

N1ER Traction Control System

Esineiden internet (Iot) moduuli-harjoitustyö

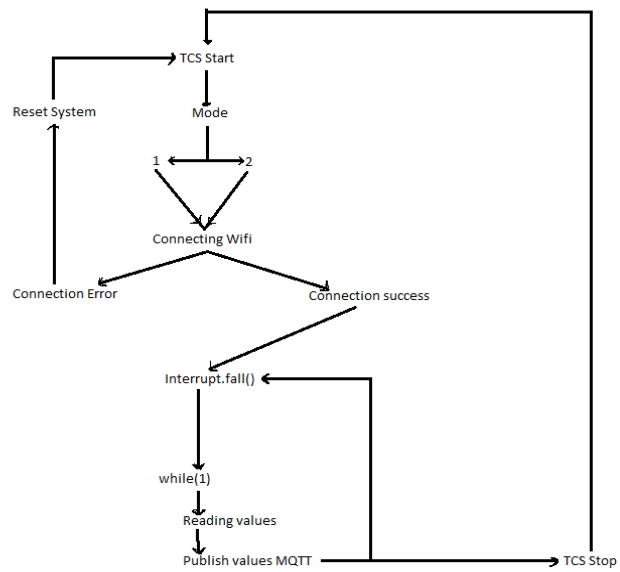
Marko Nopri INTIM19A6

Johdanto

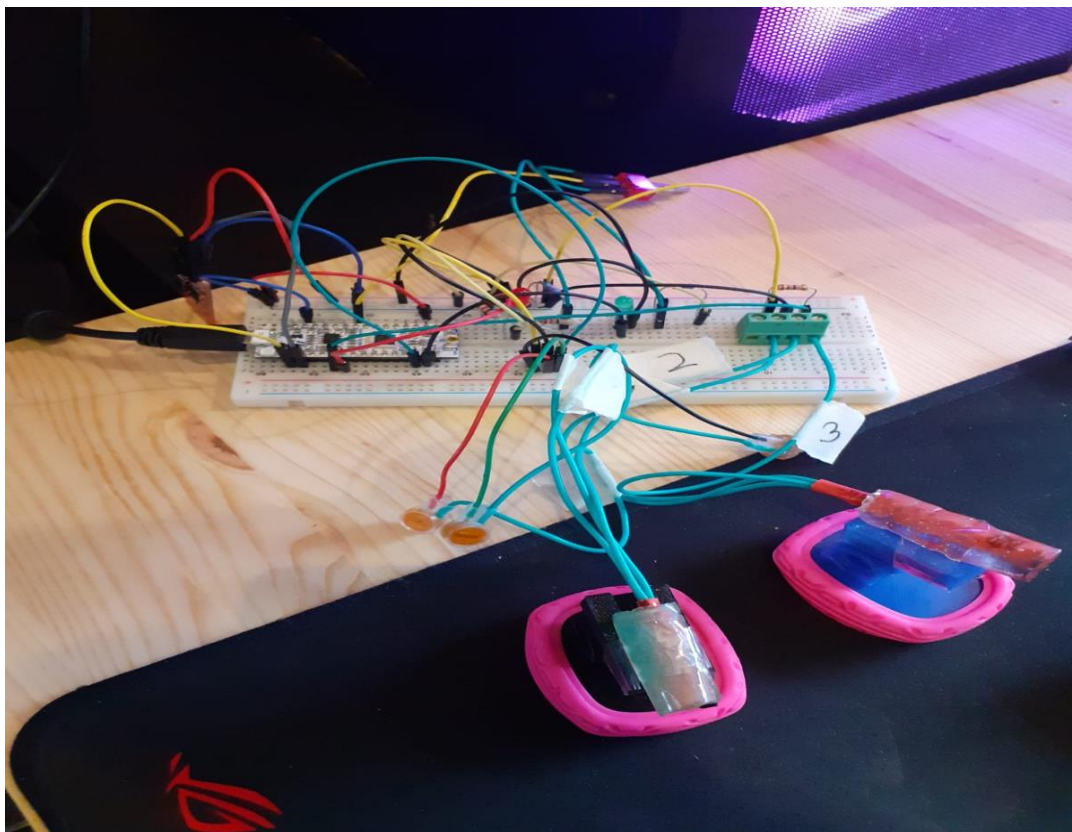
Tämä raportti käsittelee Esineiden Internet (Iot)-moduulin harjoitustyötä. Harjoitustyön aiheena on N1ER kilpa-autoon prototyyppi TCS-järjestelmä (Traction Control System). Raportissa käsitellään tavoitteen, siihen käytetyt ratkaisut, miten lopputulos muuttui tavoitteesta ja analyysin lopputuloksesta ja testauksesta. Työssä on käytetty koulusta saatuja ja omia resursseja. Tavoite järjestelmässä on mikrokontrolleri, kaksi Hall-anturia, kaksi magneettia (yksi per akseli), näyttö, joka indikoi järjestelmän tilan ja mahdollisen toimenpiteen, wifi-moduuli, jolla tiedon saa siirrettyä pilveen ja pilvipalvelu, josta datan voi lukea.

