



# Sulautetun vaihteiston ohjain- laitteen kehitys

Oskari Revonniemi

OPINNÄYTETYÖ  
Joulukuu 2024

Sähkö- ja automaatiotekniikka,  
Automaatiotekniikka

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Sähkö- ja automaatiotekniikka,  
Automaatiotekniikka

REVONNIEMI, OSKARI:  
Sulautetun vaihteiston ohjainlaitteen kehitys

Opinnäytetyö 49 sivua, joista liitteitä 1 sivua  
Joulukuu 2024

---

Opinnäytetyössä toteutettiin toimiva ja pelkistetty ohjausratkaisu mahdollistamaan Mercedes-Benzin valmistaman 722.6-automaattivaihteiston käyttämiseksi manuaalisesti rata-autoissa. Harrastajien keskuudessa on ilmennyt tarvetta edulliselle, yksinkertaiselle sekä helppokäyttöiselle ohjausratkaisulle. Työn teoriaosuudessa esitellään 722.6-automaattivaihteiston rakenne sekä toiminta, ohjainlaitteen ja näyttömoduulin piirilevyissä käytetyt komponentit, tiedonsiirtoväylät sekä ohjelmointiympäristö.

Opinnäytetyön suunnitteluosuudessa kartoitettiin laitteen suunnittelukriteerit, kuten kustannukset, helppokäyttöisyys, käyttökohde ja käyttöympäristö. Edellisten kriteerien perusteella valittiin komponentit, suunniteltiin piirilevyt sekä luotiin ohjelmat ohjainlaitteeseen ja näyttömoduuliin. Viimeisenä testattiin ohjausratkaisua vaihteiston irrallisen venttiilipöydän kanssa.

Työn lopputuotoksena syntyi valmis ja toimiva mikrokontrollerilla varustettu ohjausratkaisu, joka sopii harrastekäyttäjille. Ratkaisu vastaa loppukäyttäjän tarpeeseen saatavuuden, toimivuuden, helppokäyttöisyyden sekä edullisuuden näkökulmista. Ohjausratkaisun avulla mahdollistetaan Mercedes-Benzin 722.6-automaattivaihteiston käyttö rata-autoilussa.

---

Asiasanat: ohjainlaite, automaattivaihteisto, mikrokontrolleri

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Degree Programme in Electrical and Automation Engineering  
Automation Engineering

REVONNIEMI, OSKARI:

Developing an Embedded Transmission Controller

Bachelor's thesis 49 pages, appendices 1 pages  
December 2024

---

This thesis focuses on the development of a streamlined and practical control system to manually operate the Mercedes-Benz 722.6 automatic transmission, specifically designed for racing applications. The project addresses a growing interest within the automotive enthusiast community for a cost-effective, straightforward, and user-friendly manual control solution. The theoretical portion examines the inner workings of the 722.6 transmission, alongside a detailed exploration of the hardware components, communication interfaces, and software environments utilized in the control unit and display module.

During the design process, key factors such as affordability, usability, intended application, and operating conditions were identified and used to inform the selection of components. Circuit boards were designed, and custom software was developed for both the controller and display module. Testing was conducted using a standalone valve body from the transmission.

The result is a fully operational, microcontroller-based control system designed with hobbyists in mind. It fulfills the requirements of end-users by offering a functional, accessible, and affordable solution, enabling the manual use of the 722.6 transmission in race car settings.

---

Key words: controller, automatic transmission, microcontroller

## SISÄLLYS

1	JOHDANTO .....	7
2	TEORIA .....	8
	2.1 Sulautettu järjestelmä.....	8
	2.2 722.6 automaattivaihteiston toimintaperiaate .....	9
	2.3 Mikrokontrolleri.....	11
	2.4 Arduino-kehitysalusta ja Arduino IDE ohjelmointiympäristö .....	12
	2.5 MOSFET-kanavatransistori .....	12
	2.6 Pulssinleveysmodulaatio .....	15
	2.7 Optoerotin .....	16
	2.8 Tiedonsiirtoväylä .....	17
	2.8.1 CAN-väylä .....	18
	2.8.2 SPI-väylä .....	20
	2.8.3 I <sup>2</sup> C-väylä .....	21
3	VAATIMUSMÄÄRITTELY .....	23
	3.1 Vaatimusmäärittelyn perusteet.....	23
	3.2 Laitteiston vaatimukset.....	24
	3.3 Toiminnallinen kuvaus.....	25
4	LAITTEISTO .....	27
	4.1 Laitteistovalinnan pohjatiedot.....	27
	4.2 ATmega32U4-mikrokontrolleri.....	27
	4.3 Microchip MCP2515 CAN-kontrolleri ja NXP TJA1050 CAN-lähetin- vastaanotin.....	28
	4.4 IRF540N PWM-ulostulokanavat.....	29
	4.5 Optoerotetut-sisääntulokanavat .....	30
	4.6 Näyttömoduuli .....	31
	4.7 Vaihteiston ohjainlaite .....	33
5	OHJELMA.....	35
	5.1 Suunnittelukäytännöt.....	35
	5.2 Ohjelman suunnittelu .....	35
	5.3 Ohjelman toteutus .....	38
6	TESTAUS .....	41
	6.1 Lähtökohdat .....	41
	6.2 Moduulitestaus .....	41
	6.3 Integraatiotestaus.....	42
7	POHDINTA .....	44
	LÄHTEET.....	46

LIITTEET .....	49
Liite 1. Vaihteiston ohjainlaitteen arkkitehtuuri. ....	49

**LYHENTEET JA TERMIT**

A/D	Analog-to-Digital Converter, analogia-digitaalimuunnin
D	Drive, Ajovaihde
IDE	Integrated Development Environment, Ohjelmointiympäristö
LED	Light-emitting diode, valodiodi
N	Neutral, Vapaavaihde
P	Park, Parkkivaihde
PWM	Pulse Width Modulation, Pulssinleveysmodulaatio
R	Reverse, Peruutusvaihde
IDE	Integrated Development Environment, Ohjelmointiympäristö
JST	Japan Solderless Terminal
USB	Universal Serial Bus

## 1 JOHDANTO

Moottoriurheilussa manuaalivaihteistoja suositaan niiden keveyden, kestävyys-  
den, käytettävyyden ja automaattivaihteistoja paremman hyötysuhteen vuoksi.  
Useissa autoissa käytetään tehdasvalmisteisia vaihteistoja tai vähintäänkin teh-  
daskuoria, joihin on asennettu suoraampaiset vaihderattaat sekä suorakytken-  
tärenkaat. Korkeammissa luokissa ajavat harrastajat käyttävät myös täysin moot-  
toriurheiluun suunniteltuja suorakytkentävaihteistoja, jotka on varustettu sekven-  
tiaalivaihtekepillä. Matalamman tason harrastajilla vaihteiston valintaan vaikut-  
taa usein vaihteiston hinta, mikä rajoittaa vaihtoehtoja. Automaattivaihteistojen  
yleistyminen tuotantoautoissa tuo lisähaasteita vaihteistovalintaan, koska kestä-  
viä manuaalivaihteistoja on yhä vähemmän tarjolla kustannustehokkaaseen hin-  
taan. Tämän seurauksena automaattivaihteisto muodostuu monille harrastajille  
potentiaaliseksi vaihtoehdoksi sen hyvä saatavuuden ja kustannustehokkaan  
hinnan vuoksi.

Tämän opinnäytetyön tavoitteena on kehittää moottoriurheilukäyttöön tarkoitettu,  
kustannustehokas ja käyttäjäystävällinen vaihteiston ohjainlaite Mercedes-Benz  
722.6 -automaattivaihteistolle. Tavoitteena on tarjota harrastajille taloudellisesti  
saavutettavissa oleva sekä suorituskyvyltään kilpailukykyinen vaihtoehto. 722.6-  
automaattivaihteisto on sähköisesti ohjattu, jolloin vaihteiston käyttö edellyttää  
sähköistä ohjainlaitetta.

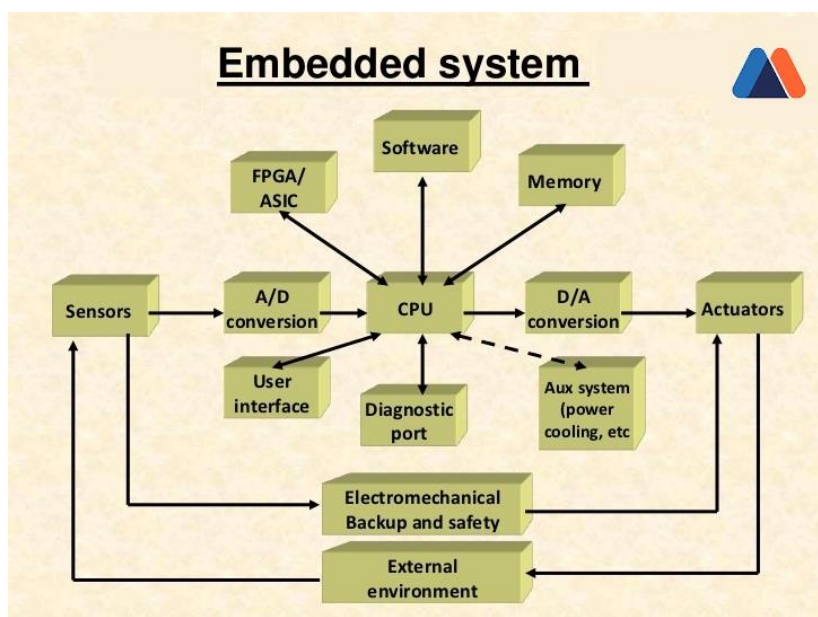
Vaihteiston ohjainlaitteen komponenttien valinta perustuu ensisijaisesti toiminnal-  
listen määrittelyjen kautta saatuihin vaatimuksiin. Nämä vaatimukset kattavat  
käyttäjäkunnan tarpeet, komponenttien kustannukset, helppokäyttöisyyden ja  
saatavuuden. Kun laitteen vaatimukset on määritelty, valitaan projektiin sopivat  
komponentit näiden kriteerien pohjalta. Ottaen huomioon, että ohjainlaitteiden  
suunnittelun tarkoituksena on tuottaa niitä harrastajien käyttöön, on kustannus-  
systä päätetty suunnitella ohjainlaite ja näyttömoduuli itse, sen sijaan että tur-  
vaututtaisiin valmiisiin komponentteihin, joihin tehtäisiin ainoastaan ohjelmistolli-  
sia muutoksia. Kustannustehokkuuden ja laitteiston käyttötarkoituksen vuoksi on  
päädytty kehittämään oma ohjainlaite sekä erillinen näyttömoduuli. Molemmissa  
laitteissa suunnitellaan piirilevy, joka sisältää mikrokontrollerit tarkoittaen, että  
laitteisto täyttää sulautetun järjestelmän määritelmän.

## 2 TEORIA

### 2.1 Sulautettu järjestelmä

Sulautetut järjestelmät ovat tietokoneohjelmiston ja laitteiston yhdistelmiä, jotka ovat suunniteltu suorittamaan sille määrättyä tehtävää. Ne voivat toimia osana suurempaa järjestelmää tai itsenäisesti ja niiden toiminnallisuus voi olla joko ohjelmoitavissa tai kiinteästi määritelty. Nykyään sulautettuja järjestelmiä käytetään laajalti erilaisten laitteiden ohjauksessa, kuten teollisuuskoneissa, kulutuselektronikassa, ajoneuvoissa sekä lääketieteellisissä laitteissa. Sulautettujen järjestelmien markkinoiden odotetaan kasvavan merkittävästi tulevaisuudessa osittain tekoälyn ja virtuaalitodellisuuden investointien kasvun vuoksi. Sulautetut järjestelmät tulevat näin ollen olemaan yhä keskeisemmässä roolissa niin teollisuudessa kuin kuluttajatuotteissakin. (Gillis, A. 2024.)

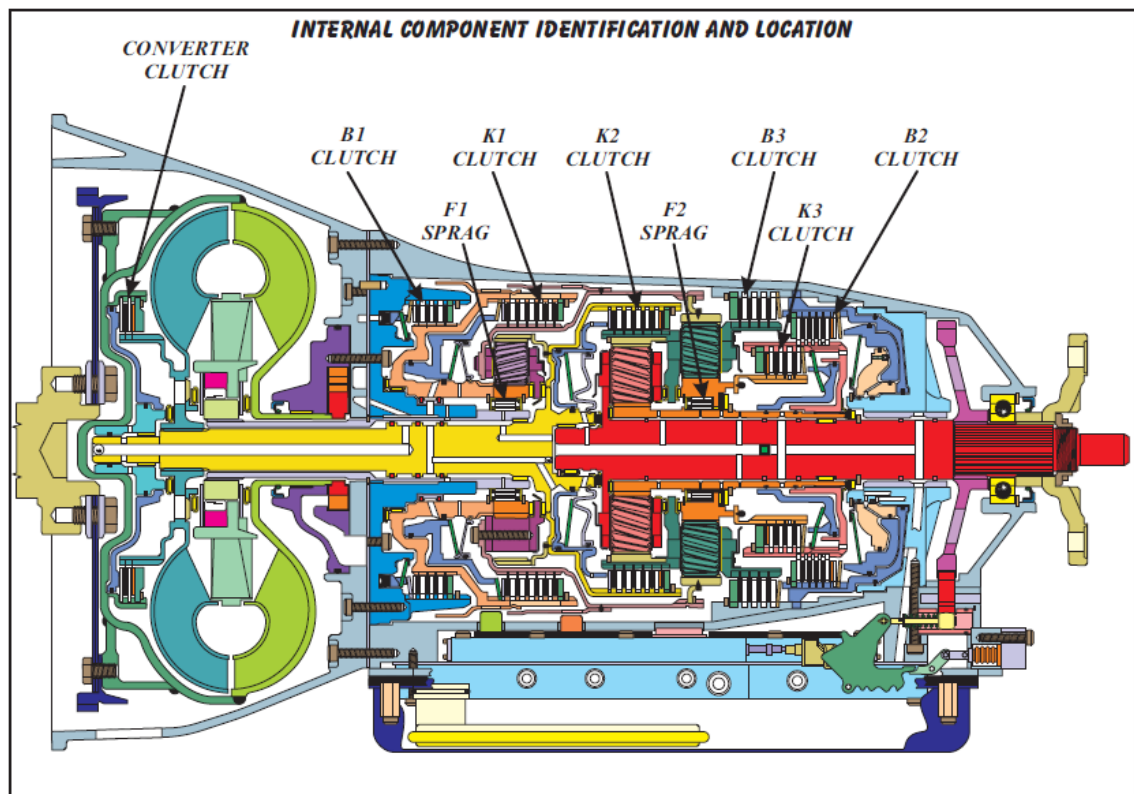
Tyypillisesti sulautettu järjestelmä sisältää mikroprosessorin tai mikrokontrollerin, muistia sekä syöte- ja tulostelaitteita. Kuvassa 1 on esitettyä sulautetun järjestelmän arkkitehtuurikuva. Sulautettujen järjestelmien käyttäjärajapinta voi vaihdella yksinkertaisista painikkeista ja LED-valoista monimutkaisiin graafisiin käyttöliittymiin. Joissakin tapauksissa järjestelmät hyödyntävät myös etäkäyttöliittymiä. (Gillis, A. 2024.)



KUVA 1. Sulautetun järjestelmän arkkitehtuuri (SWATHI, P. 2019.)

