

Óbudai Egyetem

Bánki Donát Gépész és Biztonságtechnikai Mérnöki Kar
Mechatronikai és Autótechnikai Intézet



TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

**Saját tervezésű önállóan navigáló mobilrobot
tervezése és kivitelezése**
**(Design and realization of an autonomous mobile
robot)**

Szerző: Rippert János
Mechatronikai mérnök BSc.szak, III. évf.

Konzulens: Dr. Nagy István
egyetemi docens

Lezárás dátuma: Budapest, 2014.04.23.

Tartalomjegyzék

1. BEVEZETÉS	2
2. A robotváz tervezése	3
3. Az elektronikus alkatrészek kiválasztása	5
3.1 Mikrokontroller	5
3.2 DC motorok	7
3.3 H-híd	8
3.4 Léptetőmotor vezérlő	9
3.5 Léptető motor	10
3.6 Infravörös távolságszenzor	11
3.7 Ultrahangos távolságszenzor	14
3.8 LIPO akkumulátor	17
3.9 Inkrementális tárcsa, enkóder és infravörös LED	18
4. A parkolás menete lépésről lépésre	25
4.1 A robot feltérképezi környezetét és kiválasztja a legközelebbi pontot	25
4.2 Abba az irányba fordul, amerre a legközelebbi falrész van	27
4.3 Elindul a fal felé és attól 35 mm-re megáll	32
4.4 Elfordul a 90 fokkal jobbra a fallal párhuzamosan.....	35
4.5 Az infravörös szenzort elforgatja a fal felé 90 fokkal	36
4.6 A robot elindul a fal mellett egy adott távolságban követve azt	37
4.7 Falkövetés közben helyet keres	39
4.8 Ha megtalálja a helyet, akkor beparkol	42
5. Eredmények kiértékelése.....	51
6. Összegzés és fejlesztési lehetőségek	52
7. Köszönetnyilvánítás	53
8. Irodalomjegyzék:.....	54

1. BEVEZETÉS

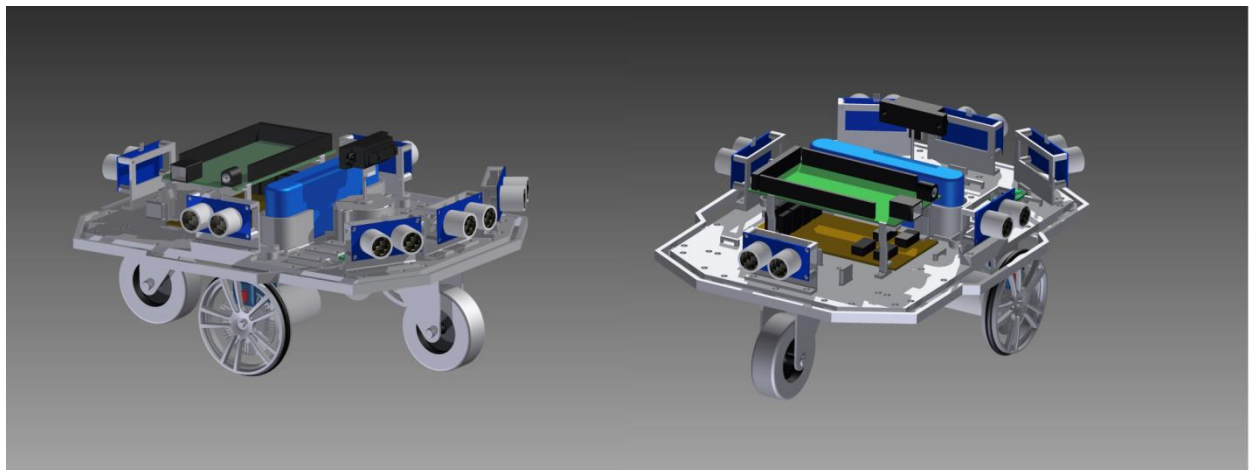
Ma egyre elterjedtebb a járműveknél az intelligens parkolás és az intelligens manőverezés. Különböző cégek különböző megoldásokkal próbálják a parkolás mechanizmusát megvalósítani. A parkolási műveletnek jól átgondoltnak, precíznek, zökkenőmentesnek és hibátlanak kell lennie, különben komoly balesetek származhatnak belőle a való életben. Azért választottam ezt a témát, mert nagyon sok fejleszteni való van még.

A parkolást egy általam tervezett saját konstrukciójú mobilrobottal valósítom meg. Inventor tervezőprogrammal terveztem meg a vázat, melynek alkatrészei 3D-s nyomtatóval lettek kinyomtatva. Az elektronikai alkatrészeket hosszas válogatás után, több szempont alapján választottam ki. A robot tájékozódását 6 darab ultrahangos és egy darab infravörös szenzor segíti. Az infravörös szenzort egy léptetőmotorral forgatom körbe, így a környezet feltérképezésére is alkalmas. A robot haladása során megtett utat inkrementális tárcsákkal, inkrementális jeladóval – odométeres technikával mérem. A rendszert működtető programot Arduino Mega mikrokontrollerre Arduino programozási környezetben fejlesztettem. Ebben a projektben kihasználom a differenciál hajtás adta előnyöket: hirtelen irányváltás, úgy, mint helyben fordulás, kis parkolási helyigény.

A dolgozat bemutatja a váz tervezésének és az alkatrészek kiválasztásának szempontjait. Az elektronikus alkatrészek működésének bemutatását szintén magába foglalja a dolgozat. Tartalmazza még a parkolási folyamat leírását: a komplett folyamat kisebb alfolyamatokra való bontását. A mozgást leíró tényezők és a szabályzás bemutatása is dolgozatom részét képezi.

2. A robotváz tervezése

A robotváz tervezése során megvizsgáltam sokféle lehetőséget, hogy miből és milyen technológiákkal lehet a robotvázat elkészíteni. A választásom több szempont alapján a műanyag 3D-s nyomtatásra esett. Ennek több oka is volt. Elsőként fontos volt, hogy a szerkezet kialakításánál ne ütközzek nagyon szoros technológiai korlátokba. Másodszor csak kis átalakításokkal lehessen kivitelezni a megtervezett alkatrészeket. Harmadszor a váz legyen könnyű, merev, jól megmunkálható és olcsó. Ezenfelül fontos szempont volt, hogy a szerkezet könnyen bővíthető és átalakítható. A váz alkatrészeit magam terveztem az Autodesk Inventor programmal, a rendelt alkatrészek befoglaló méreteit és csatlakozási pontjait figyelembe véve.



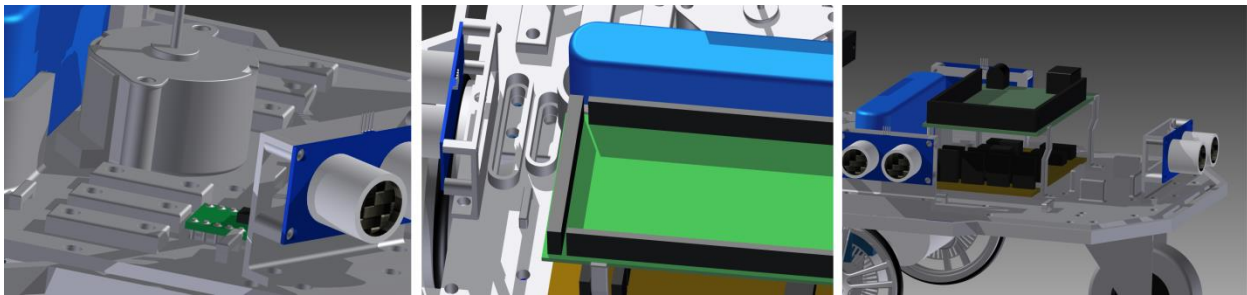
1. ábra A mobilrobot Inventoros összeállítása

A robot befoglaló méretei 265mm*200mm*135mm. A váz alapját két részre osztva kellett megtervezni, mert a 3D-s nyomtató munkaterülete 200mm*200mm*200 mm-es. A két részt 12 darab furatos összefogató elemmel rögzítettem egymáshoz. Hat darab van felül és hat darab van a váz másik oldalán. A robot köré ütközőket terveztem, amelyek csak kis keresztmetszetűek, így kellően rugalmasak ahhoz, hogy a robot ütközéskor ne károsodjon.

A robot szerkezete lehetővé teszi, hogy három, illetve négykerekű járműveket is tudjon szimulálni. A mobilrobot differenciál hajtású, azaz oldalsó kerekeit külön-külön motor hajtja. Egy, vagy két darab bolygókerék szerelhető fel rá, így lehet három illetve négykerekű. Ahhoz hogy mindkét verziójában stabilan tudjon maradni, terveztem a vázra hosszanti hornyokat, amikben a kereket lehet csúsztatni előre és hátra. A csavar feje jól beillik a hornyba és abban

csúszik. Háromkerekű verziójában a kerekeket, illetve a kerekeket tartó felfüggesztést előre lehet tolni 20 milliméterrel, így ekkor jelentős stabilitás érhető el. Én a négykerekű verziót választottam a négykerekűek tesztelése miatt. A hajtott kerekek pontosan a robot közepénél helyezkednek el, így a robot helyben meg tud fordulni.

Az akkumulátor a mobilrobot közepén kapott helyet, ezért az akku tömegéből adó tehetetlenségi nyomatékkal nem kell olyan nagymértékben megküzdenie a DC motoroknak. Helyspórolás miatt a H-hídat és az Arduino Mega mikrokontroller NYÁK-ot egymás fölé raktam. A két NYÁK lapot 4 darab tartó pillér tartja el egymástól. A H-híd csak nagyon kevés hőt termel a nagyteljesítményű FET-eknek és a kisteljesítmény DC motoroknak köszönhetően, így nem kell attól félnem, hogy a termelt hő kárt tesz az Arduino NYÁK-ban.



2. ábra Rögzítő elemek, állítható kerék, távtartók

Az infravörös és az ultrahangos távolság szenzorokhoz olyan felfogató-rögzítő alkatrészeket terveztem, melyek segítségével a szenzorok vázhoz való rögzítése nagyon egyszerű.

A kábelköteget elvezetésének megkönnyítésében a vázba néhány furatot terveztem. Az Arduino Mega, a H-híd és a léptető motor vezérlőt külön kapcsolókkal tudom ki- és bekapcsolni. A kerekeket úgy lettek megtervezve, hogy egy 56*4mm-es O-gyűrűt rá lehessen tenni. Ez azért kell, mert a műanyag kerék és a talaj között nem volt elég nagy a súrlódás és a kerekek kipörögtek. A problémát az imént leírt gumi O-gyűrűvel oldottam meg. Ennek rögzítéséről a kerék külső peremén egy félkör alakú vágat gondoskodik. A vágat pontosan úgy lett méretezve, hogy az O-gyűrű ne tudjon elfordulni a keréken.

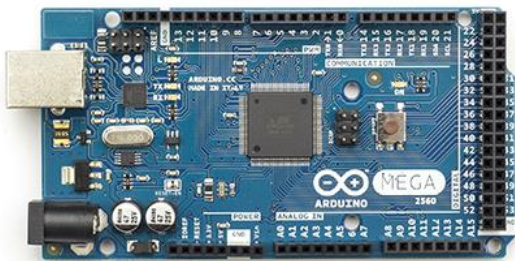
A 2 darab csapágyazott bolygókerék IKEA gyártmányú.

3. Az elektronikus alkatrészek kiválasztása

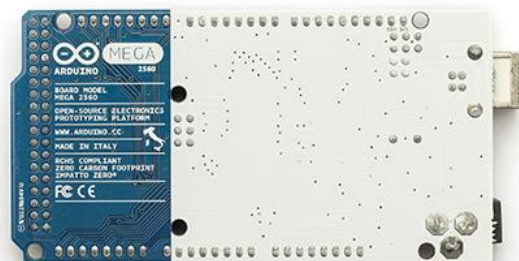
3.1 Mikrokontroller

A választott mikrokontroller: Arduino Mega. [1] Atmel gyártmányú ATmega 2560-as 8 bites chip a központi egysége. Ez a chip egy számára tervezett Arduino NYÁK-on foglal helyet. Számítógépről USB porton lehet rá programot írni. Az Arduino működtetéséhez 5voltos egyenáram szükséges. Külső tápfeszültség esetén 6-20V a működéshez szükséges minimum és maximum feszültség. Nagyobb feszültségből úgy állít elő kisebbet az eszköz, hogy a felesleges feszültséget elfüti. Az ajánlott tápfeszültség 7-12V. Hét V alatt a rendszer működése nem teljesen stabil. Tizenkét V fölött a „tápkocka” annyira felmelegedhet, hogy kárt tehet a NYÁK lapban. Maximálisan 500mA nagyságú áramot tud leadni.

Kép [2]



Arduino Mega 2560 R3 Front



Arduino Mega2560 R3 Back

3. ábra Arduino Mega 2560

54 darab digitális I/O láb van rajta. Az I/O lábakon leadott maximális áram lábanként 40mA. Ebből 15 PWM kimenet. Továbbá 16 darab analóg bemenet is van rajta. A memóriachipből három féle is van rajta. [3] Flash, SRAM és EEPROM. Az SRAM felejtő memória, míg a Flash és az EEPROM nem. A felejtő memóriák áramtalanításkor elvesztik a tartalmazott adatokat. A nem felejtő memória évekig képes megőrizni a ráírt adatokat. A Flash memóriában az Arduino program fejlesztőkörnyezetében létrehozott program vázlat tárolódik, az SRAM-ban pedig a program során használt változók.

Az ATmega chip órajele 16 MHz. Ez a sebesség elegendő a tervezett feladat végrehajtásához.

M is a PWM?

A vezérlésben és a szabályzásban használt analóg jeleket helyettesíteni lehet olyan digitális jelekkel, amelyek hosszútávon hasonlóak. Úgynevezett digitális impulzus-sorozatokkal. Ennek frekvenciáját kellően nagyra kell beállítani, máskülönben komoly problémák merülhetnek fel a vezérelt, vagy szabályozott eszközben. Például egy lámpa fényének impulzusát úgy kell beállítani, hogy az ne legyen zavaró, ne vibráljon. Az én esetemben az egyenáramú motorokat szabályozom ilyen jelekkel. Ennél az a fontos szempont, hogy a motorok ne csináljanak lökésszerű fordulatszám változásokat.

Az egyik ilyen digitális jeleket használó módszer a PWM. Pulse Width Modulation, magyarul Impulzus-szélesség modulált jel. Ezek a jelek olyan állandó periódus idejű és frekvenciájú jelek, amelyeknél a feszültség beállítása a jel kitöltési tényezőjétől függnek.

