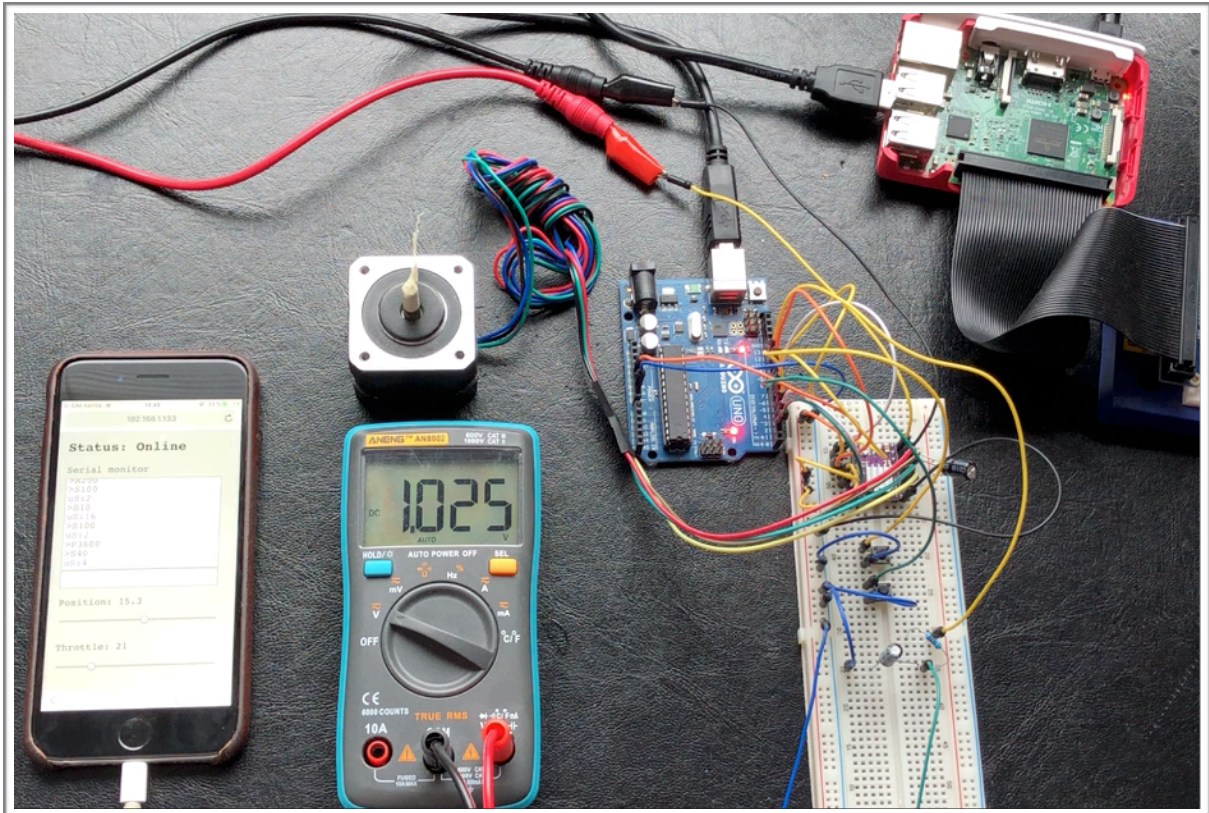


# Sähköperämoottorin etäohjaus



## *Harjoitustyöraportti*

Markus Polvilampi  
<jumapolv@student.jyu.fi>  
TIEA345 IoT-prototyypit  
Kevät 2019

Lyhennelmä	3
Laitteisto	3
Sähköperämoottori	3
Portaaton DC-moottorin nopeussäädin	4
Askelmoottori	4
Askelmoottorin ohjain	4
Minitietokone Raspberry Pi 3	5
Mikrokontrolleri Arduino Uno R3	5
GPS-moduuli + kompassi	6
Muut laitteet	6
KytKentä	6
Ohjelmointi	8
Raspberry Pi	8
Arduino Uno	8
Askelmoottorin ohjaus	9
Jännitelähtö 0..5V	10
Sarjaliikenne	11

## Lyhennelmä

Tavoitteena on automatisoida veneen sähköperämoottorin ohjaus. Lopullisena visiona on järjestelmä, joka ohjaa perämoottorin nopeutta ja kulmaa itsenäisesti GPS paikkatiedon ja digitaalisen kompassin avulla. Tällöin voidaan toteuttaa eri toimintoja kuten: reittipistenavigointi, kurssin pito ja paikan pito (ns. ilma-ankkuri).

