymmdd-hhmmss*’ un no attēla atpazītais automašīnas numurs. Tā kā automašīnas iebraukšanas laiks sakrīt arī ar uzņemtā attēla datnes *JPG* nosaukumu, tad katru *CSV* ierakstu var viegli sasaistīt ar kameras uzņemto attēlu. Maģistra darba ietvaros izstrādāto programmu (skriptu) teksts ir iekļauts ne tikai šī maģistra

darba 2.-7.pielikumā, bet pieejams arī *GitHub*⁹⁰, nodrošinot risinājuma pilnveidošanas iespējas.

Tomēr risinājuma uzstādīšanas un testēšanas rezultātā konstatēti šādi trūkumi:

- 1) kustības sensors ģenerē attēla *JPG* datnes arī tad, ja kustību nav radījusi automašīna, piemēram, teritorijā ir ienācis gājējs. Šādos gadījumos *CSV* datnē pretim laikam ir ievadīts nevis automašīnas numurs, bet tukšums, ko gan var viegli nofiltrēt un izmantot tikai tos ierakstus, kas satur arī automašīnas numuru;
- 2) *Raspberry Pi* kamera ne vienmēr uzņem tik kvalitatīvu attēlu ar redzamu teritorijā iebraucošu automašīnu, lai sekmīgi nostrādātu automašīnas numura atpazīšana, t.i.bieži vien ‘*alpr*’ komandas rezultāts ir teksts „No license plates found.”. Tie pamatā ir gadījumi, kad automašīnas numurs ir netīrs vai teritorijā ir iebraukusi automašīna ar citas valsts numuru, piemēram, Krievijas. No sākuma vidēji šāds rezultāts bija katram septītajam automašīnas attēlam. To izdevās uzlabot, novietojot kameru tuvāk teritorijā iebraukšanas vietai – barjerai – tā, lai kamera atrodas aptuveni 5 metru attālumā no iespējamās automašīnas. Rezultātā izdodas atpazīt praktiski katru automašīnu, ja vien tā nav pārāk netīra vai izmanto ne Eiropas standarta numura zīmi. Tomēr būtu noderējusi arī risinājuma testēšana un precizitātes novērtējums dienā, naktī, sniegā, lietū, pavasara vai rudens dubļu sezonā, kas nav izdevies šī maģistra darba izstrādes laikā. Pastāv augsts risks, ka stiprā sniegā, lietū un rudens dubļu laikā automašīnas numura atpazīšana nenotiks precīzi vai nenotiks vispār.
- 3) ja risinājuma darbība slēgtā teritorijā, kur automašīnu plūsma ir neliela un katra automašīna apstājas pie barjeras brīdī, kad tiek veikta tās fotofiksēšana, ir veiksmīga, tad šī risinājuma tests uz ielas, kur ir intensīvāka satiksme, cieš neveiksmi. Risinājums būtu būtiski jāpilnveido, ja plānots izmantot to intensīvas satiksmes laikā, iespējams, izmantojot jaudīgāku mikrodatoru un labāku kameru, jo *Raspberry Pi* nespēj fiksēt visas automašīnas, kas ātri brauc pa ielu. Daži no attēliem ir miglaini, dažos automašīna ir nofotografēta no lielāka attāluma, nekā nepieciešams tās sekmīgai numura zīmes nolasīšanai un atpazīšanai ar *OpenALPR*. Vārdu sakot, ir jāņem vērā, ka ar šo risinājumu nebija plānots aizstāt fotoradaru.
- 4) šis risinājums neder teritorijas novērošanai tumsā, jo tajā ir izmantota *Raspberry Pi* standarta kamera, kas labi darbojas tikai dienas gaismā. Tomēr to var viegli pilnveidot darbībai naktī, aizvietojot standarta kameru ar *NOIR* kameru, kas ir paredzēta filmēšanai tumsā. Šādā gadījumā gan būs jāveic arī papildus ieguldījumi

⁹⁰ <https://github.com/HaraldsPlass/camera/tree/master>

risinājuma uzlabošanai un infrasarkanās gaismas radīšanai, kas ir labi aprakstīti dažādos interneta avotos;

- 5) ja datiem *CSV* formātā ir ļoti neliels apjoms un tos var bez problēmām sinhronizēt ar *Google Drive* ik pēc 5 minūtēm, tad attēlu datnes *JPG* formātā, kas pie intensīvākās satiksmes ir saģenerētas vismaz 10 minūtē, ir pietiekami apjomīgas, lai aizkavētu ne tikai kameras darbību, bet apturētu pat visas iekārtas *Raspberry Pi* darbību. Tāpēc attēlu sinhronizācija ar *Google Drive* risinājumā ir plānota tikai reizi diennaktī 1am, kad ir tumsa un nedarbojas kameras modulis;
- 6) ja neskaita veltītās programmēšanas stundas (darba izmaksas), tad šajā maģistra darbā realizēto risinājumu var izveidot par 30 EUR vai 47 EUR (mobila risinājuma gadījumā, kuru vismaz 24 stundas darbina ārējs barošanas bloks, ja kameras uzstādīšanas vietā nav pieejams stacionārs enerģijas avots). Risinājums ir testēts un darbojas arī gadījumā, ja tā darbināšanas vietā nav pieejams *WiFi*. To var lieliski aizvietot ‘*Personal hotspot*’ vai ‘*WiFi tīklāja*’ iespēja, kas pieejama pat viedtālrunī, kuru var iegādāties par 60 EUR. Tātad, ja risinājuma darbināšanas vietā nav pieejama ne enerģija, ne *WiFi* tīkls, tad izmaksas sadārdzinas: par 17 – EUR uz ārējās baterijas rēķina, par 60 EUR – ja blakus risinājumam ir jāizveido *WiFi* ar lētāko tirgū pieejamo viedtālruni un, piemēram, priekšapmaksas mobilo sarunu karti (lētākais starta komplekts ar bezlimita internetu uz nedēļu izmaksā 1,50 EUR). Lai samazinātu šīs izmaksas, var atteikties no prasības redzēt *Google Drive* datus ar teritorijā iebraucošām automašīnām (numuru un laiku) ik pēc 5 minūtēm, atliekot šo sinhronizāciju līdz brīdim, piemēram, naktij, kad risinājumu var noņemt, ienest iekšā uzlādēt, kur ir pieejams *WiFi* tīkls un var automātiski notikt datu un attēlu sinhronizācija.

Pārskatot šī maģistra darba ietvaros veikto pasaulē pieejamo mikrodatoru salīdzinājumu pēc cenas, veiktspējas un pieslēgumvietām, iespējams, vēl mazākas risinājuma izmaksas varētu panākt *Raspberry Pi Zero* vietā izmantojot mikrodatoru *Orange Pi Lite*, kuram ir gan kameras pieslēgumvieta *CSI*, gan *WiFi*. Arī šim modelim tirgū ir pieejama lēta Ķīnā ražota kamera. Interesants varētu būt arī risinājums ar *Raspberry Pi Zero* konkurentu *C.H.I.P.*

Jebkurā gadījumā *Raspberry Pi* pierāda, ka šis mikrodators lieliski noder ne tikai mācību nolūkos, kam tas pamatā sākotnēji bija radīts, bet arī kā ierīce dažādiem risinājumiem lietu interneta (*Internet of Things - IoT*) pasaulē.

IZMANTOTĀ LITERATŪRA UN AVOTI

1. „A few thoughts about today’s single-board micro computers,” *Iqjar*. [tiešsaiste]. - [atsauce 23.01.2017.]. Pieejams: <http://iqjar.com/jar/a-few-thoughts-about-micro-computers/>.
2. Alex Bradbury, Russel Winder, *Learning Python with Raspberry Pi*. John Wiley & Sons Ltd., 2014, p. 290.
3. A.Martelli, *Python in a nutshell*. USA: O'Reilly, 2006, p. 695.
4. A. Mappuji, N. Effendy, M. Mustaghfirin, F. Sondok, R. P. Yuniar, S. P. Pangesti, „Study of Raspberry Pi 2 quad-core Cortex-A7 CPU cluster as a mini supercomputer,” in *Proc. 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2016. Pieejams: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7863250>.
5. „An overview and comparison of today’s single-board micro computers,” *Iqjar*. [tiešsaiste]. - [atsauce 23.01.2017.]. Pieejams: <http://iqjar.com/jar/an-overview-and-comparison-of-todays-single-board-micro-computers/>.
6. Craig Steiner, *The 8051/8052 Microcontroller*, Universal Publishers, 2005, p. 332.
7. E. Upton, "New product! Raspberry Pi Zero W joins the family," 28.02.2017. [tiešsaiste]. - [atsauce 07.03.2017.]. <https://www.raspberrypi.org/blog/raspberry-pi-zero-w-joins-family/>.
8. E. Upton, „Raspberry Pi Zero: the \$5 computer,” *The MagPi*, vol. 40, Dec. 2015, pp. 1-101. [tiešsaiste]. - [atsauce 23.01.2017.]. <https://www.raspberrypi.org/magpi-issues/MagPi40.pdf> un <https://www.raspberrypi.org/blog/raspberry-pi-zero/>.
9. G. Bekaroo, A. Santokhee, „Power consumption of the Raspberry Pi: A comparative analysis,” in *Proc. International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech)*, 2016. Pieejams: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7737367>.
10. J. Brodtkin, „New \$10 Raspberry Pi Zero comes with Wi-Fi and Bluetooth,” Feb. 2017. [tiešsaiste]. - [atsauce 07.03.2017.]. <https://arstechnica.com/information-technology/2017/02/new-10-raspberry-pi-zero-comes-with-wi-fi-and-bluetooth/>.
11. K. Hentschel, D. Jacob, J. Singer, M. Chalmers, „Supersensors: Raspberry Pi Devices for Smart Campus Infrastructure,” in *Proc. 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2016. Pieejams: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7575844>.
12. Lielā terminu vārdnīca. [tiešsaiste]. - [atsauce 22.02.2017.]. Pieejams: <http://www.termini.lv>.

13. Low Cost Pervasive Computing: Market Opportunities in Embedded Computing and the Internet of Things, Mind Commerce, 2012, eBook. [tiešsaiste]. - [atsauce 16.11.2016.]. Pieejams: <http://web.b.ebscohost.com>.
14. N. Owano, „Raspberry Pi gets customized OS called Raspbian,” [tiešsaiste]. - [atsauce 23.01.2017.]. <https://phys.org/news/2012-07-raspberry-pi-customized-os-raspbian.html>.
15. P. A. Shinde, Y. B. Mane, „Advanced vehicle monitoring and tracking system based on Raspberry Pi,” in *Proc. 9th International Conference on Intelligent Systems and Control (ISCO)*, 2015. Pieejams: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7282250>.
16. „Raspberry Pi sales pass another major milestone. 11 million sold,” *The MagPi*, Jan. 2017, pp. 10. [tiešsaiste]. - [atsauce 27.02.2017.]. Pieejams: <https://www.raspberrypi.org/magpi-issues/MagPi53.pdf>.
17. „Single-board computer,” *Wikipedia*, [tiešsaiste]. - [atsauce 23.01.2017.]. Pieejams: http://en.wikipedia.org/wiki/Single-board_computer.
18. T. Peters, *The Zen of Python*. [tiešsaiste]. - [atsauce 23.01.2017.]. Pieejams: <http://legacy.python.org/dev/peps/pep-0020/>.
19. Q. He, B. Segee, V. Weaver, „Raspberry Pi 2 B+ GPU Power, Performance, and Energy Implications,” in *Proc. International Conference on Computational Science and Computational Intelligence (CSCI)*, 2016. Pieejams: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7881331>.
20. W. J. Chun, „Python's "New" Division: Python 2 Versus Python 3”. [tiešsaiste]. - [atsauce 23.01.2017.]. Pieejams: <http://www.informit.com/articles/article.aspx?p=1439189>.
21. V. Patchava, H.B. Kandala, P.R. Babu, „A Smart Home Automation technique with Raspberry Pi using IoT,” in *Proc. International Conference on Smart Sensors and Systems (IC-SSS)*, 2015. Pieejams: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7873584>.