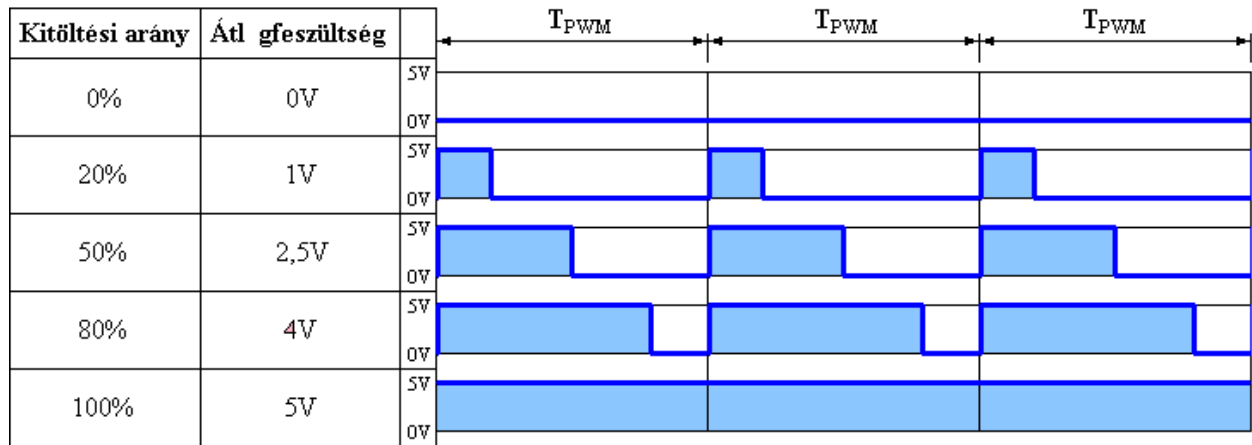
A PWM [4] jelek digitális elemekkel történő megvalósításához szükség van egy állandó órajellel léptetett számlálóregiszterre, amelynek teljes átfutása adja a PWM jel T_{PWM} periódusidejét. A számlálóregiszter minden túlsordulásakor új periódus kezdődik, a PWM kimenet szintje logikai "1"-re vált. Egy, a számlálóregiszterrel megegyező hosszúságú "kitöltési ciklus" regiszterbe kell elhelyezni a kitöltési tényezőnek megfelelő értéket, amely folyamatos összehasonlításra kerül a számlálóregiszterrel. Amikor a számlálóregiszter értéke eléri a "kitöltési tényező" regiszter értékét, a PWM kimenet szintje "0"-ra vált. A PWM jelek pontosságát a kitöltési tényező beállíthatóságának pontossága, vagyis a számlálóregiszter hossza határozza meg.

A PWM jelek két alapjellemzője:

- a felbontás, amelyet a számlálóregiszter hossza, azaz bitszáma határoz meg (például "8-bites PWM jel");
- és a frekvencia, amely a számlálóregiszter hosszától és léptető jelének frekvenciájától függ.

A képen a PWM jelek láthatóak különböző kitöltési tényezőkkel.

Kép [5]

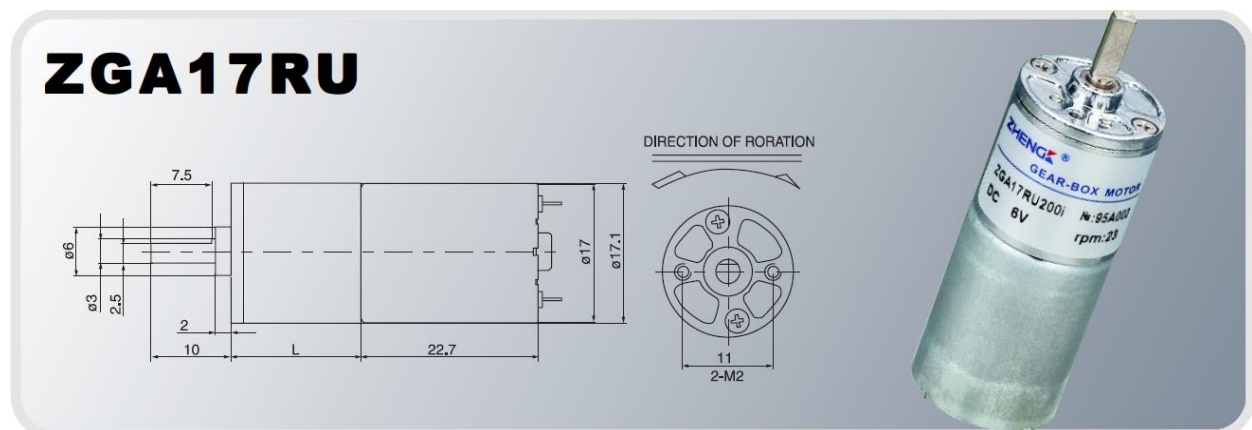


4. ábra PWM jelek különböző kitöltési tényezővel

3.2 DC motorok

Zheng márkájú [6] 12 voltos egyenáramú áttételes motorokat választottam. Ezek a motorok viszonylag kis fogyasztásúak és nagy nyomatékot képesek leadni az 1:100-as áttételnek köszönhetően (maximális nyomaték 0.75Nm). A motor tengelyén mérhető maximális fordulatszám 100 fordulat/perc. Folyamatos maximális teljesítménye 0.8W. A befoglaló méretei: 64.2mm x 25mm x 25mm. Könnyen rögzíthető az áttételházon lévő menetes furatok miatt.

Kép:[7]



5. ábra Zheng 12V 100rpm DC motor

A robot maximális sebességét a kerék kerülete és a fordulatszám szorzataként határoztam meg.

A kerék kerülete:

$$K = 2 * R * \pi \quad (3.2.1)$$

ahol az $R=32\text{mm}$.

$$K = 2 * 32 * \pi = 201.0619\text{mm}$$

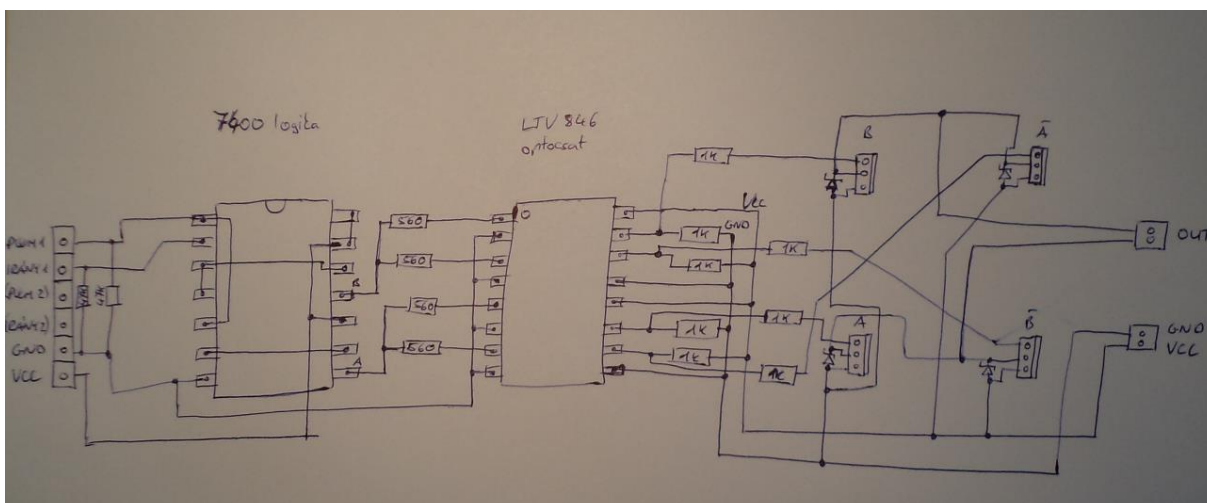
A kerületet meg kell szorozni a maximális forgási sebességgel:

$$v = \frac{100}{60} * K \quad (3.2.2)$$

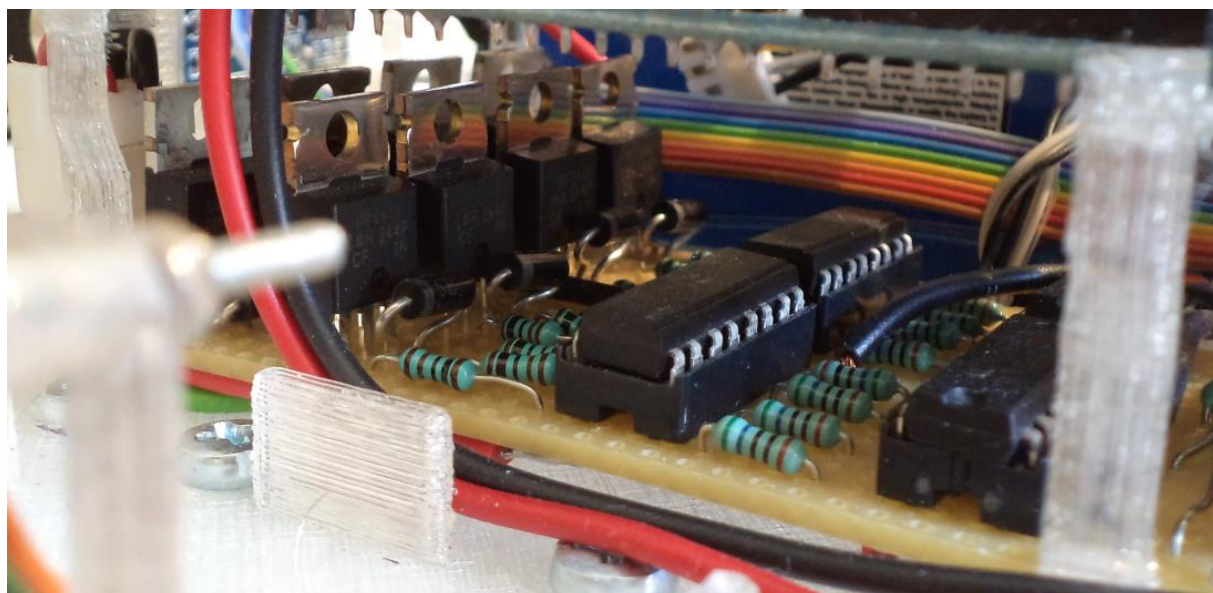
$$v = \frac{100}{60} * 201.0619 = 0.335 \frac{m}{s}$$

3.3 H-híd

Saját tervezésű és saját kivitelezésű kétcsatornás H-hídat alkalmazok a motorok meghajtására. Egyetlen próba NYÁK-ra furatszerelt alkatrészeket építettem be. Két külön fizikai részre osztható az áramkör a galvanikus leválasztás miatt. Az egyik a vezérlő rész, a másik a teljesítmény rész. A vezérlő rész az Arduino mikrokontrollertől kapja a jelet (PWM és irány). A teljesítmény rész maga a H-híd. Két motor hajtható meg vele. FET-eket használok a jobb teljesítmény miatt. A vezérlő rész 5 voltos, míg a teljesítmény rész 5 és 55 volt között tápfeszültséggel működik. Megfelelő hűtés mellett akár 41 Amper áramot is képes leadni.



6. ábra H-híd kapcsolási rajz



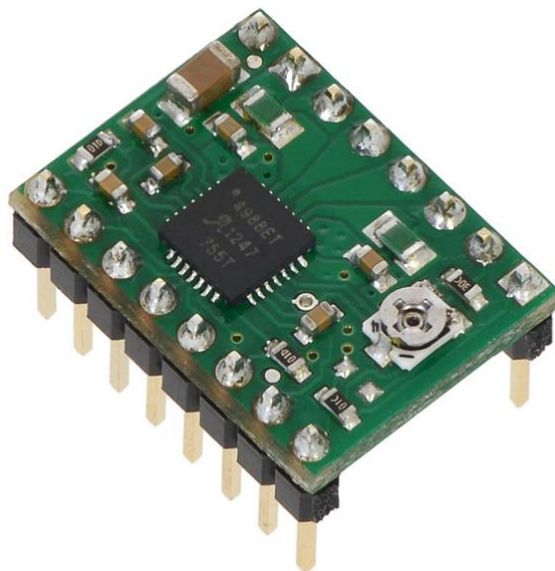
7. ábra Összeszerelt H-híd

3.4 Léptetőmotor vezérlő

Egy Plolu A4988 [8] léptető motor vezérlőt használok. Ez egy nagyon megbízható, olcsó és jó minőségű vezérlő. Csak bipoláris, vagy bipolárisra alakítható hat, vagy nyolc kivezetéses unipoláris léptető motor tud kezelni. A bipoláris léptető motor egymástól két független tekercset tartalmaz. Minden motornál meg van adva, hogy a két tekercs közötti egész lépés hány fok. A

mikrosztepping mód ezt az egy lépést tudja több részre osztani. A Pololu tudja felezni, negyedelni, nyolcadolni és akár tizenhatodolni ezt az egy lépést. Ennél az alkatrészénél is elkülönül a vezérlő és a teljesítmény rész. A vezérlő az 5 voltos mikrokontrollerről, míg a teljesítmény rész 8 és 35 volt közötti külső tápfeszültséggel működőképes. Megfelelő hűtés mellett tekercsenként 2 Amper áramot tud biztosítani. Vezérlése nagyon egyszerű.

Kép: [9]



8. ábra Pololu léptetőmotor vezérlő

3.5 Léptető motor

Egy JAPAN SERVO CO. Márkájú TYPE KH42JM2B140E [10] típusú léptető motort használok. Azért választottam ezt, mert ezt tudja vezérelni a Pololu léptető motorvezérlő. Ez egy hat kivezetésű unipoláris léptető motor, amely 5 és 16 voltos feszültség között használható és 1.2A árammal is képes léptetni. Egy lépés 1.8 fok. Ezzel forgatom az infravörös távolságszenzort. Nem használom a mikroszteppinget, mert a mikrokontroller memóriája nem elég nagy ahhoz, hogy ennyi adatot eltároljon egy tömbben.

