

Prosessipäiväkirja

17.6.19 Komponentit ja suunnittelu

Pohdimme erilaisia vaihtoehtoja mikroprosessorin suhteen ja päädyimme toistaiseksi **Raspberry Pi Zeroon** (<https://www.adafruit.com/product/3708>). Tämä mikroprosessori on riittävän pieni, luultavasti tarpeeksi tehokas ja sisältää riittävän määrän kytkentöjä ainakin yhteen kameraan ja nettiyhteyteen. Jos tarvitsemme enemmän potkua toimintojen pyörittämiseksi, käytämme isokokoisempaa Raspberry Pi 3:sta.

Langatonta yhteyttä varten mahdollisesti geneerinen nettitikku tai Pycomin **FiPy** (<https://pycom.io/product/fipy/>). FiPy on halpa ja yhteensopiva Zeron kanssa, suhteellisen pienikokoinen ja näyttää tukevan wifiä sekä suomalaista mobiiliverkkoa. Antenniksi FiPy:n LTE-M (<https://pycom.io/product/lte-m-antenna-kit/>), sillä se on riittävän pieni ja varmasti yhteensopiva FiPy:n kanssa. Tarvitsemme vielä SIM-kortin yhteyden muodostamiseksi.

Pallo vaikuttaa järkevimmältä muodolta, sillä se on periaatteessa helpoin heittää ja ominaisuuksiin pystyy vaikuttamaan ulkokuorella. Halutessa esimerkiksi vierimistä voidaan estää jollain liitettävillä ulkoisilla palasilla. Vastaavasti vieriminen voi taas olla haluttu ominaisuus. Kuutiota olisi paljon hankalampi saada vierimään ulkoisilla lisäyksillä. Kuutio vaatii periaatteessa myös kuusi kameraa ja on mahdollista, että emme saa mahdutettua tai koodattua kuutta kameraa toimimaan tämän kurssin puitteissa. Pallon voi rakentaa alle kuudellakin kameralla, joten valinnanvaraa on enemmän. Muut geometriset muodot, kuten pyramidit ja raketit tuntuvat turhan monimutkaisilta.

Painotus tulee olemaan laitteen ominaisuus väistämättä, sillä painoa on erittäin hankala jakaa tasaisesti pienen pallon sisälle. Näin ollen painotus kannattaa ottaa jo suunnittelussa huomioon ja hyödyntää sitä toiminnallisuuden parantamisessa. Esimerkiksi painottamalla pallon tiettyä osaa voimme ennustaa, että tasaisella alustalla kamera luultavasti pysähtyy niin, että painotettu osa on alhaalla. Voimme ottaa tämän huomioon kameroiden paikkojen suunnittelussa, tässä tapauksessa esimerkiksi kameran osoittaminen alas painotettuun suuntaan olisi hölmöä. Jos emme saa siirrettyä painopistettä tarpeeksi pelkkien komponenttien avulla, voi olla perusteltua lisätä jotain raskasta ainetta kuoren rakenteeseen niin, että pallo käyttäytyy halutulla tavalla laskeutumisessa.

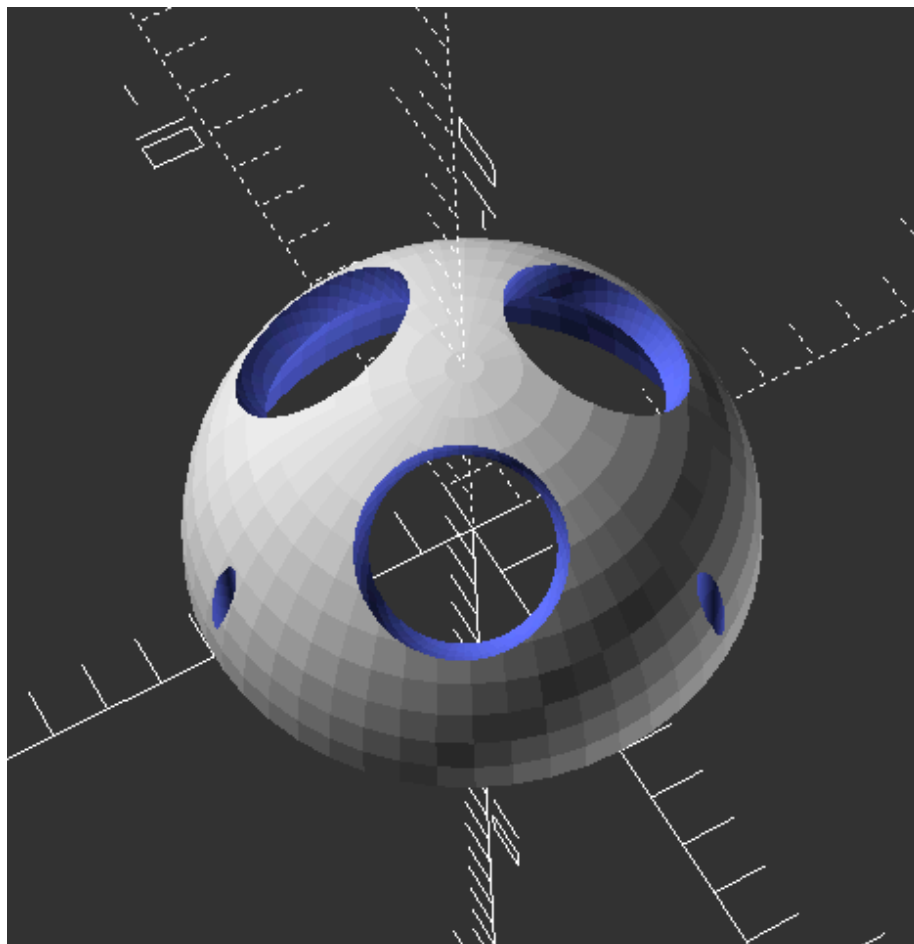
Mielessämme käväisi idea siitä, että painotettu osa mahdollistaisi pallon ylemmän kuoren pyörimisen, jolloin kameroita voisi suunnata vielä laskeutumisen jälkeen. Tällöin pallo jakautuisi ikäänkuin ylempään ja alempaan osaan, ylhäällä olisivat kevyet komponentit johtoineen ja alemmassa osassa kaikki painava. Pienen ajatusleikin kautta tajusimme kuitenkin, että kaksiosainen pallo hajoaisi helpommin laskeutuessa, joten emme nyt toistaiseksi lähde toteuttamaan tätä ideaa.

