

Debreceni Egyetem Informatikai Kar
Adattudomány és Vizualizáció Tanszék

Digitális aktuátor szenzorrendszer fejlesztése önvezető modell autókhoz

Témavezető:

Dr. Kovács László

egyetemi docens

Készítette

Mári Ferenc

Mérnökinformatikus

BSC

Girászi Tamás

Programtervező

informatikus

MSC

Zolnai Csaba

Mérnökinformatikus

BSC

Debrecen, 2022

Tartalomjegyzék:

Kivonat	4
1. Bevezetés.....	5
2. Irodalmi áttekintés.....	7
2.1 „Dave” a modellautó	7
2.1.1 Miért kell a kerekek forgási sebességét mérni:.....	8
2.1.2 Miért kell kerékelfordulást mérni:	8
2.1.3 A Hall szenzorról	9
2.2. Kerék elfordulás	11
2.2.1 Alap Koncepció	11
2.2.2. Analóg-digitális konverter (ADC)	13
2.2.3. 3D Tervezés	15
2.2.4. Mágnes pozíciója	16
2.2.5. Álló megoldás	18
2.2.6 A Fekvő megoldás	20
2.2.7. Teleszkóp alapú kerékösszekötő tengely tervezése	22
2.2.8. Mágnes csere.....	24
2.2.9. Kábelproblémák	25
2.2.10. Szoftveres háttér.....	28
2.2.11. Eredmények	32
2.3 Sebességmérés.....	34
2.3.1 A sebesség.....	35
2.3.1 Fizikai Alapok.....	39

2.3.2 Mozgásformák	39
2.3.4 A sebesség felhasználása	40
3. Összefoglalás.....	48
4. Saját munka leírása.....	51
4.2 Sebességmérő szenzor:	51
Köszönetnyilvánítás	53
Hivatkozások.....	54

Kivonat

Napjainkban a vezetéstámogatási rendszerek fejlesztésének talán a legfőbb ágai a szenzor fejlesztés és a szenzor integráció. A szenzorok két csoportba sorolhatóak, perceptuálisak és aktuátor visszajelző szenzorok. Az önvezető autók tervezése és legyártása költséges, és időigényes folyamat, ezért minden lehetőséget, ami időt és pénzt spórolhat, érdemes kipróbálni. A szenzorok tesztelése és az adatgyűjtés egy roppant kritikus pontja a fejlesztésnek. Ha nem képes az akadályokat megfelelően észlelni az autónk, az akár a teljes rendszert és annak környezetét is veszélyeztetheti. Ezen folyamatok költséghatékonyabbá, biztonságosabbá és egyszerűbbé tétele miatt egyre többen döntenek úgy, hogy kicsinyített, modell autókon tesztelik le új ötleteiket, technológiáikat. Nem mellesleg a modell méretben a hallgatóknak lehetőségük van kézzel foghatóbb formában megismerkedni és elsajátítani az 1:1 arányú autóknál is használt technológiákat.

A labor kutatásunk során egy digitális aktuátor szenzorrendszer fejlesztésén dolgozunk. Jelen munkánkban egy egyedi kerék szögelfordulás, valamint egy kerekenként sebességet mérő szenzort terveztünk és ennek a prototípusain dolgozunk. Az elkészült érzékelők alkatrészeit 3D nyomtatóval készítettük el. Ezeket a modellautóra lehet rögzíteni és adatokat lehet vele gyűjteni. A kerékelfordulás mérését egy automata kerékirány kalibráló programhoz, valamint a pontosabb mérések és kormányzás érdekében készítjük. A hely szűkössége miatt a kerék összekötő rudat is újra kellett terveznünk. A kész szenzorhálózatot és a programot egy valós méretű önvezető autó is hasznosíthatja.

A sebességmérő szenzor az egyik leglényegesebb eszköz az autóiiparban. Ennek segítségével meg tudjuk állapítani az autó haladási sebességet. Ezeket a szenzorokat kerekenként szereltünk fel. Az implementáció jellegéből adódóan és a differenciálmű működési elvét alapul véve számos egyéb mérőszám is megállapítható, valamint képesek vagyunk mérni a többi szenzor pontosságát is.

1. Bevezetés

Manapság a különböző vezetéstámogatási rendszerek használata az autókban a mindennapi életünk részévé vált. Nagyon meg tudja könnyíteni a napi ingázást vagy akár a hosszabb utakon előforduló kellemetlenségeket, amit az utazás jelent számunkra. Mindezek mellett ki ne szeretné saját magát és a családját biztonságban tudni. Ezen rendszerek biztonság tekintetében rohamos léptékű fejlődésnek indultak az elmúlt időszakban. A mai modern világban már bárki dönthet úgy, hogy besétál egy autókereskedésbe és ott a piacon kapható legkomfortosabb és legbiztonságosabb vezetéstámogatási eszközökkel felszerelt autót vásárol magának. Mindezt a kényelmet és biztonságot a szenzorok egyre csökkenő ára, viszont növekvő komplexitásuk teszi lehetővé számunkra. Ezek azonban még mindig nem kellően pontos és megbízható eszközök. Ebből kifolyólag a vezetéstámogatási rendszerek fejlesztésének talán a legfőbb ágai a szenzor fejlesztés és a szenzor integráció. Annak ellenére, hogy a szenzorok ára egyre alacsonyabb, egy önvezető autó tervezése és legyártása nagyon költséges, időigényes és meglehetősen bonyolult folyamat. Gondoljunk csak bele, hogy mennyire nagy anyagi és szellemi ráfordítást kíván egy ilyen prototípus megálmodása és megépítése. Ezért minden lehetőséget, ami költséget vagy időt spórolhat mindenképp érdemes kipróbálni. A szenzorok és szenzorrendszerek tesztelése valamint az adatgyűjtés is egy roppant kritikus pontja a fejlesztésnek, mert ha bármilyen akadályt, közlekedési táblát vagy útburkolati jelet nem tud megfelelően észlelni az autónk, az akár a teljes rendszert és annak környezetét is veszélyeztetheti. Egy félre sikerült teszt akár az egész projektet is veszélybe sodorhatja. Emiatt mind anyagi, mind személyi biztonsági tekintetében fontos, hogy a tesztelést ne valós méretekben és ne valós környezetben történjen. A folyamatok költséghatékonyabbá, biztonságosabbá és egyszerűbbé tétele miatt egyre több vállalat és egyetemi kutató csoport dönt úgy, hogy kicsinyített, modell méretben és kontrolált környezetben teszteli le új ötleteiket, technológiákat. Ez a megoldás anyagilag sem terheli meg a kisebb költségvetéssel dolgozó egyetemeket sem. Mindazon által a gyűjtött adatokat és technológiákat a továbbiakban nagyon egyszerűen, kisebb változtatásokkal átültethetik egy valódi autóra. Nem mellesleg a modell méretben a hallgatóknak lehetőségük van kézzel foghatóbb formában megismerkedni és elsajátítani az 1:1 arányú autóknál is használt technológiákat. Később ezáltal értékes szakemberré válhatnak ebben a rohamosan növekvő és fejlődő iparágban. Kutatásunk során mi is egy ilyen kicsinyített modellautón teszteljük az ötleteinket. Célunk, hogy egy digitális aktuátor szenzorrendszert hozzunk létre.

Munkánk által egy egyedi kerék szögelfordulás, valamint egy kerekenként sebességet mérő szenzort terveztünk és ennek a prototípusain dolgozunk. Kerékelfordulás mérő szenzorunkkal egyszerűen detektálható, hogy a kerék hány fokos szögelfordulást végez éppen. Számításaink szerint egy kb~ 4-5°-os szögelfordulás detektálása már kellően pontosnak számít a mi olvasatunkban.

Emellett a kerekenként sebességet mérő szenzor a kerekek forgási sebességét tudja mérni, külön-külön. A sebességértékekkel különböző mérőszámokat tudunk megállapítani, amelyek az autót jellemzik. A szögelfordulással kombinálva egy olyan sokoldalú rendszert kapunk, amely számos baleseti tényező korrigálását és a veszélyhelyzetek könnyebb kezelését teszi lehetővé. Mivel modell autókon nincs lehetőség arra, hogy ellenőrizzük a kiadott utasítás ténylegesen megtörtént-e, ezért különösen fontos, hogy ezeket a szenzorokat megfelelő pontossággal készítsük el. Számos hasznos információt tudunk velük ezáltal kinyerni. Munkánk két fő részre bontható. Az első ezek közül, amelynek a kifejtésével szeretnénk kezdeni, a kerék elfordulásának szögét detektáló rendszer. A későbbiekben pedig a kerekek forgási sebességének mérésére szolgáló rendszerünket szeretnénk részletezni.

2. Irodalmi áttekintés

2.1 „Dave” a modellautó



1. ábra DAVE [1]

Dave vagyis „University of **D**ebrecen **A**utonomous **V**ehicl**E**”, (lásd 1. ábra), az Egyetem aktuális 1:16 méretarányú önvezető modell autója egy NVIDIA Jetson Nano-val, amely 128 magos Maxwell GPU, Quad-core ARM A57 1.43 GHz processzort, valamint 4 GB memóriát tartalmaz. Ahogyan már a bevezetőben is említettük a fejlődéshez szükséges kutatásokat sokkal gyorsabban és biztonságosabban lehet elvégezni ekkora méretarányban, nem is beszélve a költséghatékonyságról. Gondoljunk csak bele mennyivel drágább lenne egy valós méretű tesztautó parkot fenntartani. Nem beszélve a tesztpályák karbantartásáról és megfelelő állapotban tartásáról. Ideális esetben ezeket fedett helyen kellene kialakítani, ugyanis fontos, hogy a tesztkörnyezet kontrollált legyen. Mielőtt veszélyes helyzetekben vagy rossz látási viszonyok közt, netalántán síkos vagy csúszós útfelületeken tesztelnénk és tanítanánk az autó tanuló modelljét, először tökéletes körülmények között kell az ideális utat, útfelületet és felfestéseket, táblákat megtanítani neki. Csak ezután lehetne az ideálistól eltérő veszélyes esetekre felkészíteni. Szensorok tekintetében a nagyobb autóktól semmivel sem kapnánk több adatot a kutatásunk elősegítésére mint a kisebb autóktól. Így e helyett modellautókat használva több kisebb csoportokra osztva a fejlesztőgárdát, mindenki külön-külön dolgozhat a saját modell autóján amíg nem tökéletesítik szenzorjaikat és/vagy szoftveres megoldásaikat. Legyen az akár, a mi „Digital Steering” [2] csapatunk vagy akár a neurális hálót csiszoló tanulók csoportja. Mindenki kényelmesen, a saját tempójában dolgozhat a projektjén. Később az elkészült szenzorokat és egyéb megoldásokat felszereljük egy közös teszt modellautóra és

kipróbáljuk, hogy hogyan működnek egyszerre együtt, egy közös autón. Ezzel a megközelítéssel nagyon gyors tempóban lehetünk képesek a fejlődésre és nem kizárt, hogy néhány éven belül az egyetem önvezető autó gárdája már akár a nemzetközi top csapatok közé is emelkedhet önvezető autó fejlesztés terén.

2.1.1 Miért kell a kerekek forgási sebességét mérni:

Az autók alapvető kialakításából adódóan a sebesség ismeretében nem csak a szabályos haladást tudjuk ellenőrizni, hanem számos más szenzor pontosságát tudjuk mérni, illetve egyéb mérőszámokat tudunk megállapítani. Egy kritikus helyzet kialakulásának kezdetén, a rendszerre ható torziós erőhatások alacsonyak és az emberek ezt nem mindig tudják észlelni, viszont ezek számításokkal megállapíthatóak. Ezek a behatások fokozatosan, de rövid idő alatt nőnek, és egy bizonyos szintet átlépve balesetet okoznak. Az időben történő észleléssel ez elkerülhető. Ehhez előnyt tud jelenteni, ha meg tudjuk mérni a sebességet pláne ha ezt kerekenként külön külön vagyunk képesek megtenni, és ennek ismeretében végzünk számításokat valamint hajtunk végre műveleteket.

Sebességmérő eszközök számos módon készíthetők. Az autóiipari megoldások számunkra nem voltak alkalmazhatóak, a modellautó nem képes ezeket az eszközöket meghajtani, illetve a mechanikai kialakítása nem teszi lehetővé, vagy ha mégis akkor az egyes implementációk túlságosan nagy erő/hajtás veszteséggel járnának.

2.1.2 Miért kell kerékelfordulást mérni:

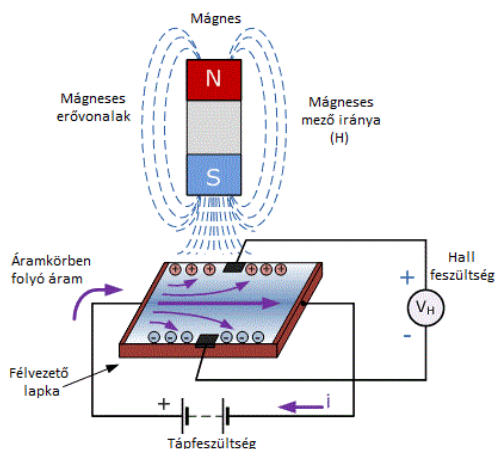
Alapvetően egy önvezető autó építésekor [13] a mérnököknek rengeteg szempontot kell figyelembe venni. Ezek közül az egyik, ha nem a legfontosabb kritérium az, hogy megbízhatóan és nagy pontossággal tudják követni a jármű mozgását. Erre rengeteg megoldás létezik, kezdve az egyes népszerű GPS alapú technológiákkal, viszont ezek csak a térbeli mozgás lekövetésére szolgálnak. Hogy ezt a mozgást és azt, hogy merre és mennyire fog elkanyarodni a jármű, egy a járműre erősített kerékelfordulás mérő műszer használata elengedhetetlen. Ezen információ felhasználásával a modellautó és később a teljes méretű önvezető autó is képes lesz egy automata önkalibrációt elvégezni teljesen egyedül. Egy kalibráció abból áll, hogy az autó indulás előtt ellenőrzi, hogy a kerék a való világban is ugyan úgy áll-e, mint ahogyan az autó belső memóriája szerint állnia kell. Ha nem így van, akkor addig fordítja a kerekeket a való világban vagy a virtuálisban (implementáció kérdése), amíg a két érték teljesen meg nem egyezik. Ez egy nagyon fontos lépés hiszen, ha a két érték eltér akkor az önvezető modell

utasításai hamar elcsúsznak a valóságtól és az balesethez vezethet. Jelenleg ezt a lépést manuálisan kell megtenni, de az elfordulás szenzorral ezt automatizálni lehet. Ezzel pénzt és időt spórolva mind a fejlesztőknek mind pedig a későbbi vég felhasználóknak. A valóságban lévő kerékállás szögének pontos ismerete, az irányítás pontosságát és az önvezető modell pontosságát is egyszerre javítja. Hosszas utánajárás eredményeként jutottunk arra, hogy jelenleg nincs a piacon számunkra megfelelő szenzor. Ez nemvolt nagy meglepetés mivel számunkra meglehetősen kicsi helyre kell egy nagyon pontos és tartós megoldás, mindezt természetesen a lehető legkedvezőbb áron. Az elmozdulás mérők, amiket mi találtunk, mind vagy túlságosan nagyok voltak a projektünkhöz, általában 5 vagy akár 20 centisek. Továbbá a költségvetésünkhöz mérten elrettentően drágák is, kb. ~150.000ft-ba kerültek. A legnagyobb baj velük viszont az volt, hogy nem egy mozgó kis modell autóra tervezték őket, hanem ipari gépekhez, ahol apró és stabil környezetben való elmozdulások mérésére alkalmazzák őket, vagy esetleges tudományos mérésekhez lettek tervezve szintén stabil felszínen. Számunkra a folyton rázkódó és mozgó modellautón lévő apró helyre ezért egyik megoldás sem volt alkalmazható. Ezért láttunk hát neki egy saját mérőszenzor megalkotásának.

