eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea

INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Informatika Ingeniaritzako Gradua

Konputagailuen Ingeniaritza

Gradu Amaierako Lana

MPU-9250 azelerometro eta giroskopioaren erabilera SPI eta I2C busen bidez

Egilea

Jon Ayuso Hernandez

Donostia, 2021eko irailaren 5a

eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Informatika Ingeniaritzako Gradua

Konputagailuen Ingeniaritza

Gradu Amaierako Lana

MPU-9250 azelerometro eta giroskopioaren erabilera SPI eta I2C busen bidez

Egilea

Jon Ayuso Hernandez

Zuzendariak

Jose Ignacio Martin Aramburu

Izaskun Etxeberria Uztarroz

Laburpena

Mundua fenomeno fisiko desberdinek inguratzen dute eta sentsoreei esker fenomeno fisiko horiek sortzen dituzten seinaleak seinale elektriko bihur daitezke. Proiektu honetan, azelerazioa eta biraketa fenomeno fisikoak detektatzen dituen MPU-9250 gailua landu da. Gainera, gailutik lortzen diren datuak tratatzeko mikroprozesadore baten beharra dago eta ondorioz gailuaren eta mikroprozesadorearen artean komunikazioa egon behar da. Horretarako bi bus desberdin erabili dira, I^2C busa eta SPI busa.

Bus mota bakoitzerako kode zati bat planteatu da komunikazioa arazorik gabe gauzatzen dela egiaztatzeko eta gailutik eskuratzen diren datuak ikuskatzeko.

Eskuratzen diren datu horiei esker, objektu batek jasaten dituen azelerazio eta biraketak (lehen aipatu diren bi fenomeno fisikoak) kalkulatzen dira, objektuak duen orientazioa estimatuz.

Hori da proiektu honen muina, MPU-9250 gailua eta mikroprozesadore bat erabiliz, robot baten muturreko pintzaren orientazioa beti konstante mantentzea posizio horizontalean.

Lehenik eta behin, lana aurrera eramateko beharrezkoak diren kontzeptu teorikoak ikertuko dira (gailuaren xehetasunak, bi bus komunikazioen funtzionamendua...). Ondoren, planteatzen den arazoarentzat ebazpen bat proposatuko da. Azkenik, ebazpen horren kodea idatziko da eta funtzionamendua egokia dela egiaztatuko da.

Gaien aurkibidea

Irudien aurkibidea

Taulen aurkibidea

1. KAPITULUA

Sarrera

Lehen atal honetan, lanaren sarrera bat aurkezten da, proiektu hau gauzatzeko egon den motibazioa, lana inguratzen duen Erabilera praktikoa eta proiektua gauzatzeko erabili den metodologia deskribatuz. Azkenik dokumentuan aurkituko diren kapitulu desberdinen laburpen txiki bat egingo da.

1.1 Motibazioa

Objektu batek duen abiadura, orientazioa eta jasaten dituen azelerazioei buruz informazioa ezagutu behar denean, IMU [Hazry et al., 2009] (*Inertial Measurement Unit*) bat erabiltzea da aukera egokiena. Tresna konplexu hauek azelerometro, giroskopio eta magnetometro batez osatuak daude eta gehienbat hegazkinak, barkuak, sateliteak... maniobratzeko erabiltzen dira. Baina hau ez da IMUen edo azelerometroaren eta giroskopioaren usadio bakarra [Ahmad et al., 2013]. Izan ere, gaur egun garrantzi teknologiko haundia duen errealitate birtuala gailu hauetaz baliatzen da. [Desai et al., 2014][You and Neumann, 2001]

Beste erabilera sinpleagoak ere baditu, telefono mugikorren orientazioa detektatzearena bezala adibidez (mugikorra posizio horizontalean edo bertikalean dagoenean pantailaren orientazioa ere aldatzeko). Proiektu honen helburua sistema txertatuen esparruan barneratuz MPU-9250 gailua, nabigazioa edo errealitate birtuala bezain konplexua ez den beste jarduera zehatz baterako erabilera bat lantzea eta probatzea.

1.2 Erabilera praktikoa

Biraketak, azelerazioak eta orientazioa detektatzeko gai den MPU-9250 gailua eta honekin komunikazioa gauzatzeko beharrezkoak diren komunikazio-busak egoki erabiltzea dira lan honen ardatza. Lehen aipatu den bezala, MPU-9250 gailua azelerometro, giroskopio eta magnetometro batek osatzen dute, baina lan honetarako azelerometroa eta giroskopioa nahikoak dira bete nahi den ataza gauzatzeko.

Proposatzen den erabilera praktikoa honakoa da: Sistema txertatuen diseinua irakasgaian erabiltzen den beso robotikoaren bukaeran dagoen pintza beti posizio horizontalean mantentzea, robotak egiten dituen mugimenduak edozein izanda ere. Beso robotiko hau 4 serbok osatzen dute eta Explorer 16 txartelaren bidez eta irakasgaian landu den kodearen bidez kontrolatzen dira. Baina lan honetan proposatu den erabilera praktikoa gauzatzeko, azkeneko serboaren mugimenduak MPU-9250 gailuak eskaintzen dituen datuen arabera kontrolatuko dira. Horretarako, lehenik eta behin MPU-9250 gailua ondo menderatu behar da, eskuratzen dituen datuak interpretatu behar dira eta erabili den Arduino Uno plakako mikroprozesadorearen eta MPU-9250 gailuaren arteko komunikazioa gauzatzeko beharrezkoak diren I^2C komunikazio-busa eta SPI komunikazio-busak kontrolatu behar dira.

1.3 Dokumentuko kapituluak

Dokumentu honetan hainbat kapitulu aurkitzen dira egin den proiektua deskribatzen dutenak. Hauek dira kapitulu horiek:

- Proiektuaren Helburuen Dokumentuaren antolaketa: Kapitulu honetan proiektuaren antolaketa deskribatzen da, ezarri diren helburuak adieraziz, proiektua ataza desberdinak aurkeztuz eta denbora estimazioak eta desbiderapenak ezagutaraziz.
- Ikerketa: Kapitulu honetan proiektua aurrera eramateko beharrezkoak diren jakintza teorikoak azaltzen dira.
- Inplementazioa: Lan honetan planteatu den erabilera praktikoari eta bi bus desberdinen bidezko komunikazioari eman zaien ebazpenak aurkezten dira.
- Ondorioak: Proiektuaren azkeneko kapitulu honetan, atera diren ondorioak azaltzen dira eta etorkizunerako planteatzen diren hobekuntzak aurkezten dira.

2. KAPITULUA

Proiektuaren Helburuen Dokumentua

2.1 Proiektuaren Helburuen Dokumentuaren antolaketa

Dokumentuaren atal honetan ezarri diren helburuak aztertuko dira, proiektua garatzeko egindako plangintza eta denbora estimazioak azalduz.

2.2 Helburuak

Proiektu honen hasieran ezarri ziren helburu nagusiak hurrengoak izan dira:

- **MPU-9250 moduluaren azterketa:** Landu behar den gailua menderatu behar da eta horretarako lehenik ondo ulertu behar da zer neurtzen duen, nola eskuratzen diren datuak, ze konexio dituen... Laburbilduz, lehen helburua gailu honek nola funtzionatzen duen aztertzea eta ulertzea da.
- **I^2C bus bidezko komunikazioa erabili:** Proiektu honetan, MPU-9250 gailuarekin komunikazioa ezartzeko, bi komunikazio-bus erabiltzea planteatu da, horietako bat I^2C busa. Hori dela eta, komunikazio protokolo honen alderdi teorikoa menderatzea eta inplementatzeko gai izatea da helburuetako bat
- **SPI bus bidezko komunikazioa erabili:** Aurreko helburuarekin jarraituz, beste komunikazio-busa SPI da. Helburu honekin protokolo hau menderatzea, bai alderdi teorikotik eta bai alderdi praktikotik bilatzen da.

- **MPU-9250 moduluarekin praktika bat garatu:** Behin gailua aztertuta eta behar den aurrekari teorikoa menderatuta, erabilera praktiko bat erakusten duen jarduera bat planteatu.
- **Arduino ingurunea menderatzea:** Arduino Uno plakaren hardwarea eta hau programatzeko erabiltzen den softwarea eta programazio lengoaia ikasi. Honekin batera, proposatuko diren jarduerak Arduino ingurunean inplementatzeko gaitasuna eskuratu.
- **Arduino liburutegiak:** Arduino Uno plaka programatzeko eta MPU-9250 gailua kontrolatzeko laguntza eskaintzen duten liburutegiak aztertu eta hauetaz baliatzeko kapaz izatea.

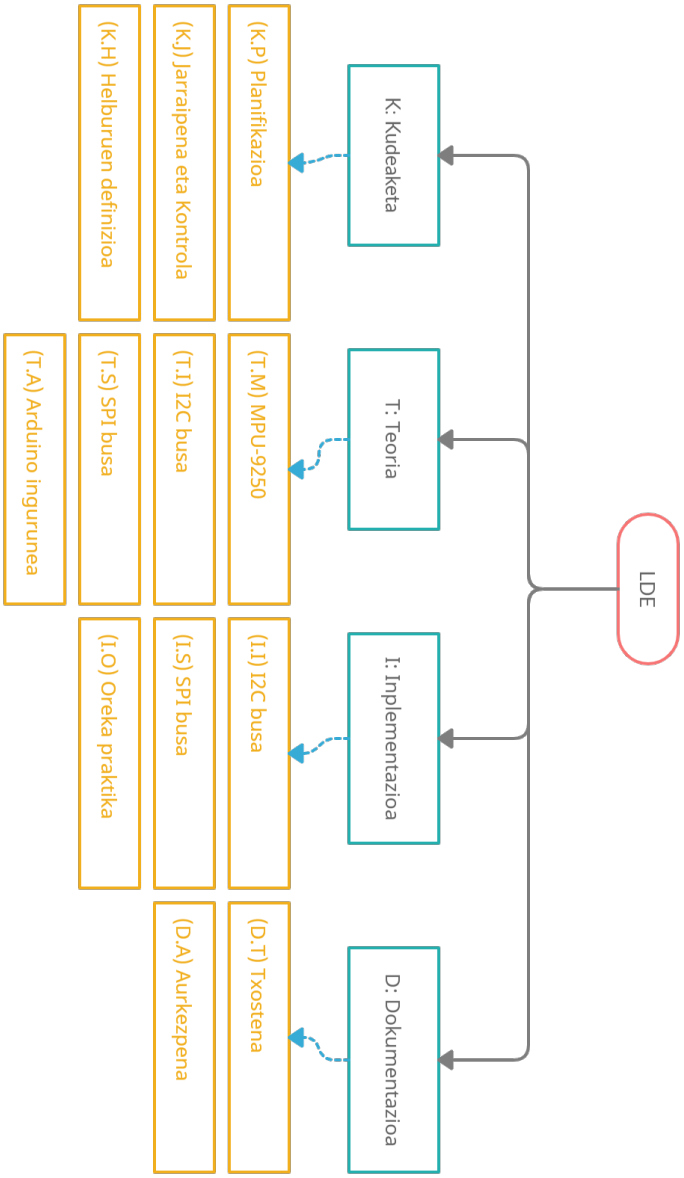
2.3 Plangintza

Azpiatal honetan, proiektuaren atazak deskonposatuko dira, LDE diagrama aurkezten da (2.1 irudia) eta denbora estimazio bat egingo da.

2.3.1 Atazen deskonposaketa

- **K: Kudeaketa**
  - **(K.P) Planifikazioa:** GrAL-ren plangintza garatuko da. Proiektua ataza desberdinetan banatuko da, bakoitzarentzat modu grafikoan adierazten duten diagramak diseinatuko dira eta denbora estimazioak eta desbiderazioak aztertuko dira.
  - **(K.J) Jarraipena eta Kontrola:** Proiektua aurrera joan ahala, zuzendariekin egindako bileren bidez erabaki desberdinak hartuko dira sortutako arazoei irtenbide bat ematen saiatzeko, planteamendu berriak eztabaidatzeko eta prozesua modu zuzen batean gauzatzen ari dela ziurtatzeko.
  - **(K.H) Helburuen definizioa:** Proiektuak izango dituen helburu desberdinak ezarriko dira, bakoitzari azalpen bat emanez.

- **T: Alderdi teorikoaren ikerketa**
  - **(T.M) MPU-9250:** gailuari buruz ikertuko da eskuliburuan azaltzen den informazioa aztertuz eta barneratuz. Honekin batera, Azelerometroa eta Giroskopioari buruzko alderdi teorikoak ikertuko dira hauen xehetasunak ondo ulertu eta menderatzeko.
  - **(T.I) I^2C busa:** I^2C komunikazio-busari buruzko teoria barneratuko da ondoren hau praktikara eramateko gai izateko.
  - **(T.S) SPI busa:** SPI komunikazio-busari buruzko teoria barneratuko da ondoren hau praktikara eramateko gai izateko.
  - **(T.A) Arduino ingurunea:** Erabiliko den inguruneari buruzko informazioa aztertu eta menderatzeko ahalmena eskuratuko da proiektua garatzeko beharrezkoak izango diren software eta hardwarea ezagutu eta ondo ulertzeko
- **I: Inplementazioa**
  - **(I.I) I^2C busa:** I^2C komunikazio-busa erabiliz, MPU-9250 gailuaren eta Arduino Uno plakaren artean komunikazioa ahalbidetzen duen praktika bat inplementatu Arduino softwarea erabiliz.
  - **(I.S) SPI busa:** SPI komunikazio-busa erabiliz, MPU-9250 gailuaren eta Arduino Uno plakaren artean komunikazioa ahalbidetzen duen praktika bat inplementatu Arduino softwarea erabiliz.
  - **(I.O) Oreka praktika:** MPU-9250 gailua erabiliz, proiektuko praktika nagusia (oreka mantentzea ahalbidetzen duena) Arduino Uno plakan inplementatu eta probatu.
- **D: Dokumentazioa**
  - **(D.T) Txostena:** GrAL-ean egindako proiektua zehatz-mehatz deskribatzen duen txostenaren dokumentazioa gauzatuko da. Behin proiektua amaitzean, txosten hau izango da dokumentazio bezala entregatuko dena.
  - **(D.A) Aurkezpena:** GrAL-ean egindako proiektuaren defentsa egingo da. Proiektua tribunalaren aurrean aurkezteko beharrezkoak izango diren gidoia, baliabide grafikoak etab. prestatuz.



2.1 Irudia: *LDE diagrama*

2.3.2 Denbora estimazioak eta desbiderapenak

Desbiderapenak aztertuz, guztira +16h-ko desbiderapen bat egon da. +5h alderdi Teorikoan eta +11h Inplementazioan.

- Teoria (+5h): Hasiera batean 101 ordu estimatu ziren teoria inguruko ikerketa egiteko, baina amaieran 105 ordu behar izan dira. Lehenik, MPU-9250 gailuaren inguruko ikerketan +6h-ko desbiderapena egon da, gehien bat gailua espero zena baina konplexuagoa zelako eta menderatu behar ziren xehetasun ugari zeudelako. I^2C busaren kasuan -2h-ko desbiderapena egon da aurkitu den erreferentziak ikerketa lana asko erraztu duelako. Azkenik Arduino-ren ikerketa prozesuak +1h-ko desbiderapena izan du gehien bat Arduino IDE softwarearen erabilera batzuk kontrolatzen uste zena baina zerbait gehiago behar izan delako.

- Inplementazioa (+11h): Hasiera batean 70 ordu estimatu ziren inplementazioak aurrera eramateko, baina 81 ordu behar izan dira, hau da, +11h-ko desbiderapena egon da. 2 azpiatazek sortu dute desbiderapen hau: SPI bus bidezko inplementazioan +6h-ko desbiderapena egon da hardware arazo baten ondorioz. Kodea ondo egon arren SPI-k ez zuen behar bezala funtzionatzen eta egun batzuk igaro ondoren, ikerketa eta kanpotik jasotako laguntzari esker aurkitu da akatsa, 3.1.5 atalean azaltzen dena.

  Beste aldetik, erabilera praktikoaren inplementazioak +5h-ko desbiderapena izan du 4.4.1 atalaren amaieran azaltzen den planifikazio aldaketa baten ondorioz.

Ataza	Azpiataza	Denbora estimazioa	Estimazioa guztira	Denbora erreala	Atazak guztira
K	K.P	20	44	20	44
	K.J	22		22	
	K.H	2		2	
T	T.M	51	101	57	106
	T.I	21		19	
	T.S	21		21	
	T.A	8		9	
I	I.I	21	70	21	81
	I.S	21		27	
	I.O	28		33	
D	D.T	75	85	75	85
	D.A	10		10	
Orduak	**Guztira**	300	300	316	316

2.1 Taula: Denbora desbiderapenak

3. KAPITULUA

Ikerketa

Dokumentuaren zati honetan, proiektua garatzeko eskuratu behar izan diren jakintza teorikoen azalpen bat emango da. 4 azpiatal desberdinetan banatuko da ikerketa prozesua: MPU-9250 gailua, I^2C komunikazio busa, SPI komunikazio busa eta Arduino ingurunea.