Kép: [11]

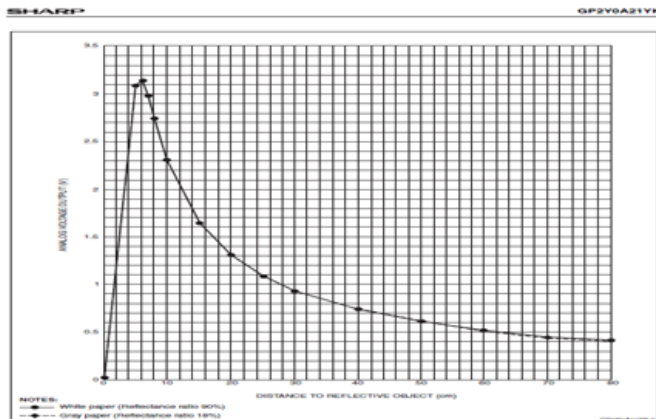


9. ábra KH42JM2B140E léptetőmotor

3.6 Infravörös távolságszenzor

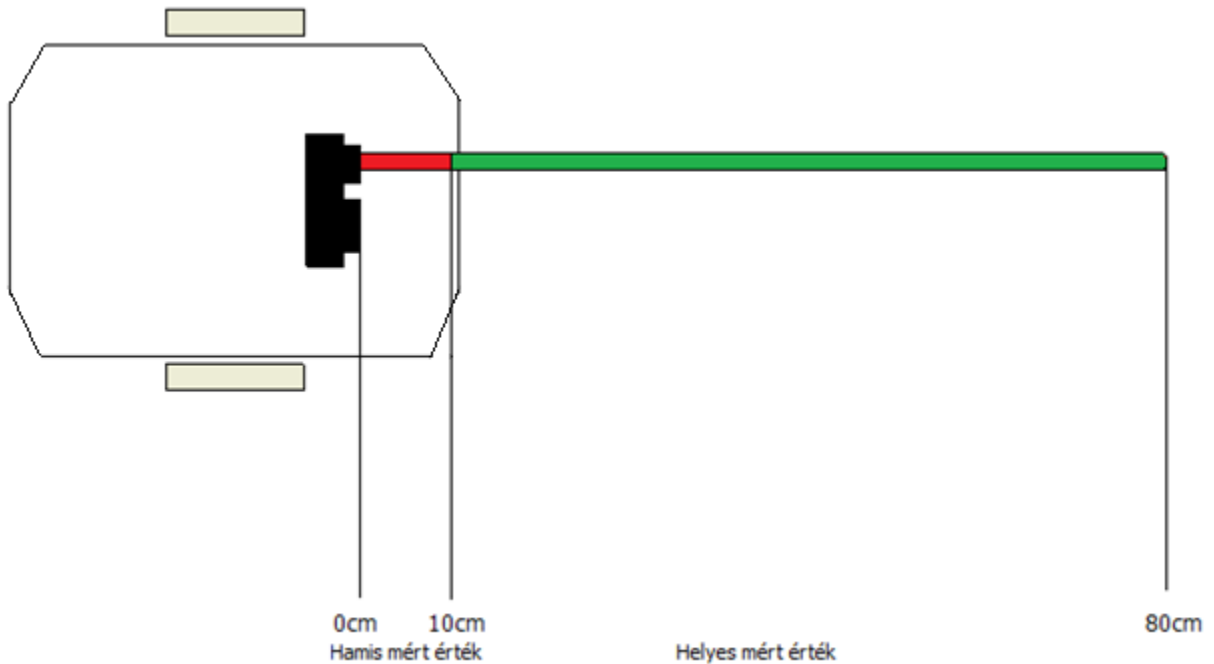
Sharp 2Y0A21YK [12] típusú infravörös távolságszenzort használok. Ez a szenzor olyan célokra jó, ahol nem olyan fontos, hogy messzire ellásson a robot. Az infravörös LED fényimpulzusokat ad ki 32 milliszekundumonként és a tárgyról visszaverődött fényt egy infravörös foto tranzisztor érzékeli (IR fény hullámhossza $850\text{nm} \pm 70\text{nm}$). A kibocsátott fénynyaláb csak kismértékben szóródik a távolság növekedésével, így a szenzor jó megközelítéssel egy pontban lát [13]. A mikrokontrollerbe érkező mérési adatok analóg jelek. Erősségük attól függ, hogy milyen messziről érkezik vissza az érzékelt fény. A jel a távolsággal fordítottan arányos és nemlineáris.

Kép: [14]



10. ábra Sharp 2Y0A21YK kimeneti feszültség-távolság karakterisztikája

Minél messzebb van az objektum, annál gyengébb a visszavert fény. A fenti feszültség-távolság függvényen jól látszik, hogy ha egy tárgy közelebb van, mint a katalógusban leírt legkisebb mérési távolság, akkor a szenzor által a mikrokontroller felé küldött jelek nem lesznek értelmezhetőek.



11. ábra Az infravörös szenzor elhelyezése a roboton a rossz méréstartomány kiszűrésére

Az Arduino azt hiheti, hogy a tárgy távol van, miközben nagyon közel. Ezért a szenzort úgy helyeztem el a roboton, hogy az semmiképpen ne legyen 20 mm-nél közelebb semmilyen tárgyhöz. [15] A szenzor hatótávolsága 10-től 80 cm. Az infravörös szenzor 5 voltos tápellátással működik, de a kimeneti jele maximum 3.3 V. A bekapcsolási késleltetése 44 milliszekundum, átlagos áram fogyasztása 30 milliamper. Ezt impulzusszerűen veszi fel. Reseteléskor nagy lehet a mikrokontrolleren a feszültségesés, ezért érdemes egy 10 mikroFarad-os kondenzátort bekötni a GND és a Vcc közé.

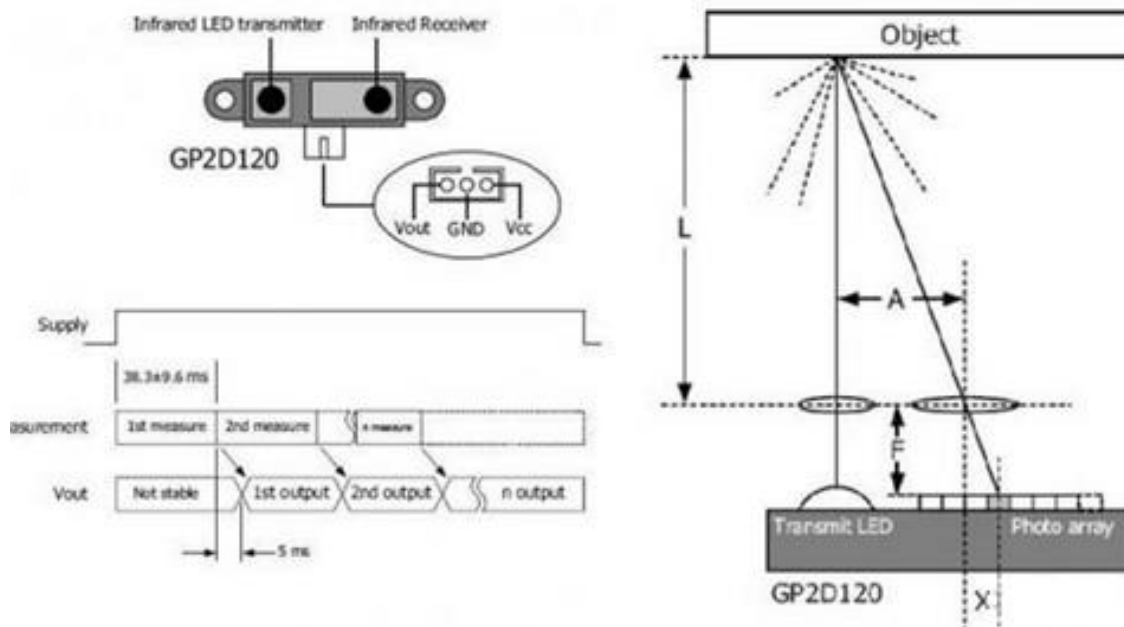
Kép:[16]

Sharp GP2Y0A21YK



12. ábra Sharp 2Y0A21YK infravörös szenzor

Kép: [17]

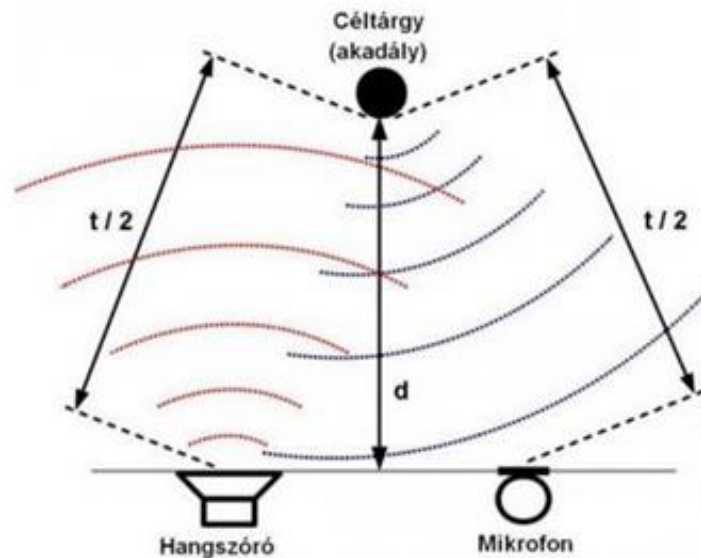


13. ábra Sharp 2Y0A21YK infravörös szenzor kimenetei (balfent), mérési ideje (bal lent), és a távolságmérést szemléltető ábra (jobb)

3.7 Ultrahangos távolságszenzor

A HC-SR04 ultrahangos távolságmérő szenzor az emberi fül számára nem hallható 40KHz-es ultrahangot bocsát ki. [18] A szonár elv alapján működik. A szenzor két fő eleme az ultrahangokat kibocsátó hangszóró (adó) és a visszaverődött ultrahangok érzékelésére szolgáló mikrofon (vevő).

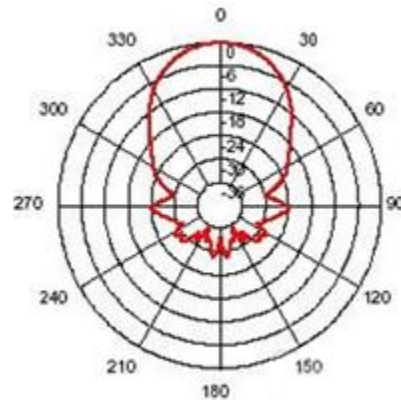
Kép: [19]



14. ábra Az ultrahangos szenzor távolság mérési elvét szemléltető ábra

A magasabb frekvenciájú hangok kevesebb energiát disszipálnak, ezért az ultrahangok koncentráltabb, szűkebb nyalábban terjednek, mint a hagyományos hanghullámok. Ez a tulajdonság segít a szenzornak abban, hogy csak az előtte lévő viszonylag szűk szögtartományban elhelyezkedő tárgyakat észlelje. A HC-SR04 15 fokban képes érzékelni.

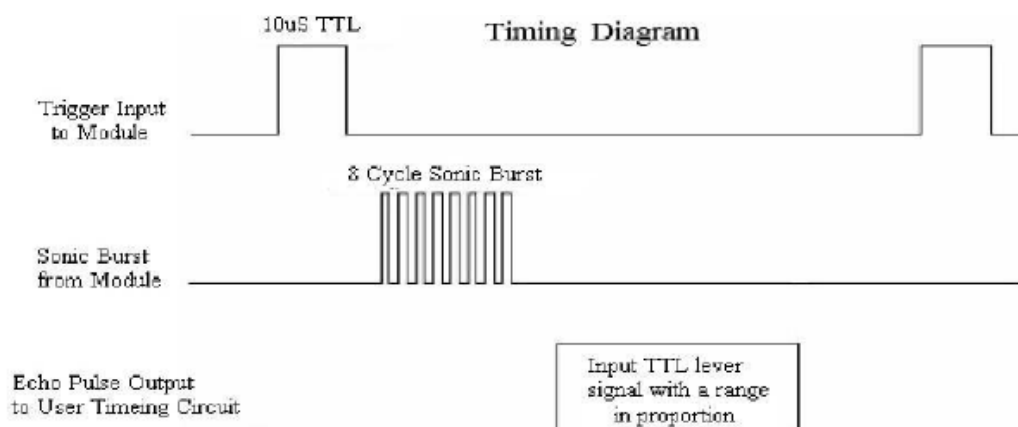
Kép: [20]



15. ábra HC-SR04 ultrahangos szenzor mérési tartománya

A szenzornak négy lába van [21]. A VCC, a GND és emellett egy Trigger és egy Echo láb. Úgy működik, hogy ha a Trigger lábra egy 10 milliszekundumos (magas) impulzust adunk, akkor az „adó” kiad 8 darab 40KHz-es jelet. A „vevő” ekkor lesz aktív. Méri azt az időt, ami ahhoz kell, hogy a kimenő impulzusok visszaverődjenek. A pontos távolság kiszámításához ismerni kell a hangsebességet. Ez 340m/s szobahőmérsékleten és tengerszinten. A távolság a $s=t*v$ képlettel számolható. Ebben az esetben $(MértIdő*hangsebesség)/2=távolság$. Azért kell elosztani kettővel a szorzatot, mert a hang az utat kétszer tette meg: „adó” – tárgy, tárgy - „vevő”.

Kép: [22]



16. ábra HC-SR04 ultrahangos szenzor mérési idejei

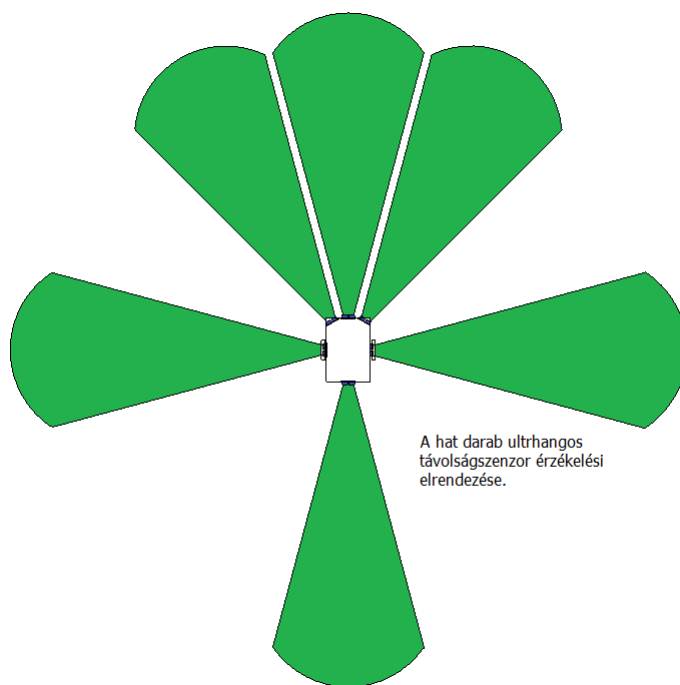
A HC-SR04 érzékelési távolsága: 2 -400cm. A szenzor áramfogyasztása 15 mA.

Kép: [23]



17. ábra HC-SR04 ultrahangos

A robot 6 db ultrahangos szenzorral van ellátva:



17. ábra A mobilroboton elhelyezett ultrahangos szenzorok érzékelési területei

Három darab ultrahangos szenzor van az robot elején. Ebből kettő harminc fokos szöget zár be a középsővel. Kettő van oldalt és egy hátul.

3.8 LIPO akkumulátor

A választásom a lithium-polimer akkumulátor fajtára esett. Ez nagyon könnyű, kis veszteségű és a legjobb töltéssűrűségű akkumulátor. [24] Nagyon nagy áram leadására képes. Folyékony elektroliton nem, vagy csak nagyon kis mennyiségben tartalmaz. Vékony és rugalmas falú cellákat lehet belőle létrehozni, mert nem kell vastag fallal védekezni a folyadék kifolyása ellen. A többcellás LiPo akkumulátoroknál fontos, hogy az összes cella feszültsége azonos legyen. Egy cella névleges minimális feszültsége 3V, maximális 4.2V. Ezen értékek alatt vagy felett a cella visszafordíthatatlan károsodást szenved. Például ha már csak 9V van egy 11.4V névleges feszültségű, 3 cellás akkumulátorban, akkor nem biztos, hogy minden cella feszültsége 3V. Ha első cella feszültsége 2.8V, a másik kettő 3.1V, akkor az egyik cella már károsodott. Folyamatosan monitorozni kell az akkumulátort használat közben. Az akkukon van egy úgynevezett balanszer csatlakozó, mely pont arra szolgál, hogy töltéskor az összes cellát azonos feszültségűre tudjuk feltölteni.