Assistenttimme tuntee entisen opiskelijan, joka on tehnyt diplomi-tasoisen tutkielman usean kameran käyttämisestä Raspberryn tyylisillä mikroprosessoreilla. Juttelimme tämän henkilön kanssa ja alla on listattuna merkittävimmät pointit tästä keskustelusta:

- Suunnittelimme alunperin, että voisimme kytkeä usean laadukkaan kameran ns. multiplekseriin, joka käytännössä päästäisi vain yhden kameran kerrallaan läpi mikroprosessorille ja käyttäjä voisi valikoida mitä kameraa haluaa katsella. Asiantuntijan mukaan tämä on erittäin hankala toteuttaa, sillä kamerat lähettävät niin paljon raaka-dataa (luokkaa gigabittejä sekunnissa) ja kameraväylät sisältävät niin paljon ns. "mustaa magiaa", että pienikin virhe koodissa tai multiplekserissä rikkoo koko systeemin ja tekee kameroista käyttökelvottomia, eikä syytä ole kovinkaan helppo löytää. Päätimme siis jättää tämän idean pois projektistamme.
- Vaihtoehdoksi asiantuntija tarjosi valmiiden kameraväylien käyttöä sellaisinaan. Raspberry Zerossa on vain yksi kameraväylä, mutta isommissa versioissa on kaksi kameraväylää.
- Voisimme myös käyttää useampaa Raspberryä, mutta pallon koko, sähkön kulutus, lämmöntuotto ja datan käyttö langattomalla verkolla lisääntyisivät merkittävästi.
- Yleisesti ottaen kameraväylät ovat haastavia väyliä, sillä ne on suunniteltu hyödyntämään kameran dataa koko kennosta hyvällä frameratella. Niiden käyttäminen yhteensopivilla kameroilla on mahdollista, mutta liikkumatilaa ei ole kovinkaan paljoa, jos ei halua syöksyä syvälle matalan tason koodaamiseen.
- Jos tulevaisuudessa halutaan rakentaa projekti, joka käyttää useampaa kunnon kameraa, NVidialla on olemassa kehitysalustoja, jotka tukevat tätä ominaisuutta paremmin.
- SPI-väylän käyttö still-kuvilla on mahdollista, mutta tämä väylä on hidas, eikä välttämättä ole optimaalinen tälle projektille.
- **Parhaaksi ideaksi osoittautui USB-webcamien käyttö**, sillä Raspberry/Linux tukee natiivina useampaa kameraa USB-väylän kautta ja kamerat pakkaavat datan automaattisesti, joten voimme streamata paremmalla laadulla langattoman yhteyden kautta. Mahdollisesti läppäreiden webcamit ovat sopivan pieniä ja laadukkaita.
- USB-liitäntä kannattaa kolvata tilan säästämiseksi.
- Tarvitaan USB-hub, joka syöttää virtaa webcameille, sillä Raspberryn USB-virta ei ole riittävä. Onneksi Raspberryt ja useimmat USB-hubit toimivat viidellä voltilla, joten kytkemisen samaan akkuun pitäisi olla helpohkoa.
- Pystymme siis toteuttamaan alkuperäisen multipleksaus-idean yksinkertaisemmin USB-webcameilla koodaamalla mikä kamera streamaa kerrallaan. Näin voimme streamata parempilaatuista kuvaa, kuin useammalla kameralla yhtä aikaa. Voi olla mahdollista myös antaa käyttäjälle valinta usean tai yhden kameran väliltä.
- USB-protokollan rakenteen takia webcamit saattavat vaihtaa paikkanumerointia bootatessa, mutta tämä ei välttämättä ole ongelma, sillä kamera pyörii lennon aikana niin paljon, että käyttäjän täytyy joka tapauksessa tarkistaa jokaisen kameran suuntaus erikseen.

- Toteutuksessa ja suunnittelussa pitää ottaa huomioon myös lämmöntuotto ja mahdollinen jäähtytys. Raspit käyvät kuumana, joten useampi mikroprosessori voi osoittautua ongelmalliseksi, erityisesti haastavissa ympäristöissä, kuten palavassa rakennuksessa. Jonkinlainen ilmanvaihto voi olla paikallaan, mutta tämä
- Jos päädyimme käyttämään USB-webcameja, IR-ominaisuudet jäävät luultavasti pois, ellemme löydä jotain teollisuustason webcameja, jotka kykenevät näkemään näitä taajuuksia. Tätä pimeänäön puutetta voidaan kompensoimalla suuntaamalla suunniteltuja LED-valoja kameroiden osoittamaan suuntaan, valaisemaan pimeää ympäristöä tarvittaessa. Sotilaskäyttöön tämä voi sellaisenaan olla huono ratkaisu, mutta pelastustehtävissä mahdollisesti monikäyttöinen ratkaisu. Käytännössä kuitenkin tämän prototyypin toteutuksessa ratkaisu on riittävän hyvä.
- Muut kehitysalustat kuin Raspberry Pi:t ovat paljon huonommin dokumentoituja, joten tämän kurssin puitteissa ei ole järkeä etsiä muita vaihtoehtoja paremman videokuvan takia.

