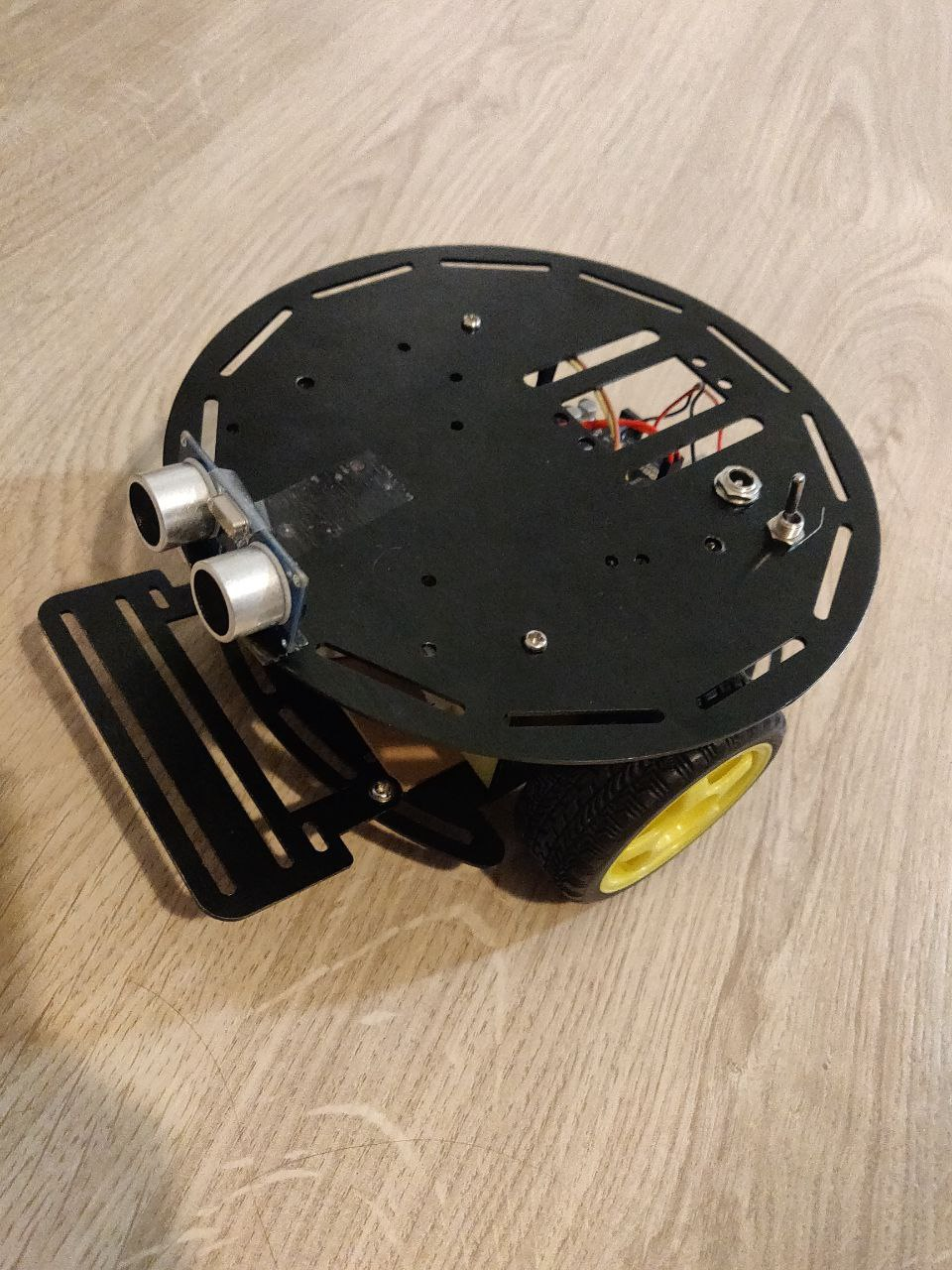
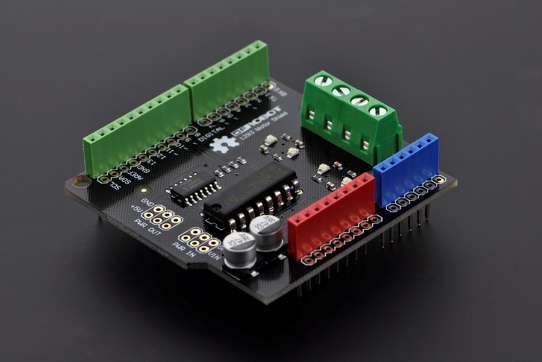
**ARDUINO ROBOT**

**Ultraääniohjattu**

**Toimintaperiaate**

Arduino ohjaa robotin. Se mittaa etäisyyden ultraäänianturilla. Ja kääntyy jos edessä on este. Robotti toimii 9V paristosta.

