VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Tomi Knuuttila

KAPPALEEN ASENNON TUNNISTUS SULAUTETUSSA JÄRJESTELMÄSSÄ

Case, Tietolaite Oy

Tekniikka
2016

VAASAN AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma

TIIVISTELMÄ

Tekijä	Tomi Knuuttila
Opinnäytetyön nimi	Kappaleen asennon tunnistus sulautetussa järjestelmässä
Vuosi	2016
Kieli	suomi
Sivumäärä	51 + 1 liite
Ohjaaja	Jukka Matila

Opinnäytetyön aiheena on InvenSense MPU6050-asentotietoanturin käyttöönotto ja sen tuotteena yleiskäyttöisen ajurin kehitys. Työ on osa Tietolaite Oy:n tuotekehitystä. Tarve kehittää ajuri kyseiselle laitteelle syntyi tarpeesta pystyä tarkastelemaan kappaleen asentoa erilaisissa sovelluskohteissa. Sovelluskohteita voisivat olla esimerkiksi älypuhelimet, robotit tai vaikkapa kameroiden vakautusjärjestelmät.

MPU6050 on 6-akselinen asentotietoanturi, joka laskee mittauksen raaka-arvoista valmiita kvaternioita. Anturin muodostuu kahdesta kolmiakselisesta MEMS-anturista, kuudesta 16-bittisestä A/D-muuntimesta, 1024-tavun FIFO-puskurista ja laskentaa suorittavasta prosessorista.Ohjelmisto kirjoitettaan C-kielellä käyttäen hyväksi Tietolaite Oy:n T-Plat.E kirjastoa. Ajuri pyörii itsenäisenä tehtävänään (task) FreeRTOS-käyttöjärjestelmässä.

Opinnäytetyön tuloksena kehitetyllä ajurilla pystytään mittaamaan asentotieto luotettavasti sovelluskohteesta riippumatta.

Avainsanat sulautetut järjestelmät, C-kieli, i²c, MEMS

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Information Technology Degree Programme

ABSTRACT

Author	Tomi Knuuttila
Title	Object orientation recognition in embedded system
Year	2016
Language	Finnish
Pages	51 + 1 Appendice
Name of Supervisor	Jukka Matila

The subject of this thesis is introduction of Invensense MPU6050-motion tracking sensor and thus develop portable driver for it. Work is done a part of product development of Tietolaite Oy. Demand for developing driver to MPU6050 originated from need to be able to track position of object in different applications. Applications can contain for example smartphones, robotics or camera stabilization systems.

MPU6050 is 6-axis motion tracking sensor that calculates quarternions from raw measurement values. Sensor consists of two three axis MEMS-sensors, six 16-bit A/D-converter, 1024 byte FIFO-buffer and calculation performing processor. Software is written in C-language utilizing Tietolaite T-Plat.E library. Driver itself runs as independent task in FreeRTOS-operating system.

As a result of this thesis the developed driver can measure motion reliably not depended on the application.

Keywords Embedded systems, C-language, i²c, MEMS

SISÄLLYS

KÄSITELUETTELO

MEMS	Micro Electro Mechanical Systems, mikrosysteemi
QFN	Quad Flat No-leads on integroitujen piirien pakkaustekniikka.
Gyroskooppi	Kulmanopeusanturi
DMP	Digital Motion Processor, tuotenimi.
Task	Pre-emptiivinen prosessi
Co-Task	Co-operativiinen prosessi
OSA	Operating System API, käyttöjärjestelmä riippumaton käyttöjärjestelmä rajapinta
Kalman-suodin	Digitaalinen suodin, joka estimoi dynaamisen järjestelmän tilaa aikasemman mittausdatan perusteella.
API	Applications Programming Interface, ohjelmointirajapinta.
Stand-alone	Ohjelmisto joka ei vaadi käyttöjärjestelmätoimintoja toimiakseen.
Hyvyysluku	Värähtelijään varastoituneen energian ja yhdessä jaksossa häviävän energiamäärän suhde.
Vaihelukittu silmukka	Ohjausjärjestelmä, jonka ulostulosignaalin vaihe on suhteessa sisään tulevan signaalin vaiheeseen.
CMOS	Complementary Metal Oxide Semiconductor, kanavatransistoreihin perustuva mikropiiritekniikka.
Demoduloida	Alkuperäisen signaalinen erottaminen moduloidusta signaalista.

KUVIOLUETTELO

LIITELUETTELO

1 JOHDANTO

MEMS-anturien kehitys on mahdollistanut asentotiedon mittauksen kustannustehokkaalla tavalla. Tämä antaa mahdollisuuden toteuttaa joukon erilaisia sovelluskohteita, jotka eivät vielä joitain vuosia sitten olleet mahdollista tai taloudellisesti järkeviä. Tällaisia sovelluskohteita ovat esimerkiksi älypuhelimet, robotiikan sovellukset, kameroiden vakautusjärjestelmät, lennokkien vakautusjärjestelmät tai esimerkiksi pelikonsolien ohjausjärjestelmät. MEMS-laitteiden yleisesti pienestä koosta johtuen, voidaankin asentotiedon mittaus sulauttaa osaksi mitä tahansa järjestelemää suhteellisen vaivattomasti. Tästä hyvänä esimerkkinä toimivat älypuhelimet, joissa asentotiedonmittaus on osana useita toimintoja aina karttapalveluista kameran vakautukseen.

Työ tehtiin osana TJK Tietolaite Oy:n tuotekehitystä. Työn tavoitteena oli luoda luotettava ja alustariippumaton laiteajuri InvenSensen MPU6050-asentotietoanturille. Kehitettävä laiteajuri tuli osaksi Tietolaitteen T-Plat.E-ohjelmistoalustaa. T-Plat.E sisältää modulaarisen alustariippumattoman koodikirjaston, joka tarjoaa alustan ajurin kehittämiselle. Näin työssä voitiin keskittyä ajurin toiminnan kannalta olennaisempien osien kehittämiseen ja esimerkiksi I²C-ajurin kehittämiseen alustakohtaisesti ei ole tarvetta. Työhön kuului ajurin ohjelmistosuunnittelut sekä käytännön toteutus. Näiden lisäksi suoritettiin joukko mittauksia piirin sekä ohjelmiston suorituskyvyn takaamiseksi.

2 PROJEKTIN TAUSTA

2.1 TJK Tietolaite Oy

TJK Tietolaite Oy on Vaasassa sijaitseva sulautettujen järjestelmien tuotekehityspalveluja tarjoava yritys. Tietolaite sai alkunsa vuonna 1988 perustetusta toiminimestä. Toiminnan laajetessa, vuodesta 1994 alkaen yritysmuodoksi muutettiin osakeyhtiö. Yritys tuottaa sulautettuun ohjelmointiin ja elektroniikkasuunnitteluun liittyviä palveluja. Yrityksen toiminnan pääpaino on Suomessa mutta asiakkaita on myös ulkomailla.

2.2 Projektin tavoitteet

Ajurin suunnitteluprosessi aloitettiin kartoittamalla projektin tavoitteet. Projektin tavoitteiksi valikoitui mahdollisimman luotettavan ja tarkan asentotiedon tarjoava ajuri. Laitteen lisäominaisuudet jäivät toissijaisiksi tavoitteiksi, sillä niiden käyttöön ei tässä projektissa ollut tarvetta. Erityisen tärkeäksi nousi DMP-prosessorin käyttäminen. Suunniteltava ajuri tulee ensisijaisesti osaksi sulautetun järjestelmän ohjelmistoa, joissa rajattujen resurssien vuoksi laskentatehon säästäminen on tärkeää. Koska valmistajan tarjoamaa ohjelmistoa haluttiin käyttää DMP:n ohjauksessa, oli yksi tavoitteista valmistajan tarjoaman DMP-prosessorin ohjelmiston yksinkertainen päivittäminen.

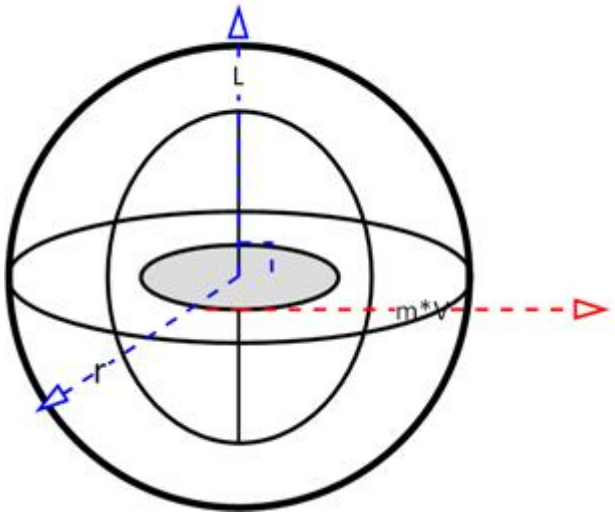
3 TEORIA

3.1 MEMS

Microelectromechnical systems (MEMS) on löyhästi määritelty termi järjestelmille, joita yleisesti kuvaa pieni koko. Tarkan määritelmän mukaan MEMS-laitteen mittasuhteiden tulisi olla mikrometriluokassa ja käyttää elektronisia että mekaanisia komponentteja. Kuitenkaan monet MEMS-laitteet eivät täytä tätä tarkkaa määritelmää. Esimerkiksi mikrofluidistiset kanavat eivät välttämättä sisällä ollenkaan elektronisia komponentteja. /1/

3.2 Gyroskooppi

Perinteinen gyroskooppi eli kulmanopeusanturi on pyörivästä levystä sekä kardaanisesti kiinnitetystä rengas systeemistä muodostuva mekaaninen laite. Gyroskoopin uloimmalla kehällä on yksi vapausaste, sisemmällä kehällä on kaksi vapausastetta ja pyörivällä levyllä on kolme vapausastetta. Näin pyörivä levy pystyy säilyttämään asentonsa riippumatta siitä mitä uloimmalla kehällä tapahtuu. Gyroskoopin levyn pyörimismomentti pyrkii siis vastustamaan pyörimismääränsä suunnanmuutosta ja näin säilyttää orientaationsa.



Kuvio 1. Gyroskoopin toimintaperiaate.

Gyroskooppien toiminta perustuu liikkeen säilymisen lakiin sekä kappaleiden pyörimisliikkeen säilymiseen. Pyörimismäärä on aina kohtisuoraan kulmanopeutta vasten:

$$L = J\omega \tag{1}$$

missä ω on kulmanopeus ja hitausmomentti J on:

$$J = m \times r^2 \tag{2}$$

Myös muulla tavalla toteutettuja laitteita kutsutaan gyroskoopeiksi. Nykyaikaisemmat gyroskoopit ovat optisia, esimerkiksi laser ja kuituoptiset gyroskoopit. Lasergyroskoopit mittaavat kahdesta vastakkaisesta suunnasta tulevien lasersäteiden erotusta. Kuituoptisissa gyroskoopeissa tarkkaillaan valon kulkua eri suuntiin kuitusilmukassa. Optiset gyroskoopit ovatkin tarkempia eivätkä kulu mekaanisesti samalla tavalla kuin perinteiset gyroskoopit. Tästä huolimatta ne ovat harvinaisempia, johtuen suuremmasta koosta ja korkeasta hinnasta verrattuna MEMS-gyroskooppeihin.

3.3 MEMS-gyroskooppi

Mikrosysteemi kulmanopeusanturit ovat kaikkein yleisin gyroskooppityyppi. Tämä johtuu niiden pienestä koosta, painosta sekä suhteellisen edullisesta hinnasta. MEMS-gyroskooppi onkin yleinen varuste, esimerkiksi nykyaikaisissa älypuhelimissa.

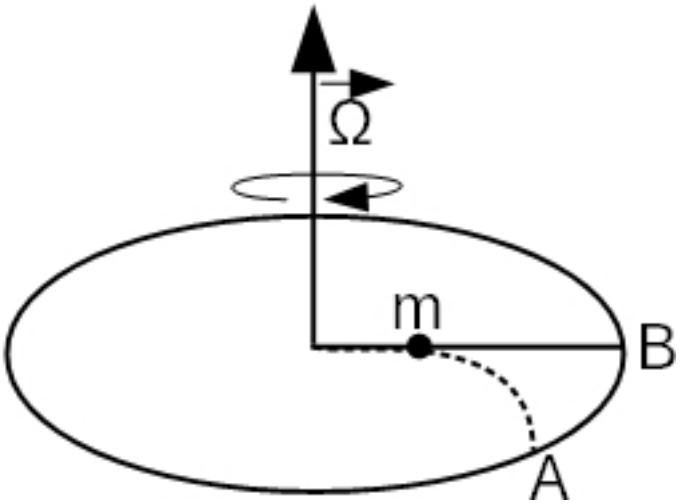
3.4 Coriolisvoima

Coriolisvoima vaikuttaa massaan, joka siirtyy pyörivän kappaleen keskipisteestä reunaa kohti. Kappaleen kulkiessa keskipisteeltä reunalle näennäisesti suoraa linjaa, saavuttaa se kappaleen reunalla eri pisteen kuin mitä alkuperäinen määränpää oli. Kuviossa 2 massa m liikkuu keskeltä kohti pyörivän kappaleen reunaa. Saapuessaan reunalle, se ei saavuta pistettä B vaan pisteen A, johtuen coriolisvoimasta.

$$\overrightarrow{X''} = -2\vec{\Omega} \times \overrightarrow{X'} \quad (3)$$

missä $\overrightarrow{X'}$ on pyörivän kappaleen kiihtyvyysvektori ja $\vec{\Omega}$ on kappaleen kulmanopeusvektori. Newtonin lain mukaan coriolisvoima $\vec{F_C}$ saadaan kertomalla corioliskiihtyvyys kappaleen massalla.

$$\overrightarrow{F_C} = m\overrightarrow{X'} = -2m\vec{\Omega} \times \overrightarrow{X'} \quad (4)$$



Kuvio 2. Coriolis-ilmiön periaate.