2.1.3 A Hall szenzorról

Alapvetően a Hall szenzor egy olyan eszköz, amely a Hall-effektus segítségével képes detektálni a mágneses terek változását. A Hall szenzor kimeneti feszültsége egyenesen arányos a térerősséggel. Ily módon tökéletesen alkalmas mágneses mérések végzésére. Kicsi mérete, egyszerű struktúrája és hosszú üzemideje miatt nagyon széles körben használható. Fő felhasználási területe azonban a közelségérzékelés, a helymeghatározás és a sebesség valamint mozgásirány érzékelés [3]. Az automatizálásban, számítógépekben, és gyakran még az iparban is előszeretettel alkalmazzák különféle szenzorok formájában. Működését tekintve elég egyszerű, hiszen csupán egy áram alá helyezett félvezetőből áll. Az eszközt mágneses térbe helyezve a mágneses fluxus olyan erőt fejt ki a félvezető anyagra, amely a töltéshordozókat a

félvezető széle felé téríti el. Az elektronok oldalirányú mozgása miatt a félvezető két szélén potenciálkülönbség jön létre, amely fizikailag mérhető, (lásd 2. ábra).

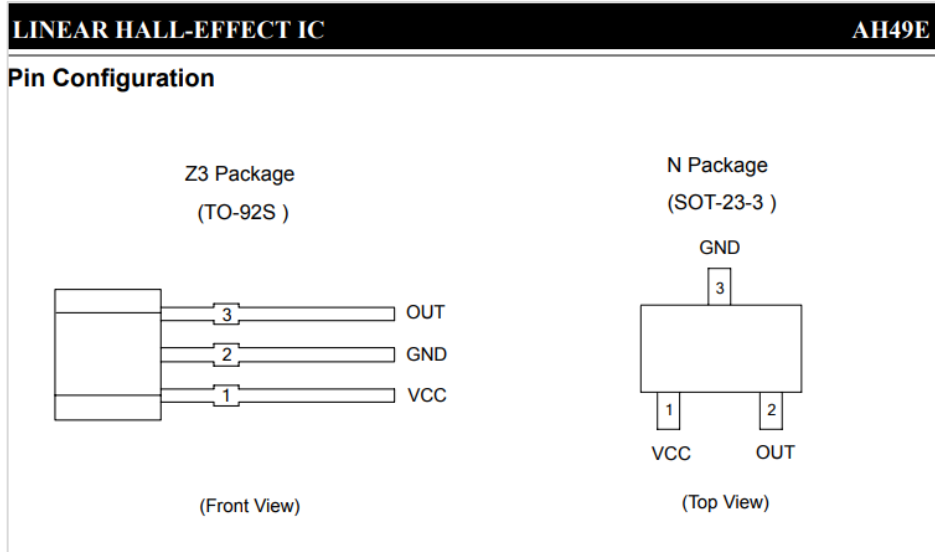


2. ábra Hall szenzor működése [4]

Hall szenzor, digitális vagy akár analóg kivitelben is kapható. A mi esetünkben mindkét szenzortípust használjuk, a kerékelforduláshoz analóg, a kerekek forgási sebességéhez pedig digitális/switch-es [5] szenzorokat. Az analóg szenzor használatakor a kimeneti feszültség pontos meghatározásához a következő egyenlet használata szükséges:

$$V_h = \frac{k * I * B}{d}$$

ahol, **V_h** a Hall feszültség voltban megadva, **k** a Hall-effektus együtthatója, **I** az érzékelőn áthaladó áram amperben megadva, **B** a mágneses fluxus sűrűsége Teslában megadva, **d** pedig az érzékelő vastagsága mm-ben megadva [6]. A Hall szenzorok lábainak kiosztása a (3. ábra)-n látható.

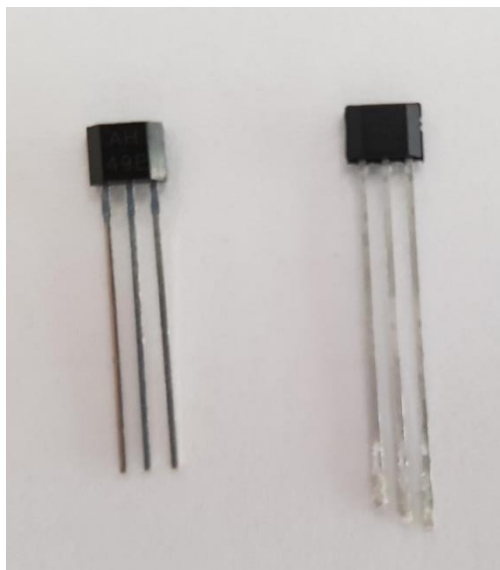


3. ábra AH49E pin configuration datasheet [7]

2.2. Kerék elfordulás

2.2.1 Alap Konceptió

Mindenekelőtt fontos letisztázni az alapvető problémákat és a megvalósítani kívánt célt. Fő célunk egy olyan eszköz létrehozása, amely segítségével egyszerűen legalább a fentebb leírt pontossággal képesek legyünk meghatározni az autónk kerekei által bezárt szöget. Ez egy igen fontos eszköz lenne, amivel folyamatosan tudnánk monitorozni az autó menetirányát. Erre a célra több ötletet is felvettünk, példaképpen az ultrahangos távolságmérés használatát. Alacsony ára miatt jó választásnak tűnt, viszont ezt a megoldást hamar elvetettük, hiszen a távolságmérés ezen formálja nem kellően megbízható vagy elége pontos. Csaba fejében felötlött a lézeres optikai érzékelő használatát is. Az ötlet alapja, egy fényérzékelő felszerelése az autó karosszériára és a kerekek innen mért távolságának megállapítása volt. Ugyan elég pontos mérést biztosítana, de sajnos erről a koncepcióról borsos ára és komplikált működése miatt kellett letennünk. Később Ferenc ötletét is átbeszéltük, miszerint egy forgó szenzor (rotational sensor) beépítésével mérjük a kerék összekötő tengely elmozdulását. Az utóbbi két ötlet merőben hasonló, viszont az egyszerűség miatt az előző fejezetben említett hall szenzor mellett tettük le végső voksunkat, (lásd 4. ábra).



4. ábra Hall szenzorok, balra analóg, jobbra digitális

Ebben a döntésben a Hall szenzor három számunkra döntő tulajdonsága játszott a legnagyobb szerepet. Először is egy súrlódás mentes szenzort szerettünk volna készíteni és egy hall szenzorral ezt tökéletesen meglehet valósítani. Másodszor a hall szenzoros ötletünk tűnt a legmegvalósíthatóbbnak, valamint legegyszerűbbnek. Harmadszor pedig a szenzor megfizethetősége, ugyanis így kényelmesen és bátran kísérletezhettünk vele. Ez későbbiekben a fő előnye is lehet a szenzorhálózatunknak, hiszen csak egyszerű és megfizethető alkatrészekből áll. Az alap ötlet, hogy Hall szenzorokból egy szenzorrendszert alkossunk meg, mellyel könnyen detektálhatóvá válik a kerekek elfordulásának szöge. A megközelítések merőben eltértek egymástól, azonban abban egyet értettünk, hogy a szenzorokat a padlólemezre kell rögzíteni. Efölött pedig, a már Ferenc által boncolgatott összekötő tengelyre erősített mágnes használata lehet célravezető. Abban is egyetértésre jutottunk, hogy a pontosabb mérés érdekében több szenzor használata is szükséges, ugyanis kevés szenzor használata esetén nem elég pontos a mérésünk. Végző soron mindketten beláttuk, hogy a Hall szenzorokkal való távolságmérés a legköltséghatékonyabb és a szenzorok halmozása által a legpontosabb mérési forma is. Mindezek után el kellett döntenünk, hogy switch-es, azaz mágneses tér hatására aktiválódó és konstans digitális jelet adó szenzort vagy lineáris, azaz analóg jelet egy mágnes távolságától függően változó jelet visszaadó szenzort használjunk. A véleményeink ebben is eltértek. Csaba a digitális szenzort preferálta hiszen így kisebb sűrűségben implementálhatók a szenzorok és elméletben egyszerűbb működést tesz lehetővé. Ha a mágnes éppen az egyik „Switch” típusú szenzor előtt helyezkedik el, akkor az magas jelet bocsát ki és így egyszerűen

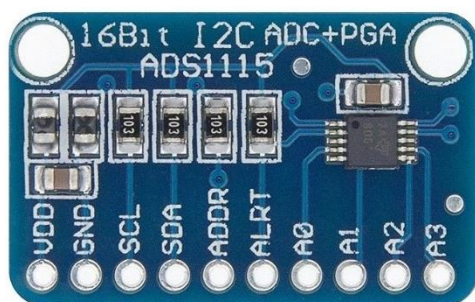
megmondható, hogy hol tart éppen a mágnes. Gyakorlatban viszont teljesen más eredményeket kaptunk ezzel a megoldással. Könnyen előfordult olyan eset, ahol egyszerre több szenzor is aktiválódott a mágnes közelsége miatt, ezáltal összezavarva a központi jelfeldolgozó egységet, amely nem tudta eldönteni, hogy a két, három aktív switch közül melyik mond igazat, azaz, hogy hol van pontosan a mágnes. Ferenc megközelítése sokkal jobban működött, ő több egymáshoz közel elhelyezett lineáris szenzorokból álló szenzorrendszert álmodott meg. Az érzékelők mindegyike egy analóg jelet bocsát ki, melyeket egy vagy több analóg-digitális konverterrel alakítunk át. A küldött jelek közül a maximális értéket kiválasztva egyszerűen eldönthető, hogy éppen pontosan hol áll a mágnes. Ezzel a megoldással már nem áll főt az előbb említett switch-es megoldás problémája. Ezen a megközelítéssel, ha több szenzor érzékeli is a mágnest, a szenzorok mérése elég pontos ahhoz, hogy mindig tudjunk választani közülük egy maximum értéket. A maximum érték ezesetben pedig megmondja, hogy melyik érzékelőhöz van legközelebb a mágnes. Így esett a választásunk az „AH49E” típusú analóg Hall szenzorra.

2.2.2. Analóg-digitális konverter (ADC)

A projektünkben használt „AH49E” szenzorok analóg Hall szenzorok. Ezek analóg jeleket továbbítanak a központi jelfeldolgozó egység felé. Az adatok feldolgozására mi a fejlesztés során eleinte egy Arduino Uno fejlesztőpanelt [8] használtunk. Ezen a fejlesztőkártyán viszont csak 6 darab analóg jelfeldolgozó bemenet kapott helyet. Így a mi munkánk során szükséges 15 szenzor szolgáltatja jelet nem tudnánk feldolgozni. Ezt az akadályt mindenféleképpen ki kellett küszöbölnünk, hogy megfelelő pontossággal végezhessük el a méréseket. Erre a megoldás Csabától jött, miszerint használjunk analóg- digitális konvertereket (ADC). A beérkező jeleket ezáltal egyszerűen átalakíthatjuk olyan digitális jelekké, amikkel már a mérési adatok feldolgozhatók. Erre a feladatra egy „ADS1115” [9] típusú ADC-t választottunk, (lásd 5.

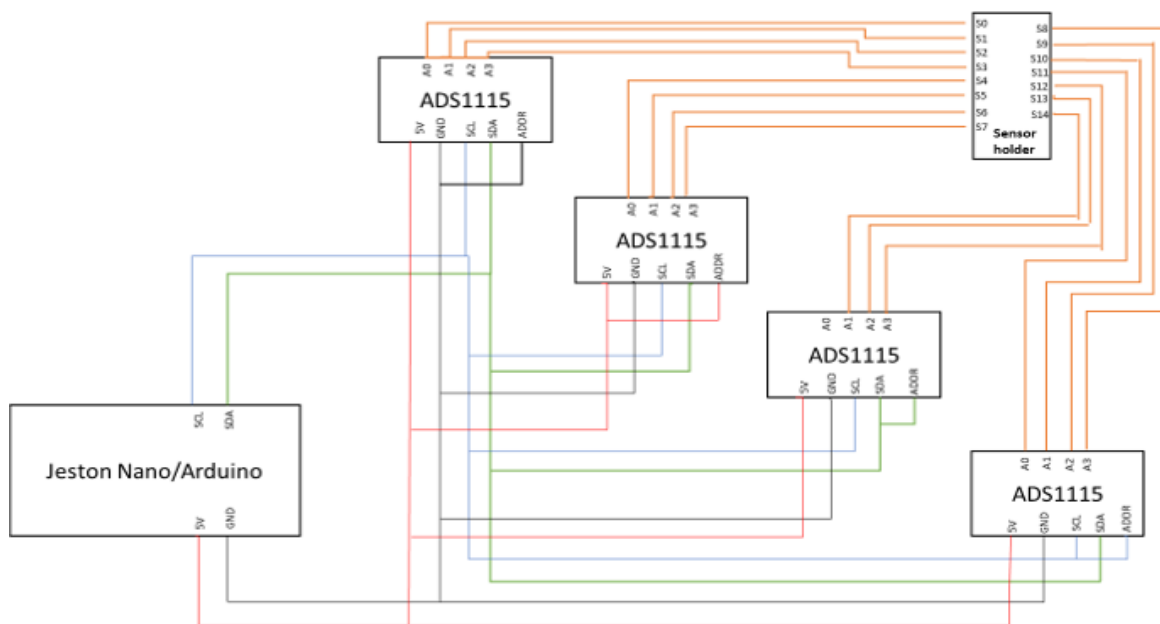
ábra) alacsony fogyasztása, kicsi mérete, 16 bites pontossága és I2C szabványon való kommunikációja miatt.

Ezen paraméterek mellett ez a legjobb választás, hiszen beépített programozható jelerősítővel is rendelkezik, ami akár 16x-os erősítést is lehetővé tesz. Ily módon a gyengébb jeleket is elég erőssé tehetjük a detektálásra. Mindezek mellett kellően gyors 860 minta/perces adatsebességet biztosít. Működését tekintve még egy 4:1-es multiplexer is helyet kapott a 4 analóg bemenet



5. ábra ADS1115 Analóg-digitális konverter

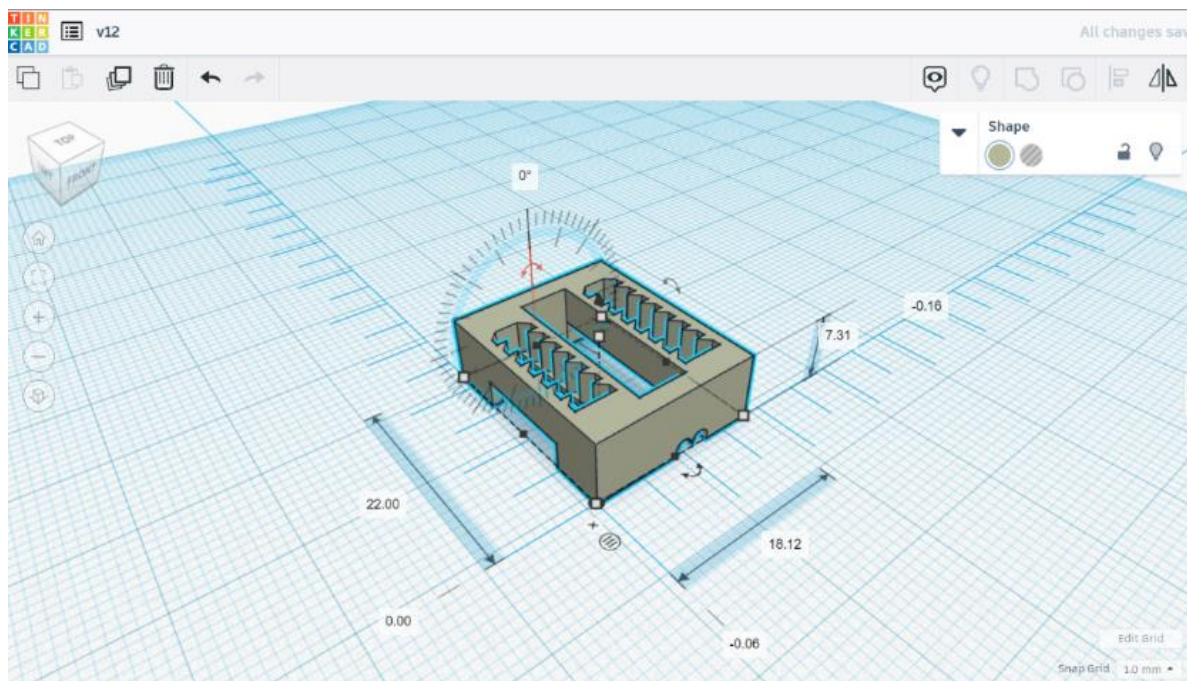
elkülönítésére. A mi nézőpontunkból viszont a legfontosabb tulajdonsága, hogy 4 ilyen eszköz összekapcsolását is támogatja, amivel akár 16 szenzor által küldött jel feldolgozása is lehetséges. Az összekapcsolt eszközök segítségével a mi 15 hall szenzorunk könnyen feldolgozhatóvá válik a központi feldolgozó egységnek. Ezen átalakítók segítségével a későbbiekben bármilyen központi egység használhatóvá válik a jelek feldolgozására, legyen szó akár egy Raspberry Pi-ről, Arduino Uno-ról vagy egy Jetson Nano-ról is, (lásd 6. ábra).