Teoria, laitetoiminta.

Tavoite oli kehittää kilpa-autoon luistonesto järjestelmä, joka mittaa etu- ja taka-akselin vauhtia. Oikeassa tilassa pienentää etu- ja taka-akselin nopeuseroa ja päivittää dataa pilveen, jota voi seurata varikolta. Auton kehitys on kesken, joten järjestelmää ei pääse testaamaan radalla kurssin aikana. Testaus tapahtuu testipenkissä ja lopputulos on ainoastaan teoreettinen. Itse tavoite on saada mikrokontrolleri lukemaan anturiarvoja, algoritmin toiminta ja datan siirto pilvipalveluun.



Kuva 1. Järjestelmän algoritmi



Kuva 1.2 TCS-prototyyppi.

Mikrokontrolleri saa HALL-antureilta pulssitiedon etu- ja taka-akselilta. Molemmissa akseleissa on magneetti, jota anturi lukee. Jokaisen kierroksen aikana mikrokontrolleri saa anturilta pulssin. Tämän pulssin avulla saadaan laskettua mm. RPM, nopeus (m/s, joka muutetaan km/h) ja pulssitaajuus.

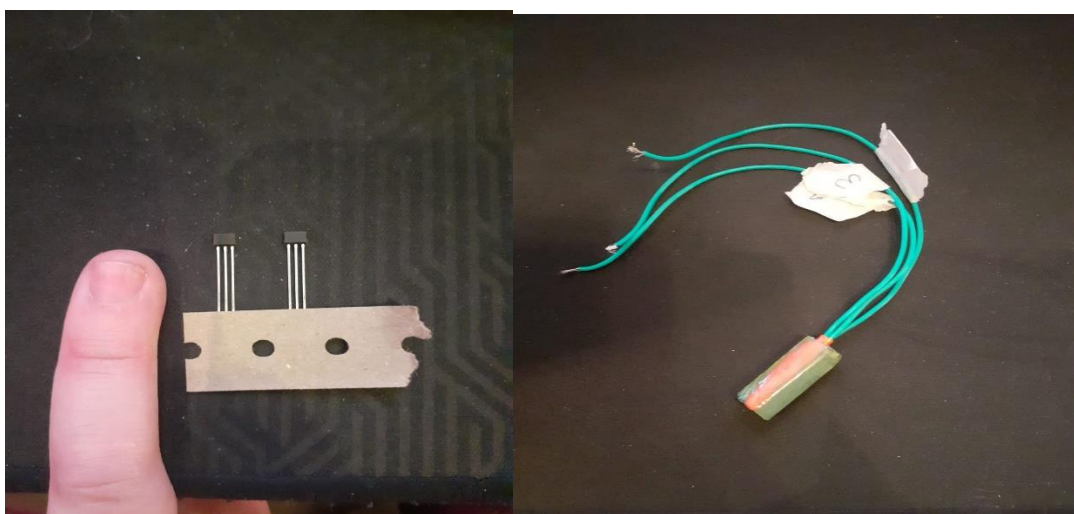
Käyttäjä voi halutessaan sammuttaa järjestelmän napista, käytännössä tämä katkaisee järjestelmältä sähkön. Käyttäjä voi vaihtaa TCS-toimenpiteen vahvuutta. Kaksiasentoinen kytkin kytkee AnalogIN pinniin tietyn jännitteen, järjestelmä tunnistaa jännitteen suuruuden ja muuttaa tämän tiedon avulla luistokertoimen rajaa. Järjestelmä tulostaa sarjaporttiin tiedon, kun slip on ylittänyt rajan ja toimenpide tehdään. Samalla LED syttyy ja vilkahtaa muutaman kerran.