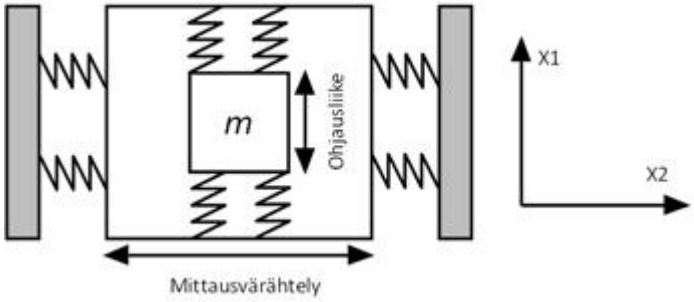
3.5 Värähtelevän rakenteen gyroskooppi

Värähtelevän rakenteen gyroskooppi, eli coriolis-värähtelevä gyroskooppi, on kulmanopeusanturin tyyppi, joka käyttää värähtelevää rakennetta kulmanopeuden laskemiseen. Tämä perustuu siihen, että värähtelevä objekti jatkaa värähtelyä samassa tasossa, vaikka ulkoinen runko ympärillä muuttaisikin asentoaan. Coriolis-ilmiö aiheuttaa sen, että objekti aiheuttaa voiman ulkoiseen runkoon. Mittaamalla tätä voimaa voidaan laskea rungon pyörimisnopeus.

3.6 Värähtelevä kaksitilainen gyroskooppi

Lähes kaikki MEMS-gyroskoopit perustuvat kahden ortogonaalisesti asetetun tunnetun massan värähtelyn tilaan, joka on havainnollistettu kuviossa 3. Ortogonaalisuudesta johtuen, tilat eivät ole keskenään interaktiossa. Ohjaustilan

massan liike ei aiheuta liikettä mittaustilan suunnassa. Resonaattori virittyy värähtelemään ohjaustilassa suunnassa X1, mutta kulmataajuuden pysyessä nollassa, liike mittaustilassa pysyy nollassa. Coriolisvoima virittää resonaattorin suunnassa X2, jos kulmataajuus nousee X1 akselilla. Näin ollen mittaustilan massan värähtelyn amplitudi on verrannollinen kulmataajuuteen.



Kuvio 3. Kaksitilaisen gyroskoopin toimintaperiaate.

Ohjaus- (X_d) ja mittaustilan (X_s) siirtymän yhtälöt ovat

$$X_d = H_d F_d \tag{5}$$

missä, H_d ohjaustilan vastefunktio ja F_d on ulkoinen voima, joka vaikuttaa ohjaustilan liikkeeseen.

$$X_s = \in_Q X_d + H_s F_c \tag{6}$$

missä H_s on mittaustilan vaste ja $\in_Q$ on fraktio ohjaustilan värähtelystä, joka on interaktiossa mittaustilan kanssa johtuen kvadratuurivirheestä. F_c on coriolisvoima joka virittää mittaustilan.

$$F_C = -2m_d \Omega X'_d \tag{7}$$

jossa m_d on massa m rungon sisällä eli ohjaustilan massa, Ω on kulmataajuus ja X'_d on ohjaustilan massan kiihtyvyys.

$$H_d(\omega) = \frac{1}{m_d} \frac{1}{\omega^2{}_{0d} - \omega^2 + j\omega\omega_{0d}/Q_d} \equiv H_{Rd} + jH_{id} \quad (8)$$

$$H_s(\omega) = \frac{1}{m_s} \frac{1}{\omega^2{}_{0s} - \omega^2 + j\omega\omega_{0s}/Q_s} \equiv H_{Rd} + jH_{id} \quad (9)$$

missä m_x on massa, Q_x on hyvyysluku, ω_{0x} on resonoiva taajuus. Sekä ohjaus- että mittaustilan resonaattorit värähtelevät viritystaajuudessa ω, mutta niiden amplitudi ja vaihe eroavat toisistaan

Useimmissa gyroskoopeissa mittaussignaalia mitataan vaihelukitulla silmukalla, jolloin sekä signaalinvaihe että amplitudi ovat tärkeitä.

3.7 Kapasitiivinen gyroskooppi

Lähes kaikki kaupalliset gyroskoopit perustuvat kapasitiiviseen mittaukseen ja aktuaatioon. Laite otetaan käyttöön kapasitiivisen aktuaattorin avulla mahdollisimman suuren värähtelyn amplitudin saavuttamiseksi. Coriolisvoiman ollessa suoraa verrannollinen värähtelyn nopeuteen, suuri amplitudi helpottaa pienten kulmanmuutosten mittausta.

Kapasitiiviset aktuaattorit vaativat suuren etujännitteen, vaikka laite viritettäisiin resonanssitilassa. Toinen huomionarvoinen seikka on, että suurellakin värähtelyn amplitudilla ja kulman muutoksella coriolisvoimat jäävät pieniksi. Näin coriolisvoiman havainnointi on vaikeaa ja heikko signaali katoaa helposti kohinan tai muiden häiriöiden sekaan. /2/

3.8 Kvadratuurivirhe

Ideaalisessa tilanteessa ohjaus- ja mittaustilat ovat täydellisen ortogonaaliset, jolloin ilman coriolisvoiman vaikutusta mittaustila ei pääse virittymään. Käytännössä kuitenkin valmistuksen varianssista johtuen, tapahtuu interaktioita tilojen välillä. Tällöin esimerkiksi ohjaustila ei ole täydellisesti suunnassa X1 ja sen liikkuessa tapahtuu liikettä myös X2 suunnassa. Tätä virhettä kutsutaan kvadratuurivirheeksi.

Nimi kvadratuurivirhe tulee coriolis-signaalin ja kvadratuurivirheen vaihe-erosta. Ideaalinen kvadratuurivirhe olisi 90º vaihe-ero, tällöin signaalit voitaisiin erottaa toisistaan synkronisella demodulaatiolla. Myös suuri kvadratuurisignaali on ongelmallinen, sillä se asettaa kohtuuttomat vaatimukset signaalin vahvistamiselle. Tällöin on olennaista minimoida kvadratuurivirhe jo mittauselementillä.

Piipohjaisissa gyroskoopeissa kvadratuurivirhettä kompensoidaan sähköisen säädön avulla käyttäen hyväksi sähköistä jousivoimaa levykondensaattoreissa. Lineaarinen kapasitiivinen jousivoima saadaan yhtälöstä:

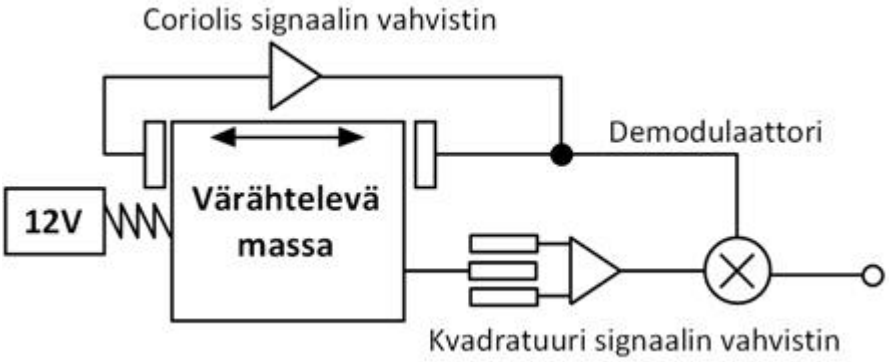
$$k_{0e} = \frac{V^2{}_{dc} C_0}{d_2} \tag{10}$$

missä C_0 on kapasitanssi ja d on elektrodien väli. Säätämällä etujännitettä V_{dc} voidaan sähköistä jousivoimaa säätää kompensoimaan mekaanisten jousien varianssia. /3/

3.9 Mittauspiiri

Mittauspiiriltä vaaditaan useita ominaisuuksia kuten värähtelyn ylläpidon, coriolis-signaalin- ja lämpötilanmittauksen, korkean etujännitteen luomisen sekä lämpötilan kompensointi-, ohjaus- ja kalibrointilogiikan.

Coriolis-signaalin vahvistamiseksi käytetään transimpedanssivahvistinta muuntamaan pieni virta jännitteeksi. Korkean vahvistuksen lisäksi tulee vahvistimen toimia laajalla alueella kvadratuurivirheen sietämiseksi. Myös signaalin vaihetta tulee ylläpitää coriolis ja kvadratuurisignaalien erottamiseksi.



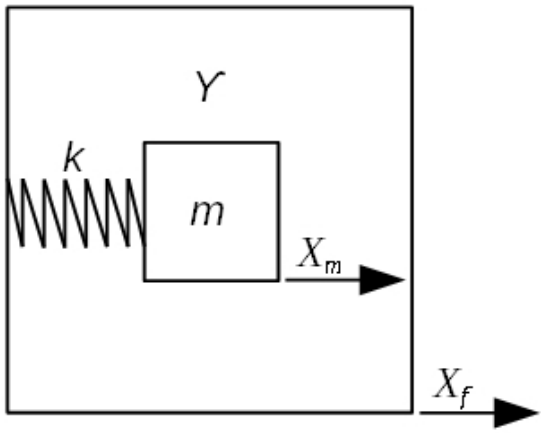
Kuvio 4. Yksinkertainen gyroskoopin mittauspiiri.

Kvadratuuri ja coriolis-signaalien erottamiseksi vahvistimen jälkeinen signaali kerrotaan ohjaus-signaalilla ja suodatetaan matalapäästösuotimella coriolis-signaalin demoduloimiseksi.

Myös lämpötilavaihtelun vaikutukset gyroskoopin mittaustarkkuuteen tulee ottaa huomioon. Tätä varten mittauspiirin tulee sisältää lämpötila-anturi, jonka avulla voidaan laskea lämpötilavaihtelun vaikutusta kun anturi on kalibroitu useille eri lämpötila-alueille.

3.10 Kiihtyvyysanturi

Kiihtyvyysanturit ovat sähkömekaanisia laitteita, jotka mittaavat kappaleen kiihtyvyyttä. Mitattava liike voi olla staattista, kuten maan vetovoima, tai dynaamista, kuten anturin heiluttaminen. Suurin osa kiihtyvyysantureista perustuu pietsosähköisiin kiteisiin. Useimmat nykyaikaiset kiihtyvyysanturit ovat MEMS-antureita. /4/



Kuvio 5. Kiihtyvyysanturin toimintaperiaate.

3.11 Kiihtyvyysanturin toimintaperiaate

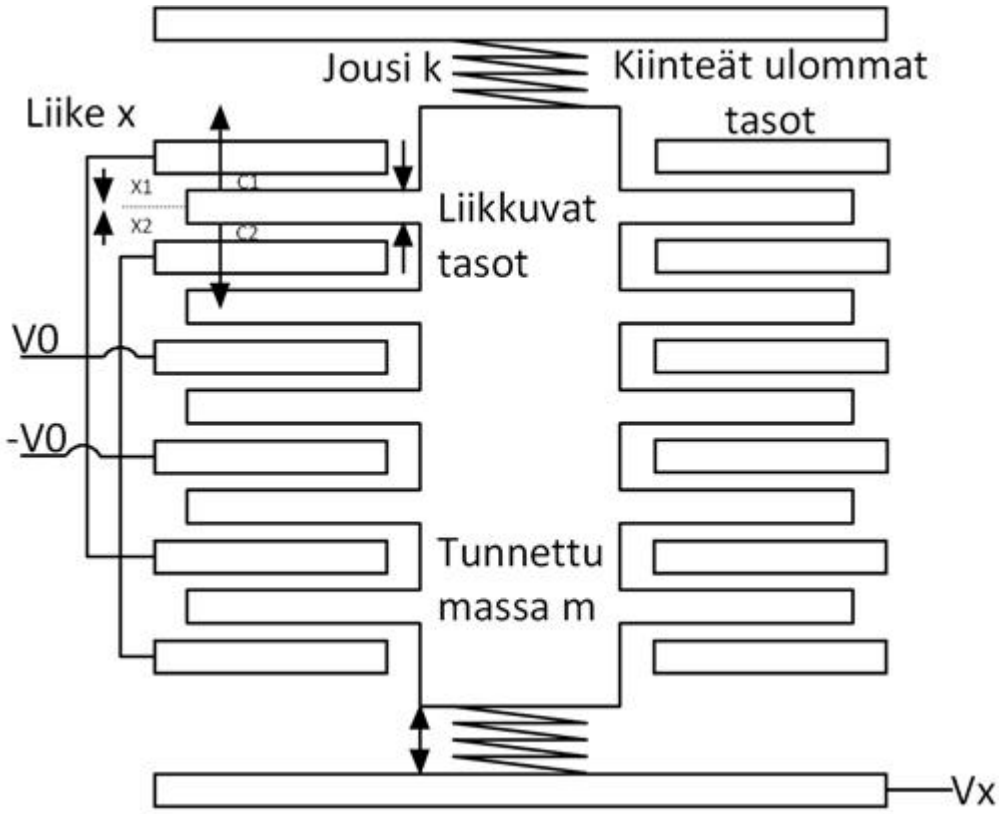
Kiihtyvyysanturit muodostuvat yleisesti neljästä pääosasta. Nämä ovat tunnettu massa m, joka on kiinnitetty jousen k avulla runkoon. Resonoinnin välttämiseksi massaa m on hidastamassa kaasu tai öljy *Y*. Yhdellä tunnetulla massalla m voidaan mitata useita eri akseleita yhtäaikaisesti, mutta useimmissa useita akseleita mittaavissa antureissa käytetään yhtä massaa yksittäistä akselia kohden.

