



Tuomo Kupari

Lämpötila-anturi häiriöisessä tilassa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja automaatiotekniikka

Insinöörityö

2.12.2023

Tiivistelmä

Tekijä: Tuomo Kupari
Otsikko: Lämpötila-anturi häiriöisessä tilassa
Sivumäärä: 30 sivua + 4 liitettä
Aika: 2.12.2023

Tutkinto: Insinööri (AMK)
Tutkinto-ohjelma: Sähkö- ja automaatiotekniikka
Ammatillinen pääaine: Elektroniikka
Ohjaajat: Lehtori Juha Kallunki
Insinööri Heikki Saari

Virtapiireihin syntyy häiriöitä. Korkeatehoiset laitteet synnyttävät paljon häiriöitä ympäristöönsä ja sisäisiin osiin, kuten lämpötila-antureihin. Laitteen lämpötiloja seuraavat anturit pitää valita huolellisesti arvioiden erilaisia lämpötilan mittaustapoja anturien luotettavuuden kannalta. Laitteen tulee toimia luotettavasti käyttöympäristössään. Tämän työn tarkoituksena oli selvittää työhön valikoituneen lämpötila-anturin häiriönsietokykyä.

Tässä tutkimustyössä tehdään katsaus työn laitteen lääketieteellisen standardin vaatimuksiin ja kirjallisuuskatsaus häiriön lähteisiin sekä poistoon. Digitaalista lämpötila-anturia testataan havainnoiden häiriöisessä tilassa ja yritetään arvioida sen toimivuutta uuden laitteen kehittämiseen.

Työhön kerätyn aineiston ja havainnoinnin pohjalta valikoitunut anturi ei osoittautunut luotettavaksi käyttökohteessaan ilman lisäkomponentteja. Toisen anturin valinta tulee olemaan luotettavampi ja laskemaan laitteen tuotannon kustannuksia.

Avainsanat: lämpötila-anturit, EMC, EMI

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

Abstract

Author: Tuomo Kupari
Title: Temperature sensor in noisy environment
Number of Pages: 30 pages + 4 appendices
Date: 02 December 2023

Degree: Bachelor of Engineering
Degree Programme: Electrical and Automation Engineering
Professional Major: Electronics
Supervisors: Juha Kallunki, Senior Lecturer
Heikki Saari, Engineer

Electrical circuits have noise. High power devices will radiate lots of noise to their environment and parts, such as temperature sensors. These sensors need to be chosen with care, considering the most fitting and reliable temperature measurement method. The device must work reliably in its environment. The purpose of the work was to test reliability of a sensor in a noisy environment.

This thesis takes a look at medical standard requirements of the studied device and clarifies causes of electromagnetic interference and its avoidance. A chosen digital thermal sensor was tested in the work.

As result, information was obtained on the subject. The sensor tested in the thesis was not reliable for the device without additions. Another type of sensor would be preferable and may decrease manufacturing costs.

Keywords: Temperature sensor, EMC, EMI

Sisällys

Lyhenteet

1	Johdanto	1
2	Sähköisten lääketieteellisten laitteiden IEC 60601 -standardi	1
3	Häiriön lähteitä	3
4	RFI-häiriöiden testaaminen	6
5	Anturien valinta	7
5.1	Ensimmäinen TMP126 testi	9
5.2	Toinen TMP126 testi	14
5.3	Kolmas TMP126 testi	16
6	Häiriöiden poisto	17
7	CRC	25
8	TMP126 testaaminen STM32 mikrokontrollerilla	26
9	SPI sarjaväylän kaapelin pituus	27
10	Yhteenveto	31
	Lähteet	32

Liitteet

Liite 1: Lämpötila-anturien taulukointi

Liite 2: TMP126 Arduino IDE -koodi 1

Liite 3: TMP126 Arduino IDE -koodi 2

Liite 4: TMP126 STM32 -koodi

Lyhenteet

- CRC: *Cyclic Redundancy Check*. Datapaketin loppuun lisätty tarkastussumma, joka lasketaan lähetetystä viestistä ja jolla voidaan verifioida lähetetyn viestin virheettömyys.
- I²C: *Inter-Integrated Circuit*. Puolidupleksinen, synkroninen protokolla, jossa on kaksi johdinta ja joka toimii isäntä-orja-ohjauksella. Yksi johtimista, SDA, on datalinja, ja toinen johdin, SCL, on kellolinja. Väylässä voi olla useampi orjalaite, joilla on omat osoitteensa.
- IDE: *Integrated Development Environment*. Ohjelmointiympäristö, ohjelma, jolla käyttäjä voi kirjoittaa tai oikolukea koodiaan tai kääntää sen ohjelmatiedostoksi. Tässä työssä käytettiin muun muassa Arduino IDE -ohjelmointiympäristöä, jolla koodaaminen pohjautuu C-kieleen.
- IEC: *International Electrotechnical Commission*. Maailmanlaajuinen standarditehtäväorganisaatio. Sen tehtävänä on edistää kansainvälistä yhteistyötä kaikissa sähköisen ja elektroniikan alan standardien kysymyksissä toimien yleismaailmallisena referenssinä.
- SPI: *Serial Peripheral Interface*. Täysidupleksinen, synkroninen protokolla, jossa on neljä johdinta ja joka toimii isäntä-orja-ohjauksella. Johtimien CKL tai SCK on sarjakello. Isäntälaitte valitsee SS- tai CS-johtimella orjalaitteen. Kaksi muuta johdinta ovat MISO ja MOSI: MISO on master-laitteen sisääntulo ja MOSI on ulostulo.

1 Johdanto

Laitteiden lämpötilojen seuraaminen antaa arvokasta tietoa, jota voidaan käyttää hyväksi laitteen kunnan seuraamiseksi ja huollon tarpeen arvioimiseksi ennalta ehkäisevästi tai laitteen vioittuessa. Erilaiset ylikuumentumissuojaukset ehkäisevät laitteen ennenaikaiselta rikkoutumiselta ja parantavat käyttöturvallisuutta. Korkeajännitteisten laitteiden piirit ovat itsessään häiriöisiä ja säteilevät elektromagneettista säteilyä lähikenttäänsä. Piiriin valitut vääränlaiset komponentit saattavat häiriintyä, lähettää vääränlaista tietoa, sekoittaa digitaaliset tietoväylät ja johtaa lopulta laitteen vioittumiseen.

International Electric Company Oy valmistaa korkeatehoisia vahvistimia ja magneetin teholähteitä. Tässä työssä käydään läpi laitteelle vaadittavaa lääketieteellistä standardia, häiriöiden lähteitä sekä niiden poistoa, pohditaan, mikä olisi vahvistimelle sopiva lämpötila-anturi ja testataan työhön valikoitua anturia. Työn päämääränä on valita vahvistimelle sopiva anturi, testata sitä ja määrittää, voidaananko sitä käyttää uudenlaisen nestejäähdytetyn vahvistimen kehittämiseen.

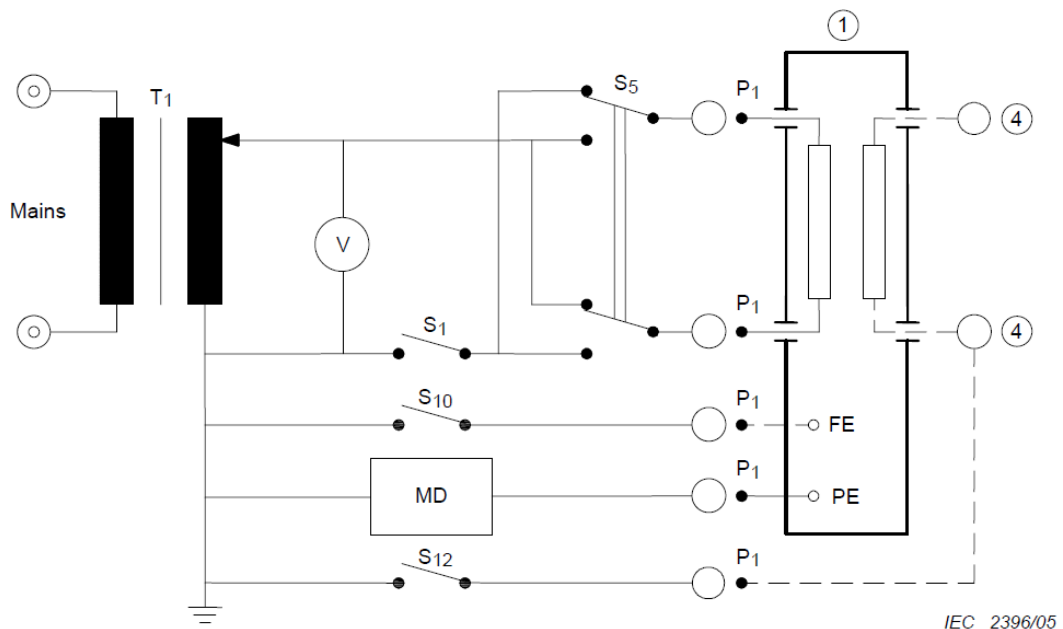
2 Sähköisten lääketieteellisten laitteiden IEC 60601 -standardi

Sähköisten laitteiden pitää olla turvallisia käyttäjilleen. Turvallisuutta tuo asetettujen standardien noudattaminen. Näitä standardeja luo ja valvoo International Electrotechnical Commission (IEC). Uusin versio sähkökäyttöisten lääkintälaitteiden turvavallisuusvaatimuksista on tämän työn aikana IEC 60601-1:2020 painos 3.2. Ensimmäinen osa sisältää yleisen turvallisuuden ja oleellisen toiminnan määrittämiä. Toinen osa IEC 60601-1-2 käsittää tarkemmin elektromagneettista yhteensopivuutta, sen vaatimuksia ja testausta. (1.)

Yleistäen lääketieteellisillä laitteilla tulee olla kaksi eristävää suojausta, Means of Protection (MOP), jotta estetään virran vuotamisen normaalissa käytössä kosketeltavissa olevilta osilta (accessible part) ja kontaktia tarvitsevilta laitteen toiminnan kannalta oleellisilta osilta (applied part). Suojaukset kategorisoidaan

korkeampaan potilasta suojaavaan eristykseen (MOPP) ja matalampaan käyttäjää suojaavaan eristykseen (MOOP). Näiden hiipimisetäisyyden ja eristyksen vaatimukset ovat erilaiset. Eristyksenä voivat toimia MOPP- tai MOOP-eristyksen vaatimukset täyttävä kiinteä eriste, hiipimisetäisyys, ilmaeriste tai eristävät komponentit. Eristeen pitää kestää laitteen käyttöiän mukaan mekaanista kulumista sekä lämmön ja kosteuden aiheuttamaa kulumista. (1.)

Toissijainen piiri tulee eristää syöttöpiiristä vähintään yhdellä suojauksella (MOP), kuten kuvassa 1 käytetään muuntajaa T_1 (1). Toissijaisen piirin tehollähteenä voi myös toimia akku.



Kuva 1. Yleisissä vaatimuksissa on annettu esimerkki syöttöpiirin mittaamisesta. Kytkimien S_5 , S_{10} ja S_{12} eri kombinaatiot mitataan S_1 normaalissa, suljetussa tilassa ja S_1 vikatilassa kytkimen ollessa avoin. (1.)

Ankarammin eristettyjen F-tyyppisten osien tulee olla eristetty kaikista muista osista yhden potilasta suojaavan eristyksen verran (MOPP). Tällöin virtaa, joka olisi suurempi kuin vikatapauksen sallittu potilasvuotovirta, ei saa kulkea potilaaseen liittimen ja maan välillä, kun ulkoisen lähteen ei-tarkoituksellinen jännite kytketään potilaaseen. F-tyyppiset osat jaotellaan BF- tai CF-tyypin liityntäosiin. Sydämenläheiseen soveltuvien liityntäosien tulee olla CF-tyyppiä (1.)

B-tyyppisen liityntäosan (applied part) tulee erottaa potilasta suojaavan eristyksen (MOPP) verran maadoittamattomista metallisista kosketeltavista osista (accessible part). Poikkeuksia ovat tilanteet, joissa kosketeltava osa on fyysisesti kosketuksissa käytön kohteisen osan kanssa, jolloin osat määritellään yhdeksi samaksi osaksi. Toinen mahdollinen poikkeus on, että metallisen kosketeltavan osan mahdollisuus osua jännitelähteen tai vuotovirran kanssa on merkittävän pieni. B-tyypin laitteen ei-toivottu liittyminen maahan voi vioittaa laitteen. B-tyyppisen liityntäosan potilaan puoleinen piiri voi olla liitetty suojamaahan. (1.)

3 Häiriön lähteitä

Työn kohteena olevassa korkeajännitteisestä vahvistimesta, jolla tuotetaan korkea-amplitudisia virran ja jännitteen muutoksia, syntyy sähkökentän muutoksia. Maxwellin yhtälön mukaan muuttuvasta sähkökentästä seuraa muuttuva magneettikenttä, mikä toimii myös päinvastoin. Nämä muuttuvat kentät saattavat häiritä ympäristössä olevia sähköisiä laitteita, mitä kutsutaan sähkömagneettiseksi häiriöksi (EMI, electromagnetic interference). Uutta vahvistinta suunniteltaessa pitää ottaa huomioon sen kaikkien järjestelmien kyky toimia tässä häiriöisessä tilassaan eli vahvistimen sähkömagneettinen yhteensopivuus (EMC, electromagnetic compatibility). Sähkömagneettiseen yhteensopivuuteen kuuluu myös laitteen elektromagneettisen säteilyn hallinta, jottei laite säteile elektromagneettisesti häiritsevästi ympäristöönsä.