6. ábra Szenzorhálózat kapcsolási rajza

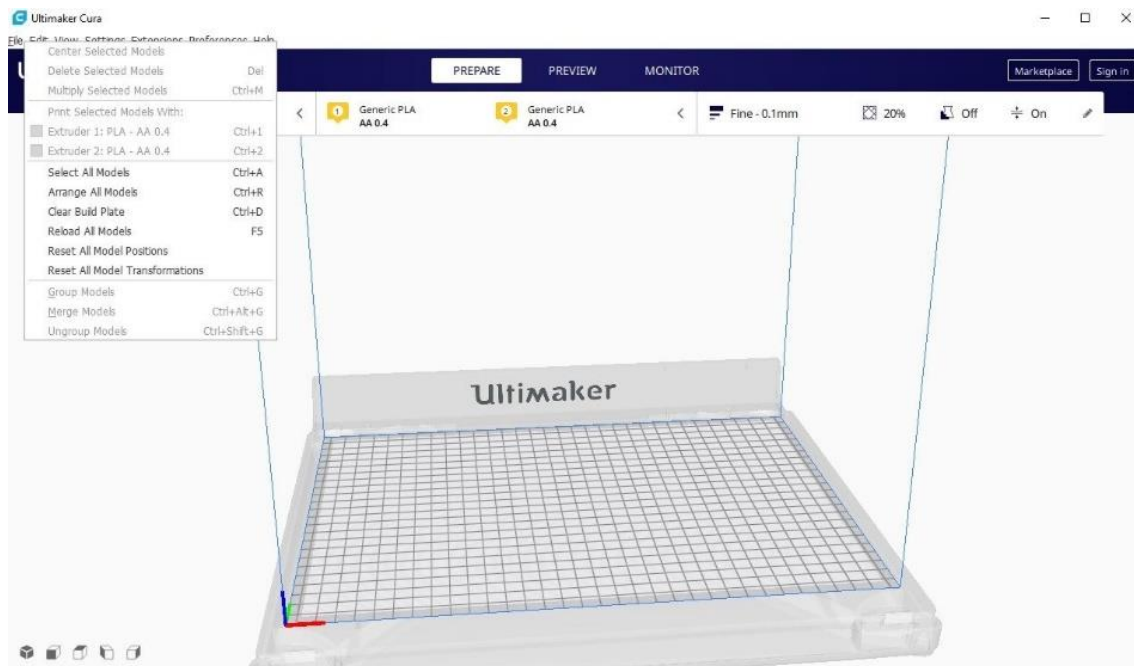
2.2.3. 3D Tervezés

Mivel ennyire kicsi és a mi céljainknak megfelelő szenzor jelenleg nincs a piacon, így mindent nekünk kellett megtervezni. Ebbe természetesen beletartozik a szenzortartó ház és a kerékösszekötő rúd tervezése is. Minden elemet 3D nyomtatással készítettük el. A nagyon szűkre szabott hely miatt ezeket először viszont tized milliméter pontosan meg kellett tervezni. Csaba vezetésével a TinkerCAD nevű ingyenes szoftvert használtuk erre a célra (lásd 7. ábra).



7. ábra TinkerCAD tervezőszoftver

Több tucatnyi 3D modellt készítettünk a különböző megoldásaink kipróbálására, tesztelésére. Ezzel a prototipizálás folyamata is nagyban felgyorsult. Az elkészült 3D modelleket a CURA Ultimaker nevű szoftverrel szimuláltuk és slice-oltuk felkészítve ezzel őket a nyomtatásra (lásd 8. ábra). A modellek kinyomtatására pedig egy Prusa i3 MK3S+ nyomtatót használtunk.

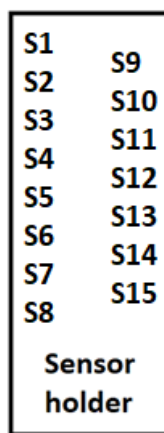


8. ábra Cura Ultimaker 3D nyomtató szoftver

2.2.4. Mágnes pozíciója

Munkánk megkezdésekor úgy terveztük, hogy a mágnes egy sor szenzor előtt halad el, amelyek az előttük elhaladó mágnes helyzetét detektálják a maximum érték kiválasztásával. Sajnos jó pár teszt és jó néhány prototípussal később, kellett rá jönnünk, hogy ez az ötlet nem megfelelő, mivel a Hall szenzorok, többek között a mi AH49E szenzorunk sem mér minden irányba mindig egyformán. [10] Az „arca” és teteje a mi általunk használt mágneseket nem megfelelően méri. Erre született Csaba egyszerű de elegáns megoldása miszerint a mágnesek oldalát használjuk. Ehhez azonban teljesen újra kellett tervezni a tartót, hogy a mágnes a két sor szenzor között mozogjon (lásd 12. ábra).

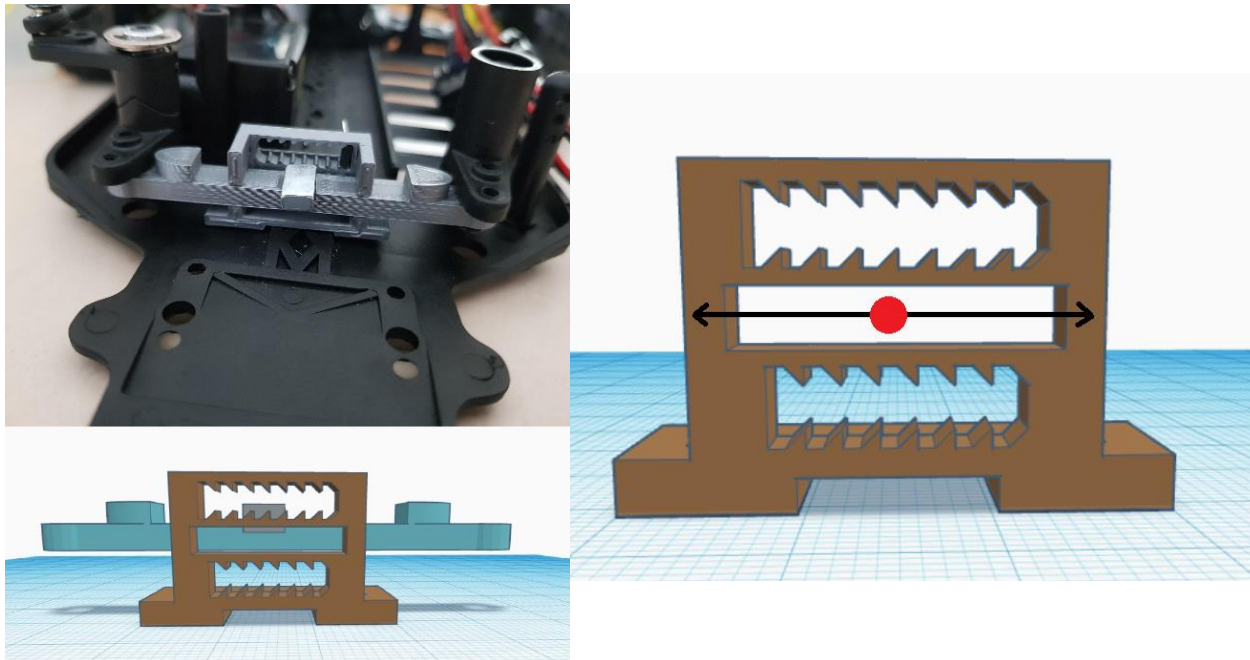
Egy másik jelentős probléma volt az, hogy a mérés során könnyen előfordulhat olyan pozíció, ahol a mágnes pontosan két szenzor között helyezkedik el hibahatáron belül. Így a munkánk nagyban bonyolódna és nem tudnánk a néhány fokos kitűzött mérési pontosságot elérni. Szerencsére, ha Ferenc ötlete szerint két külön sorba rendezzük az érzékelőket, egymás fölé és alá, míg a mágnes a két sor között egyfajta alagútban vagy csatornában haladna. Így könnyen kiküszöbölhető az előbb említett probléma a mérési pontossággal, sőt így még a tervezettnél is pontosabb lett a szenzorhálózatunk. Csupán annyit kell tennünk, hogy átfedésben helyezzük el, (lásd 9. ábra) a két sornyi szenzorjainkat, így egy csapásra majdnem megduplázva a szenzoraink sűrűségét és remélhetőleg a teljes hálózat pontosságát.



9. ábra Szenzorok elhelyezkedése

Ezt követően Csaba meg is tervezte egy 3D tervező programban (TinkerCad) a szenzortartót, (lásd 10. ábra jobb oldali ábrája). Ez az első prototípus talán egy hirdetőtáblára hasonlít leginkább, viszont reklám felület helyett a két szenzorsorunk és a mágnes csatornája kapott helyet rajta. A láb területén pedig egy másik csatorna vagy kivágás van amire azért van szükség, hogy a modellautó padlólemezén lévő „gerinc” vagy másnéven megerősítésre kényelmesen és síkon rátudjon férti és stabilan álljon a szenzorhálózatunk, ne lötyögjön balra jobbra. A kinyomtatást követően megpróbáltuk összeállítani az eszközt a kisautón, de újabb probléma adódott. Ezúttal az, hogy nemes egyszerűséggel a mért adatok egyáltalán nem voltak pontosak. Rengeteg szenvedés és keresés után rájöttünk, hogy ez magából a szenzortartó kialakításából adódott. A fő gond az, hogy a mágnes nem pontosan a szenzorok közötti sávban mozog, hanem valamivel elcsúszva, (lásd 10. ábra bal oldali ábrája) szinte majdnem csak az egyik sor felett halad. Ezt a problémát mindenképpen meg kell oldani, ha pontos mérési eredményeket akarunk. A megoldás ezúttal szerencsére olyan egyszerű volt, mint amilyennek hangzik és hamar meg is született. Csak a mágnes pozícióját kellett állítani a tengelyen. Az így elkészített szenzorhálózat

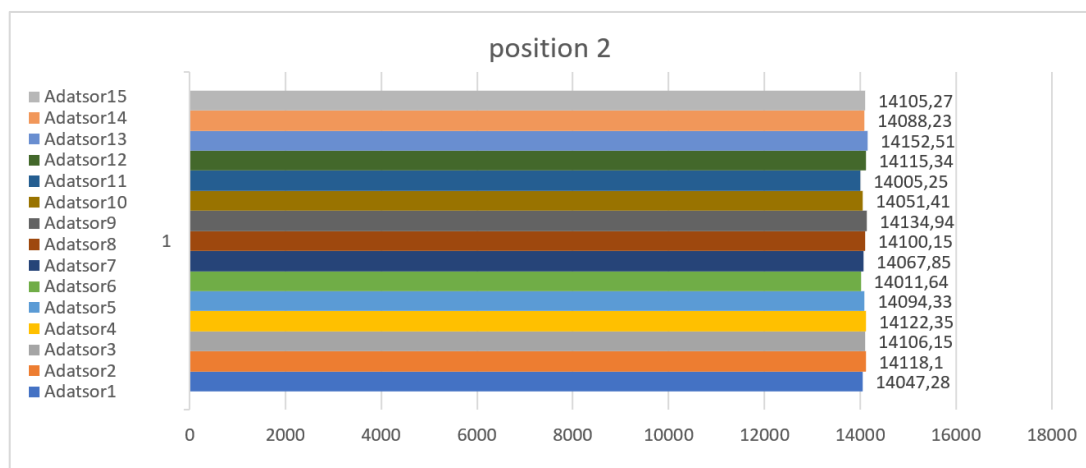
és tengely + mágnes páros már kellően pontos eredményeket biztosított, viszont egy újabb nem várt viszont annál súlyosabb problémát eredményezett. A kialakítás túl magasra sikerült, így már nem fért be az autóba a kardántengely alá.



10. ábra Elcsúszott mágnes

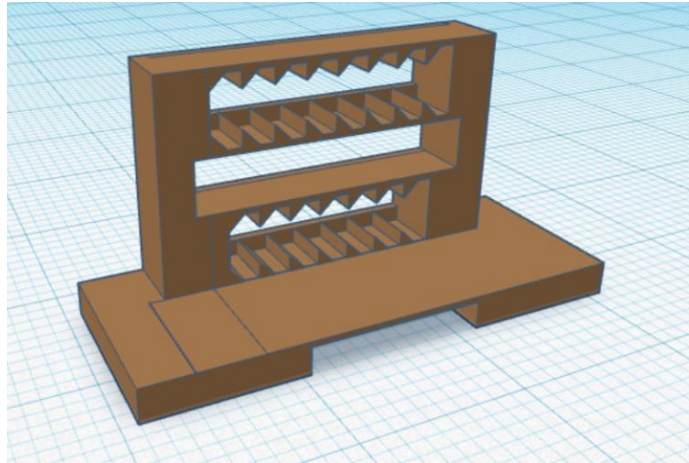
2.2.5. Álló megoldás

A projekt keretében elsőként egy álló szenzorhelyezést szerettünk volna megvalósítani. Ez több okból is ideális megoldásnak tűnt. Az előbbi pontban említett megoldás használatával, miszerint a mágnes a szenzorsor fölött halad, ez egy tökéletesen működő koncepciónak ígérkezett. Méreteit tekintve is tökéletesen beleillett az autó váza által szabott szűk keretekbe. Egy ilyen elhelyezkedésű szenzortartó esetén a szenzorok lábai mind egy irányba néznek. Ezáltal könnyű a forrasztásuk és a kábelezés is egyszerűsödik, ugyanis minden láb egy irányba néz. Ez a megoldás nagyon egyszerűnek és tökéletesen működőnek ígérkezett. A szenzorrendszer összeszerelése után azonban hamar nyilvánvalóvá vált, hogy sajnos nem megfelelően működik a megoldásunk. A méréseket megkezdve azonnal feltűnt, hogy nem elég pontosak az adatok. A szenzorok egyéni jellemzője ugyanis az, hogy ha a szenzor tetejéhez közelítjük a mágneset, nem kapunk olyan pontos mérési eredményeket, mintha azt az oldalához közelítenénk. Mint az a (11. ábra)-n is jól látható az értékek nem különíthetők el eléggé ahhoz, hogy valóban megállapítható legyen a maximum érték.