Tämän kokouksen jälkeen päädyimme lähtemään kehittämään palloa, jossa on kolme webcamia eri suuntauksilla ja laajakulma-linsseillä. Kokeilemme erilaisten Raspberry-versioiden toimintaa webcamien kanssa ja valitsemme sen, joka toimii parhaiten pienimmässä koossa. Jos pallo kasvaa ihan liian suureksi, meidän täytyy ehkä ruveta miettimään jotain heitto-mekanismia, kuten liipaisimella irtaavaa kahvaa. Tällöin pallo heitettäisiin kahvasta ja liipasinta painettaisiin sopivalla hetkellä, jotta pallo lentäisi halutulla tavalla ja kahva jäisi käteen. Tämä ratkaisu on toki hieman erikoinen ja mahdollisesti epäolennainen, jos lopullinen tuote saadaan kuitenkin pienempään kokoon.



Tutkimme aiemmin ehdotetun Pycom FiPyn dokumentaatiota ja huomasimme seuraavanlaisen tarkennuksen:

SIM card

If you intend on using the LTE CAT-M1 or NB-IoT connectivity of the FiPy you will need to insert a SIM card into your FiPy. It should be noted that the FiPy does not support regular LTE connectivity and you may require a special SIM. It is best to contact your local cellular providers for more information on acquiring a LTE CAT-M1/NB-IoT enabled nano SIM.

Tarvitsemme siis tietynlaisen IoT SIM-kortin jos haluamme saada tämän komponentin toimimaan. Lisäksi komponentti on niin uusi, että googlettaminen ei paljastanut dokumentaation lisäksi ainuttakaan valmista esimerkki projektia, joten kehitys saattaisi tyssätä yksinkertaisiin ongelmiin. Vaihtoehtoisesti voisimme käyttää nettitikkoa, esimerkiksi Huaweiin E3372H, joka on ilmeisesti saatu toimimaan jonkin Linux-distrin kanssa ihan näppärästi tavallisella SIM-kortilla. Emme tosin tiedä, että toimiiko tämä nettitikku Raspberryn kanssa.

Projektiin sopivin **USB-webcam** vaikuttaisi olevan ELP 480P 30W04MT (<http://www.webcamerusb.com/elp-vga-640480-30fps-color-image-cmos-ov7725-sensor-usb-camera-with-36mm-lens-p-149.html>). Kamerassa on säädettävä linssi 30 - 150 asteen kulmille ja pystymme pienentämään piirilevyn kokoa leikkaamalla reunat pois. Yli 120 asteen kulma on tärkeä, jotta pystymme välttämään katvealueita. Kamera tukee Linuxia natiivina, joten ongelmia ei pitäisi tulla. Käytännössä pieni koko, USB-protokolla ja linssien kulma olivat tärkeimmät kriteerit kameran valinnassa. IR-tukea tässä ei ole.

3.6mm lens



20.6.19 Lisätietoa langattomaan yhteyteen

Tutkimme FiPyn dokumentaatiota lisää sekä juttelimme informaatioteknologian opiskelijan kanssa FiPyn 4G-ominaisuuksista ja tajusimme, että moduuli tarjoaa korkeintaan vain parin megan yhteyden, joten kaista ei tule mitenkään riittämään tähän projektiin. Ainoaksi 4G vaihtoehdoksi jää tavallinen USB-nettitikku, joka pitäisi saada toimimaan Raspberry Zeron kanssa. Raspberryssä on sisäänrakennettu WiFi, joten aloitamme kehityksen siitä ja tutkimme sivussa mahdollisia nettitikkuja. Bluetooth-yhteyden kaistanleveys on samaa kokoluokkaa FiPyn kanssa, joten **WiFi on toistaiseksi ainoa varteenotettava vaihtoehto**.

24.6.19 Rungon ensimmäinen prototyyppi

Tulostimme ensimmäisen mallin rungon prototyypistä. Pallon ulkoinen halkaisija on 10 cm ja pallon sisäinen halkaisija on 8 cm. Pallo tuntuu ihan sopivan kokoiselta jokaisen kädessä. Lopullisen materiaalin on oltava jotain kestävämpää ja joustavampaa, mutta pystymme tämän mallin avulla sovittamaan komponentit sisälle sopivaan muodostelmaan ja testaamaan painotuksen vaikutusta pallon toimintaan. Tavoitteena on tuottaa nyt toimiva prototyyppi tämän rungon avulla ja kehittää iskunkestävää rakennetta myöhemmin.