Harjoitustyö rajautuu tarvittavien ohjausten tuottamiseen (moottorin nopeus ja ohjauskulma) web-rajapinnan kautta. Moottorin ohjauskulmaa ohjataan askelmoottorilla, joka on tuttua tekniikka lukuisista 3d-tulostimista, ja saatavilla edullisesti.

Perämoottorin nopeutta ohjataan DC-moottorin nopeussäätimellä, jolle annetaan nopeusasetusarvo jänniteviestinä 0-5V.

Laiterajapinnassa on Arduino Uno, joka ohjaa askelmoottoriohjainta sekä perämoottorin nopeussäädintä. Arduino saa asetusarvot ohjauksille Raspberry Pi:ltä, johon se on kytketty USB-kaapelilla ja kommunikoii sarjaliikenteellä.

Raspberry Pi toimii HTTP-serverinä ja mahdollistaa selainpohjaisen responsiivisen ohjauksen hyödyntäen websocket-protokollaa.

Raspberry Pi ohjelmointiin javascriptillä Node.js ympäristössä, mikä yhdenmukaisti koodia käyttöliittymän ohjelmoinnin kanssa.

Arduino Uno ohjelmointiin C++:lla käyttäen Arduino-ohjelmistokehystä.

Lähdekoodit löytyvät git-repostista: [https://github.com/ThatRick/TIEA345\\_harkka](https://github.com/ThatRick/TIEA345_harkka)-kansiosta.

Live-action video toiminnasta: <https://youtu.be/lcH9WNf7BH0> (pahoittelut efekteistä :)

## Laitteisto

### Sähköperämoottori<sup>1</sup>

- Malli: Minn Kota Endura C2 30
- Käyttöjännite 12V
- Maksimivirta 30A
- Tehoalue 96 - 360 W
- Kiinteät tehovalinnat: (eteen/taakse) 5/3
- Työntövoima: 13,6 kg
- 180° käännettävä ohjauspää



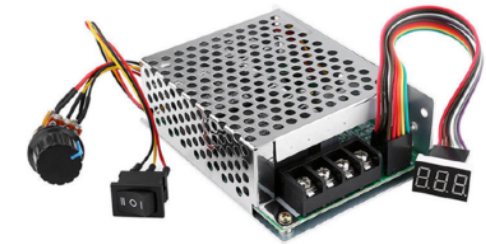
---

<sup>1</sup> [https://www.marinea.fi/minn\\_kota\\_endura\\_30](https://www.marinea.fi/minn_kota_endura_30)

Moottorin alkuperäinen tehonsäätö on toteutettu portaattaisella jännitejaolla, joka heikentää moottorin hyötysuhdetta sitä enemmän, mitä pienemmällä teholla ajetaan. Tämän vuoksi nopeuden säätöön käytetään erillistä jännitemuunninta, jonka hyötysuhde säilyy hyvänä koko nopeusalueella. Lisäksi tehon säätäminen elektronisesti helpottuu.

### Portaaton DC-moottorin nopeussäädin<sup>2</sup>

- Käyttöjännite 10 - 55V
- Maksimivirta 60A
- Suunnanvaihto
- perustuu PWM-tekniikkaan
- asetusarvon näyttö 0 - 100%



Säätimen lähtöjännitettä säätää potikka, jonka jännitealue on 0..5V. Potikan tilalle tuodaan ohjausjännite Arduinolta. Lisäksi tarvitaan kaksi binääriohjausta eteen- ja taaksesuunnalle.