PIELIKUMI

1.pielikums. Vienplates datoru salīdzinājums pēc izlaišanas gada, cenas un sistēmas mikroshēmā

Nosaukums	Modelis	Izlaišanas gads/ mēnesis	Cenu diapazona kategorija	Sistēma mikroshēmā
Arduino Uno ⁹¹	R3	2010/09	1 ⁹²	Atmel ATmega328P
armStone ⁹³	A5	2013/02	5 ⁹⁴	Freescale Vybrid VF6xx
armStone ⁹⁵	A9	2012	5 ⁹⁶	Freescale i.MX6
Arndale Board ⁹⁷	5250-AA	2012/10	5 ⁹⁸	Samsung Exynos 5 Dual (5250)
Arndale Octa Board ⁹⁹	Exynos 5420	2013/10	5 ¹⁰⁰	Samsung Exynos 5 Octa (5420)
Banana Pi ¹⁰¹	M1	2014/03	1 ¹⁰²	Allwinner A20
Banana Pro ¹⁰³		2014/12	3 ¹⁰⁴	Allwinner A20
Banana Pi ¹⁰⁵	M2	2015	2 ¹⁰⁶	?
Banana Pi ¹⁰⁷	M3	2015	3 ¹⁰⁸	?
BeagleBoard ¹⁰⁹	D	2012/10	3 ¹¹⁰	TI OMAP 3530
BeagleBoard-xM ¹¹¹	C2	2010/09	4 ¹¹²	TI DaVinci DM3730
BeagleBone ¹¹³	A6A	2011/10	3 ¹¹⁴	TI Sitara AM335x
BeagleBone Black ¹¹⁵	C	2013/04	2 ¹¹⁶	TI Sitara AM335x
Boardcon EM210 ¹¹⁷	A8	2012/07	3 ¹¹⁸	Samsung K4T1G164QF, Boardcon CM210-III

⁹¹ <https://www.arduino.cc/en/Main/arduinoBoardUno>

⁹² <https://store.arduino.cc/product/A000066>

⁹³ <https://www.fs-net.de/en/products/armstone/armstonea5/>

⁹⁴ <https://www.fs-net.de/en/products/armstone/armstonea5/>

⁹⁵ <https://www.fs-net.de/en/products/armstone/armstonea9/>

⁹⁶ <https://www.fs-net.de/en/products/armstone/armstonea9/>

⁹⁷ http://www.arndaleboard.org/wiki/index.php/Main_Page

⁹⁸ <http://arndaleboard.com/us/?menuType=product&mode=list&lcate=001&mcate=001>

⁹⁹ http://www.arndaleboard.org/wiki/index.php/Main_Page

¹⁰⁰ <https://liliputing.com/2013/11/179-arndale-octa-dev-board-features-samsung-5420-processor.html>

¹⁰¹ http://www.bananapi.com/product/detail_10.html

¹⁰² https://ru.aliexpress.com/store/product/In-stock-BPI-M1-Banana-Pi-M1-A20-Dual-Core-1GB-RAM-on-board-wifi-singel/302756_32492142619.html?spm=2114.12010612.0.0.Yju2zj

¹⁰³ <https://www.amazon.com/SainSmart-Pi-Banana-Mainboard-Cortex-A7-Dual-core/dp/B00RD86QPA>

¹⁰⁴ <https://www.amazon.com/SainSmart-Pi-Banana-Mainboard-Cortex-A7-Dual-core/dp/B00RD86QPA>

¹⁰⁵ https://ru.aliexpress.com/store/product/Freeshipment-for-Banana-PI-M2-BPI-M2-quad-core-single-board-computer-linux-board/302756_32252431797.html?detailNewVersion=&categoryId=200003315

¹⁰⁶ <http://www.banana-pi.org/m2u.html>

¹⁰⁷ <http://www.banana-pi.org/m3.html>

¹⁰⁸ <https://www.amazon.com/Banana-Pi-Computer-Octa-Core-Ethernet/dp/B0181N6FZE>

¹⁰⁹ <https://www.element14.com/community/docs/DOC-83315/l/beaglebone-black-wireless-coming-soon>

¹¹⁰ <http://beagleboard.org/>

¹¹¹ <http://beagleboard.org/beagleboard-xm>

¹¹² <http://beagleboard.org/beagleboard-xm>

¹¹³ http://beagleboard.org/static/beaglebone/latest/Docs/Hardware/BONE_SRM.pdf

¹¹⁴ <https://www.adafruit.com/products/513>

¹¹⁵ <https://www.amazon.com/Beagleboard-BeagleBone-Black-Beaglebone-Devkit/dp/B00CHYOLHK>

¹¹⁶ <https://www.element14.com/community/docs/DOC-84108?ICID=BBB-feature-link>

¹¹⁷ <http://www.armdesigner.com/EM210/>

¹¹⁸ 2016.gada 21.novembra e-pasts, <https://github.com/HaraldsPlass/camera/tree/prices>

Boardcon EM3288 ¹¹⁹	A17	2016/04	4 ¹²⁰	Rockchip RK3288
C.H.I.P. ¹²¹		2015/05	1 ¹²²	Allwinner A10
Cosmic+ Board ¹²³		2013/07	3 ¹²⁴	?
Cubieboard ¹²⁵	1	2012/09	3 ¹²⁶	AllWinner A10
Cubieboard ¹²⁷	2	2013/06	3 ¹²⁸	Allwinner A20
Cubieboard ¹²⁹	3	2013/10	3 ¹³⁰	Allwinner A20
Cubieboard ¹³¹	4 CC-A80 Octo	2014	4 ¹³²	Allwinner A80
Cubieboard ¹³³	5	2016/03	4 ¹³⁴	Allwinner H8
Cubox-i ¹³⁵	i2	2013/11	3 ¹³⁶	i.MX6 Solo i.MX6 Dual Lite
Cubox-i ¹³⁷	i2eX	2013/12	4 ¹³⁸	i.MX6 Dual i.MX6 Quad
Cubox-i ¹³⁹	i4Pro	2013/13	5 ¹⁴⁰	i.MX6 Quad
Dragonboard ¹⁴¹	Snapdragon 410 (APQ8016)	2015/09	3 ¹⁴²	Qualcomm® Snapdragon 400
DreamPlug ¹⁴³	MW302	2011/02	2 ¹⁴⁴	Cortex M4
WinSystems ¹⁴⁵	EPX-C380- D	2015/10	4 ¹⁴⁶	?
WinSystems ¹⁴⁷	E3845 SBC	?	5 ¹⁴⁸	?
Firefly ¹⁴⁹	RK3288	2014	4 ¹⁵⁰	Rockchip RK3288 (28nm HKMG

¹¹⁹ http://www.armdesigner.com/EM3288_SBC/

¹²⁰ 2016.gada 21.novembra e-pasts, <https://github.com/HaraldsPlass/camera/tree/prices>

¹²¹ <http://makezine.com/2015/11/28/chip-vs-pi-zero/>

¹²² <https://getchip.com/pages/chip>

¹²³ <http://phytec.com/products/single-board-computers/cosmic-vybrid/>

¹²⁴ <http://phytec.com/products/single-board-computers/cosmic-vybrid/>

¹²⁵ <http://cubieboard.org/>

¹²⁶ <https://www.newit.co.uk/shop/Cubieboard>

¹²⁷ <http://cubieboard.org/>

¹²⁸ <https://www.newit.co.uk/shop/Cubieboard>

¹²⁹ <http://cubieboard.org/>

¹³⁰ <https://www.newit.co.uk/shop/Cubieboard>

¹³¹ <http://cubieboard.org/>

¹³² <https://www.newit.co.uk/shop/Cubieboard>

¹³³ <http://cubieboard.org/>

¹³⁴ <https://www.newit.co.uk/shop/Cubieboard>

¹³⁵ <https://www.solid-run.com/freescale-imx6-family/cubox-i/cubox-i-specifications/>

¹³⁶ <https://www.solid-run.com/freescale-imx6-family/cubox-i/cubox-i-specifications/>

¹³⁷ <https://www.solid-run.com/freescale-imx6-family/cubox-i/cubox-i-specifications/>

¹³⁸ <https://www.solid-run.com/freescale-imx6-family/cubox-i/cubox-i-specifications/>

¹³⁹ <https://www.solid-run.com/freescale-imx6-family/cubox-i/cubox-i-specifications/>

¹⁴⁰ <https://www.solid-run.com/freescale-imx6-family/cubox-i/cubox-i-specifications/>

¹⁴¹ <http://www.96boards.org/product/dragonboard410c/>

¹⁴² https://www.arrow.com/en/products/dragonboard410c/arrow-development-tools?utm_source=product-page&utm_medium=96boards&utm_term=organic&utm_content=dragonboard40c&utm_campaign=dragonboard410c-arrow

¹⁴³ <http://www.globalscaletechnologies.com/c-5-dreamplugs.aspx>

¹⁴⁴ <http://www.globalscaletechnologies.com/c-5-dreamplugs.aspx>

¹⁴⁵ <http://www.winsystems.com/product/epx-c380-d2-1/>

¹⁴⁶ <http://www.embeddedstar.com/weblog/tag/winsystems/page/2/>

¹⁴⁷ <http://www.embeddedstar.com/weblog/tag/winsystems/page/2/>

¹⁴⁸ http://estore.bcmcom.com/product_info.php?products_id=114

¹⁴⁹ http://en.t-firefly.com/en/firenow/firefly_rk3288/specifications/

¹⁵⁰ <https://ru.aliexpress.com/store/1710008>

				Process)
Firefly ¹⁵¹	RK3288 Plus Reload	?	5 ¹⁵²	Rockchip RK3288 (28nm HKMG Process)
Foxconn ¹⁵³	AT-5570	2012/11	4 ¹⁵⁴	AMD A45 FCH
Forlinx ¹⁵⁵	OK335xD	2013/03	5 ¹⁵⁶	TI Sitara AM335x Cortex-A8
Gizmo Board ¹⁵⁷	Gizmo 2	2013/02	5 ¹⁵⁸	AMD GX210HA
GoWarrior ¹⁵⁹	TIGER Board	2015/09	3 ¹⁶⁰	ALi M3733-AFAAA
Gumstix ¹⁶¹	Overo EarthSTOR M + Summit	2012/01	4 ¹⁶²	Texas Instruments AM3703
Hackberry ¹⁶³	A10	2012/09	3 ¹⁶⁴	Allwinner A10
HiKey ¹⁶⁵	Rev A1	2015/02	4 ¹⁶⁶	HiSilicon Kirin 620
HummingBoard ¹⁶⁷	i1	2014/07	3 ¹⁶⁸	Freescale i.MX6
HummingBoard ¹⁶⁹	i2	2014/08	3 ¹⁷⁰	Freescale i.MX6
HummingBoard ¹⁷¹	i2eX	2014/09	3 ¹⁷²	Freescale i.MX6
Inforce ¹⁷³	6540 Snapdragon 805 (APQ8084)	2014/10	5 ¹⁷⁴	Qualcomm® Snapdragon™ 805 (APQ8084 SoC)
Inforce ¹⁷⁵	6410plus	2015/06	4 ¹⁷⁶	Qualcomm Snapdragon 600 (APQ8064)

¹⁵¹ http://en.t-firefly.com/en/firenow/firefly_rk3288/specifications/

¹⁵² <https://ru.aliexpress.com/store/1710008>

¹⁵³ <http://www.foxconnchannel.com/ProductDetail.aspx?T=NanoPC&U=en-us0000019>

¹⁵⁴ <http://www.moblive.lv/foxconn-at-5570-barebone-amd-brazos-c70-ddr3-hdmi-schwarz.64863-388.pg?!=en&p=1>

¹⁵⁵ http://www.forlinx.net/products_detail/productId=40.html

¹⁵⁶ <http://www.cnx-software.com/2014/10/23/forlinx-ok335xd-industrial-single-board-computer-supports-android-linux-and-wince/>

¹⁵⁷ <http://www.gizmosphere.org/products/gizmo-2/>

¹⁵⁸ <https://liliputing.com/2014/11/gizmo-2-199-single-board-pc-amd-embedded-cpu.html>

¹⁵⁹ <http://www.cnx-software.com/2015/12/10/gowarrior-tiger-development-board-with-ali-m3733-soc-to-support-android-debian-and-freertos/>

¹⁶⁰ <http://www.cnx-software.com/2015/12/10/gowarrior-tiger-development-board-with-ali-m3733-soc-to-support-android-debian-and-freertos/>

¹⁶¹ <https://store.gumstix.com/index.php/overo-earthstorm-com.html>

¹⁶² <https://store.gumstix.com/index.php/overo-earthstorm-com.html>

¹⁶³ <https://liliputing.com/2012/09/hackberry-a10-now-shipping-for-65-android-linux-dev-board.html>

¹⁶⁴ <https://liliputing.com/2012/09/hackberry-a10-now-shipping-for-65-android-linux-dev-board.html>

¹⁶⁵ <http://www.96boards.org/product/hikey/>

¹⁶⁶ <http://www.lenovator.com/product/90.html>

¹⁶⁷ <https://www.solid-run.com/freescale-imx6-family/hummingboard/hummingboard-specifications/>

¹⁶⁸ <https://www.solid-run.com/freescale-imx6-family/hummingboard/hummingboard-specifications/>

¹⁶⁹ <https://www.solid-run.com/freescale-imx6-family/hummingboard/hummingboard-specifications/>

¹⁷⁰ <https://www.solid-run.com/freescale-imx6-family/hummingboard/hummingboard-specifications/>

¹⁷¹ <https://www.solid-run.com/freescale-imx6-family/hummingboard/hummingboard-specifications/>

¹⁷² <https://www.solid-run.com/freescale-imx6-family/hummingboard/hummingboard-specifications/>

¹⁷³ <http://www.inforcecomputing.com/products/single-board-computers-sbc/qualcomm-snapdragon-600-inforce-6410plus-sbc?scroll=learnmore>

¹⁷⁴ <http://www.inforcecomputing.com/products/>

¹⁷⁵ <http://www.inforcecomputing.com/products/single-board-computers-sbc/qualcomm-snapdragon-805-inforce-6540-sbc?scroll=learnmore>