Azpiatal bakoitzean, gero inplementazio eta garapen prozesuan garrantzitsuak eta beharrezkoak izango diren kontzeptu teorikoak azaldu eta landuko dira.

3.1 MPU-9250

MPU-9250, TDK InvenSense® enpresak sortutako *Inertial Measurement Unit* gailu bat da (3.1 irudia). Gailu hau, enpresa berdinak sortutako lehenengo generazioko MPU-9150 gailuaren eboluzio bat da, kontsumo baxuagoa, tamaina txikiagoa (bere tamaina, lehenengo generazioko gailuarena baina %44 txikiagoa da) eta neurketak egiteko gailu hobetuak dituena.

Azelerometro, giroskopio eta magnetometro batez osatua dago. Bakoitzak 3 ardatzetan (*x, y, z*) neurketak egiteko kapaza dena [IPS, 2014], beraz MPU-9250 gailua 9 ardatzeko *Motion Tracking* gailua (Mugimenduaren segimendua egiteko gailua) dela esaten da.

Neurketak gauzatzeko, gailuak duen ardatz bakoitzean 16 biteko ADC (*Analog-to-Digital Converter*) bat dauka, beraz guztira 9 ADC ditu inkorporatuak [IPS, 2014].

3.1.1 Arkitektura

Gailu honek guztira, byte bateko 100 erregistro ditu, baina proiektu honetan ez dira denak beharrezkoak. Hauek dira proiektuan erabili diren erregistroak:

- 27 erregistroan, giroskopioaren konfigurazioa ezartzen da. 4. eta 3. bitetako balioak definituz adieraz daiteke ze neurketa eskala erabili nahi den. (Eskala mota desberdinak 3.1.3 azpiatalean adierazten dira).
- 28 erregistroan, azelerometroaren konfigurazioa ezartzen da. 4. eta 3. bitetako balioak definituz adieraz daiteke ze neurketa eskala erabili nahi den. (Eskala mota desberdinak 3.1.2 azpiatalean adierazten dira).
- 59,60,61,62,63 eta 64 erregistroetan, azelerometroaren 3 ardatzen balioak gordetzen dira. Ardatz bakoitzeko datuak bi bytetan gordetzen dira, horregatik erabili behar dira 6 erregistro datu guztiak gordetzeko. Proiektuan programatu diren kodeetan erregistro hauen helbide hamaseitarra erabili da (3B,3C,3D,3E,3F eta 40).
- 67,68,69,70,71 eta 72 erregistroetan, giroskopioaren 3 ardatzen balioak gordetzen dira. Ardatz bakoitzeko datuak bi bytetan gordetzen dira, horregatik erabili behar dira 6 erregistro datu guztiak gordetzeko. Proiektuan programatu diren kodeetan erregistro hauen helbide hamaseitarra erabili da (43,44,45,46,47 eta 48).

Aipatu den bezala, MPU-9250 gailuak azelerometro bat, giroskopio bat eta magnetometro bat ditu txertatuak (azken hau ez da proiektuan erabili behar izan baina garrantzi handia duenez teorian aztertuko da). Elementu guztiak ondo menderatzeko banaka aztertu behar dira.

3.1.2 Azelerometroa

Azelerometroa, jasaten dituen azelerazio indarrak neurtzen dituen sentsore bat da. Orokorrean 3 ardatzeko azelerometroak erabiltzen dira 3 dimentsioko neurketak eskuratzeko (x, y, z).

Newtonen bigarren legearen arabera [Ruiza et al., 2004] (3.1) ekuazioa betetzen da. Hau da, gorputz baten gainean eragindako indarra, bere masaren eta jasandako azelerazioaren menpe dago, eta ondorioz, azelerazioa, objektuaren gainean eragindako indarrarekiko

3.1 Irudia: *MPU-9250 gailua.*
Iturria: Sparkfun. https://n9.cl/smqgn

proportzionala izango da masa konstantea denean (3.2). Gure kasuan indar hori grabitazio indarra izango da (g).

$$F = m \cdot a \tag{3.1}$$

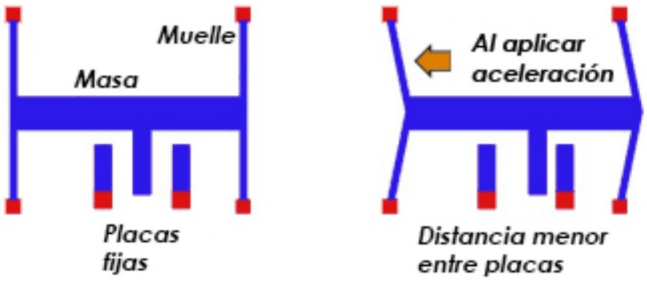
$$a = \frac{F}{m} \tag{3.2}$$

Grabitazio indarren neurketa honi azelerometroaren neurketa estatikoa deritzo eta objektu batek ardatz bertikalarekiko jasaten duen desbiderapen angelua (grabitazio indarraren norabidearekiko zenbat inklinatzen den ardatz bertikala) kalkulatzeko erabiltzen da. Neurketa estatikoaz aparte, azelerometroarekin neurketa dinamikoak ere egiten dira, dardarak, kolpeak, mugimendu txikiak... neurtzeko.

Azelerometro mota desberdinak daude eta fabrikatzaile bakoitzak bere diseinu propioak osatzen ditu, baina funtsean denek egitura berdinak erabiltzen dituzte. Hiru dira azelerometro klase nagusiak:

MEMS azelerometro kapazitiboak

Azelerometro mota hau izaten da ohikoena [AZL, 2020] izan ere, proiektu honetan erabili den MPU-9250 gailuak mota honetako azelerometro bat du barneratua. MEMS (*MicroElectroMechanical-Systems*) [Iannacci, 2017] egiturako azelerometro hauek funtsean kondentsadore baten egitura dute. Masa jakin bat ezartzen da malguki batzuetara lotua eta bestetik kapazitate plaka batzuk ezartzen dira posizio finko batean. (ikus 3.2 irudia). Azelerazio bat gertatzen denean, masak desplazamendu bat jasaten du eta masaren eta plaken arteko distantzia aldatzen da kapazitatearen aldaketa bat sortuz. Kapazitate aldaketa hauek aztertuz neurtu daiteke jasandako azelerazioa.



3.2 Irudia: *MEMS azelerometro kapazitiboaren egitura.*
Iturria: Ingeniería Mecafenix https://n9.cl/t3b02

Azelerometro mota hauen abantaila nagusia beren tamaina txikia da. Baina erabilera industrialerako zehaztasun txikia dute beraz gailu eramangarrietan erabiltzen dira gehienbat.

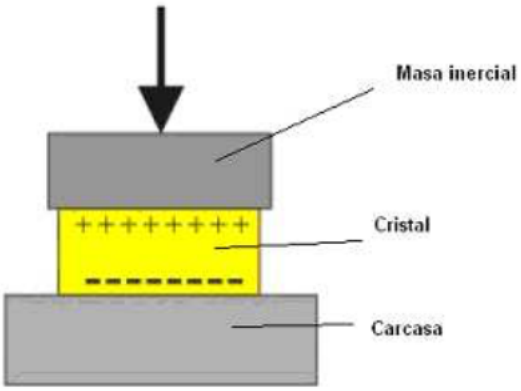
MPU-9250 gailuak duen azelerometroak neurketa tarte desberdinak ditu: $\pm 2g$, $\pm 4g$, $\pm 8g$ eta $\pm 16g$ tarteko neurketak egin ditzake eta eman nahi zaion erabileraren arabera aukeratzen da zein den tarte egokiena. Tarte txiki batek zehaztasun handia eskaintzen du baina neurtzen diren indarren muga kontuan izanik ezingo dira balio oso altuak neurtu. Beste aldetik tartea haundia bada, sentsoreak sentsibilitate handiagoa izango du baina ez da tarte txikian bezain zehatza izango.

Proiektu honetan planteatu diren jardueretarako $\pm 8g$-ko tarte bat nahikoa da sentsibilitatearen eta zehaztasunaren arteko oreka mantentzen delako eta, ez direnez oso indar handiak eragingo, oso zaila izango da tartetik irteten den balio bat jasatea.

Azelerometro piezoelektrikoak

Azelerometro hauek kristal piezoelektrikoen propietateetan oinarritzen dira. "Piezo", grezieratik datorren hitza da eta "estutu"esan nahi du. Hemendik azelerometro mota hauen funtzionamendua ondoriozta daiteke.

Kristalaren deformazioa eragitean (masa zehatz batek eragindako deformazioa azelerazio bat jasatearen ondorioz), bere ezaugarri elektrikoak aldatzen dira eta korronte elektriko bat sortzen da. Korronte horren azterketa eginez ondoriozta daiteke zenbatekoa izan den azelerazioa.



3.3 Irudia: *Azelerometro piezoelektrikoa.*
Iturria:[MAM, 2008]

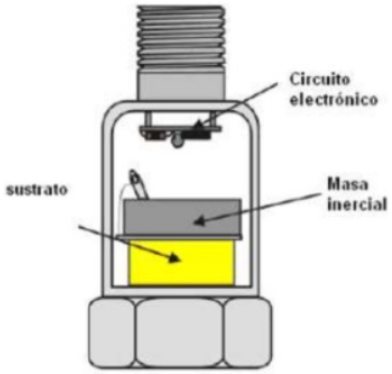
Sentsore hauen egitura 3.3 irudian ikusten dena da. Beheran oinarri finko bat ezartzen da, mugituko ez dena, eta ondoren masa mugikor baten eta oinarriaren artean kristala ezartzen da. Azelerazio bat gertatzen denean, masak kristala estutuko du korronte elektriko bat sortuz.

Azelerometro mota hauek sentsibilitate eta zehaztasun oso altuak dituzte eta horregatik gehienbat neurketa sismiko oso zehatzak lortzeko erabiltzen dira.

Azelerometro piezoerresistenteak

Sentsore klase hau ez da azelerometro piezoelektrikoaren oso desberdina. Oinarrian funtzionamendua antzekoa da baina oraingoan masaren eta oinarriaren artean jartzen den elementua, kristal piezoelektriko bat izan ordez, substratu bat jartzen da (ikus 3.4 irudia).

Sentsoreak azelerazio bat jasaten duenean, masak substratua estutzen du honen erresistentzia aldatuz.



3.4 Irudia: *Azelerometro piezoerresistentea.*
Iturria:[MAM, 2008]

Erresistentzia hori zirkuitu baten elementu bat da (azelerometroan bertan integratua dagoena) eta bertatik pasatzen den korrontearen intentsitatea aztertuz kalkula daiteke zenbatekoa izan den jasandako azelerazio indarra.

3.1.3 Giroskopioa

Giroskopioa objektu baten orientazioa zehazten laguntzen duen sentsorea da. Orientazioari buruzko informazioa eskaintzeko, sentsore hauek abiadura angeluarra neurtzen dute. 3.3 ekuazioan ikusten denez, abiadura angeluarrak ardatz batekiko jasaten den desplazamendu angeluarra ($d\theta$) denbora unitatean adierazten du, eta bere unitatea radian segundoko edo gradu segundoko da.

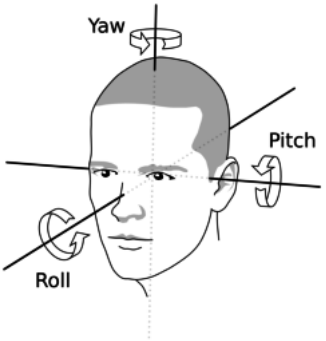
$$\omega = \frac{d\theta}{dt} \tag{3.3}$$

Abiadura angeluarra kalkulatuz, giroskopioak objektu batek jasaten dituen mugimenduak neurtzen ditu. Norabidearen arabera hiru abiadura angeluar neurketa aurkitzen dira (ikus 3.5 irudia):

- *Roll*. Biraketa horizontala objektua aurretik begiratuta (3.5 irudiaren adibidean buruaren mugimendua eskuinera eta ezkerrera inklinatuz).

- *Pitch*. Biraketa bertikala objektua aurretik begiratuta (3.5 irudiaren adibidean buruaren mugimendua gora eta behera baiezko keinua eginez).

- *Yaw*. Biraketa horizontala objektua goitik begiratuta (3.5 irudiaren adibidean buruaren mugimendua eskuinera eta ezkerrera ezezko keinua eginez).



3.5 Irudia: *Roll, Pitch, yaw.*
Iturria:[Jantunen et al., 2016]

Giroskopio mota desberdinak aurki daitezke: laser-eraztun, zuntz optiko, likido eta bibrazio giroskopioak. Lan honetan erabili den MPU-9250 gailuak bibrazio motako MEMS giroskopio bat du bere barnean.

MPU-9250 gailuko giroskopioak neurketa tarte desberdinak ditu: $\pm 250^{\circ}/s$, $\pm 500^{\circ}/s$, $\pm 1000^{\circ}/s$ eta $\pm 2000^{\circ}/s$ tarteko neurketak egin ditzake eta eman nahi zaion erabileraren arabera aukeratzen da zein den tarte egokiena. Azelerometroarekin gertatzen zen bezala, tarte txiki batek zehaztasun haundia eskaintzen du baina neurtzen den abiadura angeluarraren muga kontuan izanik, ezingo dira bira oso azkarrek sortzen duten balio altua neurtu. Beste aldetik tartea haundia bada, sentsoreak sentsibilitate haundiagoa izango du baina ez da tarte txikian bezain zehatza izango.

Proiektu honetan planteatu diren jardueretarako $\pm 1000^{\circ}/s$-ko tarte bat nahikoa da sentsibilitatearen eta zehaztasunaren arteko oreka mantentzen delako eta ez da espero tarte horretatik irteten den balioak jasatea.

3.1.4 Magnetometroa

Sentsorearen azken elementu hau eremu magnetikoak neurtzeko eta eremu magnetiko zehatz baten norabidea eta indarra neurtzeko gai da [Bai and Bai, 2019] [You, 2018]. MPU-9250 gailu honetan, eremu magnetiko zehatz hori lurraren ipar magnetikoa da. Hau da, elementu honi esker, MPU-9250 gailua gai da momenturo iparra non dagoen aztertu eta honekiko duen desbiderapena neurtzeko.

MPU-9250 gailuan erabiltzen den magnetometroa, MEMS motako teknologiarekin dago eraikia. 3 ardatzetako Hall-efektuko [Ramsden, 2011] sentsore magnetiko batez baliatzen da iparra non dagoen detektatzeko ardatz bakoitzean.

3.1.5 Komunikazioak

Aurreko ataletan aipatu den bezala, mikroprozesadorearen eta MPU-9250 gailuaren artean komunikazioa egon behar da gailuak eskuratzen dituen datuak gero mikroprozesadoreak tratatu ahal izateko. Horretarako I^2C busa eta SPI busak dira aukera bakarra. Proiektu honetan bi komunikazio busak lantzea erabaki da gailuak eskaintzan dituen komunikazio aukera posible guztiak ikertu eta lantzeko.

MPU-9250 gailua defektuz I^2C busa erabiliz komunikatzeko dago prestatua, eta SPI busa erabiltzeko aldaketa fisiko bat gauzatu behar da:

- MPU-9250 gailuak bi *jumper* ditu SJ1 eta SJ2, eta defektuz SJ2 *jumper*-a GND erreferentziara dago eztainatua, horrek eragiten du I^2C busaren kasuan AD0 hankatxoa beti 0 izatea eta SPI busaren kasuan, SDO hankatxoak 0V jasotzea eta ez bidaltzea daturik hankatxo horretatik (hankatxo hauen garrantzia 3.2 eta 3.3 ataletan azaltzen da). Horregatik, SPI busa erabili nahi baldin bada SJ2 *jumper*-a behar bezala eztainatu behar da dagokion AD0/SDO hankatxora.

3.2 I^2C Busa

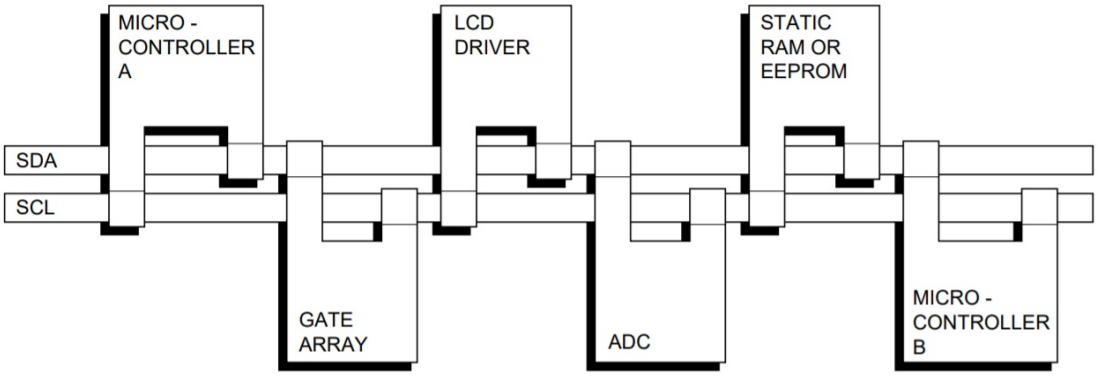
I^2C busa (*Inter-Integrated Circuit)* bi noranzko serie bus sinkronoa da [I2C, 2014]. Zirkuitu bateko elementu desberdinak komunikatzeko erabiltzen da, adibidez periferiko desberdinak mikrokontroladoreekin komunikatzeko. 1982an garatu zuen Philips Semiconduc-

tors enpresak eta hasiera batean telebista barruko txip desberdinen artean komunikatzeko sortu zen.