3.12 Pietsoresistiivinen mittaus

Pietsoresistiivisessä mittauksessa jouseen *k* integroidaan pietsoresistiivistä materiaa, jolloin jousen liikkuessa sen resistanssi muuttuu. Mittamaalla jousen resistanssin muutosta voidaan kiihtyvyys päätellä. Pietsoresistiiviset anturit ovat vakaita ja helppoja toteuttaa. Ne kuitenkin tuottavat enemmän kohinaa ja eivät ole yhtä pienitehoisia kuin muut mittaustekniikat.

3.13 Kapasitiivinen mittaus

Kapasitiivinen mittaus perustuu kapasitanssin muutoksiin tunnetun massan liikkuessa suhteessa runkoon. Kapasitiiviset anturit ovat yleisimmin käytetty anturityyppi, sillä niissä on hyvä kohinasuhde ja alhainen virrankulutus.



Kuvio 6. Kapasitiivisen kiihtyvyysanturin periaate.

Tyypillinen MEMS-kiihtyvyysanturi muodostuu liikkuvasta tunnetusta massasta, joka on mekaanisesti kiinnitetty referenssi pisteenä toimivaan runkoon jousilla k. Ulkoiset kiinteät tasot ja sisemmät liikkuvat tasot muodostavat kondensaattorin. Tunnetun massan liike mitataan tarkastelemalla kapasitanssin muutosta tasojen välissä. Tasojen väliin jäävä vapaan-tilan kapasitanssit C_1& C_2, ovat suoraa sidoksissa tunnetun massan siirtymiin x_1& x_2.

$$C_1 = \in_A \frac{1}{x_1} = \in_A \frac{1}{d+x} = C_0 - \Delta C$$
$$C_2 = \in_A \frac{1}{x_2} = \in_A \frac{1}{d-x} = C_0 + \Delta C \qquad (11)$$

, missä A on elektrodien pinta-ala, d on niiden välimatka ja $\in$ on väliaineen permittiivisyys. Jos kiihtyvyyttä ei ole, kapasitanssit C_1 ja C_2 ovat yhtä suuret, koska $x_1 = x_2$. Tunnetun massan siirtymä johtuu kiihtyvyyden kasvusta. Jos $x \neq 0$ niin kapasitanssin muutos on:

$$C_2 - C_1 = 2\Delta C = 2 \in_A \frac{x}{d^2 - x^2} \tag{12}$$

Tästä mittaamalla ΔC, voidaan x:n siirtymä selvittää yhtälöstä

$$\Delta C x^2 + \in_A x - \Delta C d^2 = 0 \tag{13}$$

Pienille siirtymille termin ΔCx^2 merkitys on vähäpätöinen, joten se voidaan jättää huomioimatta.

$$x \approx \frac{d^2}{\in_A} \Delta C = d \frac{\Delta C}{C_0} \tag{14}$$

Tästä nähdään että siirtymä on verrannollinen kapasitanssin muutokseen ΔC./5/

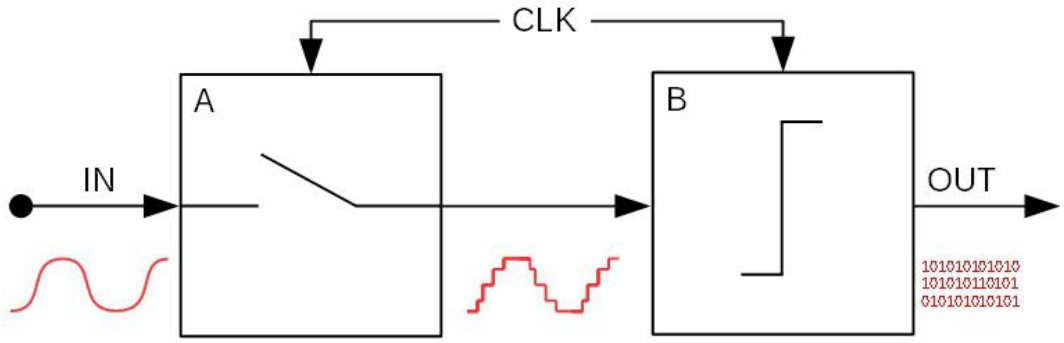
3.14 Pietsosähköinen mittaus

Pietsosähköinen mittaus perustuu latauksen purkautumiseen pietsosähköisissä materiaaleissa kun massan liike aiheuttaa niissä rasitusta. Pietsoelektroninen materiaali generoi virran, joka on suhteellinen kiihtyvyyden muutokseen. Pietsosähköistä mittausta käytetään usein makroskooppisissa sovelluksissa, mutta ei yleisesti mikrolaitteissa./5/

3.15 A/D-muunnin

A/D-muunnin muuntaa analogisen signaalin digitaaliseksi viestiksi.

A/D-konversio voidaan jakaa kahteen osaan, näytteenottoon ja kvantisointiin. Näytteenotto A muuntaa jatkuvan signaalin diskreetiksi signaaliksi jonka jälkeen kvantisointi B muuttaa jatkuvan amplitudi jakauman diskreeteiksi tasoiksi jotka voidaan ilmaista digitaalisessa muodossa. /6/



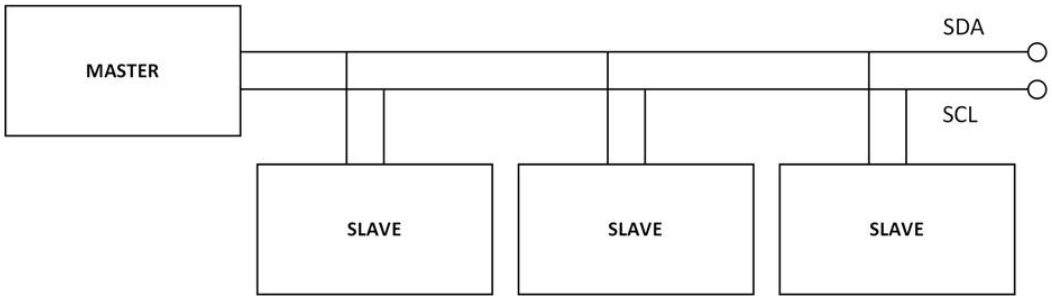
Kuvio 7. A/D-muuntimen toimintaperiaate.

3.16 Digital Motion Processor

Digital Motion Processor on MPU6050-piiriin sulautettu prosessori joka suorittaa liikkeen prosessointialgoritmejä pääprosessorin puolesta. DMP vastaanottaa dataa kiihtyvyysantureilta, gyroskoppeilta sekä mahdollisilta kolmannen osapuolen antureilta ja prosessoi vastaanotetun datan. Prosessoitu data voidaan lukea suoraa DMP:n rekistereistä tai se voidaan puskuroida FIFO-puskuriin. DMP voi myös generoida keskeytyksen yhteen piirin ulkoisista pinneistä.

DMP:n tarkoitus on vähentää kuormitusta sekä aikavaativuudessa että pääprosessorin laskennasta. Tyypillisesti liikkeen prosessointialgoritmit pitäisi suorittaa korkealla tahdilla tarkkojen tuloksien takaamiseksi matalalla viiveellä. Tätä vaaditaan, vaikka sovellus päivittäisi arvoja paljon hitaammin. DMP-prosessoria voidaankin käyttää tehon tarpeen minimointiin, ajoituksien yksinkertaistamiseen, ohjelmistoarkkitehtuurin yksinkertaistamiseen ja pääprosessorin resurssien säästämiseen./7/

3.17 I²C-väylä



Kuvio 8. I²C-väylän topologia

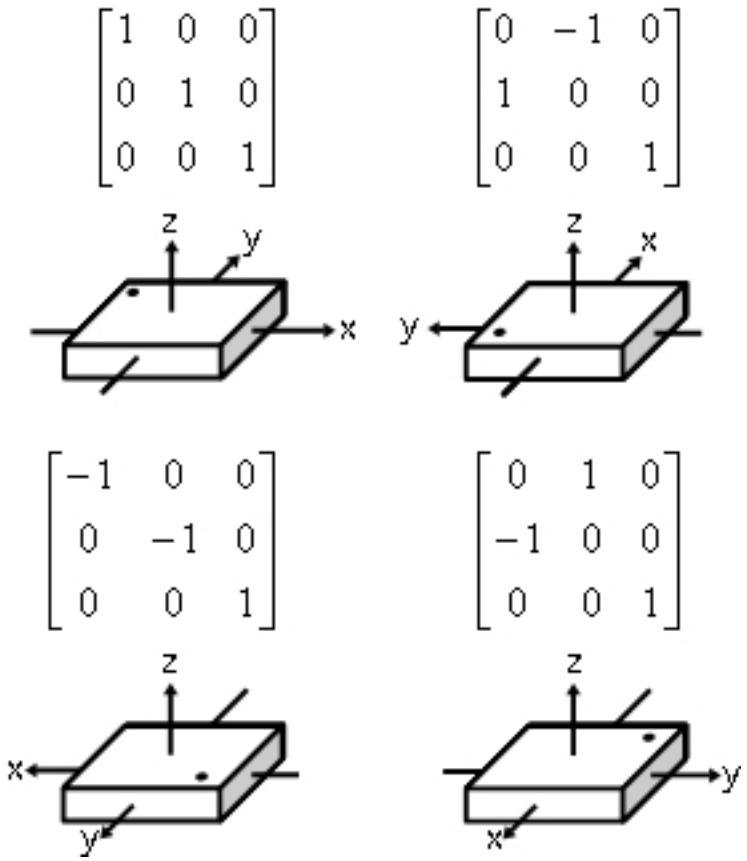
I²C on yksinkertainen kaksisuuntainen tiedonsiirto- ja ohjausväylä. Se muodostuu kahdesta johtimesta: data- ja kellolinjasta. Jokaisella väylän laitteella on oma yksilöllinen osoitteensa ja jokainen laite voi toimia joko lähettäjänä tai vastaanottajana. Väylällä on aina vähintään yksi Master-laite ja yksi Slave-laite. Master-laite tarjoaa kellopulssin ja aloittaa viestien lähettämisen, Slave-laite ei voi aloittaa keskustelua ollessaan Slave-tilassa. Yhdellä väylällä voi olla myös useita Master-laitteita. /8/

3.18 Digitaalinen matalapäästösuodatus

Digitaalinensuodin, joka suorittaa mikroprosessorilla matemaattisia operaatioita diskreetille signaalille. Yleensä suodin muodostuu A/D-muuntimesta, mikroprosessorista sekä lisälaitteista. Mikroprosessorin ohjelmisto toteuttaa suotimen ohjelmallisesti suorittamalla matemaattisia operaatioita A/D-muuntimelta tulevalle diskreetille signaalille.

3.19 Orientaatiomatriisi

Anturin orientaatio ajurille määritellään alustuksen aikana orientaatiomatriisina. Orientaatiomatriisit tai rotaatiomatriisit ovat yhdeksän komponentin neliömatriiseja ,jotka määrittävät kappaleen asennon kolmiulotteisessa vektoriavaruudessa.



Kuvio 9. Anturin orientaatiot matriiseina kun z-akseli osoittaa ylöispäin.

Rotaatio z-akselin ympärillä määritellään:

$$R_z = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

Rotaatio y-akselin ympärillä määritellään:

$$R_y = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{16}$$

Rotaatio x-akselin ympäri määritellään:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \quad (17)$$

Joiden avulla voidaan selvittää anturin orientaatio matriisikertolaskulla

$$R_z \cdot R_y \cdot R_x = R_{orientaatio} \quad (18)$$

Esimerkiksi jos kuvataan anturin kääntämistä 90° asteen kiertoa Z-akselin ympäri saadaan

$$R_{orientaatio} = \begin{bmatrix} \cos 90^\circ & -\sin 90^\circ & 0 \\ \sin 90^\circ & \cos 90^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos 0^\circ & 0 & -\sin 0^\circ \\ 0 & 1 & 0 \\ \sin 0^\circ & 0 & \cos 0^\circ \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 0^\circ & -\sin 0^\circ \\ 0 & \sin 0^\circ & \cos 0^\circ \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (19)$$

3.20 Gyroskoopin liukuma

Gyroskoopin liukuma aiheutuu yhdistelmästä useita tekijöitä. Siihen vaikuttavat lämpötila, skaalauskerroin, mekaaninen varianssi sekä esiasetettu korjaus. Pienikin virhe näissä osissa gyroskoopin funktiota aiheuttaa liukumaa pitkällä aikavälillä, johtuen kerroinvaikutuksesta.

Esimerkiksi 0,0001° asteen liukuma 200 Hz näytteenottotaajuudella kertautuu 60 minuutin aikana 72° asteen mittausvirheeksi.

$$G_{virhe} = 0{,}0001^\circ \cdot 200 \cdot 60 \cdot 60 = 72^\circ \quad (20)$$

Koska lyhyellä aikavälillä pieni virhe lasketaan mukaan jokaiseen edelliseen asentotietoon.

3.21 Anturifuusio

Anturifuusion keskeinen idea on yhdistää useamman anturin mittauksia yhdessä siten, että ne tukevat toisiaan. Nämä mittaustulokset käsitellään algoritmilla joka yhdistää kummankin mittaustuloksen, esimerkiksi painotettu keskiarvo kahdesta

eri mittauksesta tai Kalman-suodin. Tällä tavalla voidaan suodattaa mittaustuloksista virheitä ja tällä tavalla parantaa mittaustarkkuutta.