## 2.2 722.6 automaattivaihteiston toimintaperiaate

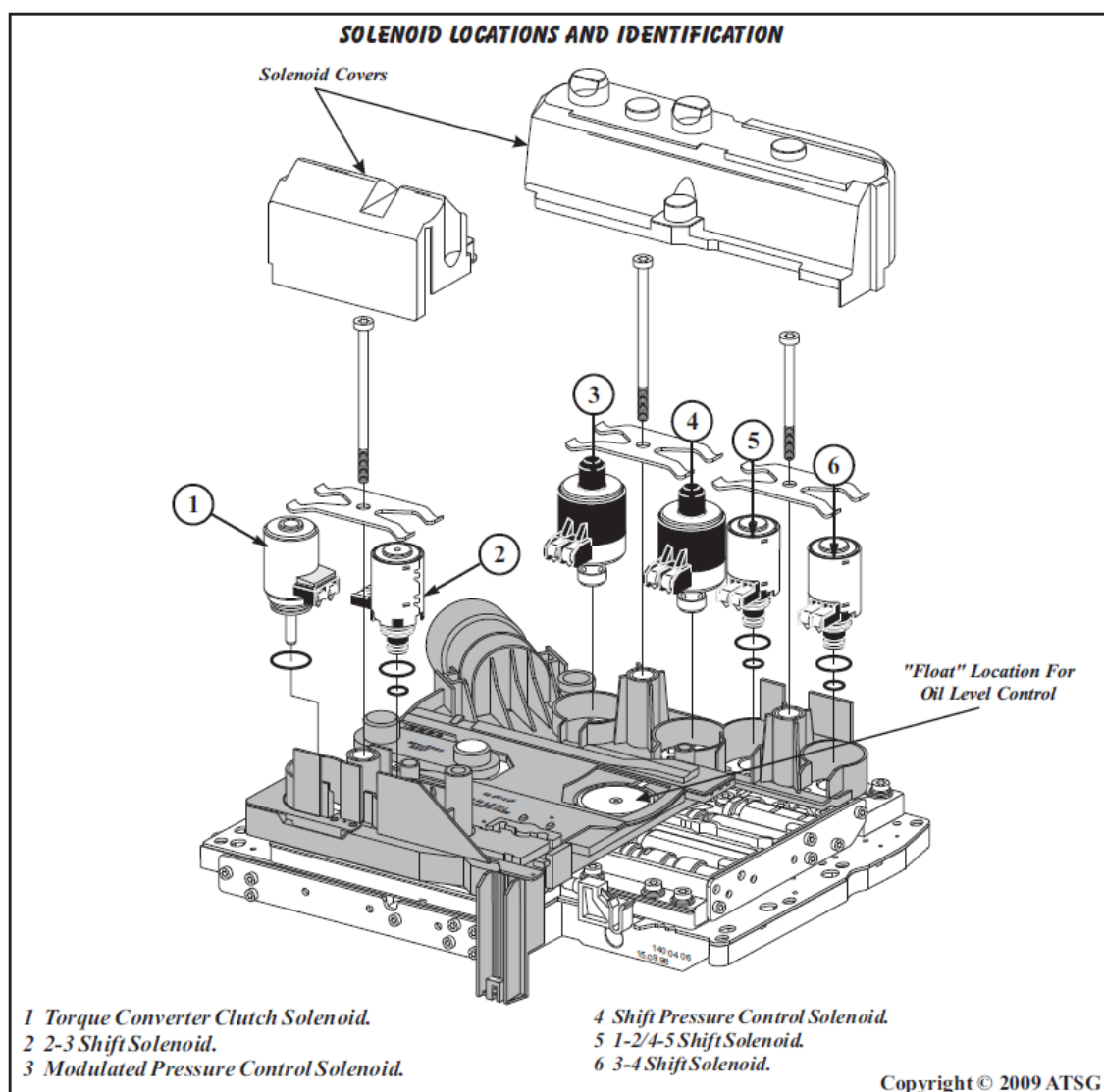
Vuonna 1996 Mercedes-Benz toi markkinoille heidän ensimmäisen täysin sähköjätun automaattivaihteistonsa, 722.6-vaihteiston, joka tunnetaan myös nimellä 5G-Tronic. Vaihteisto on viisivaihteinen, joka koostuu kolmesta yhdistetystä planeettavaihdesarjasta, kolmesta monilevyisestä voimansiirtokytkimestä kolmesta monilevyisestä jarrukytkimestä sekä kahdesta vapaakytkimestä. Kuvassa 2 on esitetty poikkileikkaus vaihteistosta, jossa on esitettynä voimansiirtokytkimet K1, K2 ja K3, jarrukytkimet B1, B2, B3 sekä vapaakytkimet F1 ja F2. (ATSG 2009, 7–8.)



KUVA 2. Poikkileikkaus 722.6-vaihteistosta (ATSG 2009, 8.)

Kuvassa 3 on esitettynä vaihteistossa sijaitsevat kuusi solenoidia, joista kolme solenoidiventtiiliä ohjaavat vaihteistossa olevia voimansiirto- ja jarrukytkimiä vaihteistoöljyn paineella. Neljäntenä vaihteistossa oleva solenoidiventtiili säätelee voimansiirto- ja jarrukytkimelle menevän vaihteistoöljyn painetta ja viidentenä oleva solenoidiventtiili säätelee vaihteen vaihdossa käytettävää vaihteistoöljyn painetta. Vaihteiston kuudennella solenoidiventtiilillä ohjataan turbiinissa sijaitsevaa lukkytkintä. Lisäksi vaihteistossa on mekaaninen venttiilikara, joka liitetään

vivustolla vaihtevalitsimeen. Vaihtevalitsin liikuttaa venttiilikaraa, joka valitsee vaihteiston tilojen: parkki (P), Peruutus (R), vapaa (N) ja ajo (D) välillä. (ATSG 2009, 12–15, 24–25.)

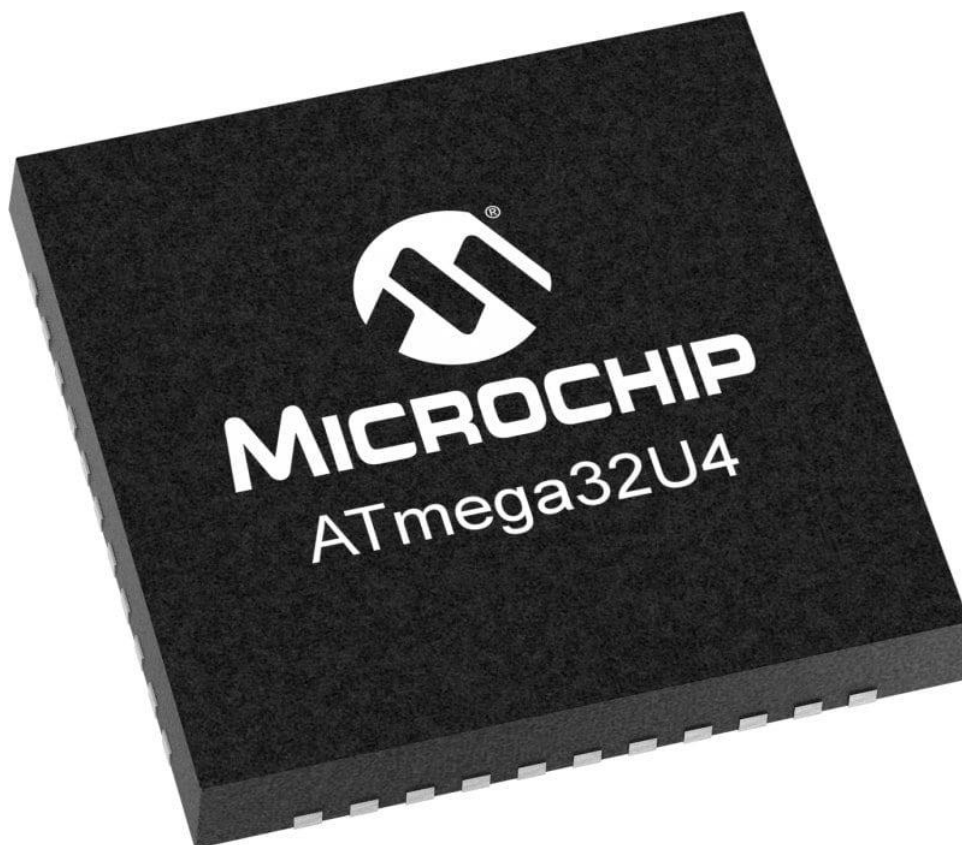


KUVA 3. 722.6-vaihteiston venttiilipöytä (ATSG 2009, 12.)

Vaihteiston voiteluun sekä kytkimien ohjaukseen tarvittava paine tuotetaan öljypumpulla, joka sijaitsee vaihteiston etuosassa. Öljypumppu pyörii aina, kun auton moottori on käynnissä. Ensimmäisenä komponenttina moottorin ja vaihteiston välissä on momentinmuunnin, joka sisältää sähköisesti ohjatun levykytkimen, joka parantaa hyötysuhdetta ja siitä johtuen polttoainetaloudellisuutta. (ATSG 2009, 11).

## 2.3 Mikrokontrolleri

Mikrokontrolleri on laite, jossa yhdistyy mikroprosessori, muisti ja I/O-rajapinta. Mikrokontrollerissa nämä kaikki ovat integroituna yhteen pakettiin, mikä mahdollistaa pienten ohjainlaitteiden tehokkaan valmistuksen. Mikrokontrollereita käytetään esimerkiksi ajoneuvoissa ja kodinkoneissa. Mikrokontrolleri on käytännössä pieni tietokone, joka pystyy suorittamaan samantyyppisiä toimintoja kuin tietokone. Suorituskyky on kuitenkin rajoittuneempaa pienemmän laskentatehon sekä muistinmäärän vuoksi. Mikrokontrolleihin integroidun I/O-rajapinnan ansiosta niihin voidaan liittää useita erilaisia sisääntulolaitteita, kuten kytkimiä ja antureita, sekä niillä voidaan ohjata erilaisia laitteita. Jotta digitaalisesti toimiva mikrokontrolleri voi lukea ja tuottaa analogiasignaaleja, on siihen integroitu A/D-muunnin. A/D-muunnin muuttaa mikrokontrollerin sisääntulokanavaan tulevan analogiasignaalin digitaaliseen muotoon, jolloin mikrokontrolleri pystyy käsittelemään signaalia. (Lutkevich 2019.) Kuvassa 4 esitetään Atmega32U4-mikrokontrolleri, joka on esimerkki tällaisesta laitteesta yhdistäen mikroprosessorin, muistin ja I/O-rajapinnan yhteen yksikköön.



KUVA 4. Atmega32U4-mikrokontrolleri (Atmel. 2015).

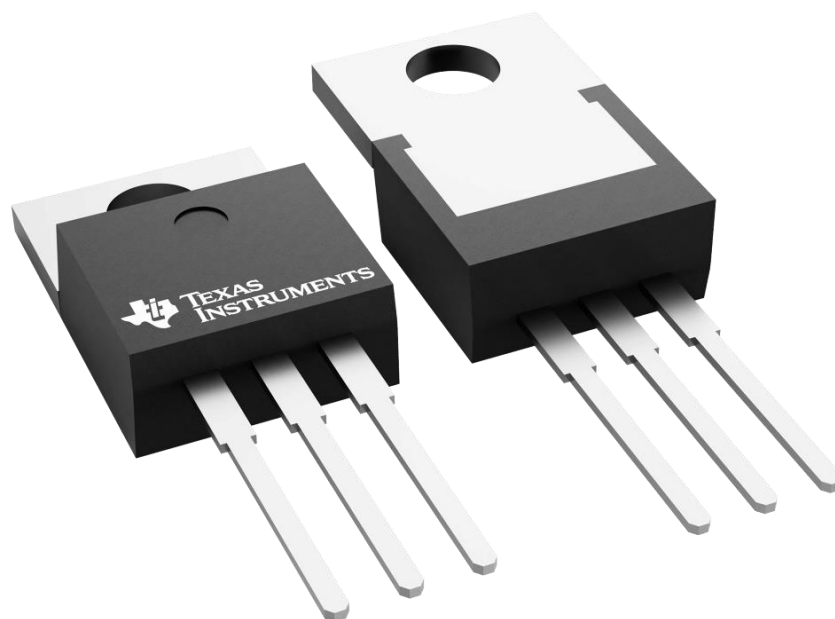
## 2.4 Arduino-kehitysalusta ja Arduino IDE ohjelmointiympäristö

Arduino IDE on Arduino-yrityksen tarjoama avoimen lähdekoodin kehitysympäristö. Arduino erikoistuu kehitysalustojen valmistukseen erityisesti ATmega-mikrokontrollereille. Kehitysalustassa mikrokontrollerin pinnit tuodaan liittimille, jotta niitä voidaan helposti käyttää liitosjohtojen kanssa erilaisten sulautettujen järjestelmien kehittämiseen. Jotta kehitysalustalla voidaan suorittaa ohjelmia, tarvitaan ensiksi kehitysympäristö, jossa ohjelma voidaan kirjoittaa ja kääntää. Tästä johtuen Arduino on kehittänyt oman kehitysympäristönsä. Arduinon kehitysympäristö tarjoaa muutamia kirjastoja jo itsessään, mutta lisäksi kehitysympäristöön on löydettävissä laajalti eri tahojen tekemiä kirjastoja. (Arduino. 2024.)

Arduino tarjoaa monipuolisia työkaluja ja resursseja ohjelmoijille sekä DIY-harrastajille. Arduinon tuoteryhmät sisältävät erilaisia kehitysalustoja, antureita ja moduuleja, jotka mahdollistavat monenlaisten projektien toteuttamisen. Lisäksi yhteisöllisyys on tärkeä osa Arduino-kokemusta: käyttäjät voivat jakaa projektejaan, hankkia ideoita sekä saada tukea muilta harrastajilta. Arduino tekee elektronikan ja ohjelmoinnin oppimisesta saavutettavaa ja hauskaa, minkä vuoksi se on suosittu valinta kouluissa ja harrastajien keskuudessa ympäri maailmaa. (Arduino. 2024.)

## 2.5 MOSFET-kanavatransistori

MOSFET on lyhenne englanninkielisistä sanoista Metal Oxide Semiconductor Field Effect Transistor ja tarkoittaa kanavatransistoria, jossa hilaelektrodi on eristetty kanavasta metallioksidilla (esitetty kuvassa 5). FET-tekniikka on alun perin kehitetty 1930-luvulla, noin 20 vuotta ennen bipolaaritransistorin kehittämistä. Ensimmäiset signaalitason FET-transistorit kehitettiin 1950-luvun loppupuolella, mutta teho-MOSFETit ovat olleet saatavilla vasta 1970-luvun puolivälistä alkaen. (Balogh 2018, 2.)

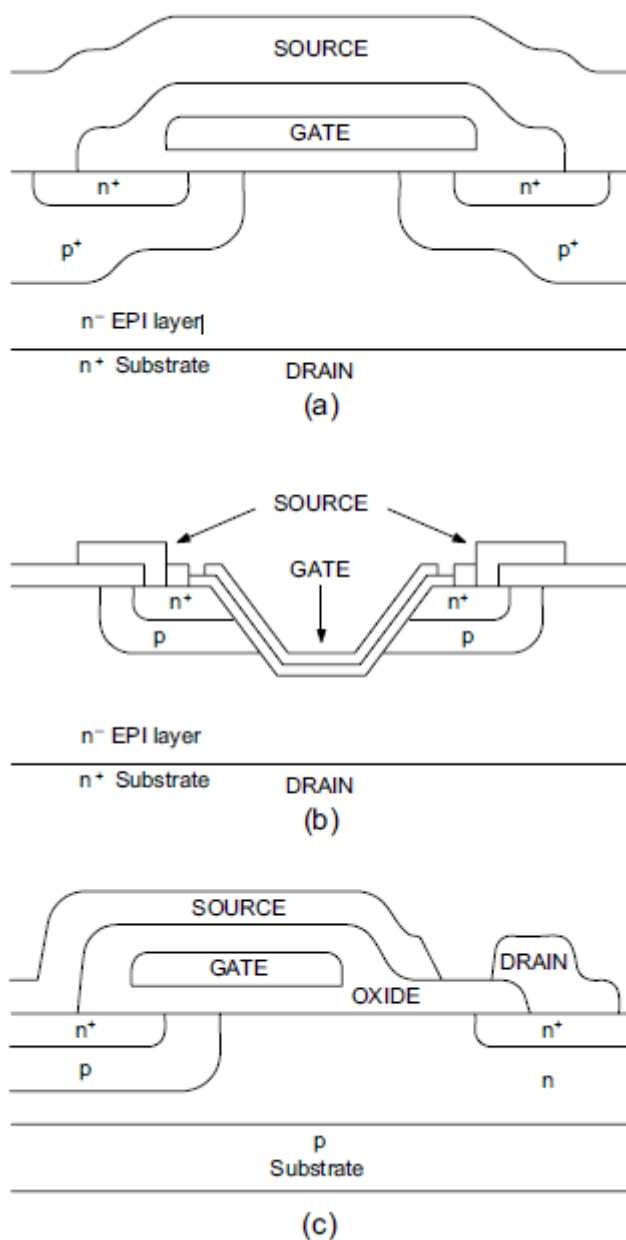


KUVA 5. MOSFET-komponentti (Texas Instruments. nd.).