Hátránya, hogy a gyengébb borítás miatt fokozottan sérülés érzékeny. Nem szabad túl nagy árammal tölteni.

A robotba egy 1500mAh-ás 3 cellás Zippy lightmax LiPo akkut választottam, melynek névleges feszültsége 11.1V. A megengedett feszültsége 12.4 és 9 V között változhat. A mikrokontroller stabil tápellátása miatt maximum 12 V-ig töltöm fel. A rendszer maximális áramfelvétele körülbelül 2.5 Amper.

Kép: [25]

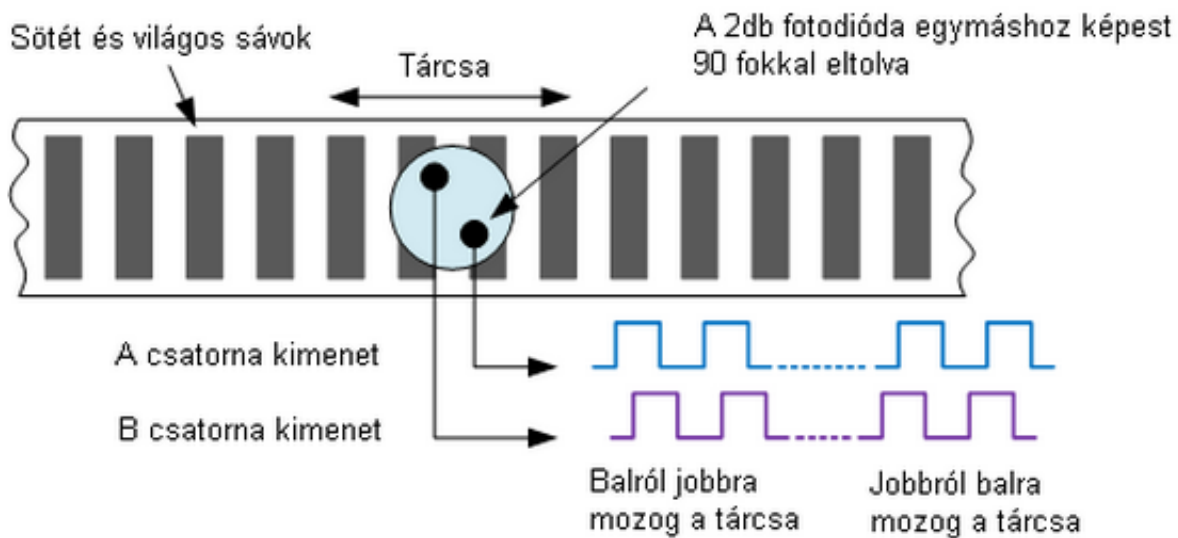


18. ábra LiPo akku

3.9 Inkrementális tárcsa, enkóder és infravörös LED

A kerékre felerősítve található az inkrementális tárcsa. Ez készülhet üvegből, fémből, vagy műanyagból is. [26] Az én esetemben vetítőfóliára lett nyomtatva lézer nyomtatóval a minta. A minta a fólián a tárcsa pereméhez közel elhelyezett osztásokból áll. Ezek az osztások feketék. A szögjeladó működési elve: ha a két osztás között átjut a fény, akkor az enkóder érzékeli és feldolgozza a jelet, ha pedig a rovátkára esik a fény, akkor nem jut át, ezáltal az enkóder nem érzékel semmit. Ezek az osztások határozzák meg a szögjeladó felbontását. Ha 50 darab sötét és 50 darab világos osztás van, akkor annak a szögjeladónak a fizikai felbontása 50.

Kép: [27]



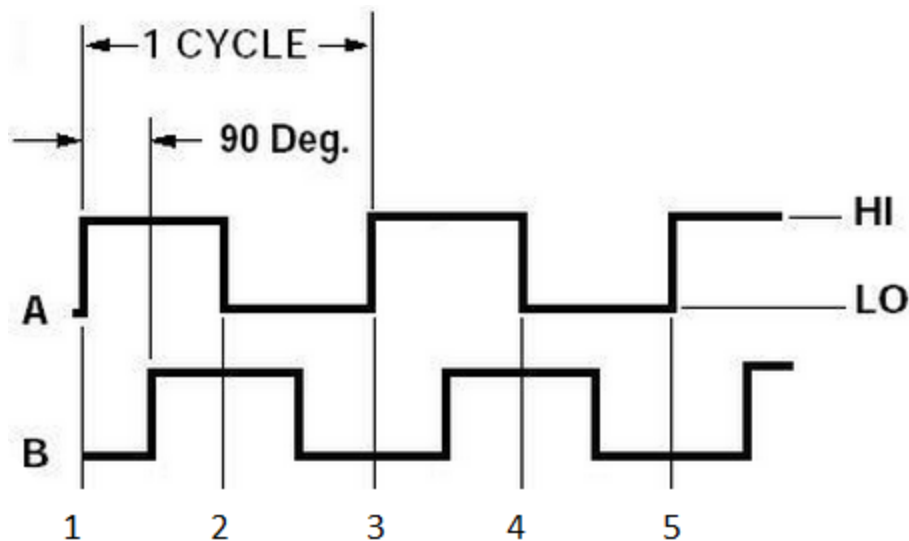
19. ábra A szögjeladó működése

Ezt a műanyag korongot megvilágítom egy darab LED diódával az egyik oldalról, míg a másik oldalon két darab fotodióda érzékeli a fényt. A fotodiódák 90 fokkal el vannak forgatva egymáshoz képest, ezáltal egymáshoz 90 fokban eltolt fázisú jelet adnak. A fotodiódák közel szinuszos jeleit négyszögjellé alakítva a jeladó kimenetein megjelenik 2 darab egymástól 90 fokban eltolt impulzus sorozat. Az egyik impulzus sorozat az A csatorna jele a másik impulzus sorozat a B csatornájé.

Normál egyszeres dekódolás:

Az A és B csatorna [28] kimenetein impulzusok jelennek meg, ha az inkrementális tárcsa elfordul. Ebből kell kiszámolni az elfordulás szögét vagy távolságát. Ha az A vagy B csatornán lévő impulzusokat megszámloljuk, akkor az így kapott szám alapján tudjuk, hogy mennyi az elfordulás. De ez csak az egyik irányba igaz, mert ha a másik irányba fordul a tárcsa, akkor szintén ilyen impulzusok fognak megjelenni és nem tudjuk, hogy előre, vagy hátra forog a tárcsa. A forgás irányához egyidejűleg mindkét csatorna jelét vizsgálni kell.

Kép: [29]



20. ábra Az A és B csatorna jelei, egyszeres dekódolás

Ezen az ábrán nyomon követhetjük az A és B csatorna jeleinek változását. A fontosabb állapotokat az 1,2,3,4,5 számok jelzik. Ha balról jobbra forog a tárcsa, akkor a sorrend időben az 1-es, 2-es, 3-as, 4-es és az 5-ös pont. Az egyszeres dekódoláshoz a felfutó és lefutó éleket figyeljük. Ha az A dekódoló a felfutó éleket figyeli, akkor az 1-es a 3-as és az 5-ös pontokban fogja a lépést számolni. Az ábrán jól látható, hogy a B csatorna ilyenkor mindig alacsony (LOW) szintű. Ha megfordítjuk a forgásirányt, akkor az események fordított sorrendben következnek be. Tehát először az 5-ös, majd a 4-es, 3-as, 2-es és az 1-es. Ha ismét a felfutó élet figyeljük, akkor az A csatornán először a 4-es, majd a 2-es állapotoknál fog lépést számolni. Ilyenkor a B csatorna jelszintje mindig magas.

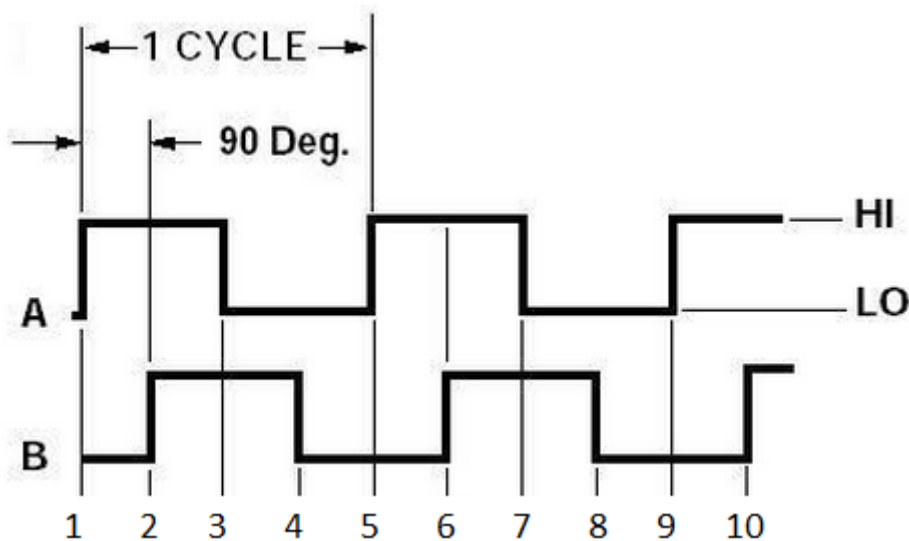
Kétszeres dekódolás:

Ha az A csatornán felfutó él érkezik és a B csatorna alacsony szintű, illetve, ha az A csatornán lefutó él érkezik és a B csatorna magas szintű, akkor a szögadó lépett egyet balról jobbra. Ha az A csatornán felfutó él érkezik és egyidejűleg a B csatornán magas szint van, illetve ha az A csatornán a lefutó él mellé alacsony B csatorna szint tartozik, akkor a szögadó jobbról balra lépett egyet.

Négyszeres dekódolás:

A négyszeres dekódolásnál a szögjeladó fizikai felbontását megnégyszerezjük. Itt már kicsit bonyolultabb a dekódolási metódus. Az A és B csatornán is figyeljük mind a felfutó, mind a lefutó éleket. A mellékelt rajzon látható, az összes állapot váltás az A és B csatornán.

Kép: [30]



21. ábra Az A és B csatorna jelei, négyszeres dekódolás

A négyszeres dekódolásnál elméletileg lehetséges a lépés hiba felismerés is. Ezt az teszi lehetővé, hogy a szögjeladó bármely állapotában tudjuk, hogy mi lehet a következő két állapot. Azért kettő, mert jobbra és balra is elmozdulhat a tengely. Ennek a lehetőségnek a megértéséhez következzen a négyszeres dekódolás állapotgépe. Melynek segítségével könnyedén megérthetjük ennek a hibafelismerésnek elvét.

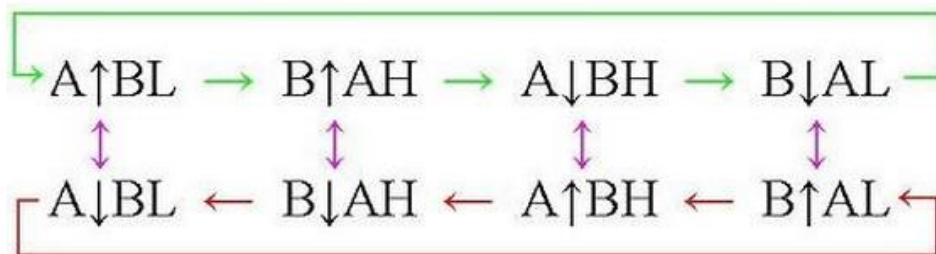
Négyszeres dekódolás állapotgépe:

Kép: [31]

Balról Jobbra lép a szögadó egy lépést	A felfutó él és B alacsony	B felfutó él és A magas	A lefutó él és B magas	B lefutó él és A alacsony
Jobbról Balra lép a szögadó egy lépést	A felfutóél és B magas	B lefutó él és A magas	A lefutó él és B alacsony	B felfutó él és A alacsony

22. ábra Négyszeres dekódolás állapotgépe 1

Kép: [32]



↑: Felfutó él

↓: Lefutó él

←, →, ↑↓: Lehetséges irányok

23. ábra Négyszeres dekódolás állapotgépe 2

A balról jobbra történő lépést a zöld szín jelöli. A balról jobbra történő lépést jelenti a vörös szín. A lila szín azt mutatja, amikor irányt vált a szögjeladó. Az A és B jelzi az A és B csatornákat. Az L és H pedig az alacsony és magas logikai szinteket jelentik. A szögjeladó utolsó ismert állapota az A csatornán egy felfutó és a B csatornán pedig egy alacsony jelszint. Ezt az állapotgép bal felső sarkában láthatjuk. Ezt az állapotot két lépés követheti. Az első lehetőségben a jeladó továbblép, így a B csatornán felfutó él és az A csatornán magas érték lesz. Vagy második lehetőségként a szögjeladó visszalép. Ilyenkor Az A csatornán lefutó él lesz, míg a B csatorna

alacsony jelszintet fog mutatni. Ezeken kívül nincs más lehetőség. Ha ezen kívül mást érzékel a mikrokontroller, akkor lépéshiba történt és valami probléma van a rendszerrel.

Honeywell márkájú HLC2701 [33] típusú enkódert használok. Úgy lett tervezve, hogy a mozgását és az irányát is érzékelje a mechanikai mozgásnak. A detektor egy monolitikus IC, amely tartalmaz két darab fotodiódát, szűrőt és Schmidt-triggert. A fotodiódák 90 fokkal vannak egymástól elfordítva. A szűrő a pontosabb, zajmentesebb jelekért felelős, a Schmidt-trigger pedig jó felfutású és lefutású négyszögjelet állít elő. A kimeneti jel érzékenysége és az infravörös LED hőmérséklet miatt fellépő hibája kompenzálva lett. A fotodiódák egy fekete műanyag tokba kerültek bele, amely jól átengedi az infravörös fényt, de a látható fényt megszűri. Az enkóder felbontása 0.229mm. Fogyasztása 7mA.

Honeywell márkájú, SEP8506-001 [34] típusú, 935nm fényhullámhosszú infravörös LED-et építettem be az enkóder mellé. Ennek hullámhossza körülbelül megegyezik az enkóder optimális hullámhossz érzékelésével. A LED 50 fokban oldalra világít. Fogyasztását egy ellenállással állítottam be 15mA-re.

Kép: [35][36]



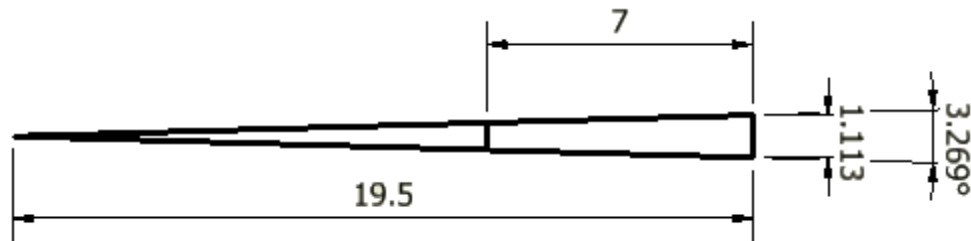
24. ábra HLC2701 enkóder, SEP8506-001 LED

Az inkrementális tárcsa tervezése [37].