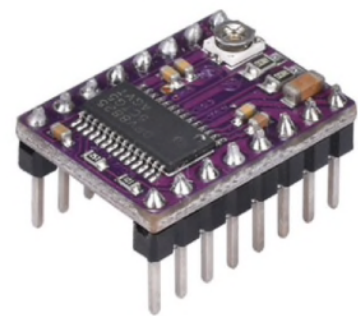
### Askelmoottori<sup>3</sup>

- Tyyppi: Bipolar stepper, Nema 17
- Vääntö 45Ncm
- 200 askelta/kierros



### Askelmoottorin ohjain<sup>4</sup>

- Malli DRV8825
- Resoluutio 1/32 askelta
- Maksimivirta 2.2A
- Käyttöjännite 8.2 – 45 V
- Ohjausjännite: 3.3 - 5V



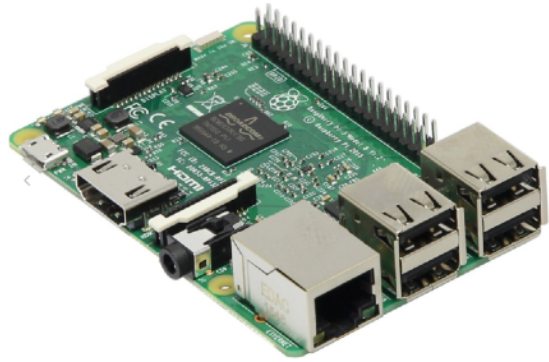
<sup>2</sup> <https://www.aliexpress.com/item/Regulating-Current-DC-Speed-Controller-DC10-55V-12V-24V-36V-60A-PWM-Motor-Speed-Controller-CW/32900621166.html>

<sup>3</sup> <https://www.aliexpress.com/item/4-Lead-Nema-17-Stepper-Motor-Nema-17-Motor-42BYGH-40mm-17HS4401-45Ncm-64oz-in-1m/32855316300.html>

<sup>4</sup> <https://www.aliexpress.com/item/3D-Printer-StepStick-DRV8825-Stepper-Motor-Driver-Carrier-Reprap-4-layer-PCB-RAMPS-replace-A4988/32618856994.html>

### Minitietokone Raspberry Pi 3<sup>5</sup>

- Broadcom BCM2837 64-bit ARMv8 Quad-core 1.2GHz
- Muisti 1Gt
- Wifi ja bluetooth
- Linux



Ensimmäisessä vaiheessa Pi tarjoaa Web-rajapinnan, jonka avulla perämootoria voidaan ohjata etänä. Tässä harjoitustyössä web-rajapinta mahdollistaa perämootorin ohjauskulman ja nopeusohjeen asetuksen. Web käyttöliittymässä on myös pääte, jolla voi tarkkailla ja muuttaa Arduino-ohjelman parametreja.

Tätä on jatkossa tarkoitus laajentaa GPS-ominaisuuksilla.

### Mikrokontrolleri Arduino Uno R3<sup>6</sup>

- Suoritin ATmega328 16MHz
- Käyttöjännite 5V
- Digitaaliset IO:t 14 kpl (joista 6 mahdollista käyttää PWM-ulostuloina)
- Analogiset sisääntulot 6 kpl



Erillinen mikrokontrolleri tarvittiin siitä syystä, että Raspberry Pi toimii 3.3V jännitteellä, ja moottorin nopeusohjaimen logiikkajännite on 5V.

Lisäksi saavutetaan perustoimintojen riippumattomuus Raspberry Pi:stä. Tärkeimmät paikallisohjaukset (eteen/taakse, nopeuspotikka) on tarkoitus kytkeä suoraan Arduinoon, jolloin perämootorin ohjaus onnistuu Raspberry Pi:n häiriötilanteissa.

Lopullisessa toteutuksessa Uno korvautuu todennäköisesti jollain pienemmällä 5V-logiikalla, mikä mahtuu suoraan perämootorin kotelon sisälle nopeusohjaimen kanssa.

---

<sup>5</sup> <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

<sup>6</sup> <https://store.arduino.cc/arduino-uno-rev3>

## GPS-moduuli + kompassi<sup>7</sup>

- Malli NEO-M8N
- Päivitystaajuus 0.1s
- Rajapinta Serial UART

Seuraavassa vaiheessa on tarkoitus kytkeä GPS-moduuli Raspberry Pi:n sarjanporttiin, jolloin saadaan luettua suunta ja paikkatiedot automaattiohjausta varten.



## Muut laitteet

Näiden pääkomponenttien lisäksi on tarkoitus kytkeä Raspberry Pi:hin jännite ja virtamittaukset akulle ja moottorille. Näiden avulla voidaan laskea moottorin kuorma ja hyötysuhde, sekä arvioida jäljellä oleva ajoaika.