¹⁷⁶ <http://www.inforcecomputing.com/products/>

Intel Galileo ¹⁷⁷	Gen 2	2013/10	2 ¹⁷⁸	Intel Quark SoC X1000
Inventami ¹⁷⁹	Entry	2015/12	5 ¹⁸⁰	Freescale i.MX6
Inventami ¹⁸¹	Full	2015/12	5 ¹⁸²	Freescale i.MX6
MarsBoard ¹⁸³	A10 New	2013/03	2 ¹⁸⁴	Allwinner A10
MarsBoard ¹⁸⁵	A20 New	2013/04	3 ¹⁸⁶	Allwinner A20
MarsBoard ¹⁸⁷	RK3066	2014/05	4 ¹⁸⁸	Rockchip RK3066
MinnowBoard ¹⁸⁹	Turbot	2013/04	4 ¹⁹⁰	64-bit Intel® Atom™ E38xx Series System on a Chip
Imagination Technologies ¹⁹¹	VL-62851 Creator Ci20	2014/08	3 ¹⁹²	Ingenic JZ4780
MiraBox ¹⁹³		2014/06	4 ¹⁹⁴	Armada 370
Rikomagic ¹⁹⁵	MK802 IV	2013/05	3 ¹⁹⁶	Rockchip 3188
Rikomagic ¹⁹⁷	MK902	2014/09	3 ¹⁹⁸	Rockchip RK3188
MYIR ¹⁹⁹	MYD-AM335X	2013/12	4 ²⁰⁰	Texas Instruments (TI) Sitara AM335x
MYIR ²⁰¹	Rico Board	2015/3	3 ²⁰²	TI's AM437x Sitara
MYIR ²⁰³	Z-turn Board	2015/3	3 ²⁰⁴	Xilinx Zynq-7000 family

¹⁷⁷ <https://software.intel.com/en-us/iot/hardware/galileo>

¹⁷⁸ http://www.intel.com/buy/us/en/product/emergingtechnologies/intel-galileo-gen-2-board-462661?icid=Intel_maker

¹⁷⁹ <http://www.inventami.com/>

¹⁸⁰ <http://www.inventami.com/>

¹⁸¹ <http://www.inventami.com/>

¹⁸² <http://www.inventami.com/>

¹⁸³ http://www.marsboard.com/new_marsboard_a10_feature.html

¹⁸⁴ <http://www.cnx-software.com/2013/03/06/49-90-marsboard-is-yet-another-allwinner-a10-development-board/>

¹⁸⁵ http://www.marsboard.com/new_marsboard_a20_feature.html

¹⁸⁶ http://www.hotmcu.com/new-marsboard-a20-dev-board-p-67.html?cPath=34_33

¹⁸⁷ <http://www.waveshare.com/product/MarsBoard-RK3066-Pro.htm>

¹⁸⁸ <http://www.waveshare.com/product/MarsBoard-RK3066-Pro.htm>

¹⁸⁹ <https://minnowboard.org/learn-more/high-level-technical-specifications>

¹⁹⁰ <http://store.netgate.com/Turbot.aspx>

¹⁹¹ <http://eu.mouser.com/new/imagination-technologies/imagination-technologies-ci20/>

¹⁹² <http://eu.mouser.com/ProductDetail/Imagination-Technologies/VL-62851/?qs=tCmD4XIZ%2fDlxiQwKR9F%252bA%3d%3d>

¹⁹³ <https://www.globalscaletechnologies.com/p-58-mirabox-development-kit.aspx>

¹⁹⁴ <https://www.globalscaletechnologies.com/p-58-mirabox-development-kit.aspx>

¹⁹⁵ <http://www.rikomagic.co.uk/>

¹⁹⁶ <https://www.cloudsto.com/store/mini-droid/rkm-mk802iv/rikomagic-quad-core-mk802iv-8gb-flash-bluetooth-dhl-express-shipping-detail.html>

¹⁹⁷ <http://www.rikomagic.co.uk/>

¹⁹⁸ <https://www.cloudsto.com/store/mini-droid/rkm-mk902/rikomagic-mk902-quad-core-8gb-flash-dhl-express-shipping-detail.html>

¹⁹⁹ <http://www.myirtech.com/list.asp?id=466>

²⁰⁰ <http://www.myirtech.com/list.asp?id=466>

²⁰¹ <http://www.myirtech.com/list.asp?id=510>

²⁰² <http://www.myirtech.com/list.asp?id=510>

²⁰³ <http://www.myirtech.com/list.asp?id=502>

²⁰⁴ <http://www.myirtech.com/list.asp?id=502>

MYIR ²⁰⁵	MYC-IMX28X	2014/5	2 ²⁰⁶	Freescale i.MX28
NanoPC ²⁰⁷	T1	2014/01	3 ²⁰⁸	Samsung Exynos 4412
NanoPi ²⁰⁹	2	2015/11	1 ²¹⁰	Samsung S5P 4418
NanoPi ²¹¹	NEO (256 MB, 512 MB)	2016/07	1 ²¹²	Allwinner H3
Nitrogen6x ²¹³	Rev 3	2013/06	5 ²¹⁴	Freescale i.MX6 Quad
Nvidia ²¹⁵	Jetson TK1	2014/04	5 ²¹⁶	Nvidia Tegra K1
ODROID ²¹⁷	C1+	2015/07	1 ²¹⁸	Amlogic S805
ODROID ²¹⁹	C2	2016/03	2 ²²⁰	Amlogic S905
ODROID ²²¹	W	2014/08	1 ²²²	Broadcom BCM2835
ODROID ²²³	XU4	2015/07	3 ²²⁴	Samsung Exynos 5 Octa (5422)
OLinuXino ²²⁵	A10 LIME	?	1 ²²⁶	Allwinner A10
OLinuXino ²²⁷	A13 base, MICRO, WIFI	?	1 ²²⁸	Allwinner A13
OLinuXino ²²⁹	A20 LIME, MICRO	2013/06	3 ²³⁰	Allwinner A20
OLinuXino ²³¹	A20 LIME2	2014/09	3 ²³²	Allwinner A20
Orange Pi ²³³		2015/01	1 ²³⁴	AllWinner H3

²⁰⁵ <http://www.myirtech.com/list.asp?id=475>

²⁰⁶ <http://www.myirtech.com/list.asp?id=475>

²⁰⁷ <http://www.cnx-software.com/2014/01/20/69-nanopc-by-friendlyarm-features-samsung-exynos-4412-soc/>

²⁰⁸ <http://www.cnx-software.com/2014/01/20/69-nanopc-by-friendlyarm-features-samsung-exynos-4412-soc/>

²⁰⁹ http://nanopi.org/NanoPi-2_Feature.html

²¹⁰ <http://nanopi.io/nanopi2.html>

²¹¹ <http://nanopi.io/nanopi-neo.html>

²¹² http://www.friendlyarm.com/index.php?route=product/product&product_id=132

²¹³ <http://eu.mouser.com/new/Boundary-Devices/boundary-devices-nitrogen6x/>

²¹⁴ http://eu.mouser.com/ProductDetail/Boundary-Devices/Nit6Q_W/?qs=IGUynppghXy6PkBfEcaxHQ%3d%3d

²¹⁵ <https://developer.nvidia.com/embedded/develop/hardware>

²¹⁶ <https://www.amazon.com/NVIDIA-Jetson-TK1-Development-Kit/dp/B00L7AWOEC>

²¹⁷ http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143703355573

²¹⁸ http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143703355573

²¹⁹ http://odroid.com/dokuwiki/doku.php?id=en:c2_hardware

²²⁰ http://www.hardkernel.com/main/products/prdt_info.php?g_code=G145457216438

²²¹ http://www.hardkernel.com/main/products/prdt_info.php?g_code=G140610189490&tab_idx=1

²²² http://www.hardkernel.com/main/products/prdt_info.php?g_code=G140610189490

²²³ http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143452239825

²²⁴ http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143452239825

²²⁵ <https://www.olimex.com/Products/OLinuXino/A10/A10-OLinuXino-LIME/open-source-hardware>

²²⁶ <https://www.olimex.com/Products/OLinuXino/A10/A10-OLinuXino-LIME/open-source-hardware>

²²⁷ <https://www.olimex.com/Products/OLinuXino/A13/open-source-hardware>

²²⁸ <https://www.olimex.com/Products/OLinuXino/A13/open-source-hardware>

²²⁹ <https://www.olimex.com/Products/OLinuXino/A20/open-source-hardware>

²³⁰ <https://www.olimex.com/Products/OLinuXino/A20/open-source-hardware>

²³¹ <https://www.olimex.com/Products/OLinuXino/A20/open-source-hardware>

²³² <https://www.olimex.com/Products/OLinuXino/A20/open-source-hardware>

²³³ <https://www.amazon.com/Orange-Pi-Single-Computer-Cortex-A7/dp/B018W6OTIM>

²³⁴ <https://www.amazon.com/Orange-Pi-Single-Computer-Cortex-A7/dp/B018W6OTIM>

Orange Pi ²³⁵	Plus	2015/02	2 ²³⁶	AllWinner H3
Orange Pi ²³⁷	Plus 2	2015/02	2 ²³⁸	AllWinner H3
Orange Pi ²³⁹	Zero	2016/11	1 ²⁴⁰	AllWinner H2
Orange Pi ²⁴¹	PC	2015/02	1 ²⁴²	AllWinner H3
Orange Pi ²⁴³	One	2016/01	1 ²⁴⁴	AllWinner H3
Orange Pi ²⁴⁵	Lite	2016/01	1 ²⁴⁶	AllWinner H3
Orange Pi ²⁴⁷	PC Plus	2016/04	1 ²⁴⁸	AllWinner H3
Orange Pi ²⁴⁹	Plus 2E	2016/04	1 ²⁵⁰	AllWinner H3
PandaBoard ES ²⁵¹	Rev. 3	2011/12	5 ²⁵²	TI OMAP 4460
Adapteva Parallella ²⁵³		2013	4 ²⁵⁴	Zynq Z7010
pcDuino ²⁵⁵	Lite	2013/10	1 ²⁵⁶	Allwinner A10
pcDuino ²⁵⁷	v2	2013/09	3 ²⁵⁸	Allwinner A10
pcDuino ²⁵⁹	3	2014/02	3 ²⁶⁰	Allwinner A20
pcDuino ²⁶¹	3 Nano	2014/07	2 ²⁶²	Allwinner A20
PC Engines APU ²⁶³	APU.1D, AP U.1D4	2014/04	4 ²⁶⁴	AMD Embedded G-Series T40E APU

²³⁵ <http://www.orangepi.org/>

²³⁶ https://ru.aliexpress.com/store/product/NEW-orange-pi-plus-Allwinner-A31s-Dual-Core-1GB-RAM-Open-source-development-board/1553371_32248189300.html

²³⁷ <http://www.orangepi.org/orangepiplus2/>

²³⁸ <https://ru.aliexpress.com/store/1553371>

²³⁹ <http://www.orangepi.org/orangepizero/>

²⁴⁰ <https://ru.aliexpress.com/store/1553371>

²⁴¹ <http://www.orangepi.org/orangepipc/>

²⁴² <https://ru.aliexpress.com/store/1553371>

²⁴³ <http://www.orangepi.org/orangepione/>

²⁴⁴ <https://ru.aliexpress.com/store/1553371>

²⁴⁵ <https://ru.aliexpress.com/store/1553371>

²⁴⁶ <https://ru.aliexpress.com/store/1553371>

²⁴⁷ <http://www.cnx-software.com/2016/04/26/orange-pi-plus-2e-development-board-with-gigabit-ethernet-2gb-ram-16gb-emmc-flash-to-sell-for-35/>

²⁴⁸ <http://www.cnx-software.com/2016/04/26/orange-pi-plus-2e-development-board-with-gigabit-ethernet-2gb-ram-16gb-emmc-flash-to-sell-for-35/>

²⁴⁹ <http://www.cnx-software.com/2016/04/26/orange-pi-plus-2e-development-board-with-gigabit-ethernet-2gb-ram-16gb-emmc-flash-to-sell-for-35/>

²⁵⁰ <http://www.cnx-software.com/2016/04/26/orange-pi-plus-2e-development-board-with-gigabit-ethernet-2gb-ram-16gb-emmc-flash-to-sell-for-35/>

²⁵¹ <http://pandaboard.org/content/pandaboard-es>

²⁵² https://www.aliexpress.com/price/pandaboard_price.html

²⁵³ https://en.wikipedia.org/wiki/Adapteva#Parallella_project

²⁵⁴ <https://www.amazon.com/Adapteva-P1601-DK03-Parallella-16-Desktop-Computer/dp/B0091UD6TM>

²⁵⁵ <http://www.linksprite.com/linksprite-pcduino-lite/>

²⁵⁶ <https://www.amazon.com/pcDuino-Lite-WiFi/dp/B00HDFRMPQ>

²⁵⁷ <http://www.linksprite.com/linksprite-pcduino/>

²⁵⁸ <https://www.amazon.com/pcDuino-V2-Arduino-headers-android/dp/B00HCZH6WG>

²⁵⁹ <https://www.amazon.com/pcDuino-Lite-WiFi/dp/B00HDFRMPQ>

²⁶⁰ <https://www.amazon.com/PcDuino-Dual-Core-Allwinner-Compatible-Arduino/dp/B00NXRYPC6>