Järjestelmän ohjelmakoodi perustuu keskeytykseen. Joka kerta, kun Hall-anturi vaihtaa tilaa, keskeytys ajaa halutun funktion ja suorittaa tarvittavan matematiikan, CountOne ja CountTwo kasvavat. Kun pulssi saadaan uudelleen, timer arvo luetaan, resetoidaan ja käynnistetään uudelleen. Järjestelmä pyörittää tätä kaavaa jatkuvasti ja lukee arvoja. Järjestelmä saa laskettua molempien renkaiden vauhtia, nämä arvot sijoitetaan kaavaan,

Luistokerroin = (Takanopeus / Etunopeus – 1) * 100.

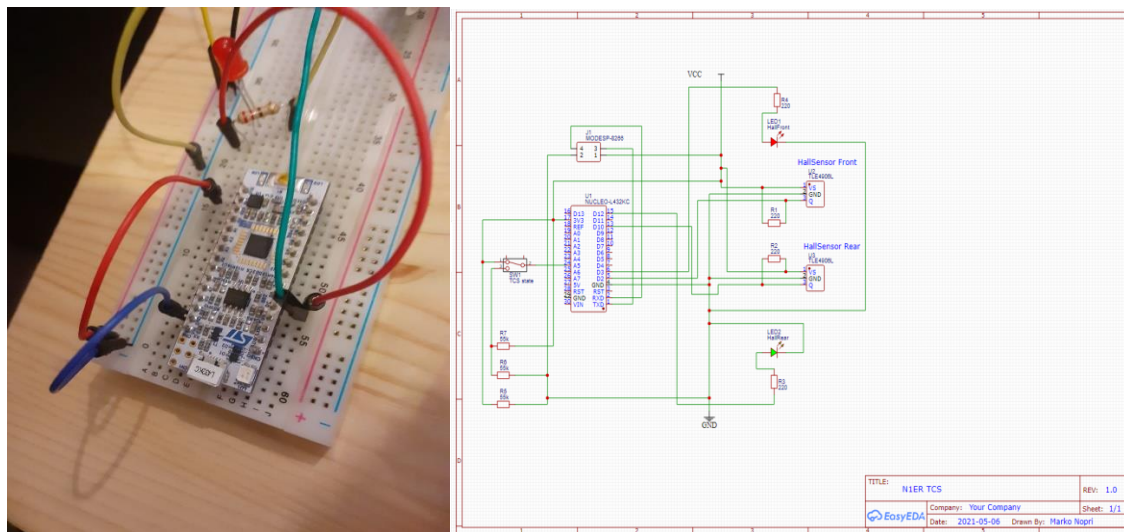
Käynnistyessä järjestelmä käynnistää internet yhteyden, jos yhteyttä ei saada tulostetaan siitä sarjaporttiin virheilmoitus. Muussa tapauksessa edetään seuraavan vaiheeseen. Seuraava vaihe on arvojen lukeminen. Luetut arvot tulostetaan sarjaporttiin. Kun järjestelmä saa onnistuneesti internet yhteyden, se alkaa julkaisemaan arvoja MQTT-protokolla pilvipalveluun. Arvot syötetään JSON-muodossa, jotka siirretään myöhemmin erilaisiin mittareihin ja graafisiin tauluihin.

Järjestelmässä käytetään Infineon TLE4906L-hall kytkimiä. Hall-kytkin on pintaliitos komponentti, jonka päälle on valettu epoksihartsilla lieriön muotoinen suojaava kuori. TLE4906L reagoi magneettiin, sulkemalla tai avaamalla kytkimen.



Kuva 2. Infineon TLE4906L pintaliitos komponentti ja valmis anturi.

Järjestelmän ohjaimena toimii STI Nucleo L432KC-mikrokontrolleri.