Lisäksi moottorin nopeusohjaimen alkuperäiset ohjauskytkimet ja nopeudensäätöpotikka tullaan kytkemään suoraan Arduinon tuloihin, jolloin voidaan ohjata perämoottoria paikallisesti.

Lisäksi täytyy tarkastella mahdollisuutta ohikytkeä moottorin nopeusohjain, jolloin voidaan käyttää moottoria suoraan akuston jännitteellä nopeusohjaimen (mallia Kiina) vikaantuessa.

## KytKentä

Arduino kytkeytyy vaivattomasti Raspberry Pi:hin USB-sarjaportin kautta. Kun Arduinon kytkee Pi:hin, ilmestyy automaattisesti portti nimeltä `/dev/ttyACM0`, johon sarjaliikenneyhteys voidaan muodostaa.

Askelmoottorin ohjain sisältää oman jänniteregulaattorin, joka ottaa jännitteen 12V akulta VMOT-portista. Maa täytyy kytkeä Arduinon maahan, jotta logiikkatasot ovat kohdillaan molemmissa päissä. Tarkemmat signaaliyhteydet selitetty Ohjelmointi-osiossa.

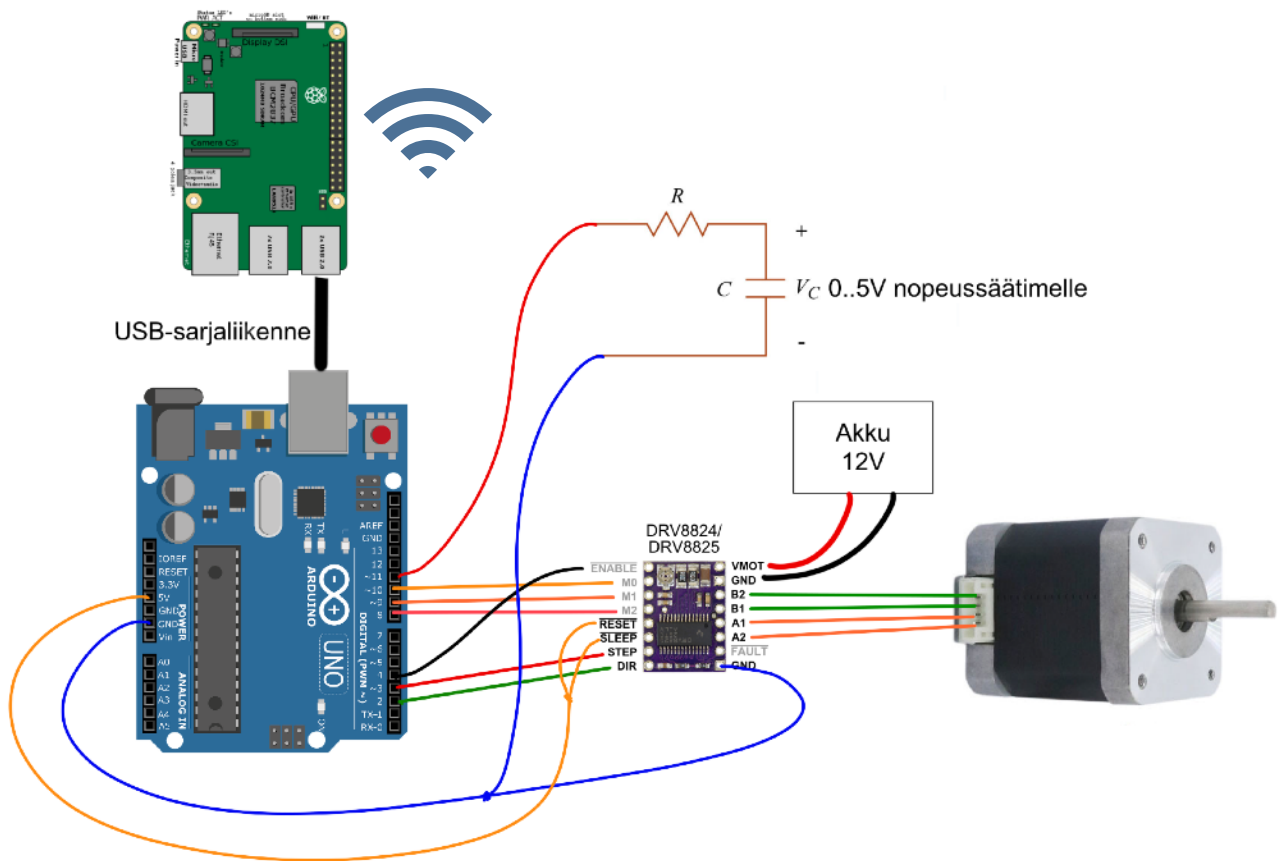
---

<sup>7</sup> <https://www.aliexpress.com/item/1pcs-OCDAE-NEO-M8N-Flight-Controller-GPS-Module-with-On-board-Compass-M8-Engine-PX4-Pixhawk/32852195849.html>



Arduino Uno ei sisällä analogista jännitelähtöä, vaan ainostaan pulssimoduloituja 5V lähtöjä, joten tasainen 0..5V jännitetaso täytyy muodostaa alipäästösuodattimen avulla. Sopivien R ja C arvojen laskentaan käytin nettistä löytynyttä laskuria<sup>8</sup>. Tässä sovelluksessa nopealla vasteajalla ei ole suurta merkitystä, vaan tavoitteena on tasainen jännitetaso. Valituilla arvoilla saadaan rajataajuudeksi 7.23Hz, mikä on huomattavasti alempi kuin Arduinon PWM taajuus 490Hz, joten lähtevä jännitte on riittävän tasainen nopeussäätimelle.

- $C = 2.2\mu\text{F}$
- $R = 10\text{k}\Omega$
- rajataajuus: 7.23Hz
- rippelijännite = 0.0056V
- asettumisaika 0%→90% = 0.05s



<sup>8</sup> <http://sim.okawa-denshi.jp/en/PWMtool.php>