11. ábra Mért adatok

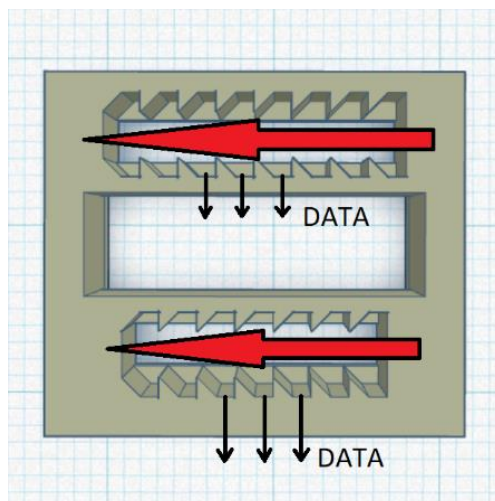
Tulajdonképpen így a mérési adatok nem használhatóak. Emiatt a hiba miatt meg kellett változtatnunk az egész koncepciót. Ez a szenzortartó teljes újra tervezését vonta maga után. Ezúttal egy olyan megoldásra váltottunk, melynél a mágnes a két szenzorsor között mozog. Egy ilyen megoldás több okból is előnyösebbnek ígérkezett. Elsőként a mágnes a szenzortartó egy olyan részen mozog, amely pontosan a két szenzorsor között mozog. Ennek a méretei pontosan a mágneshez igazodnak, ezáltal elkerülve azt az esetet, ahol a mágnes véletlen elmozdul valamelyik szenzorsor irányába. Másrészt az így mért adattok így sokkal pontosabbak és megbízhatóbbak. Kevésbé fordulnak elő az előbb említett pontatlansági problémák. A 3D tervek modifikálása viszonylag hamar megtörtént és ezek után ki is nyomtattuk az új szenzortartót, (lásd 12. ábra). Amint elkészült belepróbáltuk a karosszériába, de sajnos ismét egy roppant bosszantó probléma ütötte fel a fejét, amit már az előző fejezetben is megemlítettünk. A szenzortartó túl magasra sikerült és beleért a kardántengely síkjába. Eme problémának más megoldás nem lévén kénytelenek voltunk egy teljesen más hozzáállású új eszközt tervezni és alkotni az alapokról. Több egymástól teljesen különböző megoldást is megfontoltunk köztük azt is, hogy a szenzortartók kb~45 fokos szögben megdöntve helyezzük el az autóban. Végül megoldás gyanánt az egyszerűsége és praktikumra hivatkozva egy fekvő megoldás mellett döntöttünk.



12. ábra Álló megoldás

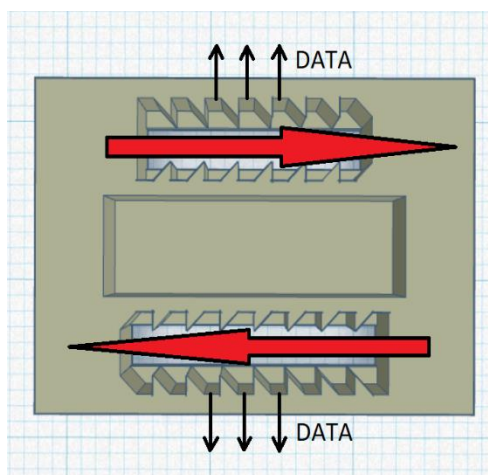
2.2.6 A Fekvő megoldás

Mivel helyhiány és az előbb taglalt problémák miatt kénytelenek voltunk egy fekvő szenzortartó megoldást alkalmazni, ebből logikusan következik, hogy minden már megtervezett és kinyomtatott alkatrészt újra ki kell mérni tervezni, kinyomtatni és tesztelni. Viszont az új megoldás más rejtett előre nem várt problémákat is okozott, amiket meg kellett és sikeresen meg is oldottunk. Az első probléma maga a tartó vastagsága. Mindenképpen szükséges, hogy a tengely alá férjen anélkül, hogy megakadályozná a rudat a szabad mozgásban. Mindemellett a súrlódás és az ebből fakadó kopáshoz és anyagfáradáshoz vezető rendellenességek elkerülése is fontos cél. A tartó teljes átdolgozásával és egy szenzortartó „fül” megtervezésével drasztikusan tudtuk vékonyítani a tartót is. A tervezés végeztével a tartó szinte semmivel sem volt vastagabb vagy szélesebb, mint ha csak a szenzorokat egymásmellé tettük volna bármi nemű tartó nélkül. Egy teljesen másik eredetű új nehézség a kábelek vezetése és összeforrasztása. Álló megoldásnál a kábeleket egyszerűen csak hozzáforrasztottuk a lábakhoz, hiszen a hely bőven a rendelkezésünkre állt a tartó oldalán. Egy fekvő megoldásnál viszont, ahol nagyon fontos, hogy minél alacsonyabb legyen a szenzortartó hálózat és hogy kényelmesen beférjen a tengely alá a szerkezet, egy teljesen más hozzáállást követel. Eleinte a



13. ábra Fekvő megoldás szenzorirányai

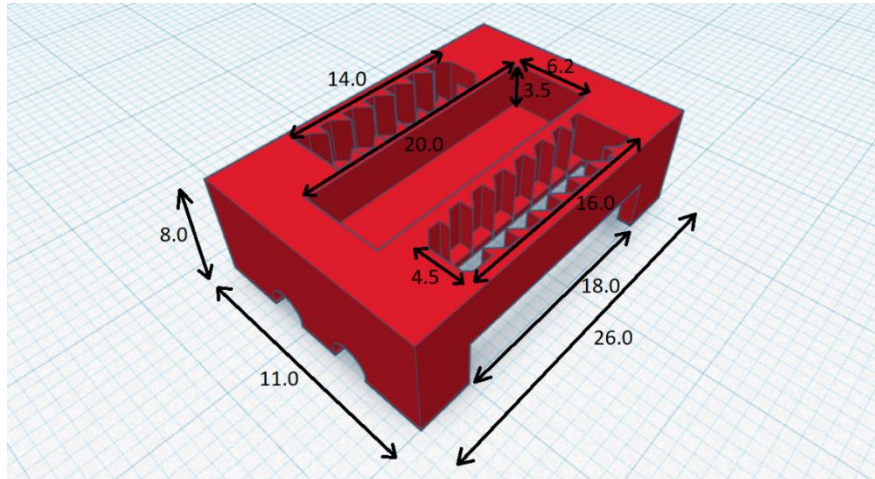
szenzorok egy irányba álltak ezzel együtt az érzékelők DATA lábai is ugyan abba az irányba néznek. Mint azt a (13. ábra) is jól mutatja a felső sor DATA kimenetei egy fekvő megoldásnál szinte megközelíthetetlenek és még a mágnes szabad mozgását is gátolhatják. Megoldásként azt láttuk logikusnak, hogy az egyik sort megfordítsuk. Így az egyik sor balra, a másik jobbra néz, ezáltal a DATA lábak a ház szélén jönnek ki nagyban megkönnyítve a csomagolást, a kábelvezetést és a forrasztást, (lásd 14. ábra).



14. ábra Fekvő megoldás fordított szenzorirányban

A szenzor 3 lába közül (VCC, GND, DATA) helytakarékoság miatt az összes GND és VCC lábat összeforrasztottuk. Mindezt úgy, hogy a két szélső lábat lehajtottuk majd a többi ezekhez töben hozzáforrasztottuk és a felesleget levágtuk. Ezzel a megoldással rengeteg helyet nyertünk mivel nem kell minden egyes lábhoz (3*15) egyesével egy-egy kábelt forrasztani és értékes helyet, időt, valamint nyersanyagot spóroltunk. Így végül, mint már említettük a tartó alig

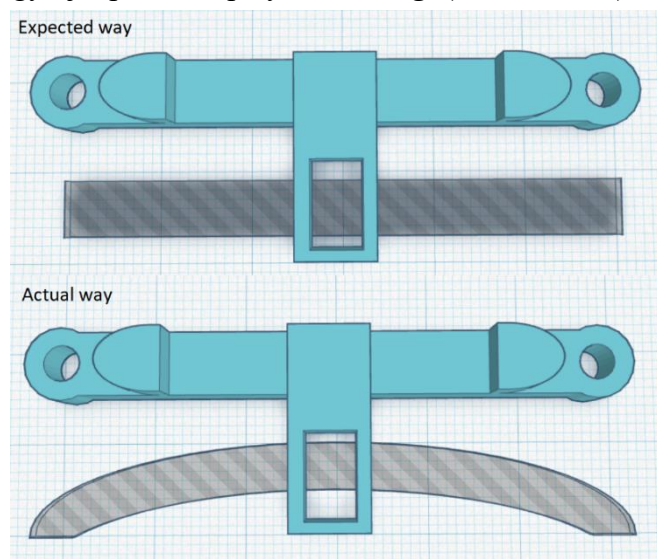
vastagabb egy szenzor vastagságánál, ami szükséges is, ha be akarunk férti az összekötő tengely alá, (lásd 15. ábra). A jövőben a csomagoláson egy egyéni nyák megtervezésével és legyártásával szeretnénk javítani. A nyáklap még több helyet szabadítana fel és egyszerűsítene az összeszerelést is.



15. ábra Szenzortartó végső paraméterei (mm-ben értendő)

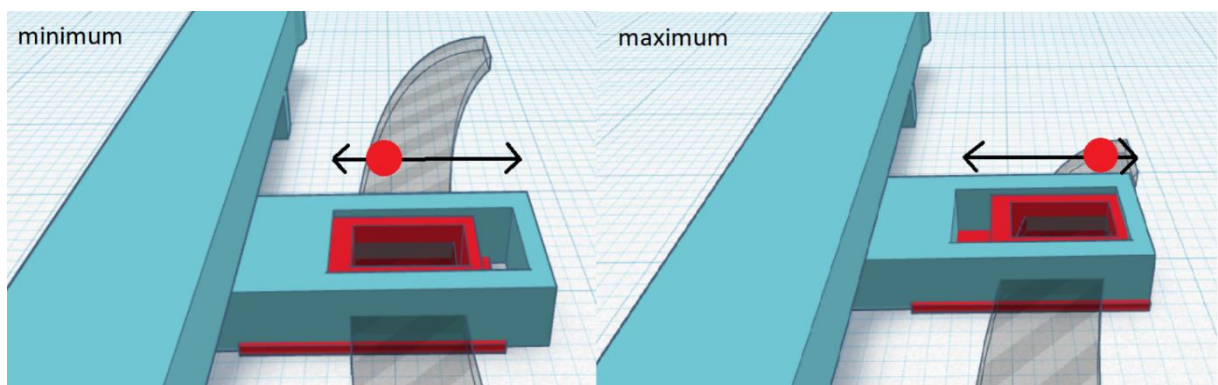
2.2.7. Teleszkóp alapú kerékösszekötő tengely tervezése

Az új fekvő megoldásunkkal ugyan rengeteg helyet nyertünk és valamivel talán elegánsabb is lett a szenzorhálózatunk viszont egy új kellemetlenség is felütötte a fejét. Ahogyan az első 3D nyomtatott fekvő szenzortartót a kocsira próbáltuk, azonnal észleltük, hogy valami nincs rendben. Mi ugyanis azzal számoltunk, hogy a tengely, ami a két elülső kereket összeköti és aminek az elmozdulását próbálnánk mérni egyenesen mozog. Ez viszont sajnos nem így van, hanem útja során egyfajta parabola pályát tesz meg, (lásd 16. ábra).



16. ábra A tengely elmozdulása

Eleinte, amikor álló szenzorhálózattal dolgoztunk ez fel sem tűnt és nem is okozott problémát. A jelenlegi fekvő koncepciónál azonban komoly fejfájást okozott és csak a sokadik megoldás bizonyult megfelelőnek. Először egy íves szenzortartót láttuk a jó megoldásnak. A mágnes útjának megfelelő ívben hajlított tartó képes lenne ezt a problémát orvosolni. Viszont amint elkezdtünk komolyan foglalkozni ezzel a megközelítéssel világossá vált, hogy ez nem a legpontosabb megoldás és túlságosan komplikált is. Nem is beszélve a gyártási problémákról és a meglehetősen nagy hely veszteségről, amit ezzel a konfigurációval elszenvednénk, egy egyenes megoldással szemben. Az alternatívák között volt több különböző mozgású és alakú tengely és szenzortartó ötlet is. Néhány ilyen terv a 3D tervezési szakaszig is eljutott mielőtt elvetettük volna őket. Mindezek helyett lett Ferenc ötlete a nyerő, miszerint a szenzor tartó változtatása helyett nyomtassunk saját rudat. Ez a mágnes rögzítését és pozicionálását is jelentősen megkönnyebbítené. A jövőben pedig, mint egy szenzor csomag a tengely + szenzortartót együtt bármelyik másik kisautóra is fel lehetne szerelni és az így képessé válna a kerekek elfordulásának mérésére. Mindössze ezen két standard alkatrész felszerelésével akár modell akár 1:1 méretű önvezető autó is képessé válna a kerekeinek elfordulásának mérésére. A parabolikus pálya okozta gondot az új konfiguráció úgy oldja meg, hogy a mágnest nem közvetlenül van rögzítve a tengelyre, hanem egy kis tartóban kap helyet. Ez a tartó pedig képes a kar majdnem teljes hosszában fel és le is mozogni, (lásd 17. ábra).



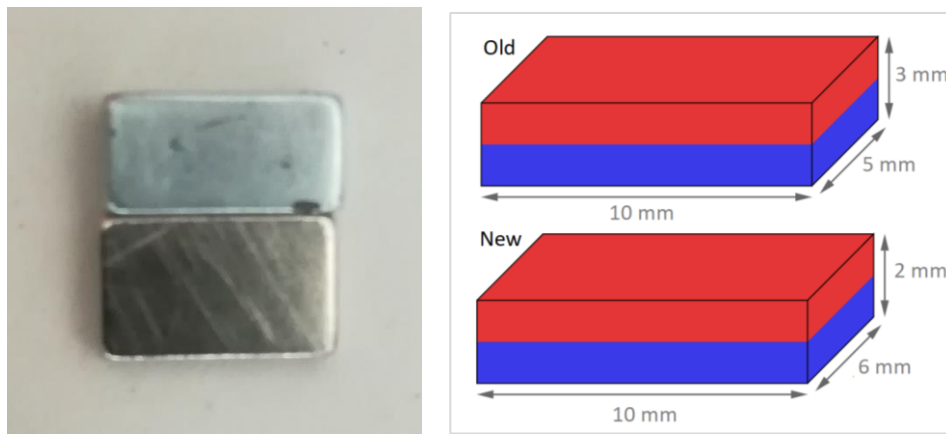
17. ábra Mágnes tartó elmozdulása

Ezáltal a mágnes képes végig a neki kiépített alagútban maradni és az összes Hall szenzor előtt elmozdulni egy egyenes mentén miközben a tengely a saját parabola ívén halad továbbra is. Így az íves pályán mozgó összekötő tengely elmozgását egy lineáris úton tudjuk pontosan és valós időben mérni. Az viszont, hogy elméletben van egy jól működő megoldás nem jelenti azt, hogy ez ugyan olyan jól képes teljesíteni a valóságban is. Rengeteg aprólékos mérésre, számításra és legalább egy fél tucat 3D-s tervre volt szükség ahhoz, hogy a prototípusok folyamatos

frissítésével és apró pontosításaival elérjük a tökéletes vastagságot, magasságot és a mágnesstartó sínjének megfelelő hosszát.

2.2.8. Mágnes csere

A prototípezés folyamatának részeként a mérési pontosság javítása érdekében egy vastagabb (3 mm) mágnesről egy jelentősen vékonyabb (2 mm) vastagságú mágnesre voltunk kénytelenek váltani (lásd 18. ábra).



18. ábra Régi és Új mágnes

Az új mágnesnek nem csak a vékonysága és magasabb gyártási minősége emelte a régi mágnes fölé, hanem a sokkal erősebb és jóval egyenletesebb, kiegyensúlyozott és kiszámítható mágneses tere is. Viszont a vékonyságnak ára van, mégpedig a magasság. Az új mágnes, a régi (5 mm) magasság helyett (6 mm) magas és mivel a mágnes hosszában helyezkedik el a tartóban, így jelentősen át kellett dolgozni a tartót a tengellyel és a kis mágnesstartóval együtt. Ez sokkal nehezebb és összetettebb feladat, mint elsőre hallatszik mivel a szenzor már sem szélesebb sem vastagabb nem lehet ugyanis nincs elég hely. Ezért egyfajta végső megoldásként a tartó falainak drasztikus vékonyításával kísérleteztünk. Ezt mindaddig folytattuk, amíg a végső tervből nyomtatott elemek elég kicsik nem lettek ahhoz, hogy beférjenek a magasabb új mágnessel együtt a szenzortartóba. Ezen folyamat során több fals prototípust is kinyomtattunk. Ha túl vékonyra vettük a falakat a 3D modellező programban még szépnek és pontosnak látszódtak a terveket, viszont ezen pontosságra (amikor már negyed milliméterekről beszélünk) a Prusa 3D nyomtatónk nem volt képes. Gyakran előfordult, hogy a kinyomtattott elemeknek egyáltalán nem volt fala vagy ha volt is, akkor görbére és hajszál vékonyra sikeredett, ami már statikailag nem volt eléggé masszív a szenzorok helyén tartására. A rengeteg mérés és pontosítás végeztével viszont sikerült egy olyan kompromisszumos koncepciót létrehoznunk, amely már

megfelelt a céljainknak. A tervezési és gyártási periódus befejeztével végül mind a ketten úgy gondoltuk, hogy nagyon is megérte ez a sok extra munka ugyanis a mérési adatok látványosan javultak a jobb mágnes alkalmazásának köszönhetően. A későbbiekben már mechanikai problémák sem jelentkeztek, tehát kutatásunk ezen részét tökéletes kivitelezéssel zártuk, nem beszélve a rengeteg tervezés közben elsajátított tapasztalatról.