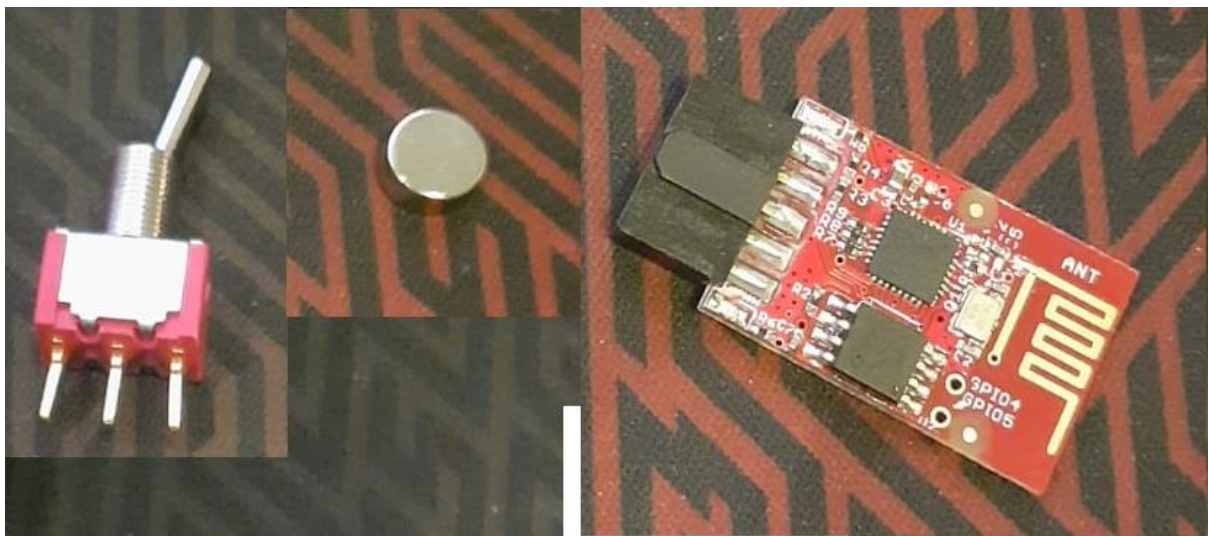


Kuva 3. STI Nucleo L432KC & järjestelmän kytkentäkaavio.

Internetyhteys saadaan käyttämällä Olimex ESP8266 Wifi-moduulia.

TCS-toimenpidettä ohjataan kaksiasentoisella kytkimellä.

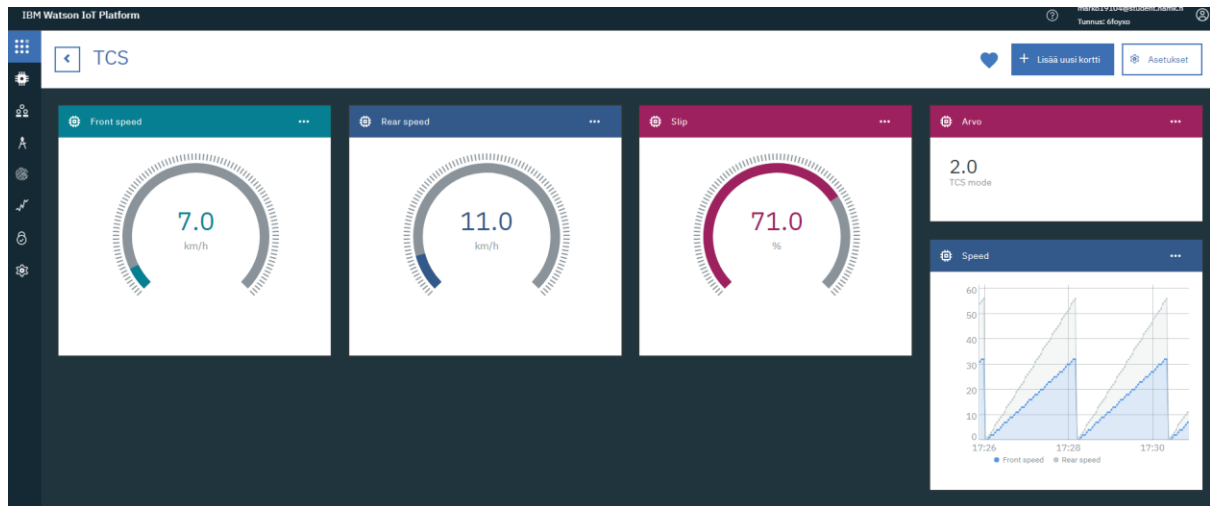
Järjestelmän tilaa luetaan LED:llä. Hall-antureiden tilan, Järjestelmän ON/OFF-tila ja toimenpide indikoidaan LED:llä.