Kohinan voi kategorisoida kolmeen alueeseen: luontaisesti syntyvään, satunnaisten fyysisten systeemien heiluntaan, kuten lämpötilakohina, ihmisen tekemään kohinaan muun muassa moottorit, kytkimet, tietokoneet, ja luonnonilmiöihin, kuten salamet sekä auringon energeettiset purkaukset. Kohinaa ei voida poistaa, mutta sen aiheuttamaa häiriötä voidaan vaimentaa, kunnes se ei enää aiheuta häiriötä piirin toimivuuteen. (2.)

Itsestään selvä syy kohinan piiriin yhdistymiseen on johdin. Johto, joka kulkee häiriöisen ympäristön läpi, johtaa häiriötä toiseen piiriin. Ratkaisu tähän on estää häiriön pääseminen johtimeen tai häiriön suodattaminen. Toinen yleinen

tapaus on, kun kaksi piiriä jakaa yhteisen impedanssin. Tällöin molempien piirien virrat aiheuttavat häiriöitä toisen piirin toimintaan, mitä kutsutaan galvaaniseksi kytkeytymiseksi. Tämä voi olla yhteismuotoista tai eromuotoista virtaa. Kahden johtimen väliin voi syntyä kapasitiivista johtamista, kun niiden välillä on riittävän suuri potentiaalinen ero. Tätä voidaan ehkäistä johtimien suojaamisella tai erottamalla ne kauemmaksi toisistaan. Yhteismuotoinen virta ei ole välttämättä haitallista, koska se ei aiheuta piiriin transienttipiikkejä. Yhteinen maadoitus tai virtalähde voivat olla galvaanisen kytkeytymisen syytä. Kohinan jännite on suoraan suhteessa häiriön lähteen taajuuteen. (2.)

Materiaalien huonot tai epäpuhtaat liitoskohdat voivat saada aikaan purskekohinaa, jota kutsutaan myös popcornkohinaksi. Purskekohina alun perin havaittiin puolijohdediodeissa, ja sitä ilmenee myös integroiduissa piireissä. Jos purskekohinaa vahvistetaan kaiuttimeen, se kuulostaa aivan kuin popcornin pamahtelulta. Purskekohina esiintyy lyhyinä, epäsäännöllisinä sarjoina mikrosekunneista sekunteihin ja saa aikaa vähäisiä tason muutoksia. Purskekohinan voimakkuus on suurimmillaan piireissä, joissa on korkea impedanssi, koska purskekohina on virtaan liittyvä ilmiö. Koska purskekohina johtuu valmistuksen virheistä, voi sitä poistaa parantamalla valmistusmenetelmiä. (2.)

Kaikki vastukset tuottavat jännitekohinaa, joka voi olla termistä kohinaa tai muuta kohinaa, kuten raekohinaa tai kontaktien välistä kohinaa. Kaikkia näitä kohinoita pystytään vähentämään tai eliminoimaan paitsi termistä kohinaa. Suurempaa tehoa kestävillä vastuksilla esiintyy vähemmän kohinaa. Reaktanssista ei generoidu termistä kohinaa. (2.)

Terminen kohina, jota kutsutaan resistanssiksi tai Johnson-kohinaksi, syntyy resistanssin elektronien liikkeistä. Se asettaa piirin kohinan alimmaistason. John B. Johnson kertoo vuoden 1927 julkaisussaan (3, s. 50) löydöstään, että epäsäännöllistä jännitettä esiintyy kaikissa johtimissa ja että se on verrannollinen lämpötilaan. Harry Nyquist täydensi tätä matemaattisesti käyttäen termodynamiasta päättelyä ja loi kaavan

$$V_t = \sqrt{4KTBR} \quad (1)$$

jolla voidaan laskea resistanssin aiheuttama, avoimen piirin tehollisarvojäännitteen kohina. K on Boltzmannin vakio, T absoluuttinen lämpötila, B kohinan kaistanleveys ja R resistanssi. Termistä kohinaa esiintyy kaikissa resistanssisissa elementeissä. Kohinaa pystytään vähentämään pienentämällä resistanssia, taajuutta tai käyttölämpötilaa. (2.)

Raekohinaa syntyy, kun virta kulkee potentiaalisen eron eli kahden eri jännitetason yli. Tällöin virran nopeus heilahtelee, mikä johtuu elektrodien epäsäännöllisestä emissiosta. Raekohinaa ilmenee yleensä puolijohteissa ja elektroniputkissa. (2.)

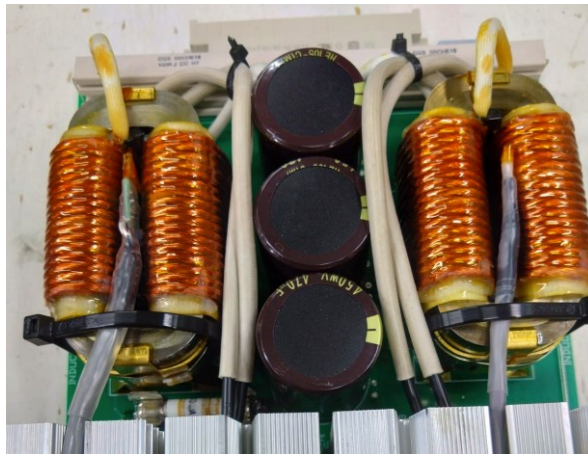
Eri materiaalista valmistettujen ja niiden välille tehtyjen epäihanteellisten kontaktien välille voi syntyä kohinaa, jota kutsutaan $1/f$ -kohinaksi tai pinkiksi kohinaksi. Se on tyypillistä matalilla taajuuksilla, ja sen voimakkuuden vaihtelu seuraa Gaussin käyrää. Sitä esiintyy kytkimissä, releissä, transistoreissa, diodeissa, hiilimassavastuksista ja hiilimikrofoneissa. (2.)

Valkoiseksi kohinaksi kutsutaan kohinaa, jolla on yhtenäinen tehon jakautuminen taajuuteen nähden, mikä viittaa, että se koostuu monesta eri taajuuden komponentista. Tämä on tyypillistä termiselle kohinalle, mutta monella muulla kohinan lähteellä on samanlaista kohinaa, jota voidaan myös kutsua valkoiseksi kohinaksi. (2.)

Kaikki piirin osat säteilevät elektromagneettisia kenttiä aina, kun sähköinen lataus purkautuu. Elektromagneettista säteilyä vastaan rakennetut suojaukset pitää maadoittaa johtimen molemmista päistä. Tällöin suojassa kulkee virtaa, joka täytyy huomioida. Suojuksista ei tosin ole hyötyä matalataajuisia häiriöitä vastaan. (2.) EMC-suorituskykyyn vaikuttavat parametrit, joita tulee suunnittelussa tarkkailla, ovat taajuus, amplitudi, aika, impedanssi ja mittasuhteet (4).

4 RFI-häiriöiden testaaminen

Lämpötiloja tulisi mitata uudessa vahvistimesta lähtökuristimista, lähdön RC-häiriönsuotopiiristä, pääteasteesta, jäähdytysnesteestä, ottoilman lämpötilasta sekä ilmankosteudesta. Tässä työssä paneudutaan erään kortin lähtökuristimen lämpötilan mittaukseen. Lämpötila-anturia PT100 ja työhön valittua lämpötila-anturia TMP126 aseteltiin ensimmäiseen testiin yhtä aikaa kuvassa 2.



Kuva 2. Testissä käytetty vahvistinkortti, johon molemmat anturit sijoitettiin.

Radiotaajuushäiriömittauksissa (radio frequency interference, RFI) testataan laitteen sietoa ulkopuolisille häiriöille, mikä on tavallaan tämän työn tarkastelun kohteena. Radiotaajuushäiriö usein yhdistetään sähkömagneettisen yhteensopivuuden (EMC) yhdeksi tekijäksi. Euroopan unionin sietokykyjen standardi IEC EN 61000-6-1 (5) määrittää kolme kriteeriä. Kriteeri A:n mukaan laitteen tulisi toimia tarkoituksen mukaan testauksen aikana ja jälkeen. Tämä standardin kriteeri määrittää laitteen hyväksytyyn RF-sietokyvyn. Radiotaajuushäiriöittäuksia tulisi tehdä lähettämällä vahvistettua häiriösignaalia testattavaan laitteeseen (DUT) eristetyssä huoneessa, jotteivat häiriösignaalit pääsisi ympäröivään alueeseen. Kriittisimmässä tilassa ovat sisään- ja ulostulokaapelien läheisyydessä sijaitsevat matalajännitteiset piirit. Radiotaajuushäiriömittauksia on kallis ja vaikea toteuttaa. (2.)

Radiotaajuuksien vastustuskykyä voi kuitenkin testata alustavasti sijoittamalla häiriösignaalin lähteen piirin kriittisiin kohtiin ja seurata piirin toimivuutta. Yksi tapa on käyttää pientä harjamoottoria, kuten Dremel-työkalua yhden tuuman päässä testattavasta laitteesta (2). Tämän työn testattavan anturin sijoittaminen ulostulokuristimeen tulee vastaamaan eräänlaista vastaavaa testiä.

5 Anturien valinta

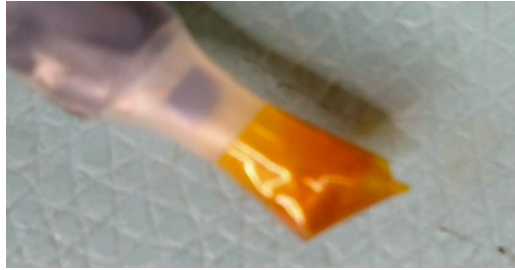
Anturien valinnasta kerättiin vertailua varten taulukko (liite 1) ja vertailtiin digitaalisia ja analogisia antureita. Analogisissa antureissa etuna on niiden häiriönsietokyky verrattuna digitaalisiin antureihin. Toisaalta piirin impedanssin vaihtelu monimutkaistaa piirisuunnittelua sekä tarvittavat lisäkomponentit lisäävät lopputuotteen tuotantokustannuksia.

Digitaalisten anturien valinnassa vaikuttaa niiden tietoväylän häiriönsietokyky. Häiriöinen ympäristö saattaa korruptoida tietoliikenteen kokonaan. Piirille rakennetut suodattimet saattavat häiritä piirin mittauskapasitanssia ja resistanssia.

Analogisissa antureissa on usein hankalaa niiden epälineaarisuus ja mahdolliset yksilölliset erot tuotantoprosessissa. Epälineaarisen käyrän lämpötilan määrittämisestä voi seurata monimutkainen kaava, joka voi olla hankala ratkaista, jos halutaan seurata kansainvälistä 1990- lämpötila-asteikkoa (ITS-90 standardi). Tosin piirisuunnittelulla ja lisäosilla lineaarisuutta voidaan parantaa. PT100-lämpötila-anturin lämpötila voidaan ratkaista Callendar-Van Dusen kaavalla

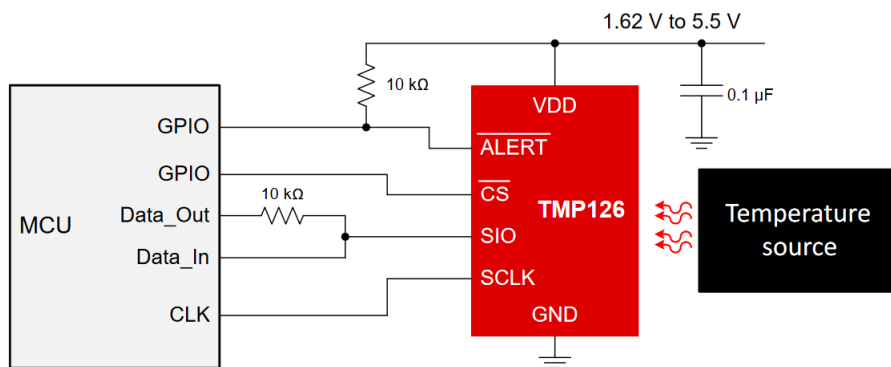
$$R_t = R_0(1 + At + Bt^2 + C(t - 100)t^3) \quad (2)$$

jossa R_t on sen hetkinen lämpötila, R_0 on resistanssi nollassa celsiusasteessa ja A, B sekä C ovat vakioita (6). PT100-lämpötila-anturia, kuvassa 3, käytettiin vertailukohtana analogisena lämpötila-anturina.



Kuva 3. Lämpötila-anturia PT100 käytettiin testin vertailukohtana.

Työhön valikoitui Texas Instrumentsin TMP126 anturi, joka käyttää SPI-protokollaa. SPI-protokollan etuna verrattuna I²C-protokollaan on CS-johdin tai SS-johdin, jolla voi valita haluamansa orjalaitteen. Sarjaliikenne, joka käyttää I²C-protokollaa, saattaa mennä kokonaan sekaisin, jos yksikin sarjaväylän laitteista häiriintyy. TMP126 anturilla on CRC tarkistussumma, jolla voi havaita tiedonsiirron virheitä. Anturia testattiin adapterikortilla ja Arduino Megalla kuvan 4 mukaisella kytkennällä.



Kuva 4. Datalehdessä julkaistu kytkentäkaavio (7).

Kuva 5 on ensimmäisessä testissä käytetystä kortista. Adapterikortti ei ollut täysin yhteensopiva, mutta sillä pääsi alkuun.



Kuva 5. Lämpötila-anturin TMP126 ensimmäinen testiin käytetty kortti.