MOSFET on puolijohdekomponentti, jota käytetään jänniteohjattuna kuormakyt-kimenä. MOSFETin pääkomponentit ovat hila (gate), lähde (source) ja nielu (drain) sekä lähteen ja nielun välissä sijaitseva eristekerros. MOSFETin hilalla on MOS-rakenteesta johtuen kapasitanssi, joka on ladattava, jotta MOSFET saadaan johtavaan tilaan. Hilan lataaminen tapahtuu johtamalla hilaan jännite. Kun hilan varaus ylittää kynnyksvarauksen, MOSFET saadaan johtavaan tilaan. (Storey 2017, 339.)

MOSFETin päällekytkentävaiheessa hilaan johdetaan jännite, jolloin hilan kapasitanssi ladataan. Kun hila on varattu, MOSFET on valmis johtamaan virtaa lähteen ja nielun välillä. Koska MOSFETin toiminta perustuu hilavaraukseen, riippuu tehohäviö hilan latausajasta. MOSFETin ominaisuudet, kuten sen kyky toimia jännitteen ohjaamana kytkimenä, tekevät siitä erittäin arvokkaan komponentin elektronisissa laitteissa. Sitä käytetään laajasti esimerkiksi vahvistimissa, kytkimissä ja tehoelektronikan sovelluksissa. Lisäksi MOSFETin resistanssi kasvaa lämpötilan noustessa, joka mahdollistaa rinnakkaiskäytön. Jos toinen MOSFET vie enemmän virtaa sekä lämpenee enemmän, sen resistanssi kasvaa ja virta pienenee. Näin rinnakkain kytketyt MOSFETit tasaavat virran automaattisesti. Seuraavassa kuvassa 6 on esitettyinä kolme erityyppistä MOSFET-rakennetta: ensimmäisenä kaksoishajutettu rakenne, joka esiteltiin 1970-luvulla sekä seuraa-

vana edistyksellisempi V-ura rakenne, jolla MOSFETin suorituskykyä saatiin parannettua, mutta toisaalta se on monimutkaisempi valmistaa. Viimeisenä lateraalinen rakenne, jolla MOSFETille saavutettiin huomattavasti pienempi kapasitanssi, jolloin kytkentänopeus kasvoi ja hilan ohjaamiseen vaadittiin vähemmän energiaa. (Balogh 2018, 4–5.)



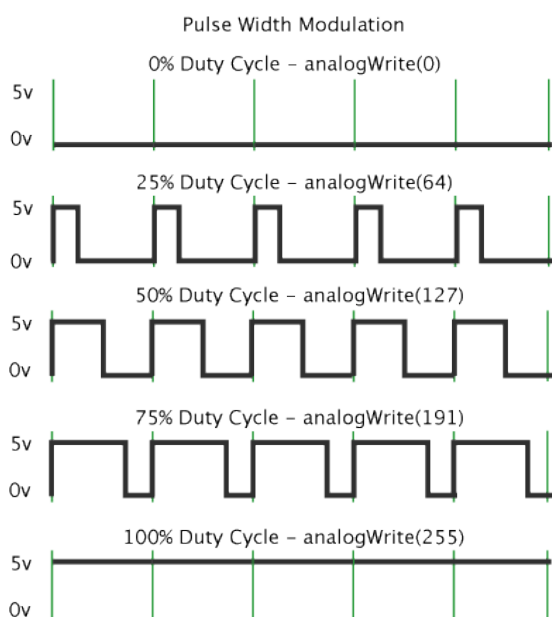
KUVA 6. Kolme erityyppistä MOSFET-rakennetta (Balogh 2018, 4).

## 2.6 Pulssinleveysmodulaatio

Pulssinleveysmodulaatio (PWM) on tehokas tapa ohjata kuormia digitaalisilla teholähteillä. Esimerkiksi ATmega32U4-mikrokontrollerissa ei ole ollenkaan muunninta, joka muuntaisi mikrokontrollerilla olevan sisäisen digitaalisen ohjausarvon analogiseksi ulostulosignaalksi. Mikrokontrollerilta voidaan tuottaa ulostulona PWM-signaalia, jolla voidaan toteuttaa tehonsäätö ulkoiselle laitteelle. (Atmel. 2015.)

PWM-signaalin kaksi keskeistä parametria ovat kytkentätaajuus ja työjakson pituus, jotka ovat esitettyinä kuviossa 1. Kytkentätaajuus riippuu kuormalaitteesta, jota ohjataan PWM-signaalilla. Esimerkiksi lämmitysvastukselle tai moottorille taajuus voi olla melko pieni, kun taas nopeasti reagoiville laitteille kuten releelle tai LEDille, taajuuden on oltava paljon korkeampi. Keskeinen määrittävä tekijä on kuormalaitteen vasteaika. (Dietrich 2022.)

Työjakso on PWM-signaalin suhteellinen kytkentäaika ja se ilmaistaan prosentteina. Jos työjakso on 100 %, signaali on koko ajan päällä. Jos työjakso lasketaan 50 %:iin, signaali on päällä puolet pulssista ja pois toisen puoliskon. Moduloitavan jännitteen ollessa 12 voltia, 50 %:n työjakso tuottaa kuorman näkökulmasta 6 voltin jännitteen, jolloin laitteen teho puolittuu. (Dietrich 2022.)

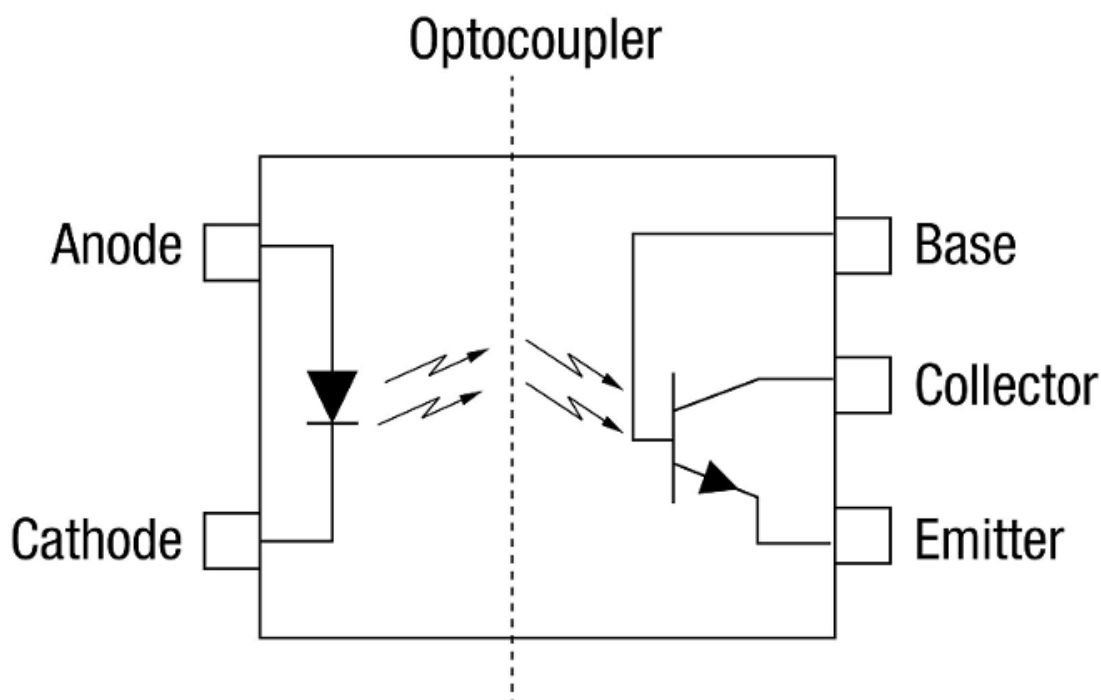


KUVIO 1. PWM-ohjausjakso Arduinin analogWrite-funktiolla (Hirzel 2024).

## 2.7 Optoerotin

Optoerottimet ovat keskeisiä komponentteja elektronisissa järjestelmissä, jotka vaativat galvaanista erotusta sekä suojasta ylijännitteiltä, jotka voivat vaurioittaa laitetta. Tämä erotus estää virran suoran kulun piirien välillä, jolloin ylijännitepiikit tai muut mahdolliset häiriöt eivät pääse siirtymään jännitettä lukevaan laitteeseen. (Texas Instruments 2019, 1.)

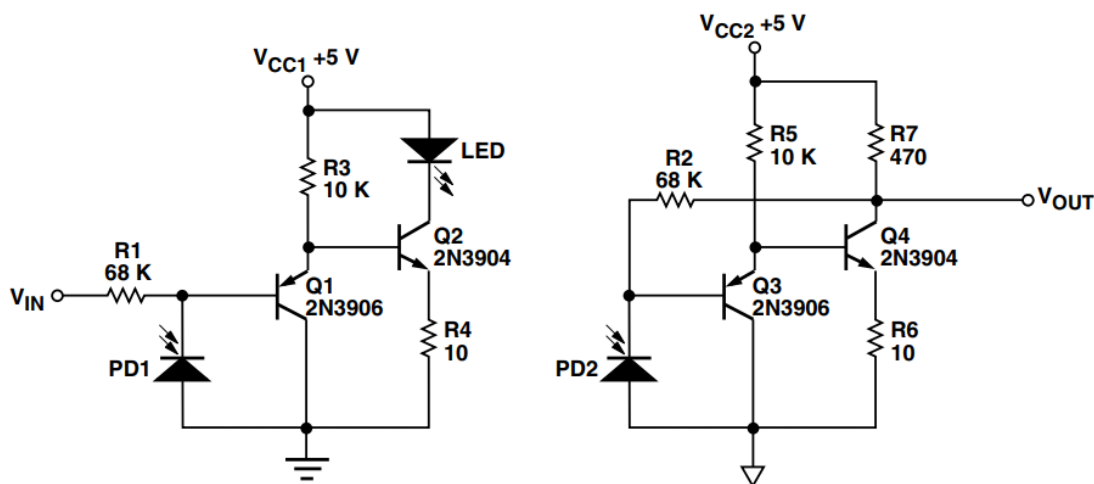
Optoerottimen toiminta perustuu valon avulla tapahtuvaan signaalin siirtoon. Optoerottimessa on syöttöpuolella LED (valodiode), vastaanotto puolella fotodiode ja ulostulopuolella ohjainpiiri. Ohjainpiiri ja LED-piiri rakennetaan tyypillisesti CMOS (Komplementaarinen metallioksidipuolijohde) -teknologialla, ja eristys koostuu yleensä epoksimassasta. Yleisesti optoerottimen sisääntulopuolelle eli LEDille on oma virtalähde ja fotodiodille oma, jolloin signaalille saadaan tuotettua parempi erotus. (Ahonen & Haikio & Salonen. 2022, 77–79.)



KUVA 7. Optoerottimen piirrosmerkki (Texas Instruments 2019).

Optoerottimia on olemassa monentyypisiä ja yllä olevassa kuvassa 7 on esillä yhdellä LEDillä sekä photodiodilla varustettu optoerotin. Alla olevassa kuviossa

2 on esitettyä analogiasignaalien erotukseen tarkoitettu optoerotin, joka on varustettu yhdellä LEDillä ja kahdella photodiodilla. Kun optoerottimessa on kaksi photodiodia, voidaan ohjauspuolelle rakentaa takaisinkytkentä toista photodiodia käyttäen. Tällöin saavutetaan hyvä tarkkuus tulo- ja lähtösignaalien välisissä jännitetasoissa. (Texas Instruments 2019, 18.)



KUVIO 2. analogiasignaalin tarkoitettujen optoerotinkytken piirroskaavio (Texas Instruments 2019, 14).

## 2.8 Tiedonsiirtoväylä

Kommunikaatio on elektroniikan keskeinen ja nopeasti kasvava osa-alue, joka kattaa radiolähetykset, television, matkapuhelimet, internetin sekä ohjaus- ja mittausjärjestelmät. Kommunikaatio tarkoittaa tiedon siirtoa paikasta toiseen ja sitä sovelletaan sekä pitkän että lyhyen kantaman järjestelmissä, joita kutsutaan usein telekommunikaatioksi. Näiden periaatteiden ymmärtäminen on välttämätöntä alalla työskenteleville (Storey 2017, 768.)

Väylä tai tiedonsiirtoväylä on tapa siirtää tietoa sähköisesti laitteiden välillä.

Tiedonsiirtoväyliä käytetään tietokoneissa ja muissa järjestelmissä, joissa täytyy siirtää dataa laitteiden välillä. Väylän fyysinen ratkaisu voi olla toteutettu yhdellä

tai useammalla kaapelilla, riippuen väylän nopeudesta. Väylä on useammin pienemmissä tiedonsiirtomäärissä sarjaväylä ja suuremmissa tiedonsiirtomäärissä rinnakkaisväylä. (Rouse, M. 2023.)

Dataväylät voivat toimia joko rinnakkais- tai sarjaväylinä, riippuen siitä, miten data kuljetetaan. Rinnakkaisväylää käytetään suurempia nopeuksia vaativissa kohteissa, joissa täytyy kuljettaa useampia bittejä kerrallaan. Tyypillisiä esimerkkejä rinnakkaisväylistä ovat PCI Express ja Small Computer System Interface (SCSI) -yhteydet, joissa data kulkee useilla johdoilla samanaikaisesti. Sarjaväylät, kuten USB (Universal Serial Bus) ja SATA (Serial Advanced Technology Attachment), käyttävät yksittäistä johtoa datan lähettämiseen ja vastaanottamiseen komponenttien välillä, ja niissä on vähemmän johtoja. Tämä tekee niistä yksinkertaisempia kuin rinnakkaisväylät. (Rouse, M. 2023.)