25.6.19 Langaton yhteys

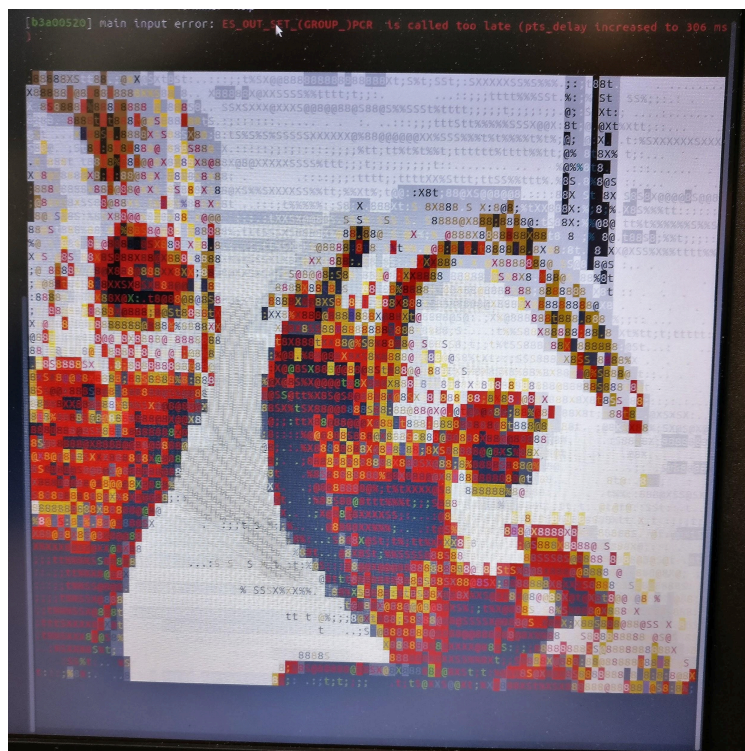
Raspberry Zero W ja monet pienet komponentit olivat saapuneet postissa. Kameran saapuvat vasta parin päivän päästä. Käynnistimme Zeron uusimmalla Rasbian Stretch -käyttöjärjestelmällä ja pyörittelimme Wifin Access Pointin virallisten ohjeiden mukaan toimintaan. Wifi siirtää nyt dataa kumpaankin suuntaan sujuvasti, mutta tarvitsemme kunnon ohjelmiston videon siirtämiseen.

Lisäksi saimme kaiuttimesta kuulumaan ääntä ihan sopivan lujaa pienen vahvistinpiiriin kautta. Testasimme myös LEDejä onnistuneesti.

Laitoimme metalliset painot pallon rungon sisälle ja liimasimme ne pohjaan kiinni. Menimme ulos heittelemään palloa nurmikolle. Totesimme, että painotus yksinään ei riitä kaikissa tapauksissa kääntämään palloa oikein päin. Tasaisella maastolla, kuten puhtaalla lattialla, painotus toimi paljon paremmin ja pallo suoristui aina oikein päin. Pohdimme, että pieni soikeus yläkanteen ja litteys alakanteen voisi tehdä pallolle hyvää. Yläkannen täytyy olla mahdollisimman kevyt alakanteen nähden. Mahdollisesti myös pieni solenoidi yläkannessa voisi auttaa tökkäämällä maata jos pallo jää päällelleen.

26.6.19 Videostreamin setuppaamista langattomasti

Lopulliset kamerat eivät ole vielä saapuneet, mutta kytkimme geneerisen USB-webcamin Raspberry Zeroon ja kokeilimme erilaisia tapoja streamata videota reaaliajassa wifin yli. Ajattelimme, että helpoin tapa olisi käyttää VLC Media Playerin sisäänrakennettua stream-ominaisuutta, sillä VLC on asennettavissa myös Raspberrylle. Painimme tämän toiminnon kanssa konsolin sekä VLC:n graafisen käyttöliittymän avulla, mutta emme jostain syystä saanut kunnon videokuvaa streamattua. Onnistuimme kyllä saamaan videokuvaa wifin yli ASCII-koodattuna konsolin sisällä, joka oli sinällään yllättävän humoristinen käänne, mutta emme lopulta saaneet parempia tuloksia aikaiseksi.