²⁶¹ <http://www.linksprite.com/linksprite-pcduino/>

²⁶² <https://www.amazon.com/pcDuino-pcDuino3-Nano/dp/B00OQEQMDU>

²⁶³ <http://www.pcenines.ch/apu.htm>

²⁶⁴ <https://pcenines.ch/order1.php?c=4>

PC Engines APU ²⁶⁵	APU.2C2, A PU.2C4	2016/03	3 ²⁶⁶	AMD Embedded G series GX-412TC
phyBOARD ²⁶⁷	Wega 5V, 12-24V	2013	3 ²⁶⁸	phyCORE-AM3354
phyBOARD ²⁶⁹	Mira Solo, Quad	2014	4 ²⁷⁰	phyCORE-i.MX 6, i.MX 6Solo
PINE ²⁷¹	A64	2015	1 ²⁷²	ARM Cortex A53 64-Bit
PINE ²⁷³	A64+ 2GB	2016	1 ²⁷⁴	ARM Cortex A53 Quad-Core
Radxa Rock ²⁷⁵	base, Lite	2014/01	3 ²⁷⁶	Rockchip RK3188
Raspberry Pi ²⁷⁷	A	2012/02	1 ²⁷⁸	Broadcom BCM2835
Raspberry Pi ²⁷⁹	B	2012/10	1 ²⁸⁰	Broadcom BCM2835
Raspberry Pi ²⁸¹	A+	2014/11	1 ²⁸²	Broadcom BCM2835
Raspberry Pi ²⁸³	B+	2014/07	1 ²⁸⁴	Broadcom BCM2835
Raspberry Pi ²⁸⁵	Zero	2015/11	1 ²⁸⁶	Broadcom BCM2834
Raspberry Pi ²⁸⁷	Zero Wireless	2017/02	1 ²⁸⁸	Broadcom BCM2835
Raspberry Pi ²⁸⁹	2 Model B	2015/02	1 ²⁹⁰	Broadcom BCM2836
Raspberry Pi ²⁹¹	3 Model B	2016/02	1 ²⁹²	Broadcom BCM2837
Rikomagic ²⁹³	MK802 base, + / II	2012/08	1 ²⁹⁴	Allwinner A10
RIoTboard ²⁹⁵	?	2014/01	1 ²⁹⁶	Freescale i.MX6 Solo

²⁶⁵ <http://www.pceingines.ch/apu2.htm>

²⁶⁶ <https://pceingines.ch/order1.php?c=4>

²⁶⁷ <http://www.phytec.de/produkt/single-board-computer/phyboard-wega/>

²⁶⁸ <http://www.phytec.de/produkt/single-board-computer/phyboard-wega/>

²⁶⁹ <http://www.phytec.de/produkt/single-board-computer/phyboard-mira/>

²⁷⁰ <http://www.phytec.de/produkt/single-board-computer/phyboard-mira/>

²⁷¹ <https://www.pine64.org/?product=pine-a64-board>

²⁷² <https://www.pine64.org/>

²⁷³ <https://www.pine64.org/?product=pine-a64-board-2gb>

²⁷⁴ <https://www.pine64.org/>

²⁷⁵ http://www.megaleecher.net/Radxa_Rock_Reviewed#axzz4RrY3V8WO

²⁷⁶ http://www.megaleecher.net/Radxa_Rock_Reviewed#axzz4RrY3V8WO

²⁷⁷ <https://www.raspberrypi.org/products/model-a/>

²⁷⁸ <https://www.element14.com/community/docs/DOC-70725/l/raspberry-pi-model-a-256mb>

²⁷⁹ <https://www.raspberrypi.org/products/model-b/>

²⁸⁰ <https://www.raspberrypi.org/blog/price-cut-raspberry-pi-model-b-now-only-25/>

²⁸¹ <https://www.raspberrypi.org/products/model-a-plus/>

²⁸² <https://www.raspberrypi.org/blog/raspberry-pi-model-a-plus-on-sale/>

²⁸³ <https://www.raspberrypi.org/products/model-b-plus/>

²⁸⁴ <https://www.raspberrypi.org/blog/price-cut-raspberry-pi-model-b-now-only-25/>

²⁸⁵ <https://www.raspberrypi.org/blog/raspberry-pi-zero/>

²⁸⁶ <https://shop.pimoroni.com/products/raspberry-pi-zero>

²⁸⁷ <https://www.raspberrypi.org/products/pi-zero-wireless/>

²⁸⁸ <https://shop.pimoroni.com/products/raspberry-pi-zero-w>

²⁸⁹ <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

²⁹⁰ <https://www.raspberrypi.org/blog/raspberry-pi-2-on-sale/>

²⁹¹ <https://www.raspberrypi.org/documentation/hardware/raspberrypi/>

²⁹² <https://shop.pimoroni.com/products/raspberry-pi-3>

²⁹³ <http://www.rikomagic.co.uk/forum/viewtopic.php?f=2&t=305>

²⁹⁴ <https://www.amazon.com/Rikomagic-MK802-Android-Rockchip-RK3066/dp/B00A0I4YJK>

²⁹⁵ <http://riotboard.org/>

²⁹⁶ <https://www.amazon.com/RIOTBOARD-MCIMX6-SOLO-DEVELOPMENT-PLATFORM/dp/B00IIXTZ98>

RouterBOARD ²⁹⁷	RB450G	2012/12	3 ²⁹⁸	Qualcomm Atheros AR7161
RouterBOARD ²⁹⁹	RB953GS-5HnT	2014/06	5 ³⁰⁰	Qualcomm Atheros QCA9558
Snowball ³⁰¹	SKY-S9500	2011/08	5 ³⁰²	ST-Ericsson NovaThor A9500
Supermicro ³⁰³	E100-8Q	2014/11	5 ³⁰⁴	Intel® Quark™ SoC X1021
TBS ³⁰⁵	2910 Matrix	2014/01	4 ³⁰⁶	Freescale i.MX6 Quad
Tronsmart Draco ³⁰⁷	Octo Meta, Octo Telos	2014/11	5 ³⁰⁸	Allwinner A80
UDOO ³⁰⁹	Dual Basic, Dual, Quad	2013/04	4 ³¹⁰	Freescale i.MX6 + Atmel SAM3X8E
UDOO X86 ³¹¹	Basic, Advanced, Ultra	2016/11	3 ³¹²	Intel® Braswell X5-E8000, N3160
Utilite ³¹³	Value, Standard, Pro	2013/07	4 ³¹⁴	Freescale i.MX6
Utilite2 ³¹⁵	4GB, SSD	2015/02	5 ³¹⁶	Qualcomm Snapdragon 600 (APQ8064M)
VIA ³¹⁷	APC 8750 / Rock	2012/05	3 ³¹⁸	WonderMedia WM8750
VIA ³¹⁹	Springboard VAB-600	2013/11	3 ³²⁰	WonderMedia WM8950
Wandboard ³²¹	Solo, Dual, Quad	2013	3 ³²²	Freescale i.MX6
Graperein ³²³	G4418 SBC	2015/06	2 ³²⁴	Samsung S5P4418
Graperein ³²⁵	G6818 SBC	2015/06	2 ³²⁶	Samsung S5P6818

²⁹⁷ <https://routerboard.com/rb450g>

²⁹⁸ <https://routerboard.com/rb450g>

²⁹⁹ <https://routerboard.com/RB953GS-5HnT-RP>

³⁰⁰ <https://routerboard.com/RB953GS-5HnT-RP>

³⁰¹ <https://web.archive.org/web/20130607204534/http://igloocommunity.org/>

³⁰² <https://web.archive.org/web/20130607204534/http://igloocommunity.org/>

³⁰³ <https://www.supermicro.com/products/system/compact/iot/sys-e100-8q.cfm>

³⁰⁴ <https://www.amazon.com/SUPERMICRO-SYS-E100-8Q-Details-SuperMicro-Compact/dp/B0176UEQUG>

³⁰⁵ <http://www.tbsdtv.com/products/tbs2910-matrix-arm-mini-pc.html>

³⁰⁶ <http://shop.tbsdtv.com/dvbs2-tv-tuner-pci-c-/matrix-arm-mini-pc-p-97.html>

³⁰⁷ <http://androidpcreview.com/tronsmart-draco-aw80-review/2117/>

³⁰⁸ <https://www.amazon.com/Tronsmart-AW80-Allwinner-Android-Bluetooth/dp/B00PU0Y8T4>

³⁰⁹ <https://en.wikipedia.org/wiki/UDOO>

³¹⁰ <https://en.wikipedia.org/wiki/UDOO>

³¹¹ <https://www.kickstarter.com/projects/udoo/udoo-x86-the-most-powerful-maker-board-ever>

³¹² <https://www.kickstarter.com/projects/udoo/udoo-x86-the-most-powerful-maker-board-ever>

³¹³ <http://www.compulab.co.il/utilite-computer/web/utilite-models>

³¹⁴ <http://www.compulab.co.il/utilite-computer/web/utilite-models>

³¹⁵ <http://www.compulab.co.il/utilite-computer/web/utilite2-models>

³¹⁶ <http://www.compulab.co.il/utilite-computer/web/utilite2-models>

³¹⁷ <http://uk.rs-online.com/web/p/single-board-computers/7689782/>

³¹⁸ <http://uk.rs-online.com/web/p/single-board-computers/7689782/>

³¹⁹ <http://www.viaspringboard.com/via-vab-600.html>

³²⁰ <http://www.viaspringboard.com/products/via-vab-600-springboard-kit.html>

³²¹ <http://www.wandboard.org/>

³²² <http://www.wandboard.org/>

³²³ <https://www.graperain.com/ARM-Embedded-S5P4418-Single-Board-Computer/>

³²⁴ https://graperain.en.alibaba.com/product/60521145030-801964843/Graperain_G4418_SBC_Quad_Core_ARM_Board_Linux.html

³²⁵ <https://www.graperain.com/ARM-Embedded-S5P6818-Single-Board-Computer/>

³²⁶ https://graperain.en.alibaba.com/productgrouplist-803068171/6818_series.html

2.pielikums. OpenALPR un saistīto bibliotēku instalācijas skripts

(openalpr_install_v3.sh)

```
#!/bin/bash
# izveido izvaddatu - LOG datni alpr install.log
exec >>(tee alpr_install.log)
exec 2>&1
# ieraksta datumu LOG datnē
date
# atjauniņa Pi, jo pretējā gadījumā var būt problēmas ar bibliotēku kompilēšanu
sudo apt-get update && sudo apt-get upgrade -y --force-yes
date
# lejupielādē un uzinstalē nepieciešamās bibliotēkas priekš tesseract
date
sudo apt-get install -y --force-yes autoconf build-essential git cmake pkg-config
date
sudo apt-get install -y --force-yes autoconf automake libtool
date
sudo apt-get install -y --force-yes autoconf-archive
date
sudo apt-get install -y --force-yes pkg-config
date
sudo apt-get install -y --force-yes libpng12-dev
date
sudo apt-get install -y --force-yes libjpeg8-dev
date
sudo apt-get install -y --force-yes libjpeg-dev
date
sudo apt-get install -y --force-yes libtiff5-dev
date
sudo apt-get install -y --force-yes zlib1g-dev
date
sudo apt-get install -y --force-yes libjasper-dev
date
sudo apt-get install -y --force-yes libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
date
sudo apt-get install -y --force-yes libxvidcore-dev libx264-dev libgdk2.0-dev libatlas-base-
dev
date
sudo apt-get install -y --force-yes gfortran python2.7-dev python3-dev
date
sudo apt-get install -y --force-yes libcurl4-openssl-dev liblog4cplus-1.0-4 liblog4cplus-dev
uuid-dev
date
sudo apt-get install -y --force-yes uv4l uv4l-raspicam uv4l-raspicam-extras uv4l-server uv4l-
mjpegstream
date
sudo apt-get install -y --force-yes libleptonica-dev

# noklonē openalpr bibliotēkas datnes:
date
cd /home/pi
git clone https://github.com/openalpr/openalpr.git
date
# izveido direktoriju un lejupielādē saistītās bibliotēkas
date
cd openalpr
mkdir libraries
cd libraries

echo "git clone https://github.com/tesseract-ocr/tesseract.git"
date
git clone --depth 1 https://github.com/tesseract-ocr/tesseract.git
date
git clone https://github.com/tesseract-ocr/tessdata.git
date
wget http://www.leptonica.com/source/leptonica-1.74.1.tar.gz
date
wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.1.0.zip
date
wget -O opencv_contrib.zip https://github.com/Itseez/opencv_contrib/archive/3.1.0.zip

# atarhivē leptonica
date
tar -zxvf leptonica-1.74.1.tar.gz
# atarhivē opencv-2.4.9.zip
date
unzip opencv.zip
```

```

date
unzip opencv_contrib.zip

# kompilē leptonica:
date
cd leptonica-1.74.1
date
sudo ./configure
date
sudo make -j4
date
sudo make install

# kompilē tesseract:
cd ../tesseract
date
sudo apt-get install -y --force-yes autoconf automake libtool
date
sudo ./autogen.sh
date
sudo apt-get install -y --force-yes autoconf-archive
date
sudo ./configure --enable-debug
LDFLAGS="-L/usr/local/lib" CFLAGS="-I/usr/local/include"
date
sudo make -j4
date
sudo make install
date
sudo ldconfig

# instalē openCV 3 ar Python 2.7
cd /home/pi/openalpr/libraries/
date
wget https://bootstrap.pypa.io/get-pip.py
date
sudo python get-pip.py
date
sudo pip install virtualenv virtualenvwrapper
date
sudo rm -rf ~/.cache/pip
date
echo 'export WORKON_HOME=$HOME/.virtualenvs' >> ~/.profile
echo 'source /usr/local/bin/virtualenvwrapper.sh' >> ~/.profile
date
source ~/.profile
date
mkvirtualenv cv
source ~/.profile
date
workon cv
date
pip install numpy

cd /home/pi/openalpr/libraries/opencv-3.1.0
date
mkdir build
cd build
date
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D INSTALL_C_EXAMPLES=OFF
-D INSTALL_PYTHON_EXAMPLES=ON -D
OPENCV_EXTRA_MODULES_PATH=/home/pi/openalpr/libraries/opencv_contrib-3.1.0/modules -D
BUILD_EXAMPLES=ON ..
date
sudo make -j4
date
sudo make install
date
sudo ldconfig
cd ../virtualenvs/cv/lib/python2.7/site-packages/
date
ln -s /usr/local/lib/python2.7/site-packages/cv2.so cv2.so

# kompilē openALPR:
cd /home/pi/openalpr/src

# nano CMakeLists.txt
# pievieno šādas rindas sākumā, kur jau ir SET() apgalvojumi
# SET(OpenCV_DIR "/home/pi/openalpr/libraries/opencv-2.4.11/release")
# SET(Tesseract_DIR "/home/pi/openalpr/libraries/tesseract")