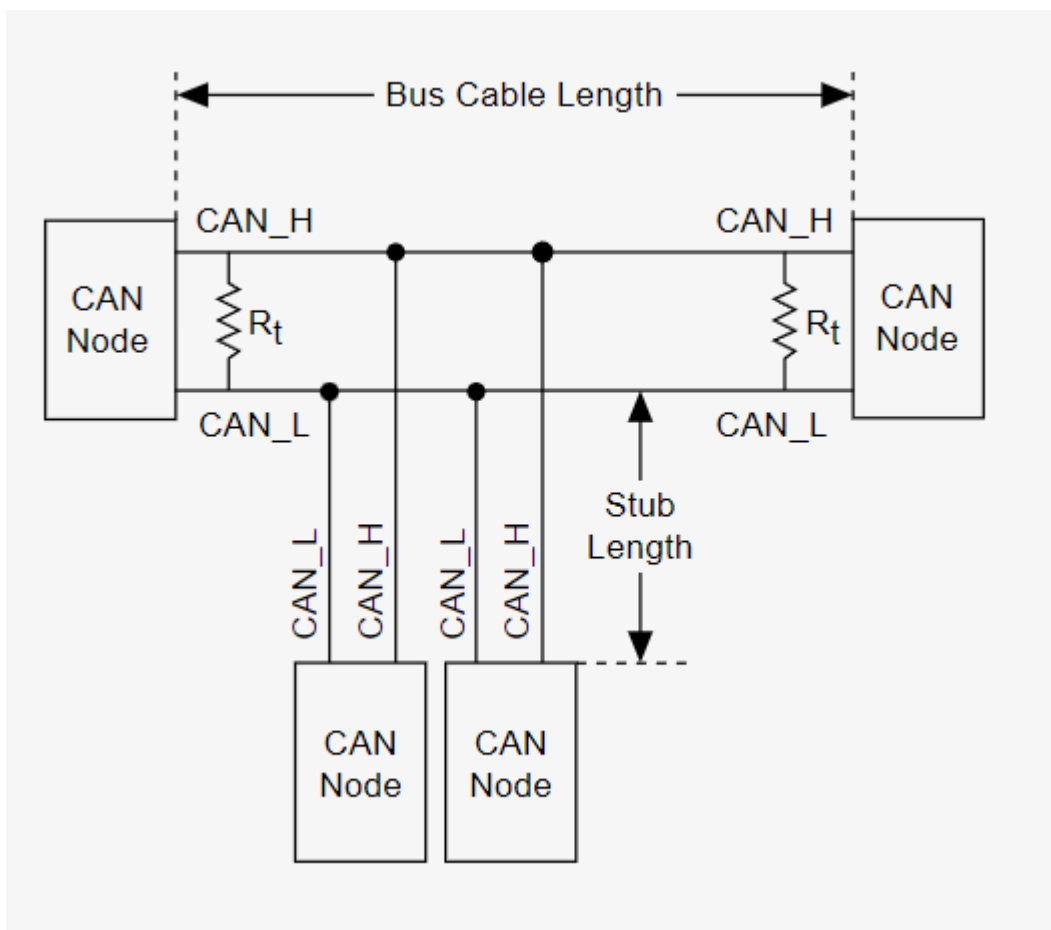
### **2.8.1 CAN-väylä**

Controller Area Network (CAN) -väylä on Robert Bosch GmbH:n vuonna 1986 esittelemä väyläjärjestelmä. CAN-väylän kehitys sai alkunsa 1980-luvun alussa, kun Boschissa työskentelevät insinöörit pyrkivät löytämään ratkaisun ajoneuvojen kasvaviin datansiirtoon liittyviin vaatimuksiin. Aikaisemmat väyläjärjestelmät eivät vastanneet näitä tarpeita, minkä vuoksi uuden väyläjärjestelmän kehittäminen alkoi vuonna 1983. (Zhu 2010, 11.)

CAN-väylän tavoitteena oli ensisijaisesti uusien toimintojen mahdollistaminen ajoneuvoissa, ja johdotuksen vähentäminen oli vain miellyttävä sivutuote. Merkittävä yhteistyö Mercedes-Benzin ja piirivalmistaja Intelin kanssa varhaisessa vaiheessa sekä useiden yliopistojen akateeminen tuki auttoivat muokkaamaan CAN-väylästä erittäin luotettavan ja tehokkaan datanvälitysprotokollan ajoneuvoihin ja myöhemmin myös moniin muihin sovelluksiin. Kuviossa 3 esitetty CAN-väylän yksinkertainen ja tehokas topologia. (CiA nd.)

CAN-väylän standardointi alkoi 1990-luvun alussa ja vuonna 1993 julkaistiin ISO 11898-standardi, joka määritteli CAN-protokollan sekä fyysisen kerroksen bitti-

nopeuksille aina 1 Mbit/s asti. Myöhemmin standardi laajennettiin kattamaan laajennetut kehysmuodot sekä uudet fyysiset kerrokset, mikä vahvisti CAN-väylän asemaa teollisuudessa. (CiA nd.)



KUVIO 3. CAN-väylän topologiakuva (Emerson 2023).

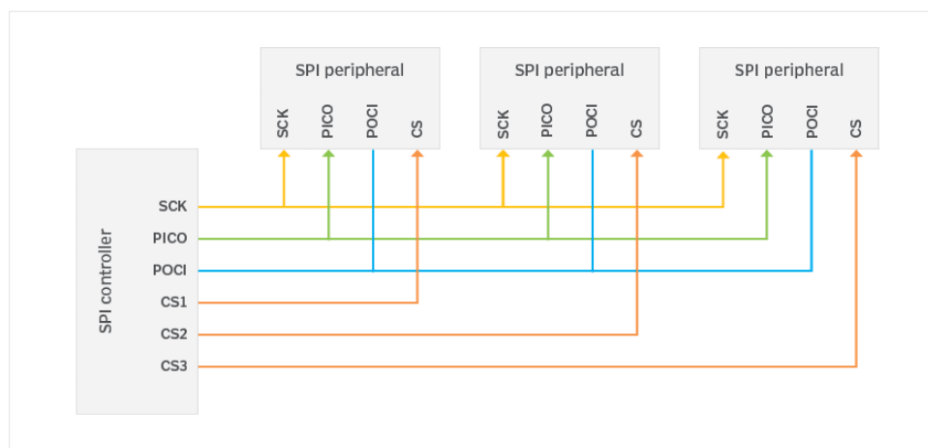
Vaikka CAN-väylä kehitettiin alun perin käytettäväksi henkilöautoissa, se on löytänyt tiensä moniin muihin sovelluksiin, kuten teollisuusautomaatioon, merenkulkuun ja jopa lääketieteellisiin laitteisiin. CAN-väylän menestys perustuu sen kykyyn tarjota luotettavaa ja tehokasta datansiirtoa haastavissakin olosuhteissa. Nykypäivänä lähes jokainen Euroopassa valmistettu uusi henkilöauto sisältää vähintään yhden CAN-väylän, ja sen käyttö on laajentunut kattamaan lukemattomia muita ajoneuvoja ja laitteita ympäri maailmaa. (Zhu 2010, 11.)

## 2.8.2 SPI-väylä

Serial Peripheral Interface (SPI) -väylä (KUVA 8) on tietoliikenneprotokolla, jota käytetään yleisesti mikrokontrollerien ja yhden tai useamman oheispiirin väliseen lyhyen matkan kommunikaatioon tietokoneissa sekä sulautetuissa järjestelmissä. SPI-väylä kehitettiin 1980-luvulla Motorolan toimesta tarjoamaan rajapinta, joka tukee sarjamuotoista, synkronista ja kaksisuuntaista tiedonsiirtoa. Tämä mahdollistaa tiedon siirtämisen molempiin suuntiin samanaikaisesti toisin kuin puolikaksisuuntaisessa kommunikaatiossa, jossa vain yksi solmu voi lähettää tietoa kerrallaan. (Sheldon 2023.)

SPI-väylä koostuu yleensä neljästä johtimesta, jotka kuljettavat erityyppisiä signaaleja ohjausyksikön ja oheispiirien välillä. Alkuperäisessä nimeämisessä käytettiin isäntä-orja-arkkitehtuuria, jossa isäntä ohjasi orjia. Teknologia-alalla on kuitenkin pyritty siirtymään pois tästä terminologiasta edistämään kulttuurisesti herempää kielenkäyttöä. Nykyään puhutaankin enemmän ohjain ja oheislaite termeistä. (Sheldon 2023.)

### Standard SPI



KUVA 8. SPI-väylän topologiakuva (Sheldon 2023).

SPI-väylän toiminta perustuu siihen, että ohjainyksikkö aloittaa kommunikaation ja luo kellostussignaalin, joka synkronoi tiedonsiirron ohjainyksikön ja oheispiirin välillä. Tiedonsiirto tapahtuu bittien muodossa kellostussyklin tahdissa, mikä tarkoittaa sitä, että mitä nopeampi kellostaus, sitä suurempi on tiedonsiirtonopeus.

Koska ohjainyksikkö tuottaa kellotussignaalin, se hallitsee datavirran kulkua. (Sheldon 2023.)

SPI-väylä mahdollistaa tiedonsiirron ohjainyksikön ja monenlaisten oheispiirien, kuten siirtorekisterien, muistipiirien, I/O-piirien, SD-muistikorttien, datamuuntimien, Flash-muistien, multimediakorttien, sensorien ja jopa muiden mikrokontrollereiden välillä. SPI-väylää voidaan käyttää monissa sovelluksissa, joissa tarvitaan nopeaa ja tehokasta tiedonsiirtoa piirien välillä. (Sheldon 2023.)

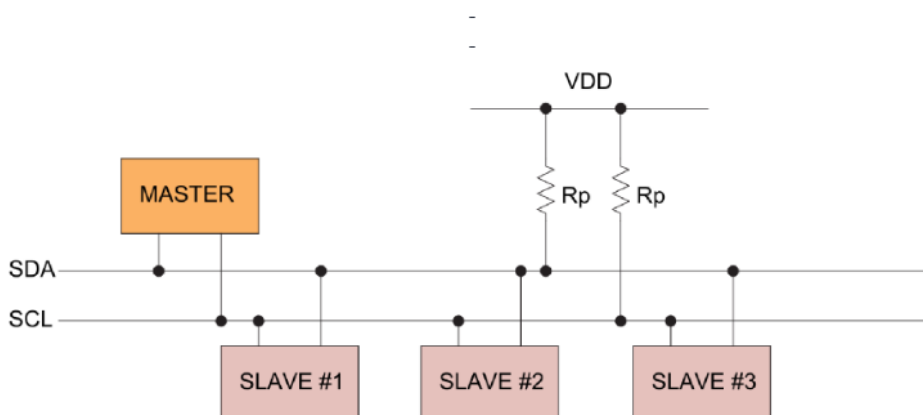
Kommunikointia varten SPI-väylässä voi olla yksi ohjain ja yksi tai useampi oheislaite. Monen oheispiirin konfiguraatioita varten on kaksi peruslähestymistapaa: standardi ja ketjutus. Standardikonfiguraatiossa jokaisella oheispiirillä on oma valintasiignaali, joka mahdollistaa suoran yhteyden ohjainyksikön ja oheispiirin välillä. Ketjutuskonfiguraatiossa data kulkee ohjainyksiköstä oheispiiriin ketjumaisesti, ja vain viimeinen oheispiiri palauttaa datan ohjainyksikölle. SPI-väylän avulla voidaan siis rakentaa joustavia ja tehokkaita järjestelmiä monenlaisiin sovelluksiin elektroniikassa ja sulautetuissa järjestelmissä. (Sheldon 2023.)

### 2.8.3 I<sup>2</sup>C-väylä

Inter-Integrated Circuit (I2C) -väylä (KUVIO 4) on kaksijohtiminen sarjaväylä, jonka kehitti alun perin Philips Corporation kulutuselektroniikan käyttöön. Kyseessä on kaksisuuntainen väylä, joka on helposti toteutettavissa missä tahansa elektroniikkapiirissä. I2C mahdollistaa yksinkertaisen viestinnän integroitujen piirien välillä. Väylän yksinkertaistamiseksi ja yhteyksien minimoimiseksi käytetään sarjadata- ja sarjakellolinjaa sekä yhteistä maatasoa kaiken kommunikaation välittämiseen. I2C on saavuttanut laajan hyväksynnän ja toiminut jopa System Management Bus-protokollan prototyyppinä, joka on I2C:n alajoukko. (Afzal 2016.)

I2C-väylän kommunikaatioprotokolla noudattaa isäntäorjahierarkiaa, jossa isäntälaite kellottaa väylää, kutsuu orjalaitteita ja kirjoittaa tai lukee dataa orjalaitteiden rekistereistä. Orjalaitteet vastaavat ainoastaan silloin, kun niitä isäntälaitteen toimesta pyydetään osoitteidensa kautta. Tämä edellyttää, että orjalaitteiden

osoitteiden välillä ei ole päällekkäisyyksiä. Orjat eivät koskaan aloita datansiirtoa. (Afzal 2016.)



KUVIO 4. I2C-väylän topologiakuva (Afzal 2016).

Datansiirto tapahtuu vaihtamalla linjojen tilaa vetämällä alas ja vapauttamalla ylös. Bittien siirto tapahtuu laskevien kellonreunojen aikana. Vakionopeus datansiirrolle on 100 kbit/s, kun taas nopeassa tilassa siirtonopeus on 400 kbit/s. (Afzal 2016.)

I2C-väylä voi tukea useita laitteita, sekä orja- että isäntälaitteita, ainoana rajoituksena on väylän kapasitanssi (400 pF) ja osoiteavaruus (128 uniikkia osoitetta), kun laitteita lisätään. Datasiiro I2C-väylällä koostuu 8-bittisistä tavuista, jotka sisältävät orjan osoitteen, rekisterinumeron ja siirrettävän datan. Siirto väylän yli on joko luku- tai kirjoitusoperaatio, joka rakentuu useista aliprotokollista kuten alku- ja loppuehdot, toistuvat alku bitit, osoitetavu, datansiirron bitit ja kuittaus bitit. (Afzal 2016.)

Kommunikointi I2C-väylällä mahdollistaa monimutkaisia datasiirrot, kuten kirjoituksen ja lukemisen saman transaktion aikana tai jopa osoitteen vaihdon eri orjalaitteille yhden I2C-transaktion sisällä. Tämä monipuolisuus tekee I2C:stä tehokkaan väylän monenlaisissa sovelluksissa, kuten antureiden, muistipiirien ja muiden oheispiirien hallinnassa kulutuselektronikassa sekä teollisissa sovelluksissa. (Afzal 2016.)

## 3 VAATIMUSMÄÄRITTELY

### 3.1 Vaatimusmäärittelyn perusteet

Vaatimusmäärittely on keskeinen osa jokaista projektia, koska sillä varmistetaan, että lopputuote vastaa alkuperäistä visiota mahdollisimman tarkasti. Mitä tarkemmin vaatimusmäärittely tehdään, sen realistisempi aikataulu ja budjetti projektille saadaan. Vaatimusmäärittelyä tehdessä olisi hyvä tarkastella laitteiston suorituskykyä, hintaa, huoltoystävällisyyttä, turvallisuutta sekä kehityskustannuksia (Jokinen 1991. 30–31.)

Yksi vaatimusmäärittelyn haasteista on ymmärtää miten "vaatimukset" määritellään ja tulkitaan. Vaatimukset voivat esiintyä monessa eri muodossa: yleisluontoisista ja abstrakteista kuvauksista nopeisiin luonnoksiin tai jopa tarkasti määriteltyihin, matemaattisen yksityiskohtaisiin spesifikaatioihin. Tämä monimuotoisuus johtuu siitä, että eri sidosryhmillä on erilaisia tarpeita ja näkökulmia, jotka vaativat vaihtelevaa tarkkuutta ja esitystapaa. Lisäksi sidosryhmien kyky ymmärtää ja ilmaista näitä vaatimuksia vaihtelee; esimerkiksi liiketoiminnan edustaja ja suunnitteluinsinööri voivat tarkastella vaatimuksia täysin eri tavoin. Nämä erot johtavat usein merkittäviin vaihteluihin vaatimusten laadussa. Seuraavassa luvussa tarkastelemme tarkemmin sidosryhmien tarpeita ja niiden vaikutusta vaatimusten esittämiseen (Laplante & Kassab 2022.)

Koska kyseessä on pienin kehitysresurssein tehty laitteisto ja tuotantomäärät tulevat todennäköisesti olemaan kohtuullisen pieniä, määrittelyssä painotetaan kehityskustannuksia enemmän kuin valmistuskustannuksia. Kyseisen ohjainlaitteen vaatimusmäärittely aloitetaan tarkastelemalla seuraavia asioita: käyttökohde, käytettävyys, kustannukset ja käyttöympäristö. Näiden vaatimusten perusteella saadaan hyvä käsitys siitä minkälainen vaihteiston ohjainlaitteen tulisi olla ja minkälaisia ratkaisuja siinä tulisi tehdä.