MEMS-gyroskoopit ovat alttiita liukumalle ja kiihtyvyysanturit ovat alttiita mekaaniselle kohinalle. Mittaustulosten parantamiseksi voidaan käyttää yhdistelmää useamman anturin mittaustuloksista eli anturifuusiota. Tässä projektissa gyroskoopin liukumaa kompensoidaan käyttämällä anturifuusiotilaa DMP-prosessorilla. DMP-prosessorin anturifuusiokäsittely käyttää gyroskoopin mittaustulosten tukena kiihtyvyysanturin mittaustuloksia. Näin saadaan tarkempi arvio anturin orientaatiosta. /9/

DMP-prosessorin käyttämää laskenta-algoritmiä ei ole dokumentoitu InvenSensen toimesta, joten sen käyttämästä laskenta-algoritmista ei ole varmuutta.

3.22 Kvaternio

Kvaterniot ovat nelikomponenttinen kompleksilukujen laajennus, jonka avulla voidaan kuvata kappaleen asentoa kolmiulotteisessa avaruudessa. Niitä voidaan kuvata nelikomponenttivektoreina, skalaarina ja kolmikomponenttivektorina tai kompleksilukuna, jolla on kolme eri imaginääriosaa.

Kvaterniota, jonka skalaariosa on 0, kutsutaan puhtaaksi kvaternioksi, vektori kvaternioksi tai vektoriksi. Kvaterniota, jonka vektoriosa on 0, kutsutaan skalaari kvaternioksi tai pelkäksi skalaariksi.

Esimerkki kvaterniosta kompleksilukunotaatiosta:

$$\overset{\circ}{q} = q_0 + iq_x + jq_y + kq_z \tag{21}$$

Missä q_0, q_x, q_y, q_z ovat reaalilukuja ja *i, j* ja *k* ovat imaginäärikomponentit, joilla on seuraavat ominaisuudet:

$$\begin{aligned} &i^2 = j^2 = k^2 = -1 \\ &ij = k, jk = i, ki = j \\ &ji = -k, kj = -i, ik = -j \end{aligned} \tag{22}$$

Kvaternion konjugaatti on

$$\overset{\circ}{q}^{*} = q_0 - iq_x - jq_y - kq_z \tag{23}$$

Kvaternion ristitulo on

$$(\overset{\circ}{p}\cdot\overset{\circ}{q}) = p_0q_0 + p_xq_x + p_yq_y + p_zq_z = (\overset{\circ}{p}\overset{\circ}{q}^{*} + \overset{\circ}{p}\overset{\circ}{q}^{*})/2 \tag{24}$$

Kvaternion itseisarvon toinen potenssi on pistetulo ko. kvaternion itsensä kanssa.

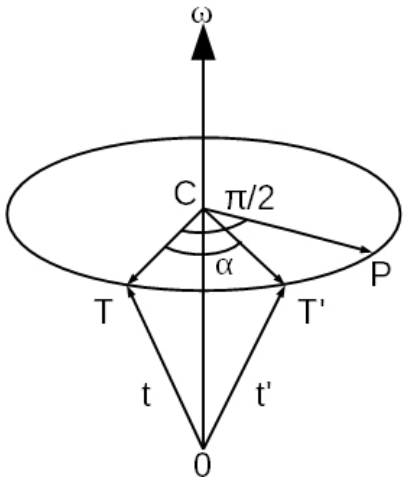
$$\left|\overset{\circ}{q}\right|^2 = \overset{\circ}{q}\cdot\overset{\circ}{q} \tag{25}$$

Kvaternioiden pituus saadaan kaavalla

$$Q_{pituus} = \sqrt{{q_0}^2 + {q_1}^2 + {q_2}^2 + {q_3}^2}$$

Yksikkö kvaternioilla tämän pituuden tulisi olla 1.

3.23 Kappaleen kierto kvaternioiden avulla



Kuvio 10. Vektorin t rotaatio vektoriksi t'.

Käyttämällä kvaternioiden skaalarin ja vektorin yhdistelmänotaatiota käsitellään kuvion 10 vektorin t kulman α kokoinen rotaatio yksikkövektorin ω ympärillä. Määritetään vektorin t projektio yksikkövektorin ω suuntaisesti.

$$\vec{OC} = (t \cdot \omega)\omega \tag{26}$$

Tästä voidaan päätellä että

$$\vec{CT} = t - (t \cdot \omega)\omega \tag{27}$$

$$\vec{CP} = \omega \times t \tag{28}$$

Jolloin $\vec{CT}'$ voidaan laskea

$$\vec{CT}' = \cos\alpha\,\vec{CT} + \sin\alpha\,\vec{CP} \tag{29}$$

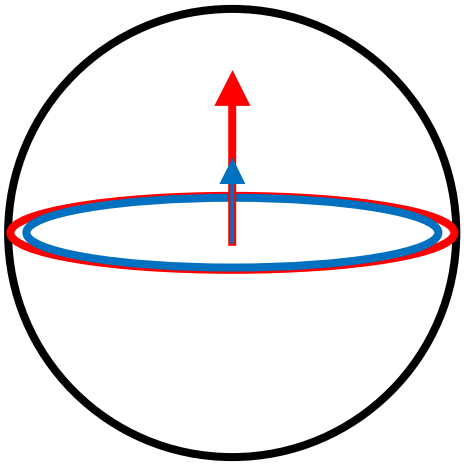
Jonka jälkeen voidaan laskea rotaation jälkeinen vektori t’

$$t' = \vec{OC} + \vec{CT}' \tag{30}$$

Vertaamalla täsmäävätkö vektoreiden t ja t’ ristitulo voidaan varmistaa, että kyseessä on rotaatio./10/

3.24 Gimbalin lukko

Gimbalin lukko on tilanne jossa yksi vapausasteista menetetään kun kaksi pyörivää akselia osoittaa samaan suuntaan kolmesta gimbaalista muodostuvasta laitteessa.



Kuvio 11. Gimbalin lukon periaatekuva.

Esimerkki ilmiöstä rotaatio matriiseina. Jos kulma $\beta = \frac{\pi}{2}$ ja $R_{lukko} = R_x \cdot R_y \cdot R_z$:

$$R_{lukko} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \cdot \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \cdot \begin{bmatrix} \cos\nu & -\sin\nu & 0 \\ \sin\nu & \cos\nu & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{31}$$

Laskemalla $\cos\frac{\pi}{2} = 0$ ja $\sin\frac{\pi}{2} = 1$ kääntyy yhtälö muotoon:

$$R_{lukko} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \cos\nu & -\sin\nu & 0 \\ \sin\nu & \cos\nu & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{32}$$

Josta matriisin kertolaskun jälkeen saadaan

$$\begin{aligned} R_{lukko} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \cdot \begin{bmatrix} \cos\nu & -\sin\nu & 0 \\ \sin\nu & \cos\nu & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ R_{lukko} &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & \sin\alpha\cos\nu + \cos\alpha\sin\nu & -\sin\alpha\sin\nu + \cos\alpha\cos\nu \\ 0 & -\cos\alpha\cos\nu + \sin\alpha\sin\nu & \cos\alpha\sin\nu + \sin\alpha\cos\nu \end{bmatrix} \end{aligned} \tag{33}$$

$$R_{lukko} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & \sin(\alpha+\nu) & \cos(\alpha+\nu) \\ 0 & -\cos(\alpha+\nu) & \sin(\alpha+\nu) \end{bmatrix} \tag{34}$$

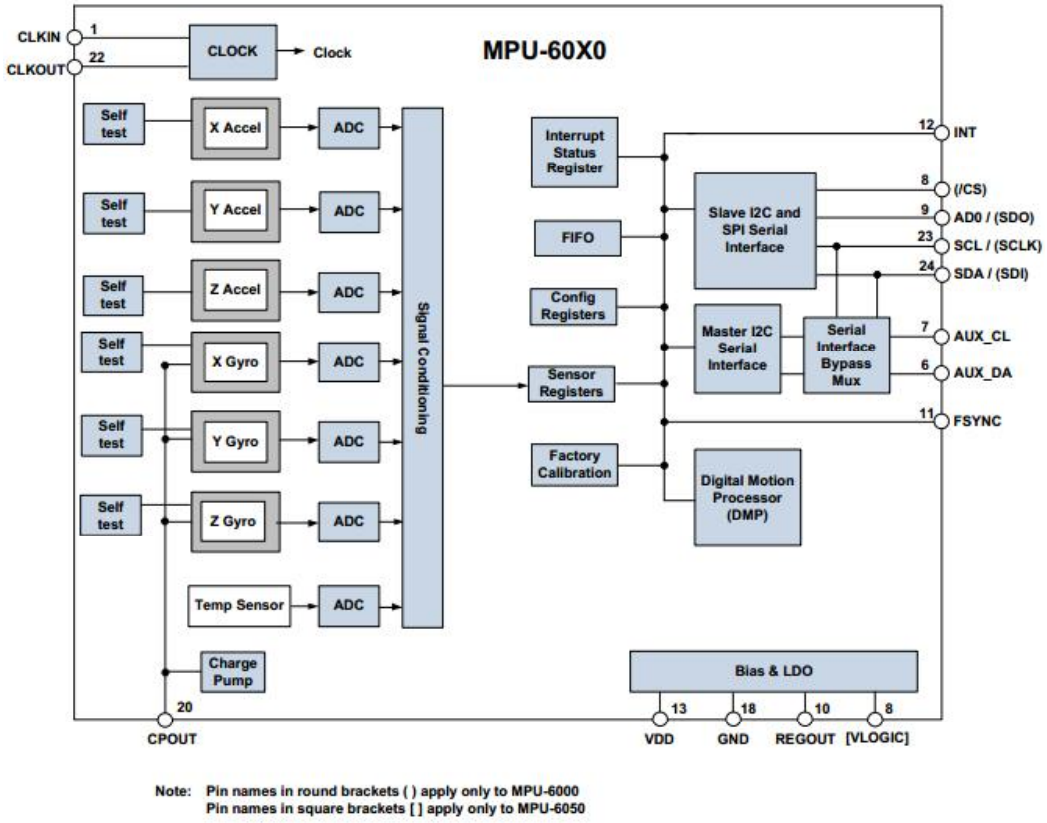
Jolloin α tai ν muutos ei vaikuta vaan orientaatio pysyy z-akselin suuntaisena. Ainut tapa vaikuttaa orientaatioon on muuttaa β arvoa.

3.25 Muita tapoja laskea kappaleen asento

Kappaleen asento 3-ulotteisessa avaruudessa voidaan laskea myös ilman kvaternioita. Kappaleen asennon laskenta käyttämällä Euler-kulmia tai kiertomatriiseja on täysin mahdollista. Niiden käyttäminen on kuitenkin laskennallisesti hitaampaa ja ne ovat alttiita Gimbalin lukolle.

4 MPU6050-ASENTOTIETOANTURI

MPU6050 on QFN-pakattu 4*4*0,9 mm kokoinen asentotietoanturi. Se sisältää 3-akselisen gyroskoopin, 3-akselisen kiihtyvyysanturin, lämpötila-anturin, 6-kappaletta 16-bittisiä A/D-muuntimia, ohjelmoitavan digitaalisen alipäästösuotimen, 1024-tavun FIFO-puskurin, I^2C-väylän sekä asentotietolaskentaa suorittavan Digital Motion Processor- prosessorin. Piirissä on myös mahdollisuus laajentaa mittausta 9-akseliin käyttämällä kolmannen osapuolen antureita, esimerkiksi kompassia.



Kuvio 12. MPU6050 lohkokaavio.

Se sisältää 3-akselisen konfiguroitavan MEMS-gyroskoopin. Yksittäisellä tunnetulla massalla mitataan yksittäistä akselia. Jokaisella akselilla on perässään erillinen 16-bittinen A/D-muunnin mittaustarkkuuden takaamiseksi. Mittaustulokset voidaan lukea suoraan rekisteristä, puskuroida FIFO-puskuriin tai viedä DMP-prosessorille, konfiguraatiosta riippuen. Gyroskooppi on

konfiguroitavissa neljään eri mittausherkkyystilaan välillä ±250 - ±2000 astetta per sekuntti.

MPU6050 sisältää matalatehoisen MEMS-kiihtyvyysanturin. Kiihtyvyysanturi on konfiguroitavissa neljään eri tilaan välillä ±2g - ±16g. Mitattaville akseleille integroidut A/D-muuntimet mahdollistavat jatkuvan näytteenottamisen ilman tarvetta erilliselle multiplekserille. Kiihtyvyysanturi mahdollistaa asennon päättelyn sekä koputuksen havaitsemisen.

DMP-prosessori suorittaa asentotietolaskentaa arvoista, jotka saadaan A/D-muuntimilta. Se muuntaa 16-bittiset mittausarvot valmiiksi kvaternioiksi ja näin ollen laskee kuormitusta sovelluksen pääprosessorilta. DMP-prosessorin laskemat arvot voidaan lukea suoraan rekisteristä tai ne voidaan puskuroida FIFO-puskuriin I^2C-väylän liikenteen tarpeen vähentämiseksi. Näin voidaan välttää väylän ruuhkautumista useampia mittauslaitteita sisältävissä sovelluksissa.

I^2C-väylä on pääasiallinen tapa kommunikoida MPU6050-asentotietoanturin kanssa. MPU6050 tarjoaa kolme erilaista I^2C-toimintoa. Se osaa toimia sekä Master- että Slave-tilassa. Näiden lisäksi on erillinen läpikulkutila, joka mahdollistaa ulkoisen prosessorin ja ulkoisen anturin kommunikaation MPU6050:n I^2C-väylän kautta.