# Ohjelmointi

## Raspberry Pi

Pi toimii ylemmän tason ohjaimena, joka tarjoaa web-käyttöliittymän perämoottorin ohjaukseen. Jatkossa Pi:n osuus kasvaa kun GPS-toimintoja kehitetään, mutta tällä hetkellä se toimii lähinnä välittäjänä.

Web-palvelin on toteutettu Node.js javascript runtime-ympäristössä<sup>9</sup> käyttäen *express*-kirjastoa<sup>10</sup> http-serverin luontiin, sekä *socket.io* -kirjastoa<sup>11</sup> kaksisuuntaiseen, jatkuvaan websocket tiedonsiirtoon palvelimen ja selaimen välillä.

Palvelin jakaa staattisen */public* kansion joka sisältää html-, javascript- ja CSS-tyylitiedostot. Selain suorittaa html-sivulle linkitetyn *main.js* tiedoston, joka luo websocket-yhteyden palvelimeen. Html-sivulla on range input -elementeillä luodut liukusäätimet, joilla voidaan ohjata askelmoottorin kulmaa ja moottorin jänniteohjetta reaaliaikaisesti.

Sivulla on myös teksti-logi, joka näyttää Arduinon sarjaliikenneporttiin lähettämän datan, sekä syöttökenttä, jolla voidaan lähettää komentoja Arduinolle.

Sarjaliikenneyhteyden muodostamiseen Node.js ympäristöstä Aduinoon käytetään *serialport*-kirjastoa<sup>12</sup>.

HTTP-serverin ja sarjaliikenne-yhteyden toteutus onnistui kohtuullisen helposti kirjastojen oman dokumentaation ja esimerkkikoodien avulla ja vei vain 80 riviä javascript koodia.

## Arduino Uno

Uno on ohjelmoitu Arduino<sup>13</sup> ympäristöllä, mikä on C++ -kieleen pohjautuva ohjelmistokehys. Se tarjoaa helpostilähestyttävän rajapinnan mikrokontrollerien ohjelmointiin ja paljon valmiita kirjastoja eri laitteiden ohjaukseen.

Harjoitustyö tehtiin Platform.IO<sup>14</sup> nimisellä IoT-kehitysympäristöllä, joka toimii lisäosana Visual Studio Code<sup>15</sup> -kehitysympäristölle.

Platform.IO tarjoaa tuen monille IoT-laitealustoille, joista Arduino yksi. Se hoitaa kirjastojen hallinnan, koodin kääntämisen ja siirron mikrokontrollerille.