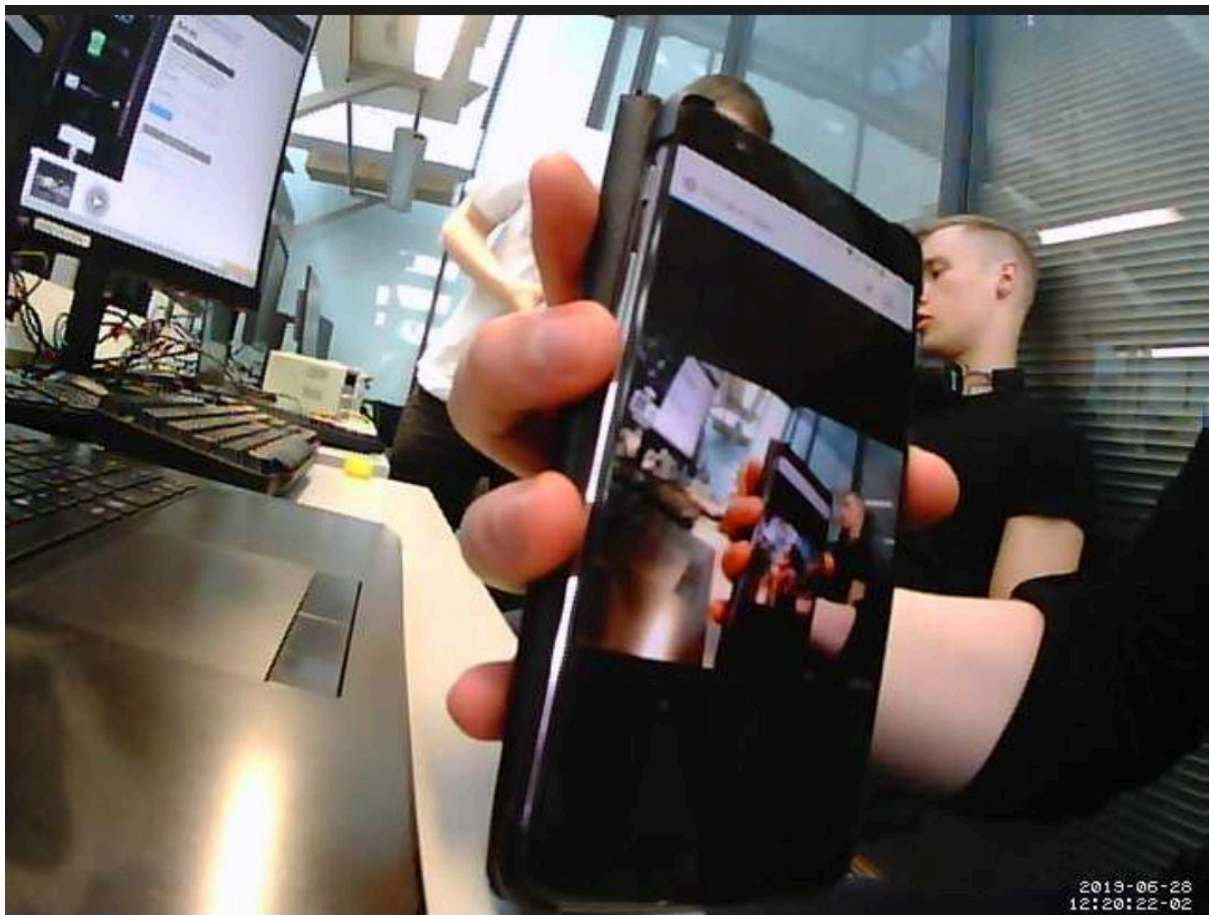


27.6.19 Videostream muilla työkaluilla

Aloitimme rakentamaan Netcastilla putkia (pipe) useita lähteitä varten, eli jokaiselle webcamille oma putki. Netcast vaikuttaisi mahdollistavan parhaan suorituskyvyn lopullisessa prototyypissä. Saimme still-kuvia näistä läpi, mutta tämä vaatii vielä lisää kehitystä. Tutkimme samalla myös OpenCV:tä, joka mahdollistaisi kuvantunnistuksen. Emme välttämättä kehitä tätä ominaisuutta sen enempää, mutta olisi hauska saada se toimivaksi ominaisuudeksi mahdollista jatkokehitystä varten. Testailimme myös Raspin ääniulostuloja GPIO-pinneistä, mutta äänenlaatu oli kauhea. Tarvitsemme ilmeisesti I2S-vahvistimen, jos haluamme parantaa äänenlaatua tai vaihtoehtoisesti voisimme kytkeä ulkoisen äänikortin, johon yhdistäisimme sekä mikin, että kaiuttimen. Päädyimme kuitenkin toistaiseksi siihen, että kaikki ulostulot tulevat GPIO-pinneistä ja sisääntulot USB-porteista.

28.6.19 Videostream wifin kautta onnistuneesti

Saimme webcamin streamaamaan wifin kautta hyvälaatuista "videota" yhden kuvan sekunnissa Raspberryn Motion-kirjaston avulla. Kävelimme kännykän kanssa toisiin huoneisiin ja pääsimme n. 10m päähän linnuntietä niin, että väliin jäi betoniseiniä ja hissikuilu. Vasta tällöin yhteys alkoi pätkimään. Latenssi oli huomattava, noin 5-15 sekuntia ja frameratea ei saanut säädettyä suuremmaksi millään. Ilmeisesti Motion-kirjasto on tarkoitettu lähinnä valvontakameroita varten, joten meidän täytyy ruveta kehittämään projektiin sopivampaa ratkaisua..



1. - 5.7. Langattoman yhteyden jatkokehittämistä

Jatkoimme kehitystä Netcatin ja OpenCV:n avulla, sillä nämä työkalut vaikuttivat tarjoavan parhaan mahdollisen suorituskyvyn videostreamiin. Kohtasimme kuitenkin suuria ongelmia datan siirtämisen suhteen, emmekä saaneet mitään järkevää edistystä moneen päivään. Suurin ongelma oli esimerkkien puute ja dokumentaation vaikeaselkoisuus. Emme ole kovin perehtyneitä aiheeseen, joten tämä työkalu osoittautui liian haastavaksi tähän projektiin, mutta Netcat/OpenCV voisi tarjota mielenkiintoisen vaihtoehdon kokeneemmille kehittäjille.

Kokeilimme myös MotionEyeOS-käyttöjärjestelmää, joka on suunniteltu streamaamaan useita videostreameja nettiin valvontakameramaisesti. Käyttöjärjestelmä oli kuitenkin valmis paketti, eikä varsinaisesti mahdollistanut muutoksia tai jatkokehitystä. Yritimme myös jatkaa kehitystä ffmpeg-kirjaston avulla, mutta emme koskaan saaneet tätä kirjastoa asennettua, sillä asennuksessa kesti niin kauan ja saimme VLC:n toimimaan odotellessa.. Ilmeisesti olimme käyttäneet vääriä IP-osoitteita ensimmäisissä yrityksissä.

VLC streamasi kolmea kameraa näppärästi 5-10fps, 640 x 480 resoluutiolla ja noin kolmen sekunnin latenssilla Wifiin yli Raspberry Pi 3 B+:lla. Kokeilimme samaa Zerolla, mutta Zero oli täysin riittämätön tähän ratkaisuun (vain yksi ydin ja kolme kameraa saman USB portin kautta). Zeron streami jäättyi pienemmilläkin resoluutioilla, eli teho ei yksinkertaisesti riittänyt. Joudumme siis jatkamaan kehitystä hieman isokokoisemmalla Raspberry Pi 3 B+:lla, joka johtaa hieman suurempaan pallon kokoon.