```

```
# darma to, izmantojot komandu SED
date
sed -i '11 i SET(Tesseract_DIR "/home/pi/openalpr/libraries/tesseract")'
/home/pi/openalpr/src/CMakeLists.txt
sed -i '11 i SET(OpenCV_DIR "~/openalpr/libraries/opencv-3.1.0/build")'
/home/pi/openalpr/src/CMakeLists.txt

date
sudo cmake ./
date
sudo make -j4
date
sudo make install
date
sudo cp -r /usr/local/lib/* /lib

# notestē OpenALPR bibliotēku, izmantojot standartattēlu ar ASV numura zīmi
date
wget http://plates.openalpr.com/ea7the.jpg
date
alpr -c us ea7the.jpg
date
```

3.pielikums. Kustības sensora programmas kods (record.py)

```
#!/usr/bin/python

# Šī (record.py) ir kustības sensora programma, kas analizē pikseļu izmaiņas attēlā,
# lai noteiktu kustību. Kustības gadījumā tiek ģenerēti attēli, izmantojot datumu un laiku JPG
datnes nosaukumā.

progVer = "ver 1.0"

import os
mypath=os.path.abspath(__file__) # atrod pilnu šīs programmas ceļu - atrašanās vietu
baseDir=mypath[0:mypath.rfind("/")+1] # izvelk tikai ceļa atrašanās vietu (izņemot skripta
nosaukumu)
baseFileName=mypath[mypath.rfind("/")+1:mypath.rfind(".")]
progName = os.path.basename(__file__)

# Pārbauda vai eksistē konfigurācijas datne un importē to, ja eksistē.
# Izvada kļūdas paziņojumu, ja neeksistēCheck
configFilePath = baseDir + "config.py"
if not os.path.exists(configFilePath):
    msgStr = "ERROR - Missing config.py file - Could not find Configuration file %s" %
(configFilePath)
    showMessage("readConfigFile", msgStr)
    quit()
else:
    # Nolasa konfigurācijas mainīgos no datnes config.py
    from config import *

if verbose:
    print("----- Loading Python Libraries -----")
else:
    print("Note: verbose=False (Disabled) Set verbose=True to Display Detailed Messages.")

# importē pārējās python bibliotēkas, kas nepieciešamas tālāk programmas darbībai.
import sys
import glob
import time
import datetime
import picamera
import picamera.array
import numpy as np
import pyexiv2
from PIL import Image
from PIL import ImageFont
from PIL import ImageDraw
from fractions import Fraction

#=====  
# Sistēmas mainīgie  
# Nevajadzētu būt vajadzībai mainīt  
#=====  
SECONDS2MICRO = 1000000 # Tiek izmantots, lai konvertētu sekundes uz mikrosekundēm  
nightMaxShut = int(nightMaxShut * SECONDS2MICRO) # default=5 sec  
nightMinShut = int(nightMinShut * SECONDS2MICRO) # zemākais kameras slēgšana iestatījums  
pārejai no dienas uz nakti (vai otrādāk)  
testWidth = 128 # rgb attēlu straumes platums, ko izmanto kustības sensora un  
dienas / nakts izmaiņām  
testHeight = 80 # rgb attēlu straumes augstums, ko izmanto kustības sensora un  
dienas / nakts izmaiņām  
daymode = False # default - pēc noklusējuma vienmēr vajadzētu būt False  
progNameVer = "%s %s" %(progName, progVer)  
motionPath = baseDir + motionDir # Šajā direktoriņā tiks saglabāti attēli pēc kustības  
motionNumPath = baseDir + motionPrefix + baseFileName + ".dat" # datne *.dat, lai saglabātu  
mainīgā currentCount lielumu  
timelapsePath = baseDir + timelapseDir # Šajā direktoriņā paredzēts glabāt Time Lapse (laika  
sprīža) attēlus pēc kustības  
timelapseNumPath = baseDir + timelapsePrefix + baseFileName + ".dat" # datne *.dat, lai  
saglabātu mainīgā currentCount lielumu  
lockFilePath = baseDir + baseFileName + ".sync"  
  
#-----  
def shut2Sec (shutspeed):  
    shutspeedSec = shutspeed/float(SECONDS2MICRO)  
    shutstring = str("%.3f sec") % ( shutspeedSec )  
    return shutstring
```

```

#-----
def showTime():
    rightNow = datetime.datetime.now()
    currentTime = "%04d%02d%02d_%02d:%02d:%02d" % (rightNow.year, rightNow.month,
rightNow.day, rightNow.hour, rightNow.minute, rightNow.second)
    return currentTime

#-----
def showMessage(functionName, messageStr):
    if verbose:
        now = showTime()
        print ("%s %s - %s " % (now, functionName, messageStr))
    return

#-----
def showDots(dotcnt):
    if motionOn and verbose:
        dotcnt += 1
        if dotcnt > motionMaxDots + 2:
            print("")
            dotcnt = 0
        elif dotcnt > motionMaxDots:
            print("")
            stime = showTime() + " ."
            sys.stdout.write(stime)
            sys.stdout.flush()
            dotcnt = 0
        else:
            sys.stdout.write('.')
            sys.stdout.flush()
    return dotcnt

#-----
def checkConfig():
    if not motionOn and not timelapseOn:
        msgStr = "Warning - Both Motion and Timelapse are turned OFF - motionOn=%s
timelapseOn=%s"
        showMessage("checkConfig", msgStr)
    return

#-----
def logToFile(dataToAppend):
    if logDataToFile:
        logFilePath = baseDir + baseFileName + ".log"
        if not os.path.exists(logFilePath):
            open(logFilePath, 'w').close()
            msgStr = "Create New Data Log File %s" % logFilePath
            showMessage(" logToFile", msgStr)
        filecontents = dataToAppend
        f = open(logFilePath, 'a')
        f.write(filecontents)
        f.close()
    return

#-----
def displayInfo(motioncount, timelapsecount):
    if verbose:
        print("")
        print("Note: To Send Full Output to File Use command - python -u ./%s | tee -a
log.txt" % progName)
        print(" Set logDataToFile=True to Send checkIfDay Data to File %s.log" %
progName)
        print("")
        print("%s" % progNameVer)
        print("----- Settings -----")
        print("")
        print("Config File .. Title=%s" % configTitle)
        print(" config-template filename=%s" % configName)
        print("Image Info ... Size=%ix%i Prefix=%s VFlip=%s HFlip=%s Preview=%s" %
(imageWidth, imageHeight, imageNamePrefix, imageVFlip, imageHFlip, imagePreview))
        shutStr = shut2Sec(nightMaxShut)
        print(" Low Light. twilightThreshold=%i nightMaxShut=%s nightMaxISO=%i
nightSleepSec=%i sec" % (twilightThreshold, shutStr, nightMaxISO, nightSleepSec))
        print(" No Shots . noNightShots=%s noDayShots=%s" % (noNightShots, noDayShots))
        print("Motion ..... On=%s Prefix=%s threshold=%i(How Much) sensitivity=%i(How
Many)" % (motionOn, motionPrefix, threshold, sensitivity))
        print(" forceTimer=%i min(If No Motion)" % (motionForce/60))
        print(" Number of previous images to use to check for motion =%i" %
(motionAverage))
        print(" Use video port for motion image capture? %s" % (useVideoPort))

```

```

    print("          motionPath=%s" % (motionPath))
    if motionQuickTLOn:
        print("    Quick TL . motionQuickTLOn=%s    motionQuickTLTimer=%i sec
motionQuickTLInterval=%i sec (0=fastest)" % (motionQuickTLOn, motionQuickTLTimer,
motionQuickTLInterval))
    else:
        print("    Quick TL . motionQuickTLOn=%s    Quick Time Lapse Disabled" %
(motionQuickTLOn))
        print("Time Lapse ... On=%s    Prefix=%s    Timer=%i sec    timeLapseExit=%i sec
(0=Continuous)" % (timelapseOn, timelapsePrefix, timelapseTimer, timelapseExit))
        print("          timelapsePath=%s" % (timelapsePath))
        if createLockFile:
            print("gdrive Sync .. On=%s    Path=%s    Note: syncs for motion images only." %
(createLockFile, lockFilePath))
            print("Logging ..... verbose=%s (Details to Console)    logDataToFile=%s" % (
verbose, logDataToFile ))
            print("          logfilePath=%s" % ( baseDir + baseFileName + ".log" ))
            print("----- Log Activity -----")
        checkConfig()
    return

#-----
def checkImagePath():
    # Pārbauda attēlu direktorijas un izveido tās, ja tās jau neeksistē
    if motionOn:
        if not os.path.isdir(motionPath):
            msgStr = "Creating Image Motion Detection Storage Folder" + motionPath
            showMessage ("checkImagePath", msgStr)
            os.makedirs(motionPath)
    if timelapseOn:
        if not os.path.isdir(timelapsePath):
            msgStr = "Creating Time Lapse Image Storage Folder" + timelapsePath
            showMessage ("checkImagePath", msgStr)
            os.makedirs(timelapsePath)
    return

#-----
def getCurrentCount(numberpath, numberstart):
    # Izveido datni *.dat, lai saglabātu mainīgo currentCount vai nolasītu datni, ja tā jau
eksistē
    # Izveido numberPath datni, ja tā neeksistē
    if not os.path.exists(numberpath):
        msgStr = "Creating New File " + numberpath + " numberstart=" + str(numberstart)
        showMessage("getCurrentCount", msgStr)
        open(numberpath, 'w').close()
        f = open(numberpath, 'w+')
        f.write(str(numberstart))
        f.close()
    # Nolasa numberPath datni, lai iegūtu pēdējo ciparu pēc kārtas
    with open(numberpath, 'r') as f:
        writeCount = f.read()
        f.closed
        try:
            numbercounter = int(writeCount)
        except ValueError: # Atrasta bojāta datne *.dat, jo nevar konvertēt uz integer
            # Mēģina noteikt, vai šī ir kustība vai timelapse (laika sprādis pēc noklusējuma
konfigurācijas datnē netiek izmantots)
            if numberpath.find(motionPrefix) > 0:
                filePath = motionPath + "/*.jpg"
                fprefix = motionPath + motionPrefix + imageNamePrefix
            else:
                filePath = timelapsePath + "/*.jpg"
                fprefix = timelapsePath + timelapsePrefix + imageNamePrefix
            try:
                # Skenē visjaunāko datni attēlu direktorijā un mēģina iegūt numbercounter
                newest = max(glob.iglob(filePath), key=os.path.getctime)
                writeCount = newest[len(fprefix)+1:newest.find(".jpg")]
            except:
                writeCount = numberstart
            try:
                numbercounter = int(writeCount)+1
            except ValueError:
                numbercounter = numberstart
            msgStr = "Invalid Data in File " + numberpath + " Reset numbercounter to " +
str(numbercounter)
            showMessage("getCurrentCount", msgStr)
            f = open(numberpath, 'w+')
            f.write(str(numbercounter))
            f.close()

```

```

        f = open(numberpath, 'r')
        writeCount = f.read()
        f.closed
        numbercounter = int(writeCount)
    return numbercounter

#-----
def postImageProcessing(filename):
    # Šo izmanto, ja ir nepieciešams parādīt laika tekstu tieši uz attēla
    rightNow = datetime.datetime.now()
    if createLockFile and motionOn:
        createSyncLockFile(filename)

#-----
def getImageName(path, prefix):
    # veido attēlu datņu nosaukumus, izmantojot datetime yyyyymmdd-hhmmss formātā
    rightNow = datetime.datetime.now()
    filename = "%s/%s%04d%02d%02d-%02d%02d%02d.jpg" % ( path, prefix ,rightNow.year,
rightNow.month, rightNow.day, rightNow.hour, rightNow.minute, rightNow.second)
    return filename

#-----
def takeDayImage(filename):
    # Uzņem dienas attēlu, izmantojot exp=auto un awb=auto
    with picamera.PiCamera() as camera:
        camera.resolution = (imageWidth, imageHeight)
        time.sleep(0.1) # iemieg uz īsu brīdi, lai kamera varētu iegūt korekcijas
        iestatījumus
        if imagePreview:
            camera.start_preview()
            camera.vflip = imageVFlip
            camera.hflip = imageHFlip
            camera.rotation = imageRotation # Labāk lietot mainīgos imageVFlip un imageHFlip
            # dienas Automātiskais režīms
            camera.exposure_mode = 'auto'
            camera.awb_mode = 'auto'
            camera.capture(filename, use_video_port=useVideoPort)
        msgStr = "Size=%ix%i exp=auto awb=auto %s" % (imageWidth, imageHeight, filename)
        dataToLog = showTime() + " takeDayImage " + msgStr + "\n"
        logToFile(dataToLog)
        showMessage(" takeDayImage", msgStr)
    return

#-----
def takeNightImage(filename):
    dayStream = getStreamImage(True)
    dayPixAve = getStreamPixAve(dayStream)
    currentShut, currentISO = getNightCamSettings(dayPixAve)
    # Uzņemt vājas gaismas - nakts attēlu (ieskaitot krāsas zonas)
    with picamera.PiCamera() as camera:
        # Uzņemt Zemas Gaismas Attēlu
        # Uzstādīt framerate 1/6 fps, pēc tam iestatīt aizveri
        camera.resolution = (imageWidth, imageHeight)
        if imagePreview:
            camera.start_preview()
            camera.vflip = imageVFlip
            camera.hflip = imageHFlip
            camera.rotation = imageRotation # Labāk lietot mainīgos imageVFlip un imageHFlip
            camera.framerate = Fraction(1, 6)
            camera.shutter speed = currentShut
            camera.exposure_mode = 'off'
            camera.iso = currentISO
            # Dodiet kameram pietiekami ilgu laiku, lai nomērītu AWB
            # (tā vietā var arī lietot fiksētu AWB)
            time.sleep(nightSleepSec)
            camera.capture(filename)
        shutSec = shut2Sec(currentShut)
        msgStr = "Size=%ix%i dayPixAve=%i ISO=%i shut=%s %s" % ( imageWidth, imageHeight,
dayPixAve, currentISO, shutSec, filename )
        dataToLog = showTime() + " takeNightImage " + msgStr + "\n"
        logToFile(dataToLog)
        showMessage(" takeNightImage", msgStr)
    return

#-----
def takeQuickTimeLapse(motionPath, imagePrefix, daymode, motionNumPath):
    msgStr = "motion Quick Time Lapse for %i sec every %i sec" % (motionQuickTLTimer,
motionQuickTLInterval)
    showMessage("Main", msgStr)
    checkTimeLapseTimer = datetime.datetime.now()