Az eddigiekben megfogalmazottak és az eszközök adatlapjainak felhasználásával inkrementális tárcsát terveztem.

Az 55 osztásos inkrementális tárcsa tervezése.

A tárcsa átmérője 400 mm. Az osztások magassága 7mm, így az osztás alja a középponttól 12.5mm-re, a teteje pedig 19.5 mm-re. 110 osztás van a tárcsán. Ekkor fizikailag 55 lépést mérhetünk vele egy körülfordulás alatt.



25. ábra Inkrementális osztás tervezése

A 360 fokot leosztottam 110-zel és így megkaptam, hogy 1 osztás 3.269 fokra van a másiktól. Azért terveztem ilyen nagyra az osztásközöket (az enkóder maximális felbontásához képest), mert 3D-s nyomtatásból, a szerelésből és a minta kinyomtatásából eredő pontatlanságok miatt már nem tudja a maximális felbontást az enkóder. Az enkóder fotodiódáira az osztásoknak pont merőlegesen kell esni, különben nem fog pontos értékeket adni az A és a B csatorna.

Az 55 lépéses fizikai felbontást szoftveresen megnégyszereztem, így egy körülfordulás alatt 220 lépést mér.

A lépési hibát úgy számoltam ki, hogy a robot kerekének területét „Kk” elosztottam az egy kerék körbefordulása alatt mért lépések számával.

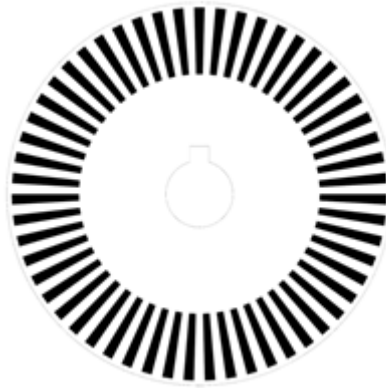
$$Kk = 2 * r * \pi = 2 * 32 * \pi = 201.0619mm \quad (3.9.1)$$

Egy körbefordulás alatt mért lépések: 220.

$$egy\ osztással\ megtett\ út = \frac{201.0619}{220} = 0.91mm$$

A hiba 0.91mm.

A kész inkrementális tárcsa:

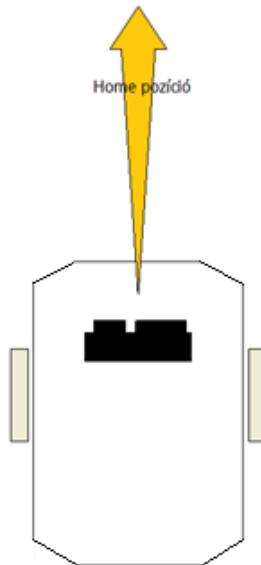


26. ábra Inkrementális tárcsa

4. A parkolás menete lépésről lépésre

4.1 A robot feltérképezi környezetét és kiválasztja a legközelebbi pontot

Elsőként a robot feltérképezi a környező területet, hogy hol találja a legközelebbi pontot.

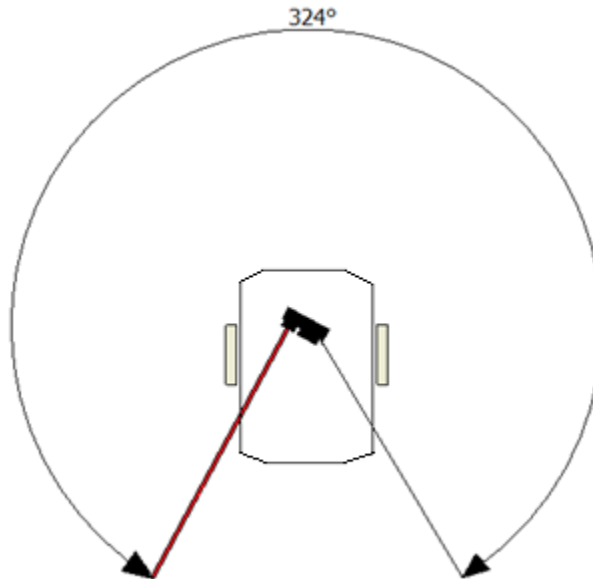


27. ábra A robot Home Pozíciója

A robot alaphelyzetben az úgynevezett Home Pozícióban áll [38]. Ekkor az infravörös szenzor előre néz. Mozgás közben mindig ez a kiinduló pont, mindig ehhez a ponthoz viszonyítja a mozgását. A feltérképezéshez az infravörös távolságszenzort használom, amit a léptető motorral forgatok körbe. Ez a szenzor kiváló az ilyen műveletekhez, mivel csak egy pontban érzékel, az érzékelési tartomány nem növekszik a robottól való távolsággal. A léptető motor egy lépéssel 1.8 fokot lép. A szenzor méréstartománya határán (800 mm) a köríven a két mérési pont közötti távolság 2.73mm.

Az Arduino fejlesztő környezetben írt program úgy térképezi fel a környezetet, hogy a Home Pozícióból balra 90 lépéssel-162 fokkal- elforgatja az infravörös szenzort. Ezután lépésenként forgatja körbe 324 fokot. Ez 180 lépést jelent. A szenzor minden lépés után megméri a távolságot és eltárolja egy 180 elemű tömbben. A szenzor átlagos válaszüzeje 39

milliszekundum, így a két lépés között eltelt időt 50 milliszekundumra állítottam be. A 180-adik lépés után visszaáll Home Pozícióba. A szenzor jel túl zajos volt, ezért szűrőként egy 10 μ F-os kondenzátort iktattam be az áramkörbe.



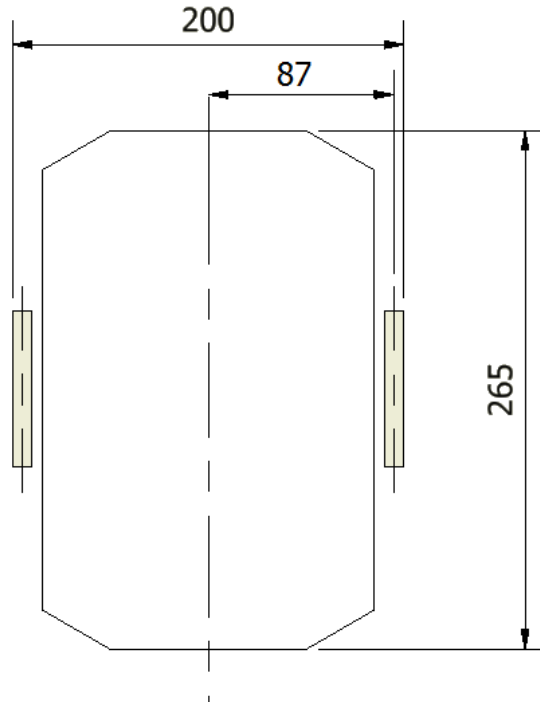
28. ábra Feltérképezés területe

Ezek után a program meghatározza a tömbben, hogy melyik volt a legközelebbi objektum. Infravörös szenzornál a szenzor érzékelőjébe minél erősebb fény jut vissza, annál nagyobb jelet dekódol. A szenzor által mért értékek közül a legnagyobbat keresi meg és az ehhez tartozó sorszámot. Ez a sorszám megmutatja, hogy a Home Pozíciótól hány lépésre helyezkedik el. Ha a sorszám kisebb, mint 90, akkor az $(\text{abs}(0-90+\text{sorszám}))$ képlettel, ha nagyobb, akkor a $(\text{sorszám}-90)$ képlettel állapítja meg a keresett legközelebbi objektumot.

4.2 Abba az irányba fordul, amerre a legközelebbi falrész van

Második lépésként a robot abba az irányba fordul, amerre a legközelebbi objektumot észlelte.

A robot befoglaló méretei:



29. ábra A robot befoglaló méretei

Meghatároztam a robot fordulókörét. Mivel ez egy differenciál hajtású robot és azáltal, hogy a kerekek pontosan a váz hosszúságának felénél helyezkednek el, a robot egyhelyben képes megfordulni.

A fordulókör átmérője 174mm. Ebből kiszámolható a fordulókör kerülete „K”.

$$K = 2 * R * \pi$$

$$K = 2 * 87 * \pi = 546.63mm$$

Az így kapott értéket felhasználva kiszámoltam az egy léptető lépés által megtett körívet „C”.

$$C = K / \frac{360}{1.8} \quad (4.2.1)$$

$$C = \frac{546.63}{\frac{360}{1.8}} = 2.733mm$$

A kapott körív értéke 2.733mm.

A parkolás menetének első részénél a program meghatározza, hogy a Home Pozíciótól balra, vagy jobbra hány lépésnyit kell elfordulni ahhoz, hogy pontosan a legközelebbi objektum felé forduljon a robot. Az ott használt képleteket és még egyéb képleteket felhasználva kiszámoltam, hogy az inkrementális tárcsán hány osztásnyit kell egyik, vagy másik irányba menni kerekenként.

Ha sorszám kisebb, mint 90, akkor:

$$x1 = (abs(0 - 90 + sorszám) * 2.733)[mm] \quad (4.2.2)$$

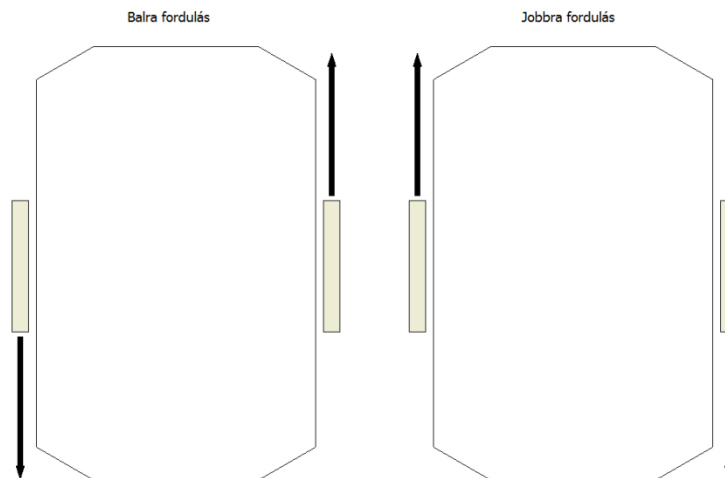
Ha pedig a sorszám nagyobb, mint 90, akkor:

$$x2 = ((maximum - 90) * 2.733)[mm] \quad (4.2.3)$$

X1 milliméternyit kell balra, vagy x2 milliméternyit kell jobbra fordulni. Ahhoz, hogy megkapjam az inkrementális osztásszámot „O” az imént kapott értékeket el kell osztani az egy osztáshoz szükséges kerék elfordulással.

$$O = \frac{x}{0.9139} \quad (4.2.4)$$

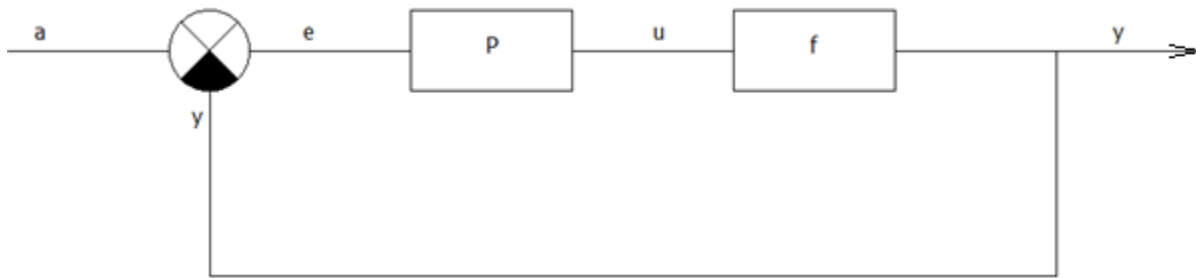
Ha a sorszám kisebb 90, akkor „O” osztásnyit kell a robotnak balra, ha pedig nagyobb 90, akkor jobbra fordulnia. Forduláskor a kerekek egymással ellenkező irányba forognak. A robot balra fordulásánál a bal kerék hátra, a jobb előre, míg a jobbra fordulásánál a bal előre és a jobb hátra forog.



30. ábra Fordulás mechanizmusai

A forgást nem akkor kell abbahagyni, amikor az inkrementális számlálás hibája elérte a nullát, mert akkor a robot a tehetetlensége miatt továbbfut és a célállomástól és akár több centimétert is megtehet. E probléma megoldására P szabályzást alkalmazok a mozgás során. Azért választottam a P szabályzást, mert a rendszernek nagy a tehetetlensége.

A P szabályzás blokkábrája.



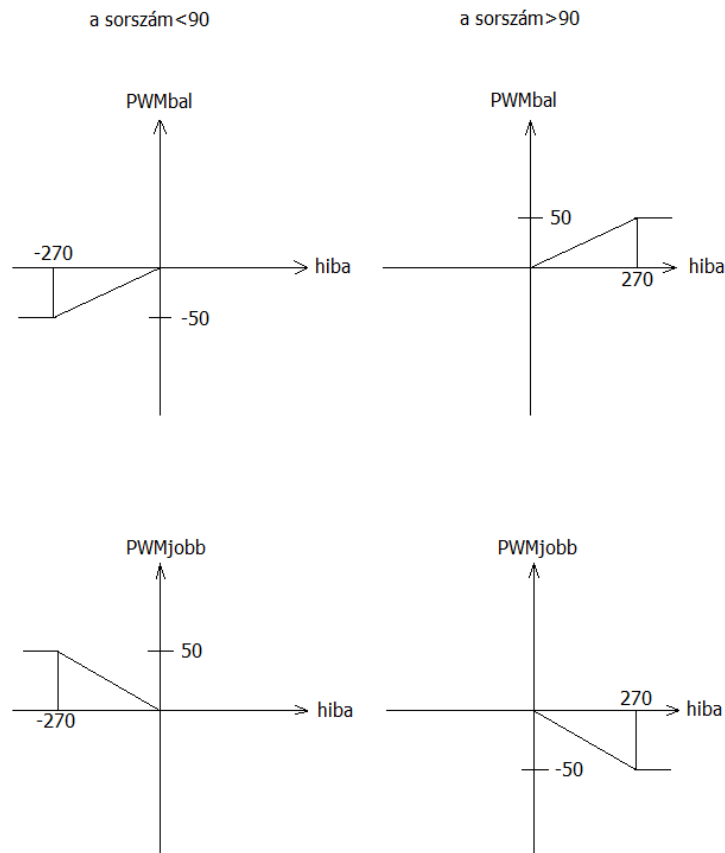
31. ábra P szabályzó hatásvázlata

Jelmagyarázat:

- a – szabályozott jellemző
- e - hiba (a-y)
- P – P tag
- u - P-vel beszorzott jel
- f - folyamat

Az „a”-hoz közelít a rendszer. A kimenetre adott jel vissza van csatolva. A hiba az „a” jelből kivont visszacsatolási jelből áll. A P tag egy szorzó, amely felerősíti a hibát. Minél nagyobb a P, annál gyorsabban megy végbe a folyamat. A szabályzás addig működik, amíg a hiba nem lesz nulla.