Bus mota hau bi seinaletaz baliatzen da komunikazioa gauzatzeko. Alde batetik SDA eta bestetik SCL. Honekin batera garrantzitsua den terminologia gehiago dago:

- **SDA** (*Serial Data*) datu lerroa da, hemendik pasatzen dira komunikatu nahi diren bitak.
- **SCL** (*Serial Clock*) erloju-lerroa komunikazioa sinkronizatzeko erabiltzen da.
- **Igorlea** busean datuak idazten dituen osagaia da.
- **Hartzailea** busetik datuak irakurtzen dituen osagaia da.
- **Nagusia** transferentzia kontrolatzen duen osagaia da, hau da, komunikazioari hasiera eman, erloju-seinalea sortu eta kontrolatu, eta transferentziari amaiera ematen dion osagaia da. Nagusiaren papera igorleak zein hartzaileak bete dezake.
- **Morroia** nagusiak transferentzia egiteko aukeratutako osagaia. Hau ere igorlea edo hartzailea izan daiteke.

Bus mota hau multi-master motakoa da, hau da, busaren kontrola lortu dezakeen osagai bat baino gehiago konekta daiteke busera.3.6 irudian aurki daiteke I^{2}C konexio adibide bat 2 mikrokontroladore eta hainbat periferiko lotuta bus berdinera.



3.6 Irudia: *I^{2}C konexio adibidea*
Iturria:[I2C, 2014]

3.2.1 Byte baten transmisioa

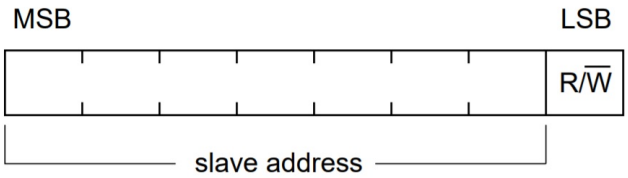
Datuak transferitzerako orduan, komunikazioak baldintza eta urrats batzuk bete behar ditu (transmisioren hasiera eta amaieraren definizioa, datuak ondo bidali edo jaso direla adierazten duen oharra...).

SDA lerroan jartzen diren datu guztiek 8 biteko luzera izan behar dute eta MSB (*Most Significant Bit*) transmititzen da lehenengo.

Komunikazio urratsak modu argi batean azaltzeko adibide bat erabiliko da. Byte baten transmisioa 3.8 irudian adierazten da.

Lehenik START baldintza egon behar da, datu-transferentzia bati hasiera emateko baldintza delako. Erloju-lerroa (SCL) maila altuan (H) dagoenean eta datu lerroa (SDA) maila altutik (H) baxura (L) pasatzen denean, gertatzen da START baldintza. START baldintza hau, transmisioa egin nahi duen gailuak transmititzen du, nagusi gisa jokatuko duenak (transmisio busa libre badago gauzatuko da).

Transmisio bateko lehenengo bytearen (ikus 3.7 irudia) lehenengo 7 bitek morroiaren helbidea adierazten dute eta 8. bitak, hau da, LSB-ak (*Least Significant Bit*) eragiketa zein izango den adierazten du. Bere balioa 0 bada nagusiak idazketa gauzatuko du morroian, eta bere balioa 1 bada nagusiak irakurketa gauzatuko du morroian.

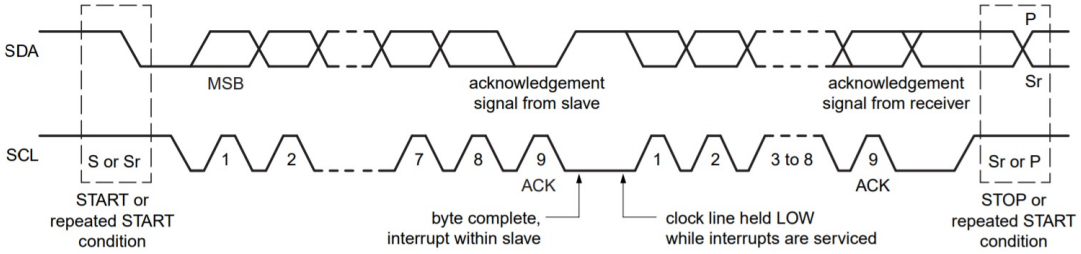


3.7 Irudia: *Lehenengo bytearen egitura*
Iturria:[I2C, 2014]

Ondoren byte bat transmitituko du nagusiak. Bertan, morroiaren helbidea eta eragiketa mota adierazten du. Morroiak onespen-seinalea (ACK *acknowledgment*) bidaliko du libre badago. Nagusiak SDA lerroa libre uzten du morroiak maila baxura (L) pasa dezan 9. pultsuan (3.8 irudian agertzen den SDA 9. bita). Pasatzen ez badu, NACK bat (*Not Acknowledgment*) egon dela ondorioztatzen da eta nagusiak jakingo du morroiak ez duela informazioa eskuratu.

ACK egon bada, SDA lerroa berriro maila altura (H) pasatzen denean igorleak datua transmitituko du busetik. Bukaeran, hartzaileak ACK seinalea aktibatuko du (hartzailea nagu-

sia bada ez da beharrezkoa ACK seinalea, hartzaileak berak erabakitzen duelako transmisioa bertan amaitzen den edo ez) eta beste datu bat transmitituko da edo STOP baldintza transmitituko du nagusiak. Amaierako baldintza hau START baldintzaren antzekoa da, SCL maila baxuan (L) dagoenean eta SDA maila baxutik maila altura pasatzen denean, amaitutzat ematen da komunikazioa eta busa libre geratzen da berriro beste nagusi batek datu-transferentzia bat egin nahi duen arte.



3.8 Irudia: *I^2C komunikazioa*
Iturria:[I2C, 2014]

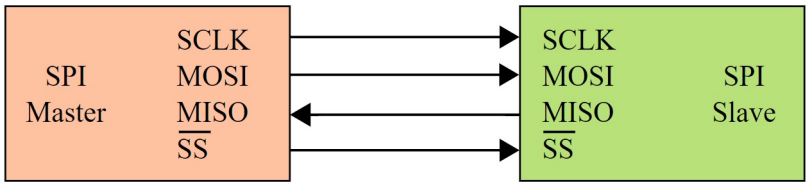
3.3 SPI Busa

SPI (*Serial Peripheral Interface*), Motorola enpresak 1985 urtean garatu zuen bus serie sinkrono bat da, eta I^2C ren modura, nagusi-morroi egitura erabiltzen du komunikazioa gauzatzeko [Kalinsky and Kalinsky, 2002]. 4 lerro erabiltzen ditu transmisioak ahalbidetzeko: 2, transferentziaren kontrola mantentzeko, eta beste 2, datuak bidali eta jasotzeko, horregatik bus hau Full Duplex motakoa da (datuak daramatzaten seinaleak bi noranzkoetan doaz aldi berean).

SPIk lehen aipatu den bezala 4 lerro seinale erabiltzen ditu komunikazioak gauzatzeko nagusiaren eta morroiaren artean (ikusi 3.9 irudia):

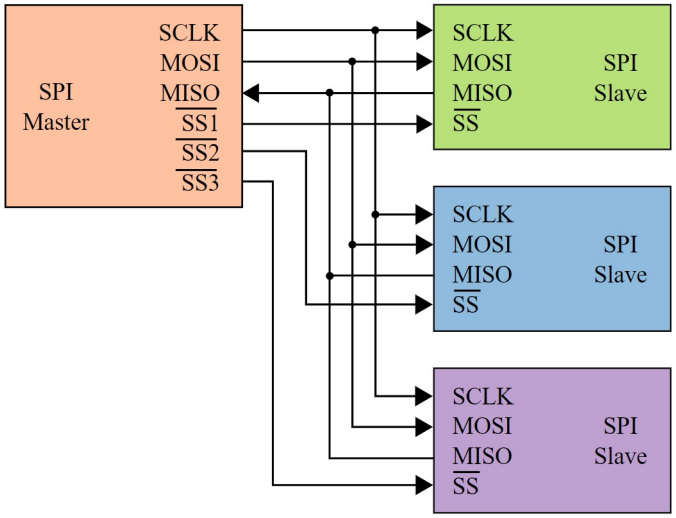
- **SCLK** (*Serial Clock*), nagusiak sortzen duen erloju-seinalea da. Morroi guztiek jasotzen dute erloju-seinale horretara sinkronizatzeko.
- **MOSI** (*Master Out Slave In*), nagusiak morroiari informazioa bidaltzeko lerroa da.
- **MISO** (*Master In Slave Out*), morroiak nagusiari informazioa bidaltzeko lerroa da.
- **CS/SS** (*Slave Select*), morroi bakoitzarentzako aukeraketa seinale propio bat. Lerro honi esker nagusiak morroi bat aukeratu dezake komunikatzeko. SS pin hau maila

baxuan (L) dagoenean, morroia nagusiarekin komunikatzen ari da eta maila altuan (H) badago, ez du kasurik egiten nagusiak bidalitako agindu edo informazioari.

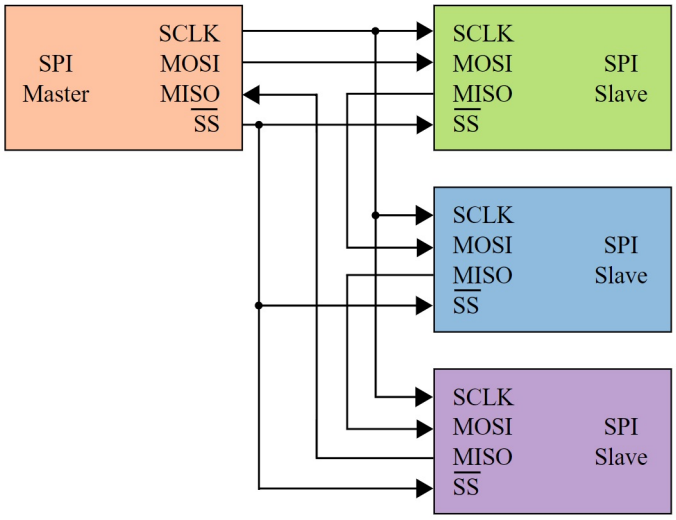


3.9 Irudia: *SPI protokoloaren lerroak*
Iturria: Wikimedia Commons. https://n9.cl/tolps

3.9 irudian ikusten da nagusi batek eta morroi batek osatutako egitura, baina normalean nagusi bat eta morroi bat baina gehiagok osatzen dute egitura 3.10 irudian ikus daitekeen moduan. Nabari daiteke morroi anitzeko egitura honek duen ahulezia: morroi bakoitzeko SS lerro bat behar denez geroz eta morroi gehiago, orduan eta kable gehiago gehitu beharko dira. Baina egitura honi irtenbidea emateko, kate egiturako beste lotura bat erabil daiteke (ikus 3.11 irudia).



3.10 Irudia: *SPI protokoloaren lerroak morroi anitzekin*
Iturria: Wikimedia Commons. https://n9.cl/d20lp



3.11 Irudia: *SPI protokoloaren lerroak morroi anitzekin kate egitura*
Iturria: Wikimedia Commons. https://acortar.link/NKSo8J

3.3.1 Transmisio baten formatua

SPI bitartez datu transmisio bat egin nahi denean (read edo write eragiketa batekin) gutxienez 2 byte transmititzen dira. Lehenengo bytean SPI erregistroaren helbidea bidaltzen da eta hurrengoan (edo hurrengoetan) datuak, beti egitura berdina mantenduz (3.12 irudian agertzen den bezala). Datuak beti MSB bitetik hasita bidaltzen dira, beraz transmisio batean bidaliko den lehenengo bitak irakurketa (1) edo idazketa (0) bat den adieraziko du eta hurrengo 7 bitak erregistroaren helbidea. Hortik aurrera bidaltzen diren byteak transmititu nahi den informazioa daramate.

Transmisioa hasteko, nagusiak, berak eta morroiak onartzen duten erloju-seinale bat konfiguratzen du eta ondoren morroia aukeratzen du bere SS lerroa maila baxura (L) pasatuz. Hortik aurrera nagusiak informazioa bidaltzen du MOSI lerrotik eta morroiak MISO lerrotik aldi berean 3.13 irudian ikusten den adibidean bezala.

Erloju-pultsu bakoitzean bit bat transmititzen da MOSI eta MISO lerroetatik, baina pultsu hori modu desberdinetara konfiguratu daiteke. Modifikazio honetaz nagusia arduratzen da eta SCLK pultsua sortzeaz gain CPOL (*clock polarity*) eta CPHA (*clock phase*) definitzen ditu SPI Mode zein den adierazteko [Mot, 2000].

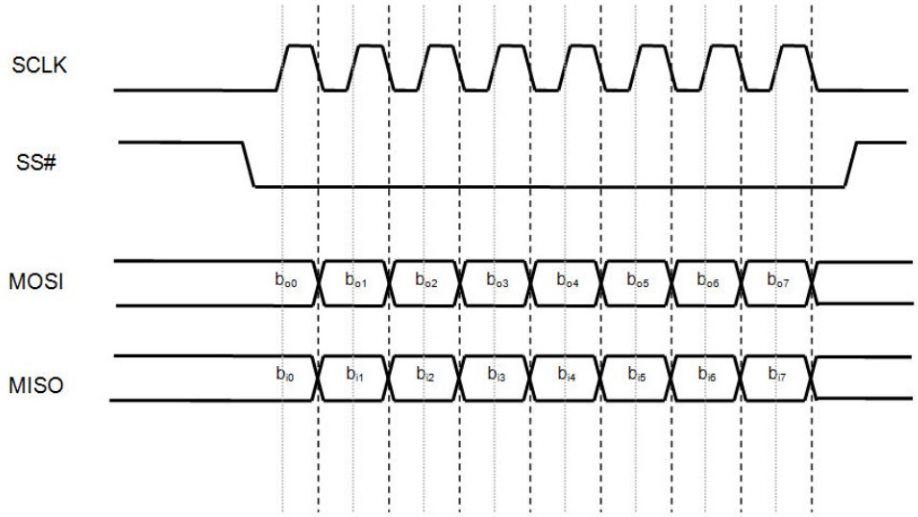
SPI Address format

MSB							LSB
R/W	A6	A5	A4	A3	A2	A1	A0

SPI Data format

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

3.12 Irudia: *SPI transmisio byten egitura*
Iturria:[I2C, 2014]

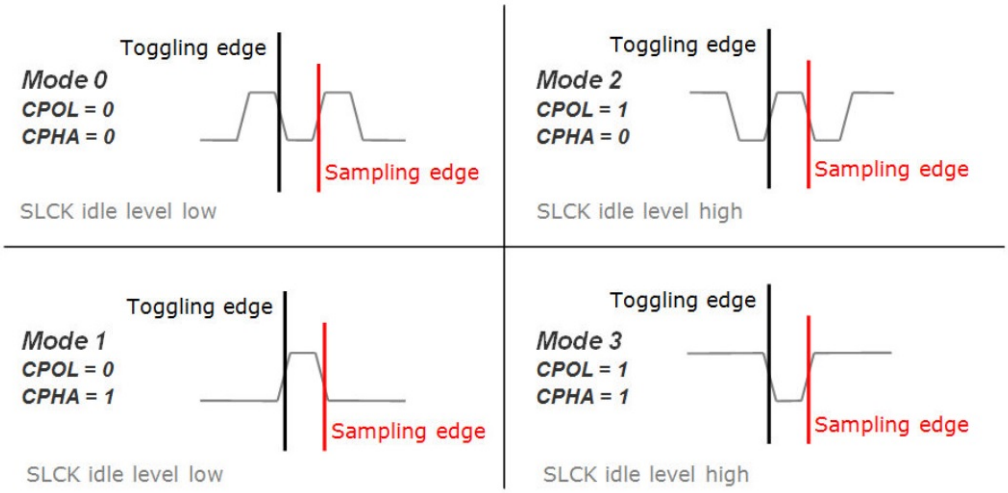


3.13 Irudia: *SPI transmisio baten adibidea*
Iturria: byteparadigm. https://cutt.ly/pWeOZB2

- **CPOL** erlojuaren polaritatea adierazten du. 0 edo 1 balioa har dezake.
  - CPOL=0 bada, erlojua 0n edo maila baxuan egongo da gelditua (idle egoeran) eta ziklo bakoitza 1era edo maila altura igotzen den pultsu bat da.
  - CPOL=1 bada, alderantziz. Erlojua 1en edo maila altuan egongo da gelditua (idle egoeran) eta ziklo bakoitza 0ra edo maila baxura jaisten den pultsu bat da.