8. - 12.7. Streamauksen ensimmäinen toimiva iteraatio

Videostreamin lisäksi saimme nyt myös äänen kulkemaan kumpaankin suuntaan VLC:tä käyttämällä. Kirjoitimme scriptin Raspberryille, joka avaa viisi VLC-instanssia ja aloittaa streamauksen automaattisesti bootauksesta. Kolme instanssia vastaa videoiden streamauksesta ja kaksi instanssia audion kaksisuuntaisesta streamauksesta. Lämpärin puolella kirjoitimme vastaavan scriptin toiseen suuntaan, jotta yhteyden käynnistäminen on helppoa. Käytännössä streamaus on nyt saavuttanut ensimmäisen toimivan iteraationsa.

Suoritamme myöhemmin vielä testauksia ja optimointeja ja dokumentoimme ne tähän, mutta alustavasti kykenimme keskustelemaan tämän streamin välityksellä muutaman sekunnin latenssista huolimatta.

Rungon kehitys lähti myös käyntiin kovalla teholla. Suunnittelimme kameroille kehikon, joka kiinnitetään pallon sisälle ruuveilla. Pohdimme, että pallon ulkokuoren voisi rakentaa metallista ja kiinnittää muoviseen sisäkuoreen niin, että väliin jää kumisia pehmusteita tai jousituksia. Ulkokuoreen kiinnitetään ledejä, joiden toiminnallisuutta voi tällä hetkellä ohjailla läppäritä SSH-yhteyden kautta komennoilla.

Testasimme myös alustavasti akunkestoa nykyisillä toiminnoilla. 3,7v 2000 mAh LiPo-akku jaksoi pyörittää 5 voltilla streamausta noin 10 - 15 minuuttia LEDien välkkyessä ja kaiuttimen toistaessa musiikkia. Tilaa voisi olla vielä isommallekin akulle, mutta etenemme nyt toistaiseksi tällä kokoonpanolla.

15. - 19.7. Rungon kehitystä ja toiminnallisuuden lisäystä

Tämän viikon suurin fokus pyöri rungon rakenteen ympärillä. Suurin osa ajasta käytettiin rungon 3D-printtauksen suunnitteluun ja metallisen ulkokuoren hiomiseen. Olemme löytäneet halvan ja näppärän vaihtoehdon metallikuoren toteuttamiseen: Ikean metallinen kulho, jonka halkaisija on noin 12 cm ja joka muodostaa täydellisen ympyrän, kun kaksi kulhoa laitetaan vastakkain. 3D-printatut pallonpuoliskot suunniteltiin mahtumaan näiden kulhojen sisälle ja porasimme metalliin sopivan kokoisia aukkoja, jotta kamerat, ruuvit ja langaton signaali pääsevät kuoresta läpi.

Mietimme, että metallinen kuori saattaisi vaimentaa langattoman signaalin tehokkuutta, mutta käytännössä emme huomanneet mitään vaikutusta streamiin pallon ollessa suljettuna. Suurempi antenni, joka olisi esimerkiksi upotettu ulkokuoren metalliin, olisi yleisesti ihan hyvä idea, mutta ilmeisesti Raspberry Pi 3 ei tue ulkoisen antennin liittämistä ilman työlästä nikkarointia ja käyttöjärjestelmän ohjelmointia. Toivottavasti sisäänrakennettu antenni on riittävän hyvä tähän prototyyppiin.