Anturille tehtiin Arduino IDE -ohjelmalla koodi testaamista varten. Koodissa anturille lähetetään 16-bittisiä viestejä, jotka voi jakaa kahteen 8-bittiseen viestiin. Ensimmäisen viestin ensimmäisessä 8-bittisessä osassa annetaan komento, jolla voidaan enableoida CRC, määrittää datablokkien määrä CRC:tä varten, enableoida dataosoitteen automaattinen suureneminen ja anturin lukeminen tai kirjoittaminen. Jälkimmäiset 8-bittiä määrittävät alioisoitteen.

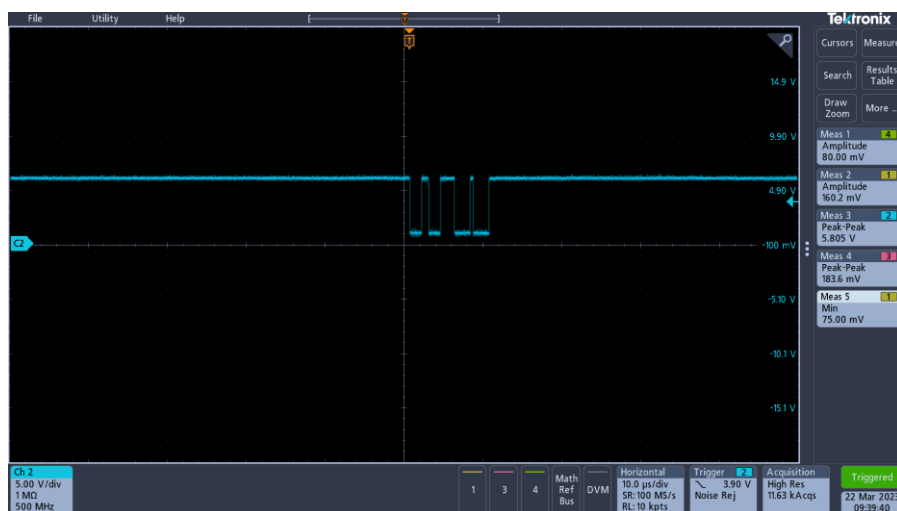
TMP126 lämpötila-arvo on luettavissa rekisteristä 00h. Yksinkertaisimmillaan tämän rekisterin voi lukea lähettämällä luku 01h. Tällöin koko 16-bittinen viesti on binäärisesti 0000 0001 0000 0000. TMP126 lähettää lämpötilan takaisin 16-bittisenä viestinä, joista kaksi viimeistä bittiä ovat nollia eivätkä tarkoita mitään. Toisin sanoen lämpötila-arvo lähetetään 14-bittisenä lukuna, joka muunnetaan desimaaliluvuksi. Tämä luku pitää vielä kertoa arvolla 0,03125, jolloin saadaan sen hetkinen lämpötila celsiusasteina. (7.)

5.1 Ensimmäinen TMP126 testi

Testissä (22.3.2023) yritettiin testata TMP126 häiriönsietokykyä käyttäen Arduino Mega kehityskorttia 9 VDC:n patterilla. SPI-väylän nopeus oli Arduinon vakionopeus eli 4 MHz. Anturi juotettiin valmiiksi jo löytyvään muunneltuun adapterikorttiin (RE914) ja suojattiin kirkkaalla W1H-9.0-X-UP kutistesukalla. Kutistesukan pää suojattiin Kapton-teipillä. Kaapelina käytettiin kolmesta kierreparillisesta johtimesta koostuvaa tietokonekaapelia Belden 9503. Testissä

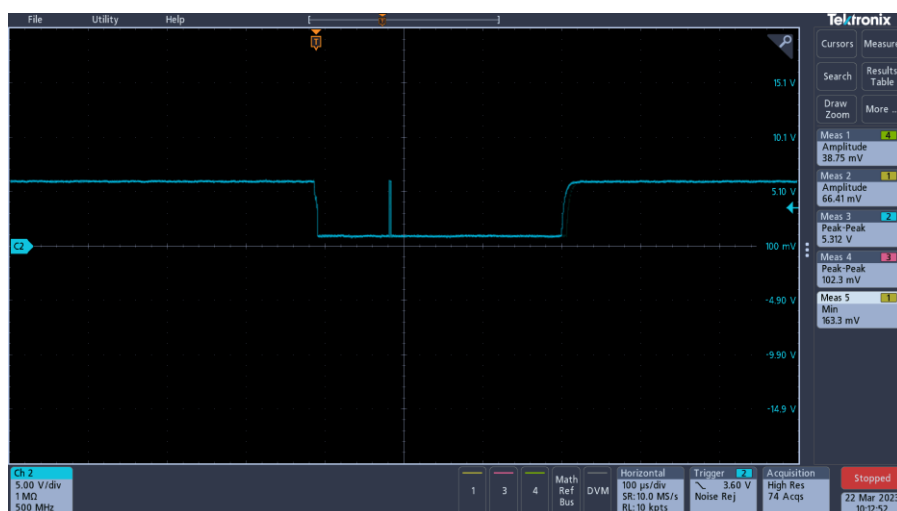
käytettiin toisen liitteen koodia. Kytchentäkaaviona käytettiin luvun 6 kuvaa 4 ilman ALERT-pinniä.

Anturi toimi täydellä testisignaallilla hetken aikaa, kun MOSI-pinnistä mitattiin differentiaali mittapäällä. Differentialimittapäät olivat väärinpäin, jolloin saatiin kuvan 6 ja 7 näköistä signaalia.



Kuva 6. MOSI-pinnistä mitattu Arduinon ohjaussignaali anturille. Differentiali-mittapäät ovat kytketty väärinpäin sattuneista syistä. Siksi signaali näkyy väärinpäin.

Kuvassa 7 näkyy lämpötila-anturilta tulevan signaalin pyöristyminen lopussa. Tämä viittaisi liian suureen kapasitanssiin johtimessa.



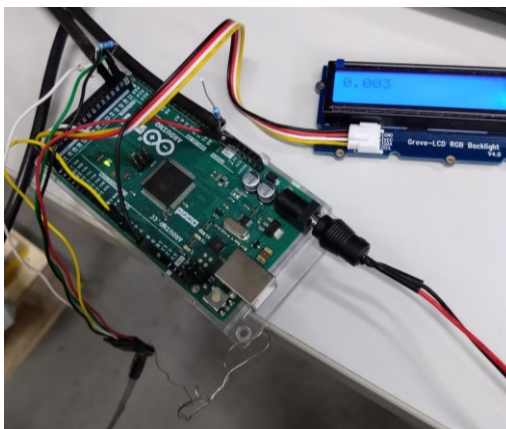
Kuva 7. MISO-pinnistä mitattu anturilta tuleva signaali, kun kaapelin suojaeriste on maadoitettu Arduinoon.

Anturi toimi paremmin, kun kaapelinsuojan yhdisti Arduinoon maahan, kuten kuvassa 8.



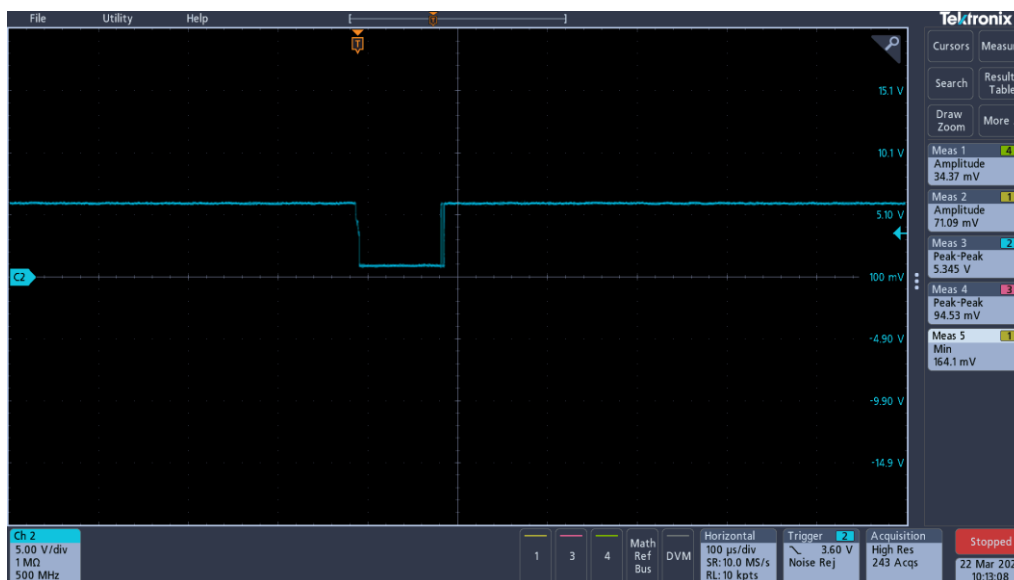
Kuva 8. Anturin kaapelinsuoja piti maadoittaa Arduinoon, jotta lämpötila oli luettavissa.

Muutoin lämpötila-anturi ei lähettänyt mitään, ja lämpötila näytti nolaa, kuten kuvassa 9.



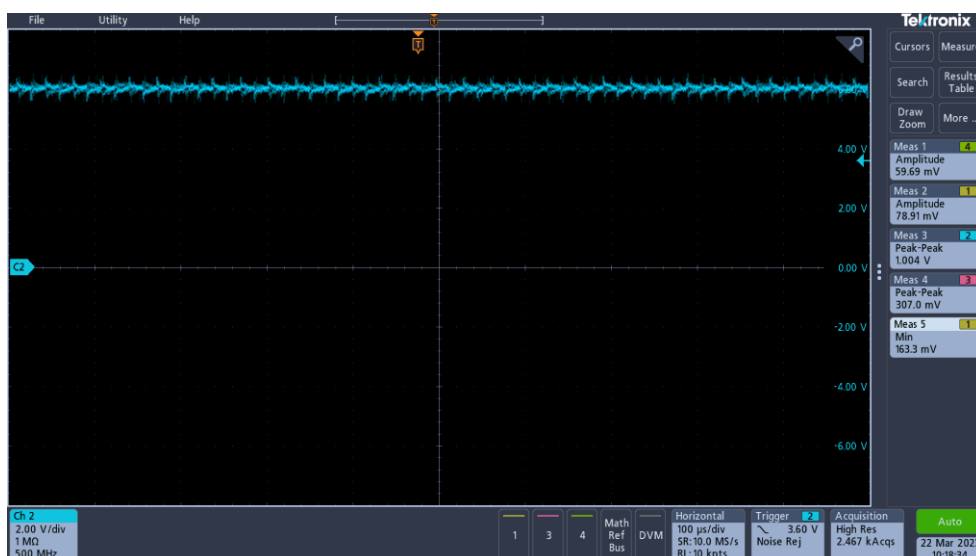
Kuva 9. Ilman kaapelinsuojan maadoitusta lämpötilasignaali häiriintyi ja näytti nolaa.

Oskilloskoopilla otettiin kuvat 10 ja 11. Kuvassa 10 kaapelinsuoja ei ollut maadoitettu, ja lämpötila-anturi näyttää nolaa astetta.



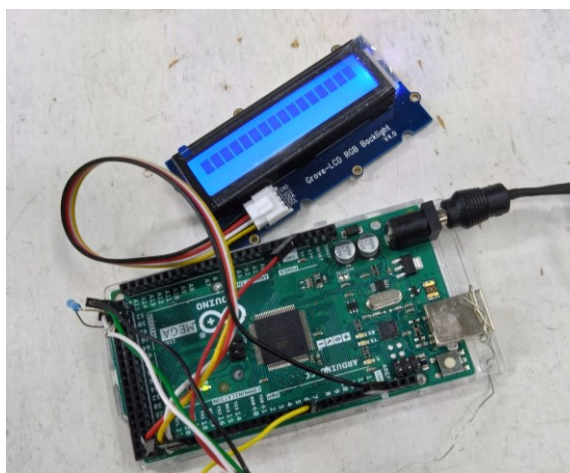
Kuva 10. Ilman kaapelinsuojan maadoitusta lämpötilasignaali ei ollut luettavissa MISO-pinnistä.

Kuvassa 11 Arduino sekosi kokonaan. Nollatasossa näkyi heiluntaa.



Kuva 11. MISO-pinnista mitattu signaali, kun järjestelmä häiriintyi.

Lämpötilan lukeminen onnistui vahvistin enableituna, kun Arduinon MISO-pinniä mittattiin differentiaalimittapäällä. Muutoin Arduinon yhdistetty LCD-näyttö mustui, kuten kuvassa 12.



Kuva 12. Ilman mittapäätä Arduinon yhdistetty näyttö häiriintyi.

Kuvassa 13 Arduino maadoitettiin kabinetin runkoon, jolloin näyttö toimi. Lämpötila-arvo jäättyi pian kuitenkin paikoilleen.

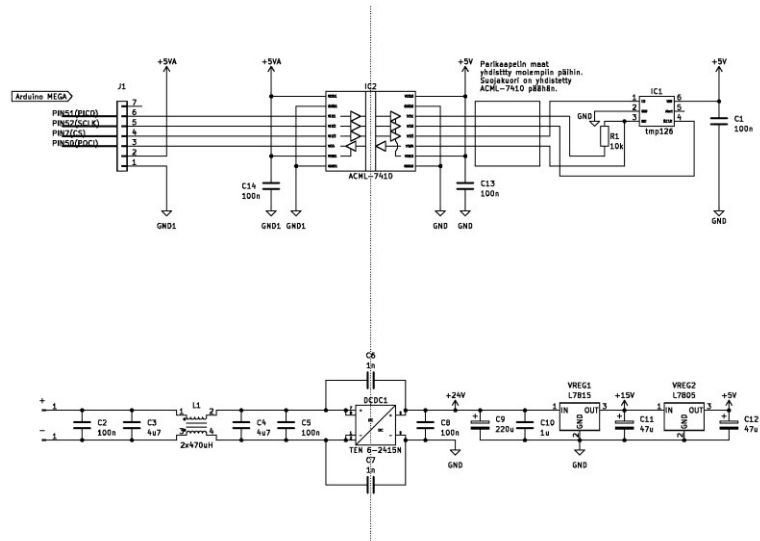


Kuva 13. Arduinon maadoitus runkoon sai näytön toimimaan, mutta lämpötila-arvo jäättyi paikoilleen.