**Komponentit: Arduino Motor Shield L293 SKU DRI0001**



Moottoriohjain, sisältä kaksikanavainen

L293B ohjain, joka ohja DC moottorit.

Ja 74C00 NAND logiikkapiiri, joka vähentää

moottoriohjauksen Arduinon pinnin määrä

6:sta 4:ään. Piirilevy ohjaa 2 moottoria,

niiden sunnanvaihto ja nopeus PWM:n kautta.

Käyttää liitimet 4 ja 7 suunnanvaihto-ohjauksen,

6 ja 5 nopeuden vaihtoon.

 **Ultrasonic Distance Sensor - HC-SR04**

Mittaa etäisyyden, mittaus alue on 2 – 400 cm.

Toimii 2:lla pinnillä. Trig antaa pulssi ja Echo lue

esteestä heijastuu signaali. Koodissa signaalin

pituus on 10uS. Etäisyys mitataan kaavalla:

(sign. aika pituus – aika nyt) \*0.034/2 = cm

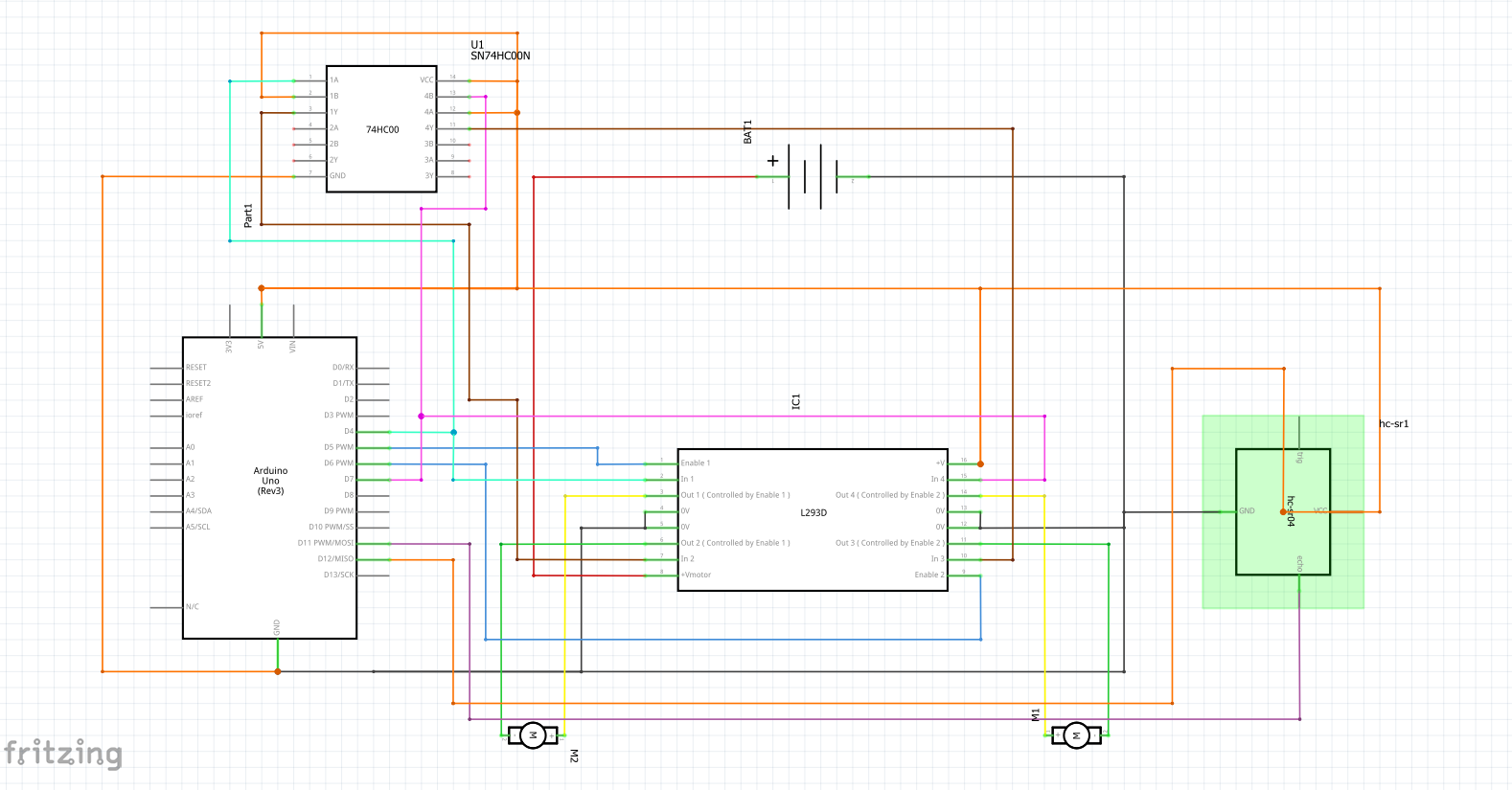
Niin saadaan etäisyys senttimetreinä.