### 3.2 Laitteiston vaatimukset

Käyttökohteen näkökulmaa tarkasteltaessa 722.6-vaihteisto vaatii toimiakseen vähintään vaihteenvalitsimen, joka liikuttaa vaihteistossa olevaa mekaanista karraa, sekä lisäksi kolmekanavaisen ohjauksen, joka ohjaa vaihteiston venttiilipöydässä olevia vaihtosolenoidia. Vaihteistossa on näiden lisäksi kaksi solenoidia joista toisella ohjataan vaihteiston pakoille menevää linjapainetta, ja toisella vaihteiden vaihtoihin käytettävää painetta. Mikäli kahta edellistä solenoidia ei ohjata, ovat vaihteiston linja- sekä vaihtopaine maksimissaan, jolloin rata-autossa näitä solenoideja ei ole välttämätöntä ohjata. Vaihteistossa on myös solenoidi turbini- lukkoa varten, jota on hyvä ohjata, jotta vaihteiston tuntuma saadaan jämsämmäksi. Vaihteistossa on myös kaksi pyörintänopeusanturia sekä öljynlämpötila-anturi. Vaihteiston toiminnan perusteella saadaan ohjainlaitteelle vaatimukseksi kuusi ulostulokanavaa sekä kolme sisääntulokanavaa Koska nopeusanturit ovat hall-tyyppisiä, niitä voidaan lukea normaalilla digitaalitulolla. Öljynlämpötila-anturi on resistiivinen, joten ohjaimen vaaditaan vähintään yksi analogiamittaus. Jos vaihteiston kanssa käytetään vakio vaihteenvalitsinta mekaanisen karan liikuttamiseen, on ohjainlaitteessa oltava myös CAN-väylä, jotta vaihteenvalitsimen asentotieto saadaan luettua ohjaimelle.

Käyttöympäristön kannalta on tarpeellista varmistaa, että komponenttien lämmönsietokyky sekä häiriön sietokyky ovat riittävällä tasolla. Lisäksi ohjainlaite olisi syytä koteloida, jotta se saadaan suojattua ympäristövaikutuksilta. Lämmönsietokyvyn kannalta varmistetaan, että komponenttien käyttölämpötila skaala on riittävän laaja. Näyttömoduuliin valittiin käytettäväksi OLED-näyttö sen laajan käyttölämpötilan takia. Näytön ja ohjainlaitteen väliseen kommunikaatioon käytetään CAN-väylää sen hyvän häiriönsietoisuuden takia. Piirilevysuunnittelussa sijoitetaan erotuskondensaattoreita käyttöjännitelinjoihin lähelle komponentteja ja käytetään erotusta tai muuta häiriön poistoa tulo- sekä lähtökana- vissa. Ohjainlaitteelle valittiin alumiininen kotelo sen mekaanisen- sekä häiriönsuojauksen takia. Näyttömoduulin kotelo 3D tulostetaan, sillä se on kustan- nustehokkain tapa.

### 3.3 Toiminnallinen kuvaus

Vaihteiston ohjainlaitteiston tulisi kokonaisuutena toimia seuraavasti: vaihteenvaihtinsin tuottaa tilatietoa, joka päivittyy CAN-väylää pitkin näyttömoduulille ja vaihteiston ohjainlaitteelle, vaihteenvaihtinsin tila vaikuttaa vaihteiston ohjainlaitteen tilaan. Vaihteenvaihtinsin tila luetaan näyttömoduulille vain sen näyttämiseksi näytöllä informaatiotarkoitukseen. Vaihteenvaihtinsin tuottaa tilatiedot parkki, pakki, vapaa, ajo sekä vaihteen vaihto isommalle ja vaihteenvaihto pienemmälle. Vaihteenvaihtinsimella 1–5 vaihteen vaihto toiminnot voidaan kytkeä vaihteiston ohjainlaitteelta päälle käyttäen DIP-kytkintä nro 1.

Sekventiaalishifterissä on sisällään kaksi mikrokytkintä, jotka tuottavat digitaalisen asentosignaalin vaihteiston ohjainlaitteelle. Sekventiaalishifterillä voidaan vaihtaa vaihteita isommalle ja pienemmälle mikäli sekventiaalishifteri on neutraalissa tilassa ja kumpikaan mikrokytkin ei ole aktiivisena. Kun sekventiaalishifterin keppi liikutetaan jompaankumpaan ääriasentoon, kyseisen asennon mikrokytkin aktivoituu. Sekventiaalishifterin toimisuunta voidaan muuttaa DIP-kytkimellä nro 2, jos sekventiaalishifteri halutaan asentaa eripäin. Sekventiaalishifterin päähän sijoitetulla napilla voidaan kytkeä turbiinilukko päälle tilanteissa, joissa ei haluta turbiinin aiheuttamaa luistoa, joka vaikuttaa auton vääntövasteeseen. nappi vaikuttaa turbiinilukon tilaan vain silloin kun vaihteen vaihtinsin on asennolla vapaa tai ajo. jos turbiinilukko on napilla saatettu aktiiviseen tilaan ja vaihteenvaihtinsin liikutetaan asentoon parkki tai pakki, turbiinilukko kytkeytyy automaattisesti pois päältä. Turbiinilukon nappi on ohjelmoitu siten, että nappia on painettava vähintään 200 ms ja enintään 2 s, jotta turbiinilukon tila vaihtuu.

Näyttömoduulin tarkoitus on tuottaa vaihteiston ohjainlaitteen käyttäjälle tilatietoa vaihteenvaihtinsin asennosta, vaihteiston ohjainlaitteen tilakoneen tilasta, eli kytkeytyneestä vaihteesta, 1–5. Näyttömoduulilla näytetään vaihteistoöljyn lämpötila ja tuotetaan virheviestejä tilanteessa, jossa jokin sisääntulokanava on juuttunut maadoitettuun tilaan, vaihteen vaihtinsin tai vaihteiston ohjainlaite puuttuu CAN-väylältä. Näyttömoduulin sivuun on sijoitettu kaksi nappia, joilla voidaan säätää näyttömoduulin kirkkautta.

Vaihteiston ohjainlaitteen tulisi lukea vaihteenvälitsimen tila CAN-väylää pitkin. Sekventiaalishifteri on kytketty vaihteiston ohjainlaitteen digitaalisiin sisääntuloihin, joista luetaan sekventiaalishifterin tila. Tilatiedoille on tehty suodatusta seuraavasti, mikäli sekventiaalishifteriltä tuleva tilatieto jää päälle yli kahdeksi sekunniksi, tai edellisen vaihtenvaihto on vaihteiston ohjainlaitteella kesken, ei vaihteen vaihtoa suoriteta. Kun vaihteenvälitsin on tilassa parkki tai pakki, vaihteeksi voidaan valita vain ensimmäinen tai toinen vaihde. Kun vaihteenvälitsimen tila on vapaa tai ajo, vaihteeksi voidaan valita yhdestä viiteen mikä vain vaihde, mikäli valittuna on toista vaihdetta suurempi vaihde, kun vaihteenvälitsin liikutetaan vapaalta tai ajolta, pakille tai parkille, vaihteiston ohjainlaitteen tilakoneen vaihteeksi palautetaan toinen vaihde, tämä johtuu siitä, että vaihteiston venttiilipöydän rakenne aiheuttaa saman vaihdoksen venttiilipöydän tilassa. Vaihteiston ohjainlaitteeseen on liitetty myös nappi, jolla voidaan palauttaa vikatilanteessa vaihteiston ohjainlaitteen tilakone takaisin toiselle vaihteelle. Tämä tilanne saattaa esiintyä esimerkiksi silloin, kun vaihteistoöljy on päässyt kuumenemaan liikaa ja öljyn viskositeetin muutos sekoittaa vaihteiston venttiilipöydän oikeanlaisen toiminnan.

## 4 LAITTEISTO

### 4.1 Laitteistovalinnan pohjatiedot

Laitteistovalinnoissa pyrittiin noudattamaan vaatimusmäärittelyssä saatuja vaatimuksia. Kustannusten kannalta pyrittiin valitsemaan komponentteja, jotka olisivat hinnaltaan edullisia mutta olisivat varustettu mahdollisimman kattavilla datalehdillä, jolloin säästetään kehitystyöhön käytetyissä kustannuksissa. Esimerkiksi jos piti valita kahdesta komponentista, joista ensimmäisellä oli huokeampi hinta, mutta huonompi datalehti ja toisella parempi datalehti mutta kalliimpi hinta, valittiin käytettäväksi paremmalla datalehdellä varustettu komponentti, koska paremmalla datalehdellä varustetun komponentin käyttäminen on helpompaa ja riskit virheisiin ovat pienemmät.

Käyttökohde ja käyttöympäristö puolestaan ohjasivat käyttämään automotive-luokituksen komponentteja. Automotive-komponentit tarjoavat yleensä pidemmän käyttöiän ja korkeamman luotettavuuden verrattuna tavanomaisiin kuluttajaluokan komponentteihin huolellisen testauksen sekä laadukkaiden materiaalien ansiosta (Mouser Electronics Nd, 7). Tämä on erityisen tärkeää, koska laitteisto altistuu lämpötilanvaihteluille, tärinälle ja mahdollisesti sähkömagneettisille häiriöille. Kaikkia komponentteja ei ollut mahdollista tai kannattavaa valita automotive-komponenttina hinnan tai saatavuuden vuoksi, mutta kuitenkin pyrittiin varmistamaan, että komponenteilla olisi vähintään riittävä käyttölämpötilaskaala.

### 4.2 ATmega32U4-mikrokontrolleri

ATmega32U4 on Atmelin kahdeksan bittinen mikrokontrolleri, joka toimii 16MHz kellotaajuudella. ATmega32U4-mikrokontrollerissa on 32KB flash-muistia, 2.5kB SRAM-muistia sekä 1KB EEPROM-muistia (Atmel. 2015). Yksi tärkeistä ominaisuuksista minkä takia kyseinen mikrokontrolleri valittiin käyttöön, on sen sisäinen USB-kontrolleri. Sisäisen USB-kontrollerin ansiosta USB-portin integroimiseksi ohjainlaitteeseen ei vaadita kuin muutama komponentti, joilla suojataan USB-kontrolleria ylijännitteiltä. ATmega32U4-mikrokontrollerissa on neljä

kahdeksan bittistä PWM-ulostulokanavaa, neljä ohjelmoitavaa 2–16 bittistä PWM-ulostulokanavaa, sekä kuusi ohjelmoitavaa 2–11 bittistä PWM-ulostulokanavaa (Atmel. 2015). ATmega32U4-mikrokontrollerissa on myös 12 kymmenen bittistä analogia-digitaalimuunnin sisääntulokanavaa, jotka mahdollistavat analogiamittausten tekemisen. ATmega32U4-mikrokontrollerissa SPI- ja I2C-väylät sekä laaja käyttölämpötilaskaala  $-40^{\circ}\text{C}$  -  $+85^{\circ}\text{C}$ . (Atmel. 2015.)

### **4.3 Microchip MCP2515 CAN-kontrolleri ja NXP TJA1050 CAN-lähetin-vastaanotin**

MCP2515 CAN-kontrolleri ja TJA1050 CAN-lähetin-vastaanotin ovat suosittu komponenttivalinta matalan kynnyksen ATmega-mikrokontrolleri projekteissa, joissa tarvitaan CAN-väylää. Tämä johtuu siitä, että niitä on ollut saatavilla jo 2000-luvun alkupuolelta ja ne ovat helppokäyttöisiä sekä edullisia. Mikäli mikrokontrollerin koodin haluaisi kirjoittaa Arduino IDE:llä, löytyy MCP2515 CAN-kontrollerille lukuisia eri kirjastovaihtoehtoja.

MCP2515 CAN-kontrolleri liitetään mikrokontrolleriin käyttäen SPI-väylää ja se on varustettu kahdella vastaanottopuskurilla. Kun CAN-väylää pitkin tulee viesti, CAN-kontrolleri varastoi sen puskuriinsa ja siirtää sitten SPI väylää pitkin mikrokontrollerille käsiteltäväksi. CAN-kontrolleriin voidaan myös asettaa suodattimia vain tiettyjen viestien vastaanottamiseksi. Näitä 29-bittisiä ID-suodattimia on kuusi kappaletta, ja lisäksi CAN-kontrollerissa on kaksi 29-bittistä ID-maskia. Kun mikrokontrollerilla halutaan lähettää viesti CAN-väylälle, mikrokontrolleri lähettää viestin SPI-väylää pitkin CAN-kontrollerin lähetyspuskuriin, josta se sitten lähettää viestin väylälle. MCP2515 CAN-kontrollerilla on myös erilaisia asetus rekistereitä, joista voidaan esimerkiksi asettaa väylän nopeus sekä esimerkiksi vastaanotettavien viestien ID-maskeja käyttöön, Myös nämä toiminnot suoritetaan SPI-väylän kautta. (Microchip. nd.)

TJA1050 CAN-lähetin-vastaanotin on suunniteltu käytettäväksi auto- ja teollisuussovelluksissa. Se toimii rajapintana CAN-kontrollerin ja fyysisen väylän välillä. TJA1050 päätoiminto on signaalin muuntaminen. Se muuntaa CAN-kontrol-

lerin TXD (lähetysdata) kanavasta tulevan signaalin CAN High ja CAN Low -kanavien väliseksi differentiaalisignaaliksi, joka lähetetään CAN-väylään. Vastavasti se muuntaa CAN-väylältä vastaanotetun differentiaalisignaalin takaisin digitaaliseksi signaaliksi RXD (vastaanottodata) kanavaan CAN-kontrollerille. (NXP. 2003.)

#### 4.4 IRF540N PWM-ulostulokanavat

722.6-automaattivaihteiston solenoidien ohjaamiseksi vaihteiston ohjainlaitteeseen toteutettiin kuusi ulostulokanavaa IRF540N-mosfetilla. IRF540N-mosfettien hilanohjauspiiri toteutettiin optoerottimilla, jolloin mikrokontrollerin 5V logiikkajännitteen sijaan hilan lataamiseen voidaan käyttää solenoidien käyttöjännitettä, joka on 14.5V. Ulostulokanavat toteutettiin mosfeteilla, jotta solenoideja voidaan ajaa PWM-ohjauksella, jolloin solenoidien asentoa voidaan säätää portaattomasti.