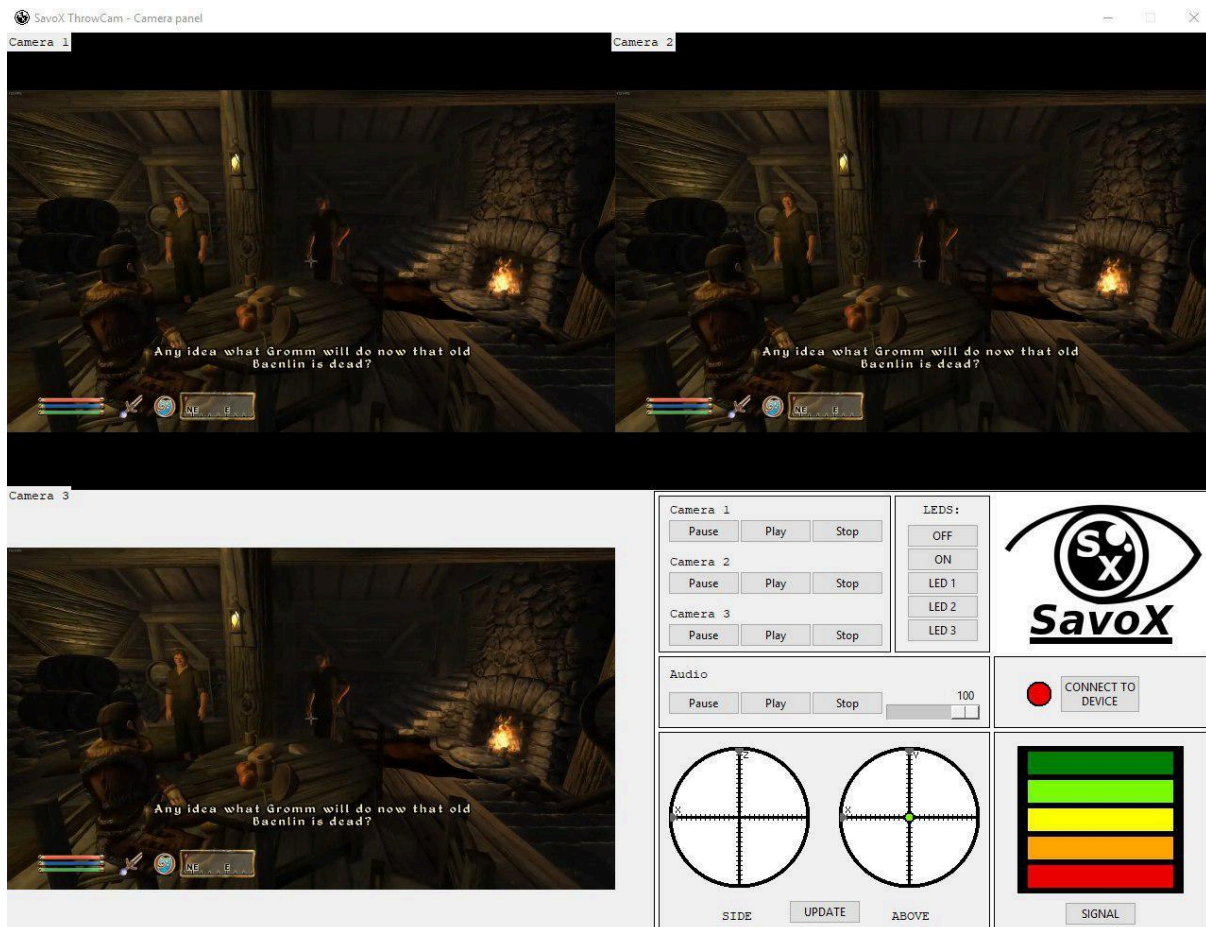


22.7. - 26.7. Käyttöliittymän ohjelmointia

Aloitimme viikon laittamalla asento-anturin toimintaan. Käytimme yksinkertaista MMA8452-kiihtyvyysanturia asennon tunnistamisessa. Pienellä Python-koodilla pallon asento pystytään laskemaan hyvin tarkasti, kun pallo on pysähtynyt maahan. Aiomme luoda mittari-ohjelmiston, joka näyttää pallon yleisen kallistuksen sivusta katsottuna, sekä kallistumisen suunnan ylhäältä katsottuna. Lisäksi koodasimme Pythonilla pienen ohjelman, joka lukee wifi-signaalin vahvuutta ja arvioi sen perusteella laitteen etäisyyden.

Lähes kaikki pallon toiminnallisuus on nyt periaatteessa valmiina, mutta ilman yhtenäistä ohjelmistoa, josta kaikkea voisi hallita näppärästi. Tästä syystä keskitimme kaiken kehitystyön sopivan käyttöliittymän ohjelmointiin. Löysimme Python-VLC kirjaston, jonka avulla pitäisi pystyä käyttämään VLC:n ominaisuuksia Python-koodin avulla.

Kirjaston esimerkeistä löytyi ohjelma, joka käytännössä upottaa VLC:n lukeman videon Tkinter-nimisen GUI-kirjaston luomaan ikkunaan ja mahdollistaa usean videon samanaikaisen katselun yhdestä ikkunasta. Esimerkistä löytyi myös painikkeita ja liukuja, jotka asetimme sopivaan järjestykseen pallon toimintojen ohjaamista varten. OpenSSH-kirjastolla pitäisi pystyä siirtämään dataa Raspberryn ja tietokoneen välillä ja käskyttää palloa tekemään haluttuja asioita.



29.7. - 2.8. Rungon pehmustuksen kehitystä ja testailua

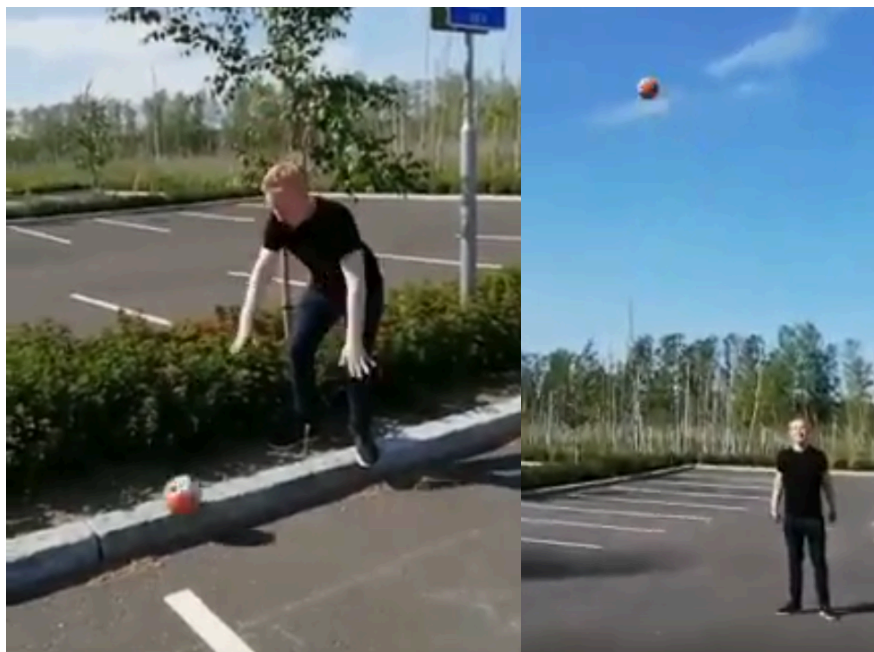
Testasimme alkuvuikosta käyttöliittymää onnistuneesti. Kaikki toiminnot olivat ohjattavissa yhden ohjelman kautta ja kameran lähettämä video ja ääni vastaanotettiin aivan kuten pitääkin.