Testin jälkeen kutistesukka avattiin. Todettiin, että anturi ei lähettänyt signaalia, kun anturia painoi tai kaapelia varovasti väänsi anturin puolelta. Muutenkin todettiin anturin prototyyppi epäluotettavaksi. Päätettiin tilata yhteensopivampaa SOT-23-6 adapterikorttia ja sijoittaa ohituskondensaattori lähemmäksi VCC-jalkaa.

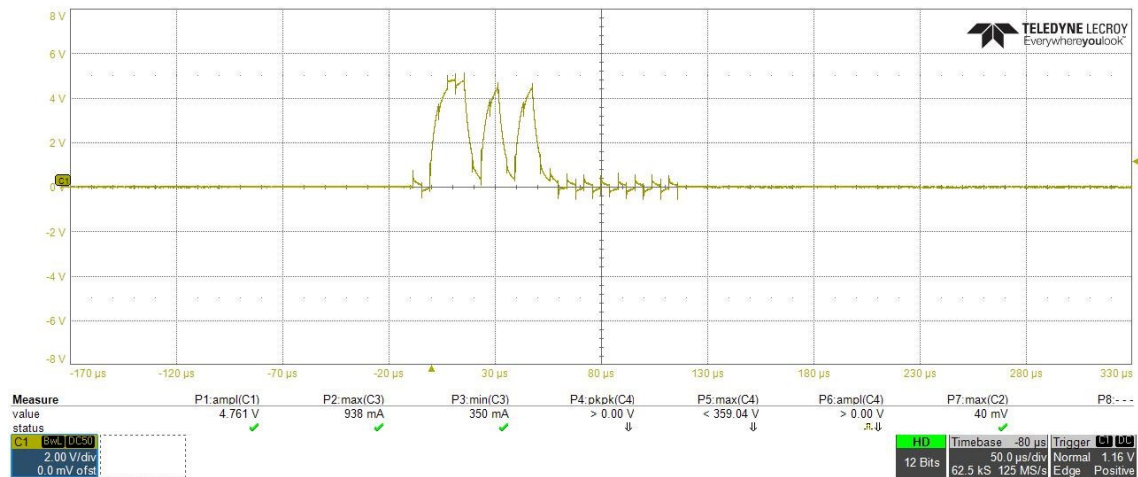
5.2 Toinen TMP126 testi

Tässä testissä (12.5.2023) erotettiin Arduino ja lämpötila-anturin piiri toisistaan, ettei Arduinon häiriintyminen vaikuttaisi lämpötila-anturin toimivuuden arviointiin. Arduino ja anturi eristettiin ACML-7410 CMOS -digitaalisella isolaattorilla ja eristävällä jännitteenmuuntimella. Testaukseen käytettiin hyväksi valmista piirilevyä, josta löytyivät kyseiset osat. Erona kuvan 14 piirikaavioon on, että testaukseen käytettiin kaapelia, joka sisälsi kolme parijohdinta. Yksi pari saattoi tällöin sisältää kaksi datajohdinta, jotka saattoivat häiritä toisiaan. Ohjelmointiin käytettiin paranneltua kolmannen liitteen koodia.



Kuva 14. Piirikaavio testauksessa käytetystä piiristä.

Arduino toimi moitteettomasti testin aikana. LCD-näytöllä ei tapahtunut mitään ylimääräistä. Anturi toimi noin muutaman kymmenen sekunnin ajan, kunnes se alkoi näyttämään nollaa. Huomattiin, että ohjaussignaali kymmenen kilo-ohmin vastuksen yli vääristyy (kuva 15). Signaalissa näkyy myös kelloväylän häiriö.



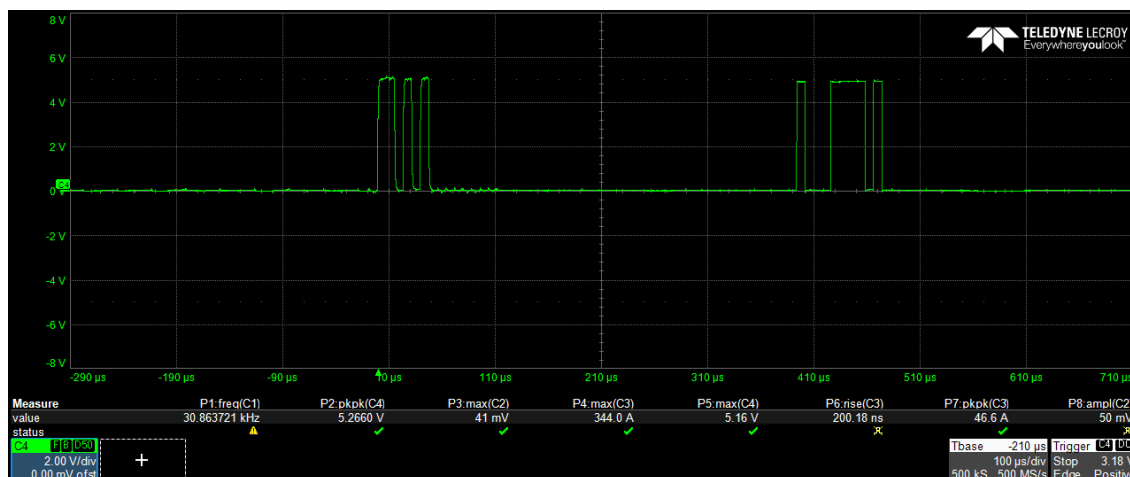
Kuva 15. Ohjaussignaali vääristyy kymmenen kilo-ohmin vastuksen yli.

5.3 Kolmas TMP126 testi

Tämän testin (23.5.2023) tarkoituksena oli parantaa edellisen testin ohjaussignaalia. Erona edelliseen testiin on, että tässä testissä käytettiin parikaapelia, joista jokaisen datalinjan pari yhdistettiin molempien päiden maahan. Luvun 5.2 kuvan 14 piirikaavion R1 pienennettiin yhteen kilo-ohmiin. Arduinon ohjelmointia ei muutettu vaan käytettiin yhä liitteen 3 koodia.

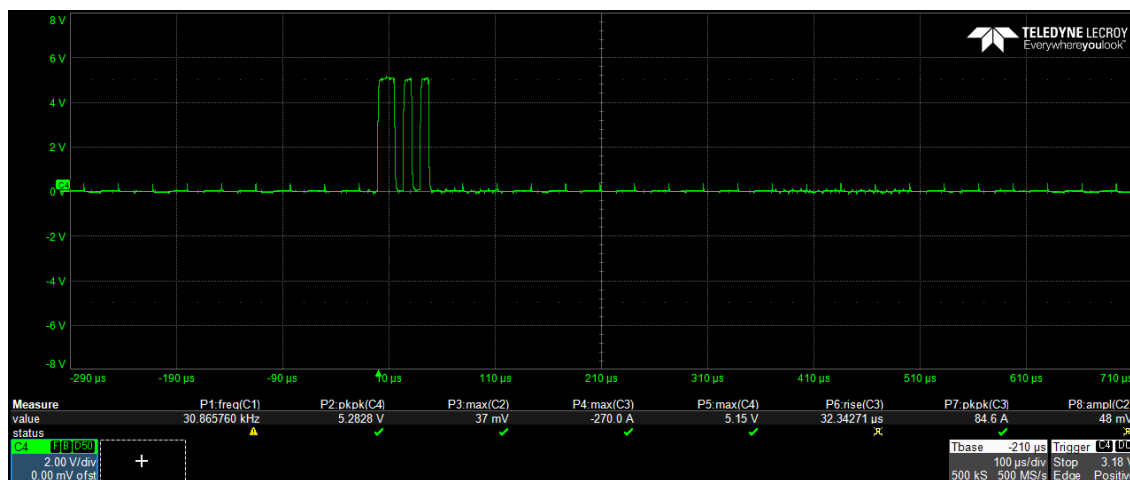
Tämä testi toimi parhaiten tähän mennessä. Anturi antoi lukemia, mutta satunnaisesti noin puolen minuutin välein näytti hetken aikaa nollaa. Käytettäessä virtalähteenä kahta yhdeksän voltin paristoa laboratoriovirtalähteen sijaan, anturi toimi testin aikana moitteettomasti. Yksi paristo toimi Arduinon virtalähteenä, ja toinen sijoitettiin +15 voltin linjaan. Tämä viittaisi, että tehollähteen linjassa aiheutuisi häiriöitä anturiin.

Datan pitää kulkea kahteen suuntaan, jotta lämpötila-arvo saadaan luettua. Binaarisen tiedon, joka on mikrosekuntien mittainen, pitäisi olla tulkittava kummassakin päässä. Lämpötila-anturilta palaava signaali näytti nyt paremmalta ilman pyöristymisiä (kuva 16).



Kuva 16. Ohjaussignaali ja anturin lähettämä vastaus.

Kun anturi ei lähetä mitään takaisin, voisi olettaa, että pyyntösignaali on korruptoitunut jo anturille saapuessaan, mutta se näkyi hyvin palaavassa signaalissa (kuva 17). Tämä viittaisi myös, että virransyötön piirissä saattaisi olla häiriöitä.



Kuva 17. Kuva tilanteesta, kun anturi näyttää nolaa.

6 Häiriöiden poisto

Tämän työn suurin ymmärtämys tuli sähkönsiirtymisen ajattelemisesta fysikaalisena ilmiönä. Piirin energia ei ole jännitteessä tai virrassa vaan sähkö- ja magneettikentissä. Nämä kentät eivät sijaitse piirien linjoissa tai kuparitasoissa vaan

niiden välissä. Sähkölinjojen energian siirtymistä kutsutaan elektromagneettiseksi aalloksi. Aallot kulkevat matalinta impedanssia pitkin – eli piirin kuparirei-
tit toimivat aallon ohjaajina. Energia kulkee piirin meno- ja paluureitin dielektri-
sessä tilassa. Signaalireittiin syntyvät kentät maa- ja tehoreittien välillä. (2.)

Digitaalisen piirin häiriöt tulevat useimmiten piiristä itsestään, maalinjan häiri-
östä, teholähteen häiriöstä, siirtolinjojen heijastuksista tai tietolinjojen ristivaiku-
tuksesta eli ylikuulumisesta (2). Tässä työssä lämpötila-anturi sijoitetaan niin lä-
helle häiriön lähde, että ulkoinen lähde vaikuttaa piiriin huomattavasti. Huomi-
oitavaa on, että kaikki piiriin lisätyt komponentit lisäävät vahvistimen lopullista
tuotannon kustannusta

Ferriittien lisääminen kaapelin signaalilinjoihin saattaisi vähentää heiluntaa. Ku-
vassa 17 signaalilinjaa alhaallaoloajan oskillointia voi verrata edelliseen kuvaan.
Ferriitit tarjoavat edullisen vaihtoehdon erottaa piiriin kohdistuvaa korkeataa-
juista resistanssia. Ferriitit vaimentavat parhaiten ei-toivuttuja signaaleja yli 10
MHz:n taajuudella ja toimivat pieninä vaihtovirran vastuksina lähes nollan ohmin
impedanssina noin 1 MHz taajuudella (2). Eli ferriittejä voi ajatella tavallaan
vaihtosähkön vastuksina, jotka vastustavat hieman tai ei juuri lainkaan matala-
taajuista vaihtosähköä tai tasasähköä.

Yksinkertainen tapa parantaa piirin toimivuutta olisi rakentaa Faradayn häkin ta-
painen, 360 asteen suoja. Tällöin pitää ratkaista, miten lämpötila johtaa suojan
läpi. Suojaa heikentävät mahdolliset aukot riippuen niiden lukumäärästä, taa-
juudesta ja aukkojen välimatkasta. Suojaa ei välttämättä tarvitse maadoittaa,
jolloin se voi olla missä vain potentiaalissa. Useissa tapauksissa suoja pitää yh-
distää piirin yhteiseen pisteeseen, mikä ehkäisee potentiaalisen eron suojan ja
piirin välissä. Jos laite käyttää vaihtovirtaa, laitteen suojamaadoitus toimii virran
reittinä laitteen vikaantuessa. Maadoitus ehkäisee myös staattisen sähkön va-
rautumista suojaan. (2.)

Kaapelinsuojan yhdistäminen piiriin maahan voi muuttaa suojan antenniksi. Yli
100 kHz:n taajuudella toimivien piirien kaapelin suojauksen tulee olla

yhtenäinen ja maadoitettu molemmista päistä laitteen runkoon, joka ei ole yhteydessä piiriin maahan. Häiriön virta kulkee suojaossa, koska kaapelin päissä on maapotentiaalinen ero. Pienet suojaamattomat pätkät kaapelinsuojan ja rungon välissä, niin sanotut siansaparot, ovat alttiita häiriöille yli 100 kHz taajuuden datalinjoissa.

Kierretyistä parikaapeleista on hyötyä magneettiseen kenttään kytkeytymistä vastaan mutta ei kapasitiivista sähkökenttää vastaan. Balansoimattomien kierrettyjen parikaapelien käytöstä ei ole hyötyä, elleivät linjat ole toistensa differentiaalisia pareja, ja kierrosten määrä pitää olla vähemmän kuin kahdeskymmenesosa niissä kulkevasta taajuudesta. Jos kaapelin molemmat terminaalit ovat balansoituja, on kiertämisestä hyötyä. Kierretyn parikaapelin signaalin tulee virrata yhtä lailla ja vastakkaiseen suuntaan. (2.) Kierrettyjen parikaapelien käyttäminen on luultavasti paras edullinen ratkaisu signaalien ja virtalähteiden yhdistämiseen (4).