```

```

keepTakingImages = True
filename = getImageName(motionPath, imagePrefix)
while keepTakingImages:
    yield filename
    rightNow = datetime.datetime.now()
    timelapseDiff = (rightNow - checkTimeLapseTimer).total_seconds()
    if timelapseDiff > motionQuickTLTimer:
        keepTakingImages=False
    else:
        filename = getImageName(motionPath, imagePrefix)
        time.sleep(motionQuickTLInterval)

#-----
def createSyncLockFile(imagefilename):
    # Ja nepieciešams, izveido bloķēšanas datni (lock file), lai norādītu apstrādāt
    nepieciešamās datnes
    if createLockFile:
        if not os.path.exists(lockFilePath):
            open(lockFilePath, 'w').close()
            msgStr = "Create gdrive sync.sh Lock File " + lockFilePath
            showMessage(" createSyncLockFile", msgStr)
            rightNow = datetime.datetime.now()
            now = "%04d%02d%02d-%02d%02d%02d" % ( rightNow.year, rightNow.month, rightNow.day,
            rightNow.hour, rightNow.minute, rightNow.second )
            filecontents = now + " createSyncLockFile - " + imagefilename + " Ready to sync using
            sudo ./sync.sh command."
            f = open(lockFilePath, 'w+')
            f.write(filecontents)
            f.close()
        return

#-----
def getStreamImage(isDay):
    # Iegūst attēlu straumi atmiņā, balstoties uz dienas režīmu (daymode)
    with picamera.PiCamera() as camera:
        time.sleep(0.5)
        camera.resolution = (testWidth, testHeight)
        with picamera.array.PiRGBArray(camera) as stream:
            if isDay:
                camera.exposure_mode = 'auto'
                camera.awb_mode = 'auto'
                camera.capture(stream, format='rgb', use_video_port=useVideoPort)
            else:
                # Uzņem zemas gaismas attēlu
                # Uzstāda kadrus sekundē uz 1/6fps, tad aizver slēģi
                # pātrina līdz 6s
                camera.framerate = Fraction(1, 6)
                camera.shutter_speed = nightMaxShut
                camera.exposure_mode = 'off'
                camera.iso = nightMaxISO
                # Dodiet kameram pietiekami ilgu laiku, lai nomērītu AWB
                # (tā vietā var arī lietot fiksētu AWB)
                time.sleep( nightSleepSec )
                camera.capture(stream, format='rgb')
            return stream.array

#-----
def getStreamPixAve(streamData):
    # Aprēķināt vidējo pikseļu vērtības noteiktā plūsmā (izmanto, lai noteiktu dienas / nakts
    vai krāsas nosacījumus)
    pixAverage = int(np.average(streamData[...],1))
    return pixAverage

#-----
def getNightCamSettings(dayPixAve):
    # Aprēķināt attiecību starp slēdža un ISO vērtībām, pielāgojot tās
    if dayPixAve <= twilightThreshold:
        ratio = ((twilightThreshold - dayPixAve)/float(twilightThreshold))
        outShut = int(nightMaxShut * ratio)
        outISO = int(nightMaxISO * ratio)
    else:
        ratio = 0.0
        outShut = nightMinShut
        outISO = nightMinISO
    # veic dažas robežpārbaudes, lai izvairītos no iespējamām problēmām
    if outShut < nightMinShut:
        outShut = nightMinShut
    if outShut > nightMaxShut:
        outShut = nightMaxShut
    if outISO < nightMinISO:

```

```

        outISO = nightMinISO
    if outISO > nightMaxISO:
        outISO = nightMaxISO
    msgStr = "dayPixAve=%i ratio=%.3f ISO=%i shut=%i %s" % ( dayPixAve, ratio, outISO,
outShut, shut2Sec(outShut))
    showMessage(" getNightCamSettings", msgStr)
    return outShut, outISO

#-----
def checkIfDay(currentDayMode, dataStream):
    # Mēģina noteikt, vai tā ir diena, nakts vai krēsla
    dayPixAverage = 0
    if currentDayMode:
        dayPixAverage = getStreamPixAve(dataStream)
    else:
        dayStream = getStreamImage(True)
        dayPixAverage = getStreamPixAve(dayStream)

    if dayPixAverage > twilightThreshold:
        currentDayMode = True
    else:
        currentDayMode = False
    return currentDayMode

#-----
def timeToSleep(currentDayMode):
    if noNightShots:
        if currentDayMode:
            sleepMode=False
        else:
            sleepMode=True
    elif noDayShots:
        if currentDayMode:
            sleepMode=True
        else:
            sleepMode=False
    else:
        sleepMode=False
    return sleepMode

#-----
def checkForTimelapse (timelapseStart):
    # Pārbauda, vai timelapse taimeris ir beidzies
    rightNow = datetime.datetime.now()
    timeDiff = ( rightNow - timelapseStart).total_seconds()
    if timeDiff > timelapseTimer:
        timelapseStart = rightNow
        timelapseFound = True
    else:
        timelapseFound = False
    return timelapseFound

#-----
def checkForMotion(data1, data2):
    # Atrrod kustību starp divām datu plūsmām, balstoties uz jutīguma (sensitivity) un sliekšņa
(threshold) iestatījumiem
    motionDetected = False
    pixColor = 3 # red=0 green=1 blue=2 all=3 default=1
    if pixColor == 3:
        pixChanges = (np.absolute(data1-data2)>threshold).sum()/3
    else:
        pixChanges = (np.absolute(data1[...,pixColor]-data2[...,pixColor])>threshold).sum()
    if pixChanges > sensitivity:
        motionDetected = True
    if motionDetected:
        dotCount = showDots(motionMaxDots + 2) # New Line
        msgStr = "Found Motion - threshold=" + str(threshold) + " sensitivity=" +
str(sensitivity) + " changes=" + str(pixChanges)
        showMessage("checkForMotion", msgStr)
    return motionDetected

#-----
def dataLogger():
    # Nomainiet main() ar šo funkciju, lai logotu dienas / nakts pixAve datnē
    # Tādā gadījumā mainīgajam logDataToFile ir jābūt True datnē config.py
    # Var arī izdzēst record.log datni, lai iztīrītu iepriekšējos log datus
    print("dataLogger - One Moment Please ....")
    while True:
        dayStream = getStreamImage(True)
        dayPixAverage = getStreamPixAve(dayStream)

```

```

    nightStream = getStreamImage(False)
    nightPixAverage = getStreamPixAve(nightStream)
    logdata = "nightPixAverage=%i dayPixAverage=%i twilightThreshold=%i " % (
nightPixAverage, dayPixAverage, twilightThreshold )
    showMessage("dataLogger",logdata)
    now = showTime()
    logdata = now + " " + logdata
    logToFile(logdata)
    time.sleep(1)
return

#-----
def Main():
    # Galvenās programmas Main() inicializācijas un loģika cilpa
    dotCount = 0 # Skaitītājs priekš showDots() parādīšanas, ja nav atrasta kustība (sistēma
strādā)
    checkImagePath()
    timelapseNumCount = 0
    motionNumCount = 0
    try: # ja motionAverage nav iekļauta config datnē (lai tā darbojas ar iepriekšējām
versijām)
        global motionAverage
        if motionAverage > 1:
            resetSensitivity = sensitivity*150 # nomainīto pikseļu skaits, lai izraisītu
atiestatīšanu uz fona vidējo
            if resetSensitivity > testHeight*testWidth*2:
                resetSensitivity = testHeight*testWidth*2 # ierobežot resetSensitivity
            else:
                motionAverage = 1
        except NameError:
            motionAverage = 1
    try:
        global useVideoPort
        useVideoPort = useVideoPort
    except NameError:
        useVideoPort = False
    moCnt = "non"
    tlCnt = "non"
    if timelapseOn:
        if timelapseNumOn:
            timelapseNumCount = getCurrentCount(timelapseNumPath, timelapseNumStart)
            tlCnt = str(timelapseNumCount)
    displayInfo(moCnt, tlCnt)
    daymode = False
    data1 = getStreamImage(True).astype(float) # Visām funkcijām ir jāstrādā ar float int
vietā - vienkārši aizņem vairāk atmiņas
    daymode = checkIfDay(daymode, data1)
    data2 = getStreamImage(daymode) # inicializē data2, lai lietotu to galvenajā (main) cilpā
    if not daymode:
        data1 = data2.astype(float)
    timelapseStart = datetime.datetime.now()
    checkDayTimer = timelapseStart
    checkMotionTimer = timelapseStart
    forceMotion = False # Izmanto, lai piespiestu veidoties kustības attēlam, ja nav
kustības priekš motionForce laiks pārsniegts
    msgStr = "Entering Loop for Time Lapse and/or Motion Detect Please Wait ..."
    showMessage("Main", msgStr)
    dotCount = showDots(motionMaxDots) # atiestata kustības punktiņus
    # Programmas galvenā cilpa (loop) sākas šeit. Izmantot Ctrl+C, lai izietu, ja programma
tiek izpildīta no termināļa sesijas.
    while True:
        # izmanto data2, lai pārbaudītu daymode pret data1, kas var būt vidējais, kas mainās
lēni, un data1 var neatjaunināties
        if daymode != checkIfDay(daymode, data2): # ja daymode ir mainījies, atiestata
fonu, lai izvairītos no neesošas kustības (false motion trigger)
            daymode = not daymode
            data2 = getStreamImage(daymode) # iegūst jaunu plūsmu
            data1 = data2.astype(float) # atiestata (reset) fonu (background)
        else:
            data2 = getStreamImage(daymode) # šis dabūt otro plūsmu kustības analīzei
            rightNow = datetime.datetime.now() # atjauno rightNow laiku
            if not timeToSleep(daymode): # Neuzņemt attēlus, ja noNightShots vai noDayShots
iestatījums ir derīgs (validācija)
                if timelapseOn:
                    takeTimeLapse = checkForTimelapse(timelapseStart)
                    if takeTimeLapse:
                        timelapseStart = datetime.datetime.now() # atiestata (reset) time lapse
laiku
                    dotCount = showDots(motionMaxDots + 2) # atiestata kustības punktus
                    msgStr = "Scheduled Time Lapse Image - daymode=" + str(daymode)

```

```

        showMessage("Main", msgStr)
        imagePrefix = timelapsePrefix + imageNamePrefix
        filename = getImageName(timelapsePath, imagePrefix)
        if daymode:
            takeDayImage(filename)
        else:
            takeNightImage(filename)
        dotCount = showDots(motionMaxDots)
    if motionOn:
        # SVARĪGI: - Nakts kustības uztveršana var nedarboties ļoti labi
        # ņemot vērā ilgo ekspozīcijas laiku un vājo apgaismojumu (var mēģināt
uzstādīt sarkanu zaļā vietā)
        # Tāpat var būt nepieciešams īpašs nakts sliekšņa (threshold) un jutīguma
(sensitivity) iestatījums (nepieciešama lielāka testēšana)
        motionFound = checkForMotion(data1, data2)
        if motionAverage > 1 and (np.absolute(data2-data1)>threshold).sum() >
resetSensitivity:
            data1 = data2.astype(float)
        else:
            data1 = data1+(data2-data1)/motionAverage
            rightNow = datetime.datetime.now()
            timeDiff = (rightNow - checkMotionTimer).total_seconds()
            if timeDiff > motionForce:
                dotCount = showDots(motionMaxDots + 2) # Jauna Rinda
                msgStr = "No Motion Detected for " + str(motionForce / 60) + " minutes.
Taking Forced Motion Image."
                showMessage("Main", msgStr)
                checkMotionTimer = rightNow
                forceMotion = True
            if motionFound or forceMotion:
                dotCount = showDots(motionMaxDots + 2) # Jauna Rinda
                checkMotionTimer = rightNow
            if forceMotion:
                forceMotion = False
            imagePrefix = motionPrefix + imageNamePrefix
            # pārbauda, vai kustības Quick Time Lapse opcija ir ieslēgta (On). Šī
opcija aizstāj motionVideoOn
            if motionQuickTLOn and daymode:
                filename = getImageName(motionPath, imagePrefix)
                with picamera.PiCamera() as camera:
                    camera.resolution = (imageWidth, imageHeight)
                    camera.vflip = imageVFlip
                    camera.hflip = imageHFlip
                    time.sleep(.1)
                    # izmanto pakāpienu, lai cilpotu laika sprīža (time lapse) secībā,
bet tas nešķiet ātrāks, jo attēli tiek rakstīti
                    camera.capture_sequence(takeQuickTimeLapse(motionPath,
imagePrefix, daymode, motionNumPath))
                    # motionNumCount = getCurrentCount(motionNumPath, motionNumStart)
            else:
                filename = getImageName(motionPath, imagePrefix)
                if daymode:
                    takeDayImage(filename)
                else:
                    takeNightImage(filename)
                motionNumCount = postImageProcessing(filename)
            if motionFound:
                dotCount = showDots(motionMaxDots)
            else:
                dotCount = showDots(dotCount) # parāda progresu punktus, ja nav noteikta
kustība
        return

#-----
if __name__ == '__main__':
    try:
        if debug:
            dataLogger()
        else:
            Main()
    finally:
        print("")
        print("+++++")
        print("%s - Exiting Program" % progName)
        print("+++++")
        print("")