5 AJURIN SUUNNITTELU

5.1 Vaatimusmäärittely

Ajurin tuli pystyä initialisoimaan MPU6050 ja sen lisälaiteet, erityisesti DMP-prosessori sen tarjoamien hyötyjen saavuttamiseksi. Ajurin on pystyttävä käyttämään InvenSensen tarjoamaa valmista ohjelmistoa DMP-prosessorille siten, että se on päivitettävissä uusimpaan versioon siirtämällä uusi lähdekoodi ajurin projektikansioon. Näin siksi, että InvenSense ei ole dokumentoinut DMP-prosessoria eikä sen toimintaa kuin rajapintafunktioiden osalta.

Koodin on oltava alustariippumatonta ja ajurin tulee toimia osana Tietolaitteen T-Plate-ohjelmistoalustaa. Nämä kaksi vaatimusta täydentävätkin toisiaan siten, että T-Plat.e on alustariippumaton alusta ja näin ollen sille kirjoitettu koodi on alustariippumatonta. Vaikka alkuperäinen suunnitelma olikin kirjoittaa stand-alone-ajuri, huomattiin suunnittelun aikana, että ajankäytön kannalta on tehokkaampaa kirjoittaa ajuri suoraan osaksi T-Plat.e kirjastoa.

5.2 T-Plat.e

T-Plat.E on Tietolaite Oy:n alustariippumaton ohjelmistoalusta sulautettujen järjestelmien ohjelmistokehitykseen. Se pitää sisällään niin käyttöjärjestelmä-rajapinnan kuin koodikirjaston sekä näiden integraation. Sen universaali ohjelmointirajapinta pitää huolen siitä, että kehitettävät ohjelmistot ovat helposti siirrettävissä alustalta toiselle ilman muutoksia itse ohjelmistoon.

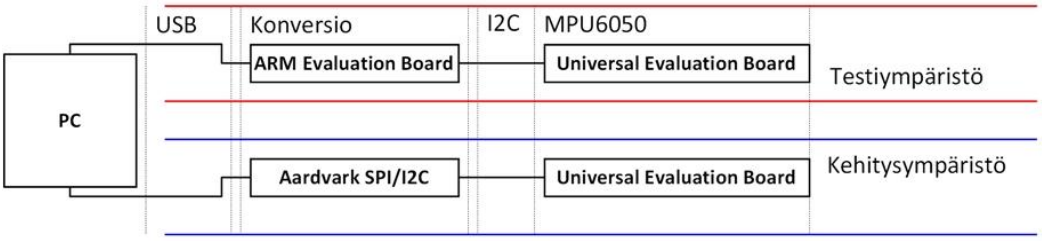
T-Plat.E:n päälle rakennettu projekti muodostuu sekä käyttöjärjestelmästä että mukaan halutuista ohjelmistokomponenteista. Ohjelmistokomponentit voivat pitää sisällään esimerkiksi erilaisia kommunikointi protokollia, ajureita sekä muita yleispäteviä ohjelmistokomponentteja. Käyttöjärjestelmä hoitaa alla omat tehtävänsä, kuten moniajon suoritusvuoron varaamisen. Sen päällä pyörivät modulit omina taskeinaa tai co-taskeinaan. Kommunikointi käyttöjärjestelmän kanssa tapahtuu OSA:n (Operating System API) kautta. Näin alla oleva käyttöjärjestelmä voidaan vaihtaa ilman muutoksia ylemmän tason ohjelmistoon.

5.3 Kehitysympäristö

Kehityksessä käytettiin on InvenSensen Universal Evaluation Board- ja ARM Evaluation Board-kehitysalustoja.

Universal Evaluation Board-kehitysalusta sisältää MPU6050-asentotietoanturin sekä pinnityksen sen kanssa kommunikointia varten. Se sisältää myös jännitteen säädön ulkoisesta $5V_{dc}$ anturille sopivaksi 3,6 V_{dc}.

ARM Evaluation Board muuntaa USB-väylän liikenteen I^2C-väylälle, ottaa käyttöjännitteen USB-väylältä ja on yhteensopiva InvenSensen omien ohjelmistojen kanssa.



Kuvio 13. Kehitys- ja testausympäristö.

Piirin toimivuuden varmistamiseksi kehityksessä käytettiin myös InvenSense Universal Datalogger-ohjelmistoa testauksessa. Käyttämällä valmistajan tarjoamaa kehitysalusta sekä testiohjelmistoa, voidaan varmistaa, että kehityksessä käytössä oleva MPU6050-asentotietoanturi toimii ja antaa valideja mittausarvoja.

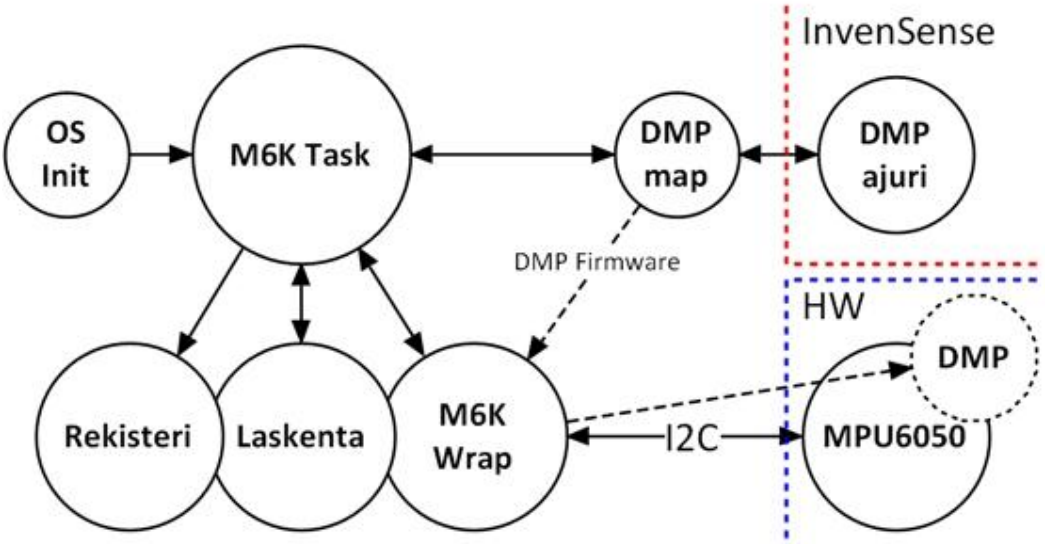
Kommunikointi MPU6050-asentotietoanturin kautta tapahtuu Aardvark SPI/ I^2C-adapterin kautta, joka muuntaa USB-liikenteen I^2C-väylälle. Käyttämällä erillistä virtalähdettä Universal Evaluation Board- kehitysalustan kanssa, voidaan Aardvark kytkeä suoraan sen pinneihin. Näin PC:ltä voidaan keskustella I^2C-väylää käyttäen MPU6050:lle.

Ohjelmiston kehitysympäristöksi valittiin virtualisoitu FreeRTOS käyttöjärjestelmä ja Microsoft Visual Studio IDE. Tämä valinta tarjosi kehittyneemmät debug- ja diagnostiikkatoiminnot sekä lokitiedostojen

kirjoittamisen suoraan tiedostojärjestelmään. T-Plat.e ohjelmistoalusta tarjoaa ohjelmistokehitykseen tuen Aardvark-adapterille joten sen kanssa kommunikointiin ei tarvinnut erikseen kirjottaa omaa ohjelmistoa.

5.4 Ajurin rakenne

Ohjelmiston rakenne muodostuu omasta ohjelmistomodulistaan sekä InvenSensen DMP-ajurista. Ajurimoduli pyörii omassa taskissaan ja keskustelee I^2C-väylän yli oman m6k_wrapper rajapintansa kautta. Projektissa ei implementoida erillistä I^2C-toteutusta vaan sen sijaan käytetään jo valmiita rajapintoja, jotka voidaan vaihtaa m6k_wrapper rajapintaan ilman tarvetta muokata koko ohjelmistoa. Tässä projektissa I^2C-toteutuksena toimi T-Plat.e-kirjastosta löytynyt iicmstr-lohko.

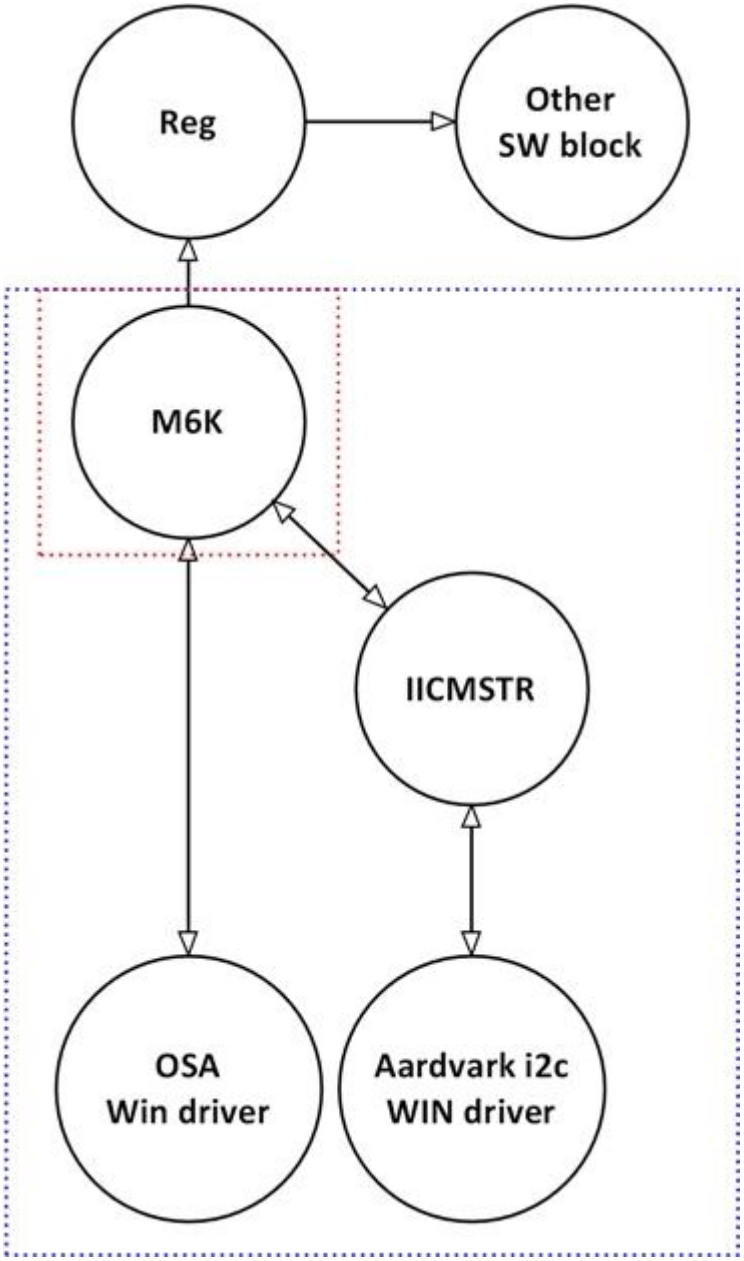


Kuvio 14. Lohkokaavio m6k ajurista.

DMP-ajuri on sulautettu osaksi m6k-lohkoa korvaamalla alustakohtaiset funktiokutsut m6k-lohkossa implementoiduilla samat toiminnot toteuttavilla funktioilla DMPmap-header-tiedostossa. Näin ajuri on päivitettävissä lisäämällä uuden version lähdekoodi osaksi projektia ja kääntämällä projektin uudelleen, jos ajurin ei vaadi uusia rajapintafunktioita.

Laskentaosa lohkossa muuntaa DMP:ltä luetut kvaterniot laiteyksiköistä liukuluvuiksi. Tämän jälkeen ne muunnetaan Euler-kulmiksi, jotka talletetaan

rekisteriin. Rekisteristä ne ovat muiden ohjelmistolohkojen käytettävissä ja näin niitä voidaan hyödyntää esimerkiksi MMI:ssä.

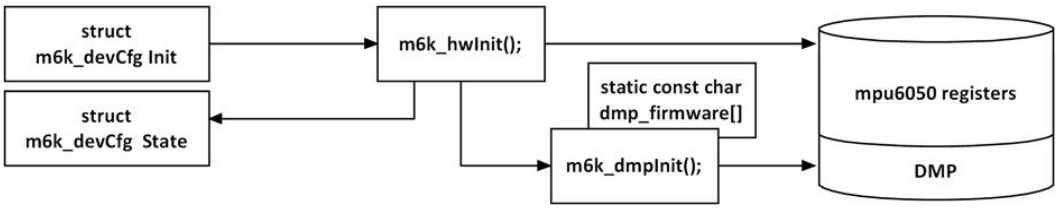


Kuvio 15 m6k-lohko osana sulautettua ohjelmistoa.

Kuviossa 15 on havainnollistettu m6k-lohkon sijoittumista osaksi sulautettua ohjelmistoa. Sinisellä on merkitty osa havainnolistaa käytettyjä ohjelmistokomponentteja ja punainen osa kirjoitettua ohjelmistoa.

5.5 Initialisointi

Käyttöjärjestelmäinitialisoinnin jälkeen m6k-lohko aloittaa initialisoimaan MPU6050-anturia. Konfiguraatio luetaan Init-structista, josta arvot ajetaan MPU6050:llan rekistereihin ja talletetaan State-structiin. Tämä siksi, että ohjelma on koko ajan tietoinen nykyisestä konfiguraatiosta, jonka anturi on vastaanottanut. Tilanteessa, jossa konfiguraatiossa esiintyy virhe, voidaan laite initialisoida uudelleen. Muiden laitteiden konfiguraation jälkeen konfiguroidaan DMP-prosessori. DMP:lle on ladattava InvenSensen tarjoama firmware ennen muita toimenpiteitä. Sen jälkeen kutsutaan DMP-prosessorin ajurin funktiota jotka hoitavat initialisonnin Init-structissa annetun konfiguraation mukaisesti. Konfiguraatiossa annetaan listaus eri ominaisuuksista, joita halutaan käyttää ohjelman suorituksen aikana.