Käytettävän mosfetin valintaperusteena käytettiin sopivaa virrankestoa solenoidien ajamiseen, oikeaa jännitealuetta sekä saatavuutta piirilevyn teettämispalvelussa. Solenoidien vastusarvot saatiin 722.6-automaattivaihteiston manuaalista. Solenoidien vastusarvot ovat välillä 2–6,5  $\Omega$  (ATSG. 2009, 23). jolloin laskennassa käytetään pienintä vastusarvoa ja jotta saadaan laskettua suurin mahdollinen solenoidien ottama virta käyttäen ohmin lakia. Virta lasketaan seuraavalla kaavalla, jossa  $I$  on yksittäisen solenoidin ottama maksimivirta,  $U$  on vaihteistonohjaimen käyttöjännite, 14,4 V auton ollessa käynnissä, ja  $R$  on yksittäisen solenoidin pienin resistanssi, eli 2  $\Omega$

$$I = \frac{U}{R} . \quad (1)$$

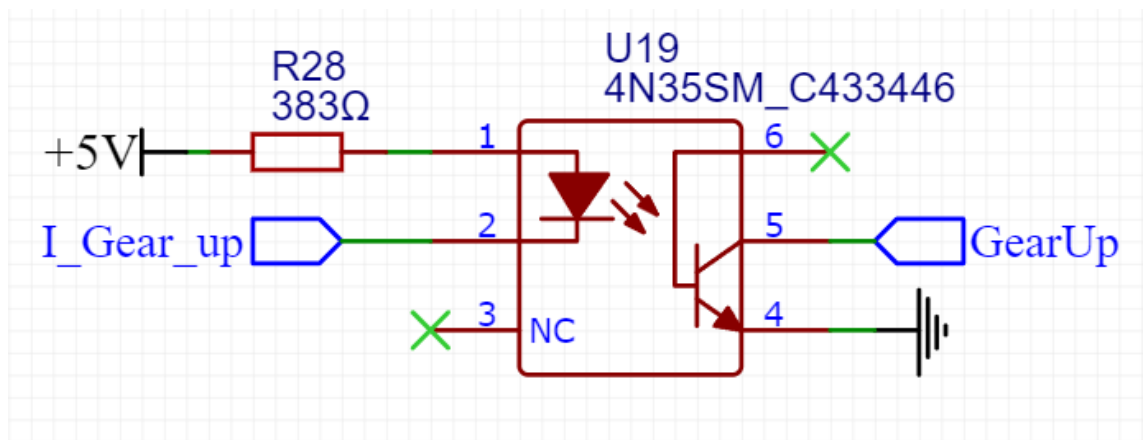
Kun sijoitetaan arvot kaavaan, tulokseksi saadaan:

$$I = \frac{14,4 \text{ V}}{2 \Omega} = 7,2 \text{ A} . \quad (2)$$

Yksittäisen solenoidin ottamaksi maksimi virraksi saadaan siis 7,2 ampeeria, jolloin IRF540N-mosfetin 33 ampeerin maksimivirta on riittävä tähän käyttötarkoitukseen. Lisäksi IRF540N-mosfetin suurin sallittu lähteen ja nielun välinen jännite-ero on 100 V (Infineon. 2003,1–2), joka ylittää reilusti vaihteiston ohjainlaitteen 14,4 V käyttöjännitteen ja näin ollen soveltuu käytettäväksi vaihteiston ohjainlaitteeseen.

#### 4.5 Optoerotetut-sisääntulokanavat

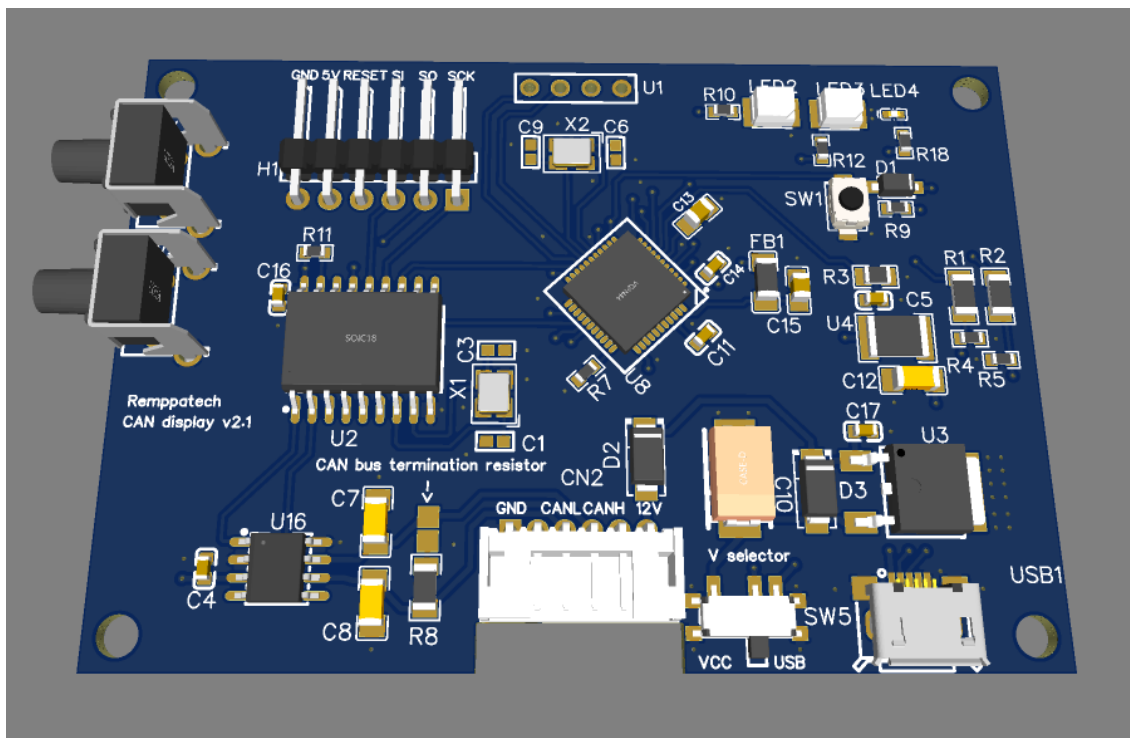
Vaihteiston ohjainlaitteen sisääntulokanavat toteutettiin käyttäen optoerottimia häiriöiden poistamiseksi sekä mikrokontrollerin sisääntulokanavia suojelemiseksi. Digitaalisisääntulojen optoerotin kytkettiin seuraavan kuvion 5 mukaisesti kytkemälle sisääntulopuolen ledin anodi etuvastuksen kautta jännitteeseen ja katodi sisääntuloliittimeen, jotta sisääntuloista saadaan maaohjattuja. Ledin etuvastus mitoitettiin ledin kynnsjännitteen ja tyypillisen virran mukaan.



KUVIO 5. Digitaalisisääntulon optoerotin.

Analogiasisääntulot kytkettiin valmistajan ohjeen mukaisesti ja niillä luetaan 0- 5V signaaleja. Seuraavassa kuviossa 6 esitettynä toteutettu kytkentä. Analogisessa optoerottimessa on toteutettu takaisinkytkentä, jolla optoerotin sisääntulopiiri säätää optoerotin ulostulojännitteen lineaariseksi sisääntulojännitteen kanssa.





KUVA 9. Näyttömoduulin ohjainkortti.

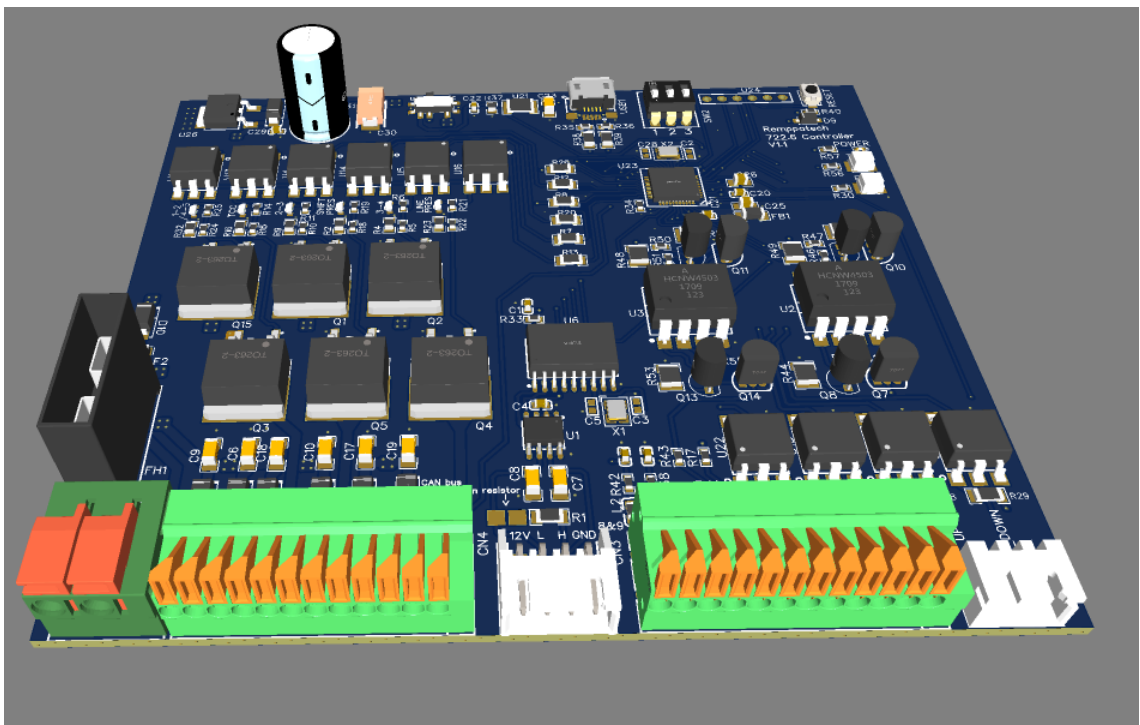
Näyttömoduulin ohjainkorttiin kasattiin myös kaksi nappia, jotta näytön kirkkautta saataisiin helposti säädettyä sekä saataisiin haluttaessa rakennettua erilaisia näkymiä ja valikkorakenteita, joita voi selata napeilla. Ohjainkorttiin kasattiin myös micro-USB -liitin, jotta ohjelman lataamisesta saataisiin tehtyä mahdollisimman yksinkertaista. Ohjainkortin vasemmasta yläreunasta löytyy liitinrima, jota käytetään bootloaderin lataamiseksi ATmega32U4-mikrokontrollerille. Bootloader on ohjelmisto, joka mahdollistaa ohjelman lataamisen mikrokontrollerille käyttäen USB-porttia.

Näyttömoduuli kasataan ruuvaamalla SSD1309 2.4" OLED-näyttö kiinni näyttömoduulin ohjainkorttiin ruuveilla. OLED-näytön ja ohjainkortin väliin asennetaan muoviholkit, jotta ohjainkortin ja OLED-näytön väliin jää riittävä rako, ettei piirilevyt kosketa toisiaan. Kun OLED-näyttö ja ohjainkortti on ruuvattu yhteen, juotetaan OLED-näytön ja ohjainkortin väliin liitinrima, jolla syötetään ohjainkortilta OLED-näytölle 5 V käyttöjännite, maa ja I2C-väylä. Tämän jälkeen ohjainkortille ladataan bootloader, koska sen latausliittimeen ei pääse käsiksi, kun näytön kotelo kasataan. Nyt OLED-näyttö ja ohjainkortti voidaan asentaa koteloon, kotelon kansi kiinnitetään kahdella ruuvilla ja kotelon kannessa olevat muoviset olakkeet pitävät OLED-näytön ja ohjainkortin paikallaan. Tämän jälkeen näyttöön voidaan

ladata ohjelma USB-portista. Ohjainkortin alareunassa on JST-tyyppinen liitin, jolla näyttömoduuli saadaan kytkettyä valmiilla kaapelilla kiinni vaihteiston ohjainlaitteeseen.

#### 4.7 Vaihteiston ohjainlaite

Vaihteiston ohjainlaitteen ohjainkortissa (KUVA 10) käytettiin mikrokontrollerina samaa ATmega32U4-mikrokontrolleria kuin näyttömoduulissa. CAN-väylän toteutuksessa käytettiin myös samoja MCP2515 CAN-kontrolleria sekä TJA1050 CAN-lähetin-vastaanotinta kuin näyttömoduulissa. Ohjainkortissa on neljä optoerotettua digitaalista maaohjattua sisääntulokanavaa, kaksi optoerotettua analogista sisääntulokanavaa sekä kaksi suodatettua digitaalista sisääntulokanavaa HALL-antureita varten. Ohjainkortissa on kuusi MOSFET-ohjattua PWM-ulostulokanavaa sekä yksi PWM-ulostulokanava suoraan mikrokontrollerilta, jolla voidaan syöttää optionaaliselle servolle asento-ohje.



KUVA 10. Vaihteiston ohjainlaitteen piirikortti.

Ohjainkortissa käytettiin jousiliittimiä sellaisissa kanavissa, joihin ei kytketä valmista johtoa, jotta vaihteiston ohjainlaite olisi mahdollisimman helppo kytkeä. Kanavissa, joihin oli mahdollista saada valmis johto, käytettiin JST-tyyppisiä liittimiä. JST-tyyppiset liittimet löytyvät ohjainkortin alareunasta, joista oikeanpuoleinen on sekventiaalishifterille ja vasemmanpuoleinen näyttömoduulille. Ohjainkortin alareunassa on kolme jousiliitintä, joista vasemmanpuoleisin on käyttöjännitteelle, keskimäinen ulostulokanaville ja oikeanpuoleinen sisääntulokanaville.

Ohjainkortin käyttöjännite tulee lattasulakkeen läpi, ja logiikan 5 V käyttöjännite tuotetaan samantyyppisellä automotive-luokituksen jänniteregulaattorilla kuin näyttömoduulinkin. Ohjainkortin yläreunaan on sijoitettu kolmella kytkimellä varustettu DIP-kytkin, jolla voidaan tarvittaessa tehdä parametointia. Vaihteiston ohjainlaitteen koteloksi valittiin alumiininen kotelo sen hyvän saatavuuden, edullisen hinnan sekä hyvän mekaanisen sekä häiriön suojauksen takia. Ohjainkortti on helppo asentaa koteloon. Kotelon liittimienpuoleinen päätylevy (kuvassa 11) 3D-tulostettiin muovista ensimmäisiin versioihin.



KUVA 11. Vaihteiston ohjainlaite koteloon asennettuna.

## 5 OHJELMA

### 5.1 Suunnittelukäytännöt

Ohjelmistosuunnitteluvaihe on sulautetun järjestelmän kehittämisen keskeinen osa. Sen tavoitteena on luoda ohjelmiston kokonaisrakenne ja määrittää, miten eri toiminnalliset moduulit liittyvät toisiinsa. Tässä vaiheessa määritellään myös moduulien rajapinnat, suunnitellaan ohjelmiston globaali tietokanta tai tietorakenne, asetetaan suunnitteluun liittyvät rajoitteet ja laaditaan ohjelmiston suunnittelukuvaus. Suunnitteluvaihe jakautuu kahteen osaan: yleissuunnitteluun ja yksityiskohtaiseen suunnitteluun. (Yin & Jiang 2024.)

Yleissuunnitteluvaiheessa ohjelmistovaatimukset muutetaan vastaavaksi ohjelmistoarkkitehtuuriksi, jossa jokaisella arkkitehtuurin osalla on selkeä tehtävä ja tarkoitus. Yleissuunnittelun tavoitteena on luoda pohja ohjelmiston rakenteelle ja valmistella seuraavaa, tarkempaa suunnitteluvaihetta. (Yin & Jiang 2024.)

Yksityiskohtainen suunnittelu keskittyy ohjelmistoarkkitehtuurin jokaisen moduulin tarkempaan määrittelyyn. Tämä sisältää algoritmien ja moduulien yksityiskohtaisen suunnittelun sekä moduulien rajapintojen tarkentamisen yleissuunnitelman pohjalta. Samalla suunnitellaan ohjelmiston yksikkötestaus, joka helpottaa myöhemmässä toteutusvaiheessa tehtävää testausprosessia. Näiden vaiheiden avulla varmistetaan ohjelmiston rakenteen selkeys ja toteutuksen tehokkuus. (Yin & Jiang 2024.)