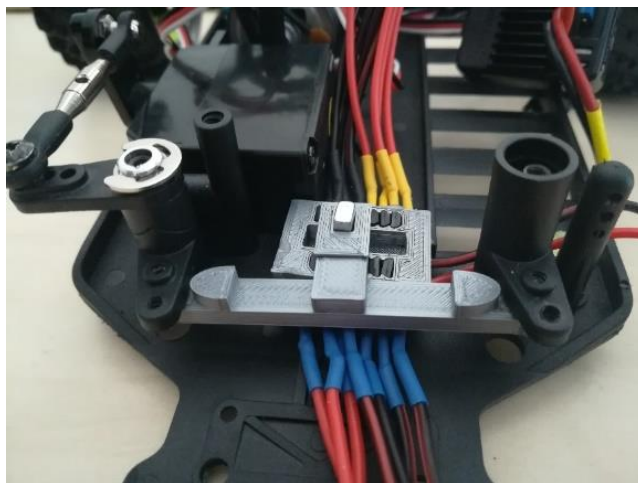
2.2.9. Kábelproblémák

Munkánk során sok-sok buktató és nem kevés probléma is jelentkezett. Közülük az egyik, ha nem a legfontosabb a kábelek mennyiségéből fakadóan a vastagságuk és hosszuk révén jelentkezett. Az elkészült és helyére illesztett eszköz lábai ugyanis túl rövidek lettek volna ahhoz, hogy ezekkel közvetlenül csatlakozzunk az „*Analog to Digital Converter*” (ADC)-khez. Amennyiben ez a probléma nem is jelentene gondot, a 4 darab ADC elhelyezése a szűkös helye miatt egy újabb megoldandó feladatot vetne elénk. A lábakat tehát szenzoronként 1 kábellel, azaz összesen 15 kábellel kellene összekötni. Nem beszélve a VCC és a GROUND lábaknál használt vezetékekről, amelyekkel együtt ez a szám már tizenhétre rúg. Ekkora mennyiségű kábelköteg már komoly előre tervezést és jól kigondolt kábelvezetést igényel. Erre a célra eleinte egy relatíve vastagabb réz huzalú hangtechnikában használt vezetéket használtunk, (lásd 19. ábra jobb oldali).



19. ábra Használt vezetékek

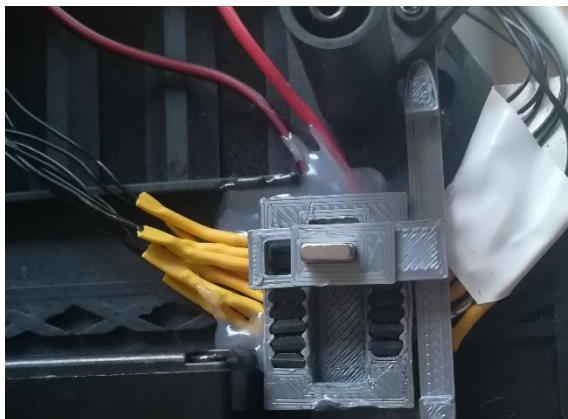
Erre az indok egyszerűen az volt, hogy nem állt rendelkezésünkre ennél vékonyabb a laborban. A 15 láb egyenkénti felforrasztását követően még egy-egy zsugorcsonyot is hasznosnak találtunk rámelegíteni a lábakra. Ezáltal is elkerülve az esetleges áthúzásokat vagy a lábak összeéréséből adódó fals jelek mérését. Mindez még azzal az előnnyel is járt, hogy a sok hajtogatástól és forrasztgatástól eddigre elgyengült/elfáradt OUT/DATA lábakat valamilyen szinten megvédte a szakadástól, töréstől. Ezek elkészülte után megpróbáltuk az autóra helyezni a mérőeszközt, (lásd 20. ábra) azonban hamar egy újabb kellemetlen akadályba ütköztünk.



20. ábra Elsőnek elkészült szenzorrendszer

A használt kábel vastagsága és a zsugorcsonyok hossza miatt az autóra helyezés túlságosan nehézkessé vált. Olyannyira súlyos lett a helyhiányunk, hogy a kábelek megfelelő helyen való elvezetése közben egymásután sorra elkezdtek letöredezni a szenzorunk adatkivezetési lábai. Megpróbáltuk menteni a menthetőt és visszaforrasztgatni egyesével a lábakat, viszont a szenzor sok ki-be helyezgetése által egyre több láb tört le vagy gyengült el. Emiatt az egész koncepciót újra kellett gondolnunk. A megoldás szerencsére nagyon hamar megszületett. Alapos kutakodás után Csaba találta ki a megoldást, miszerint használjunk egy nagyon vékony egy eres kábelt. Mivel ez előtt nem dolgoztunk vagy hallottunk ilyen vékony kábelről, úgy hittük, hogy az előző most már csak „vastag” kábelnek nevezett is elég vékony lesz, viszont az akkor még nagyon vékonynak tűnő kábelből már mindössze 15 is vastos kábelköteget alkotott. A laborban sajnos nem rendelkezünk ilyen vékony kábellel, tehát beszerzőútra kellett indulnunk. Ezután már hamar össze is szereltünk egy új szenzort amire már a jóval vékonyabb új huzalt illesztettük. Ez igen trükkös vállalkozásnak bizonyult, mivel az új kábel vékonysága növelte a forrasztás nehézségét a már így is miniatűr alkatrészekből álló szenzorhálózaton. A forrasztáshoz használt ón gyakran nem szándékozott megtapadni rajta, így mind a 15 forrasztás meglehetősen macerás

feladattá vált. Csak a 15 adatláb kapott ilyen vékonyabb új kábelt, a VCC és a GROUND maradt a régi „vastag” kábel. Ennek egyszerű az oka, ezek a kábelek (VCC, GND) a tartó oldalán jönnek ki, ahol jóval több helyel rendelkezünk, továbbá áram járta vezeték lévén is maradtak a régi kábelek. Az új megoldás nagyon jól sikerült, hiszen a kábelek tökéletesen elfértek az adott elvezetési pontokon és nem terhelték túl az adatlábakat sem, (lásd 21. ábra). Egyszóval a kábelezés megvalósítása tökéletesen sikerült.



21. ábra Végző kábelelrendezés

2.2.10. Szoftveres háttér

A fizikai hardverek készítésénél alap koncepcióként egy Arduino fejlesztőpanelt használtunk. Ennek köszönhetően a szoftverek korai változatai Arduino implementációval is rendelkeznek.

Arduino kód részlet:

```
1. #include "ADS1X15.h"
2.
3. ADS1115 ADS[4];
4. uint16_t val[16];
5. int16_t degree=0;
6. int16_t maxflag=13;
7. int16_t minflag=13;
8. int idx = 0;
9. void setup()
10. {
11.   Serial.begin(115200);
12.   Serial.println(__FILE__);
13.   Serial.print("ADS1X15_LIB_VERSION: ");
14.   Serial.println(ADS1X15_LIB_VERSION);
15.
16.   for (uint8_t i = 0; i < 4; i++)
17.   {
18.     uint8_t address = 0x48 + i;
19.     ADS[i] = ADS1115(address);
20.
21.     Serial.print(address, HEX);
22.     Serial.print(" ");
23.     Serial.println(ADS[i].begin() ? "connected" : "not connected");
24.
25.     ADS[i].setDataRate(7); // 7 is fastest, but more noise
26.   }
27.   ADS_request_all();
28. }
```

Raspberry PI kód részlet:

Az Arduino fejlesztőkártya korlátozott erőforrásai hamar feltűntek számunkra, ami miatt azonnal egy Jetson Nano-val szerettük volna folytatni a fejlesztést. Az áttérést követően az eszköznél viszont egyből, akkor még leküzdhetetlennek tűnő szoftveres kompatibilitási problémákba futottunk. Ezt elkerülendő és a saját kódunk/munkánk ellenőrzése okából egy Raspberry PI 4B board-on folytatunk inkább munkánkat. Ezen igen gyorsan sikerült a kód implementáció Python nyelven. A Szoftver és a Hardver alapos tesztelése után figyelmünket ismét a Nano felé irányítottuk.

```
1. import time
2. import board
3. import busio
4. import adafruit_ads1x15.ads1115 as ADS
5. from adafruit_ads1x15.ads1x15 import Mode
6. from adafruit_ads1x15.analog_in import AnalogIn
7.
8. # Data collection setup
9. RATE = 860
10. SAMPLES = 1000
11.
12.
13. # Create the I2C bus
14. i2c = busio.I2C(board.SCL, board.SDA)
15.
16. # Create the ADC objects using the I2C bus
17. ads1 = ADS.ADS1115(i2c, address=0x48) # grd
18. ads2 = ADS.ADS1115(i2c, address=0x49) # vcc
19. ads3 = ADS.ADS1115(i2c, address=0x4A) # data
20. ads4 = ADS.ADS1115(i2c, address=0x4B) # clk
```

Teszt kimenet:

```
1. *****
2. Ért: 20884      poz: 0
3. Ért: 20715      poz: 1
4. Ért: 19851      poz: 2
5. Ért: 21207      poz: 3
6. Ért: 20817      poz: 4
7. Ért: 20722      poz: 5
8. Ért: 6845       poz: 6
9. Ért: 15722      poz: 7
10. Ért: 20722     poz: 8
11. Ért: 23857     poz: 9
12. Ért: 14867     poz: 10
13. Ért: 19867     poz: 11
14. Ért: 18722     poz: 12
15. Ért: 21302     poz: 13
16. Ért: 35243     poz: 14
17. Ért: 4845      poz: 15
18. *****
19. Max: 14 | Min: 6 | Mágnes Poz: 13 | Szög: -12 °
20. *****
```

Jetson Nano Kód:

Két hétnyi kutatás után megszületett a problémánkra a megoldás is, amivel a Raspberry-n már működő ADC könyvtárakat és kódot a Jetson Nano-n is működésre lehet bírni. Ehhez az alábbi parancsok szükségesek egy frissen telepített Nano imagen:

```
1. Ads 16 bit i2c adc install help on Jetson Nano
2.
3. sudo apt-get update
4. sudo apt-get upgrade
5. sudo apt-get install python3-pip
6. sudo apt-get install python3.8
7. sudo python3.8 -m pip install --upgrade setuptools
8. sudo python3.8 -m pip install nano
9. sudo /opt/nvidia/jetson-io/jetson-io.py
10. ls /dev/i2c* /dev/spi*
11. sudo i2cdetect -r -y 0
```

```

12. sudo i2cdetect -r -y 1
13. sudo groupadd -f -r gpio
14. sudo usermod -a -G gpio nano
15. cd ~
16. git clone https://github.com/NVIDIA/jetson-gpio.git
17. sudo cp ~/jetson-gpio/lib/python/Jetson/GPIO/99-gpio.rules /etc/udev/rules.d
18. sudo reboot
19. python3.8 -m pip install adafruit-blinka
20. python3.8 -m pip install --upgrade adafruit_blinka
21. python3.8 -m pip install adafruit-circuitpython-ads1x15

```

Ezeket a parancsokat pontosan és sorrendben követve (a Python verziót is!) lehet sikeresen előkészíteni egy Jetson Nano-t a szenzorhálózat használatára.

A 4 db ADC tisztán látszik a kimeneten:

```

1. nano@nano:/# i2cdetect -r -y 0
2.      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
3. 00:      -- -- -- -- -- -- -- -- -- -- -- -- -- --
4. 10: -- -- -- -- -- -- -- -- -- -- -- -- -- --
5. 20: -- -- -- -- -- -- -- -- -- -- -- -- -- --
6. 30: -- -- -- -- -- -- -- -- -- -- -- -- -- --
7. 40: -- -- -- -- -- -- -- -- 48 49 4a 4b -- -- -- --
8. 50: -- -- -- -- -- -- -- -- -- -- -- -- -- --
9. 60: -- -- -- -- -- -- -- -- -- -- -- -- -- --
10. 70: -- -- -- -- -- -- -- -- -- -- -- -- -- --

```

Kiszervezés és integrálás a többi szenzorral

Miután megbizonyosodtunk arról, hogy mind a hardveres, mind a szoftveres rész is jól működik, megkezdjük a szenzorunk integrálását az autó már meglévő szenzorhálózatával. Ez hardver szinten azt jelentette hogy a megfelelő helyen kell elhelyezni az ADC-ket és azok csatlakozóit a Nano GPIO-án. Szoftveresen pedig azt hogy szenzorunk kódját egy könyvtárba kiszerveztük így azt a fő vezérlő programnak csak meg kell hívni szükség szerint.

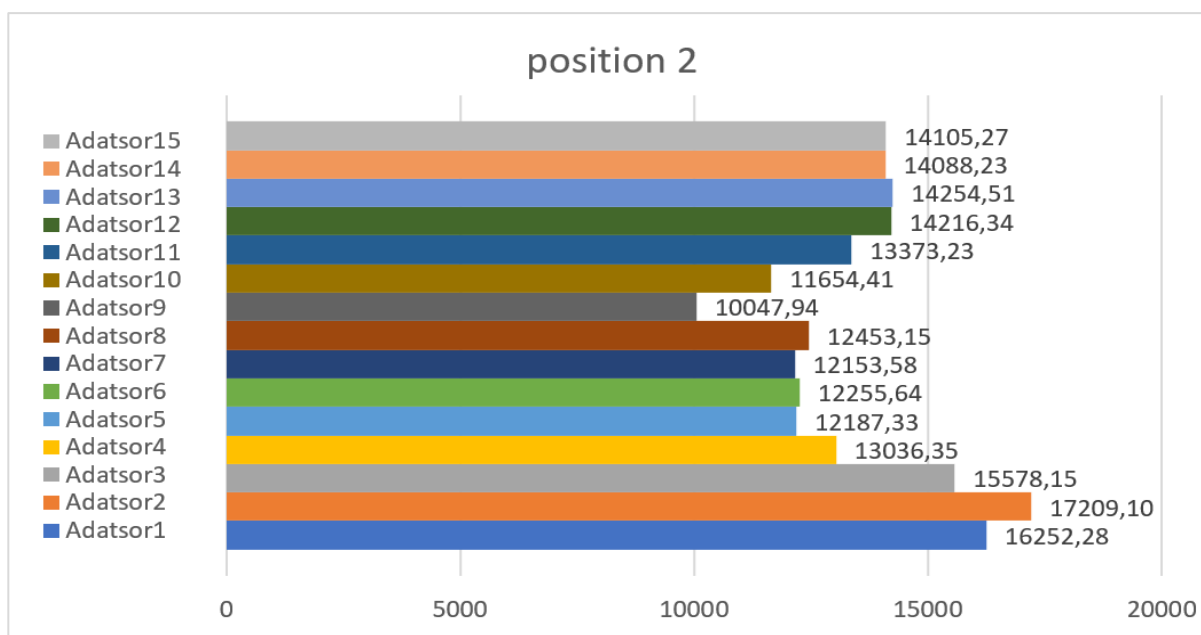
Python teszt kód:

```
1. from test_class import SteeringSensor as sensor
2.
3. def main():
4.     sensor1 = sensor
5.     print(sensor1.angle(),sensor1.magnet_poz(),sensor1.min_val(),sensor1.max_val())
```