Kuvio 16. Ajurin initialisointiprosessi.

6 AJURIN IMPLEMENTOINTI

Ajurin toteutuksen aikana päädyttiin tekemään kaksi muutosta alkuperäiseen suunnitelmaan. Ensimmäinen muutos oli siirtyminen stand-alone-toteutuksesta Tietolaitteen T-Plat.e:n käyttöön. Seuraava huomattava muutos oli InvenSensen DMP-prosessorin ajurin käyttäminen osana ohjelmistoa. Näistä muutoksista huolimatta, ohjelmistoarkkitehtuuri on toteutettu alkuperäisen suunnitelman mukaisesti.

Ajurin ohjelmistolohko kirjastoon nimettiin m6k:ksi. Jokainen ohjelmistolohkoon kirjoitettu funktio, määritys tai datatyypin määritys kirjoitettiin m6k etuliitteellä. Tämä tehtiin kirjaston ohjelmistolohkojen kanssa päällekkäisyyksien välttämiseksi ja on osa T-Plat.e kirjaston nimeämiskäytäntöä.

6.1 Stand-alone-ohjelmisto

Koska tarkoituksena oli kirjoittaa ajuri MPU6050-anturille ilman omaa I^2C-implementaatiota, päätettiin luoda erillinen Wrapper kerros I^2C-väylän kommunikaatiolle. Näin käytettävä I^2C-kirjasto olisi vaihdettavissa ilman, että ohjelman keskeisiin osiin tarvitsisi tehdä muutoksia. Tässä vaiheessa tutkittiin mahdollisuutta käyttää NXP:n LPCOpen ohjelmistokirjaston I^2C-toteutusta ohjelmistokehitysvaiheessa. Kirjaston toimintaa tutkiessa nousi ongelmaksi I^2C-kirjaston synkroninen toteutustapa, joka monimutkaistaisi rajapintoja turhaan sekä estäisi ajurin käytön osana reaaliaikakäyttöjärjestelmää.

Erilaisia vaihtoehtoja kartoittaessa huomattiin nopeasti, että vaikka ajurin kirjoittaminen tällä tavalla olisi mahdollista, pitäisi sitä muokata raskaasti ennen sen istuttamista osaksi T-Plat.e:n koodikirjastoa. Ajan säästämisen nimissä ohjelmisto päätettiinkin kirjoittaa suoraan osaksi T-Plat.e:n kirjastoa. Tämä ratkaisu mahdollisti kirjaston I^2C-ohjelmistolohkon, iiicmstr:in käytämisen osana projektia. Koska T-Plat.e on siirrettävä rajapinta, on sille kirjoitettu ohjelmisto myös lähtökohtaisesti siirrettävää.

MPU6050-asentotietoanturin kanssa samaan tuoteperheeseen kuuluu myös lähes identtinen anturi MPU6000, josta löytyy tuki myös SPI-väylälle. Ohjelmiston myöhemmän laajennettavuuden nimissä päätettiin Wrapper-taso jättää osaksi ohjelmiston toteutusta. Näin alla oleva kommunikaatioväylä voidaan vaihtaa tarvittaessa SPI-väylään helpommin.

6.2 Osana T-Plat.e-ohjelmistoalustaa

T-Plat.e ohjelmistoalustaa käytättämällä saatiin mukaan projektin kannalta oleellisia osia, esimerkiksi millisekunttikellon sekä I^2C-totetutuksen. Koska T-Plat.e pyörii oletusarvoisesti osana käyttöjärjestelmää, tuli nyt miettiä myös ohjelmiston toimintaa käyttöjärjestelmän kannalta. Ajurin tulisi pystyä toimimaan moniajoympäristössä.

Ajuria alettiinkin kirjoittamaan CoTask-prossesina, jossa se suorittaisi alustuksen, kommunikaation sekä laskennan. Tutkittaessa valmistajan tarjoamaa ajuria DMP-prosessorille huomattiin, että sen kommunikaatio on synkronista. Tästä johtuen ajurille päätettiin allokoida oma Taskinsa. Näin voitiin DMP:n kommunikaatiosta tehdä asynkronista Wrapper-tasossa.

Ohjelmiston kirjoittamista vaikeutti huomattavasti DMP-prosessorin dokumentoinnin puute. InvenSense tarjoaa dokumentaation vain DMP:n rajapinta funktioille eikä esimerkiksi sen rekistereitä ole dokumentoitu mitenkään. Näin syntyi myös tarve pystyä pitämään kirjaa siitä mitä DMP-prosessorille on alustettu jotta sitä voitaisiin ohjata luotettavasti.

6.3 DMP-ajurin integrointi

Käytännön toteutus aloitettiin määrittelemällä selkokieliset määritykset ajurin rekistereille. DMP:n käyttämien rekisterien selvittämisen apuna käytettiin anturille löytyviä avoimen lähdekoodin toteutuksia. Selkeyden vuoksi rekisterien nimet määriteltiin piirin valmistajan tarjoamien rekisterinimien mukaisesti. Päällekkäisyyksien välttämiseksi ohjelmiston symboleiden etuliitteenä käytettiin projektin tunnusta m6k.

InvenSensen ajurille tuli kirjoittaa rajapintafunktiot muuhun järjestelmään. Ajuria varten toteutettiin omat funktiot I^2C-kommunikointiin, FIFO-puskurin lukemiseen, kiihtyvyysanturin konfiguraation lukemiseen, aikaleiman hakemiseen sekä firmwaren lataukseen.

```
#define mpu_get_accel_sens      m6k_mpuGetAccelSens
#define mpu_write_mem           m6k_dmpWrite
#define mpu_read_mem            m6k_dmpRead
#define mpu_get_accel_fsr       m6k_mpuGetAccelFsr
#define mpu_read_fifo_stream    m6k_readFifoStream
#define mpu_load_firmware       m6k_dmpLoadFirmware
#define mpu_reset_fifo          m6k_resetFifo
#define get_ms                  m6k_getTimeStamp
```

Kuvio 17. Funktioiden kartoitus ajurissa toteutettuihin funktioihin.

Funktiot noudattelevat vähäisin muutoksin valmistajan esimerkkiajurista löytyviä ratkaisuja. Näin varmistetaan, että ne ovat yhteensopivia valmistajan tarjoaman ajurin kanssa.

6.4 Firmwaren lataus DMP-prosessorille

DMP-prosessoriin on aina käynnistyksen yhteydessä kirjoitettava firmware-ohjelmisto (InvenSensen ajurin mukana tarjotaan firmware DMP-prosessorille 3062-tavun kokoisena kiinteänä taulukkona.) Sen lataukseen kirjoitettiin m6k_dmpLoadFirmware-funktio valmistajan esimerkin mukaisesti. Ydin-toimintona funktiossa toimii do-while-silmukka, jossa taulukko ladataan MPU6050-piirille 16-tavun paketeissa, kirjoitettu paketti luetaan piiriltä ja varmistetaan, että paketit täsmäävät.

Master	S	AD+W		RA		DATA		DATA		P
Slave			ACK		ACK		ACK		ACK	

Kuvio 18 Purskemuotoinen kirjoitusoperaatio I^2C-väylällä.

Testiajossa huomattiin, että MPU6050-anturi kaatui aina 1516 tavun kirjoittamisen jälkeen. Laite toimi valmistajan tarjoaman testiohjelmiston kanssa, mutta koska valmistajan testiohjelmisto ei käytä DMP-prosessoria, ei voitu olla

varmoja onko kyseessä laitevika. I^2C-väylän liikennettä tarkasteltiin oskilloskooppilla yksittäisen 16-paketin tasolla ja sekä osoite että data olivat valideja. Kokonaisen kirjoitusoperaation tarkastelu oskilloskoopilla ei ollut mahdollista, johtuen viestien suuresta määrästä ja oskilloskoopin rajoittuneesta resoluutiosta.

```
do{

    this_write = min(M6K_LOAD_CHUNK, length - ii);
    /*  Write buffer     */
    m6k_dmpWrite(ii, this_write, (unsigned char*)&firmware[ii]);

    /*  Clear readbuffer to avoid invalid values */
    for(jj=0 ; jj < this_write ; jj++)  {
        cur[jj] = 0xA5;
    }

    /*  Read recently written values */
    m6k_dmpRead(ii, this_write, cur);

    /*  Compare them to original     */
    if(memcmp((unsigned char*)&firmware[ii], cur, this_write))  {
            return -2;
    }

    ii += this_write;
    DEB_MSG_ARGS(("Firmware loaded %d bytes" , ii));

}while(ii < length);
```

Kuvio 19. Firmwaren kirjoitus m6k_dmpLoadFirmware funktiossa.

Ohjelmiston toimintaa debuggerissa tarkasteltuna ei näkynyt aluksi mitään epäilyttävää ennen laitteen kaatumista. Lisäämällä puskurin käsittelyyn aikaleimat huomattiin, että kirjoitus puskuriin vei satunnaisesti useita satoja millisekuntteja. Tästä pääteltiin, että vika voisi mahdollisesti olla Aardvarkissa tai sen integraatiossa T-Plat.e:hen. Pienentämällä lähetettävän puskurin kokoa, tämä viive laski. Tutkittaessa T-Plat.e:n Aardvark-integraatiotiota, löytyi virhe, jonka vuoksi Aardvark.dll:stä kutsuttiin lukuoperaation jälkeen myös kirjoitusfunktiota, joka lähetti juuri luetun puskurin sisällön piirille. Tästä johtuen jokainen 16-tavun kirjoitusoperaatio tehtiin kahteen kertaan aiheuttaen muistiylivuodon 95 kirjoituskerralla.

Koska Aardvarkin integraatiota T-Plat.e:ssa oli käytetty aiemmissa tuotekehitysprojekteissa, ei vikaa osattu etsiä sieltä. Kyseinen virhe olikin

mahdollinen vain purskemuotoisessa lähetyksessä. Aiemmin mainittu viive kirjoitusoperaatioissa johtui kaksinkertaisen kirjoitusoperaation aiheuttamasta ruuhkasta väylällä, joka sai slave-tilassa toimivan MPU6050-anturin pitämään kellolinjaa alhaalla kunnes se pystyi taas käsittelemään uusia viestejä.

6.5 Lopullinen toteutus

Kommunikaation korjaamisen jälkeen ohjelmistoon kirjoitettiin korkeamman tason funktiot. Näiden funktioiden pääasiallinen tarkoitus on yksinkertaistaa ajurin implementointia osaksi tulevia projekteja. Niiden keskeisenä ideana on automatisoida initialisointiprosessia siten, että ajurin käyttöönotto olisi mahdollisimman yksinkertaista, eikä vaatisi syvällistä tutkimusta MPU6050-anturin toiminnasta. Niiden pääasiallinen tarkoitus on konfiguroida tarvittavat lisälaitteet alustettujen arvojen kanssa yhteensopiviksi. Esimerkiksi m6k_setSampleRate-funktio konfiguroi digitaalisen matalapäästösuodattimen alustetun näytteenottotiheyden mukaan.

Ohjelmistoon kirjoitettiin myös tuki ominaisuuksille, joita ei todennäköisesti tulla käyttämään osana sulautetun järjestelmän ohjelmistoa. Näitä ovat esimerkiksi raaka-arvojen luku suoraan gyroskoopeilta tai kiihtyvyysantureilta. Näitä toimintoja voidaan kuitenkin käyttää apuna, esimerkiksi ohjelmistokehityksen aikana, testauksessa tai selvitettäessä anturin vikatiloja.

```
Boolean m6k_chkMagnitude(
    double * q
)   {
    double mag;

    mag =   sqrt(q[0] * q[0] + q[1] * q[1] + q[2] * q[2] + q[3] * q[3]);

    if(mag > 1.000001 || mag < 0.999999)    {
        return FALSE;
    }   else    {
        return TRUE;
    }
}
```

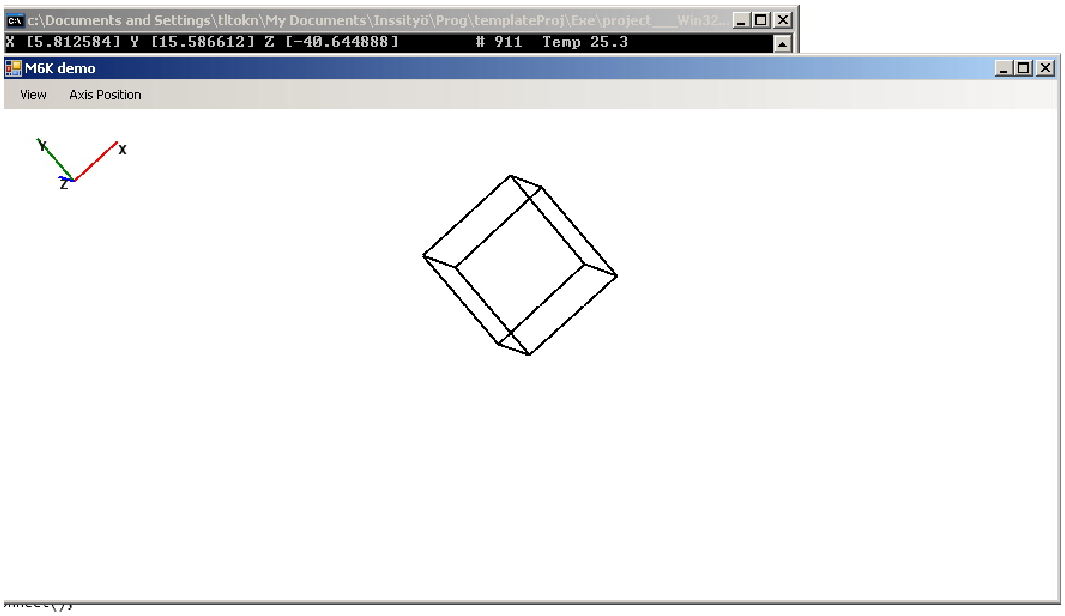
Kuvio 20 Kvaternion pituuden tarkistuksen C-implementaatio.