```

4.pielikums. Kustības sensora konfigurācijas parametri (config.py)

```
# Lietotāja Konfigurācija: mainīgo iestatījumi attēliem un programmas record.py
darbībai
# Mērķis: Kustības Noteikšanas Kamera - Kustības Sensors
# Izveidots: 20-Feb-2017
# Autors: Haralds Plass, izmantojot atvērto kodu, ko publicējis Clude Pageau
(https://github.com/pageauc/pi-timolo)

configTitle = "shoot default configuration motion and timelapse 720p images"
configName = "shoot-default-config"

# Šiem iestatījumiem ir jābūt False, ja skriptu darbina fonā, izmantojot /etc/init.d
dēmonu
verbose = True # Nosūta izvērstu programmas darbības informāciju uz
konsoli. Iestata uz False, ja skriptu darbina, izmantojot dēmonu
logDataToFile = True # reģistrē disgnostikas datus diska datnē vēlākai
pārskatīšanai. Pēc noklusējuma (default) = False
debug=False # Ieliek atklūdošanas režīmā, atgriez pikseļu vidējos
datus noskaņošanai

# Attēlu Iestatījumi
imageNamePrefix = '' # Visu attēlu datņu priekšējais prefikss, piemēram, 'mol-
'
imageWidth = 1024 # Pilna izmēra attēla platums pikseļos, default=1024
imageHeight = 768 # Pilna izmēra attēla augstums pikseļos, default=768
imageVFlip = False # Sagriezt attēlu Vertikāli, default=False
imageHFlip = False # Sagriezt attēlu Horizontāli, default=False
imageRotation=90 # Sagriezt/Rotēt attēlu. Iespējamās vērtības: 0, 90, 180
& 270. Default=0. Uzstādījums 90 uzlikts dēļ CCTV kameras izgriezuma cauruma.
imagePreview = False # Attēla parādīšana uz Raspberry Pi pieslēgta monitora.
default=False
noNightShots = False # Neuzņemt attēlus naktī. default=False
noDayShots = False # Neuzņemt attēlus dienā. default=False

# Vāja apgaismojuma Iestatījumi
nightMaxShut = 5.5 # default=5.5 sec Lielākais kameras slēgšanas
eksponēcijas laiks.
# SVARĪGI, ka 6 sec darbojas dažreiz, bet reizēm var
bloķēt RPI tā, ka būs nepieciešama restartēšana un attīrīšana
nightMinShut = .001 # default=.002 sec Mazākais kameras slēgšanas
eksponēcijas laiks, lai pārietu no dienas uz nakti vai otrādāk
nightMaxISO = 800 # default=800 Maksimālais kameras ISO nakts iestatījums
nightMinISO = 100 # mazākais kameras ISO iestatījums, lai pārietu no dienas
uz nakti vai otrādāk
nightSleepSec = 10 # default=10 sekundes - Laika periods, lai ļautu kamerai
aprēķināt vājas gaismas AWB
twilightThreshold = 40 # default=40 Gaismas līmenis, lai darbinātu dienas /
nakts pāreju pie krēslas

# web lapas iestatījumi
webDir = "/home/pi/web/" # direktoriya, kur glabājas web lapas skripts
webImageDir = "/home/pi/web/images/" # direktoriya, kur glabājas attēli
webImageFileName = "live.jpg" # attēla datnes nosaukums aktuālajam kameras
uzņemtajam attēlam
webImage = "/home/pi/web/images/live.jpg" # direktoriya un aktuālā attēla nosaukums
kopā

# Kustības Noteikšanas Iestatījumi
motionOn = True # True = kustības noteikšana ir ieslēgta. False =
nenoteikt kustību
motionPrefix = "" # Kustības Noteikšanas ģenerēto attēlu prefikss
motionDir = "mol" # Direktoriya, kurā tiks glabāti attēli
threshold = 50 # Cik daudz pikseļiem ir jāizmainās, lai būtu ieskaitīta
kustība. default=10 (1-200). set=20, 200 bija par daudz
sensitivity = 700 # Izmainījušos pikseļu skaits, lai sāktu darboties
kustības sensors - attēlu ģenerēšana. default=300 set=300, 3000 bija par daudz
(nebija kustības)
motionAverage = 1 # Attēlu skaits, kas vidēji jāpārbauda, lai noteiktu
kustību: 1 = vienkārši izmanto pēdējo attēlu. Default=100
```

```

useVideoPort = False           # Izmanto video portu, lai uzņemtu attēlus - ātrāk nekā
attēlu ports. Default=False
motionQuickTLOn = False       # Ja uzstāda True, tad uzņem ātros time lapse secības
attēlus viena attēla vietā (pārraksta motionVideoOn)
motionQuickTLTimer = 1        # Ilgums sekundēs, kas nepieciešams, lai uzņemtu ātros
time lapse secības attēlus pēc sākotnējās kustības noteikšanas. default=10
motionQuickTLInterval = 0    # Laiks starp katru Quick Time lapse laika plūduma
attēlu. 0 ir cik ātri vien iespējams
motionForce = 60 * 60        # Piespiež uzņemt vienu kustības attēlu, ja nav noteikta
kustība norādītajās sekundēs. default=60*60
motionNumOn = False          # True=On (datņu nosaukumu pēc kārtas). Neizmantojam, jo
izmantojam datums-laiks datnes nosaukumā. default=False
motionMaxDots = 100          # Kustības punktu skaits pirms jaunas rindas sākšanas
createLockFile = True        # default=False. Ja True, tad sync.sh izsauks gdrive, lai
sinhronizētu datnes ar google drive tikai tad, ja eksistēs .sync datne
                                # Lock File tiek izmantots, lai noteiktu, vai ir
uzģenerēti kustības attēli. Tas sync.sh var sinhronizēties fonā caur sudo crontab -e

# Time Lapse (laika sprīža) Iestatījumi
timelapseOn = False          # Ieslēdz timelapse. True=On False=Off
timelapseTimer = 5 * 60      # Sekundes starp timelapse attēliem. default=5*60
timelapseDir = "t14"          # Direktorijs, kurā tiks glabāti Time Lapse attēli
timelapsePrefix = ""          # Timelapse attēlu prefikss
timelapseExit = 0 * 60      # Aizvērs programmu pēc noteiktām sekundēm. 0=Continuous
default=0
timelapseNumOn = False       # True=On (datņu nosaukumi, veidoti no cipara pēc
kārtas). Neizmantojam, jo izmantojam datums-laiks datnes nosaukumā. default=False.

```

5.pielikums. Kustības sensora programmas palaišanas skripts (run.py)

```
#!/bin/sh
# Programmas record.py automātiskas palaišanas skripts palaišanai no komandrindas vai
attiecīgu ierakstu /etc/rc.local datnē
# Varētu būt nepieciešams pamainīt sleep (iemigšanas) aizkavēšanās parametru, ja tas
nedarbojas pareizi caur rc.local
# Jāpārliedzinās, vai šis skripts ir izpildāms (ir izpildāmas datnes tiesības) vai jāuzstāda
tās ar komandu:
# chmod +x run.sh
# Šis skripts aizsargā programmas record.py palaišanu vairākas reizes
# jo Raspberry Pi kamera nevar tikt darbināta vairākas reizes vienlaicīgi

progpath="/home/pi/record"
progname="record.py"
proglog="verbose.log"
progsleep=10

echo "$0 ver 1.1 written by Haralds Plass using initial script written by Claude Pageau"
echo "-----"
cd $progpath

# Pārbauda, vai eksistē programma, kuras nosaukums uzstādīts mainīgajā 'progname'
if [ ! -e $progname ] ; then
    echo "ERROR - Could Not Find $progname"
    exit 1
fi

if [ -z "$( pgrep -f $progname )" ]; then
    if [ "$1" = "start" ]; then
        echo "Start $progpath/$progname in Background"
        # laiks sekundēs (uzstādīts mainīgajā 'progsleep'), par ko aizkavēt programmas palaišanu
        # līdz iesāknējas pati iekārta Raspberry Pi Zero (ja darbina no /etc/rc.local)
        echo "Waiting $progsleep seconds ...."
        sleep $progsleep

        # nokomentēt līniju zemāk, ja nav nepieciešams ierakstīt konsoles izvaddatus
        $progpath/$progname >/dev/null 2>&1 &
        # Piezīme: uzstādīt verbose = True datnē config.py
        # tad atkomentēt līniju zemāk, lai veidotos LOG datne
        # echo "Start $progpath/$progname with log to $progpath/$proglog"
        # python -u $progpath/$progname > $progpath/$proglog &
    fi
else
    if [ "$1" = "stop" ]; then
        echo "Stopping $progname ...."
        progPID=$( pgrep -f $progname )
        sudo kill $progPID
    fi
fi

if [ -z "$( pgrep -f $progname )" ]; then
    echo "$progname is Not Running ..."
    echo "To Start $progname execute command below"
    echo "$0 start"
else
    progPID=$(pgrep -f $progname)
    echo "$progname is Running ..."
    echo "PID is $progPID"
    echo "To Stop $progname execute command below"
    echo "$0 stop"
fi
echo "Good Bye Harald's camera record program!"
```

6.pielikums. Google Drive sinhronizācijas skripts attēliem (sync_pics.sh)

```
#!/bin/bash
echo "-----"
# -----
# Šis skripts veic šādas darbības:
# 1) palaiž gdrive tikai tad, ja tas jau nedarbojas
# 2) kontrolē record.sync datni, ko ir izveidojusi record.py programma gadījumā, ja:
# (1) ir izveidojušies jauni sinhronizējami faili un (2) ja parametrs CHECK FOR SYNC FILE=true
# Iznīcina gdrive procesu, ja tas darbojas pārāk ilgi (pēc noklusējuma > 4000 sekundes vai 67
minūtes)
# Ieteicams izpildīt šo skriptu no crontab, piemēram, ik pēc 5 minūtēm, pēc komandas 'sudo
crontab -e' pievienojot crontab šādu rindu:
# */5 * * * * /home/pi/record/sync.sh >/dev/nul
# -----
DIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )" # šī skripta direktorijs
# ----- Mainīgie -----
SYNC_DIR=processed # sinhronizējamā direktorijs, kas atrodas šī skripta
direktorijs
FILES_TO_SYNC='*jpg' # apstrādājamo datņu veidi (* - visi)
CHECK_FOR_SYNC_FILE=true # true, ja ir nepieciešama sync datnes pārbaude, savādāk
uzstādīt false
SYNC_FILE_PATH=$DIR/record.sync # record programmas inicializētas sinhronizācijas apturēšanas
datne
FORCE_REBOOT=false # true, lai restartētu, ja record programma nedarbojas,
savādāk uzstādīt false
WATCH_APP='record.py' # monitorējamā programma priekš FORCE_REBOOT (ja tā
nedarbojas)
# -----
# Nomainīt direktoriju, kas sakrīt ar google drive saknes direktoriju (nepieciešams crontab)
cd $DIR
function do_gdrive_sync()
{
    # Pārbauda, vai eksistē SYNC DIR direktorijs
    if [ ! -d "$DIR/$SYNC_DIR" ] ; then
        echo "ERROR - Local Folder $DIR/$SYNC_DIR Does Not Exist"
        echo " Please Check SYNC_DIR variable and/or Local Folder PATH"
        exit 1
    fi
    # Pārbauda, vai ir sinhronizējamas datnes SYNC direktorijs
    ls -l $DIR/$SYNC_DIR/$FILES_TO_SYNC > /dev/null 2>&1
    if [ ! "$?" = "0" ] ; then
        echo "ERROR - No Matching $FILES_TO_SYNC Files Found in $DIR/$SYNC_DIR"
        exit 1
    fi
    # Pārbauda, vai eksistē attiecīga attālinātā direktorijs google drive
    # un, ja nē, tad izveido to
    echo "STATUS - Check if Remote Folder /$SYNC_DIR Exists"
    echo "-----"
    /usr/local/bin/gdrive file-id $SYNC_DIR
    if [ ! $? -eq 0 ] ; then
        echo "-----"
        echo "STATUS - Creating Remote Folder /$SYNC DIR"
        echo "-----"
        /usr/local/bin/gdrive new --folder $SYNC_DIR
        /usr/local/bin/gdrive file-id $SYNC_DIR
        if [ $? -eq 0 ] ; then
            echo "-----"
            echo "STATUS - Successfully Created Remote Folder /$SYNC DIR"
        else
            echo "-----"
            echo "ERROR - Problem Creating Remote Folder $SYNC_DIR"
            echo " Please Investigate Problem"
            exit 1
        fi
    fi
    echo "-----"
    echo "STATUS - Start gdrive Sync ...."
    echo "STATUS - Local Source Folder - $DIR/$SYNC_DIR"
    echo "STATUS - Remote Destn Folder - /$SYNC DIR"
    echo "STATUS - Files $FILES_TO_SYNC"
    echo "STATUS - Running This May Take Some Time ...."
    echo "STATUS - sudo /usr/local/bin/gdrive push -no-prompt -ignore-conflict
$SYNC DIR/$FILES TO SYNC"
    echo "-----"
    date
    START=$(date +%s)
    sudo /usr/local/bin/gdrive push -no-prompt -ignore-conflict $SYNC_DIR/$FILES_TO_SYNC
    if [ $? -ne 0 ] ; then