---

<sup>9</sup> <https://nodejs.org/en/>

<sup>10</sup> <https://www.npmjs.com/package/express>

<sup>11</sup> <https://www.npmjs.com/package/socket.io>

<sup>12</sup> <https://www.npmjs.com/package/serialport>

<sup>13</sup> <https://www.arduino.cc/reference/en/>

<sup>14</sup> <https://platformio.org>

<sup>15</sup> <https://code.visualstudio.com>

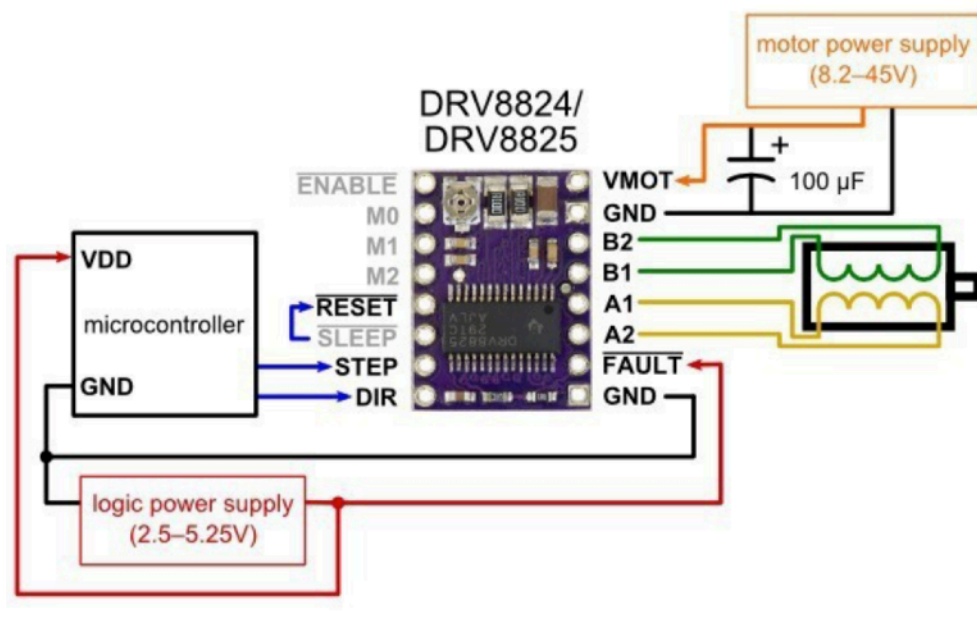


## Askelmoottorin ohjaus

Aaskelmoottoriohjaimen DRV8825 ohjaus vaatii yksinkertaisimmillaan vain kaksi signaalia: suuntavalinta (DIR) ja askel (STEP). Askel-signaali pyörittää askelmoottoria yhden askeleen signaalin nousevalla reunalla. Netistä löytyi hyvä tutoriaali<sup>16</sup>, jossa piirin ohjauksen periaatteet käydään läpi.

ENABLE-signaalilla voidaan laittaa moottorin ohjaus pois päältä, jolloin säästetään virtaa ja moottori on vapaasti pyöritettävissä. Muulloin moottorilla on jatkuva pitomomentti päällä, myös silloin kun sitä ei liikuteta.

Askel-pulssien ajoitukseen käytetään *AccelStepper*-kirjastoa<sup>17</sup>. Kirjasto osaa vakionopeuden lisäksi laskea pulssien ajoituksen annetun kiihtyvyyden mukaan. Ilman kiihtyvyyden rajoitusta



moottori voi alkaa "lipsua". Silloin moottori ei pysy enää ohjauspulssien tahdissa vääntömomentin kasvaessa liian suureksi.

Kirjaston funktiot toimivat asynkronisesti, eli ne eivät blokkaa koodin suoritusta. Kirjastosta luotavalle oliolle annetaan paikka-asetusarvo pulsseina, sekä nopeusasetus. Sen jälkeen kutsutaan toistuvasti pääloopissa run-funktiota, joka antaa pulsseja annettujen asetusten mukaisella taajuudella, kunnes paikka on saavutettu. Ohjelman looppi täytyy pitää tarpeeksi kevyenä, jotta sykli aika ei kasva liian lähelle askelmoottorin ohjaustaajuutta.