2.2.11. Eredmények

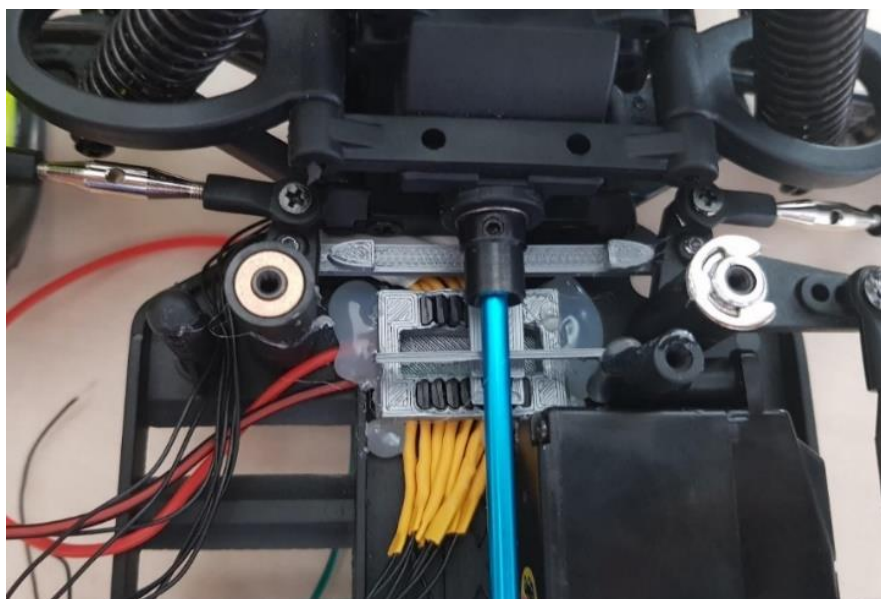
A sikeres koncepció kialakítása után készen álltunk az adatok gyűjtésére. Erre a célra egy Arduino Uno fejlesztőpanelbe vezettük a kimeneteket. A 4 darab ADC a beljük vezetett szenzorjeleket egy úgynevezett I2C kommunikációs szabvány segítségével képesek továbbítani a feldolgozó egység felé. Az említett Arduino eszköz is rendelkezik ilyennel beépítve, ami segítségével a kommunikáció zavartalanul végbe mehet a soros porton. Mindezek monitorozása/megfigyelése viszont nem triviális feladat. Külön program segítségét kellett igénybe vennünk hozzá, ennek a szoftvernek a neve pedig Putty. Ezt használva egy .txt kiterjesztésű fájlba könnyen elmentettük a küldött adatokat. Később ezeket, mint CSV fájlokat be tudjuk importálni egy Excel dokumentumba. A 15 szenzor által szolgáltatott 14 lehetséges

állapot mindegyikét egy-egy Excel táblázatba vezettük fel, és azokat vizualizáltuk is egy-egy diagrammal, mint azt a (22. ábra) is szépen mutatja.



22. ábra A 2. pozícióban gyűjtött adatok

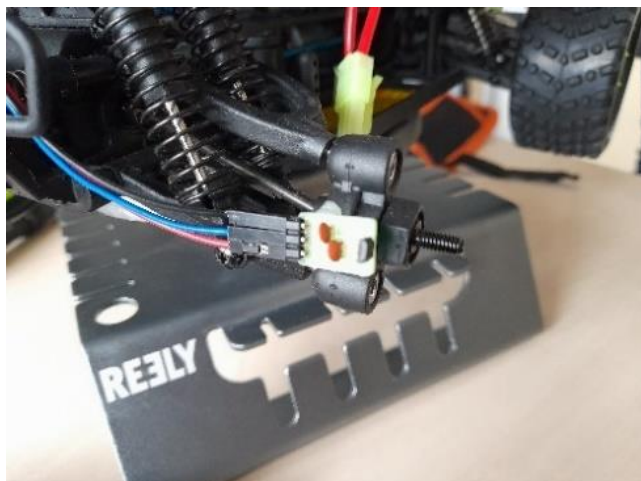
Pozíciónként közel 200 sornyi, azaz majdnem 3000 db egyéni szenzor mérési adattal rendelkezünk. Az összes pozíciót tekintve tehát 42.000 Hall szenzor mérést végeztünk el a szenzorhálózat tesztelése során. A gyűjtött eredmények nagyon biztatóak, ugyanis minden pozícióban a maximum és minimum értékek kellőképpen és tisztán látható módon elkülönülnek egymástól. Így a mi számításaink szerint nagyon pontos, körülbelül $\sim 2^\circ$ -os szögelfordulás mérhető az általunk készített eszközzel. Mindez úgy lehetséges, hogy a kerekek jobbra és balra is $15-15^\circ$ -ot képesek elfordulni. Az általunk használt szenzorrendszer pedig 14 pozíciót képes pontosan detektálni. A teljes 30° -os szögelfordulást pedig a pozíciókra vetítve könnyen megkapható a körülbelüli 2° -os ($2,1429^\circ$) pontosság. Fő célként, már a bevezetésben említett ennél jelentősen nagyobb, $4-5^\circ$ -os (~ 6 db szenzor) pontosságú elfordulás detektálását tűztük ki. Ez azért is jelentős sikernek számíthat mivel maga a modellautó irányítása, kormányzó aktuátorai nem túl pontosak, legalább is közel sem ennyire. Tehát a mi mérő eszközünk pontosabban képes mérni, mint az autó irányítani, korrigálni saját magát. Így az autó kerekeinek esetleges nemkívánt Hardveres vagy Szoftveres okból kifolyó kerék elfordulását vagy nem (eléggé) elfordulását is képesek vagyunk mérni. Kutatásunkról ezért úgy gondoljuk, hogy sikeresnek és eredményesnek bizonyult.



23. ábra Végző megoldás

2.3 Sebességmérés

A különböző autógyárak egymástól eltérő, merőben különböző megközelítéseket használnak a sebesség mérésére. Bármilyen szenzor megtervezésének [12][14][15] a legfontosabb lépése az, hogy megtaláljuk a kapcsolatot a mérendő érték és a méréshez használt jelenség között. Jó példa erre az autóiparban használt elektromos indukción alapuló sebességmérő. Ekkor az autó meghajt egy feszültség-generátor, amely annál nagyobb feszültséget állít elő, minél nagyobb a sebesség. Ez egy elegáns és nagyon egyszerű módja a sebességmérésének, viszont nem használható egy modellautó sebességmérésére, mivel nem képes meghajtani az eszközt, illetve ezek a megközelítések az autó kardántengelyhez erősítettek. Számunkra ez nem megfelelő, mivel a bővebb felhasználás érdekében sokkal hatékonyabb és hasznosabb, ha a sebesség mérését mi kerekenként oldjuk meg, (lásd 24. ábra).



24. ábra Felfüggesztésre erősített sebességmérő szenzor

2.3.1 A sebesség

Az mozgó járműveket jellemző egyik legfontosabb mennyiség a sebesség. Ez a mennyiség, amely megmutatja számunkra, hogy egy mozgást végző objektum egy egységnyi idő alatt mekkora távolságot tesz meg. A balesetek elkerülése érdekében megszabnak egy maximális haladási sebességet, ezért a legriválisabb felhasználási módja egy sebességmérő szenzornak az, hogy megállapítsuk az autó haladási sebességét.

$$v = \frac{s}{t}$$

ahol s a megfigyelés ideje alatt megtett út, t a megfigyelés időtartalma és v a sebesség. Rendszerint ez a mért érték kilométer/órában van megjelenítve. A fizikai képletekkel való számoláshoz az SI mértékegységrendszer szerinti méter/szekundumban fogjuk használni. kilométer/órából át tudjuk váltani méter/szekundumba triviális módon 3.6-al történő osztással.

Fizikai szempontból a sebesség a pozíciófüggvény idő szerinti első deriváltja. A pozíció függvény egy pontszerű test térbeli elhelyezkedéséről ad nekünk információt, mekkora távolságra van a nullponttól és egy adott inercia rendszerben a t -k időpillanatban megfigyelés kezdetétől tekintve.

$$v(t) = s'(t) = \frac{ds}{dt} = \lim_{t \rightarrow t_0} \frac{s(t) - s(t_0)}{t - t_0}$$

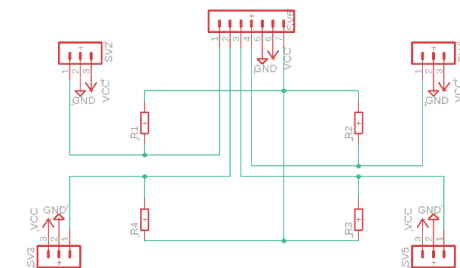
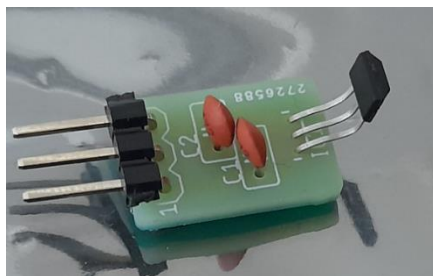
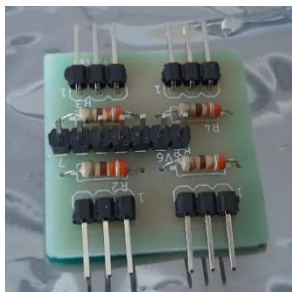
A sebesség egy vektormennyiségként is kezelhető. Ekkor nem csak arról kapunk információt, hogy mekkora utat tesz meg egységnyi idő alatt a test, hanem a test haladási irányáról is.

Modellautók esetén a legegyszerűbben kivitelezhető és a legkevesebb mérési hibát okozó megközelítés egy sebességmérő szenzor szerintünk egy fordulatszám-mérő eszközön alapuló eszköz [11]. A fordulatszám megmutatja, hogy egységnyi idő alatt hány fordulatot tesz meg egy forgó eszköz. A fordulatszám nem más, mint a frekvencia. A frekvencia az a mennyiség, ami a vizsgált periodikus jelenség ismétlődési gyakoriságát jellemzi, azaz egységnyi idő alatt hányszor történik meg az adott esemény.

$$\varphi = \frac{1}{T}$$

ahol T a periódusidő. A periódusidő az az időtartam, amely alatt egy ismétlődés megtörténik.

Ezt a szenzort mi alapos meggondolás után Hall-effektusra alapozva készítettük el. A hagyományos megközelítéssel szemben, mi megmérjük azt, hogy mekkora t idő alatt tesz a kerék meg egy fordulatot. Ez számunkra azt jelenti, hogy az autó megtett egy s kerék-kerületnyi utat egy t idő alatt, ezáltal a sebesség már kiszámolható. Ebből már elég pontosan és megbízhatóan tudunk sebességet számolni. Ez a megoldás előnyösebb is mivel, nem azt mérjük, hogy hányszor fordult meg a kerék egy másodperc alatt, hanem azt, hogy mennyi idő telt el két észlelés között, amellyel kritikus helyzetekben gyorsabb reagálást tudunk elérni.



25. ábra Sebességmérő szenzor és kapcsolási rajza

Egy kapcsoló üzemi TLE4905L hallszenzort alkalmaztunk, melyet egy egyedi tervezésű áramkörre rögzítettünk, valamint ehhez csatlakozópanelt terveztünk. Az áramkör kialakításának köszönhetően a szenzor az szűrt. A szenzort tartalmazó nyomtatott áramkört az autó felfüggesztésének nem forgó részeire rögzítettük.

Ha a mágnes elhalad forgás közben a szenzor előtt, akkor a szenzor egy megszakítást idéz elő a rendszeren belül, mely során megmérjük azt, hogy mennyi idő telt el két megszakítás között. Ezt a jelet a Jetson Nano GPIO (Általános célú ki- bemeneti csatlakozó) csatlakozóin olvassa

be. A mágneseket a kerekre helyeztük fel, amelyhez egy 3D nyomtatott tartókorongot terveztünk.



26. ábra Mágnes kerekre erősítése 3D nyomtatott elemekkel

Kezdetben egy mágnest rögzítettünk a kerekre, de beláttuk, hogy több mágnes esetén pontosabb rendszert kaphatunk, mivel két mérés között rövidebb idő fog eltelni. n darab mágnes esetén a pontosított sebességmérés

$$v = \frac{s}{\frac{n}{t}}$$

Mi az ideális mágnesszámot 2-nek állapítottuk meg ezen a modellautón. Több mágnes jelenleg az autó stabilitását és haladási képességét korlátozhatja.

Python kódrészlet a sebességmérő szenzorhoz:

```
1. import Jetson.GPIO as GPIO
2. import time
3. import math
4.
5. class SpeedSensor:
6.     def __init__(self, pin_1=None, pin_2=None, pin_3=None, pin_4=None, diameter =
7.     8, stop_time=2000):
8.         if pin_1==None or pin_2==None or pin_3==None or pin_4==None:
9.             raise Exception("Set up every pin for speedometer")
10.         self.pin_1_time = time.time()*1000
11.         self.pin_2_time = time.time()*1000
12.         self.pin_3_time = time.time()*1000
13.         self.pin_4_time = time.time()*1000
14.         self.stop_time = stop_time
15.         self.s = (diameter/100)*math.pi
16.         self.v1 = 0
17.         self.v2 = 0
18.         self.v3 = 0
19.         self.v4 = 0
```

```

20.
21.     GPIO.setmode(GPIO.BOARD)
22.     GPIO.setup(pin_1, GPIO.IN)
23.     GPIO.setup(pin_2, GPIO.IN)
24.     GPIO.setup(pin_3, GPIO.IN)
25.     GPIO.setup(pin_4, GPIO.IN)
26.
27.     GPIO.add_event_detect(pin_1,     GPIO.FALLING,     callback=self.handle_pin_1,
    bouncetime=10)
28.     GPIO.add_event_detect(pin_2,     GPIO.FALLING,     callback=self.handle_pin_2,
    bouncetime=10)
29.     GPIO.add_event_detect(pin_3,     GPIO.FALLING,     callback=self.handle_pin_3,
    bouncetime=10)
30.     GPIO.add_event_detect(pin_4,     GPIO.FALLING,     callback=self.handle_pin_4,
    bouncetime=10)
31.
32.     def handle_pin_1(self,channel):
33.         d = time.time()*1000 - self.pin_1_time
34.         self.v1 = self.s/(d/1000)
35.         self.pin_1_time = time.time()*1000
36.
37.     def handle_pin_2(self,channel):
38.         d = time.time()*1000 - self.pin_2_time
39.         self.v2 = self.s/(d/1000)
40.         self.pin_2_time = time.time()*1000
41.
42.     def handle_pin_3(self,channel):
43.         d = time.time()*1000 - self.pin_3_time
44.         self.v3 = self.s/(d/1000)
45.         self.pin_3_time = time.time()*1000
46.
47.     def handle_pin_4(self,channel):
48.         d = time.time()*1000 - self.pin_4_time
49.         self.v4 = self.s/(d/1000)
50.         self.pin_4_time = time.time()*1000
51.
52.     def run(self):
53.         if (time.time()*1000 - self.pin_1_time >self.stop_time):
54.             self.v1 = 0
55.         if (time.time()*1000 - self.pin_2_time >self.stop_time):
56.             self.v2 = 0
57.         if (time.time()*1000 - self.pin_3_time >self.stop_time):
58.             self.v3 = 0
59.         if (time.time()*1000 - self.pin_4_time >self.stop_time):
60.             self.v4 = 0
61.         return self.v1,self.v2,self.v3,self.v4
62.
63.     def run_threaded(self):
64.         if (time.time()*1000 - self.pin_1_time >self.stop_time):
65.             self.v1 = 0
66.         if (time.time()*1000 - self.pin_2_time >self.stop_time):
67.             self.v2 = 0
68.         if (time.time()*1000 - self.pin_3_time >self.stop_time):
69.             self.v3 = 0
70.         if (time.time()*1000 - self.pin_4_time >self.stop_time):
71.             self.v4 = 0
72.         return self.v1,self.v2,self.v3,self.v4
73.
74.     def shutdown(self):
75.         GPIO.cleanup()

```

2.3.1 Fizikai Alapok

Egy jól megtervezett szenzort nem csak egy adat mérésére tudunk felhasználni, hanem számos más fizikai törvényeken alapuló felhasználási módot tudunk hozzájuk találni. A különböző mozgásformák ismeretében pontosan kiszámolható a rendszerre jellemző mérőszámok. A kerekenként történő sebességméréssel ez még pontosabbá tehető. Ami számunkra és a mindennapi életben kifejezetten hasznos. Önvezető járművek modell autóval történő fejlesztése során számos kritikus szélsőállapot felismerését teheti lehetővé, ha a fizikai rendszerre gyakorolt erők és a különböző a rendszert tápláló impulzusokat mérni tudjuk, viszont ehhez szükségünk van a sebesség ismeretére. Ezzel el tudjuk érni, hogy az autó viselkedése is emberszerűbb legyen, nem melleleg kisebb változást is hamarabb képes detektálni, így könnyebben tud végül baleseteket megelőzni.