****

**DC-Moottori**

Tasajännite moottorit pyörän pyörimiseen.

**Kytkentäkaavio**

****

**Moottorin ohjaus**

**Arduinon puoli:**

Robotti saa virtaa 9V paristosta. Robotti käyttää 6 Arduinon pinniä.

Arduinon D9 pin ohjaa moottorin 1 PWM, se liitetty L293D Enable1 jalkaan.

Arduinon D6 pin ohjaa moottorin 2 PWM, se liitetty L293D Enable1 jalkaan.

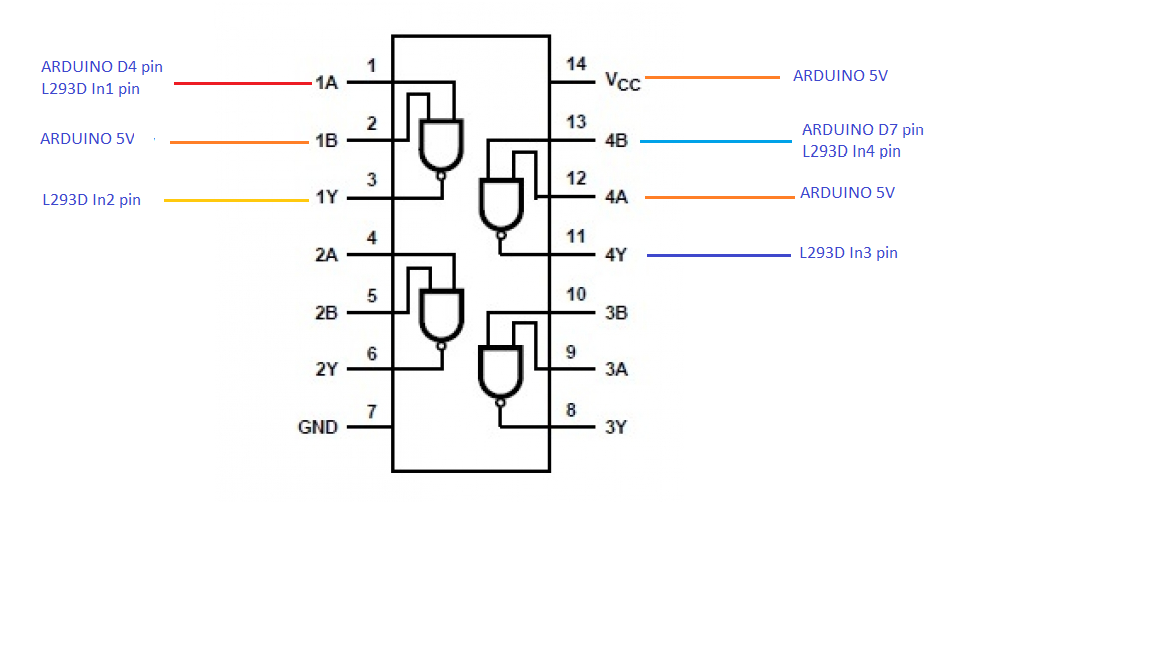
Arduinon D4 pin ohjaa suunnanvaihto ja kytketty moottorin 1 L293D In1 jalkaan ja 74HC00 1A jalkaan.