Ohjelmistoon lisättiin myös kvaternioiden pituuden tarkastelu. Tällä tavalla voidaan varmistaa, että mittausdata on validia. Testauksen aikana huomattiin, että

kvaternioiden pituuden tarkastelun tarkkuutta oli lisättävä, sillä DMP-ajuri päästi läpi joillain konfiguraatioilla kvaternioita, joiden pituus ylitti vaaditun arvon miljoonasosien tarkkuudella, aiheuttaen virheellisiä tuloksia.

6.6 Demo-ohjelmisto

Anturin toiminnan havainnolistamiseksi kirjoitettiin myös demo-ohjelmisto. Ohjelmisto kirjoitettiin c# kielellä Microsoft .NET ympäristöön. Koska m6k-ajurin kehitysympäristönä käytetään Windowsia, voivat ohjelmistot kommunikoida keskenään tiedostojärjestelmän kautta. Ohjelmisto piirtää näytölle kuution ja akseliston, jotka liikkuvat anturin liikkeiden mukaan.



Kuvio 21 Demo-ohjelmisto sekä debug-loki

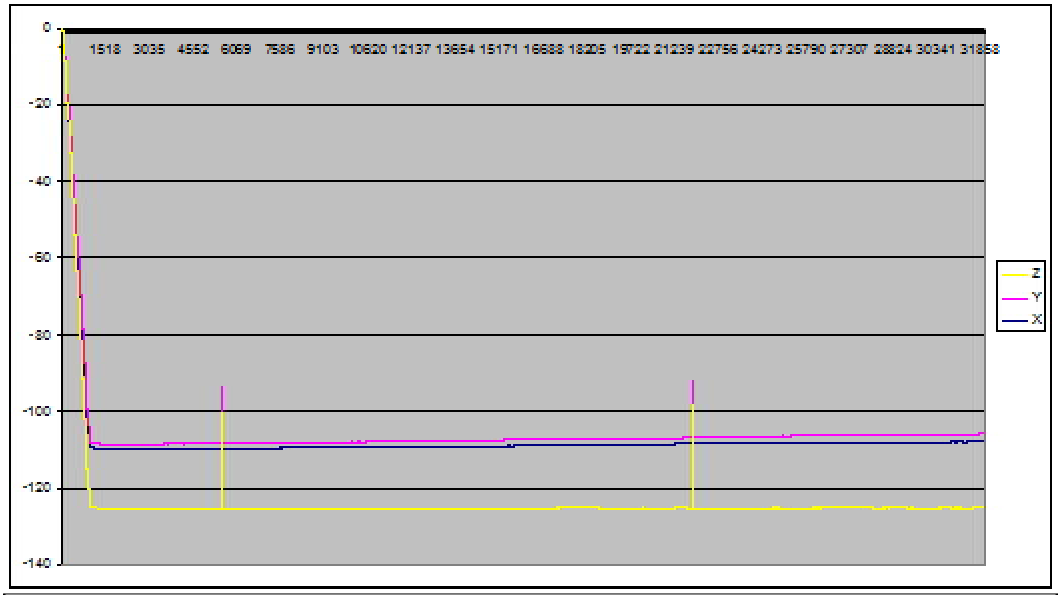
7 TESTAUS

Testi vaiheessa haluttiin vertailla eri konfiguraatioiden suorityskykyä. Koska anturia on tarkoitus käyttää aina yhdessä DMP-prosessorin kanssa suoritettiin mittaukset vertaillen 3-akselista ja 6-akselista mittausta. Käytettäessä 6-akselista mittausta on käytössä aina anturifuusiolaskenta.

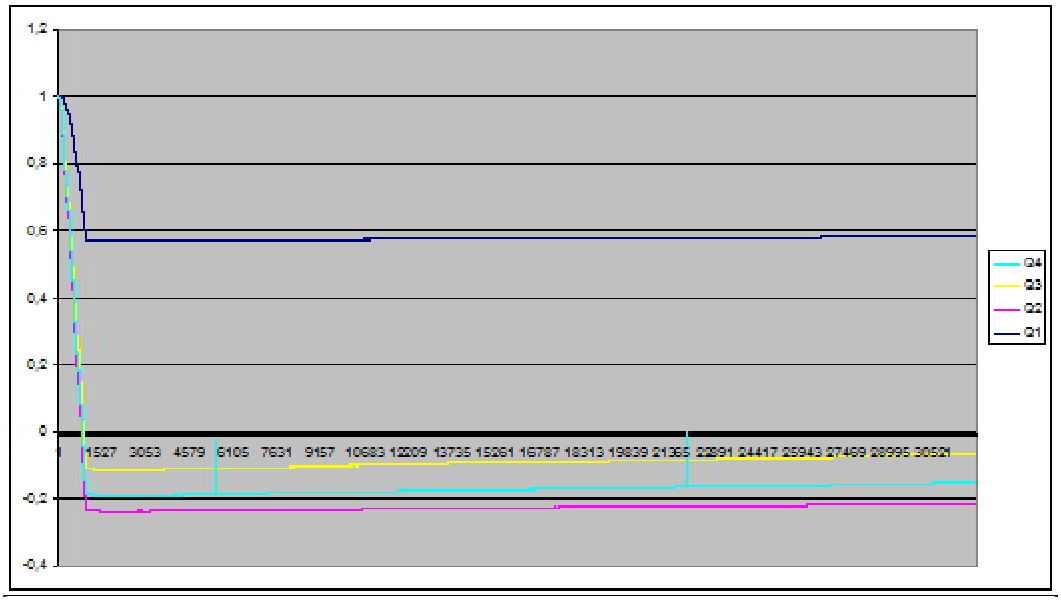
Olennainen mittaus oli myös lämpötilan vaikutus mittaustuloksiin sillä sen tiedettiin olevan MEMS-gyroskooppien heikkous.

7.1 Mittaus asetukset ja ympäristö

Alkuperäinen suunnitelma oli suorittaa mittaukset pareittain pienimmästä suurimpaan, esimerkiksi 250±°/s ja ±2g. Suoritettaessa mittauksia huomattiin kuitenkin, että mittausarvoissa esiintyi virheellisiä arvoja joillain konfiguraatioilla.



Kuvio 22 Mittauksessa esiintyvät virheeliset arvot asteina.



Kuvio 23 Virhe kvaternioina.

Asiaa tutkittaessa ei löydetty mitään selkeää syytä mikä tällaisia virheitä aiheuttaa. Konsultoitaessa valmistajaa asiasta selvisi, että DMP-prosessorin ohjelmisto toimii varmimmin tilassa, jossa kiihtyvyysanturi on konfiguroitu 2G:tä per sekuntti tilaan ja gyroskooppi 2000 asteeseen per sekuntti tilaan.

Projektin tavoitteena oli saada mahdollisimman tarkka asentotieto, joten mittaukset keskittyvät asentotiedon mittaukseen. Kiihtyvyysanturi on konfiguroitu

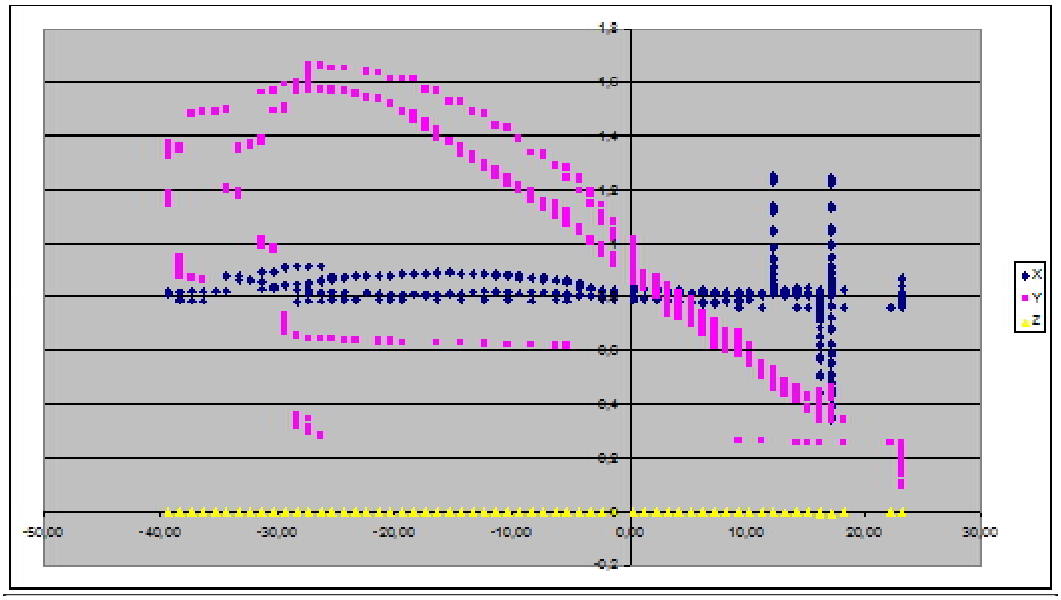
jokaisessa mittauksessa valmistajan suosittelemaan tilaan, jotta epävarmuustekijöitä saataisiin eliminoitua tuloksista. Graafeissa käytettiin mittausten ensimmäiset 32000 näytettä, jotta tulokset pysyisivät vertailukelposina.

Mittauksissa käytettiin DMP-prosessoria sillä asentotiedon laskenta luo turhaa kuormaa prosessorille. Mittaukset suoritettiin käyttämällä 8MHz kelloa ja 200Hz näytteenottotaajuutta.

Mittaukset suoritettiin kiinnittämällä MPU6050-anturi jalustaan, jossa se asetettiin silmämääräisesti suoraan jalustaan kiinnitettyjen vesivaakojen avulla.

7.2 Lämpötilan vaikutus mittaustuloksiin

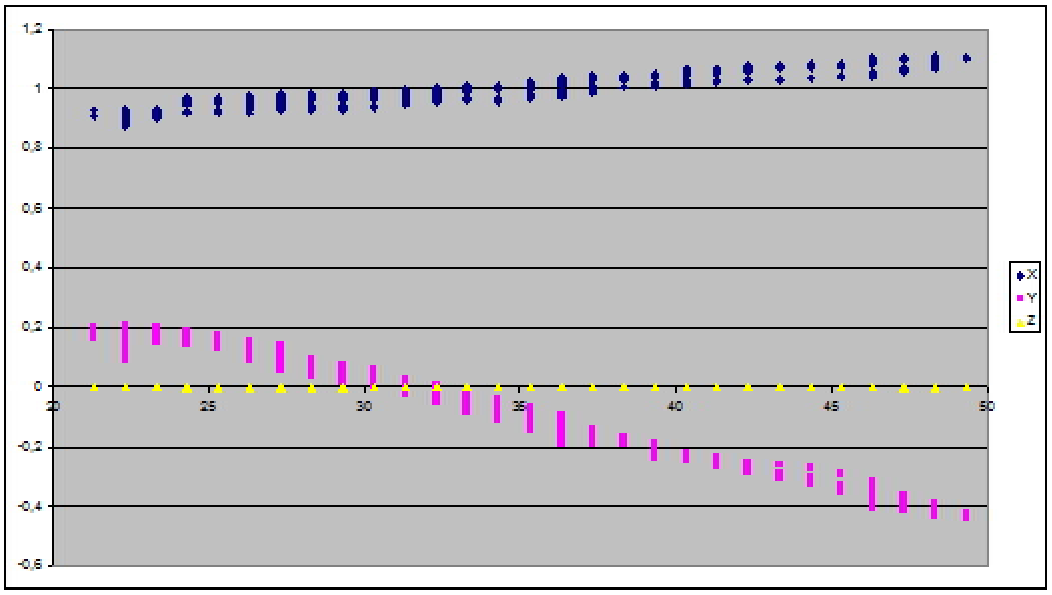
MPU6050-asentotietoanturille luvattu lämpötila-alue on -40° - +85° celsiusta. Mittauksen aikana huomattiinkin, että anturi kaatui aina lämpötilan palatessa -40° asteesta noin 15° asteeseen.



Kuvio 24. Lämpötilan laskun vaikutusta asentotietoon.

Kuviossa 21 vaaka-akselilla näkyvät lämpötilan muutokset ja pystyakselilla MPU6050-anturin akseleiden asennon muutokset. Graafilla näkyvät suuret heitot x-akselin suuntaisesti ovat suurella todennäköisyydellä virheellisiä arvoja ennen anturin kaatumista. Graafista nähdään myös selkeästi noin 1,2 asteen muutos y-

akselin kulmassa, kun lämpötila muuttuu -25°:sta +15° asteeseen.