- **CPHA** erlojuaren pultsuei dagozkien datuen biten unea zehazten du. 0 edo 1 balioa har dezake.
  - CPHA=0 bada, datuak aurreko erloju-zikloaren amaierako ertzean aldatzen dira eta datuak momentuko erloju-zikloaren hasierako ertzean hartzen dira.
  - CPHA=1 bada, datuak momentuko erloju-zikloaren hasierako ertzean aldatzen dira eta datuak momentuko erloju-zikloaren amaierako ertzean hartzen dira.

3.14 irudian aurkitzen dira konbinazio posible guztien, hau da, SPI Mode guztien azalpen grafikoak.



3.14 Irudia: *SPI Moduak*
Iturria: byteparadigm. https://cutt.ly/pWeOZB2

3.4 Arduino

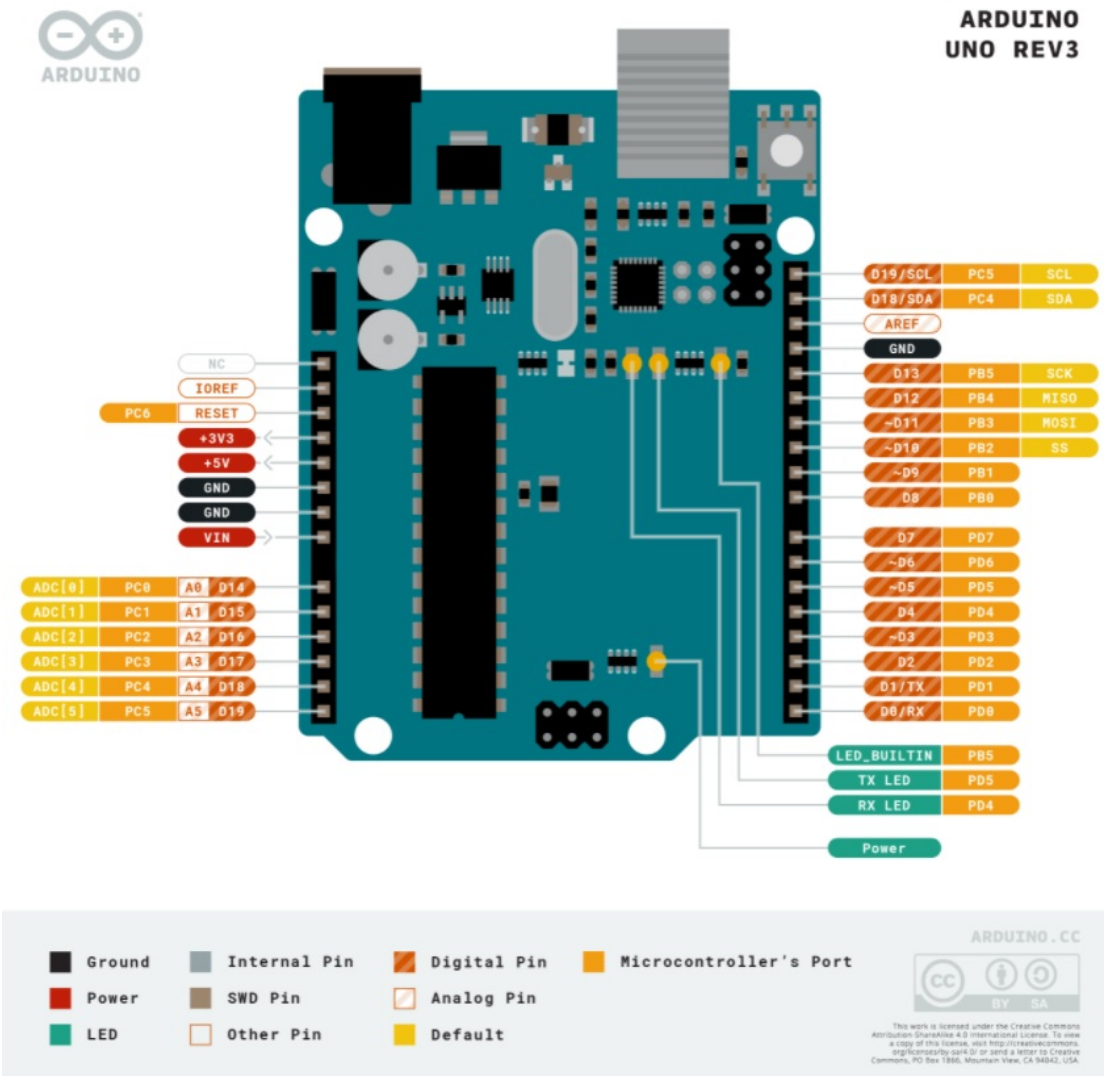
Proiektua Sistema txertatuen diseinua irakasgaira eramatea bilatzen da lan honekin. Horrekin batera, programak Explorer 16 plakan inplementatu beharko dira, eta horretarako, ondo funtzionatzen duen kode bat edo oinarri minimo bat lortu behar da. Hori horrela izanik, Arduino Uno plaka eta hori kontrolatzeko softwarea (Arduino IDE) erabiltzea erabaki da, gero proiektua Explorer 16 plakara pasatzeko prozesua errazagoa izateko asmoarekin.

Arduino hardware eta software libreko elektronika sortzeko plataforma bat da. 2003an hasi zen proiektua Italiako ikasle batzuen eskutik eta helburua eskuragarriak ziren plaka batzuk sortzea zen, elektronika eta programazioaren erabilera errazteko.

Hainbat Arduino plaka aurki daitezke baina proiektu hau gauzatzeko Arduino Uno [Ard, 2019] plaka erabili da (3.15 irudia). Hauek dira bere ezaugarriak:

- Mikrokontroladorea: ATmega328
- Hankatxoak
  - Sarrera/Irteera digitalak: 14
  - Sarrera analogikoak: 6
  - PWM: 6
  - LED: 13
- Komunikazio motak
  - SPI
  - I^2C
  - UART
- Zirkuituaren elikadura: 5V
- Sarrerako tentsioa: 7-12V
- Mikrokontroladorearen erloju abiadura: 16 MHz
- USB konektorea
- Jack bateria konektorea
- ATmega328P memoria
  - 2KB SRAM
  - 32KB FLASH
  - 1KB EEPROM (azkeneko hau ez da borratzen plaka itzaltzen denean)

Arduino Uno plaka hau elikatzeko bi aukera daude: zuzenean ordenagailura konektatzea USB konektorearen bitartez edo elikadura jack konektoretik eskaintzea energia. Programak plakan kargatzeko USB bidez konektatu behar denez ordenagailura, normalean horren bitartez elikatzen da plaka, baina behin programa kargatuta dagoela, jack konektorea erabil daiteke (EEPROM memoriari esker ez delako galtzen kargatu den azken programa).



3.15 Irudia: *Arduino Uno plakaren hankatxo-mapa*
Iturria: Arduino https://docs.arduino.cc/hardware/uno-rev3

4. KAPITULUA

Inplementazioa

Dokumentuaren kapitulu honetan proposatu diren praktiken azalpena eta ebazpenak aurkeztuko dira, lortutako emaitzak aurkeztuz eta inplementazioan zehar egon diren arazoak eta hauei irtenbidea emateko hartu diren erabakiak azalduz.

Inplementazioarekin zerikusia duten 3 helburu nagusi aurkitzen dira: I^2C bus bidezko komunikazioa erabiltzea, SPI bus bidezko komunikazioa erabiltzea eta MPU-9250 gailuarekin praktika bat garatzea. Bi bus ezberdinen bitartez komunikatzeko bi inplementazio planteatu dira bakoitza bere programa eta hardware muntaiarekin. Hirugarren helburua lortzeko 1.2 atalean planteatu den erabilera praktikoari ebazpen bat eman zaio, aurrerago azalduko dena.

4.1 Materiala

I^2C bus bidezko ebazpenean eta SPI bus bidezko ebazpenean erabili den materiala hurrengoa izan da:

- MPU-9250 gailua
- Arduino Uno plaka
- Arduino IDE softwarea
- konexio kableak

Erabilera praktikoa aurrera eramateko hau izan da erabili den materiala:

- MPU-9250 gailua
- Arduino Uno plaka
- Arduino IDE softwarea
- konexio kableak
- Beso robotikoa

4.1.1 MPU-9250 hankatxoen eskema

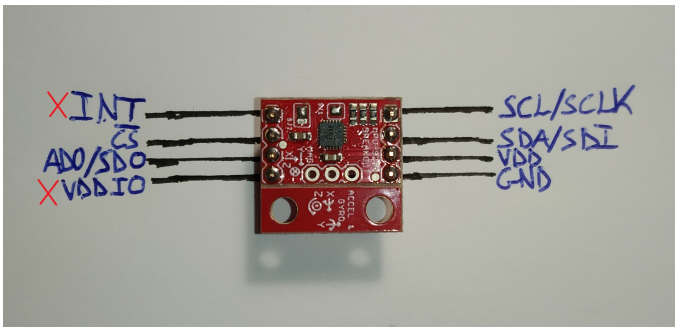
Inplementazio bakoitza azaltzen hasi baino lehen ezinbestekoa da erabiliko den sentsorearen egitura ezagutzea. 4.1 irudian ikusten denez, guztira hogeita lau hankatxo daude, baina horietako batzuk erreserbatuak daude eta proiektu honetarako ez dira denak beharrezkoak. Hogeita lau hankatxo horietatik sei bakarrik dira beharrezkoak:

Pin Number	Pin Name	Pin Description
1	RESV	Reserved. Connect to VDDIO.
7	AUX_CL	I²C Master serial clock, for connecting to external sensors
8	VDDIO	Digital I/O supply voltage
9	AD0 / SDO	I²C Slave Address LSB (AD0); SPI serial data output (SDO)
10	REGOUT	Regulator filter capacitor connection
11	FSYNC	Frame synchronization digital input. Connect to GND if unused.
12	INT	Interrupt digital output (totem pole or open-drain)
13	VDD	Power supply voltage and Digital I/O supply voltage
18	GND	Power supply ground
19	RESV	Reserved. Do not connect.
20	RESV	Reserved. Connect to GND.
21	AUX_DA	I²C master serial data, for connecting to external sensors
22	nCS	Chip select (SPI mode only)
23	SCL / SCLK	I²C serial clock (SCL); SPI serial clock (SCLK)
24	SDA / SDI	I²C serial data (SDA); SPI serial data input (SDI)
2 – 6, 14 - 17	NC	Not internally connected. May be used for PCB trace routing.

4.1 Irudia: *MPU-9250 hankatxoen eskema*
Iturria:[IPS, 2014]

- 9 hankatxoa (AD0/SDO): Hankatxo honek funtzio desberdina du erabiltzen den komunikazio busaren arabera.
  - I^2C busaren kasuan (AD0) hankatxo honekin MPU-9250ren morroi helbidearen azken bita (LSB) definitzen da. Hankatxo honi esker bi MPU-9250 gailu konekta daitezke plaka berdinera, bati AD0 bita Low egoeran jarriz eta besteari High egoeran jarriz, bi helbide desberdin lortuko lirateke bi sentsoreen helbideak desberdintzeko. MPU-9250 gailuaren kasuan, AD0 hankatxoa Low egoeran badago, helbidea 1101100**0** (68) izango da eta High egoeran badago 1101100**1** (69)
  - SPI busaren kasuan (SDO) hankatxo hau arduratzen da MISO lerroan datuak jartzeaz (*Serial Data Output*).
- 13 hankatxoa (VDD): Tentsio elikadura hankatxoa
- 18 hankatxoa (GND): Lurraren erreferentzia
- 22 hankatxoa (nCS): SPI busa erabiltzen denean bakarrik da erabilgarria. Slave select aukeraketa egiteko erabiltzen da.
- 23 hankatxoa (SCL/SCLK): Hankatxo hau erloju-seinalearen arduraduna da, SCL I^2Cren kasuan eta SCLK SPIren kasuan
- 24 hankatxoa (SDA/SDI): Hankatxo honek 2 funtzio ditu erabiltzen den komunikazio busaren arabera.
  - I^2C busaren kasuan (SDA) datu lerroaren papera betetzen du.
  - SPI busaren kasuan (SDI) hankatxo hau arduratzen da MOSI lerroan datuak jartzeaz (*Serial Data Input*).

Sentsorean bertan, fisikoki 4.2 irudian agertzen diren posizioetan daude hankatxoak.



4.2 Irudia: *MPU-9250 hankatxoen posizioa sentsorean*

4.2 I^{2}C bus inplementazioa

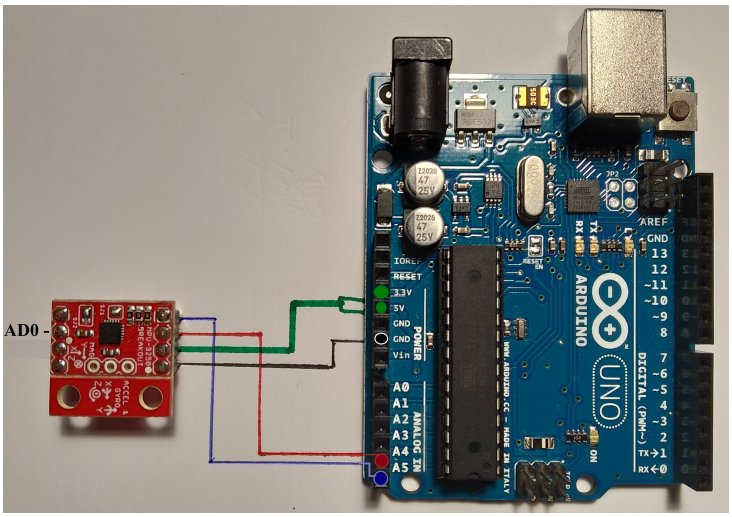
Proiektuaren zati honetan I^{2}C bus bidezko komunikazioa erabiliz, MPU-9250 gailua eta Arduino Uno plakaren artean komunikazioa ahalbidetzea eta eskuratutako datuak aztertzea lortu da.

Helburua: MPU-9250 gailua konfiguratu komunikazioa plakarekin I^{2}C busaren bitartez gauzatzeko, giroskopioaren eta azelerometroaren datuak eskuratu, eta lortutako datuak aztertu.

Loturak: Hardwarearen muntaia egokia izateko eta plakarekin komunikazioa ahalbidetzeko 4.3 irudian agertzen diren konexioak egin behar dira, hau da:

- SCL hankatxoa A5-era konektatu
- SDA hankatxoa A4-ra konektatu
- VDD hankatxoa 3.3V edo 5V-eko elikadura iturrira konektatu
- GND hankatxoa Ground erreferentziara konektatu
- AD0 hankatxoa Ground erreferentziara helbidearen azkeneko bita 0 izatea nahi bada eta 3.3V edo 5V-eko erreferentziara helbidearen azkeneko bita 1 izatea nahi bada.

Kodea: Behin loturak eginda kodea programatu da Arduino IDE softwarea erabiliz. I^{2}C busa erabiltzeko eta programatzeko "Wire.h"liburutegia erabili da eta I2Cread eta I2CwriteByte



4.3 Irudia: *MPU-9250 hankatxoen konexioa I²C busa erabiliz*

funtzioak programatu dira komunikazioa ahalbidetzeko. Kode honekin (kode osoa eranskinak atalean dago) azelerometrotik eta giroskopiotik eskuratzen diren datuak erakustea bilatzen da.

I2Cread (4.4 irudia): Funtzio honekin I²C bus bidezko komunikazioaz baliatuz irakurketak egitea lortzen da. Sarrera moduan lau parametro pasatzen dira:

- Helbidea: Irakurketa zein sentsoretik egingo den adierazten du.
- Erregistroa: Aukeratutako elementuko zein erregistrotik irakurriko den adierazten du.
- Byte kopurua: Irakurriko den byte kopurua adierazten du.
- Data: Behin datuak irakurrita hauek jasotzeko memoria tartea.

Funtzionamendua hurrengoa da: Lehenik aukeratutako elementuarekin komunikazioa hasten da honen helbidea adieraziz, ondoren irakurketa zein erregistrotik egin nahi den idazten da busean. Behin hau eginda, irakurketa prozesua hasten da adierazitako byte kopurua Data memoria tartean gordez.

```
void I2Cread(uint8_t Helb, uint8_t Erreg, uint8_t Nbytes, uint8_t* Data)
{
   Wire.beginTransmission(Helb);
   Wire.write(Erreg);
   Wire.endTransmission();

   Wire.requestFrom(Helb, Nbytes);
   uint8_t index = 0;
   while (Wire.available())
      Data[index++] = Wire.read();
}
```

4.4 Irudia: *I2Cread funtzioa*

I2CwriteByte (4.5 irudia): Funtzio honekin I^2C bus bidezko komunikazioaz baliatuz idazketa egitea lortzen da. Sarrera moduan hiru parametro pasatzen dira:

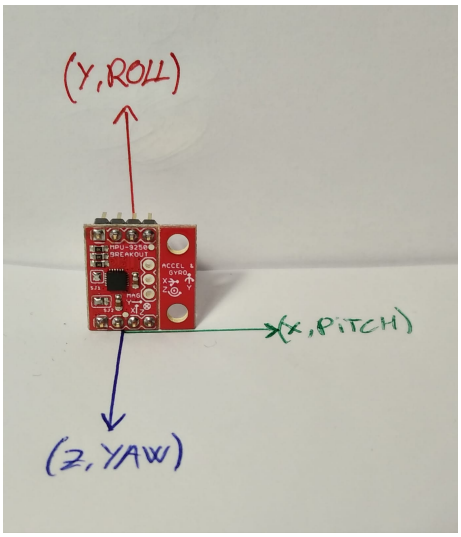
- Helbidea: Idazketa zein sentsoretan egingo den adierazten du.
- Erregistroa: Aukeratutako elementuko zein erregistroan idatziko den adierazten du.
- Data: Idatzi nahi den informazioa.