Kuva 4. Toimenpide kytkin, Magneetti akseliin ja ESP8266 wifi-moduuli.

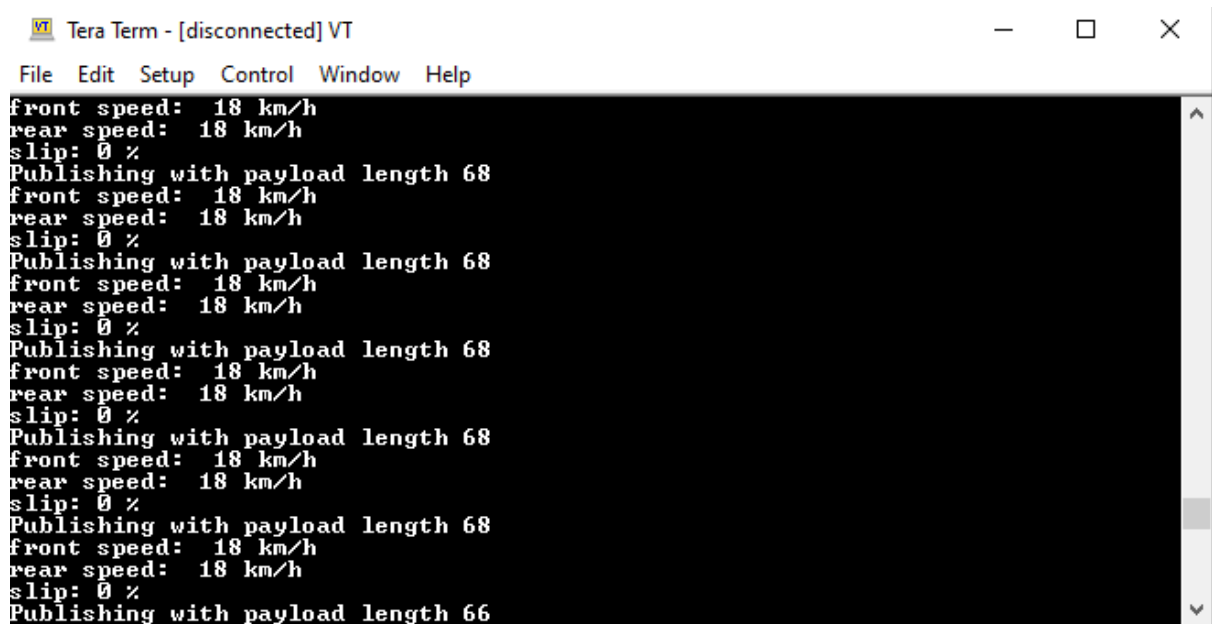
Toiminnan analyysi

Järjestelmän toiminta havaittiin toimivaksi. Algoritmin testaus tapahtui ohjelmallisesti testifunktiolla, joka simuloi renkaiden pyörimisnopeutta. Tällöin luistokerroin pystyttiin laskemaan ja arvot lähettämään pilvipalveluun. Funktio nostaa pyörien nopeutta epätasaisesti, jotta takapyörän nopeus saadaan isommaksi. Testi osoittautui onnistuneeksi.



Kuva 5. IBM Watson IoT Platform, TCS.

Toinen testi tehtiin pienellä radio-ohjattavalla autolla. Radio-ohjattava auto pyörittää oikean- ja vasemman puolen pyöriä omilla ohjauksilla. Joten testissä onnistuttiin, saamaan reaaliaikainen pyörän nopeus. Tämäkin testi osoittautui onnistuneeksi. Järjestelmä ottaa onnistuneesti internet yhteyden, keskeytys ajaa onnistuneesti funktioita, tulostettujen muuttujien arvot pitävät paikkansa ja pilvipalveluun sijoitetut arvot näkyvät mittareissa.