Kaapelin molempien päiden maadoittaminen suojaa magneettista säteilyä vastaan, kun taas yhden pään maadoittaminen suojaa kapasitiivista kytkeytymistä vastaan. Ongelmia ei synny useista maadoituskohdista, jos maat ovat erotettuja. Jos kyseessä on yhteinen maa, voi piiriin syntyä potentiaalista maaeroa. Potentiaalinen maaero johtaa EMI-ongelmiin. Häiriön määrä voi vähentyä toisessa päässä kaapelia, mutta systeemien välinen häiriö kasvaa. (4.)

Usein kaapelien suojaus on tehty punoksesta, joka kestää paremmin ja on joustavampaa kuin yhtenäinen suoja. Punos kuitenkin suojaa 60–98 prosenttia vähemmän kuin yhtenäinen suoja. (4.)

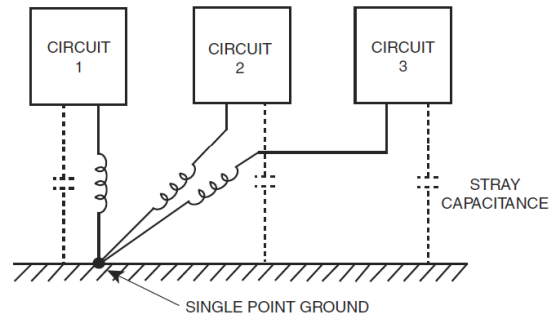
Piiriin toimintaa voi yrittää parantaa käyttämällä erilaisia filttäreitä ja erotuksia. Luvun 6.3 kuvassa 1 piirissä käytetty ACML-7410 eli digitaalinen isolaattori erotti Arduinon häiriöisestä piiristä, jolloin Arduinon puoleisessa piiriin toiminnassa ei todettu enää häiriöitä. Anturin puoleinen piiri säilyi kuitenkin vielä häiriöisenä vaikuttaen piiriin toimintaan. Piirejä voi erottaa myös käyttäen

optoerottimia, mutta ne toimisivat tässä piirissä liian hitaasti. Nämä erottimet toimivat myös suojaavina erottimina.

Lämpötila anturin SIO-datalinja ja kellolinja ovat alttiita toistensa ylikuulumiselle. Parempi piirilevysuunnittelu vähentäisi niiden keskeistä häirintää. Monikerroksinen piirilevyn suunnittelu parantaisi piirin toimintaa pienentämällä reittien impedanssia ja RF-vastustuskykyä. Signaalien paluureitin tulee tällöin seurata reittiä mahdollisimman läheltä, jotteivat datalinjat pariutuisi keskenään. Piirilevyn ulkopuoleisten kerroksien aallot levittyvät suuremmalle alueelle aiheuttaen enemmän ylikuulumista. (2.)

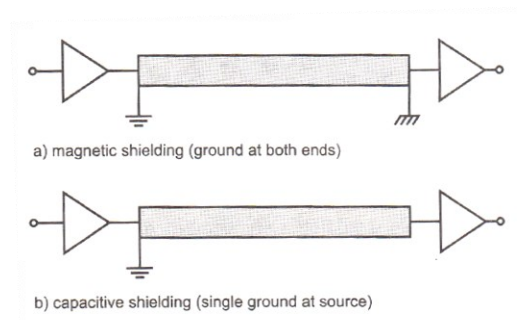
Maadoitukset jaetaan kahteen kategoriaan: suojaavaan maahan ja signaalimaahan. Jälkimmäistä ei välttämättä tarvitsisi kutsua maaksi vaan virran paluureitiksi, jonka voisi jakaa signaali- tai tehopaluuksi. Yleensä suojaavassa maassa ei kulje virtaa kuin vikatilanteissa. Jos maa on yhdistetty laitteen koteloon, kutsutaan sitä kotelomaaksi. Jos maa on taas yhdistetty vähäistä impedanssia pitkin maaperään, kutsutaan sitä suojamaaksi. Suojaava maa on yleensä yhdistetty suojamaahan, mutta se voi olla esimerkiksi kannettavassa tietokoneessa, autossa tai lentokoneessa yhdistettynä runkoon. Signaalimaa voi olla tai ei ole yhteydessä suojaavaan maahan, mutta usein niiden yhdistäminen johtaa EMC-ongelmiin. Useimmiten tilanteissa, joissa vaaditaan suojaavan maan yhdistämistä, pisteet eivät ole enää sopivia signaalimalle, mikä johtaa suunnittelun monimutkaistumiseen. Virta kulkee piirissä, joten suojamaahan kulkeva virta palaa toista reittiä takaisin. (2.)

Korkeilla taajuuksille (yli 100 kHz) yhden pisteen maata tulisi välttää, koska maajohtimet lisäävät maan impedanssia. Maajohtimet toimivat myös epätoivotuina antennina keräten ympäröivää energiaa ja säteillen ympäristöönsä. Ne tulisikin pitää lyhyempänä kuin kahdeskymmenesosa jaksonajasta. Liian pitkissä johtimissa virta harhailee matalinta impedanssia pitkin ei-toivotun kapasitanssin kautta (kuva 18). (2.)



Kuva 18. Liian pitkissä johtimissa, virta harhailee hajakapasitanssi kautta maahan (2).

Maasilmukat ovat harvoin ongelmallisia korkeilla taajuuksilla. Esimerkiksi kaapelin maadoittaminen molemmista päistä on tällöin suositeltavaa (kuva 19). Maasilmukat tuottavat ongelmia matalilla, alle 100 kHz taajuuksilla. Ongelmat assosioidaan herkkiin analogisiin piireihin kuten audio tai instrumentointi systeemien piirit. Maasilmukoiden ongelmat voidaan ratkaista käyttämällä yhtä maadoitusta tai maadoituksen hybridimallia. Hybridimaadoituksessa voidaan käyttää eriarvoisia kondensattoreita maadoituksen välissä, jolloin ne tarjoavat matalan impedanssin korkeilla taajuuksilla mutta erottavat matalilla taajuuksilla. Vaihtoehtoisesti, jos kondenssaattorit korvataan kuristimilla, ne toimivat maapisteinä matalilla taajuuksilla ja vastuksina taas korkeille taajuuksille. Maasilmukan huminaa voi myös parantaa pienentämällä maadoituksen impedanssia tai kasvattamalla kohinan marginaalia. Muita vaihtoehtoja ovat muuntajat, yhteismuotokuristimet ja optoerottimet. (2.)



Kuva 19. Kaapelin maadoituksen ero magneettista ja kapasitiivista kytkeytymistä vastaan (4).

Lattakaapelin käyttäminen on edullinen vaihtoehto useamman johtimen yhdistämiselle. Toinen niiden etu on, että johtimien asento ja orientaatio ovat muuttumattomia, joten tuotantoerien yksilöllinen vaihtelu on vähäistä. Lattakaapelin ongelmat riippuvat siitä, mitä linjaa kunkin johtimen tehtävä on johtaa. Jos kaapelissa on käytössä vain yksi maajohdin ja kaikki muut johtimet johtavat eri signaaleja, on vaarana isot johdinsilmukat, induktiivinen kytketympien ja ylikuuluminen. Ratkaisuna tähän on käyttää jokaisen signaalijohtimen kummallakin puolella maajohdinta. (2.)

Piirisuunnittelussa joskus nousee esille, pitäisikö analogisen ja digitaalisen piirin maat yhdistää vai erottaa. Analogisien ja digitaalisten piirien sekoittaminen johtaa myös niiden energioiden sekoittumiseen toisiinsa. Pääsääntöisesti piirilevy pitäisi suunnitella niin, että analogisten ja digitaalisten piirien komponentit tulisi erottaa omille alueilleen. Näiden maat olisivat kuitenkin pääsääntöisesti yhteydessä toisiinsa. Poikkeuksina voisi mainita, että jos piirit ovat liian lähellä toisiaan tai jokin piiri on hyvin herkkä häiriöille, on hyvin epätodennäköistä, että piirisuunnittelussa päädyttäisiin maiden erotukseen. Erotetut maat johtavat usein EMI-ongelmiin, koska erotetut maat voivat toimia kuten antenni. (2.)

Analogisien ja digitaalisten piirien suunnittelussa tarvitaan erilaisia tekniikoita maadoitusten, tehonjakautumisen ja yhteyksien rakentamiseen. Esimerkiksi yhden pisteen maadoitus voi olla haluttua joissakin matalataajuisissa analogisissa piireissä, mutta se voi olla häiriötä keräävä tai säteilyn lähde digitaalisessa piirissä. (2.)

Digitaaliset piirit yhdistettynä johtimien induktanssiin voivat olla merkittäviä häiriön lähteitä. Induktanssin virran muutoksista syntyvä jännite seuraa kaavaa

$$V = L \frac{di}{dt} \quad (3)$$

, jossa V on jännite, L on induktanssi ja $\frac{di}{dt}$ on virranmuutos suhteessa aikaan (2). TMP126 vie maksimissaan 160 mikroampeeria. Nousu- ja laskuaikoja ei ole ilmoitettu, mutta oletetaan niiden olevan noin 15 nanosekuntia. Jos johtimen

induktanssi olisi noin 200 nH, niin häiriön aiheuttama jännite olisi tällöin noin 2 millivolttia. Jos käyttöjännite on 3,3 V, niin johtimien induktanssi voi huomattavasti vaikuttaa piirin toimivuuteen. Häiriön määrä kasvaa lämpötilan noustessa tai johtimen pituutta lisätessä. Tosin TMP126:n pitäisi olla 175 celsiusen lämpötilassa viedäkseen 160 μ A verran virtaa, joten se tuskin tulee olemaan häiriön lähde. Toki lämpötilan nousu saattaa olla tilanne, joka jo häiriöisessä tilassa on se viimeinen niitti.

Digitaaliset piirien taajuuden nopeus ei ole suhteessa kellon taajuuteen vaan kellon nousuaikaan. Arduinon kellon muuttaminen tällöin ei välttämättä vaikuta piirin toimintaan. SIO-linjaan lisätty vastus hidastaa nousuaikaa. Koska Arduinon nousuaika on luultavasti TMP126 anturin käyttöä varten odotettua hitaampi (johtimen pituuden seurauksesta), hidastaa kymmenen kilo-ohmin vastus liikaa signaalin nousuaikaa. Toisaalta kymmenen kilo-ohmin vastus erottaa SIO-linjat toisistaan. Hitaampi nousuaika ja laskuaika vähentävät piiriin syntyvää häiriötä luvun 5 kaavan 2 mukaisesti.

Tässä työssä hankalana toimii anturin hyvän tehonsyötön toteutuminen. Hyvän tehonjakautumisen tuntomerkkinä pidetään matalaimpedanssista silmukkaa, joka syöttää jatkuvaa tasaista tasajännitettä ilman vaihtovirrasta syntyvää häiriötä. Tehonlähteen asettelu tulisi olla samanlainen ja samansuuntainen kuin maan. Tehonlähteen häiriötä voi parantaa teholähteen erotuksella, hyvällä tehoverkostolla tai tehokerroksen jakautumissysteemillä, mutta se ei ole niin tärkeää kuin hyvä maadoittaminen. Erotuskondensaattorin lisääminen lisää myös johtimen impedanssia korkeilla taajuuksilla kondensaattoreille tyypillisen ekvivalentin sarjainduktanssin myötä (ESL). (2.) Kyseessä on komponentin testaaminen, joten lopullisen piirin ennustaminen on lähes mahdotonta.

Erotuskondensaattorilla on kapasitanssisuutensa lisäksi induktanssia. Induktanssia lisäävät myös piirilevyn reitit ja läpiviennit sekä integroitujen piirien sisäiset rungot. Tämän vuoksi erotuskondensaattorin lisääminen tarkoittaa L-C verkon lisäämistä. L-C piirillä on tietty resonanssitaajuus, jolloin piirin impedanssi on matalimmillaan ja jonka voi laskea kaavalla

$$f_r = \frac{1}{2\pi\sqrt{LC}} \quad (4)$$

f_r on resonanssitaajuus (Hz)

L on induktanssi (H)

C on kapasitanssi (F).

Tämän taajuuden ala- ja yläpuolella piiristä tulee induktiivinen ja sen impedanssi kasvaa taajuuden myötä. Resonanssitaajuus on piste, jossa kapasitiivinen ja induktiivinen reaktanssi kumoavat toisensa. Korkeataajuisille piireille voi olla tehokas tapa pienentää impedanssia käyttämällä useampaa samanarvoista erotuskondensaattoria. Tosin usean kondensaattorin käyttö pienentää kokonaiskapasitanssia, josta on haittaa matalammalla käyttötaajuudella toimiville piireille, jotka vaativat enemmän kapasitanssia. Usean eriarvoisen kondensaattorin käytössä on riski luoda piikkejä impedanssien resonanssitaajuuksien väleihin. (2.)

Teholähteen jakaminen useamman piirin kanssa on hyvä erottaa RC- tai LC-suodattimilla, jotta piireistä ei kytkeytyisi häiriöitä toisiinsa. RC-suodattimen vastuksesta aiheutuu jännitteen lasku, joten suodatinta voidaan käyttää rajoitetusti. Korkeilla taajuuksilla LC-suodatin toimii erityisen hyvin. Sillä on myös resonointitaajuus. (2.)