Funtzionamendua hurrengoa da: Lehenik aukeratutako elementuarekin komunikazioa hasten da honen helbidea adieraziz, ondoren idazketa zein erregistrotan egin nahi den eta idatziko den informazioa zein den adieraziko da busean. Behin prozesu hau amaituta, komunikazioa eteten da.

```
void I2CwriteByte(uint8_t Helb, uint8_t Erreg, uint8_t Data)
{
   Wire.beginTransmission(Helb);
   Wire.write(Erreg);
   Wire.write(Data);
   Wire.endTransmission();
}
```

4.5 Irudia: *I2CwriteByte funtzioa*

Emaitzen analisia: Lortzen diren emaitzak aztertu baina lehen, sentsorearen ardatzak zein norabidetan definitzen diren ikusi behar da, emaitza behar bezala interpretatzeko. 4.6 irudian ikusten den modura kokatzen dira azelerometroaren eta giroskopioaren ardatzak [IPS, 2014].

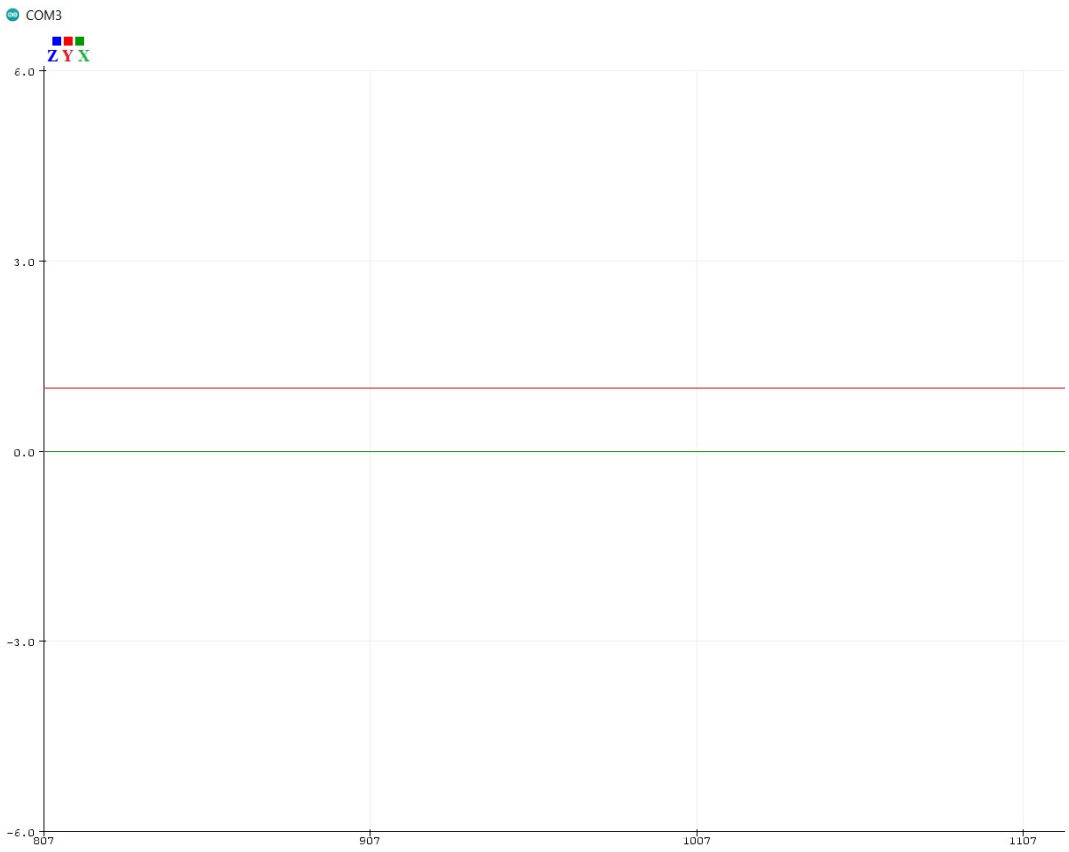


4.6 Irudia: *Azelerometro eta giroskopioaren orientazioa*

MPU-9250 gailua 4.6 irudian duen posizioan mantenduta, azelerometroaren datuak eskuratzen badira, teorikoki *y* ardatzak balio positibo bat hartuko luke (1g-ko indarra jasaten duen balio positiboa) eta beste bi ardatzek (*x, z*) 0 balioa (grabitatearen indarrak ez dielako eraginik egiten ez daudelako grabitatearen norabidean). 4.7 irudian lortu diren emaitzak aztertuz (grafikoaren ordenatuan, sentsoreak jasandako g indarraren balioa adierazten da eta abzisan denbora), ikusten da zuzena dela planteatu den emaitza teorikoa, *y* ardatzak 1g ko indarra jasaten du eta *x, z* ardatzek 0g ko indarra (*z* ardatzaren lerroa ez da ikusten *x* lerroak tapatzen duelako). Posizio honetan mantentzen bada MPU-9250 gailua, giroskopioak ez du mugimendurik detektatuko eta ondorioz balio guztiak 0 izango dira.

Giroskopioaren funtzionamendua zuzena dela aztertzeko, sentsoreari *x* ardatzarekiko +90º-ko biraketa eragin zaio eta ondoren ardatz berdinean -90ºko biraketa ba. Sentsorea posizio berdinean bukatuko du bi biraketak amaitzean baina giroskopioak jasandako mugimenduak detektatuko ditu. Emaitzak aztertuz, 4.8 irudian ikusten denez giroskopioaren funtzionamendua ere egokia da. Lehenengo mugimenduan (*x* ardatzean +90ºko biraketan) *x* balioak 800º/s-ko biraketa bat antzematen du eta mugimendua amaitzean, hau da, posizio berrira iritsi denean sentsorea, berriro 0º/s inguruko balioetara bueltatzen da sentsorea posizio egonkor batean egotera pasatzen delako. Bigarren mugimendua lehenengoaren kontrakoa da (*x* ardatzean -90ºko biraketa) eta ondorioz *x* balioak -800º/s-ko biraketa bat antzematen du gutxi gorabehera. Lehenengo mugimenduaren antzera, sentsorea berriro posizio egonkor batera iristean 0º/s inguruko balioetara bueltatzen da.

Bi mugimenduetan bereiz daiteke *z* eta *y* ardatzetan mugimendua detektatzen dela. Ho-



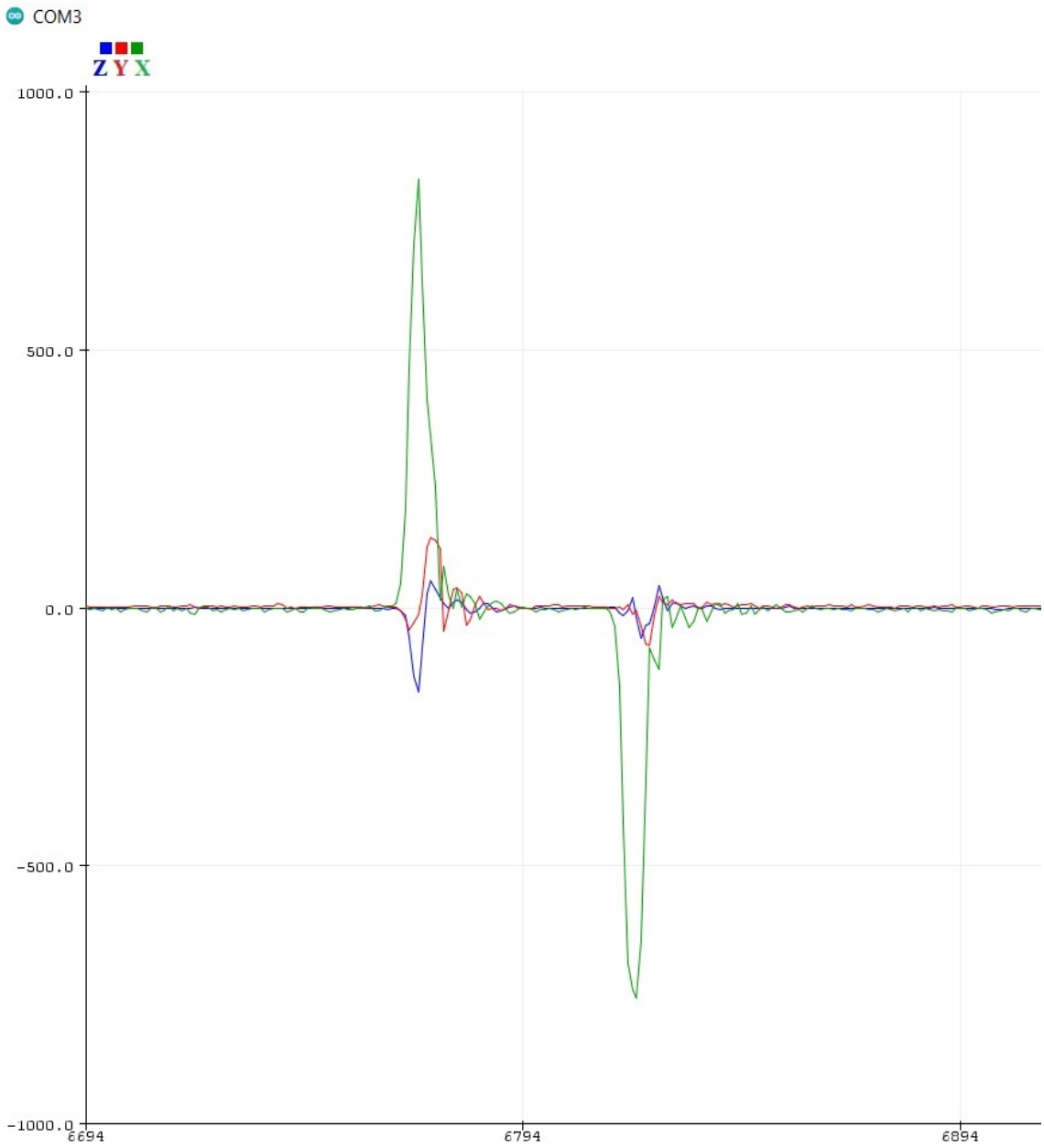
4.7 Irudia: *4.6 irudian duen posizioan lortutako azelerometroaren datuak*

ri gertatzen da sentsorea bakarrik ardatz batean mugitzea oso zaila delako eta biraketa nagusia ardatz bakar batean izanda ere, beste ardatzek biraketa jasaten dute.

Bi biraketa hauek egin diren bitartean azelerometroak eskuratzen dituen datuak aztertzen badira (4.9 irudia) ikus daiteke nola sentsorea, lehenengo biraketa egin ondoren, buruz behera geratzen den eta ondorioz *z* ardatzak -1g ko indarra jasaten duen. Ondoren, bigarren mugimendua gauzatzean, sentsoreak hasieran zuen posiziora itzultzen da eta *y* ardatzak berriro 1g ko indarra jasatera pasatzen da.

4.3 SPI bus inplementazioa

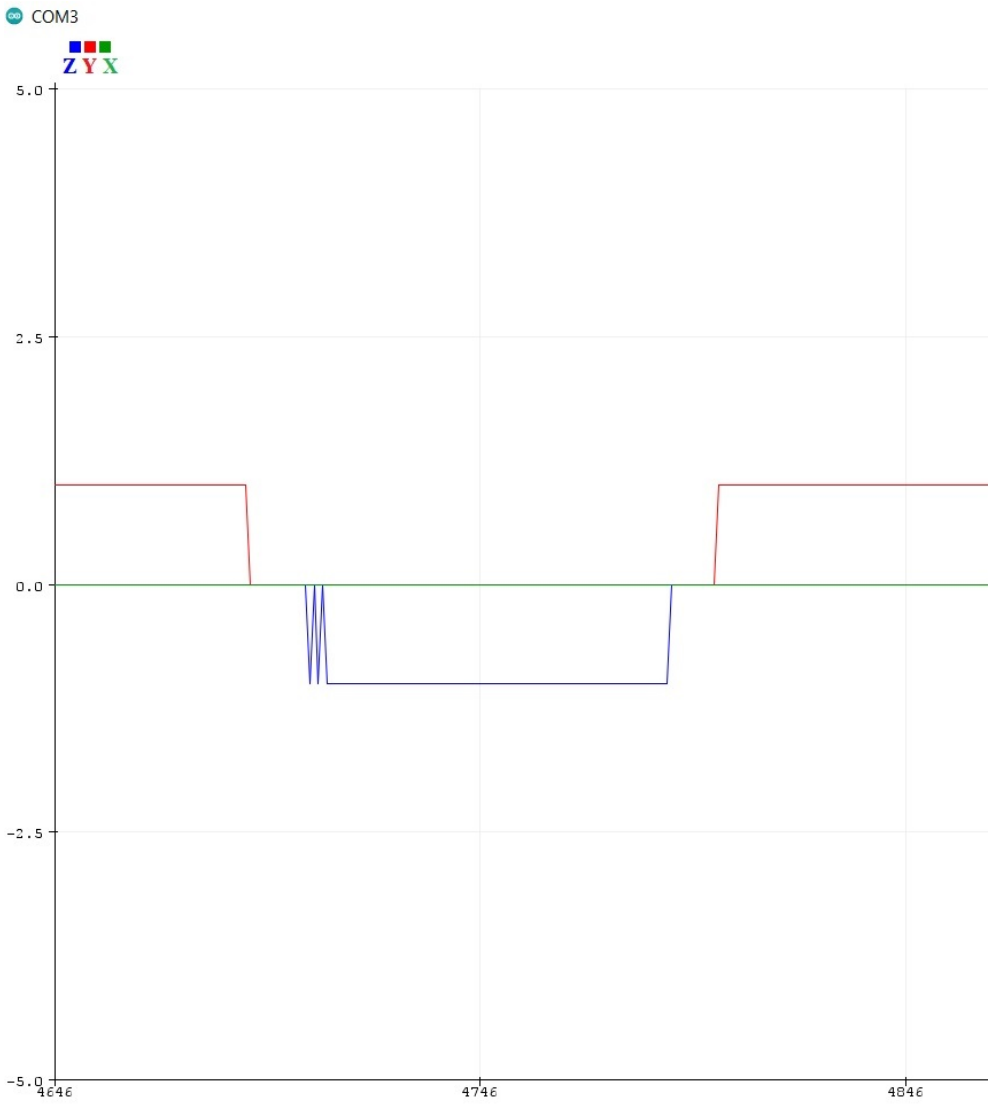
Proiektuaren zati honetan, lehen I^2C busarekin egin den antzera, SPI bus bidezko komunikazioa erabiliko da MPU-9250 gailua eta Arduino Uno plakaren artean komunikazioa ahalbidetzeko.



4.8 Irudia: *x ardatzean +90º eta -90º biraketak sortutako giroskopioaren emaitza*

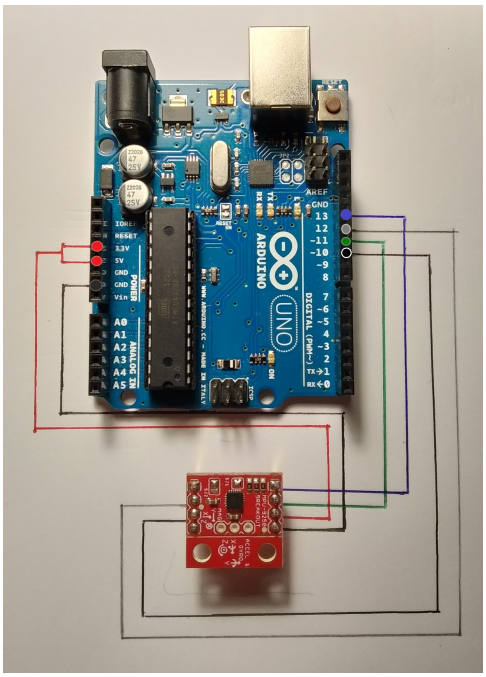
Helburua: MPU-9250 gailua konfiguratu komunikazioa plakarekin SPI busaren bitartez gauzatzeko, giroskopioaren eta azelerometroaren datuak eskuratu, eta lortutako datuak aztertu.