A szabályzási függvény meghatározására az alábbi diagramokat csináltam:



32. ábra Kanyarodási szabályzási függvények

Ezekon a diagramokon jól látszik, hogy milyen hibaértékeknél milyen értéket vesz fel a szabályzás. A függvény meredekségét kell meghatározni. Az x tengelyen a hibát, az y tengelyen a PWM jelet ábrázoltam. Negatív PWM jel önmagában nincs. Ha a függvény PWM értéke pozitív, akkor az irány is pozitív, ha pedig negatív, akkor az irány is negatív. Balra és jobbra is 270 inkrementális osztásnyit (137 fok) kell a robotnak fordulnia, hogy a teljes észlelési felületen fordulni tudjon.

Ehhez a megoldó képlet:

$$y = a * x + b \quad (4.2.5)$$

A monoton lineáris növekvő és csökkenő szakaszok kezdő és végpontjait felhasználva két ismeretlenes egyenletrendszert írtam fel az első koordináta-rendszerben lévő lineáris szakaszra (nem vízszintes).

$$I. 0 = 0x + b$$

$$II. -50 = -270x + b \quad (4.2.6)$$

Ezt a két függvényt kivonjuk egymásból, így a „b” tagok kiesnek. Ekkor egy egyenlet lesz egy ismeretlenre. Ebből ki tudjuk számolni az „x”-et.

$$-20 = -270x$$

$$x = \frac{50}{270} = 0.185$$

Az „x” értéke attól függően vesz fel pozitív és negatív értékeket, hogy a függvény növekvő, vagy csökkenő.

Így, ha a sorszám <90, akkor $PWM_{bal} = 0.185x$

$$PWM_{jobb} = -0.185x$$

Ha pedig a sorszám >90, akkor $PWM_{bal} = 0.185x$

$$PWM_{jobb} = -0.185x$$

Az „x” paraméter helyére az „e” hibát helyettesítem be. A különbség a két fordulás között, hogy bal felé történő forgásnál, amikor a bal kerék hátrafelé hajt, akkor a szabályzási hibát és az „a”-t nem kivonni kell egymásból, hanem összeadni, mert a kerék hátrafelé forgásánál az inkrementális jelek negatívak lesznek. Jobbra forduláskor ugyanígy. Ha a jobb kerék fordul hátra, akkor ott is szabályozott jellemzőt és a hiba jeleket összegezni kell. A diagramokon ez úgy látszik, hogy balra forduláskor a PWM_{bal}-nál a lineáris függvény az „x” tengely alatti tartományban van, míg PWM_{jobb}-nál felette. Jobbra forduláskor pont az ellentéte a helyzet.

Minél nagyobb P értéket adunk meg a szabályzásba, annál meredekebb lesz a függvény. Ezért kellett a szabályzás P paraméterét kísérleti úton beállítani, hogy a robot pont ott álljon meg, ahol a hiba nulla lesz.

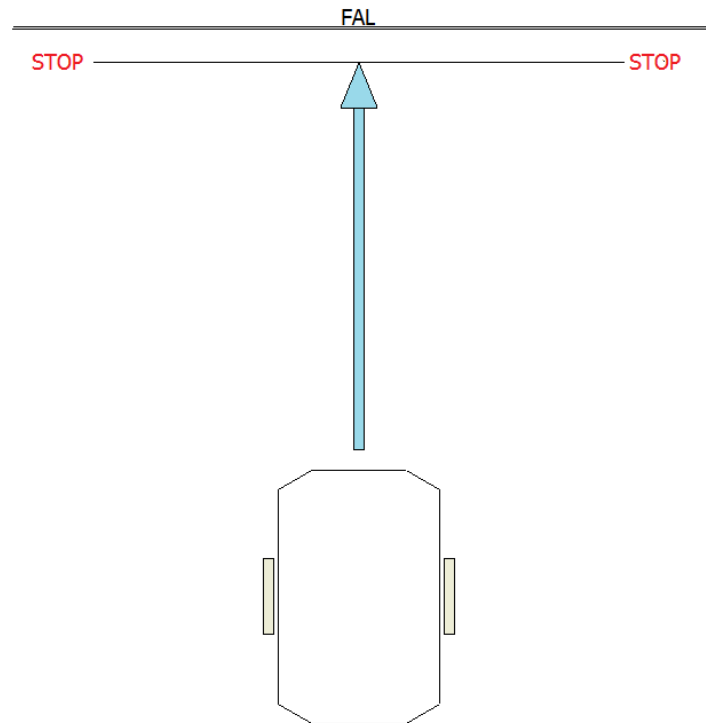
A PWM jele 0-255 közötti értéket vehet fel. A 0 értéknél a motorok állnak, a 255-ös értéknél maximális teljesítménnyel működnek. Azért korlátoztam 50-re a jelet, mert így lassabban fordul a robot és ezáltal nagyon lecsökken a tehetetlenségből adódó pontatlanság.

Ha a szabályzás azt érzékeli, hogy a robot túlfut a megadott szabályozott jellemzőn, akkor visszatolat a megfelelő pontig. (Amennyire a mérési hiba engedi.)

Ez az én robotom esetében viszonylag jó pontossági beállást eredményez.

4.3 Elindul a fal felé és attól 35 mm-re megáll

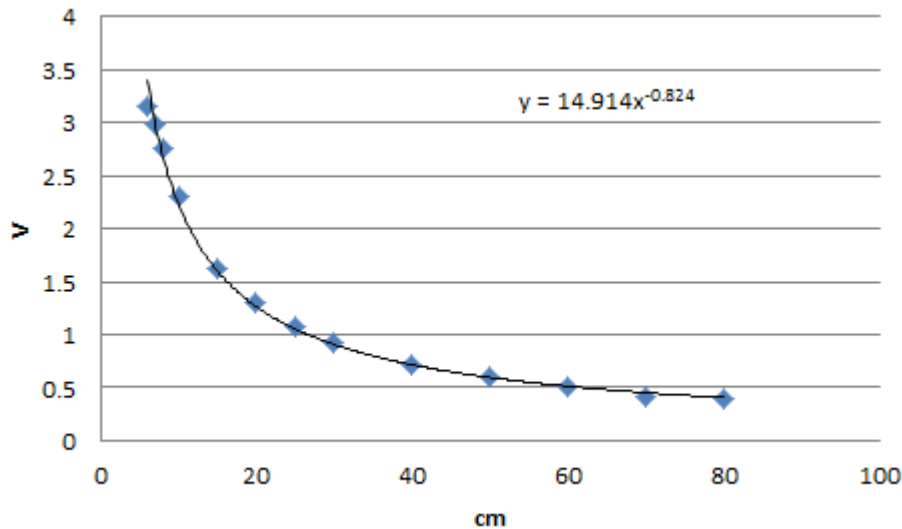
Harmadik lépésként a mért infravörös szenzor értéket felhasználva a robot elindul a fal felé és onnan 35mm-re megáll.



33. ábra A robot a faltól 35 milliméterre áll meg

A program megnézi, hogy a legközelebbi objektum tömbben tárolt sorszámához milyen érték tartozik. Ahhoz, hogy ki tudjam számolni a pontos távolságot, az infravörös szenzor kimeneti feszültség – távolság karakterisztikájára függvényt kellett illesztenem. Az adatlapban ábrázolt függvény jól leolvasható „x” és „y” koordinátáit beírtam egy Excel táblázatba. Ezekből a koordinátákból egy ponthalmazt rajzoltattam le. Az Excel beépített programja segítségével ráillesztettem egy görbét az ábrázolt pontokra. Ennek egyenlete:

$$y = 14.914 * x^{-0.824} \quad (4.3.1)$$



34. ábra Infravörös szenzorra illesztett görbe

1024-es analóg skálán a 0.01V megközelítőleg 2 egységnek felel meg. A szenzor által adott legnagyobb feszültség 3.25V, ez 650-es analóg jelerősség. Ahhoz, hogy megkapjam a mért analóg jelből a feszültséget, a következő képletet alkalmaztam.

Z - legnagyobb mért jel.

V – voltban meghatározott érték

$$V = Z/200 \quad (4.3.2)$$

Az „y” tengely értékét megkaptam. Az „x” tengelyen lévő értéket úgy határoztam meg, hogy átrendeztem az illesztett görbe egyenletét, úgy, hogy az „x” tengely kerüljön a bal oldalra. A kapott egyenlet:

$$x = {}^{-0.824}\sqrt{\left(\frac{14.914}{V}\right)} \quad (4.3.3)$$

A kapott eredménynek az inverzét kellett venni. Így centiméterben megkaptam a távolságot. „xj”

$$xj = 1/x \quad (4.3.4)$$

Beszoroztam az „xj”-t 10-zel, ekkor milliméterben kaptam meg az eredményt.

Ahhoz, hogy a robot a faltól 35mm-re álljon meg, a mért értékből ki kell vonni 100mm-t. Azért 100 millimétert, mert az ultrahangos szenzor a robot elejétől 65mm-re helyezkedik el. A tízes beszorzás után kapott eredményt, xm-et elosztotta az egy osztásnyi elmozduláshoz szükséges értékkel, 0.9139-cel. A végeredményt „L”-lel jelöltem.

$$L = \frac{xm}{10} \quad (4.3.5)$$

A parkolás második lépésénél használt P szabályzót alkalmazom ebben a lépésben is. Egy kicsit átalakítva. A szenzor maximális látótávolsága miatt a robot 800mm-t tud pontosan megtenni. Az ehhez szükséges inkrementális lépések számát az alábbi módon számoltam ki és az eredményt egészre kerekítve adtam meg.

$$\frac{800}{0.9139} = 875$$

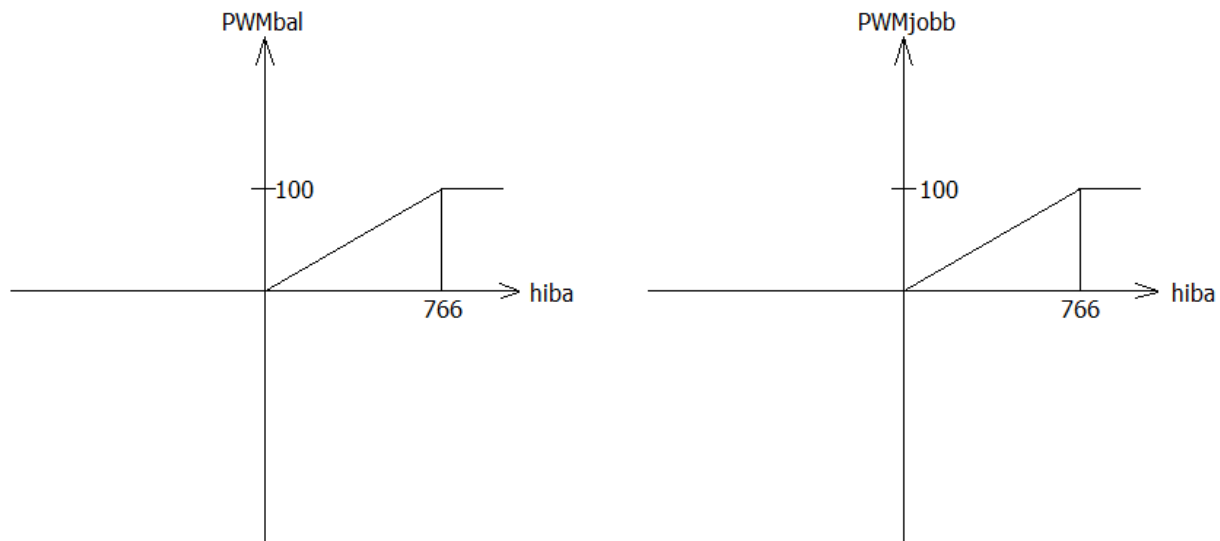
Ebből az értékből ki kell vonni azt a 100 mm-nyi utat, amit a robot biztos, hogy nem fog megtenni a folyamat során. A 100 mm út alatti osztásszám egészre kerekítve:

$$\frac{100}{0.9139} = 109$$

Ezután megkaptam azt az osztásszámot, amit a valóságban maximálisan mehet. A maximális osztásszámot „L1”-gyel jelöltem.

$$L1 = 875 - 109 = 766$$

Az „L1” révén meghatároztam a P szabályzás egyenletét mindkét motorra.



35. ábra Előrehaladási szabályzás függvénye

Egy két ismeretlenes egyenletrendszerből számoltam ki az x értéket. Ezt a bal oldali lineáris függvényre írtam fel.

A két ismeretlenes egyenletrendszer:

$$\begin{aligned}
 0y &= 0x + b \\
 100y &= 766x + b
 \end{aligned}
 \tag{4.3.6}$$

A két egyenletet kivonva egymásból és kifejezve az x-et megkapjuk a jobb és bal oldali függvényt.

$$PWM_{bal} = 0.1305x$$

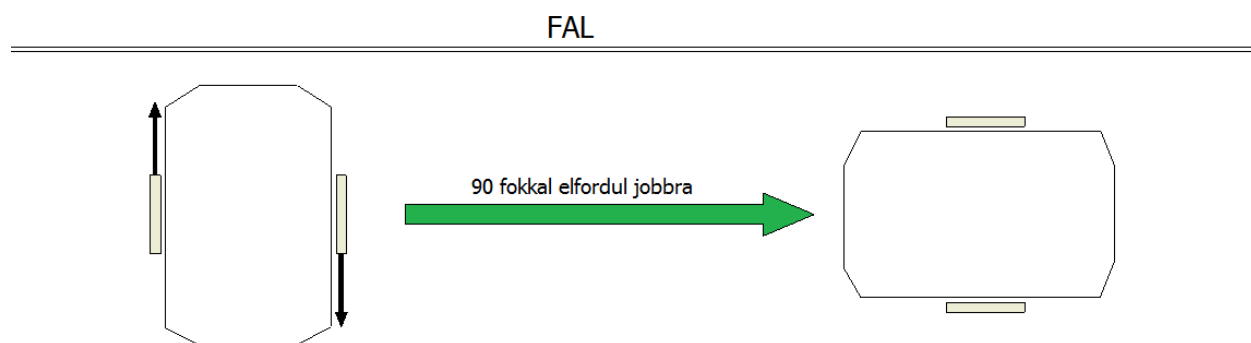
$$PWM_{jobb} = 0.1305x$$

Mindkét függvénynél a maximális PWM korlátozva van 100-on.

A szabályzás metodikáját úgy csináltam meg, mint az előző lépésben. Ha a robot továbbfut a nulla hibán, akkor visszatolat.

4.4 Elfordul a 90 fokkal jobbra a fallal párhuzamosan

Negyedik lépésként a robot elfordul jobbra a fallal párhuzamosan.



36. ábra Befordulás a fallal párhuzamosan

Itt a parkolás második lépésénél (4.2) leírt P szabályzást alkalmazom. Annak paraméterei majdnem megegyeznek az itt használtakkal. A különbség annyi, hogy itt egy fix „a”-t kell követnie a szabályzásnak és nem kell minden mérés után újraszámolni.

A fordulásnál használt szabályozott jellemzőt a fordulási körből „K” határoztam meg. Mivel 90 fokkal kell elfordulnia a robotnak, ezért a fordulási kört el kell osztani 4-gyel.