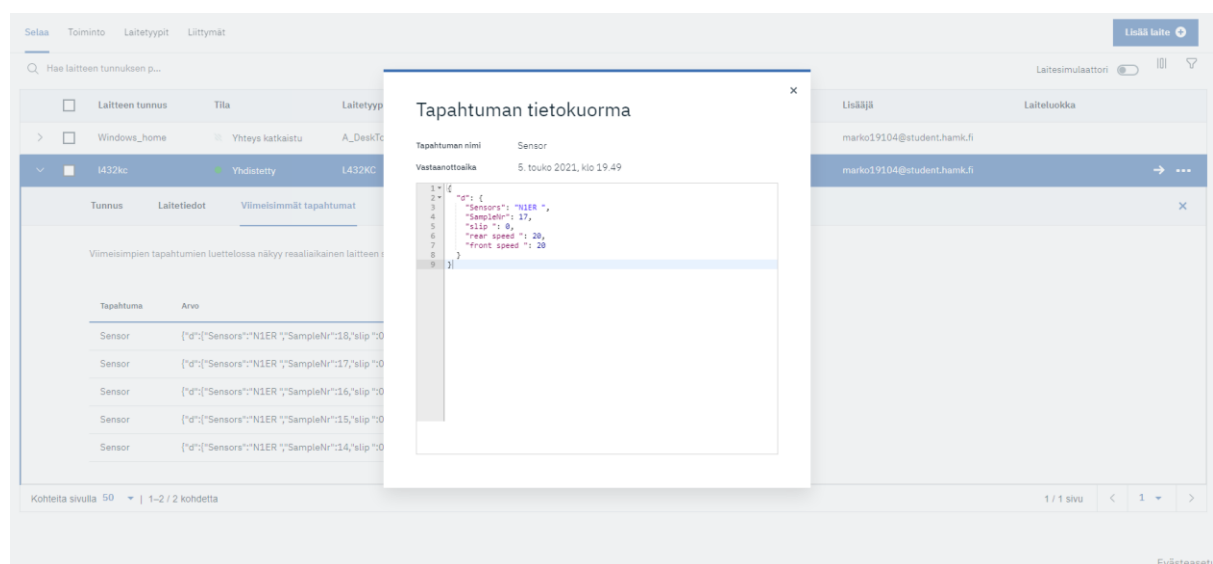


Tera Term - [disconnected] VT

File Edit Setup Control Window Help

```
front speed: 18 km/h
rear speed: 18 km/h
slip: 0 %
Publishing with payload length 68
front speed: 18 km/h
rear speed: 18 km/h
slip: 0 %
Publishing with payload length 68
front speed: 18 km/h
rear speed: 18 km/h
slip: 0 %
Publishing with payload length 68
front speed: 18 km/h
rear speed: 18 km/h
slip: 0 %
Publishing with payload length 68
front speed: 18 km/h
rear speed: 18 km/h
slip: 0 %
Publishing with payload length 66
```

Kuva 6. Sarjaportti dataa, Slip todettu toimivaksi.



Selaa Toiminto Laitetyypit Liittymät

Hae laitteen tunnuksen p...

Laitteen tunnus Tila Laitetyyppi

| Tunnus | Laitetiedot | Viimeisimmät tapahtumat |
|--------------|-------------------|-------------------------|
| Windows_home | Yhteyks katkaistu | A_DeskT... |
| I432kc | Yhdistetty | I432KC |

Viimeisimpien tapahtumien luettelossa näkyy reaaliaikainen laitteen s...

Tapahtuma Arvo

| | |
|--------|---|
| Sensor | ["d":{"Sensors":"N1ER","SampleNr":16,"slip":0 |
| Sensor | ["d":{"Sensors":"N1ER","SampleNr":17,"slip":0 |
| Sensor | ["d":{"Sensors":"N1ER","SampleNr":16,"slip":0 |
| Sensor | ["d":{"Sensors":"N1ER","SampleNr":15,"slip":0 |
| Sensor | ["d":{"Sensors":"N1ER","SampleNr":14,"slip":0 |

Kohteita sivulla 50 | 1-2 / 2 kohdetta

1 / 1 sivu

Evästeaset

Tapahtuman tietokuorma

Tapahtuman nimi Sensor

Vastaanottoaika 5. touko 2021, klo 19:49

```
1 {
2   "d": {
3     "Sensors": "N1ER ",
4     "SampleNr": 17,
5     "slip": 0,
6     "rear speed": 20,
7     "front speed": 20
8   }
9 }
```

Kuva 7. Arvojen siirto pilvipalveluun.