Loturak: Hardwarearen muntaia egokia izateko eta plakarekin komunikazioa ahalbidetzeko 4.10 irudian agertzen diren konexioak egin behar dira, hau da:



4.9 Irudia: *x ardatzean +90º eta -90º biraketak sortutako azelerometroaren emaitza*

- SCLk hankatxoa 13. konexiora lotuta
- SDI hankatxoa 11. konexiora lotuta
- VDD hankatxoa 3.3V edo 5V-eko elikadura iturrira konektatu
- GND hankatxoa Ground erreferentziara konektatu
- SDO hankatxoa 12. konexiora lotuta
- nCS hankatxoa 10. konexiora lotuta



4.10 Irudia: *MPU-9250 hankatxoen konexioa SPI busa erabiliz*

Kodea: Behin loturak eginda, I^2C busarekin egin den bezala kodea programatu da Arduino IDE softwarea erabiliz. SPI busa erabiltzeko eta programatzeko "SPI.h"liburutegia erabili da eta readBytesSPI eta writeByteSPI funtzioak programatu dira komunikazioa ahalbidetzeko. Honekin batera SPI Mode definitu da 3. Modua bezala (3.14 irudian daude modu guztien adierazpen grafikoak). Kode honekin (kode osoa eranskinak atalean dago) azelerometrotik eta giroskopiotik SPI busaren bitartez eskuratzen diren datuak erakustea bilatzen da.

readBytesSPI (4.11 irudia): Funtzio honekin SPI bus bidezko komunikazioaz baliatuz irakurketa egitea lortzen da. Sarrera moduan hiru parametro pasatzen dira:

- Erregistroa: Aukeratutako elementuko zein erregistrotik irakurriko den adierazten du.
- kontagailua: irakurri nahi den byte kopurua
- Bufferra: Irakurriko den informazioa gordetzeko memoria tartea.

Funtzionamendua hurrengoa da: Lehenik SPI komunikazioa konfiguratu behar da SPI data rate aldagaian abiadura adieraziz eta SPI modua (SPI_MODE) definituz. Ondoren, SS hankatxoa maila baxura pasatzen da adierazteko morroi horrekin komunikatu nahi dela

eta irakurri nahi den erregistroaren MSB bitari 1 balioa ematen zaio SPI.transfer(erreg | IRAKURKETA_FLAG) aginduarekin. (IRAKURKETA_FLAG 0x80 balioa du). Gero, kont aldagaian definitu den byte kopurua transmititzen da 0x00 balioarekin (Full Duplex denez aldiro dago informazioa bidaltzen eta jasotzen, baina kasu honetan garrantzitsuena informazioa jasotzea denez 0 balioa bidaltzen da, ez duelako garrantzirik bidaltzen dena). Azkenik, SS hankatxoa maila altura pasatzen da transferentzia amaitutzat emateko morroiarekin.

```
void readBytesSPI(uint8_t erreg, uint8_t kont, uint8_t* buf)
{
  SPI.beginTransaction(SPISettings(SPI_DATA_RATE, MSBFIRST, SPI_MODE));
  digitalWrite(CS_PIN, LOW);
  SPI.transfer(erreg | IRAKURKETA_FLAG);
  uint8_t i;
  for(i=0; i < kont; i++){
    buf[i]=SPI.transfer(0x00);
  }
  digitalWrite(CS_PIN, HIGH);
  SPI.endTransaction();
}
```

4.11 Irudia: *readByteSPI funtzioa*

writeByteSPI (4.12 irudia): Funtzio honekin SPI bus bidezko komunikazioaz baliatuz idazketa egitea lortzen da. Sarrera moduan bi parametro pasatzen dira:

- Erregistroa: Aukeratutako elementuko zein erregistrotan idatziko den adierazten du.
- Data: idatziko den informazioa.

Funtzionamendua hurrengoa da: Lehenik SPI komunikazioa konfiguratu behar da SPI data rate aldagaian abiadura adieraziz eta SPI modua (SPI_MODE) definituz. Ondoren, SS hankatxoa maila baxura pasatzen da adierazteko morroi horrekin komunikatu nahi dela. Gero, idazketa egin nahi den erregistroaren helbidea eta idatzi nahi den datua bidaltzen dira. Azkenik, SS hankatxoa maila altura pasatzen da transferentzia amaitutzat emateko morroiarekin.

Emaitzen analisia:

I^2C bus bidezko komunikazioarekin egin den bezala, MPU-9250 gailua 4.6 irudian duen posizioan mantenduz bi mugimendu egin dira, *x* ardatzarekiko +90º-ko biraketa bat eta

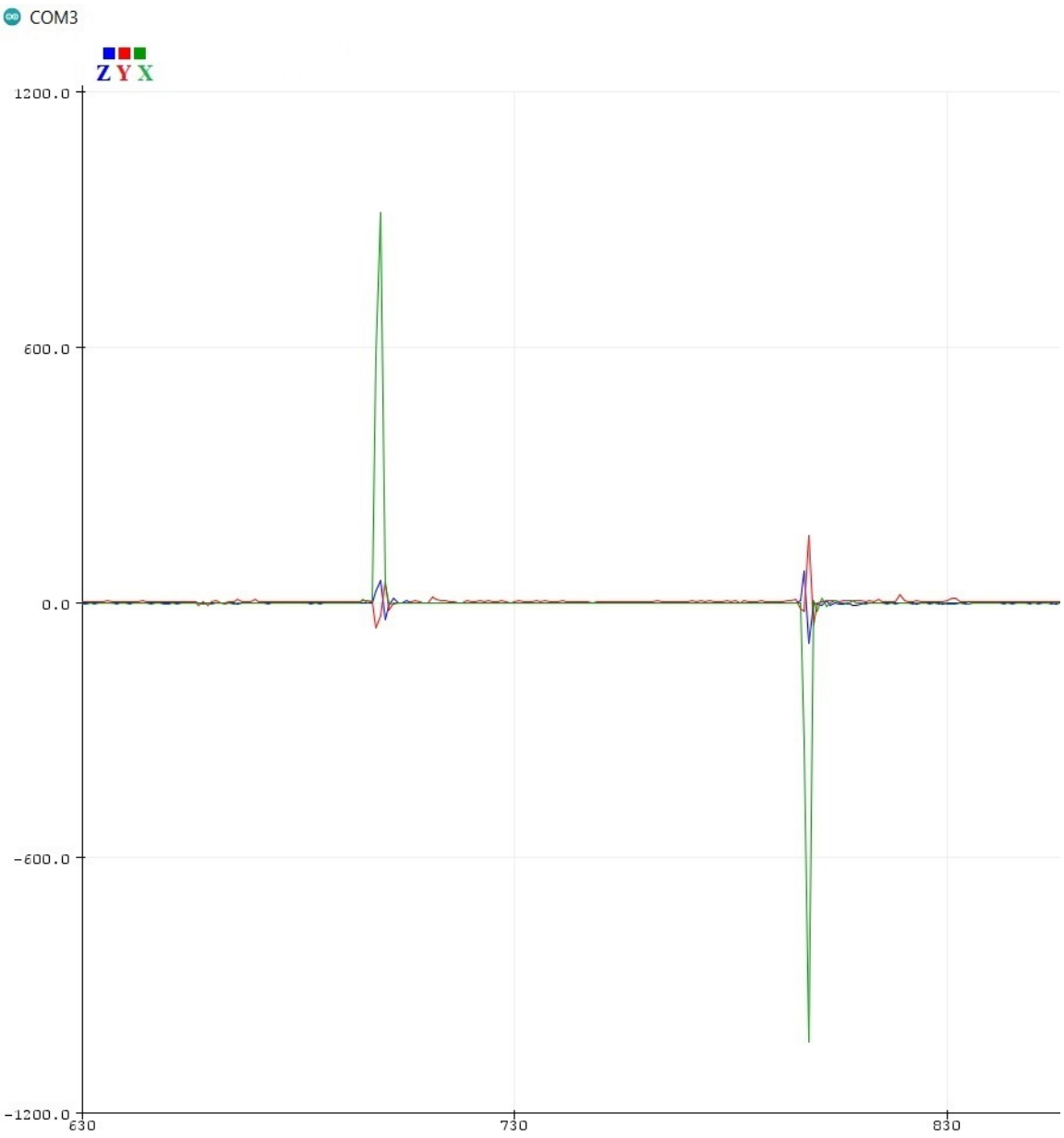
```
void writeByteSPI(uint8_t erreg, uint8_t data)
{
  SPI.beginTransaction(SPISettings(SPI_DATA_RATE, MSBFIRST, SPI_MODE));
  digitalWrite(CS_PIN, LOW);
  SPI.transfer(erreg);
  SPI.transfer(data);
  digitalWrite(CS_PIN, HIGH);
  SPI.endTransaction();
}
```

4.12 Irudia: *writeByteSPI funtzioa*

ondoren ardatz berdinean -90ºko biraketa. Teorikoki, I^2C bus komunikazioan eskuratu diren antzeko datuak eskuratu behar dira dena ondo baldin badago.

Giroskopioak lortu dituen balioak aztertuz (4.13 irudia) ikus daiteke lehen eskuratu diren balio oso antzekoak ematen direla hemen ere. Desberdintasun bakarra, *x* balioak oraingoan +800º/s eta -800º/s izan ordez, +950º/s eta -1000º/s izan direla, hau da, mugimendua gauzatzerako orduan, azken honetan (SPI bus komunikazioaren inplementazioaren proban) azkarrago mugitu Dugula MPU-9250 gailua eta ondorioz *x* balio altuagoak eskuratu dira.

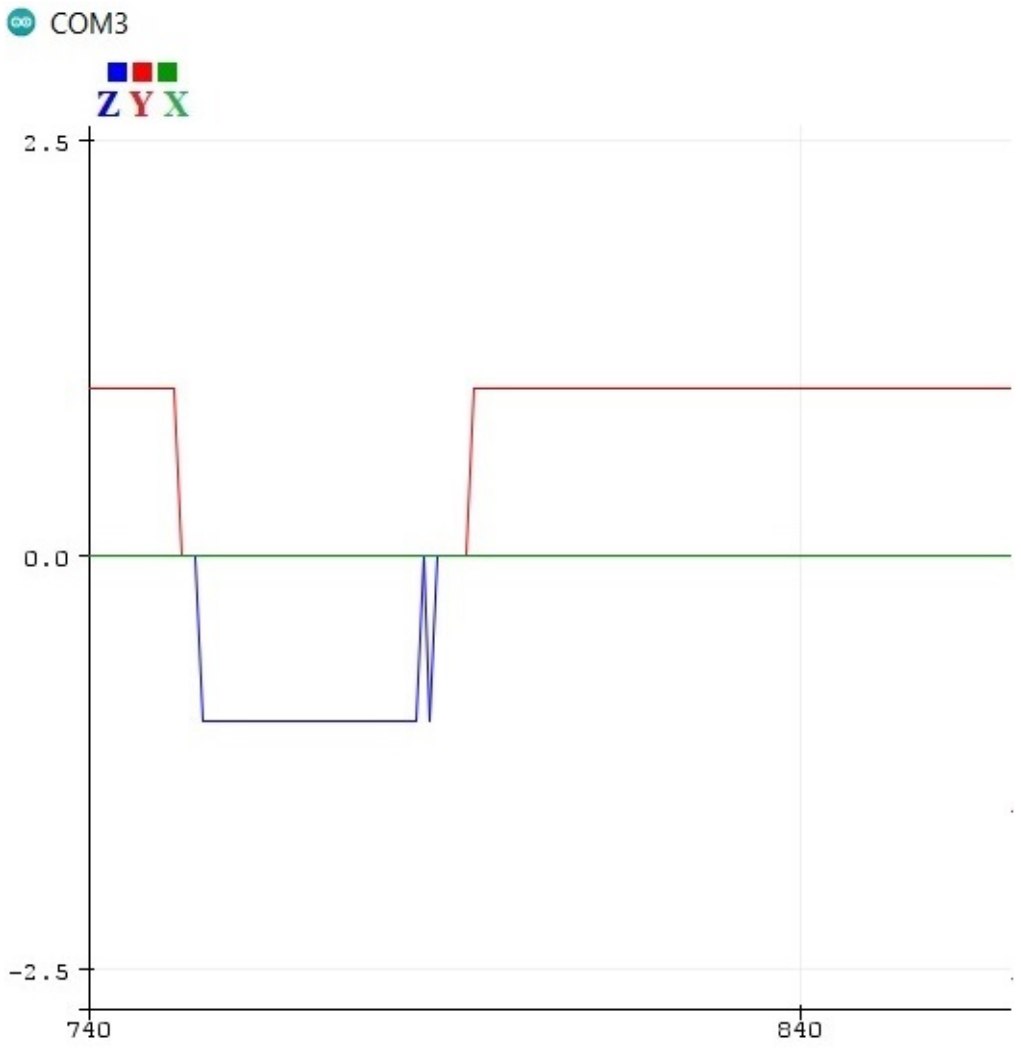
Azelerometroak bildu dituen datuak aztertzen badira, mugimendu osoan zehar (+90º-ko eta -90º-ko biraketak *x* ardatzarekiko) ikus daiteke lehen lortu diren datuen oso antzekoak direla oraingo emaitzak. 4.14 irudian antzematen da bi biraketek eragiten dituzten balio aldaketak azelerometroan.



4.13 Irudia: *x ardatzean +90º eta -90º biraketak sortutako giroskopioaren emaitza*

4.4 Erabilera praktikoaren inplementazioa

Atal honetan proiektuan planteatu den praktika nagusiari (1.2 atalean azaldu dena) ebazpen bat emango zaio MPU-9250 gailua eta Arduino Uno plaka erabiliz.



4.14 Irudia: *x ardatzean +90º eta -90º biraketak sortutako azelerometroaren emaitza*

Helburua: Sistema Txertatuen Diseinua irakasgaian erabiltzen den beso robotiko baten pintza beti posizio horizontalean mantentzea. 1.2 atalean aipatu den bezala, robota proiektu honetan landu ez den programa batekin eta Explorer 16 plakaren bitartez kontrolatzen da. Lan honetan robotaren 4. serboaren mugimenduak MPU-9250 gailuaren bitartez kontrolatzea eta ondorioz pintza beti plano horizontalarekiko paralelo mantentzea bilatu da.

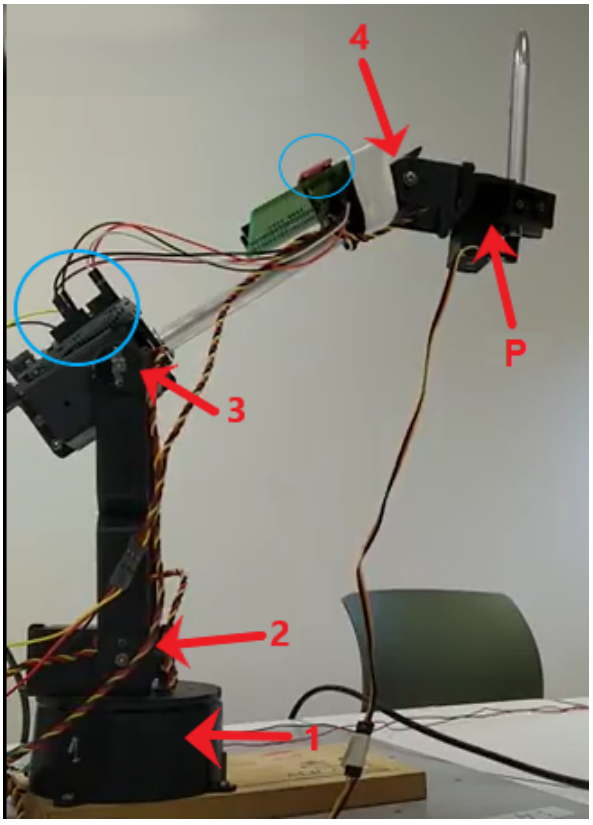
Loturak: Inplementazioa hau egiterako orduan I^2C busa erabili da Arduino Uno eta MPU-9250 gailuaren artean komunikazioa ahalbidetzeko. Gainera, beso robotikoaren 4. serboa kontrolatzeko beharrezkoak diren loturak ere egin behar direnez, hiru hankatxo gehiago lotu behar dira plakara (serboaren hiru hankatxoak).

Beraz, I^2C busarekin zerikusia duten loturak 4.2 atalean jarri diren bezala mantentzen dira

eta gehitzen diren 3 loturak serboarenak dira:

- VCC hankatxoa 5V-eko elikadura iturrira konektatua
- GND hankatxoa Ground erreferentziara konektatua
- SIGNAL hankatxoa 9. konexiora lotuta

Robotaren egitura: Lehen aipatu den bezala, proiektuan erabiliko den robota, beso robotiko bat da. Lau serbok eta muturreko pintzak osatzen dute 4.15 irudian ikusten den bezala. Honen gainean, Arduino Uno plaka (3. serboaren gainean dagoen zirkunferentzia urdinak markatzen duena) eta MPU-9250 gailua (4. serboaren ondoan dagoen zirkunferentzia urdinez markatua) jarri dira.