```

```

# Pārbauda, vai gdrive sync process ir bijis veiksmīgs
date
echo "-----"
echo "ERROR - gdrive Sync Failed."
echo "      Possible Cause - See gdrive Error Message Above"
echo "      Please Investigate Problem and Try Again"
else
# Ja veiksmīgs, tad parāda laiku un izdzēš sync datni, iesāk OpenALPR procesu un
datubāzes datnes izveidi
date
END=$(date +%s)
DIFF=$((END - START))
echo "-----"
echo "STATUS - $0 Completed Successfully"
echo "STATUS - Processing Took $DIFF seconds"
if [ -e $SYNC_FILE_PATH ] ; then
echo "STATUS - Deleting Sync Lock File $SYNC_FILE_PATH"
rm -f $SYNC_FILE_PATH
fi
fi
}

# pārbauda, vai gdrive jau darbojas, lai izvairītos no vairākām instancēm
if [ -z "$(pgrep -f gdrive)" ] ; then
if [ $CHECK_FOR_SYNC_FILE ] ; then
echo "STATUS - Script Variable Setting CHECK_FOR_SYNC_FILE=true"
if [ -e $SYNC_FILE_PATH ] ; then
# Palaiž gdrive datnēm, kas atrodas direktoriņā $SYNC_DIR
echo "STATUS - Found File $SYNC_FILE_PATH"
do_gdrive_sync
else
echo "STATUS - File Not Found $SYNC_FILE_PATH"
echo "STATUS - No Files to Sync at this Time."
fi
else
echo "STATUS - Script Variable Setting CHECK_FOR_SYNC_FILE=false"
echo "WARNING - No Sync Lock File $SYNC_FILE_PATH Required"
do_gdrive_sync
fi
else
# ja gdrive jau darbojas, tad pārbauda, cik ilgi un iznīcina procesu, ja ir sasniegts
laika limits
GDRIVEPID=$(pgrep gdrive)
if [ -z "$(sudo ps axh -O etimes | grep gdrive | grep -v grep | sed 's/^ *//' | awk '{if
($2 >= 4000) print $1}')" ]
then
RUNTIME=$(sudo ps axh -O etimes | grep gdrive | grep -v grep | sed 's/^ *//' | awk
'{if ($2 > 0) print $2}' | head -1)
echo "STATUS - Another sync.sh has run for \"$RUNTIME\" seconds."
echo "STATUS - Will kill if greater than 4000 seconds."
echo "STATUS - gdrive PID is $GDRIVEPID"
else
echo "WARNING - gdrive has run longer than 4000 seconds so kill PID $GDRIVEPID"
echo "      Killing $GDRIVEPID in 5 seconds"
sleep 5
sudo kill $GDRIVEPID
fi
fi

if $FORCE_REBOOT ; then # pārbauda, vai ir nepieciešama restartēšana
echo "-----"
echo "STATUS - Check $WATCH_APP Run Status ..."
if [ -z "$(pgrep -f $WATCH_APP)" ] ; then
echo "STATUS - $WATCH_APP is NOT Running so reboot"
echo "WARNING - Reboot in 15 seconds Waiting ...."
echo "      ctrl-c to Abort Reboot."
sleep 10
echo "WARNING - Rebooting in 5 seconds"
sleep 5
sudo reboot
else
APP_PID=$(pgrep -f $WATCH_APP)
echo "STATUS - $WATCH_APP is Running PID is $APP_PID"
fi
fi
echo ""
echo "Done ..."
exit

```

7.pielikums. Google Drive sinhronizācijas skripts datiem (sync_data.sh)

```
#!/bin/bash
echo "-----"
# -----
# Šis skripts veic šādas darbības:
# 1) palaiž gdrive tikai tad, ja tas jau nedarbojas
# 2) kontrolē record.sync datni, ko ir izveidojusi record.py programma gadījumā, ja:
# (1) ir izveidojušies jauni sinhronizējami faili un (2) ja parametrs CHECK FOR SYNC FILE=true
# Iznīcina gdrive procesu, ja tas darbojas pārāk ilgi (pēc noklusējuma > 4000 sekundes vai 67
minūtes)
# Ieteicams izpildīt šo skriptu no crontab, piemēram, ik pēc 5 minūtēm, pēc komandas 'sudo
crontab -e' pievienojot crontab šādu rindu:
# */5 * * * * /home/pi/record/sync data.sh >/dev/nul
# -----
DIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )" # šī skripta direktorijs
# ----- Mainīgie -----
SYNC_DIR=mo1 # sinhronizējamā direktorijs, kas atrodas šī skripta
direktorijs
PROC_DIR=processed # direktorijs uz kuru pārvietot OpenALPR un gdrive
apstrādātās attēlu datnes
FILES_TO_SYNC='*csv' # apstrādājamo datņu veidi (* - visi)
CHECK_FOR_SYNC_FILE=true # true, ja ir nepieciešama sync datnes pārbaude, savādāk
uzstādīt false
SYNC_FILE_PATH=$DIR/record.sync # record programmas inicializētas sinhronizācijas apturēšanas
datne
FORCE_REBOOT=false # true, lai restartētu, ja record programma nedarbojas,
savādāk uzstādīt false
WATCH_APP='record.py' # monitorējamā programma priekš FORCE_REBOOT (ja tā
nedarbojas)
# -----
# Nomainīt direktoriju, kas sakrīt ar google drive saknes direktoriju (nepieciešams crontab)
cd $DIR
function do_gdrive_sync()
{
# Pārbauda, vai eksistē SYNC DIR direktorijs
if [ ! -d "$DIR/$SYNC_DIR" ] ; then
echo "ERROR - Local Folder $DIR/$SYNC_DIR Does Not Exist"
echo " Please Check SYNC_DIR variable and/or Local Folder PATH"
exit 1
fi

# Pārbauda, vai ir sinhronizējamas datnes SYNC direktorijs
ls -l $DIR/$SYNC_DIR/$FILES_TO_SYNC > /dev/null 2>&1
if [ ! "$?" = "0" ] ; then
echo "ERROR - No Matching $FILES_TO_SYNC Files Found in $DIR/$SYNC_DIR"
exit 1
fi

# Pārbaude vai eksistē attiecīga attālinātā direktorijs google drive
# un, ja nē, tad izveido to
echo "STATUS - Check if Remote Folder /$SYNC_DIR Exists"
echo "-----"
/usr/local/bin/gdrive file-id $SYNC_DIR
if [ ! $? -eq 0 ] ; then
echo "-----"
echo "STATUS - Creating Remote Folder /$SYNC_DIR"
echo "-----"
/usr/local/bin/gdrive new --folder $SYNC_DIR
/usr/local/bin/gdrive file-id $SYNC_DIR
if [ $? -eq 0 ] ; then
echo "-----"
echo "STATUS - Successfully Created Remote Folder /$SYNC_DIR"
else
echo "-----"
echo "ERROR - Problem Creating Remote Folder $SYNC_DIR"
echo " Please Investigate Problem"
exit 1
fi
fi
echo "-----"
echo "STATUS - Start gdrive Sync ...."
echo "STATUS - Local Source Folder - $DIR/$SYNC_DIR"
echo "STATUS - Remote Destn Folder - /$SYNC_DIR"
echo "STATUS - Files $FILES_TO_SYNC"
echo "STATUS - Running This May Take Some Time ...."
echo "STATUS - Processing pictures JPG in $DIR/$SYNC_DIR and OpenALPR ...."
# inicializē automašīnas numura atpazīšanu (OpenALPR) katrai attēla datnei direktorijs
for file in "$DIR/$SYNC_DIR/"*.jpg; do
```

```

# iegūst datnes nosaukumu bez paplašinājuma un ieraksta mainīgajā full fn
full_fn=${file##*/}
# iegūst tikai datnes nosaukumu bez paplašinājuma un ceļa uz datni, ieraksta mainīgajā
fn
fn=${full_fn%. *}
# atpazīst automašīnas numuru attēlā un ieraksta rezultātu *.txt datnē
alpr -c eu $file > $DIR/$SYNC_DIR/$fn.txt
# izgriež datus par auto numuru no OpenALPR izvaddatnes *.txt 2 rindas
car_num=$(sed -r 's|.*- (.*)confidence.*|\1|' $DIR/$SYNC_DIR/$fn.txt | sed '2q;d')
# izveido datni log.csv, izmantojot šodienas datumu kā prefiksu formātā Y-M-D
# (>, ja tāda neeksistē) un ievada/pievieno (>>, ja eksistē) datus par atpazīto auto
numuru
nowdt=$(date +%Y-%m-%d_%H-%M-%S)
now=$(date +%Y-%m-%d)
log_file="$now_log.csv"
if [ ! -e $DIR/$SYNC_DIR/$log_file ]; then
    echo $fn,$car_num > $DIR/$SYNC_DIR/$log_file
else
    echo $fn,$car_num >> $DIR/$SYNC_DIR/$log_file
fi
# pārvieto apstrādāto attēla failu un auto numura atpazīšanas izvaddatus *.txt uz
direktoriiju $PROC_DIR
mv $file $DIR/$PROC_DIR/
mv $DIR/$SYNC_DIR/$fn.txt $DIR/$PROC_DIR/
done
echo "STATUS - sudo /usr/local/bin/gdrive push -no-prompt -ignore-conflict
$SYNC_DIR/$FILES_TO_SYNC"
echo "-----"
date
START=$(date +%s)
sudo /usr/local/bin/gdrive push -no-prompt -ignore-conflict $SYNC_DIR/$FILES_TO_SYNC
if [ $? -ne 0 ] ; then
    # Pārbauda, vai gdrive sync process ir bijis veiksmīgs
    date
    echo "-----"
    echo "ERROR - gdrive Sync Failed."
    echo " Possible Cause - See gdrive Error Message Above"
    echo " Please Investigate Problem and Try Again"
else
    # Ja veiksmīgs, tad parāda laiku
    date
    END=$(date +%s)
    DIFF=$((END - START))
    echo "-----"
    echo "STATUS - $0 Completed Successfully"
    echo "STATUS - Processing Took $DIFF seconds"
fi
}

# pārbauda, vai gdrive jau darbojas, lai izvairītos no vairākām instancēm
if [ -z "$(pgrep -f gdrive)" ] ; then
    if [ $CHECK_FOR_SYNC_FILE ] ; then
        echo "STATUS - Script Variable Setting CHECK_FOR_SYNC_FILE=true"
        if [ -e $SYNC_FILE_PATH ] ; then
            # Palaiž gdrive datnēm, kas atrodas direktoriājā $SYNC DIR
            echo "STATUS - Found File $SYNC_FILE_PATH"
            do_gdrive_sync
        else
            echo "STATUS - File Not Found $SYNC FILE PATH"
            echo "STATUS - No Files to Sync at this Time."
        fi
    else
        echo "STATUS - Script Variable Setting CHECK_FOR_SYNC_FILE=false"
        echo "WARNING - No Sync Lock File $SYNC FILE PATH Required"
        do_gdrive_sync
    fi
else
    # ja gdrive jau darbojas, tad pārbauda, cik ilgi un iznīcina procesu, ja ir sasniegts
    laika limits
    GDRIVEPID=$(pgrep gdrive)
    if [ -z "$(sudo ps axh -O etimes | grep gdrive | grep -v grep | sed 's/^ *//' | awk '{if
($2 >= 4000) print $1}')" ]
    then
        RUNTIME=$(sudo ps axh -O etimes | grep gdrive | grep -v grep | sed 's/^ *//' | awk
'{if ($2 > 0) print $2}' | head -1)
        echo "STATUS - Another sync.sh has run for \"$RUNTIME\" seconds."
        echo "STATUS - Will kill if greater than 4000 seconds."
        echo "STATUS - gdrive PID is $GDRIVEPID"
    else
        echo "WARNING - gdrive has run longer than 4000 seconds so kill PID $GDRIVEPID"
    fi
fi

```

```

        echo "          Killing $GDRIVEPID in 5 seconds"
        sleep 5
        sudo kill $GDRIVEPID
    fi
fi

if $FORCE_REBOOT ; then # pārbauda, vai ir nepieciešama restartēšana
    echo "-----"
    echo "STATUS - Check $WATCH_APP Run Status ..."
    if [ -z "$(pgrep -f $WATCH_APP)" ] ; then
        echo "STATUS - $WATCH_APP is NOT Running so reboot"
        echo "WARNING - Reboot in 15 seconds Waiting ...."
        echo "          ctrl-c to Abort Reboot."
        sleep 10
        echo "WARNING - Rebooting in 5 seconds"
        sleep 5
        sudo reboot
    else
        APP_PID=$(pgrep -f $WATCH_APP)
        echo "STATUS - $WATCH_APP is Running PID is $APP_PID"
    fi
fi
echo ""
echo "Done ..."
exit

```