2.3.2 Mozgásformák

A járművek alapfeladata, hogy egy megadott pontból egy másik pontba eljussanak. Ennek során állapotváltozás következik be. Minden ilyen mozgás leírása során inercia rendszerekben gondolkodunk. Ezek a rendszerek olyan rendszerek, amelyekben teljesül Newton I. törvénye. A valós fizikai környezetben a tehetetlenségi törvény alapján, egy mozgó test mindaddig nem veszít a sebességéből és ugyanabban az állapotban marad, amíg valamilyen hatás ennek megváltoztatására nem kényszeríti. Autók esetén legtöbbször fékező hatásokra kell gondolni. Ilyen fékező hatás tud kifejteni a légellenállás is vagy amikor az autót vezető sofőr fékez. A sebességet befolyásolja még az autót alkotó mechanikai berendezések is, mivel ekkor is keletkezik átviteli veszteség. A hatás-ellenhatás törvény okán elhanyagolható mértékben az áttétek passzívan ellenálnak és a mozgással ellentétes erőt fejtenek ki, tehát a jármű lassulni kezd.

Azt a folyamatot, amikor egy inercia rendszerben egy megfigyelt test pozíciója megváltozik mozgásnak nevezzük és azt mondjuk, hogy a megfigyelt test mozgást végzett. Az objektum ezen folyamat során egy mozgási görbén vagy pályán halad. A pálya azon része, amit egy t ideig történő megfigyelés alatt tekintünk, test által t idő alatt megtett útnak nevezzük. Autók szempontjából a két fő mozgásfajtát különböztetünk meg: Egyenes vonalú- és körmozgás.

Számos esetben a haladó autó tekinthető egy pontszerű testnek is a számítások csökkentése érdekében. Ekkor a testre ható vektorok eredőjével kell számolnunk, mivel az autó egy egységes rendszerként kezelhető, a tagjai nem függetlenek egymástól. Az egyes oldalanként

fellépő különböző mérőszámok kiszámolására az autó geometriai formájából adódó következtetések segítségével megállapíthatóak, erre azonban ritkán van szükség.

Egyenes vonalú mozgás során a test mozgásvektora minden időpillanatban ugyanabba az irányba mutat. Ezen belül megkülönböztetünk egyenletes és változó mozgást. Egyenletes mozgás esetén a test pozícióváltozása egyenletes, azaz a test egy konstans sebességgel halad, a mozgásvektor nagysága és iránya állandó. A rendszert tápláló impulzus is állandó.

Változó mozgás esetén nem áll fent az egyenletesség, azaz a test gyorsul vagy lassul, tehát nem egyenletesen változik a pozíciója, vagyis a test minden időpillanatban más sebességgel halad, de ugyanabba az irányba mutat a mozgásvektor. Technikailag nem teszünk különbséget gyorsulás és lassulás között, és csak a gyorsulásvektorok irányából tudunk következtetni arra, hogy éppen a test gyorsuló vagy lassuló mozgást végez. Ha a mozgás irányával ellentétes a gyorsulásvektor iránya, akkor a test lassul, míg ellenkező esetben gyorsul. Ekkor a testet mozgásra készítő impulzus folyamatosan változik.

Körmozgás esetén a test nem egyenes pályán halad, vagyis a mozgásvektor iránya eltérő, mint az megelőző időpillanatban. Ebben az esetben célszerű későbbi alkalmazások hasznosítása érdekében nem pontszerű testként kezelni az autót. Számítások során pontszerűként kezeljük és a geometriai jellemzőket figyelembe véve hajtjuk végre a műveleteket. A különböző oldalakra vetített következtetéseket az autó alaktani jellemzői alapján vonjuk le. Az autó két oldalának elfordulási szöge, vagy szögfordulása mindkét oldalon megegyezik, míg az általuk megtett ív hossza el fog térni. Lehetséges, hogy a helyváltozás során nem fog egy teljes kört leírni a test pályája, illetve az alakja nem fog egy körre hasonlítani, de tekintett időpillanatbéli állapot jellemezhető a szabályos körmozgásra jellemző törvényeknek a segítségével. Ekkor úgy tekintünk a rendszerre, hogy a test olyan körpályán halad, amelynek sugara változik, valamint az ív is eltérő lehet, mint a megelőző pillanatban.

2.3.4 A sebesség felhasználása

A sebességből származtatható a gyorsulás, amely a pozíciófüggvény második idő szerinti deriváltja, azaz a sebesség elsőrendű deriváltja. Ez a szám a sebességváltozás nagyságáról ad nekünk információt, egységnyi idő alatt mekkora mértékben változik meg a sebesség. Formálisan felírva a gyorsulás

$$a = \frac{v_1 - v_0}{t_1 - t_0}$$

ahol v_1 a t_1 és v_0 a t_0 időpillanatban mért sebessége a mozgó járműnek. Ezt fel tudjuk használni ahhoz, hogy kiszámoljuk a rendszerre ható erőket, illetve protokollok használatáról döntsünk kritikus helyzetekben.

Fontos mérőszám a lendület, mely megmutatja számunkra, hogy I impulzus milyen mértékben változtatja meg a rendszerünk állapotát, azaz mekkora mozgásmennyiséget képes egy külső behatás átadni a rendszerünknek. Autók esetén nem lesz explicit külső behatás, mivel a motor hatására fog keletkezni az állapotváltozás, de ugyanazok a mechanikai törvények segítségével jellemezhető, mint külső hatás esetén. Rendelkezik iránnyal is, amely megmutatja, hogy a beható impulzusvektor mely irányba mozdítja az autót el. Ez egy inercia rendszeren belül a lendület zárt rendszert alkot, tehát a lendületek vektoriális összege időben állandó lesz. Egy inercia rendszeren belül a lendület nem vész el, hanem átadódik a környezetébe, illetve azoknak az objektumoknak, amelyekkel érintkezik. A lendület egyenesen arányos a test tömegével és annak haladási sebességével.

$$I = m * v$$

$$I(t) = m * v(t)$$

ahol, az m a test tömege és v a haladási sebessége. A viselkedés alapú önvezető modellek számára fontos lehet az impulzus nagyságának meghatározása az egyes körülményekhez történő alkalmazkodáshoz, illetve az őt érő erőhatások modellezésében.

Az erő az a fizikai mennyiség, amely megmutatja az impulzusváltozás gyorsaságát, azaz egységnyi idő mekkora mennyiséggel nő az impulzus egy rendszeren belül. Ha ez a szám nem 0 akkor azt jelenti, hogy a test előjeles gyorsuló mozgást végez (lassít vagy gyorsít), amely triviális, hiszen megváltozik az a behatás, amely okán a test mozogni kezdett. Ekkor azt mondjuk, hogy a test gyorsul F erő hatására,

$$F = \frac{I_1 - I_0}{t_1 - t_0} = m * a$$

Körmozgás esetén fellépnek különféle erők. A két legfontosabb ilyen erő a centripetális és a centrifugális által kifejtett erőhatások. A centripetális erő iránya mindig a körpálya közepe felé mutat, míg a centrifugális ezzel ellentétes irányú, azaz le akarja taszítani a pályáról a mozgó testet. A test körpályán való mozgásba kezd akkor, ha ez a két erő egyforma. Abban az esetben, ha a centrifugális erő nagyobb, mint a centripetális erő, akkor a test kisodródik a pályáról.

A testre ható centrifugális erő

$$F = m \frac{v^2}{r} = m \cdot a_c$$

ahol

$$a_c = \frac{v^2}{r}$$

a körmozgás során fellépő centrifugális gyorsulás, ahol v a test körpályán történő haladási sebessége és r a körpályához tartozó sugár.

A centripetális erő nagysága autók esetén megegyezik a jármű által kifejtett súrlódási erővel és az autó oldala által támasztott légellenállással

$$F_{cp} = m \cdot g \cdot \mu + A \cdot k \cdot \varrho \cdot v_{cf}^2$$

ahol m a test tömege, g a gravitációs gyorsulás, μ az adott fizikai útszakaszon mért tapadási-súrlódási együttható, A az autó oldalának felszíne, k az alaktani állandója, ϱ a közeg sűrűsége és v_{cf} az autó oldalirányú sebessége, melyet a centrifugális gyorsulás idéz elő. A μ egy konstans szám, mely autóként eltérhet. Függ attól, hogy a kerék mekkora felületen érintkezik aszfalttal. Eltérhet ez az állandó akkor, ha az út vizes vagy olajos, de a gumi összetétele is befolyásolja.

$$\mu = \frac{F_{húzó}}{F_{nyomó}}$$

Ennek ismeretében meghatározhatjuk azt a maximális sebességet, amely esetén még az autó a körpályán haladni tud biztonságosan. Ekkor a centripetális erőnek nagyobbak kell lennie, mint a centrifugálisnak, azaz

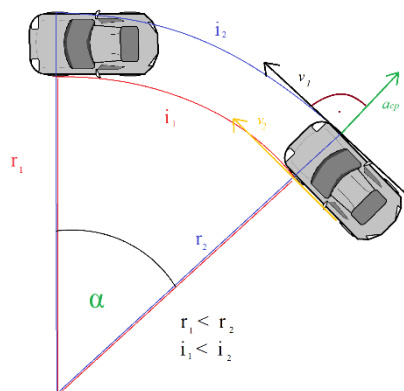
$$m \frac{v^2}{r} \leq m \cdot g \cdot \mu + A \cdot k \cdot \varrho \cdot v_{cf}^2$$

Innen kifejezve látható, hogy

$$v \leq \sqrt{\frac{m \cdot g \cdot \mu + A \cdot k \cdot \varrho \cdot v_{cf}^2}{m} \cdot r}$$

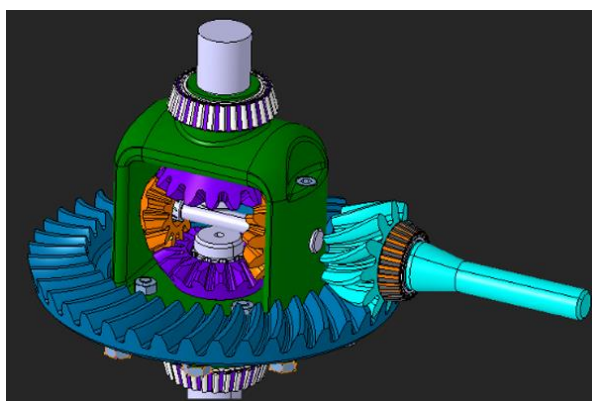
sebességgel kell haladni az autónknak egy kanyarban ahhoz, hogy ne sodródjon le az aktuális körpályáról és biztonságosan haladjon tovább.

Az autók a legtöbbször nem egyenes vonalon mozognak, hanem valamilyen összetett körmozgást végeznek. Ennek során a különböző kerekek különböző nagyságú sugarú körön mozognak és eltérő távolságot tesznek meg. Ennek okán belátható, hogy a négy kerék különböző forgási sebességgel rendelkezhetnek.



27. ábra Körmozgás szemléltetése

Kezdetben a kerekek egy merev rúddal voltak összekötve, amely egy ilyen helyzetben felesleges terhelést kap és számos kockázati tényezőt jelent. A tengelyekre akár akkora erőhatás terhelődhet relatív kis sebesség esetén is, hogy mozgás közben eltörhet, emellett az autókönnyebben meg tud csúszni az egyik hajtott oldalon, ezáltal elvesztve az irányítást a jármű felett. Az autóiparok annak érdekében, hogy a tengelyre kifejtett erőhatásokat és a megcsúszás valószínűségét csökkentsék kifejlesztették a differenciálművet, melynek célja, hogy lehetővé tegye a különböző kereket eltérő sebességgel mozogjanak.



28. ábra Differenciálmű

A (28. ábra)-n látható narancssárga fogaskerekek fogják korrigálni a különböző kerékfordulatok közötti eltérést. A fogaskerekek méretezéséből adódóan egyenesvonalú

mozgás esetén merev rúdként fog viselkedni. Eltérő kerék fordulatszámok esetén ezek a fogaskerekek elmozdulnak, így a passzívan fékeztetett kerék képes lassabb sebességgel forogni.

A differenciálmű működéséből kifolyólag tudjuk, hogy képesek a kerekek különböző forgási sebességgel forogni. Körmozgás esetén kerekenként történő sebességméréssel meg ki tudjuk számolni, hogy mekkora ívet tesz meg az autó két oldala. Tudjuk, hogy mekkora sebességgel haladt az autó és tudjuk azt is, hogy mennyi ideig mértük a sebességet. Ezeknek az értékeknek az ismeretében a sebesség kiszámítására használható képletet átrendezve

$$v = \frac{s}{t}$$

$$s = v * t$$

$$s = i = \frac{r\pi}{180} \alpha$$

azonosságot kapjuk, amely a jelen helyzetben az autó által megtett ív hosszát fogja számunkra megadni. A különböző oldalakra felírva ezt az egyenletet

$$\begin{cases} i_1 = \frac{r_1\pi}{180} \alpha_1 = v_1 t \\ i_2 = \frac{r_2\pi}{180} \alpha_2 = v_2 t \end{cases}$$

egyenletrendszer kapjuk. Ekkor viszont oldalanként

$$\begin{cases} a_{c1} = \frac{v_1^2}{r_1} \\ a_{c2} = \frac{v_2^2}{r_2} \end{cases}$$

ahol a_{c1} és a_{c2} a két oldalon fellépő centrifugális gyorsulás, melyet mérni tudunk oldalanként egy giroszkóp szenzor segítségével. A két giroszkóp szenzor használatával pontosabb eredményt kaphatunk. Ennek a gyorsulásnak mérése szükséges, mivel a körmozgás során fellépő centrifugális gyorsulás egyértelmű módon nem megállapítható. Ezzel a négy egyenlettel α_1 és α_2 , valamint r_1 és r_2 szögek és sugarak kiszámolhatóak.

A két oldal külön kezelésével ideális esetben

$$| \alpha_1 - \alpha_2 | \leq \delta$$

amely azt jelenti, hogy egy δ hibahatár között van a két oldal eltérése (a szenzorok pontatlansága miatt). Ha ennél nagyobb számot kapunk eredményül, akkor a két oldal felfüggesztésében valamely eszköz meghibásodott.

Két jelentős fogalom van még, amit mozgó testekkel kapcsolatban meg kell említenünk. Az egyik az energia a másik a munka. Az energia nem más, a testek azon jellemzője, mely megmutatja, hogy a tekintett objektum mekkora állapotváltoztató képességgel rendelkezik. Az energiát a test valamilyen kölcsönhatás segítségével tudja megszerezni, mivel energia nem tud keletkezni sem elveszni, hanem valamilyen másik energiaformába átalakul, mértékegysége a Joule(J). Ahhoz, hogy egy rendszerbe energiát tápláljunk, valamilyen mezőnek vagy kölcsönhatásnak kapcsolatba kell lépnie vele és F erőt kell gyakorolnia a testre, valamint ennek az erőnek a hatására test mozgásra kényszerül. Ekkor azt mondjuk, hogy F erő munkát végzett a testen. A munka az a mennyiség, amely megmutatja, hogy egy tetszőleges mező vagy kölcsönhatás által kifejtett erő, hány Joule energiát táplált egy rendszerbe.

$$W = F * s$$

ahol W az elvégzett munka, F a testre gyakorolt erő és s az erő hatására megtett út. A definíciókból adódik, hogy az elvégzett munka mennyisége lesz a test adott helyzetbeli energia mennyisége.