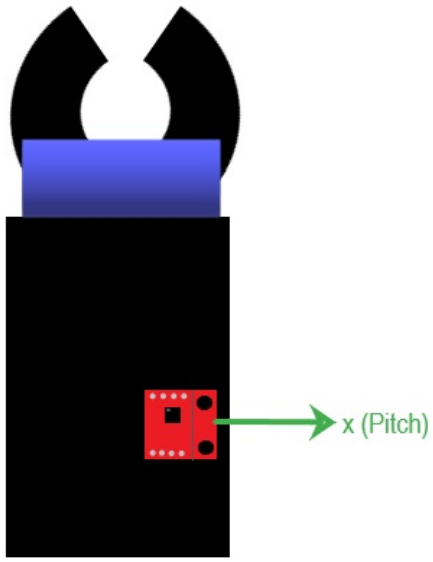
4.15 Irudia: *beso robotikoaren egitura*

Robotak 3. eta 4. serboen artean duen lotura, metalezko barra bat denez, MPU-9250 gailua ezin da zuzenean bertan jarri magnetometroak erabiltzen dituen sentsore magnetikoetan interferentziak sortzen direlako eta proiektuan erabiltzen ez den arren, sentsore osoaren

funtzionamendua kaltetzen du. Horregatik, zulodun plaka bat erabili da sentsorea eta robota ez elkar ukitzeko.

Kodea: Lehenik eta behin, MPU-9250 gailua beso robotikoaren gainean duen orientazioa jakitea oso garrantzitsua da. 4.16 irudian agertzen den bezala kokatzen da sentsorea. Irudia beso robotikoaren azken muturraren goiko bistaren irudikapena da: aurrean pintza, bere azpian beso robotikoarekin lotzen duen laugarren serboa (kolore morez marraztuta) eta azkenik laugarren serbora lotuta dagoen beso robotikoaren muturra, MPU-9250 gailua bertara lotua dagoela.

Beraz x ardatza beso robotikoaren eskuinera dago orientatuta, y ardatza pintzara (aurrera) orientatuta eta z ardatza gora (robotarekiko perpendikularrean).



4.16 Irudia: *MPU-9250 gailuaren orientazioa beso robotikoan*

4.15 irudia aztertzen bada berriro, ikus daiteke 2., 3. eta 4. serboek ardatz berdinarekiko mugimendua gauzatzen dutela. MPU-9250 gailua hartzen badugu erreferentzia puntu bezala, bere y eta z ardatzek marrazten duten planoan bakarrik mugitzen da robota aipatutako serboen konbinazioarekin. Hau da, 2. eta 3. serboen arteko zatia aurrera eta atzera inklinatzen da, eta 3. eta 4. serboen arteko zatiak ere mugimendu berdina egiten du. Lehenengo serboaren mugimenduak ez du eraginik sortzen planteatu den arazoan, sortzen duen rotazioak ez duelako MPU9250 sentsorearen orientazioa aldatzen lurraren plano horizontalarekiko, posizioa bakarrik aldatzen da.

Honek esan nahi du pintza beti posizio horizontalean mantentzeko, MPU-9250 gailuaren x ardatza hartu behar dela kontuan, laugarren serboaren biraketa horri esker alda daitekeelako pintzaren posizioa eta orientazioa y eta z ardatzek marrazten duten planoan.

Behin jakinda MPU-9250 gailuaren Pitch balio dela behar dena erabilera praktikoa gauzatzeko, hau nola lortzen den aztertu behar da.

4.4.1 Pitch balioa giroskopioaren bitartez

3.1.3 atalean aipatu den bezala, abiadura angeluarra neurtzen du giroskopioak, gradu segundoko (º/s) unitatean. Hori horrela izanik, desplazatu den gradu kopurua kalkulatzeko (Pitch balioa eskuratzeko), angelu aldaketa hori zenbat denboran gauzatu den jakin behar da. Behin une jakin bateko abiadura angeluarra jakinda (giroskopioak eskaintzen du datu hau) eta une horrek iraun duen denbora jakinda, bi balioak biderkatzen dira emaitza bezala desplazatu den angeluaren balioa eskuratuz. Ondoren, hurrengo denbora unearekin berdin egiten da eta lortzen den angelua, aurreko uneko angeluari gehitzen zaio. Hau behin eta berriz eginda, uneoro Pitch angeluaren estimazio bat lortzen da.

Prozesu honi (abiadura angeluarrak integratzeko prozesuari angeluak lortzeko) *Dead reckoning* estimazioa deritzo [Choset et al., 2005] eta epe motzera mugimenduen kontrola mantentzeko oso prozesu egokia da. Baina arazo bat du prozesu honek, edozein errore txiki (kanpoko soinuak eraginda, denbora kalkulu ez zehatzak eraginda...) metatzen joango da denboran zehar eta epe luzera ez dira emaitza zuzenak lortuko.

Beste aldetik, neurketa mota hau erlatiboa da. Hau da, momentuan dagoen posizioarekiko angelu aldaketa kalkulatzen du. Beraz, sentsorea neurketak egiten hasi denean hasierako posizioa 0º izan ordez 3º dela suposatzen bada eta angelu aldaketa +5º-koa bada, errealitatean +8º-ko Pitch balioa izan beharko luke baina 5º-ko Pitch balioa lortuko litzateke ezin duelako kontuan hartu hasieran MPU-9250 gailuak 3º-ko inklinazioa zuela.

Hasieran, erabilera praktiko hau giroskopioaren laguntzaz bakarrik ebaztea zen ideia, baina probak egin ondoren, metatzen den errorearen arazoa eta neurketa erlatiboaren arazoa ikusi dira. Horregatik azelerometroa erabiltzea erabaki da.

4.4.2 Pitch balioa azelerometroaren bitartez

3.1.2 atalean ikusi denez, azelerometroak jasaten dituen azelerazio indarrak detektatzen ditu. Grabitateak sortzen duen grabitazio indarra beti noranzko berdina du (lurraren erdigunera seinalatzen du beti), beraz azelerometroarekin beti lor daiteke erreferentzia konstante hori. MPU-9250 gailua inklinatzen bada, grabitazio indarraren erreferentzia ez denez galtzen, hau da, sentsoreak beti dakienez zein noranzkoa den behera, kalkulatu daiteke zenbat gradu biratu den beherako noranzko horrekiko. kalkulu hori (4.1) ekuazioarekin lortzen da, non A_y azelerometrotik lortzen den y ardatzaren balioa den, A_x azelerometrotik lortzen den x ardatzaren balioa eta A_z azelerometrotik lortzen den z ardatzaren balioa [Acc, 2013].

$$\arctan \frac{A_y}{\sqrt{(A_x)^2 + (A_z)^2}} \tag{4.1}$$

Baina giroskopioaren antzera, azelerometroak ere bere "ahuleziak"ditu. Azelerometroak azelerazio edo indar guztiak detektatzen dituenez, edozein mugimendu txiki edo bibrazio neurtzen ditu eta grabitatearen erreferentzia galtzen du momentu batez. Beraz epe luzerako oso fidagarria eta egonkorra da baina epe motzeko neurketak egiteko ez da aukera ona.

4.4.3 Complementary filter

Pitch balioa lortzeko giroskopioaren eta azelerometroaren aukerak ikusi ondoren, eta bakoitzaren arazoak ikusi ondoren, aukera egokiena biak konbinatzea dela: Azelerometroak epe luzera ez du grabitate indarraren erreferentzia galtzen baina epe motzera ez ditu neurketa oso fidagarriak egiten. Giroskopioak aldiz ez du erreferentzia globalik eta errore bat metatzen joaten da baina epe motzean neurketa oso zehatzak egiten ditu.

Ikusten denez azelerometroaren eta giroskopioaren alderdi onak osagarriak dira. Horregatik *Complementary filter* [Van de Maele, 2013] bat erabiltzea erabaki da, azelerometroa eta giroskopioa fusionatzeko. Bi aukeren alderdi onak aprobetxatuz, Pitch balioren neurketa zehatz eta on bat egiteko.

Complementary filter batek betetzen duen funtzioa neurketa baten zati bat hartu eta beste neurketa baten zati osagarriarekin fusionatzea da. Kasu honetan, epe motzerako giroskopioaren neurketak hartu nahi dira eta epe luzerako azelerometroaren neurketekin osatu.

Horretarako elementu bakoitzetik lortu den balioa α koefiziente batekin biderkatu behar da.

$$Pitch = GiroPitch * \alpha + AzelPitch * (1 - \alpha) \tag{4.2}$$

Koefiziente horrek adierazten du ehuneko zenbateko garrantzia ematen zaion elementu horretatik jaso den balioari, beste elementuaren koefizientea bere osagarria izanik. Beraz garrantzitsuena koefiziente egokia aurkitzea da α balioarentzat.

Koefiziente egoki hori aurkitzerako orduan, bi aukera orokor daude: α-ri 0tik gertu dagoen balio bat ematea giroskopioari garrantzi gutxi emateko eta azelerometroaren ezaugarriak indartzeko, edo α-ri 1etik gertu dagoen balio bat ematea giroskopioari emateko garrantzi haundia eta bere ezaugarriak indartzeko.

Lehenengo aukeraren kasuan, azelerometroaren ezaugarriak nabarmentzen badira, epe luzera Pitch balioa egonkorra izango da baina beste indar guztien eraginak ere asko nabarmenduko dira eta Pitch balioa ez da egonkorra izango epe motzera. Honek Pitch balio oso aldakor bat emango digu pintza une oro horizontalean mantentzea ezinezkoa bihurtuz.

Bigarren kasuan, giroskopioaren ezaugarriak nabarmentzen badira, epe motzera Pitch balio oso zehatz bat lortuko da eguneraketa azkarrak eginez eta lehen gertatzen ez zen bezala, oraingoan azelerometroaren erreferentzia egongo da (nahiz eta koefiziente baxua izan) horren ondorioz metatzen doan errorea pixkanaka zuzentzen joanez. Bigarren aukera honek aldaketa leunagoak eragiten ditu eta ondorioz Pitch balioa une oro horizontalean mantentzea errazagoa da.

Proba batzuk egin ondoren eta beste artikulu batzuetan erabili diren balioak aztertuz [Ariffin et al., 2016] [Van de Maele, 2013], α koefizienteari 0.98 balioa ematea aukeratu da. Beraz giroskopioari %98-ko garrantzia eman zaio eta, koefizienteak osagarriak direnez, azelerometroari %2-ko garrantzia. Ikusten denez azelerometroaren ekarpenak garrantzi txikia du baina beharrezkoa da posizio orokorraren erreferentzia ez galtzeko.

Complementary filter-arekin batera beste baliabide desberdin batzuk erabili dira kodea osatzeko:

- "Wire.h"eta "Servo.h"liburutegiak erabili dira I^2C bus bidezko komunikazioa eta serboaren kontrola ahalbidetzeko.
- Giroskopiotik lortzen diren abiadura angeluarrak integratu eta angeluen estimazioa lortzeko, angelu aldaketa hori zenbat denboran gauzatu den jakin behar da 4.4.1

atalean ikusi den bezala. Horretarako *millis* funtzioa erabili da momentuko denbora kalkulatu eta hurrengo iterazioan berriro erabiliz momentuko denbora eguneratzen joateko.

- Behin Pitch balioa lortu dela, serboak balio hori jaso behar du eta balio hori eta 0º balioaren arteko diferentzia mugitu behar da beti posizio horizontalean (Pitch = 0º) mantentzeko. Horretarako *map* funtzioa erabili da.

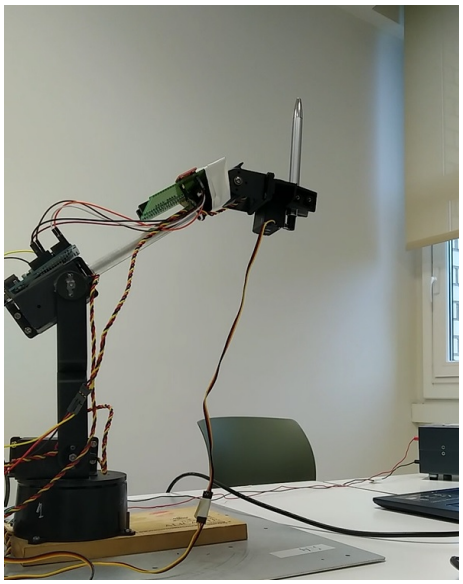
Amaitzeko, Pitch balioaren errore kalkulu bat gauzatzen duen funtzioa garatu da. Teorikoki, sentsorea egoera egonkor batean geldi badago, giroskopioak 0º/s-ko abiadura angeluarra detektatu beharko luke. Baina praktikan baliteke giroskopioak mugimendu txikiak detektatzea sentsorea geldirik egon arren. Horregatik inplementatu da funtzio hau, mugimendu txiki eta baztergarri horiek detektatzeko. Funtzionamendua hurrengoa da: MPU-9250 gailua egoera egonkor batean dagoela, 300 neurketa egiten dira giroskopioaren x ardatzean (x ardatzak ematen digu Pitch balioari buruzko informazioa) eta balioak aldagai batean metatzen joaten dira. Behin 300 neurketak eginda, balio metatuak dituen aldagaia zatitu egiten da 300ekin. Errorerik egongo ez balitz, balio metatuen batura 0 izango litzateke eta $\frac{0}{300} = 0$ denez, errorea 0 izango litzateke. Baina mugimendu txikiak antzematen baditu, emaitza lortutako errorea izango da.

Gero, errore hori giroskopioaren x ardatzeko irakurketa bakoitzean lortzen den balioari kentzen zaio errorea konpentsatzeko.

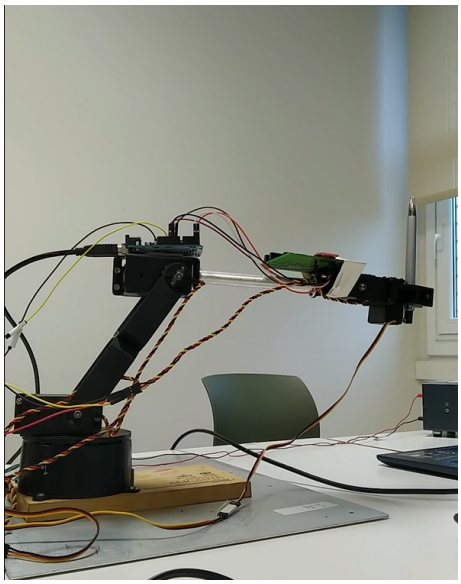
4.4.4 Lortutako emaitzak

Behin kodea amaituta, beso robotikoan muntatu da beharrezkoa zen hardwarea, beso robotikoa kontrolatzen duen kodea (Sistema Txertatuen Diseinua irakasgaian programatua izan den kodea, proiektu honetan programatu ez dena) martxan jarri da Explorer 16 plaka batean, eta mugimendu desberdinak gauzatu dira lan honetan proposatu den programak eman dituen emaitzak aztertzeko.

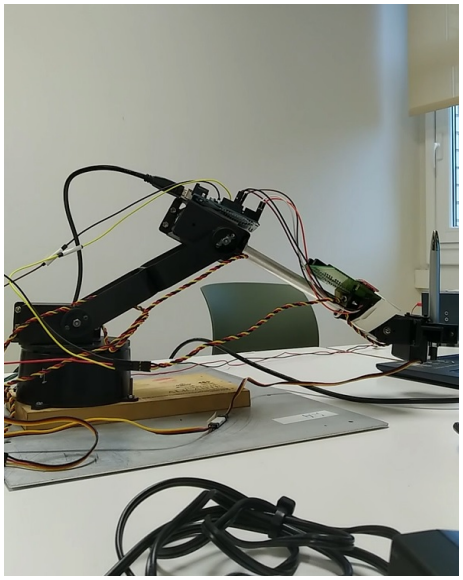
Lortu diren emaitzak espero zirenak izan dira. Besoaren muturreko pintza beti posizio horizontalean mantendu da (boligrafo bat erabili da erreferentzia bisual modura) robotak egiten dituen mugimenduak edozein izanda ere 4.17 irudian ikusten den moduan. Beraz, proposatu den erabilera praktikoa ondo ebaztea lortu dela esan daiteke.