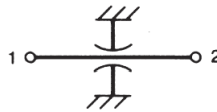
Käyttötaajuus on tärkein tekijä kondensaattorien valinnassa. Kondensaattorin maksimi käyttötaajuus riippuu kondensaattorin rakenteen induktanssista ja kondensaattorin jaloista. Jollain taajuuksilla kondensaattori alkaa värähtelemään oman induktanssinsa vuoksi. Tämän itseresonointitaajuuden alapuolella kondensaattori vaikuttaa kapasitanssina, jonka induktanssi vähenee taajuuden myötä. Kun tämä taajuus taas ylitetään, kondensaattori muuttuu induktiiviseksi, jonka impedanssi kasvaa taajuuden kanssa. (2.)

Tantaalikondensaattoreilla on vähemmän sisäistä vastusta (ESR) verrattuna elektrolyyttikondensaattoreihin, ja niitä voidaan käyttää muutaman megahertsin käyttötaajuuksilla. Elektrolyyttikondensaattorit ovat matalataajuisia kondensaattoreita, joita ei tulisi käyttää yli 25 kHz käyttötaajuuksilla. Silver Mica -

kondensaattoreita ja keraamisilla kondensaattoreilla on pieni sarjaresistanssi ja induktanssi, mikä tekee niistä käytännöllisiä korkeilla taajuuksilla. Läpilyöntäviä kondensaattoreita voidaan käyttää 500 MHz saakka, ja pintaliitoskondensaattorien käytettävyys ylittää gigahertseihin. (2.) Keraamisia kondensaattoreita luokitellaan niiden lämpötilaherkkyden ja toleranssin mukaan. NP0 tai C0G on esimerkki hyvin stabiilista keraamisesta kondensaattorista.

Korkean resistanssin arvoisilla vastuksilla kapasitiivinen reaktanssi kasvaa merkittävästi taajuuden myötä. Tällöin vastuksen impedanssi pienenee ja vaiheen kulma muuttuu. (2.)

Häiriösuojien läpisyöttäviä, kolmeliittimisiä kondensaattoreita, piirikaaviokuva 20, voi käyttää korkeiden taajuuksien pois kytkemiseksi virtalähteistä tai matalataajuisista signaaleista (2). Näitä löytyy myös pintaliitoskomponentteina.



Kuva 20. Läpisyöttävän kondensaattorin piirikaaviokuva (2).

7 CRC

TMP126 käyttää CRC-16 / CCITT-FALSE algoritmia tarkastussumman laskemiseen. Anturille lähetetyssä datassa voidaan määrittää, käytetäänkö tarkastussummaa. Jos tarkastussummaa käytetään, liitetään se lähetetyn datan loppuun. Jos lähetetty data ei vastaa tarkastussummaan, anturi ei reagoi dataan. (7.)

Datalehdessä annettua algoritmista tehtiin aliohjelma (esimerkkikoodi 1), joka liitettiin mikrokontrollerin ohjelmointiin (7).

```
#include <stdio.h>
```

```
unsigned short
```

```

crcCalc (int msglen, unsigned int msg[20])
{
    unsigned short crc = 0xFFFF;
    for (int byte = 0; byte < msglen; byte++){
        crc ^= msg[byte];
        printf ("msgbyte: 0x%X\n", msg[byte]);
        for (int bit = 0; bit < 16; bit++) {
            printf ("crc: 0x%X byte: %d bit: b%d\n", crc, byte, bit);
            if (crc & 0x8000)
                crc = (crc << 1) ^ 0x1021;
            else
                crc = (crc << 1);
        }
    }
    printf ("crc: 0x%X\n", crc);
    return crc;
}

int
main ()
{
    unsigned int msg[20];
    msg[0] = 0b01001001;
    msg[1] = 0x00;
    int msglen = 2;

    unsigned short crc = crcCalc (msglen, msg);
    printf ("%d", crc);
    return 0;
}

```

Esimerkkikoodi 1. CRC tarkastussumman käyttäminen parantaa anturin luotettavuutta, jos dataväylään syntyy häiriöitä. Koodin alkuperä on datalehdessä (7).

8 TMP126 testaaminen STM32 mikrokontrollerilla

Tässä testissä (29.8.2023) Piiristä rakennettiin sulautetumpi järjestelmä jättäen pois luvun 6 kuvan 15 lisäosa. Mikrokontrollerina toimi STM32L010K8T6, ja ohjelmoimiseen käytettiin STLINK-V3SET-laitetta. Piirin jännite aleni näin viidestä voltista 3,3 volttiin, jolloin piiri on alttiimpi RF-häiriöille. Anturi suojattiin kupariteipillä, johon juotettiin kaapelinsuoja (kuva 21). Piiri suojattiin metallilaatikkoon toisessa päässä, johon kaapelinsuoja yhdistettiin. Ohjelmoimiseen käytettiin neljänneen liitteen koodia.



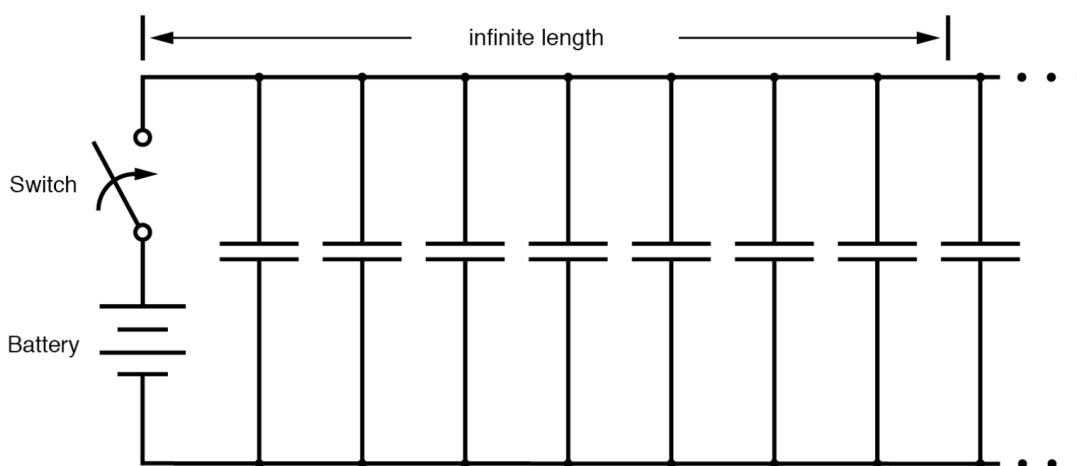
Kuva 21. Anturi häiriösuojattiin kupariteipillä.

Jo ennen itse testiä anturi ei antanut luotettavasti lämpötilalukemaa. Se oli toiminut hyvin koekytkenvälävyyllä, muttei enää metrin pituisen kaapelin päässä. Testin aikana piirin toiminta ei muuttunut, esimerkiksi LCD-näytön häiriintyminen, eli kyseessä ei ollut RFI-ongelma. Tähän myös viittaa se, että SIO-linjan vastuksen pienentäminen kymmenestä kilo-ohmista yhteen paransi aikaisemmin piirin toimivuutta.

9 SPI sarjaväylän kaapelin pituus

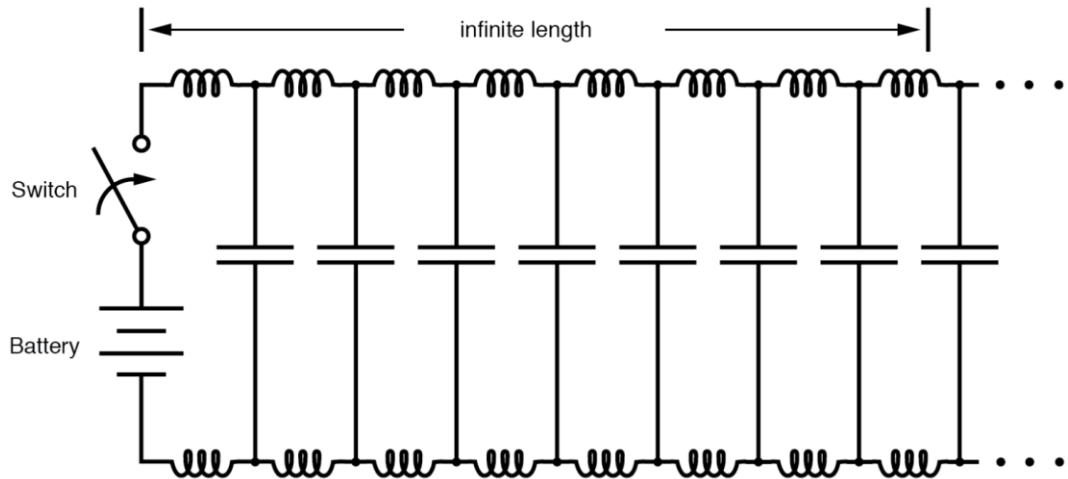
Piirin kapasitanssia voi kasvattaa kondensaattoreilla. Myös kaapelit kasvattavat piirin kapasitanssia, ja niitä voi karkeasti ajatella ikään kuin kondensaattoreina. TMP126 anturin datalehdessä (TMP126 Low-Power, 0.25°C SPI Temperature Sensor With 175°C Operation, CRC and Slew Rate Alert) annetaan

digitaalisen väylän sisään tulon maksimi kapasitanssiksi 20 pF, jos käyttötaajuus on yksi megahertsi (7). SPI-väylän maksimaalisesta pituudesta ei ole ilmoitettu. Kaapelin kapasitanssiin vaikuttaa moni tekijä, joten yksiselitteistä vastausta ei löydy. Kapasitanssin kasvaminen vaikuttaa signaalien viiveeseen. SPI, joka on synkroninen protokolla seuraten kellon nousu, lasku, ylhäällä tai alhaalla oloaika, on erityisen herkkä viiveille. Kuva 22 havainnollistaa, miten kapasitanssi kasvaa johtimien pituuden kasvaessa.



Kuva 22. Kapasitanssi kasvaa johtimen pituuden myötä, mikä kasvattaa myös signaalin viivettä (8).

Johtimien induktiivinen reaktanssi usein aliarvioidaan piirisuunnittelussa. Induktanssi vastustaa virrankulun muutoksia. Ohmin lain mukaan virran väheneminen vähentää jännitettä. Johtimien siirtäminen kauemmaksi esimerkiksi monikerroksisen piirilevyn maatasosta lisää myös johtimen induktanssia, kuva 23. Johtimien induktanssia voi vähentää lyhentämällä niitä, kasvattamalla niiden läpimitaa, lisäämällä vaihtoehtoisia virran reittejä tai pienentämällä virtasilmuksia. Johtimien läpimitan kasvattaminen vähentää tasajännitteellä tai matalilla taajuuksilla resistanssia. Reittien verkottamista käytetään korkeataajuisen signaalien impedanssin vähentämiseksi piirisuunnittelussa. Kokonaisimpedanssin vähentämiseksi johtimet, jotka kuljettavat virtaa samaan suuntaan, olisi hyvä erottaa, ja johtimet, jotka kuljettavat virtaa vastakkaisiin suuntiin, tulisi pitää mahdollisimman lähellä toisiaan. (2.)

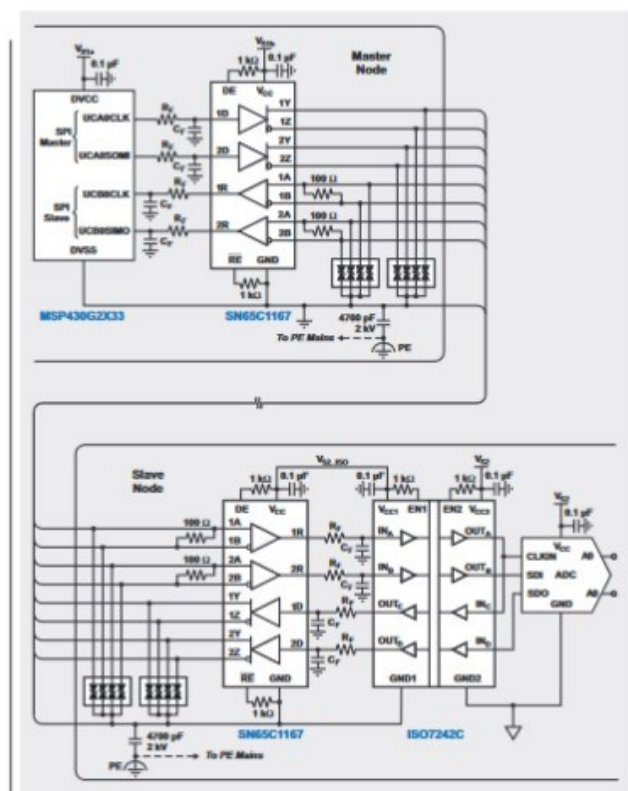


Kuva 23. Induktanssi kasvaa myös johtimen myötä, jolloin jännite häviää (8). Tämä muistuttaa balansoitua LC-suodatinta.

Digitaalisten piirien johtimia tulisi kohdella kuten siirtolinjoja, kun signaalien nousuaika on kaksi kertaa vähemmän kuin etenemisviive johtimessa. Siirtolinjojen ero johtimiin on se, että ne toimivat elektromagneettisen kentän tai energian ohjaajina paikasta toiseen eikä niinkään jännitteen ja virran. Jännite ja virta ilmenevät johtimissa, mutta ne ovat kentästä johtuvia seurauksia. Elektromagneettinen energia kulkee dielektrisessä tilassa – keskiosa on usein ilmassa. Elektromagneettinen energia kulkee valonnopeudella, mikä tarkoittaa myös sitä, että sen siirtyminen ei johdu johtimien elektronien liikkumisesta. Elektronien nopeus johtimessa on noin 0,01 m/s, mikä on 30 biljoonaa kertaa hitaampaa kuin valonnopeus tyhjiössä. Luultavasti yleisin siirtolinja on koaksiaalikaapeli. (2.)