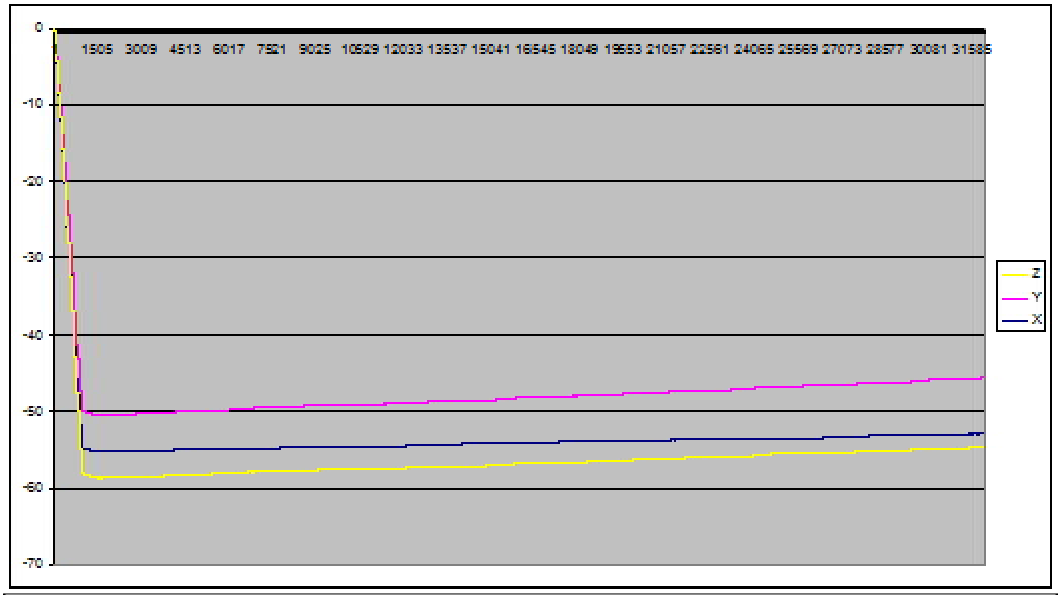


Kuvio 25. Lämpötilan nousun vaikutus akseleiden mittausarvoihin.

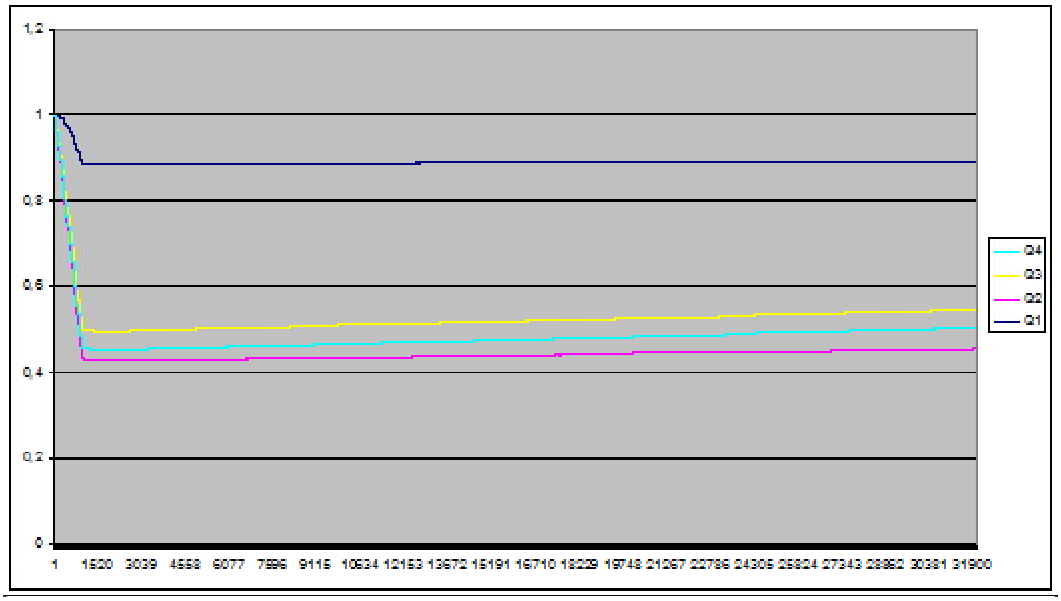
Myös lämpötilan nousu vaikutti selkeästi sekä x- että y-akseleiden mittaustuloksiin. Kummallakin akselilla asentotiedon muutos on noin ±0,4 astetta 30° asteen lämpötilan kasvun aikana.

7.3 Mittaustulokset ilman anturifuusiota

Teorian mukaan MEMS-gyroskooppi kärsii gyroskoopin liukumasta jo lyhyelläkin aikavälillä. Tätä ilmiöitä olikin havaittavissa mittaustuloksista selkeästi.

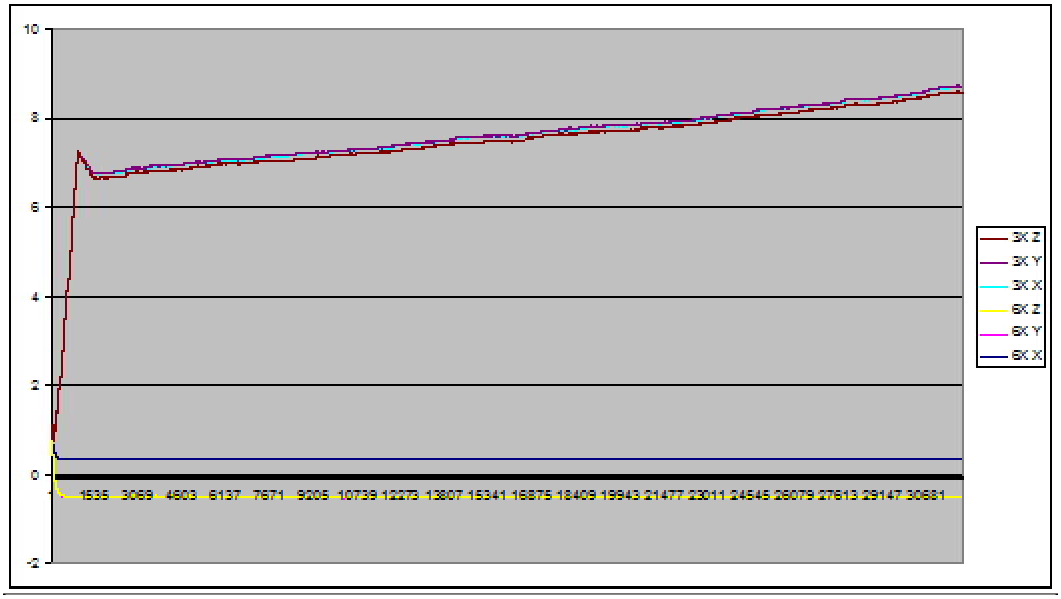


Kuvio 26. Gyroskoopin liukuma asteina käytettäessä 3-akselista laskentaa.



Kuvio 27. Gyroskoopin liukuma kvaternioina käytettäessä 3-akselista laskentaa

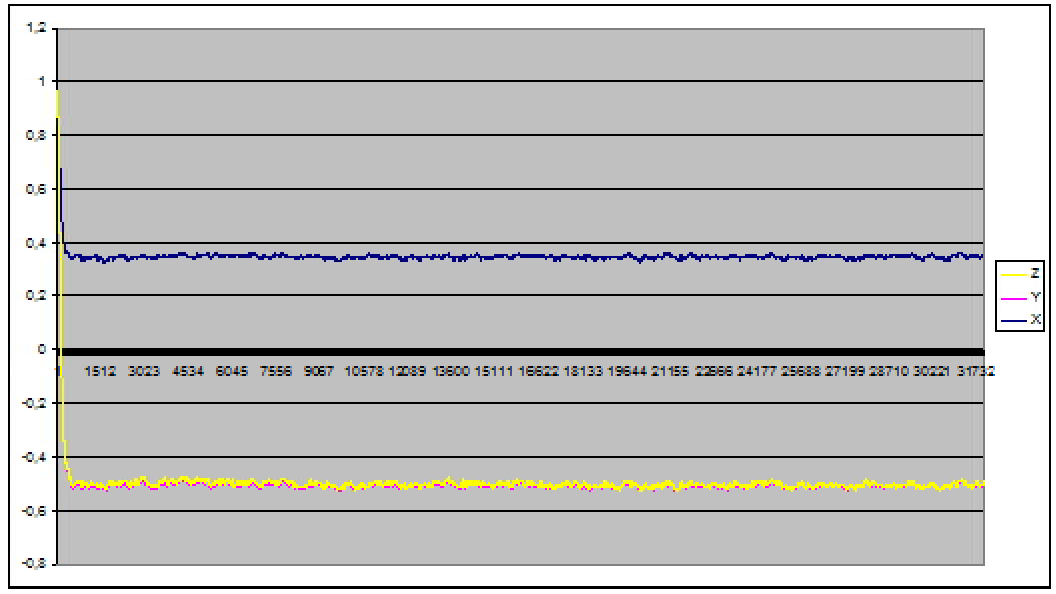
Liukumasta johtuen, anturin käyttäminen ilman anturifuusiolaskentaa on käytännössä mahdotonta ilman aktiivista korjausta. Jo lyhyelläkin aikavälillä liukuman aiheuttama virhe on useita asteita. Vaikka ongelmaa voitaisiin kompensoida ohjelmallisesti, ei sitä projektissa toteutettu prosessorin resurssien säästämiseksi.



Kuvio 28. Vertailu käyttäen 3-akselisen ja 6-akselisen laskennan välillä.

Jos vertaillaan eroa 3-akselisen laskennan ja 6-akselisen fuusiolaskenan välillä huomataan, että erot ovat merkittäviä. Noin 7° asteen erotus lähtöarvoissa 3-akselisen mittauksen käyrällä johtuu todennäköisesti DMP-prosessorin suorittamasta korjauslaskennasta ilman kiihtyvyysanturia. Samanlainen tasapainotus on nähtävillä myös 6-akselisen mittauksen käyrällä mutta huomattavasti lievempänä. Mittauksen 32000 näytteen aikana 3-akselisen mittauksen gyroskooppinen liukuma on jo noin 2° astetta. Anturifuusiolaskennalla liukumaa ei ole havaittavissa.

7.4 Mittaustulokset anturifuusiolla



Kuvio 29. Anturifuusiolaskennalla suoritettu mittaus.

Käytettäessä anturifuusiomenetelmää laskemaan asentotietoja saadaan MEMS-gyroskoopille tyypillinen liukuma poistettua mittaustuloksista. Vaikka graafilla esiintyykin selkeästi näkyvää värinää mittauksen aikana eivät mittaustulokset muutu merkittävästi 32 000 näytteen aikana. Tämä värinä vastaa MPU6050-anturin gyroskoopille ilmoitettua värinää mittauksen aikana.

Anturifuusiolla suoritettujen mittausten tulokset vahvistavat sen, että ajuri tarjoaa luotettavaa asentotietoa myös pitkällä aikavälillä.

8 YHTEENVETO

Työn tuloksena kehitettiin ajuri, jolla pystytään tarkastelemaan kappaleen asentoa luotettavasti ja tarkasti. Työn vaatimusmäärittelyn mukaisesti käytössä on DMP-prosessorille InvenSensen tarjoama oma ohjelmistonsa, joka on sulautettu osaksi ajuria.

Suurimmat haasteet projektin aikana loi dokumentaation puute sekä I^2C-väylän kommunikaatiosta löytynyt virhe, joka pitkitti kehitysprosessia. Voidaan kuitenkin olettaa, että implementoimalla erillinen I^2C-ajuri osaksi projektia, olisi aikaa mennyt varmasti yhtä paljon kuin tämän virheen paikantaminen vaati. Dokumentaation puute loi haasteita DMP-prosessorin käyttämiseen osana projektia. Erityisesti vikatilojen selvittäminen oli haasteellista, sillä DMP-prosessorin tulosten tai konfiguraation oikeellisuudesta ei ollut aina varmuutta.

MPU6050-ajurin kehitystä tullaan vielä jatkamaan tämän projektin jälkeen ja ajurin käyttäminen tuotannossa vaatisikin vielä lisää testejä. Todennäköisesti kehitetty ajuri liitetään osaksi Tietolaitteen T-Plat.e ohjelmistokehitysalustan kirjastoa kesän 2016 aikana.

LÄHTEET

/1/ Kaajakari, V. 2009. Practical MEMS. Las Vegas. Small Gear Publishing.

/2/ Kaajakari, V. 2009. Practical MEMS. Las Vegas. Small Gear Publishing.

/3/ Kaajakari, V. 2009. Practical MEMS. Las Vegas. Small Gear Publishing.

/4/ Andrejašic, M.2008. MEMS Accelerometers. University of Ljubljana. Faculty for mathematics and physics, Department of physics

/5/ Andrejašic, M.2008. MEMS Accelerometers. University of Ljubljana. Faculty for mathematics and physics, Department of physics

/5/ Kaajakari, V. 2009. Practical MEMS. Las Vegas. Small Gear Publishing.

/6/ Waltari, M . 2002 ,Circuit techniques for low-voltage and high-speed A/D converters Helsinki University of Technology, Electronic Circuit Design Laboratory

/7/ InveSense Inc. 2013 ,MPU-6000 and MPU-6050 Product Specification Revision 3.4, Sunnyvale. InveSense Inc

/8/ NXP Semiconductors. 2014 , I^2C -bus specification and user manual. NXP Semiconductors.

/9/ Jarrell, J . 2008 , Employ Sensor Fusion Techniques for Determining Aircraft Attitude and Position Information. Morgantown. ProQuest

/9/ Siciliano, B. 2008 , Springer Handbook of Robotics. Berlin. Springer-Science & Business Media.

/10/ Byung-Uk Lee. 1991, Stereo Matching of Skull Landmarks, Ph.D. Thesis, Stanford , Univ., Stanford, CA.

.