Amikor egy autó gyorsuló mozgást végez és egy v sebességet el szeretne érni, akkor a motornak munkát kell végeznie. Ez egy valós fizikai rendszerben veszteséges lesz, egyrésze hő formájában távozik, a másik része az áttételnél fog másik energiaállapotba alakulni. Amikor a test eléri a kívánt v sebességet, akkor az autó úgynevezett v sebességbeli mozgási energiája ekkor

$$W = F * s = m * a * \frac{a}{2} * t^2$$

ahol a megtett s út az a gyorsulás alatt megtett út (idő szerinti integrálással triviális módon megkapott pozíció függvénye)

$$s = a * \frac{a}{2} * t^2 + v * t + s$$

ahol $v*t$ azt a megtett utat jelöli, amit a test egyenes vonalú egyenletes mozgással megtett, és s az alap távolság, amit az autó az inercia rendszer kezdőállapotától számunkra ismeretlen

mozgási folyamat során megtett. Ez a két érték számunkra most 0, mivel mi csak a gyorsulást ideje alatt megtett úttal foglalkozunk.

$$W = m * a * \frac{a}{2} * t^2 = \frac{m * a^2 * t^2}{2} = \frac{m * v^2}{2}$$

Ez az energiamennyiség a tényleges érték, amivel az autó rendelkezni fog. Ebből már a veszteségek is levonásra kerültek.

Számunkra ez az érték kifejezetten hasznos, mivel ennek és a motor által végzett veszteségmentes munka (az áttétek nélkül végzett munka) ismeretében meg tudjuk állapítani azt, hogy az autók mechanikai alkatrészei, amelyek a hajtásért felelnek megfelelő állapotban vannak-e vagy sem. A motor által leadott munkát az autón ki tudjuk számolni a motor adottságainak mérésével (a modell autó motorjának méretbeli kialakítása ezt nem teszi lehetővé ezt). Ekkor a hajtási rendszer hatásfoka

$$H = \frac{W_{\text{átvitt}}}{W_{\text{motor}}}$$

ahol $W_{\text{átvitt}}$ az autóra gyakorolt munka, vagyis az autó mozgási energiája v sebesség esetén, és W_{motor} a motor munkája. Ha ez az érték 1 akkor az áttétek által okozott veszteség 0, de ilyen rendszer nem létezik. Egy meghatározott H_0 hatásfok esetén, ahol a mechanikai rendszer megfelelő állapotú, ha a hiba egy ε határon belül mozog akkor nincs meghibásodás.

$$|H_0 - H| \leq \varepsilon$$

Ezekből az egyenletekből egyértelműen látszik, hogy szükséges a kerekenként történő sebességmérés. Szenzoradatok ellenőrizhetőek és számos a rendszer szempontjából lényeges mérőszámok állapíthatóak meg velük a fent említett módokon. Ezen felül biztonságos vezetést támogató alkalmazások is készíthetőek velük.

Magas sebességgel történő haladás során nagy erővel történő hirtelen fékez esetén, a kerekek könnyedén a teljesen megállhatnak, de az autó viszont tovább mozoghat. Ekkor azt mondjuk, hogy a kerekek blokkoltak. A test a mozgása során elért sebességről ekkor túl nagy gyorsulással próbál lassuló mozgást végezni és fellépő erőhatás nagyobb, az autót egy stabil pályán tartó tapadási erő. A mozgó rendszer instabillás válik, mely eredménye az lehet, hogy az autó megcsúszik. Ennek könnyen baleset lehet a vége, ilyenkor a tapasztalt sofőröknek is nehezebb lehet az, hogy újra stabil állapotba hozzák az autót. A kerekenként történő sebességméréssel

detektálni tudjuk ezt a szituációt és automatikusan korrigálni tudjuk a kifejtett fékezőerőnek a csökkentésével. Ezt a rendszert nevezik ABS-nek (Anti-lock Braking System).

Egy másik szintén hasznos felhasználási módja ennek a kipörgés gátló. Ha a rendszer detektálja azt, hogy valamelyik oldalt a túlzottan nagy eltérés mutatkozik meg a forgási sebességek között, akkor az azt jelenti számunkra, hogy a kerekek kipörögnek és az autó nem fut stabilan. Ekkor a kipörögő kerekeket fékezhethetjük.

3. Összefoglalás

Az önvezető autók fejlesztésének alapköve az aktuátorrendszer és környezet diverzifikált módon történő megfigyelését lehetővé tevő szenzorhálózat kidolgozása és az ebből származó adatok elemzése. A modell autók alapját képező alváz passzív rendszernek tekinthető abban az értelemben, hogy nincsenek integrált szenzorai. Ezen túlmenően a piacon sehol nem találtunk a számunkra megfelelő méretű, célú és árú sebességmérő, illetve kerék elfordulás mérő szenzorokat. Viszont mivel „Dave”-nek szüksége volt ezekre a szenzorokra így mi magunk kezdtük meg ezek megtervezését és legyártását. Ezek a megvalósítások lehetővé teszik egy komplexebb rendszer pontos modelljének kidolgozását is. A munka során a számtalan nehézség közül a helyhiány okozta a legnagyobb kihívást. Ki kellett találnunk, hogy hogyan lehetünk képesek az autó vázába rögzíteni mind a kerékszög mérőt, mind a keréksebesség mérőt, emiatt számtalanszor át kellett dolgozzuk az elkészült szenzorokat és azok programjait.

A tervezés részeredményei, feltételezett jövőbeni körülmények és a várható környezeti terhelések alapján a Hall szenzoros megoldás mellett döntöttünk. Mivel ezzel a szenzorral súrlódás mentesen képesek vagyunk pontos adatokat kapni a fényviszonyoktól és a hőmérséklettől függetlenül. Ez egy érintésen vagy egy fényen alapuló szenzorral nem lenne lehetséges. Az érintés alapú szenzor a súrlódás miatt elkopna ráadásul ilyen kicsi modell méretnél a súrlódás a kanyarodást is befolyásolhatná, akadályozhatná. A fény alapúak pedig erős napsütésben vagy kontrasztos, árnyékos helyeken könnyen pontatlan adatokat adhatnak nem beszélve a hely igényeiről, illetve az áráról. Ezekkel ellentétben a Hall szenzorok olcsók, helytakarékosak és súrlódás mentesek. Miután a szenzort kiválasztottuk azt is el kellett dönteni, hogy ezt a szenzort hogyan használjuk. A sok mágnes, egy szenzor és a 3 szenzor 3 mágnes koncepció elvetése után a sok szenzor 1 mágnes koncepció mellett döntöttünk. Ebben a konfigurációban egy mágneset erősítünk a keresztengelyhez, ami összeköti a két első kereket. Ez a mágnes pedig egy sornyi mágnes előtt elhalad. Ahhoz viszont, hogy ezt megtudjuk valósítani saját rudat és szenzor tartót kell 3D terveznünk és nyomtatnunk.

A sebességmérő szenzorból származó adatokkal számos mérőszámot meg tudunk állapítani, melyek hasznosak a biztonságos vezetéshez. Ezekkel az értékekkel meg tudjuk becsülni az autó viselkedését, következtetéseket tudunk levonni arra vonatkozóan, hogy az autó belső szerkezete milyen állapotban van.

Mivel mind a tartó, mind a tengely és maga az egész koncepció is teljesen egyedi design, így nekünk kellett mindent az elméleti kidolgozástól a 3D-s tervezésen keresztül a prototípus gyártáson és tesztelésig bezárólag megcsinálni. Ez rengeteg 3D-s modell újra tervezéssel, finomítással járt és nagy mennyiségű prototípus verziók keletkeztek, amíg el nem jutottunk a tökéletes konfigurációhoz (lásd 29. ábra). Mire eljutottunk az ideális konfigurációhoz körülbelül két tucat különböző prototípus készült el és legalább kétszer annyi ki-nem nyomtatott 3D-s terv, valamint nagyból egy tucat kereszt tengely 3D-s terv és fél tucat prototípus tengely.



29. ábra Kinyomtatott 3D prototípusok generációi

A sebességmérő szenzort elsőnek egy Arduino Uno panel segítségével valósítottuk meg. A jelenlegi autón egy már második verzió található meg. Ennek kialakításának oka a főleg a helyhiány volt, mivel az Arduino-t egy biztonsági vázba helyeztük el, és ezt megfelelő módon nem tudtuk rögzíteni az autó vázához. Az új szenzorral ki akartuk iktatni egy közbelső számítási egységet használatát is.

A kutatás következő fázisának részeként sikeresen gyűjtöttünk adatot a szenzorhálózatból és a tartó az autóra is fel fért végül a kardántengely alá. A remélhetőleg közeljövőben ezekkel a

szenzorokkal, szenzor hálózatokkal, első lépésben a labor összes autóját fel szeretnénk szerelni velük. Miközben ezeket a szenzorokat készítsük mi vagy egy másik munkacsoport, már dolgozik az önkalibráló szoftver megoldásokon.

Ezek a szenzorok és felhasználási lehetőségeik önmagukban is kifejezetten hasznosak és már dolgozunk a fizikális modellezésen, hogy a teljes funkcionalitásukat ki tudjuk használni. Költséghatékony vezetést is el tudjuk vele érni, mivel meg lehet vele állapítani azt, hogy mikor érdemes a sebességváltóval a következő fokozatba kapcsolni. Talán az egyik leghasznosabb felhasználási módja a kerékelfordulás mérésével együtt kialakított balesetmegelőző rendszer lehet. Számos megközelítési módot, vagy csak elméleti, vagy csak vázlat szinten tudtunk kidolgozni, mivel az utón történő helyhiány és az autó szerkezeti korlátosságai számunkra nem tették lehetővé ezeknek a megvalósítását. A motor által az autóba táplált mozgási energia mennyiségét csak a sebességmérő szenzorral tudjuk megállapítani, viszont a motor tényleges munkáját mérni nem tudjuk, mivel a fizikális kialakítása miatt ez jelenleg korlátozott.

4. Saját munka leírása

A kutatási munka részeként az elérhető eszközök helyett egy tulajdonságaiban számunkra előnyösebb szenzorok önvezető modellautókra történő kifejlesztését tűztük ki célul, melyeket később közösen egy nagyobb szenzorhálózatban használva többek között öndiagnosztikát, valamint baleset elkerülésre és megcsúszás korrigálásra szeretnénk továbbfejleszteni. A fejlesztés fázisaiban mindannyian szervesen részt vettünk. Az alábbi felosztás igyekszik objektíven szemléltetni fajsúlyosan az egyes fázisokban hozzáadott kontribúcióinkat.

4.1. Elfordulás szenzor:

Mári Ferenc:

- Analóg szenzor használata, hogy elkerüljük a switchekkel adódó problémákat.
- Egy Fekvőház használata a helyhiány miatt a nem megfelelő álló helyett
- Teleszkópozó mágnesstartó tervezése a parabolikus pályán mozgó tengelyhez
- Eredeti („vastagabb”) kábelezés
- Egy szenzor, sok mágnes mérési elképzelés
- Szenzor sorok átfedésben helyezkedjenek el a köztes állapotok kiszűrése érdekében
- Szoftver Raspberry PI/Jetson Nano

Zolnai Csaba:

- Digitális szenzor használata, hogy minimalizáljuk a szenzorok mennyiségét
- Egy Állóház használata, hogy a mágnes a szenzorok között haladhasson a pontosságért
- Új, jóval vékony kábelezés és új kábelvezetés a helyhiány miatt
- Egy mágnes, sok szenzor, mérési elképzelés
- A mágnes a két szenzorsor között mozogjon, hogy a szenzorok oldalával tudjunk mérni
- Az alkatrészek 3D tervezése
- ADC-k használata az Arduino kevés analóg bemenete miatt
- Szoftver könyvtárba való kiszervezése/átírása

4.2 Sebességmérő szenzor:

Girászi Tamás:

- Hall szenzorok használata szenzorokhoz, egyéb megközelítési módok vizsgálata
- GPIO pinek használata a sebességméréshez és a driver megírása
- Kerékre szerelhető mágnesstartó vázlatos kidolgozása

- Sebességmérő szenzor elkészítése, annak továbbfejlesztése, valamint az autóra történő felszerelése
- Mechanikai és kinematikai vizsgálatok és elemzések
- Fizikai modellek és alkalmazások kidolgozása a sebességmérő szenzorhoz használatához.

Köszönetnyilvánítás

A munkánk az Önvezető Jármű Labor keretei között zajlott. Ezúton szeretnénk köszönetet nyilvánítani Dr. Kovács Lászlónak a szakmai vezetésért, és a megoldás során adott ötletekért. Köszönetet szeretnénk továbbá adni Xie Yu PhD hallgatónak az elfordulás mérő szenzor kódolása során nyújtott tanácsaiért és segítségéért, Majoros Tamás PhD hallgatónak a nyáklapok tervezésénél nyújtott segítségért, Dr. Újvári Balázsnak a fizikai modellek kialakításánál nyújtott lektorálásáért, Baranyai Dávid PhD hallgatónak az elektrotechnika terén adott tanácsaiért, Kovács Ádámnak a 3D tervezésben nyújtott segítségért. Végül, de nem utolsó sorban pedig a szüleinknek a rengeteg támogatásért és türelemért.

Hivatkozások

1. Autonomous Vehicle Research Lab - University of Debrecen
<https://inf.unideb.hu/en/autonomous-vehicle-research-lab>
2. Autonomous Vehicle Research Lab UD (autonomous-vehicle-research-group-ud.github.io) <https://autonomous-vehicle-research-group-ud.github.io/>
3. Jezny J, Curilla M. Position measurement with Hall effect sensors[J]. American Journal of Mechanical Engineering, 2013, 1(7): 231-235.
4. Electronics tutorials website - <https://www.electronics-tutorials.ws/wp-content/uploads/2018/05/electromagnetism-mag26.gif>
5. SIEMENS, "Uni- and Bipolar Hall IC Switches for Magnetic Field Applications", TLE4905L datasheet, Sept. 1997
6. Karplus R, Luttinger J M. Hall effect in ferromagnetics[J]. Physical Review, 1954, 95(5): 1154.
7. BCD, linear hall effect IC AH49E hall pinout, data sheet, 2010. aug,
<https://www.diodes.com/assets/Datasheets/AH49E.pdf>
8. Arduino Weboldal: <https://www.arduino.cc/>
9. Texas Instruments, Ultra-Small ADS111x Ultra-Small, Low-Power, I²C-Compatible, 860-SPS, 16-Bit ADCs, 2018 jan., <https://www.ti.com/lit/ds/symlink/ads1114.pdf>
10. Pepka G. Position and level sensing using Hall-effect sensing technology[J]. Sensor Review, 2007.
11. F. Zidani, D. Diallo, M. E. H. Benbouzid and E. Berthelot, "Diagnosis of Speed Sensor Failure in Induction Motor Drive," 2007 IEEE International Electric Machines & Drives Conference, 2007, pp. 1680-1684, doi: 10.1109/IEMDC.2007.383682.
12. Mittal, A., Davis, L.S. A General Method for Sensor Planning in Multi-Sensor Systems: Extension to Random Occlusion. Int J Comput Vis 76, 31–52 (2008).
<https://doi.org/10.1007/s11263-007-0057-9>
13. A. L. Meyrowitz, D. R. Blidberg and R. C. Michelson, "Autonomous vehicles," in Proceedings of the IEEE, vol. 84, no. 8, pp. 1147-1164, Aug. 1996, doi: 10.1109/5.533960
14. S. Abrams, P. K. Allen and K. A. Tarabanis, "Dynamic sensor planning" [1993] Proceedings IEEE International Conference on Robotics and Automation, 1993, pp. 605-610 vol.2, doi: 10.1109/ROBOT.1993.291892

15. Spears, W.M., Spears, D.F., Hamann, J.C. et al. Distributed, Physics-Based Control of Swarms of Vehicles. *Autonomous Robots* 17, 137–162 (2004).
<https://doi.org/10.1023/B:AURO.0000033970.96785.f2>