Analyysin perusteella tehdyt havainnot jatkokehityksestä

Järjestelmä on vasta prototyyppi. Kehitystyön ohessa, on löytynyt monta asiaa, jotka parantavat nykyisen version suorituskykyä. Ensimmäisessä versiossa käytettiin NUCLEO-L432KC-mikrokontrolleria. Järjestelmää suunnitellessa, otettiin jo huomioon seuraavan version "raudan" parannus. Version seuraavassa vaiheessa otetaan käyttöön eri ohjain. Käyttöön otetaan NUCLEO-L152RE. Kyseisessä kontrollerissa on enemmän tuloja/lähtöjä, enemmän muistia ja tehokkaampi prosessori. Ohjaimen saa kytkettyä enemmän antureita ja tiedon käsittely on nopeampaa.

OLIMEX MOD-WIFI-8266 korvataan paremmalla modulilla. Tarkoituksena on ottaa käyttöön Adafruit AirLift FeatherWing-ESP32 Wifi Co-Processor moduuli. Kyseisessä moduulissa on paremman kantaman omaava antenni, joka takaa radalla jatkuvan internet yhteyden. Lisäksi moduulissa on erillinen prosessori tiedonkäsittelyä varten. Tämä poistaa järjestelmän ohjaimelta ison kuorman. Lyhyesti sanottuna wifi moduuli hoitaa anturitiedonsiirron omalla prosessorilla.

TCS-järjestelmän datan siirto pilveen, ei ole juurikaan hyödyllistä dataa lukea. Datasta tulee hyödyllistä, kun autoon lisätään NAVIO 2-autopilot moduuli. Tämä kehitys tuo käyttäjälle paikantamisdatan. Tämän ansiosta ajetusta radasta saadaan piirrettyä kartta, ajolinjat ja kellotettua kierrosajat. Kun tämä tieto saadaan ylös, TCS-data on hyödyllisempää. Tässä tapauksessa voidaan pohtia, tarvitseeko TCS vahvuutta keventää tietyissä mutkissa tai voiko jonkun mutkan ajaa kovempaa, että kierrosaika paranee.

TCS-järjestelmää voisi hyödyntää muun datan lukemista. Tässä tapauksessa järjestelmä hoitaisi yleisesti telemetrian. Telemetria lukisi akun kennojen jännitettä, kaasun asentoa, moottorin virtoja ja kaikkea CAN-väylän dataa. Kaikki tämä data olisi hyödyllistä dataa siirtää pilveen.

Suuri kehityskohde olisi pulssitaajuuden parantaminen. Nyt järjestelmä saa yhden pulssin kierroksen aikana. Auto liikkuu 0,75 m yhden pulssin aikana. Järjestelmää saisi tarkemmaksi, jos pulsseja tulisi esim. monta kymmentä kierroksen aikana. Tämä kehitys on helppo toteuttaa. Hall-anturi lukee akselilta magneettia. Akseliin 3d-tulostetaan erillinen muoviosa, johon lisätään 4 magneettia. Tulostettu osa on helppo asentaa, rikkoutuessa se on helppo korvata uudella tulostuksella.

Pilvipalveluna järjestelmässä käytetään IBM Watson palvelua. Palvelu on osoittautunut erittäin epävakaaaksi. Palvelu ei toimi aina halutulla nopeudella. Tämä palvelu korvataan omalla palvelulla. Omaa palvelua pyöritetään Dockerilla. Käyttöön otetaan InfluxDB tietokanta ja Grafana. Grafanaan piirretään datasta luettavaa graafista käyrää. Tällä ratkaisulla saadaan nopeampi palvelu, ilman ylimää räisiä rasitteita.

Antureiden runko on tehty epoksihartsilla. Tätä varten olen tulostanut jo seuraavan anturin muotin, joka valetaan hyväksi todetulla epoksihartsilla. Epoksiharts ei tartu lujasti muoviin, joten muotin saa helposti pois itse anturin rungosta. Tällä saadaan tehtyä anturista sopivan kokoinen ja lisättyä kiristysruuvia varten pieni kolo.

Nykyisessä tilassa järjestelmä antaa sarjaporttiin tiedon ja sytyttää LED:n, kun slip raja ylittyy. Nykyisen auton moottorinohjaimeen ei pääse käsiksi ohjelmallisesti. Tähän ratkaisu on tehdä toimenpide kaasun tietoon. Nykyisessä autossa tämä toutetetaan potentiometrillä, moottorihjain saa jänniteviestin. Ideana on käyttää hyväksi PWM (Pulse width module). Tällöin jännitettä voisi muuttaa ja laskea kaasun asentoa hetkellisesti.