Askelmoottoriohjain tukee useita aliaskelemääriä (micro-stepping), joilla askelmoottorin ohjaustarkkuutta voidaan kasvattaa. Askelmoottoriohjaimen tulot M0, M1, M2 määrittävät aliaskelten lukumäärän. Vakiona moottori liikkuu yhdellä STEP-pulssilla kokonaisen

<sup>16</sup> <https://lastminuteengineers.com/drv8825-stepper-motor-driver-arduino-tutorial/>

<sup>17</sup> <https://www.airspayce.com/mikem/arduino/AccelStepper/>

askeleen, joka on tällä askelmoottorilla 1.8 astetta. ( $360^\circ / 200$  askelta kierros). Kasvattamalla aliaskelten lukumäärää, saadaan resoluutiota kasvatettua maksimissaan 0.056 asteeseen ( $360^\circ / 200 \cdot 32$ ). Aliaskelia käytettäessä askelmoottorin toiminta muuttuu hiljaisemmaksi, koska se pyörii tasaisemmin. Korkeilla aliaskelasetuksilla maksimipyörimisnopeus pienenee, koska saman kulmanopeuden saavuttamiseen tarvittava pulssitaajuus kasvaa aliaskelten määrän suhteessa.

Tässä toteutuksessa päästiin luotettavasti noin 4000 pulssiin sekunnissa tarkkuuden kärsimättä. Kokonaisia askelia käytettäessä se vastaa noin 20 kierrosta/s ( $4000 \text{ pls/s} / 200 \text{ pls/kierros}$ ). Suurimmalla tarkkuudella maksimipyörimisnopeus on vain 0.625 kierrosta/s ( $4000 \text{ pls/s} / 200 \cdot 32 \text{ pls/kierros}$ ).

Aliaskelten määrää voidaan kuitenkin dynaamisesti muuttaa vaihtamalla tulojen M0, M1, M2 tilaa ohjelmallisesti, jolloin moottoria voidaan ohjata maksiminopeudella tai -tarkkuudella asetetun pyörimisnopeusasetuksen mukaan.

Käytetyssä kirjastossa ei ollut toimintoja aliaskelten määrittämiseen, joten se toteutettiin omilla funktioilla. Ohjelma laskee funktiossa *setMaxSpeed* suurimman aliaskelmäärän, jolla saavutetaan annettu maksiminopeus, ylittämättä maksimipulssitaajuutta. Funktio *setMicroSteps* asettaa tulojen M0..2 tilat taulukon mukaisesti, annetun aliaskelmäärän mukaan.

Tällä toteutuksella voidaan valita haluttu maksimipyörimisnopeus, ja ohjelma valitsee automaattisesti suurimman mahdollisen aliaskelmäärän, jolla annettu nopeus saavutetaan ylittämättä maksimiohjaustaajuutta.

M0	M1	M2	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	1/4 step
High	High	Low	1/8 step
Low	Low	High	1/16 step
High	Low	High	1/32 step
Low	High	High	1/32 step
High	High	High	1/32 step

## Jännitelähtö 0..5V

Jännitelähtö oli yksinkertainen toteuttaa Arduinon *analogWrite*-funktioilla, joka asettaa annetun lähtöportin PWM-pulssipituuden 8 bitin resoluutiolla (0-255 -> 0-100%). PWM-signaali muutetaan jatkuvaksi jännitetasoksi (0-5V) ulkoisella alipäästösuodattimella (RC-piiri). Tarkemmin RC-piiristä kappaleessa *Kytchentä*.

## Sarjaliikenne

Sarjaliikenteelle luotiin yksinkertainen protokolla, jolla voi asettaa ja pyytää arvoja Arduinossa.

Comm-luokka on rakennettu *HardwareSerial*-objektin ympärille ja se vastaanottaa ja parsii tulevasta sarjaliikenne-datasta parametrin nimen ja arvon. Viesti loppuu aina rivinvaihtoon. Jos parametrin nimi on annettu, mutta arvoa ei, tulkitaan se kyselynä ja parametrin sen hetkinen arvo lähetetään paluuviestinä sarjaporttiin.

Raspberry Pi vastaanottaa Arduinon lähettämät viestit ja välittää ne websocket protokollalla yhteydessä oleville selainohjelmille, jotka tulostavat ne html-sivun logiin.