Siirtolinjoissa signaalin vaimentavia tekijöitä ovat johtimen resistanssista johtuva ohminen häviö sekä dielektrinen häviö, joka on taas dielektrisen materiaalin energian absorboinnista elektromagneettisen kentän levittämiseen ja elementtien lämpenemisen johtuvaa seurausta. Siirtohäviöt ovat häviävän pieniä, kun käyttötaajuus lähestyy yhtä gigahertsiä. Siirtolinjoissa ilmenee heijastuksia, kun piirin impedanssi muuttuu. Näitä muutoksia voivat olla esimerkiksi piirilevyn läpiviennit tai johtimien suorat kulmat. (2.)

SPI sarjaväylää ei ole luotu pitkän matkan datasiirtoon. Texas Instruments on julkaissut SPI-väylän pidentämisestä artikkelin Extending the SPI bus for long-distance communication. Kuvassa 24 on artikkelissa annettu esimerkki piirikaaviosta. Artikkelissa on muunnettu SPI-väylä RS-422 muotoon pitkän matkan datansiirtoon. (9).



Kuva 24. Yksi vaihtoehto olisi käyttää SN65C1167 integroituja kaksoisdifferentiaalisia -piirejä (9).

Toinen vaihtoehto on käyttää puskureita signaalin vahvistukseen. Jäljelle jääneisiin SOT-23 adaptoreihin kokeiltiin SN74LC1G17 Schmitt-trigger puskureiden toimivuutta. Yhdistin kaksi noin 25 senttimetrin pituista kaapelia ja lisäsin jokaisen datalinjaan puskurin. Toimivuus parani, mutta anturi ei vieläkaan toiminut moitteettomasti. Kaapeli jäi lyhyeksi, ja se olisi ollut hankala testata vahvistimessa turvallisesti. Puskureiden käyttö aiheuttaa jonkin verran viiveitä datalinjoihin.

10 Yhteenveto

Lämpötila-anturi TMP126 toimii hyvin lyhyillä matkoilla. Pitkän matkan SPI data-siirtoon on ratkaisuja, mutta ne lisäävät tuotteen lopullisia kustannuksia. Vaikka kolmen johtimen SPI säästää yhden johtimen verrattuna perinteiseen neljän johtimen datansiirtoon, saattaa se monimutkaistaa suunnittelua. SPI-protokollan käyttö häiriöisessä tilassa lisää epäluotettavuutta se, että tiedon pitää kulkea molempiin suuntiin. Anturilta pitää pyytää lämpötila-arvoa, ja anturi vastaa tähän. Binäärisien arvojen siirtäminen on altis häiriöille. Suurempi jännite tai differentiaalinen datansiirto, kuten MODBUS tai CAN-väylä, parantaisi huomattavasti piirin luotettavuutta, mutta lisäosat tulevat lisäämään lopputuotteen hintaa. SPI-piirissä pitää ottaa huomioon myös teholähteen silmukan häiriöttömyys. Tällöin johtimia on lopulta kuusi, joiden jokaisen häiriöttömyys pitää ratkaista. Toisen lämpötila-anturin valinta luultavasti johtaa kaikin puolin parempaan lopputulokseen ottaen huomioon yksinkertaisuuden, luotettavuuden ja häiriönsiedon.

TMP126 anturin toimivuuden testaus osoittautui tässä työssä hankalaksi toteuttaa asetetulla tavalla. Ongelmana oli kaapelin pituus, haastava sijainti ja työturvallisuus. Vaikka anturi ehkä itsessään toimisi tässä tilassa hyvin, niin SPI tyyppinen datansiirto tämän tilan läpi luotettavasti tuottaa ongelmia. Ehkä työn alkua asetelma on väärin aseteltu. Kysymyksenä voisi olla, voiko SPI siirtolinja toimia häiriöisen tilan läpi? Mielestäni tähän löytyy ratkaisuja. Mutta onko se järkevää? Mielestäni ei tässä tapauksessa.

Lähteet

- 1 International Standard – Medical electrical equipment. 2020. International Electrotechnical Commission. Geneva: IEC publications painos 3.2.
- 2 Ott, Henry W.. 2009. Electromagnetic Compatibility. New Jersey. John Wiley & Sons, Inc.
- 3 Johnson, John B. 1927. Thermal Agitation of Electricity in Conductors. Nature Publishing Group No. 2984, Vol. 119.
- 4 O’Hara, Martin. 1998. EMC at component and PCB level. Oxford. Newnes.
- 5 International Standard – Electromagnetic compatibility (EMC) – Part 6-1: Generic standards – Immunity standard for residential, commercial and light-industrial environments. 2007. International Electrotechnical Commission. Geneva: IEC publications.
- 6 Trump, Bruce. 2011. Analog linearization of resistance temperature detectors. Verkkoaineisto. Texas Instruments Incorporated. <<https://www.ti.com/lit/an/slyt442/slyt442.pdf>>. Luettu 2.11.2023.
- 7 TMP126 Low-Power, 0.25°C SPI Temperature Sensor With 175°C Operation, CRC and Slew Rate Alert. 2022. Verkkoaineisto. Texas Instruments Incorporated. <https://www.ti.com/lit/ds/sym-link/tmp126.pdf?ts=1693651555609&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTMP126>. Päivitetty kesäkuu 2022. Luettu 3.9.2023.
- 8 Kuphaldt, Tony R.. 2015. Characteristic Impedance. Verkkoaineisto. All About Circuits. <<https://www.allaboutcircuits.com/textbook/alternating-current/chpt-14/characteristic-impedance/>>. Päivitetty 11.1.2021. Luettu 30.10.2023.
- 9 Kugelstadt, Thomas. 2011. Extending the SPI bus for long-distance communication. Verkkoaineisto. Texas Instruments Incorporated. <<https://www.ti.com/lit/an/slyt441/slyt441.pdf>>. Luettu 18.10.2023.

Lämpötila-anturien taulukointi

Sensori	Interface	Hinta	Erä (kpl)	Kiinnitys	Saatavuus (M=Mouser,	Valmistaja	Paketti
ERT-JZET472J	NTC	0,02		SMD	0 (M)	Panasonic	
S-5813A-I4T1U	Analog	0,27		SMD	0 (M)	ABLIC	SNT-4A
MCP9701AT-E/LT	Analog	0,31		SMD	37 240 (M)	Microchip	SC-70-5
LMT84LPM	Analog	0,29		TH	3 991 (M)	Texas Instruments	TO-92-3
LMT86LPM	Analog	0,29		TH	3 784 (M)	Texas Instruments	TO-92-3
TMP103GYFFT	I2C	0,7	1000	SMD	0 (M)	Texas Instruments	DSBGA-4
NB-PTCO-386	RTD	0,77		TH	0 (M)	TE Connectivity	
LTC110_JTP	PTC	0,12		SMD	0 (M)	Kamaya	
YQB120N0025	PTC	0,56		TH	0 (M)	Amphenol	
LMT70YFQR	CMOS analog	0,95	500	SMD	11 (M)	Texas Instruments	WLCSP-4
SHT41I-AD1B-R3	I2C	1,67	10 000	SMD	0 (M) 0(F) 9 995(D)	Sensirion	
SHT41A-AD1B-R2	I2C	2,58	50+	SMD	7 919(M) 2425 (F) 8273 (D)	Sensirion	
TMP175AIDR	I2C	0,86	500	SMD	2 686 (M) 392 (F) 3 123(D)	Texas Instruments	SOIC-8
LMT01LPG	PCM	1,38	1 000	TH	0 (M) 0(F) 0(D)	Texas Instruments	TO-92-2
TMP06BKSZ-500RL7	PWM	2,12	2 500	SMD	3 277 (M) 475 (F) 0(D)	Analog Devices Inc.	5-pin SC-70
MAX6673A-XK+T	PWM	4,2	2 500	SMD	2 109 (M) 2518(F) 4 128(D)	Maxim Integrated	5-pin SC-70
TC72-5.0MMFTR	SPI	0,98	3 300	SMD	0 (M) 0(F) 0(D)	Microchip	DFN-8
TMP127EDBVRQ1	SPI	0,69	1 000	SMD	4 177 (M) 0(F) 1 569(D)	Texas Instruments	SOT-23-6
TMP104YFFR	SMAART	0,66	3 000	SMD	0 (M) 0(F) 602(D)	Texas Instruments	DSBGA-4
TMP126NDCKR	SPI	0,71	3 000	SMD	5 206(M) 0(F) 2 344(D)	Texas Instruments	SC-70-6
TMP126EDBVRQ1	SPI	1,24	3 000	SMD	264(M) 0(F) 0(D)	Texas Instruments	SOT-23-6
TMP126DCKR	SPI	1,24	3 000	SMD	2 492(M) 0(F) 1 716(D)	Texas Instruments	SC-70-6
TMP126EDCKRQ1	SPI	1,27	3 000	SMD	0 (M) 0(F) 2 547(D)	Texas Instruments	SC-70-6
TMP708AIDBVR	Thermostats	0,17	3 000	SMD	4 553(M) 0(F) 9 000(D Factory)	Texas Instruments	SOT-23-5
TCN75AVUJA713	I2C	0,64	5 000	SMD	4 355(M) 0(F) 19 810(D)	Texas Instruments	MSOP-8
BME280	I2C, SPI	2,87	5 000	SMD, SMT	70 680(M) 161 290 (D)	Bosch	LGA-8
HTS221	SPI	2,84	500+	SMD	0 (F)	ST	HLGA

TMP126 Arduino IDE -koodi 1

```
/*
Author Tuomo Kupari
11.3.2023

SPI Temperature test
        tmp126
CS: pin 7    -> CS (inverted) pin1
MOSI: pin 11 -> SIO pin 3
MISO: pin 12 -> 10k res SIO pin 3
SCK: pin 13  -> SCLK 4
                -> ALERT (inverted) pin 5
                -> GND pin 2
                -> VDD pin 6

MAX 10 MHz
*/

#include <SPI.h>
#include <Wire.h>
#include "rgb_lcd.h"

rgb_lcd lcd;

const int colorR = 0;
const int colorG = 0;
const int colorB = 255;

void setup() {
  lcd.begin(16, 2);
  lcd.setRGB(colorR, colorG, colorB);
  delay(1000);

  Serial.begin(9600);
  SPI.begin();

  // initialize the data ready and chip select pins:
  const int chipSelectPin = 7;
  pinMode(chipSelectPin, OUTPUT);
}

void loop() {
  int chipSelectPin = 7;
  unsigned int tempData = readRegister(chipSelectPin);
  Serial.print("TempData = ");
  float temp = tempData*0.03125;
  Serial.print(temp);
  Serial.println("\n");
  lcd.setCursor(0, 0);
  lcd.print(temp);
}

//Read from or write to register:
unsigned int readRegister(int chipSelectPin) {
  // result to return
  unsigned int result = 0;

  // Enable reading 0b01000001
```

```
uint8_t b1 = 0b01000001;
uint8_t b2 = 0b00000000;
// take the chip select low to select the device:
digitalWrite(chipSelectPin, LOW);
delay(10);
// send the device the register you want to read:
result = SPI.transfer(b1);
//Serial.println(result);
result = SPI.transfer(b2);
//Serial.println(result);

// Read first 16 bits
uint8_t b3 = 0b00000000;
uint8_t b4 = 0b00000000;
result = SPI.transfer(b3);
//Serial.println(result);
result = SPI.transfer(b4);
//Serial.println(result);

//Read another 16 bits
uint8_t b5 = 0b00000000;
uint8_t b6 = 0b00000000;
uint8_t resultB5 = SPI.transfer(b5);
//Serial.println(resultB5);
uint8_t resultB6 = SPI.transfer(b6);
//Serial.println(resultB6);
unsigned int data[] = {resultB5,resultB6};

unsigned long y = (data[0]<<8) + data [1];
// Remove two unnecessary bits
y = y>>2;

/*
CRC would be read here
uint8_t b7 = 0b00000000;
uint8_t b8 = 0b00000000;
result = SPI.transfer(b7);
Serial.println(result);
result = SPI.transfer(b8);
Serial.println(result);
*/

// Stop SPI connection pulling CS pin high
digitalWrite(chipSelectPin, HIGH);
// Return outcome
return (y);
}
```