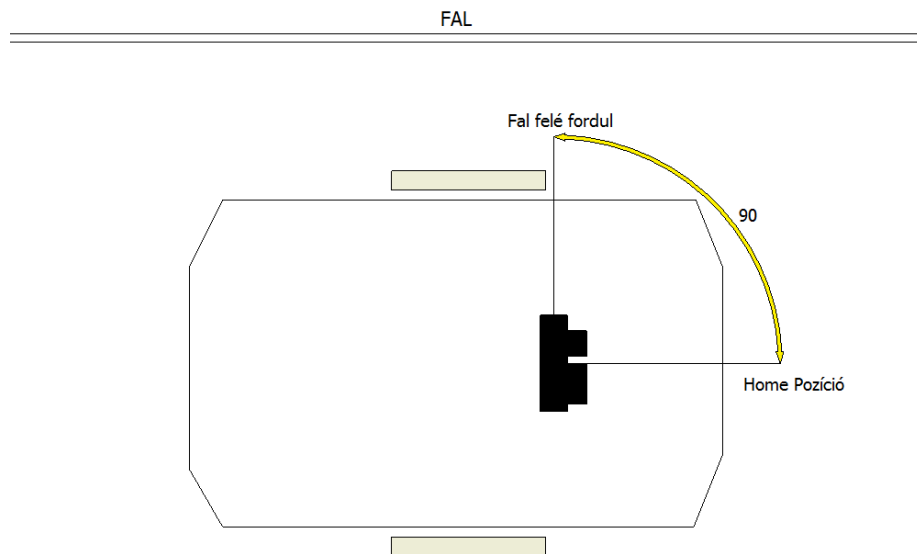
$$\frac{K}{4} = \frac{546.63}{4} = 136.6575mm$$

A kapott számot még el kell osztani az egy inkrementális osztásnyi elforduláshoz szükséges úthosszal, ami 0.9139mm.

$$\frac{136.6575}{0.9139} \approx 150 \text{ osztás}$$

150 osztás kell a bal és a jobb keréknek is elfordulnia ahhoz, hogy a robot pontosan a fallal párhuzamosan álljon. A túllépésnél most is igaz, hogy visszatolat a robot.

4.5 Az infravörös szenzort elforgatja a fal felé 90 fokkal



37. ábra Az infravörös szenzor elforgatása a fal felé

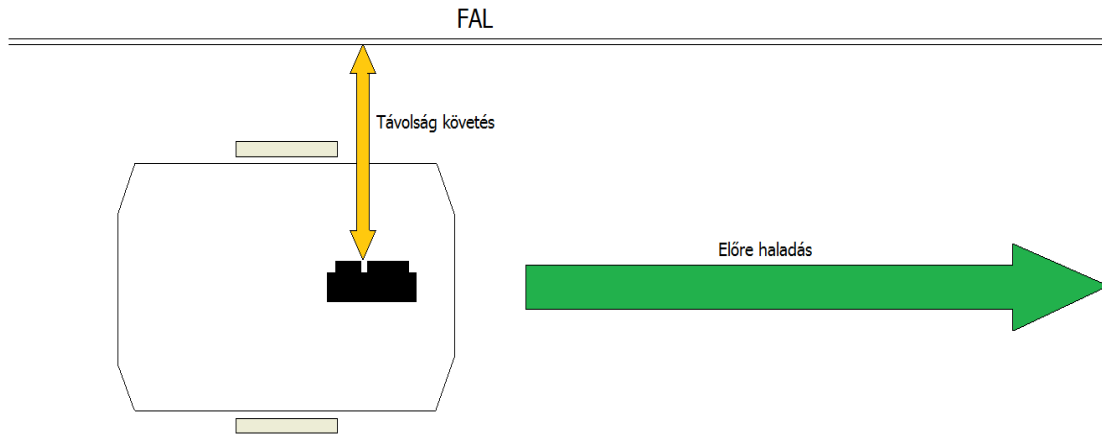
Ötödik lépésként a fal felé forgatom az infravörös szenzort a Home Pozíciótól 90 fokkal balra. Az első lépésben (4.1) alkalmazottak alapján csináltam meg. A léptetőmotor lépésszámát úgy kapom meg, hogy a 90 fokot elosztom 1.8-cal.

$$\frac{90}{1.8} = 50$$

A robotnak 50 lépést kell a balra fordulni.

4.6 A robot elindul a fal mellett egy adott távolságban követve azt

Hatodik lépésben megvalósítom a falkövetést.



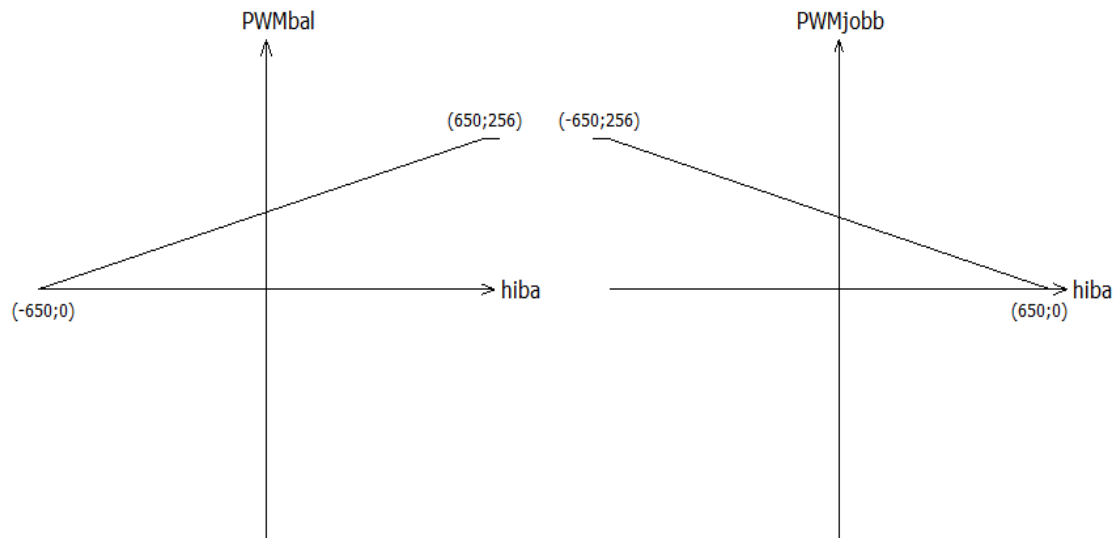
38. ábra Falkövetés

A 4.4-es parkolási lépésben a robot beállt a faltól 85mm távolságra. A szenzor ilyenkor 165 milliméterre van a faltól. A szabályozott jellemző 165mm. Ezt a távolságot tarja a faltól a robot.

Ismét az infravörös szenzort használok. A robot fő feladata, hogy egy megadott távolságban kövesse a falat. Ha valami zavar lép fel egy kis időre, azt is tudja korrigálni. A falkövetésre szintén P szabályzót használok. Ebben a részben egy kicsit át kellett dolgozni a szabályzási paramétereket.

A fő probléma abból adódott, hogy a robot nem tudott pontosan párhuzamosan beállni a fallal. Ilyenkor érzékelte, hogy a robot mm pontosan a megadott távolágban van. A hiba milyenségétől függően vagy a bal, vagy a jobb kereket kezdte el nagyon gyorsan forgatni és nekiment a falnak. Ennek korrigálására egy olyan erősségű P tagot és olyan függvényeket használok, amely nem engedi meg a hirtelen irányváltást.

A falkövetés PWM szabályzása és a hiba függvénye.



39. ábra Falkövető szabályzás függvényei

A baloldali ábráról az olvasható le, hogy ha a szenzor közeli objektumot érzékel, akkor nagy lesz a jelerősség, nagy lesz a hiba, ekkor a baloldali kerék gyorsan fog hajtani. Ha messze van az objektumtól, akkor kicsi lesz a jelerősség, így a hiba is csökkeni fog, a kerék lassabban hajt. A jobboldali ábráról ennek a fordítottja olvasható le. Minél közelebb van a robot az objektumhoz, annál lassabban hajt a jobb kerék és minél messzebb van, annál gyorsabban. A kerekek sebességét a függvény határozza meg.

A két függvény egyenletének meghatározása két ismeretlenes egyenletrendszerből:

A PWMbal egyenlete:

$$I. 0 = -650x + b$$

$$II. 256 = 650x + b \quad (4.6.1)$$

A két egyenletet összeadva és abból kifejezve megkapjuk b-t:

$$256 = 2b$$

$$b = 128$$

A b-t visszahelyettesítve a második függvénybe meghatároztam az x-et:

$$256 = 650x + 128$$

$$128 = 650x$$

$$x = \frac{128}{650} = 0.197$$

A PWMbal egyenlete:

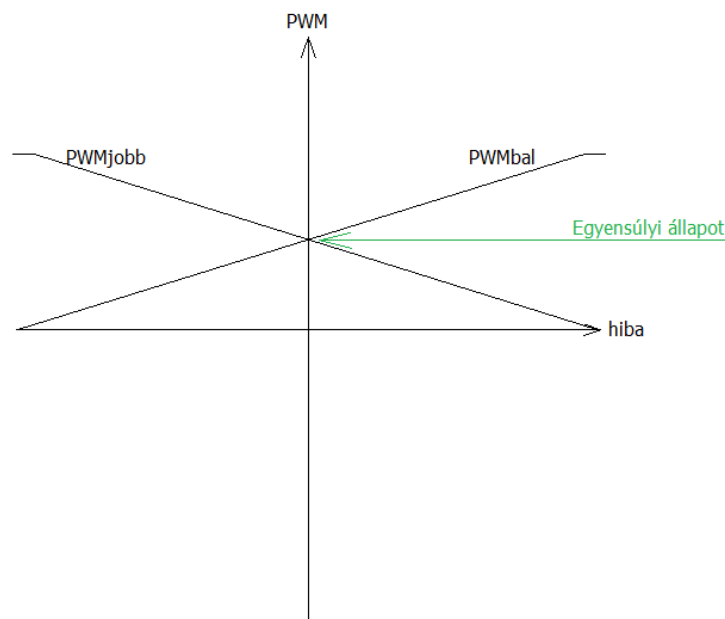
$$PWMbal = 0.197x + 128$$

A PWMjobb egyenleté: Ugyanilyen módon számoltam. Az ábrán is jól látszik, hogy a PWMjobb egyenes lejt. Ebből következik, hogy a meredekségi paramétere előjelet vált.

A PWMjobb egyenlete:

$$PWMjobb = -0.197x + 128$$

Az egyensúlyi pont ott van, ahol a két egyenes metszi az y tengelyt. A szabályzás mindig ebbe az állapotba próbálja vinni a robotot.



40. ábra Falkövető szabályzás egyensúlyi állapota

4.7 Falkövetés közben helyet keres

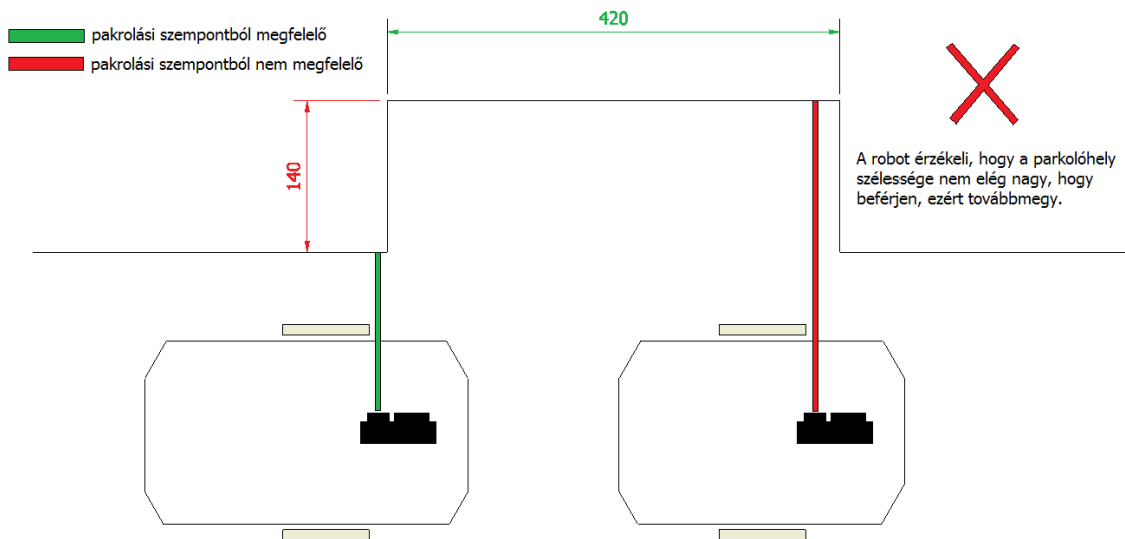
A parkolási folyamat hetedik lépése, hogy a robot falkövetés közben parkolásra alkalmas helyet keres magának.

A folyamat úgy zajlik, hogy a robot falkövetés közben folyamatosan vizsgálja az infravörös szenzorral a távolságot. Ha a távolság ugrásszerűen megnő, akkor a falkövető szabályzás megszűnik és a robot mindkét kerekére azonos PWM jel kerül. Ily módon a robot előre halad.

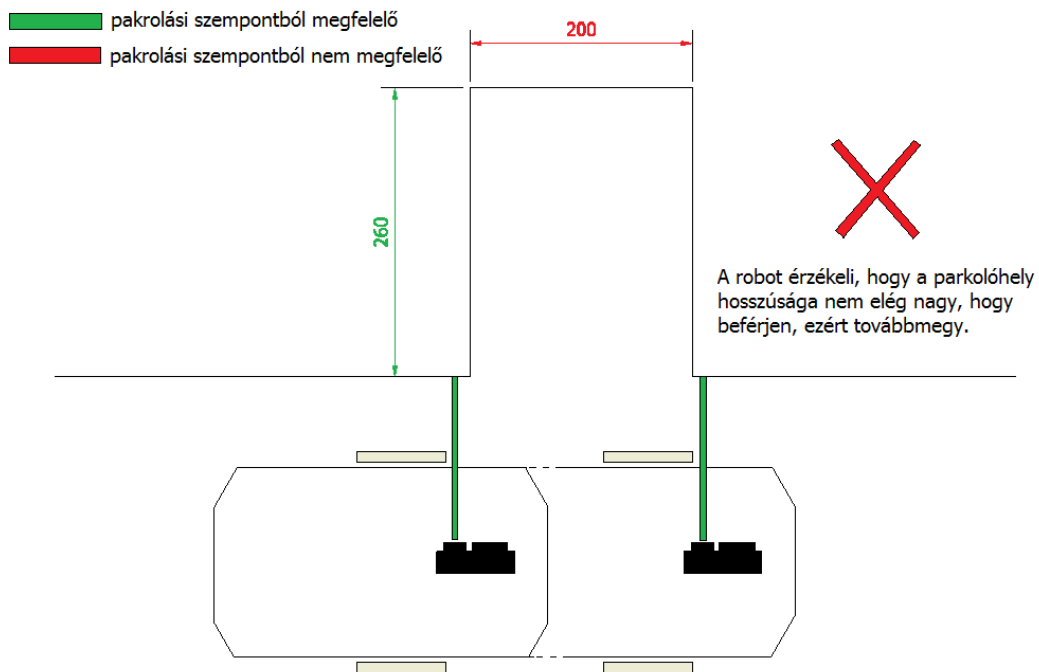
Haladás közben folyamatosan vizsgálja a távolságot és méri az inkrementális tárcsán az elfordulási osztátszámot. Ha ugrásszerű távolságcsökkenést érzékel (újra látja a falat), akkor a robot megáll. A hirtelen távolságcsökkenéskor csak azokat a szenzorértékeket veszi figyelembe, amelyek a szabályozott jellemző közelébe esnek. Az enkóder jeleit felhasználva kiszámolja, hogy mekkora a két végpont közötti távolság. Ha az kisebb, mint a parkoláshoz szükséges legkisebb táv, akkor oda nem parkol be, hanem tovább megy és másik parkoló helyet keres.