Toimintojen ollessa käytännössä valmiita, keskittimme loppuvuikon energian rungon kehitykseen. Runko koostuu tällä hetkellä 3D-printatusta sisäkuoresta, jonka sisällä on tietynlainen tukikehikko, johon elektroniikka ja kamerat ovat kiinnitettynä. 3D-printattu muovi on sinällään aika heiveröistä, eikä kestä teräviä iskuja kovin hyvin, joten olemme hankkineet Ikeasta metallisen kulhon, joka sopii muovisen kuoren päälle. Kuori kiinnitetään luultavasti liiman ja ruuvien avulla sisäkuoreen, mutta emme vielä ole tätä tehneet.

Metallinen kuori olisi sinällään ihan kestävä, mutta heiteltyämme useita erilaisia prototyypppejä, totesimme laskeutumisen iskun hajoittavan muoviset sisärakenteet kohtuullisen helposti. Tästä syystä haimme K-Raudasta putkieristeesi tarkoitettua vaahtomuovia ja leikkasimme siitä sopivan muotoisia palasia niin, että saimme pallon ympäröityä vaahtomuovilla tasaisesti. Lopputulos ei ollut erityisen esteettinen, eikä heittotuntuma tykännyt parista lisäsentistä halkaisijaan. Meidän käsille 10 - 12 cm halkaisija tuntuu vielä luontevalta, mutta tämä uusi 14 - 15 cm halkaisija on jo selkeästi liian paksu luontevaan heittämiseen. Käytännössä palloa oli helpoin heittää kahdella kädellä.

Testasimme uuden suojakuoren toimintaa erittäin aggressiivisella heittelyllä, ja pallo kesti yllättävän hyvin. Vasta kun palloa heiteltiin katukivetyksen kulmaan useaan otteeseen, sisäkuori lohkesi ohuimmasta kohtaa, siitä mistä kaksi puolipalloa kiinnitetään toisiinsa ruuveilla. Huomattavaa on kuitenkin se, ettei pallon sisällä ollut elektroniikkaa, joten sisärakenteet eivät olleet todellisen rasituksen alaisina. Heittelimme palloa kuitenkin huolimattoman kovaa, joten todellisessa käytössä näin suuria voimia ei luultavasti pääsisi ilmenemään.

Ideana vaahtomuovi on loistava, sillä se lisäsi pallon kestävyyttä erittäin paljon. Käytännössä jokin taktisempi materiaali ja kuviointi voisi olla parempi vaihtoehto lopulliseen tuotteeseen.



5. - 9.8. Komponenttien kolvaaminen

Kolvasimme komponentit kiinni toisiinsa ja parantelimme käyttöliittymän toimintaa. Laite toimi kolvauksen jälkeen normaalisti, eikä mitään ihmeellistä tapahtunut tämän asian suhteen. Seuraava askel on komponenttien sovittaminen pallon sisälle.

Testasimme myös PLA Flex -materiaalia 3D-printatussa sisäkuoressa ja tulokset olivat erinomaisia. Verrattuna perinteiseen materiaaliin, joka lohkeaisi pudotessaan muutamasta metristä, Flex kesti yllättävän hyvin edellisen viikon testien mukaista heittelyä ilman metallista suojakuorta tai vaahtomuovia. Pallo ei aluksi hajonnut minkäänlaisesta heittelystä, mutta materiaali väsyi lopulta ja lohkesi muutamasta kohtaa. Periaatteessa kuori olisi voinut olla vielä toimintakunnossa.

Tämä materiaali on siis ihanteellinen sisäkuoreen ja elektroniikan tukirakenteisiin, sillä metallikuori vaahtomuoveineen ei kykene pysäyttämään impulssin etenemistä pallon sisäpuolelle, joten sisärakenteiden tulee olla joustavia ja tarpeeksi kiinteitä elektroniikan suojaamiseksi. Tämä materiaali yhdistettynä metallikuoreen ja vaahtomuovi-pehmustukseen voisi tarjota erittäin hyvää iskunkestävyyttä tulevilla testeillä.

12.8. - 16.8. Pallon kasaaminen

Progressio edellisten viikkojen aikana on ollut luonteeltaan viimeistelevää, eikä mitään erityisen mielenkiintoista ole tapahtunut, lukuunottamatta erään välttämättömän virtakomponentin hajoamista. Uusi on laitettu tilaukseen ja saapuneen tulevalle viikolla ja pääsemme vihdoinkin testaamaan laitetta.

Kaikki ominaisuudet ovat kuitenkin toimineet lopullisella kokoonpanolla. Prototyyppi esitellään ensi viikolla, joten valmistelimme siistimmän kopion kuorista ilman vaahtomuoveja miellyttävämmän ulkomuodon vuoksi. Testaukset tulemme suorittamaan vanhoilla, vaahtomuovitetuilla kuorilla.



19. - 23.8. Viimeistely loppunäytöstä varten

Ryhdyimme toteuttamaan lyhyttä videota pallon esittelyä varten. Kasasimme myös kurssin vaatimia nettisivuja ja dokumentaatioita.

26. - 30.8. Viimeinen viikko

Viimeistelimme tämän prosessipäiväkirjan ja loppuraportin. Loppuraporttiin on koottu kaikki merkittävä informaatio heitettävään kameraan liittyen.

Linkki loppuraporttiin:

https://docs.google.com/document/d/1IcNWsNJifEo75dKJh3PFXGHHLWA6aAWgVvFyxxKcX_Y/edit?usp=sharing