### 5.2 Ohjelman suunnittelu

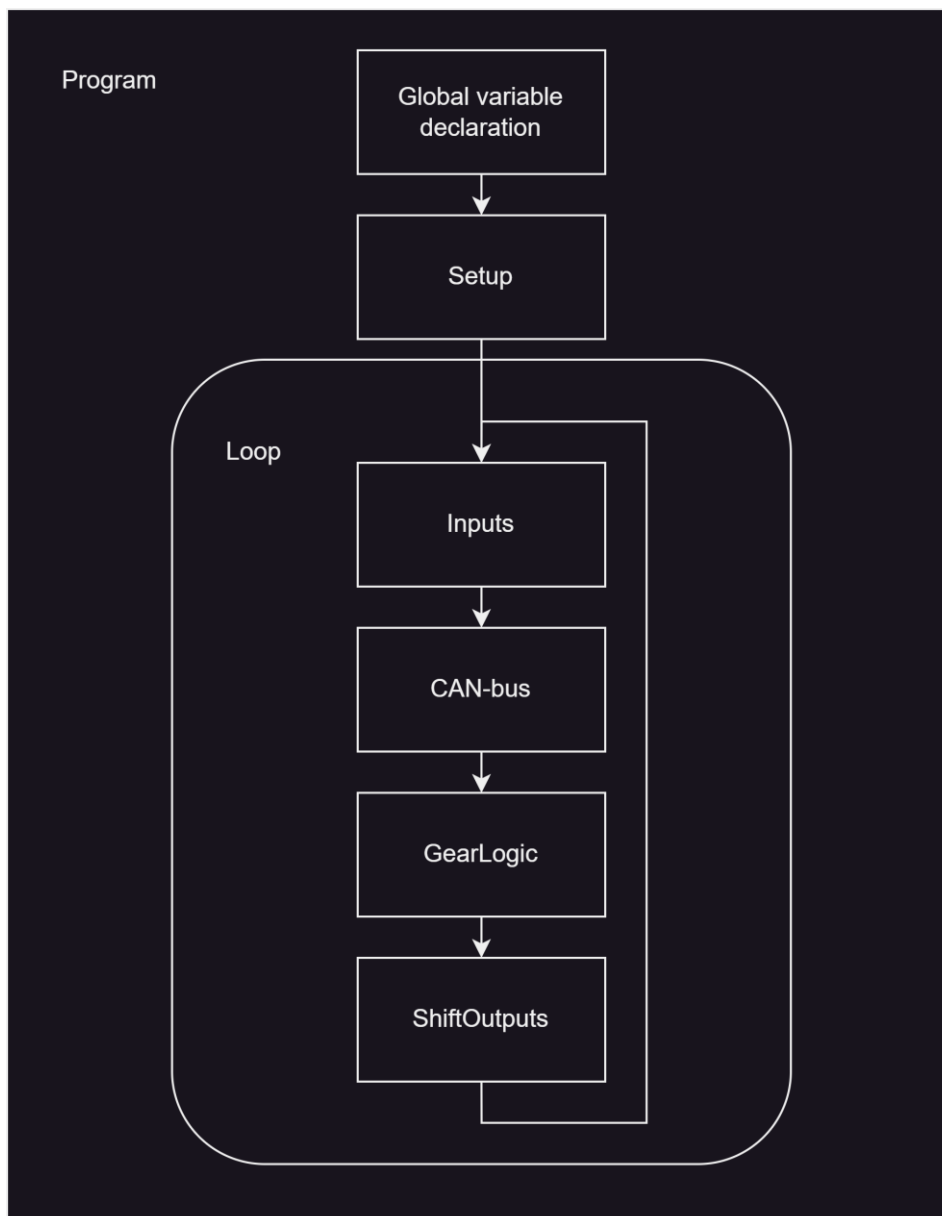
Ohjelman suunnittelussa on pyritty yksinkertaisiin ratkaisuihin, hyvään luettavuuteen sekä helppoon muokattavuuteen, sillä ohjauslaite on vielä kehitysvaiheessa. Ohjelmiston toimintoja on jaettu selkeisiin funktioihin, jotka kukin suorittavat oman rajatun tehtävänsä. Tämä lähestymistapa parantaa ohjelman ylläpidettävyyttä ja tekee koodin rakenteesta loogisemman.

Vielä parempi tapa olisi toteuttaa ohjelman eri osia olioina, mutta tässä kehitysvaiheessa sitä ei ole koettu tarkoituksenmukaiseksi ja tehokkaaksi. Olio-ohjelmointiin siirtyminen voisi tuoda mukanaan lisätyötä ja monimutkaisuutta, mikä ei tässä projektin vaiheessa tue sujuvaa kehitystä ja koodin nopeaa muokkaamista.

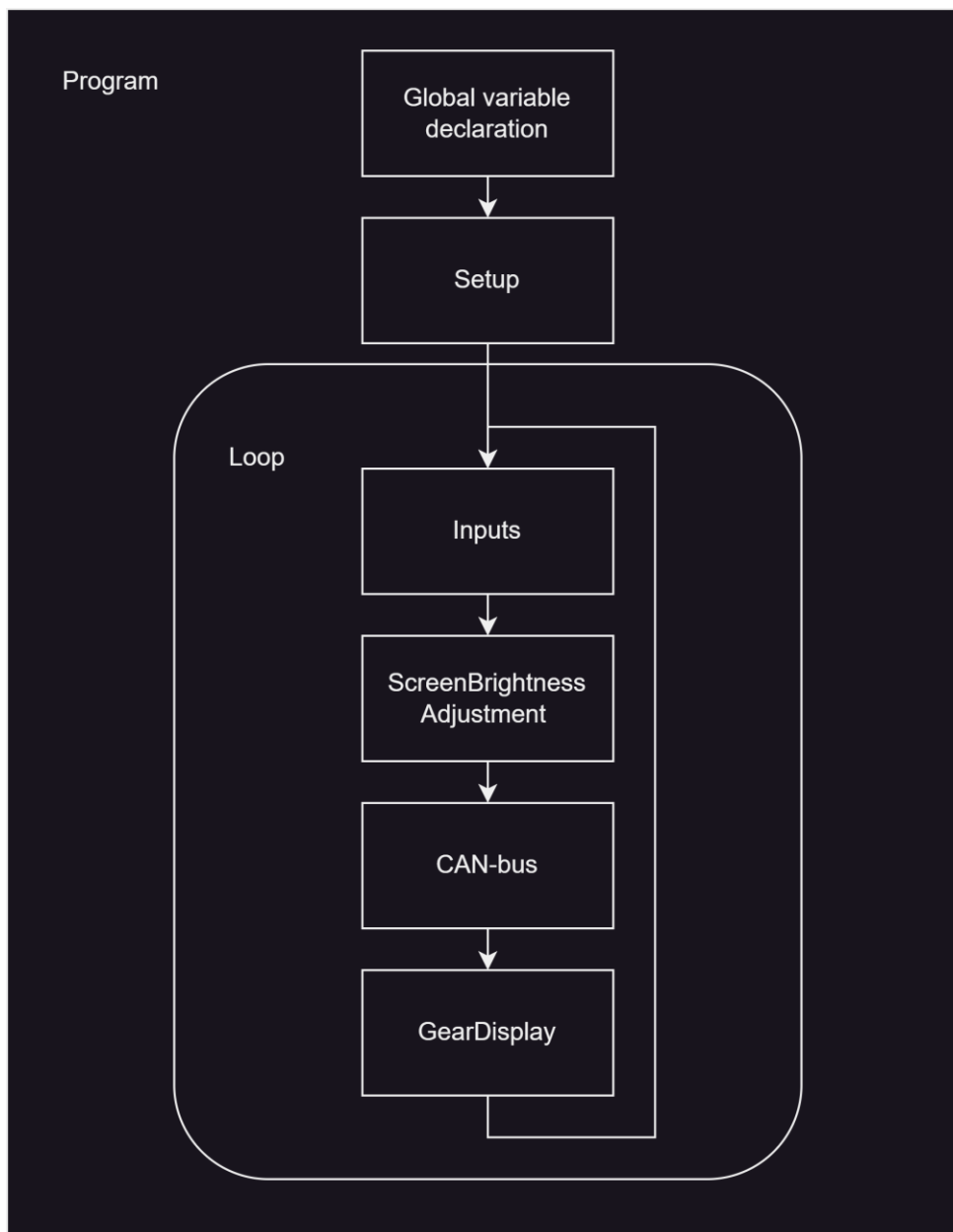
Tällä hetkellä tärkeintä on säilyttää ohjelmiston yksinkertaisuus ja muokattavuus, jotta kehitysprosessi pysyy ketteränä ja tarvittavia muutoksia voidaan tehdä helposti. Jatkokehityksessä olioiden hyödyntämistä voidaan harkita, mikäli ohjelmiston monimutkaisuus kasvaa ja olioiden tarjoamat hyödyt, kuten parempi abstraktiotaso ja uudelleenkäytettävyys, alkavat painaa vaakakupissa enemmän.

Ohjelmistoa suunniteltaessa ohjelmoinnin tueksi laadittiin toiminnallisia kaavioita, joiden avulla ohjelman eri osa-alueet voidaan toteuttaa selkeästi ja järjestelmällisesti. Toiminnalliset kaaviot valittiin työkaluksi niiden nopean toteutuksen ja helpon luettavuuden ansiosta. Ne tarjoavat yksinkertaisen ja visuaalisen tavan hahmottaa ohjelman toimintalogiikka ja rakenteet, mikä helpottaa sekä suunnittelua että toteutusta.

Kaavioiden avulla voidaan myös varmistaa, että kaikki tarvittavat toiminnot ja niiden väliset yhteydet tulevat huomioiduiksi jo suunnitteluvaiheessa, mikä vähentää virheiden ja puutteiden riskiä. Tämä lähestymistapa tukee ohjelmiston kehittämistä erityisesti tilanteissa, joissa tarvitaan nopeaa iterointia ja joustavuutta. Ohjainlaitteen ohjelman pää rakenne suunniteltiin seuraavan kuvion 7 mukaisesti ja näyttömoduulin ohjelman pää rakenne kuvion 8 mukaisesti.



KUVIO 7. Vaihteiston ohjainlaitteen ohjelman päätason rakenne.



KUVIO 8. Näyttömoduulin ohjelman päätason rakenne.

### 5.3 Ohjelman toteutus

Vaihteiston ohjainlaitteen ohjelma tehtiin Arduinon tarjoamalla ohjelmointiympäristöllä, Arduino IDEllä. Kyseinen ohjelmointiympäristö valittiin käytettäväksi sen helppokäyttöisyyden ja laajan kirjastosaatavuuden takia sekä sen tarjoaman suorran tuen ATmega32U4-mikrokontrollerille. Vaihteistonohjainlaitteen toiminto- ja laitteistorakennetta on kuvattu liitteessä nro 1.

Vaihteiston ohjainlaitteen ohjelmarakenne on toteutettu modulaarisesti, jolloin eri toiminnallisuudet on jaettu omiin moduuleihinsa ohjelmarakenteen selkeyttämiseksi ja testaamisen helpottamiseksi. Kuvassa 12 on esitetty pääohjelman rakenne, jossa eri toiminnallisuuskokonaisuudet on toteutettu omina funktioinaan. Ensimmäisenä ohjelmassa siirretään kulunut millisekuntimäärä mikrokontrollerin käynnistymishetkestä muuttujaan `currentMillis` käytettäväksi ajastuslaskennoissa.

```
void loop() {  
    currentMillis = millis();  
  
    CAN();  
  
    inputs();  
  
    gear_logic();  
  
    shift_outputs();  
}
```

KUVA 12. Vaihteiston ohjainlaitteen pääohjelma.

Seuraavana pääohjelmarakenteessa on CAN-väylä toiminnallisuus funktiona `CAN()`, jossa vastaanotetaan viestit väylältä sekä lähetetään tarvittavat viestit väylälle. Kyseinen moduuli sisältää myös tarvittavat skaalaukset, jotta CAN-viestistä saatu data saadaan siirrettyä muuttujiin helposti käytettävässä muodossa. Vaihteiston ohjainlaitteen fyysiset sisääntulokanavat käsitellään funktiossa `inputs()`, jossa sisääntulokanavien tilatiedot siirretään muuttujiin sekä toteutetaan mittausten suodatukset, joilla varmistetaan mittausten oikeellisuus.

Tilakone vaihteiston tilasta on toteutettu funktiossa `gear_logic()`, tilakoneen tiloina ovat vaihteiston viisi sähköisesti kytkettävää vaihdetta joita ohjataan sekventiaalivaihdekepillä, tai vaihtoehtoisesti Vaihteenvälitsimen + ja – toiminnoilla. Vaihteenvälitsimen ollessa asennossa vapaa tai ajo, on vaiheita mahdollista vaihtaa ensimmäisestä viidenteen asti. Vaihteiston ollessa asennossa parkki tai peruutus vaihteen voi valita ensimmäisen ja toisen väliltä. Jos vaihteenvälitsin liikutetaan asennosta ajo tai vapaa asentoon parkki tai peruutus vaihteen ollessa suurempi kuin kaksi, vaihteiston mekaniikka palauttaa vaihteen automaattisesti toiselle

vaihteelle, tällöin myös vaihteiston ohjainlaitteen tilakone muuttaa tilakoneen tilaksi vaihteen kaksi.

Viimeisenä pääohjelmatasolla on esitelty funktio `shift_outputs()`, jossa on toteutettu vaihteidenvaihtojen ohjaus PWM ohjattuja MOSFET-ulostulokanavia. Funktion sisällä on toteutettu pulssimaiset ohjaukset vaihteenvaihtosolenoidille, jolloin solenoidia pidetään päällä riittävä aika vaihteen kytkeytymiseksi. Jos halutaan lisätä vaihteiston käyttömukavuutta, on ohjattava myös linjapaine, sekä vaihtopainesolenoidia vaihdetta vaihtaessa. Jos vaihtopainetta pienennetään vaihdetta vaihdettaessa, on myös vaihteenvaihtosolenoidia ohjattava ajallisesti kauemmin, jotta vaihteen vaihto suoriutuu loppuun.

## 6 TESTAUS

### 6.1 Lähtökohdat

Hyvä laitteisto- ja laiteohjelmistosuunnittelu muodostaa perustan luotettaville järjestelmille. Pelkkä suunnittelu ei kuitenkaan takaa, että laite todella täyttää tavoitteensa, ellei sitä testata kattavasti. Nykyaikaiset sulautetut järjestelmät ovat niin monimutkaisia ja niissä on paljon ennakoimattomia laitteisto- ja ohjelmistovuorovaikutuksia, että luotettavuutta ei voida odottaa ilman realistista testausohjelmaa. (Labrosse ym. 2007.)

Tämä tarkoittaa, että tuotetta on testattava perusteellisesti ja kaikki mahdolliset vikaantumistavat on tutkittava. Esimerkiksi, jos ohjelman on tarkoitus säilyttää muuttujien arvot jännitekatkoksista huolimatta, mutta sitä ei testata kunnolla, saatetaan päätyä toimittamaan tuote, jossa on huomaamattomia mutta kriittisiä virkoja. (Labrosse ym. 2007.)

### 6.2 Moduulitestaus

Vaihteiston ohjainlaitteen testaaminen aloitettiin suorittamalla näyttömoduulin ja vaihteiston ohjainlaitteen ohjelmiston moduulitestaus. Moduulitestauksessa testataan ohjelmakokonaisuuden rakentavia itsenäisiä osia yksitellen, ja tarkastellaan vastaako moduulin toiminta haluttua toiminnallisuutta.