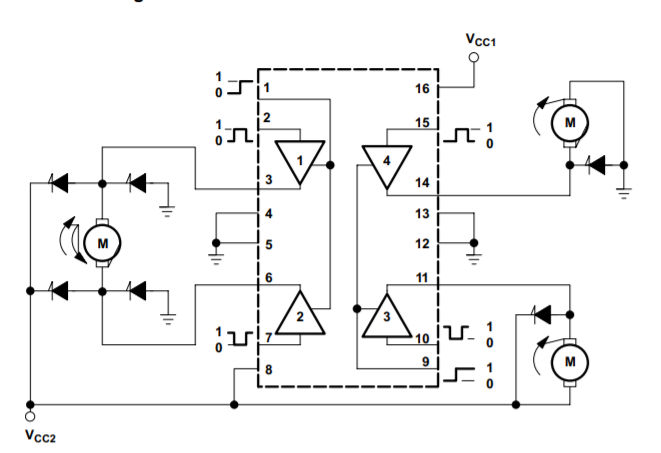
Arduinon D7 pin ohjaa suunnanvaihto ja kytketty moottorin 2 L293D In4 jalkaan ja 74HC00 4A jalkaan.

**74HC00 logiikkapiiri**

74HC00 logiikkapiirin tarkoitus vähentää Arduinosta lähtevä piuhojen määrä.

Signaali tulee vaakittuna pinnille “1B” niin uloslähtö signaali “1Y” on 1 koko ajan. Kun tule signaali “1A” pinnille Arduinosta, niin uloslähtevä signaali muuttuu nollaksi. L293D “In1” pinni tulee jännitteettömäksi. Niin signaali tule L293D “In2” pinnin ja moottorin pyörimissuunta vaihtuu.





**Arduino Motor Shield L293 SKU DRI0001**

Se piiri ohjaa moottorin toiminta.

Vcc1 liitin kytketty Arduinon 5V linjalle.

Vcc2 liitin kytketty patteriin.

Pinnit 3(Out1) ja 6(Out2) kytketty Moottoriin 1 liittimiin.

Pin 1(Enable1) kytketty Arduinon D5 liittimiin.

Pin 2 (In1) kytketty Arduinon D4 liittimiin.

Pinnit 4 ja 5 ovat maadoitusliittimet.

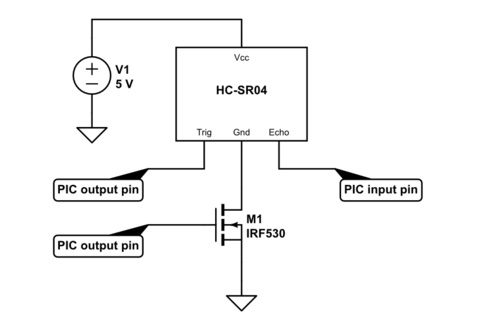
Pin 7(In2) kytketty 74HC00 1Y liittimiin.

Liitin 1 ohjaa moottorin pyörimisnopeuden PWM kautta.

Jos jännite tule 2. liittimiin, niin moottori pyörii kellon suuntaan.

Tai sen vastaan, jos signaali tule liittimeen 7.

**Ultrasonic Distance Sensor - HC-SR04**



Ultraäänietäisyysmittari Vcc kytketty Arduinon 5V linjalle.

Trig kytketty Arduinon D12 pinniin.

Echo Kytketty Arduinon D11 pinniin.

GND kytketty maihin.

Toiminnassa pinniin Trig tule HIGH signaali 10us jatkuvuudella.

Sen jälkeen Trig antaa 8 signaalia 40kHz taajuudella.

Kun heijastuttava signaali saapuu Echo pinnille se muuttuu HIGH tilaan.

Sitten Echo muuttuu LOW tilaan.

Etäisyys mitattaan saapuvan signaalin ajoista kerrattuna ääninopeuden ilmassa (340m/s).

**Koodi**

**Toimintaperiaate**

Ultraäänietäisyysmittari mittaa etäisyyden. Etäisyyden tiedosta ohjataan

moottorin toiminta. Jos edessä on este, robotti kääntyy ja ajaa eteen.

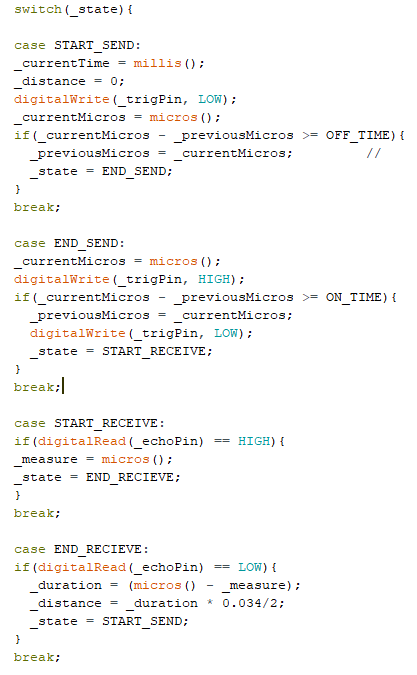
* Robotin ajonopeus laske, kun este lähenee. (Täytyy olla

epälineaarinen nopeus/etäisyys suhde. Ominaisuus estää lian ison

nopeuden ja törmäysvahinko.) Ei tehty.