TMP126 Arduino IDE -koodi 2

```
/*
Author Tuomo Kupari
11.3.2023

SPI Temperature test
                tmp126
CS: pin 7      -> CS (inverted) pin1
MOSI: pin 11   -> SIO pin 3
MISO: pin 12   -> 1k res SIO pin 3
SCK: pin 13    -> SCLK 4
                -> ALERT (inverted) pin 5
                -> GND pin 2
                -> VDD pin 6

MAX 10 MHz
*/

#include <SPI.h>
#include <Wire.h>
#include "rgb_lcd.h"

rgb_lcd lcd;

const int colorR = 0;
const int colorG = 0;
const int colorB = 255;

void setup() {
  lcd.begin(16, 2);
  lcd.setRGB(colorR, colorG, colorB);
  delay(1000);

  Serial.begin(9600);
  SPI.setClockDivider(SPI_CLOCK_DIV8);
  SPI.begin();

  // initialize the data ready and chip select pins:
  const int chipSelectPin = 7;
  pinMode(chipSelectPin, OUTPUT);
}

void loop() {
  int chipSelectPin = 7;
  unsigned int tempData = readRegister(chipSelectPin);
  Serial.print("TempData = ");
  float temp = tempData*0.03125;
  Serial.print(temp);
  Serial.println("\n");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(temp);
  delay(1000);
}

//Read from or write to register:
unsigned int readRegister(int chipSelectPin) {
  // result to return
```

```

uint16_t result = 0;

/*
x don't care | CRC Enable | L4 | L3 | L1 | L0 | Auto Increment | R/W
|
|      0      |      1      | 1 | 0 | 1 |
0 |          1          | 0 |
*/
uint8_t b1 = 0b01101010;

/*
Sub-Address
*/
SPI.beginTransaction(SPISettings(100000,MSBFIRST,SPI_MODE3));
delay(1000);
uint8_t b2 = 0b00000000;
uint16_t c1 = (b1<<8) + b2;
//Serial.println(c1);
// take the chip select low to select the device:
digitalWrite(chipSelectPin, LOW);
delay(1000);
// send the device the register you want to read:
result = SPI.transfer16(c1);
Serial.println(result);

// Read first 16 bits
uint8_t b3 = 0b00000000;
uint8_t b4 = 0b00000000;
uint16_t c2 = (b3<<8) + b4;
result = SPI.transfer16(c2);
Serial.println(result);
// Remove two unnecessary bits
unsigned int y = result>>2;

// Stop SPI connection pulling CS pin high
digitalWrite(chipSelectPin, HIGH);
// Return outcome
return (y)

```

TMP126 STM32 -koodi

```
/* USER CODE BEGIN Header */
/**
*****
*
* @file      : main.c
* @brief     : Main program body
*****
*
* @attention
*
* Copyright (c) 2023 STMicroelectronics.
* All rights reserved.
*
* This software is licensed under terms that can be found in the LICENSE
file
* in the root directory of this software component.
* If no LICENSE file comes with this software, it is provided AS-IS.
*
*****
*
*/
/* USER CODE END Header */
/* Includes -----
-*/
#include "main.h"

/* Private includes -----
-*/
/* USER CODE BEGIN Includes */
#include <string.h>
#include <stdlib.h>
#include <math.h>
/* USER CODE END Includes */

/* Private typedef -----
-*/
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define -----
-*/
/* USER CODE BEGIN PD */
#define GROOVELCD_ADDR 0x3e<<1
#define ADS1115_ADDRESS 0x48<<1
/* USER CODE END PD */

/* Private macro -----
-*/
/* USER CODE BEGIN PM */
```

```

/* USER CODE END PM */

/* Private variables -----
-*/
I2C_HandleTypeDef hi2c1;

SPI_HandleTypeDef hspi1;

/* USER CODE BEGIN PV */

/* USER CODE END PV */

/* Private function prototypes -----
-*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_I2C1_Init(void);
static void MX_SPI1_Init(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code -----
-*/
/* USER CODE BEGIN 0 */
float temperature[1];
unsigned short crcCalc (int msglen, unsigned int msg[4])
{
    unsigned short crc = 0xFFFF;
    for (int byte = 0; byte < msglen; byte++){
        crc ^= msg[byte];
        for (int bit = 0; bit < 16; bit++) {
            if (crc & 0x8000)
                crc = (crc << 1) ^ 0x1021;
            else
                crc = (crc << 1);
        }
    }
    return crc;
}
/* USER CODE END 0 */

/**
 * @brief The application entry point.
 * @retval int
 */
int main(void)
{
    /* USER CODE BEGIN 1 */
    uint8_t spi_buf[8];
    unsigned int crc_buf[8];
    uint16_t crc[1];
    /* USER CODE END 1 */

    /* MCU Configuration-----
-*/

```

```
/* Reset of all peripherals, Initializes the Flash interface and the Sys-
tick. */
HAL_Init();

/* USER CODE BEGIN Init */

/* USER CODE END Init */

/* Configure the system clock */
SystemClock_Config();

/* USER CODE BEGIN SysInit */

/* USER CODE END SysInit */

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_I2C1_Init();
MX_SPI1_Init();
/* USER CODE BEGIN 2 */
// CS pin should default high
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, GPIO_PIN_SET);

uint8_t data_t [2];
data_t[0] = 0x80;
data_t[1] = 0x28;
HAL_I2C_Master_Transmit(&hi2c1, (uint16_t)GROOVELCD_ADDR, (uint8_t*)data_t,
2, 50);
HAL_Delay(1);

data_t[0] = 0x80;
data_t[1] = 0x0C;
HAL_I2C_Master_Transmit(&hi2c1, (uint16_t)GROOVELCD_ADDR, (uint8_t*)data_t,
2, 50);

data_t[0] = 0x80;
data_t[1] = 0x01;
HAL_I2C_Master_Transmit(&hi2c1, (uint16_t)GROOVELCD_ADDR, (uint8_t*)data_t,
2, 50);

data_t[0] = 0x80;
data_t[1] = 0x06;
HAL_I2C_Master_Transmit(&hi2c1, (uint16_t)GROOVELCD_ADDR, (uint8_t*)data_t,
2, 50);
HAL_Delay(1);

char *str = malloc (6);
uint16_t result;

// Write settings
spi_buf[0] = 0b01010000;
spi_buf[1] = 0b00000011;
spi_buf[2] = 0b00000000;
spi_buf[3] = 0b01000000;
crc_buf[0] = spi_buf[0];
crc_buf[1] = spi_buf[1];
crc_buf[2] = spi_buf[2];
crc_buf[3] = spi_buf[3];
```

```

crc[0] = crcCalc(4,crc_buf);
spi_buf[4] = crc[0]>>8;
spi_buf[5] = crc[0] & 0xFF;

// Write 6 bytes starting at given address
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, GPIO_PIN_RESET);
HAL_Delay(10);
HAL_SPI_Transmit(&hspi1, (uint8_t *)spi_buf, 6, 100);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, GPIO_PIN_SET);
HAL_Delay(100);

/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    HAL_Delay(1000);
    result = 0xFF;
    while (result!=0){
        // Read alerts
        spi_buf[0] = 0b01001001;
        spi_buf[1] = 0b00000010;
        crc_buf[0] = spi_buf[0];
        crc_buf[1] = spi_buf[1];
        crc[0] = crcCalc(2,crc_buf);
        spi_buf[3] = crc[0]>>8;
        spi_buf[4] = crc[0] & 0xFF;
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, GPIO_PIN_RE-
SET);

        HAL_Delay(10);
        HAL_SPI_Transmit(&hspi1, (uint8_t *)spi_buf, 4,
100);

        HAL_SPI_Receive(&hspi1, (uint8_t *)spi_buf, 2,
100);

        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, GPIO_PIN SET);
        HAL_Delay(100);

        /*Shift to the second line of lcd*/
        data_t[0] = 0x80;
        data_t[1] = 0xC0;
        HAL_I2C_Master_Transmit(&hi2c1,
(uint16_t)GROOVELCD_ADDR, (uint8_t*)data_t, 2, 50);

        result = (spi_buf[0]<<8)+spi_buf[1];
        sprintf(str, "%d", result);
        if(result==0x01){
            data_t[0] = 0x80;
            data_t[1] = 0x01;
            HAL_I2C_Master_Transmit(&hi2c1,
(uint16_t)GROOVELCD_ADDR, (uint8_t*)data_t, 2, 50);
            HAL_Delay(1);
            print(GROOVELCD_ADDR, "Data unread");
        } else {
            print(GROOVELCD_ADDR, str);
            HAL_Delay(2);
        }
    }
}

```

```

//Read temperature
spi_buf[0] = 0b01001001;
spi_buf[1] = 0b00000000;
crc_buf[0] = spi_buf[0];
crc_buf[1] = spi_buf[1];
crc[0] = crcCalc(2,crc_buf);
spi_buf[2] = crc[0]>>8;
spi_buf[3] = crc[0] & 0xFF;
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, GPIO_PIN_RESET);
HAL_Delay(10);
HAL_SPI_Transmit(&hspi1, (uint8_t *)spi_buf, 4, 100);
HAL_SPI_Receive(&hspi1, (uint8_t *)spi_buf, 2, 100);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, GPIO_PIN_SET);
HAL_Delay(100);

/*Shift to the first line of the lcd*/
data_t[0] = 0x80;
data_t[1] = 0x03;
HAL_I2C_Master_Transmit(&hi2c1, (uint16_t)GROOVELCD_ADDR,
(uint8_t*)data_t, 2, 50);
HAL_Delay(1);

result = (spi_buf[0]<<8)+spi_buf[1];
unsigned int temp = result>>2;
float reading = temp*0.03125;
sprintf (str, "%.2f", reading);
print(GROOVELCD_ADDR, str);

print(GROOVELCD_ADDR, " crc");
sprintf (str, "%d", crc[0]);
print(GROOVELCD_ADDR, str);
HAL_Delay(2);
}
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */
/* USER CODE END 3 */
}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
    RCC_PeriphCLKInitTypeDef PeriphClkInit = {0};

    /** Configure the main internal regulator output voltage
    */
    __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);

    /** Initializes the RCC Oscillators according to the specified parameters
    * in the RCC_OscInitTypeDef structure.
    */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_MSI;

```

```

RCC_OscInitStruct.MSISState = RCC_MSI_ON;
RCC_OscInitStruct.MSICalibrationValue = 0;
RCC_OscInitStruct.MSIClockRange = RCC_MSIRANGE_5;
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_NONE;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{
    Error_Handler();
}

/** Initializes the CPU, AHB and APB buses clocks
 */
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSClockSource = RCC_SYSCLOCKSOURCE_MSI;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLOCK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_0) != HAL_OK)
{
    Error_Handler();
}
PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_I2C1;
PeriphClkInit.I2C1ClockSelection = RCC_I2C1CLKSOURCE_PCLK1;
if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
{
    Error_Handler();
}
}

/**
 * @brief I2C1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_I2C1_Init(void)
{
    /* USER CODE BEGIN I2C1_Init 0 */

    /* USER CODE END I2C1_Init 0 */

    /* USER CODE BEGIN I2C1_Init 1 */

    /* USER CODE END I2C1_Init 1 */
    hi2c1.Instance = I2C1;
    hi2c1.Init.Timing = 0x00000708;
    hi2c1.Init.OwnAddress1 = 0;
    hi2c1.Init.AddressingMode = I2C_ADDRESSINGMODE_7BIT;
    hi2c1.Init.DualAddressMode = I2C_DUALADDRESS_DISABLE;
    hi2c1.Init.OwnAddress2 = 0;
    hi2c1.Init.OwnAddress2Masks = I2C_OA2_NOMASK;
    hi2c1.Init.GeneralCallMode = I2C_GENERALCALL_DISABLE;
    hi2c1.Init.NoStretchMode = I2C_NOSTRETCH_DISABLE;
    if (HAL_I2C_Init(&hi2c1) != HAL_OK)
    {
        Error_Handler();
    }
}

```



```

/** Configure Analogue filter
*/
if (HAL_I2CEx_ConfigAnalogFilter(&hi2c1, I2C_ANALOGFILTER_ENABLE) !=
HAL_OK)
{
    Error_Handler();
}

/** Configure Digital filter
*/
if (HAL_I2CEx_ConfigDigitalFilter(&hi2c1, 0) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN I2C1_Init 2 */

/* USER CODE END I2C1_Init 2 */

}

/**
 * @brief SPI1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_SPI1_Init(void)
{
    /* USER CODE BEGIN SPI1_Init 0 */

    /* USER CODE END SPI1_Init 0 */

    /* USER CODE BEGIN SPI1_Init 1 */

    /* USER CODE END SPI1_Init 1 */
    /* SPI1 parameter configuration*/
    hspi1.Instance = SPI1;
    hspi1.Init.Mode = SPI_MODE_MASTER;
    hspi1.Init.Direction = SPI_DIRECTION_2LINES;
    hspi1.Init.DataSize = SPI_DATASIZE_8BIT;
    hspi1.Init.CLKPolarity = SPI_POLARITY_LOW;
    hspi1.Init.CLKPhase = SPI_PHASE_1EDGE;
    hspi1.Init.NSS = SPI_NSS_SOFT;
    hspi1.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_16;
    hspi1.Init.FirstBit = SPI_FIRSTBIT_MSB;
    hspi1.Init.TIMode = SPI_TIMODE_DISABLE;
    hspi1.Init.CRCCalculation = SPI_CRCCALCULATION_ENABLE;
    hspi1.Init.CRCPolynomial = 7;
    if (HAL_SPI_Init(&hspi1) != HAL_OK)
    {
        Error_Handler();
    }
    /* USER CODE BEGIN SPI1_Init 2 */

    /* USER CODE END SPI1_Init 2 */

}

```

```

/**
 * @brief GPIO Initialization Function
 * @param None
 * @retval None
 */
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure = {0};

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_4|GPIO_PIN_5, GPIO_PIN_RESET);

    /*Configure GPIO pins : PB4 PB5 */
    GPIO_InitStructure.Pin = GPIO_PIN_4|GPIO_PIN_5;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOB, &GPIO_InitStructure);
}

/* USER CODE BEGIN 4 */
void print(uint16_t adresse, uint8_t *str)
{
    uint8_t data_a[strlen(str)];
    data_a[0] = 0x40;
    int i;
    for(i=0; str[i]!=0; i++)
    {
        data_a[1+i] = str[i];
    }
    HAL_I2C_Master_Transmit(&hi2c1, (uint16_t)adresse,
    (uint8_t*)data_a, 1+strlen(str), 50);
}
/* USER CODE END 4 */

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error return state */
    __disable_irq();
    while (1)
    {
    }
    /* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT

```

```
/**
 * @brief Reports the name of the source file and the source line number
 *        where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line
    number,
    ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line)
    */
    /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */
```