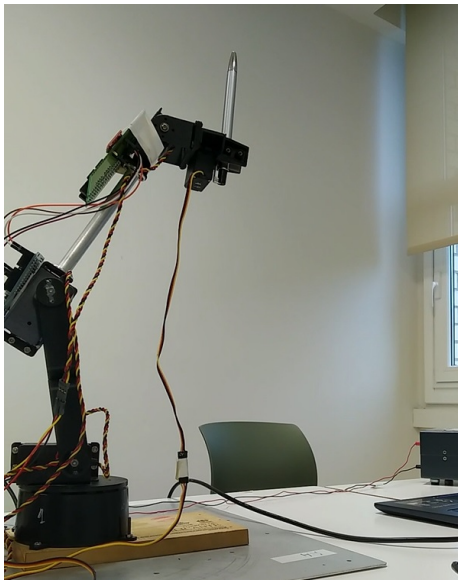
(a) Lehenengo egoera



(b) Bigarren egoera



(c) Hirugarren egoera



(d) Laugarren egoera

4.17 Irudia: Egindako probaren 4 egoera

4.4.5 Erabilera praktikoaren kodea

```
#include "Wire.h"
#include <Servo.h>

#define    MPU_HELB   0x68
#define    serbopin   9

#define    GYRO_250_DPS    0x00
#define    GYRO_500_DPS    0x08
#define    GYRO_1000_DPS   0x10
#define    GYRO_2000_DPS   0x18

#define    ACC_2G         0x00
#define    ACC_4G         0x08
#define    ACC_8G         0x10
#define    ACC_16G        0x18

Servo nireserbo;

//aldagaiak sortu datuak eskuratzeko eta tratatzeko
float accelerometer_x, accelerometer_y, accelerometer_z;
float gir_x, gir_y, gir_z;
float aurregir_x=0;
float gir_x_error = 0;
float aurrekoDenbora, momentukoDenbora, deltaDenbora,
float pitch_gir, pitch_az, pitch_total;
float momentukoErrore = 0.0;
int konta = 0;
```

```
 /*
  * I2Cread (Irakurketa funtzio lagungarria)
  *
  * Input:  Helbidea: Irakurketa zein sentsoretik egingo den
  *         Erregistroa: Aukeratutako elementuko zein erregistrotik irakurri
  *         Byte kopurua: Irakurriko den byte kopurua
  *         Data: Irakurritako datuak jasoko diren memoria tartea
  */
void I2Cread(uint8_t Helb, uint8_t Erreg, uint8_t Nbytes, uint8_t* Data){
   Wire.beginTransmission(Helb);
   Wire.write(Erreg);
   Wire.endTransmission();

   Wire.requestFrom(Helb, Nbytes);
   uint8_t index = 0;
   while (Wire.available())
      Data[index++] = Wire.read();
}
 /*
  * I2CwriteByte (Idazketa funtzio lagungarria)
  *
  * Input:  Helbidea: Idazketa egin nahi den elementuaren helbidea
  *         Erregistroa: Aukeratutako elementuko zein erregistrotan idatziko den
  *         Data: Idatzi nahi dugun informazioa
  */
void I2CwriteByte(uint8_t Helb, uint8_t Erreg, uint8_t Data){
   Wire.beginTransmission(Helb);
   Wire.write(Erreg);
   Wire.write(Data);
   Wire.endTransmission();
}
```

```
float erroreaKalkulatu()
{
  while(konta<300){
    Wire.beginTransmission(MPU_HELB);
    Wire.write(0x43);
    Wire.endTransmission(false);
    Wire.requestFrom(MPU_HELB,2);
    gir_x=Wire.read() << 8 | Wire.read();
    gir_x_error = gir_x_error + (gir_x /131);
    konta++;
  }
  gir_x_error = gir_x_error/300;
  Serial.print("Pitch errorea: ");
  Serial.println(gir_x_error);
  return gir_x_error;
}

void setup() {
  nireserbo.attach(serbopin);
  Serial.begin(9600);
  Wire.begin();
  //Giroskopioaren konfigurazioa 27. erregistroan definitu behar da
  I2CwriteByte(MPU_HELB, 27, GYRO_1000_DPS);
  //Azelerometroaren konfigurazioa 28. erregistroan definitu behar da
  I2CwriteByte(MPU_HELB, 28, ACC_8G);
  //Errorea kalkulatu
  float momentukoErrore=erroreaKalkulatu();
}
```

```
void loop() {
  aurrekoDenbora = momentukoDenbora;
  momentukoDenbora = millis();
  deltaDenbora = (momentukoDenbora - aurrekoDenbora)/1000;
  // zati 1000 denbora segundutan lortzeko
   uint8_t Buf[14];
   I2Cread(MPU_HELB, 0x3B, 14, Buf);
  accelerometer_x = (Buf[0] << 8 | Buf[1])/4096;
  accelerometer_y = (Buf[2] << 8 | Buf[3])/4096;
  accelerometer_z = (Buf[4] << 8 | Buf[5])/4096;
  gir_x = (Buf[8] << 8 | Buf[9])/ 32,8;
  gir_y = (Buf[10] << 8 | Buf[11])/ 32,8;
  gir_z = (Buf[12] << 8 | Buf[13])/ 32,8;

  //zuzendu errorea, kalkulatutako balioak erabiliz
  gir_x = gir_x - momentukoErrore ;

  //giroskopiotik lortutako pitch balioa
  pitch_gir= pitch_total + gir_x * deltaDenbora;

  //azelerometrotik lortutako pitch balioa
  pitch_az = 180 * atan2(accelerometer_y, sqrt(accelerometer_x*accelerometer_x
        + accelerometer_z*accelerometer_z))/PI;

  //Complementary filtroa erabiliz kalkulatuko dugu pitch balio egokia
  //(koefizienteak modifikatuz balio egokia aurkituz)
  pitch_total= pitch_gir * 0.98 + pitch_az * 0.02;
  //Serbora mapeatu pitch balioa
  int serbobalioa = map(pitch_total, 90, -90, 0, 180);
  nireserbo.write(serbobalioa);
}
```

4.4.6 I²C bus inplementazioaren kodea

```
#include "Wire.h
#define    MPU_HELB                  0x68

#define    GYRO_250_DPS    0x00
#define    GYRO_500_DPS    0x08
#define    GYRO_1000_DPS   0x10
#define    GYRO_2000_DPS   0x18

#define    ACC_2G        0x00
#define    ACC_4G        0x08
#define    ACC_8G        0x10
#define    ACC_16G       0x18

int16_t acc_x, acc_y, acc_z;
int16_t giro_x, giro_y, giro_z;

void I2Cread(uint8_t Helb, uint8_t Erreg, uint8_t Nbytes, uint8_t* Data)
{
   Wire.beginTransmission(Helb);
   Wire.write(Erreg);
   Wire.endTransmission();
   Wire.requestFrom(Helb, Nbytes);
   uint8_t index = 0;
   while (Wire.available())
      Data[index++] = Wire.read();
}

void I2CwriteByte(uint8_t Helb, uint8_t Erreg, uint8_t Data)
{
   Wire.beginTransmission(Helb);
   Wire.write(Erreg);
   Wire.write(Data);
   Wire.endTransmission();
}
```

```
void setup() {
  Serial.begin(9600);
  Wire.begin();
   // Azelerometroa konfiguratu
   I2CwriteByte(MPU_HELB, 28, ACC_8G);
   // Giroskopioa konfiguratu
   I2CwriteByte(MPU_HELB, 27, GYRO_1000_DPS);
}

void loop() {
   uint8_t Buf[14];
   //Balioak irakurri
   I2Cread(MPU_HELB, 0x3B, 14, Buf);

  acc_x = (Buf[0] << 8 | Buf[1])/4096;
  acc_y = (Buf[2] << 8 | Buf[3])/4096;
  acc_z = (Buf[4] << 8 | Buf[5])/4096;
  giro_x = (Buf[8] << 8 | Buf[9])/16,4;
  giro_y = (Buf[10] << 8 | Buf[11])/16,4;
  giro_z = (Buf[12] << 8 | Buf[13])/16,4;

  // Informazioa pantailaratu
  Serial.print(acc_z);Serial.print(",");
  Serial.print(acc_y);Serial.print(",");
  Serial.print(acc_x);Serial.print(",");
  Serial.print(giro_z);Serial.print(",");
  Serial.print(giro_y); Serial.print(",");
  Serial.print(giro_x);
  Serial.println();
  delay(20);
}
```

4.4.7 SPI bus inplementazioaren kodea

```
#include <SPI.h>
const int CS_PIN = 10;
#define IRAKURKETA_FLAG 0x80
#define SPI_DATA_RATE 1000000 // 1 MHz max
#define SPI_MODE SPI_MODE3

#define    GYRO_250_DPS    0x00
#define    GYRO_500_DPS    0x08
#define    GYRO_1000_DPS   0x10
#define    GYRO_2000_DPS   0x18

#define    ACC_2G         0x00
#define    ACC_4G         0x08
#define    ACC_8G         0x10
#define    ACC_16G        0x18

//16 biteko integer aldagaiak sortu ardatzetako datuak gordetzeko
int16_t  acc_x, acc_y, acc_z, gir_x, gir_y, gir_z;


void setup() {
  Serial.begin(9600);
  pinMode(CS_PIN, OUTPUT);
  digitalWrite(CS_PIN, HIGH);
  SPI.begin();
  writeByteSPI(28, ACC_8G);
  writeByteSPI(27, GYRO_1000_DPS);
}
```

```
void loop() {
  // put your main code here, to run repeatedly:
  uint8_t Buf[14];
  readBytesSPI(0x3B, 14, Buf);
    acc_x= (Buf[0] << 8 | Buf[1])/4096;
    acc_y= (Buf[2] << 8 | Buf[3])/4095;
    acc_z= (Buf[4] << 8 | Buf[5])/4096;
    gir_x= (Buf[8] << 8 | Buf[9])/16,4;
    gir_y= (Buf[10] << 8 | Buf[11])/16,4;
    gir_z= (Buf[12] << 8 | Buf[13])/16,4;

    //Informazioa pantailaratu
    Serial.print(acc_x);Serial.print(",");
    Serial.print(acc_y);Serial.print(",");
    Serial.print(acc_z);Serial.print(",");
    Serial.print(gir_x);Serial.print(",");
    Serial.print(gir_y); Serial.print(",");
    Serial.print(gir_z);
    Serial.println();
  delay(20);
}

void writeByteSPI(uint8_t erreg, uint8_t data)
{
  SPI.beginTransaction(SPISettings(SPI_DATA_RATE, MSBFIRST, SPI_MODE));
  digitalWrite(CS_PIN, LOW);
  SPI.transfer(erreg);
  SPI.transfer(data);
  digitalWrite(CS_PIN, HIGH);
  SPI.endTransaction();
}
```

```
void readBytesSPI(uint8_t erreg, uint8_t kont, uint8_t* buf)
{
  SPI.beginTransaction(SPISettings(SPI_DATA_RATE, MSBFIRST, SPI_MODE));
  digitalWrite(CS_PIN, LOW);
  SPI.transfer(erreg | IRAKURKETA_FLAG);
  uint8_t i;
  for(i=0; i < kont; i++){
    buf[i]=SPI.transfer(0x00);
  }
  digitalWrite(CS_PIN, HIGH);
  SPI.endTransaction();
}
```

5. KAPITULUA

Ondorioak

Proiektu honek MPU-9250 gailuaren funtzionamendua ikertu eta probatu ditu gailuak berak eskaintzen dituen eta beharrezkoak izan diren beste baliabide batzuk erabiliz. Komunikazioa ahalbidetzeko beharrezkoak izan diren I^2C busaren eta SPI busaren funtzionamendua ere ikertu eta probatu da.

MPU-9250 gailuaren inguruan aztertu da bere potentziala handia dela (nabigazioa, errealitate birtuala, robotika... esparruetan oso erabilia) eta sentsore konplexua izan arren oso erabilgarria dela. Gainera, ikusi da lan honetan erabili ez diren funtzionalitate desberdinak bete ditzakeela gailuak.

Proiektuan, arazo desberdinak aurkitu dira garapen fasea betetzen zen bitartean, bai hardware aldetik (SJ2 jumperraren arazoa) eta bai planteamendu aldetik (erabilera praktikoan azelerometroaren beharra). Dena den, eragozpen hauei irtenbide bat ematea lortu da estimatutako denbora gehiegi luzatu gabe eta lana aurrera eramatea posible eginez.

Bestetik, erabilera praktikoari ebazpen bat eman zaio, ikerketaren bidez lortu den jakinduria aplikatuz. Aurkeztu diren bi bus mota desberdinen bidezko komunikazioa lortu da (I^2C busa eta SPI busa), proposatutako ebazpena beso robotikoan probatu da eta espero ziren emaitzak lortu direnez, lan honetan planteatzen ziren helburuak bete direla esan daiteke, proiektua amaitutzat emanez.

5.1 Etorkizunerako lana

Proiektu honetan landutakoa sistema txertatuen diseinua irakasgaian erabiltzen den Explorer 16 txartelera pasatzea izango litzateke lehentasun handieneko lana, proiektu hau zubi modura erabiliz MPU-9250 gailua irakasgaian sartuz.

Gainera, Explorer 16 txartelak duen PIC 24H mikroprozesadorearekin, MPU-9250 gailuak dituen eta lan honetan erabili ez diren beste baliabide batzuk ikertzeko eta probatzeko aukera sortuko litzateke, gailuaren funtzionamendua sakonago aztertzeko.

Aurreko idearekin jarraituz magnetometroaren erabilera inplementatzea ideia ona litzateke. Lan honetan 2 sentsore erabili direnez guztira erabili diren 6 ardatzetako datuak aztertzen dira (3 ardatz azelerometrotik eta beste 3 giroskopiotik). Baina magnetometroa ere erabiliko balitz, 9 ardatzetatik lortuko lirateke datuak (3 ardatz magnetometrotik, 3 ardatz azelerometrotik eta beste 3 giroskopiotik). Gainera, magnetometroaren erabilerarekin iparraren erreferentzia konstantea lortuko litzateke (azelerometroaren grabitate indarraren erreferentziarekin gertatzen den bezala) teorikoki egonkortasun gehiago emanez.

Magnetometroaren erabilerarekin *Complementary filter* ez den beste filtro mota bat erabiltzea ere aukera ona izan daiteke, azelerometrotik eta giroskopiotik eskuratzen diren datuak magnetometrotik eskuratzen diren datuekin batzeko eta emaitza osatuago bat lortzeko.

Bibliografia

[Acc, 2013] Acc (2013). *Tilt Sensing Using a Three-Axis Accelerometer*. NXP Semiconductors by Mark Pedley. Rev. 6.

[Ahmad et al., 2013] Ahmad, N., Ghazilla, R. A. R., Khairi, N. M., and Kasi, V. (2013). Reviews on various inertial measurement unit (imu) sensor applications. *International Journal of Signal Processing Systems*, 1(2):256–262.

[Ard, 2019] Ard (2019). Arduino uno documentation, https://docs.arduino.cc/hardware/uno-rev3.

[Ariffin et al., 2016] Ariffin, N. H., Arsad, N., and Bais, B. (2016). Low cost mems gyroscope and accelerometer implementation without kalman filter for angle estimation. pages 77–82.

[AZL, 2020] AZL (2020). ¿cómo funciona y qué hace el acelerómetro?, https://www.tme.eu/es/news/library-articles/page/22568/como-funciona-y-que-hace-el-acelerometro/.

[Bai and Bai, 2019] Bai, Y. and Bai, Q. (2019). 4 - subsea surveying, positioning, and foundation. In Bai, Y. and Bai, Q., editors, *Subsea Engineering Handbook (Second Edition)*, pages 81–121. Gulf Professional Publishing, Boston, second edition edition.

[Choset et al., 2005] Choset, H. M., Lynch, K. M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., Thrun, S., and Arkin, R. C. (2005). *Principles of robot motion: theory, algorithms, and implementation*. MIT press.

[Desai et al., 2014] Desai, P. R., Desai, P.Ñ., Ajmera, K. D., and Mehta, K. (2014). A review paper on oculus rift-a virtual reality headset.

[Hazry et al., 2009] Hazry, D., Mohd Sofian, M. R., and Zul Azfar, A. (2009). Study of inertial measurement unit sensor.

[I2C, 2014] I2C (2014). *I2C-bus specification and user manual*. NXP Semiconductors. Rev. 6.

[Iannacci, 2017] Iannacci, J. (2017). Introduction to mems and rf-mems: From the early days of microsystems to modern rf-mems passives. In *RF-MEMS Technology for High-Performance Passives*, 2053-2563, pages 1–1 to 1–39. IOP Publishing.

[IPS, 2014] IPS (2014). *MPU-9250 Product Specification*. InvenSense. Rev. 1.

[Jantunen et al., 2016] Jantunen, T., Mesch, J., Puupponen, A., and Laaksonen, J. (2016). In *On the rhythm of head movements in Finnish and Swedish Sign Language sentences*, pages 850–853.

[Kalinsky and Kalinsky, 2002] Kalinsky, D. and Kalinsky, R. (2002). Introduction to serial peripheral interface.

[MAM, 2008] MAM, M. (2008). *4. Sensor medidor de Aceleración. Acelerómetro*, page 39–54. Universidad de Sevilla.

[Mot, 2000] Mot (2000). *SPI Block Guide*. Motorola. Rev. 3.

[Ramsden, 2011] Ramsden, E. (2011). *Hall-effect sensors: theory and application*. Elsevier.

[Ruiza et al., 2004] Ruiza, M., Fernandez, T., and Tamaro, E. (2004). Las leyes de isaac newton.

[Van de Maele, 2013] Van de Maele, P.-J. (2013). Reading a imu without kalman: The complementary filter.

[You and Neumann, 2001] You, S. and Neumann, U. (2001). Fusion of vision and gyro tracking for robust augmented reality registration. In *Proceedings IEEE Virtual Reality 2001*, pages 71–78.

[You, 2018] You, Z. (2018). Chapter 9 - magnetometer technology. In *Space Microsystems and Micro/nano Satellites*, Micro and Nano Technologies, pages 341–360. Butterworth-Heinemann.