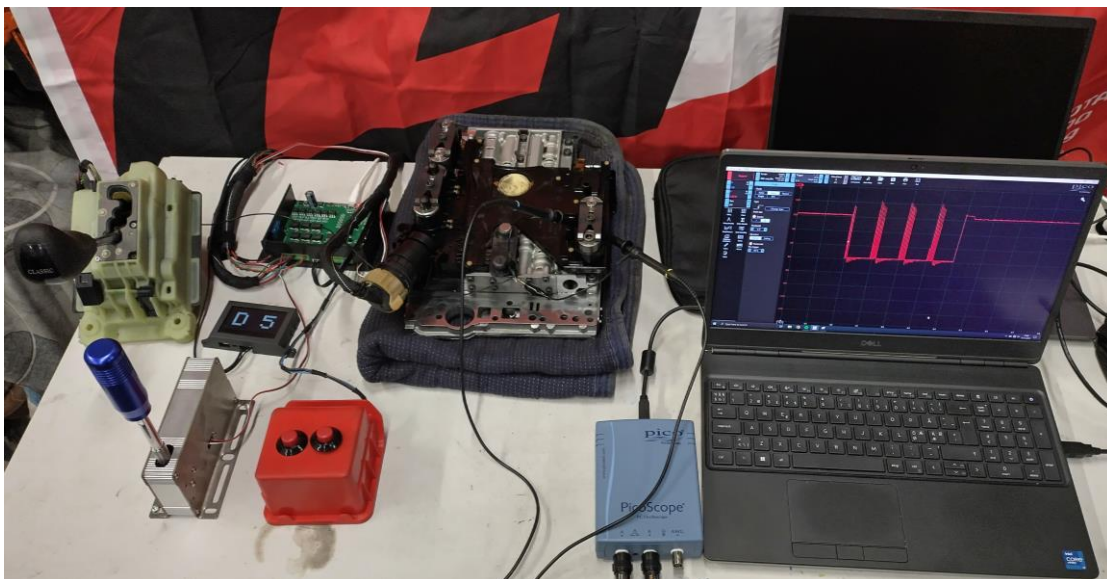
Koska Arduino IDE-kehitysympäristö ei tue debuggausta eli virheenjäljitystä ATmega32U4-mikrokontrollerin kanssa, on testausta varten tehtävä omia testausohjelmia. Testausohjelmaan sijoitetaan testattava funktio ja syötearvoja voidaan joko kysyä sarjaliikenteen kautta käyttäen `Serial.read()` -funktia. Tai kovakoodaamalla arvot muuttujiin. On myös mahdollista tehdä ohjelmarakenteita, jotka muuttavat testattavalle funktiolle syötettäviä arvoja. Funktioiden vastetta tarkasteltiin tulostamalla ulostulomuuttujien arvot sarjaliikenteen kautta käyttäen `Serial.print()` -funktia.

### 6.3 Integraatiotestaus

Integraatiotestaus aloitettiin purkamalla 722.6-automaattivaihteistosta venttiilipöytä, joka kiinnitettiin irrallisena vaihteiston ohjainlaitteeseen. Testausta varten rakennettiin johtosarja, jolla vaihteiston ohjainlaite kytkettiin venttiilipöydässä olevaan liittimeen. Lisäksi testikytkentään liitettiin vaihteenvälitsin, sekventiaalishifteri, näyttömoduuli ja kaksi nappia. Ensimmäinen nappi on turbiinilukon käyttökytkin, ja toinen resetointinappi, jolla voidaan palauttaa vikatilanteessa vaihteiston ohjainlaitteen tilakone takaisin toiselle vaihteelle.

Kun testauslaitteisto (kuvassa 13) oli saatu valmiiksi, aloitettiin varsinainen testaus. Ensimmäisenä testattiin, että näyttömoduulille päivitty vaihteenvälitsimen asento oikein sekä vaihteenvälitsimen vaihto käskyjen (vaihde isommalle, vaihde pienemmälle) vaikutus vaihteistonohjainlaitteen tilakoneeseen. Tämän jälkeen testattiin vaihteiston ohjainlaitteen tilakoneen vaihde 1–5 päivitty näytölle oikein. Vaihteenvaihtosolenoidien ohjausta tarkasteltiin oskilloskoopin avulla, jolla tarkastettiin oikea ohjausaika sekä flyback-diodin toiminta. Flyback-diodilla estetään induktiivisten kuormien aiheuttamien suurien jännitepiikkien syntyminen, kun ohjaus kytkeytyy pois päältä. Seuraavaksi testattiin napit, joilla ohjataan turbiinilukkoa sekä resetoidaan vaihteiston ohjainlaitteen tilakoneen tila takaisin toiselle vaihteelle. Turbiinilukon napin toiminta testattiin, kun vaihteenvälitsin oli asetettu ajoasentoon. Tämän lisäksi testattiin, että turbiinilukon ohjaus sammuu, kun vaihteenvälitsin liikutetaan pakkiasentoon. Resetointinapin toiminta testattiin vaihtamalla vaihteiston ohjainlaitteen tilaa ensin eri vaihteille käyttäen sekventiaalivaihtokeppiä, jonka jälkeen resetointinappia painettiin noin yhden sekunnin ajan. Lisäksi testattiin, että resetointinapille määritellyt aikarajat toimivat.

Testeissä ei ilmennyt suurempia ongelmia. Ohjainlaitteen ohjelmaan tehtiin pientä hienosäätöä nappien toiminnan osalta ja näytön ohjelmaan tehtiin muutoksia näytön kirkkauden säätöön. Testin lopputuloksena voidaan todeta, että laitteisto toimi suunnitellusti ilman ongelmia, joita ei olisi voitu ohjelmallisesti korjata. Laitteisto vaatii pidempiaikaista testausta autoon asennettuna, jotta voidaan todentaa pitempiaikainen toimivuus.



KUVA 13. Vaihteiston ohjainlaitteen testauslaitteisto.

## 7 POHDINTA

Työn tarkoituksena oli toteuttaa Mercedes-Benz automaattivaihteistoon soveltuva vaihteiston ohjainlaite, joka täyttää annetut kriteerit (käyttäjäkunnan tarpeet, komponenttien kustannukset, helppokäyttöisyys ja saatavuus). Jotta näihin vaatimuksiin päästiin, todettiin, että paras ratkaisu olisi suunnitella piirikaaviot ja piirilevyt itse, jolloin ohjainlaite toteutettaisiin sulautettuna järjestelmänä. Sulautettuun järjestelmään päätyminen ratkaisi monta ongelmaa: esimerkiksi sopivaa laitteistoa oli vaikea löytää, jossa olisi ollut oikean tyyppiset sisääntulot ja ulostulot sekä se olisi ollut yhtä kustannustehokas ja mahdollistanut jonkin tyyppisen näytön käyttämisen.

Sulautettuun järjestelmään päätyminen toi kuitenkin mukanaan useita haasteita ja uuden opettelua oli paljon. Komponenttivalinnat veivät useita viikkoja ja vaati runsaasti perehtymistä. Piirikaavioiden piirtäminen oli kohtuullisen helppoa, mutta kuitenkin hyvien käytäntöjen opettelu teki piirikaaviosta luettavampia. Piirien suunnittelu taas itsessään oli haastavaa. Sisääntulot pyrittiin suojaamaan ulkoisilta häiriöiltä ja tässä tapauksessa se toteutettiin optoerottimilla niiden helpouden takia. Optoerottimet ovat kuitenkin kalliita ja kohtuullisen kookkaita, joten jatkokehityksessä ne tullaan korvaamaan TVS-diodeilla. Piirilevyn piirtäminen ja komponenttien sijoittelu piirilevyille vaati opiskelua, jotta saavutettaisiin mahdollisimman hyvä häiriönsuojaus. Piirilevyn johtimien suunnittelussa käytettiin apuna laskuria, jotta saavutettiin riittävät virran ja lämpökuorman kestot.

Koteloinnin osalta itse ohjainlaitteen kotelointi oli helppo ratkaista valmiilla alumiinikotelolla, mutta näyttömoduulin kotelointiin ei löytynyt valmista tuotetta. Tämän vuoksi päädyttiin suunnittelemaan kotelo itse ja teettämään 3D-tulostettuna, joka tuotti hieman lisätyötä. Ohjelmoinnin osalta projekti sujui kohtuu vaivattomasti koska itse ohjainlaitteen toiminnot pidettiin ensimmäisessä versiossa kohtuullisen yksinkertaisina. CAN-väylän viestit pyrittiin kuitenkin tekemään Mercedes-Benzin väylän mukaiseksi, jolloin teoriassa vaihteen pitäisi näkyä myös auton omassa mittaristossa.

Työn aikana syntyi runsaasti ajatuksia jatkokehityksestä. Tärkeimpänä ATmega-mikrokontrollerin korvaaminen uudemmalla, esimerkiksi STM32-mikrokontrollerilla, joka on tehokkaampi, halvempi ja tarjoaa runsaasti kirjastoja valmistajan puolesta. STM32-mikrokontrollerilla saavutettaisiin etu myös CAN-väylän kanssa. Koska STM32-mikrokontrolleri ei vaadi erillistä CAN-kontrolleria, vaan pelkkä CAN-lähetin-vastaanotin riittää, jolloin säästetään komponenttien piirilevyllä käyttämästä tilasta sekä kustannuksissa. Lisäksi korvaisin itse suunnitellut MOSFET-piirit valmiilla low-side kytkimillä, joita voidaan ajaa suoraan mikrokontrollerin ulostulolla, jolloin vältytään hilanohjauspiirin suunnittelulta ja säästetään tilaa. Digitaalisääntulot kannattaisi suojata TVS-diodeilla ja analogiasääntulot kannattaisi toteuttaa operaatiovahvistimilla, jolloin mittauksesta saataisiin mahdollisimman tarkka. Lisäksi laittaisin analogiamittauksiin alipäästösuodattimet, jotta ohjelmallista suodatusta vaadittaisiin mahdollisimman vähän. Lisäksi näkisin järkeväksi ratkaisuksi korvata nykyisen näyttömoduulin mahdollisimman valmiilla näytöllä, jossa olisi kosketusnäyttö tai muu syötteenvastaanotin valmiiksi, tällä säästettäisiin kasausajassa ja kehitystyössä.

Kokonaisuutena vaihteistonohjaimesta tuli vaatimukset täyttävä ja se toimi suunnitellusti. Kustannukset pysyivät kohtuullisina eikä itse työssäkään tullut kohtuuttomia vastoinkäymisiä. Kehitystyötä tullaan jatkamaan, kunhan nykyiselle versiolle on saatu kerrytettyä riittävästi käyttötunteja, jotta voidaan arvioida laitteiston toimivuutta tarkemmin.

## LÄHTEET

Dietrich, S. 2022. Understanding the Basics of Pulse Width Modulation (PWM). Control Automation. Verkkosivu. Viitattu 20.03.2024.

<https://control.com/technical-articles/understanding-the-basics-of-pulse-width-modulation-pwm/>

Hirzel, T. 2024. Basics of PWM (Pulse Width Modulation). Arduino. Verkkosivu. Viitattu 07.05.2024. <https://docs.arduino.cc/learn/microcontrollers/analog-output/>

Texas Instruments. 2019. Optocouplers and Silicon-based Galvanic Isolation Technology – How Do They Work? Verkkosivu. Viitattu 27.02.2024 <https://www.ti.com/document-viewer/lit/html/SSZT391>

Arduino. nd. Software. Verkkosivu. Viitattu 26.02.2024. <https://www.arduino.cc/en/software>

ATSG. 2009. Mercedes, Jaguar, Daimler/Chrysler “722.6” “5 speed” (Nag1). ATSG. 10.2009. Viitattu xx.x.2024. Vaatii käyttöoikeuden. [https://atsgbookstore.com/mercedes-722-6-nag-1\\_dl/](https://atsgbookstore.com/mercedes-722-6-nag-1_dl/)

Afzal, S. 2016. I2C Primer: What is I2C? (Part 1). Analog Devices. Verkkosivu. Viitattu 29.02.2024. <https://www.analog.com/en/resources/technical-articles/i2c-primer-what-is-i2c-part-1.html>

Lutkevich, B. 2019. microcontroller (MCU). Techtarger. Verkkosivu. Viitattu 01.02.2024. <https://www.techtarger.com/iotagenda/definition/microcontroller>

Balogh, L. 2018. Fundamentals of MOSFET and IGBT Gate Driver Circuits. Texas Instruments. Verkkosivu. Viitattu 27.02.2024. <https://www.ti.com/lit/pdf/slva618>

Broadcom. 2023. HCNR200 and HCNR201 High-Linearity Analog Optocouplers Data Sheet. Verkkosivu. Viitattu 27.02.2024. <https://docs.broadcom.com/doc/HCNR200-HCNR201-High-Linearity-Analog-Optocouplers-DS>

CiA. nd. History of CAN technology. Verkkosivu. Viitattu 29.02.2024. <https://www.can-cia.org/can-knowledge/can/can-history/>

Emerson. 2023. CAN Bus Topology and Termination. Verkkosivu. Viitattu 29.02.2024. <https://www.ni.com/docs/en-US/bundle/sbrio-9627-feature/page/can-bus-topology-termination.html>

Sheldon, R. 2023. serial peripheral interface (SPI). TechTarget. Verkkosivu. Viitattu 29.02.2024. <https://www.techtarget.com/whatis/definition/serial-peripheral-interface-SPI>

Atmel. 2015. ATmega16U4/ATmega32U4. Verkkosivu. Viitattu 07.05.2024.  
[https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7766-8-bit-AVR-ATmega16U4-32U4\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7766-8-bit-AVR-ATmega16U4-32U4_Datasheet.pdf)

Texas Instruments. nd. CSD18511KCS. Verkkosivu. Viitattu 07.05.2024  
<https://www.ti.com/product/CSD18511KCS>

Rouse, M. 2023. DataBus. Techopedia. Verkkosivu. Viitattu 06.06.2024  
<https://www.techopedia.com/definition/6733/data-bus>

Microchip. nd. MCP2515. Verkkosivu. Viitattu 09.06.2024  
<https://www.microchip.com/en-us/product/mcp2515>

NXP. 2003. TJA1050 High speed CAN transceiver. Datalehti. Viitattu 23.7.2024.  
<https://www.nxp.com/products/interfaces/can-transceivers/legacy-can/high-speed-can-transceiver:TJA1050>

Infineon. 2003. IRF540N. Datalehti. Viitattu 08.8.2024.  
<https://www.infineon.com/cms/en/product/power/mosfet/n-channel/irf540n/>

Gillis, A. 2024. What is an embedded system?. TechTarget, Verkkosivu. Viitattu 06.10.2024.  
<https://www.techtarget.com/iotagenda/definition/embedded-system>

SWATHI, P. 2019. What is an Embedded System. Verkkosivu. Viitattu 06.10.2024.  
<https://swathiprabhala.blogspot.com/2019/02/embedded-systems.html>

Mouser Electronics. Nd. Datalehti. Viitattu 11.11.2024  
[https://www.mouser.com/pdfdocs/te-microchip-automotive-grade-components.pdf?srltid=AfmBOor916RloTXtTJcJG-dF9RFK4F-uxiU4AfJR0eYEUVSVW\\_k2ocU1](https://www.mouser.com/pdfdocs/te-microchip-automotive-grade-components.pdf?srltid=AfmBOor916RloTXtTJcJG-dF9RFK4F-uxiU4AfJR0eYEUVSVW_k2ocU1)

Storey, N. 2017. Electronics: A Systems Approach. 6. uud. painos. Pearson Education. Viitattu 17.11.2024

Ahonen, J., Haikio, T., Salonen, K. 2022. Elektroniikka. 3. uud. painos. Sanoma PRO. Viitattu 17.11.2024

Jokinen, T. 1991. Tuotekehitys. 2. uud. painos. Otatieto. Viitattu 18.11.2024

Zhu, Y. 2010. CAN and FPGA communication engineering : implementation of a CAN bus based measurement system on an FPGA development kit. 1. uud. painos. Diplomica Verlag. Viitattu 18.11.2024

Laplante, P., Kassab, M. 2022. Requirements engineering for software and systems. 4. uud. painos. CRC Press. Viitattu 18.11.2024

Yin, Y., Jiang, B. 2024. Embedded Software System Testing: Automatic Testing Solution Based on Formal Method. 1. uud. painos. CRC Press. Viitattu 18.11.2024

Labrosse, J., Perrin, B., Ganssle, J., Oshana, R., Walls, C., Curtis K., Adrews, J., Katz, D., Gentile, R., Hyder, K. 2007. Embedded Software. 1. uud. painos. Elsevier Science & Technology. Viitattu 19.11.2024

## LIITTEET

Liite 1. Vaihteiston ohjainlaitteen arkkitehtuuri.