**Ultraäänietäisyysmittarin ohjelmointi**

Ideana käyttää oma kirjasto etäisyysmittarin ohjaukseen. Perustus on se, että standartti pulseIn() antaa viivettä joka jumitta laiteen toiminta, ja estää moottorin normaali toiminta.

Koodi on jaettu neljään askelin:

1. START\_SEND:

Nollata etäisyyden arvo \_distance,

Pistä Trig pinni LOW-tilaan,

Tallenna nykyisen aikaan \_currentMicros,

Tarkista oliko kulunut OFF\_TIME aikaa (oletuksena 2µs). Jos aika on kulunut, ohjelma siirtyy askelin END\_SEND ja tallenna aikaan \_previousMicros.

1. END\_SEND:

Ohjelma tekee uusi aika leima \_curentMicros,

Trig muuttuu HIGH-tilaan. Tapahtuu vertailu:

Jos nykyaika \_currentMicros miinus edellinen aikaa

\_previousMicros on yhtä samaa, kun ON\_TIME

(10 µs) niin ohjelma päivittää kulunut aikaan \_previousMicros nykyaikaiseksi \_currentMicros.

Ja lopeta Trig-pinnin HIGH signaali. Ohjelma jatkuu seuravan askelin – START\_RECIEVE.

1. START\_RECIEVE:

Ohjelma vertaa tuleeko Echo pinnille HIGH signaali,

jos se tuli. Se tallenna nykyinen aikaan \_measure muuttujaan. Ja siirtää askelin END\_RECIEVE.

1. END\_REVIEVE:

Tapahtuu vertailu, muuttuuko Echo LOW-tilaan. Kun se toteutuu, ohjelma laskee signaalin pituus

mikrosekunnissa ja tallentaa sen \_duration muuttujaan.

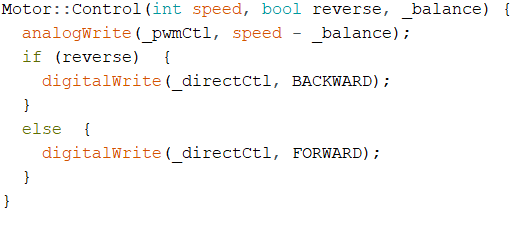
sitten laskee etäisyyden \_distance. Ja paluu START\_SEND

askeliin.



**Moottorin ohjelmointi**

Ohjelma ottaa kolme muuttujia:



int speed -> nopeuden (0-255),

bool reverse -> pyörimissuunta (FORWARD, BACKWARD)

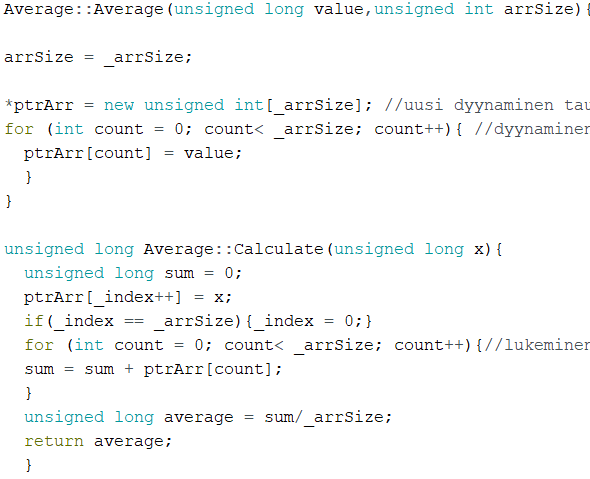
\_balance -> se vähentää nopeuden, jos toinen moottori pyörii lian nopeasti.

analogWrite lähettää parametrit Arduinon pinnille

ja ohjaa toiminnan.



**Keskiarvon laskeminen**

****Laske keskiarvon etäisyysmittarin tulevasta tiedosta

ja tasoittaa suuria arvoeroja.

Average:

Otta muuttujan arvot