Van egy másik tényező is, amely nagyban meghatározza, hogy a robot be tud-e parkolni a kiválasztott helyre. Ez a tényező pedig a parkoláshoz szükséges hely mélysége, azaz az a távolság, amely ahhoz kell, hogy a robot teljes szélességében beférjen az adott helyre. Ezt nagyon egyszerűen kiméri a mobilrobot. Amikor már nem a szabályzási program fut, hanem a pakolási helymérés program, akkor a szenzor értékeit folyamatosan figyeli a mikrokontrollerbe írt program. Ha a szenzor bármely értéke egy megadott pont fölé esik, akkor a robot oda nem fér be. Ilyenkor továbbmegy, és másik helyet keres.

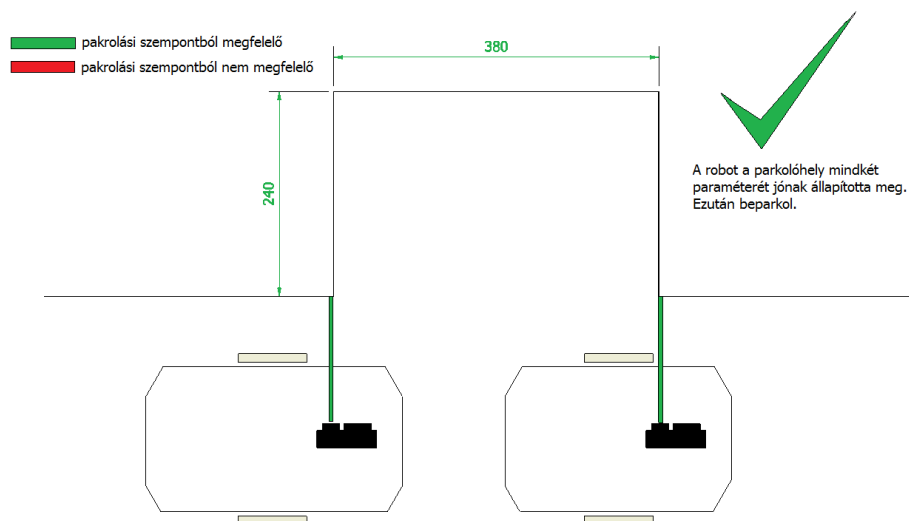
Példák a parkolási hely kiválasztására:



41. ábra Helykeresés 1



42. ábra Helykeresés 2

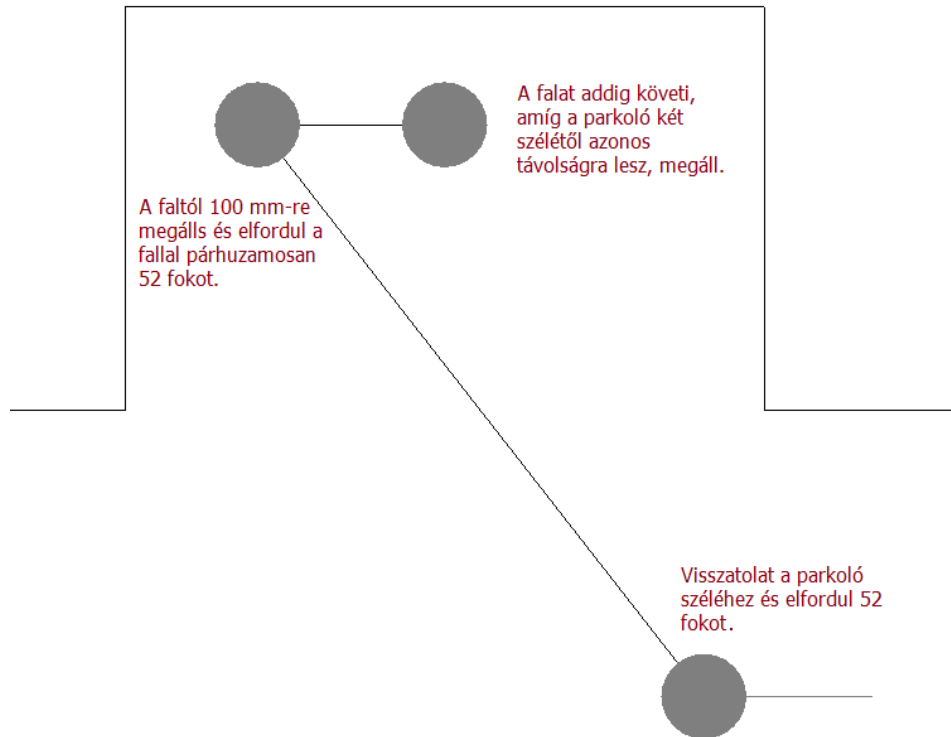


43. ábra Helykeresés 3

További korlátozások a helykeresés közben: a parkoló maximális hossza 450 milliméter szélessége 260 milliméter.

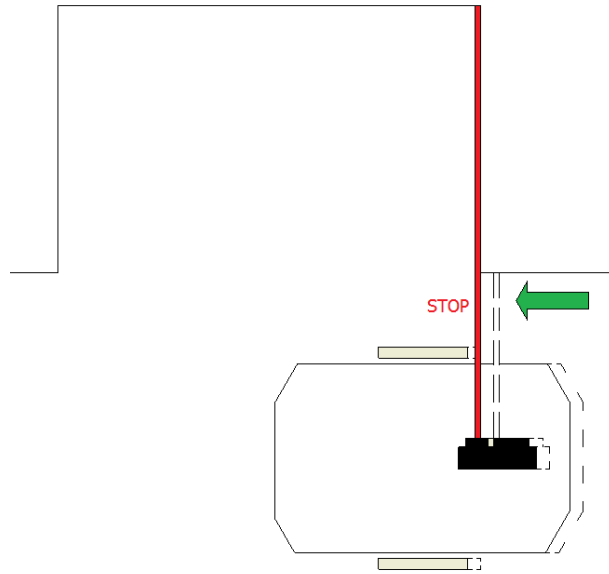
4.8 Ha megtalálja a helyet, akkor beparkol

Nyolcadik lépésként a robot beparkol a kiválasztott helyre. Ezt több lépésben valósítja meg [39].



44. ábra Beparkolás lépései

Elsőként nagyon lassan hátrátolat addig, amíg az infravörös szenzor ugrásszerű távolság növekedést nem érzékel. Ekkor megáll. A robot tudja, hogy a parkolóhely innenső szélén van éppen. A robot ismert hosszúsági (265mm) és szélességi (200mm) paramétereit alapján meghatároztam, hogy mekkora az a hely, ahová még kényelmesen befér az 360 mm hosszú és 250 mm széles.



45. ábra Beparkolás első lépése – falszegélynél megállás

Második lépésként elfordul 52 fokot úgy, hogy az eleje kifelé álljon. A fordulásnál a parkolás második és negyedik lépésben alkalmazott P szabályzást alkalmazom ugyanazokkal a paraméterekkel.

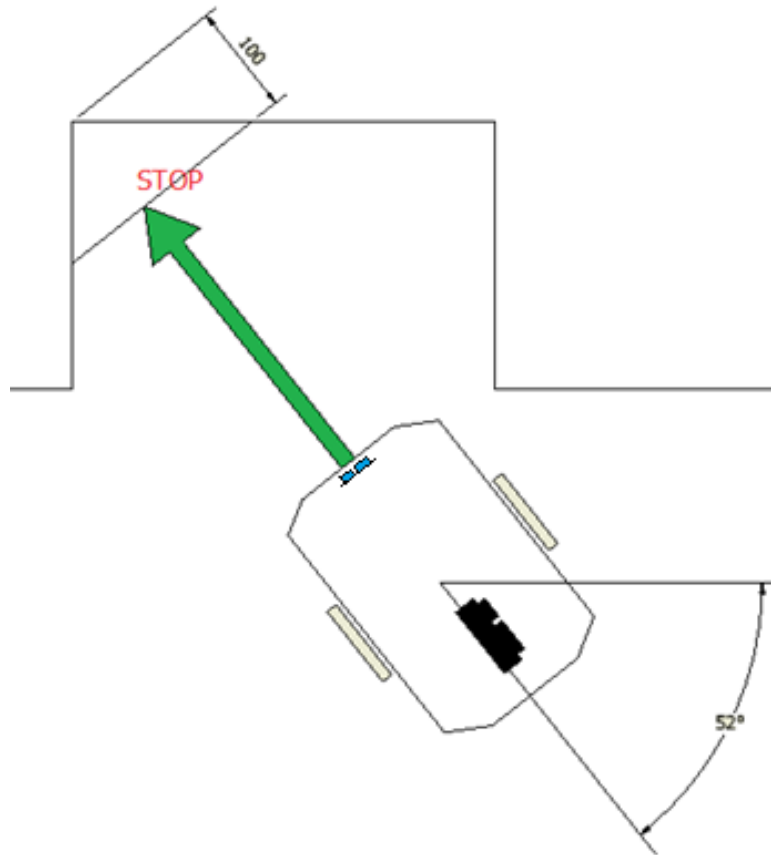
Az inkrementális lépésszám kiszámolása:

$$\frac{K}{360} * 52 = \frac{546.63}{360} * 52 = 78.96mm \quad (4.8.1)$$

Ezt elosztva az egy inkrementnyi osztáshoz szükséges távolsággal:

$$\frac{78.96}{0.9139} \approx 86 \text{ osztás}$$

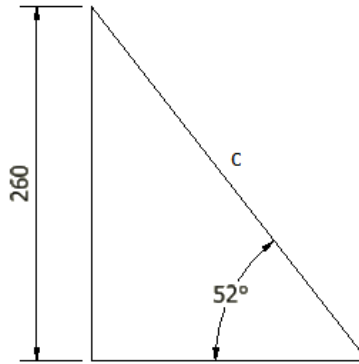
Az inkrementális lépésszám 86.



46. ábra Beparkolás második és harmadik lépése – elfordulás 52 fokkal, majd tolatás

Harmadik lépésként nagyon lassan elkezd tolatni. A hátsó ultrahangos szenzorral folyamatosan érzékeli a faltól való távolságot. 100 milliméterre a faltól megáll. A tolatásnál is P szabályzót használok. A harmadik parkolási lépében használt szabályzási függvényeket használom egy kissé átalakítva, a minél pontosabb pozicionálás érdekében.

Mivel e parkolási módban a robot hosszában áll be a parkolóhelyre, az optimális helykihasználás érdekében a túl „mély” helyekre nem így parkol be, így korlátoztam a parkolóhely szélességét maximum 260 mm-ben. Ezt átszámoltam az inkrementális osztásnyi távolságba.



47. ábra Maximális tolatási úthossz meghatározása

Az inkrementális lépések számát úgy kaptam meg, hogy a maximálisan 52 fokos derékszögű háromszög átfogóját kiszámoltam. Az egyszerű képlet rá:

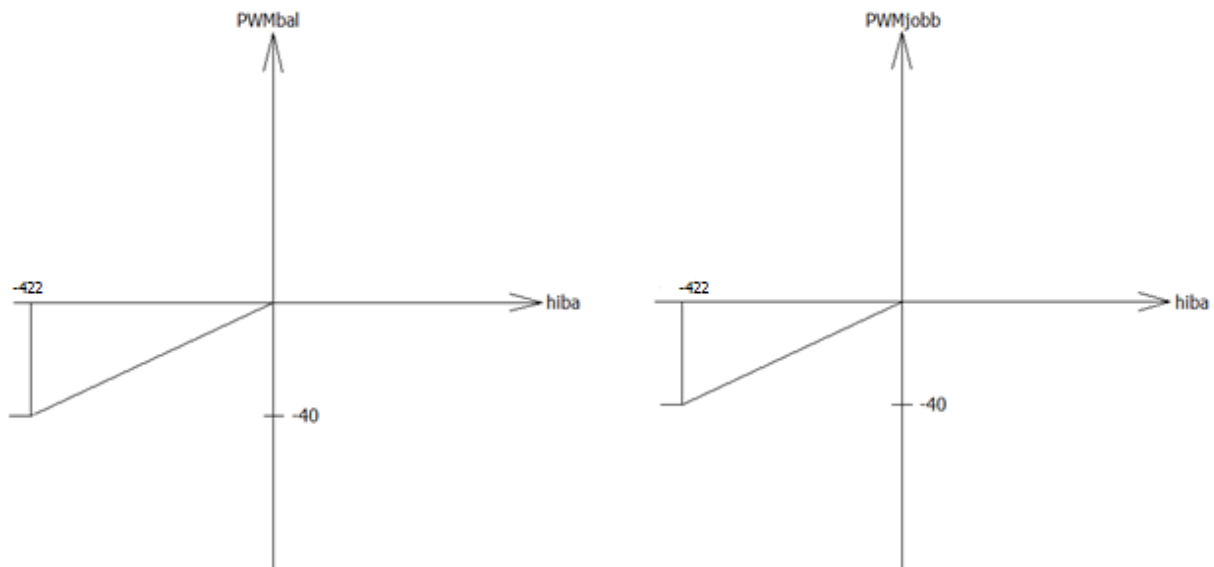
$$\cos 52^\circ = \frac{260}{c} \quad (4.8.2)$$

Ezt átrendezve:

$$c = \frac{260}{\cos 52^\circ} \approx 422 \text{ osztás}$$

A nagyon lassú hátramenet miatt a PWM-ek jelét 40-re maximalizáltam.

A kapott függvények:



48. ábra Tolatási szabályzás függvényei

A PWM-eket mínusszal jelöltem ezeken az ábrákon is. Ezzel szemléltetve, hogy a motorok a robot normális haladási irányával szemben mozognak, azaz hátra. Ilyenkor a PWM iránya alacsony.

A függvények egyenletét az eddigi módszerrel, két ismeretlenes egyenletrendszert alkotva határoztam meg.

A függvények:

$$\begin{aligned} I. -40 &= -422x + b \\ 0 &= 0x + b \end{aligned} \tag{4.8.3}$$

A két függvény kivontam egymásból és ezután kifejeztem az x -et:

$$\begin{aligned} -40 &= -422x \\ x &= \frac{40}{422} \end{aligned}$$

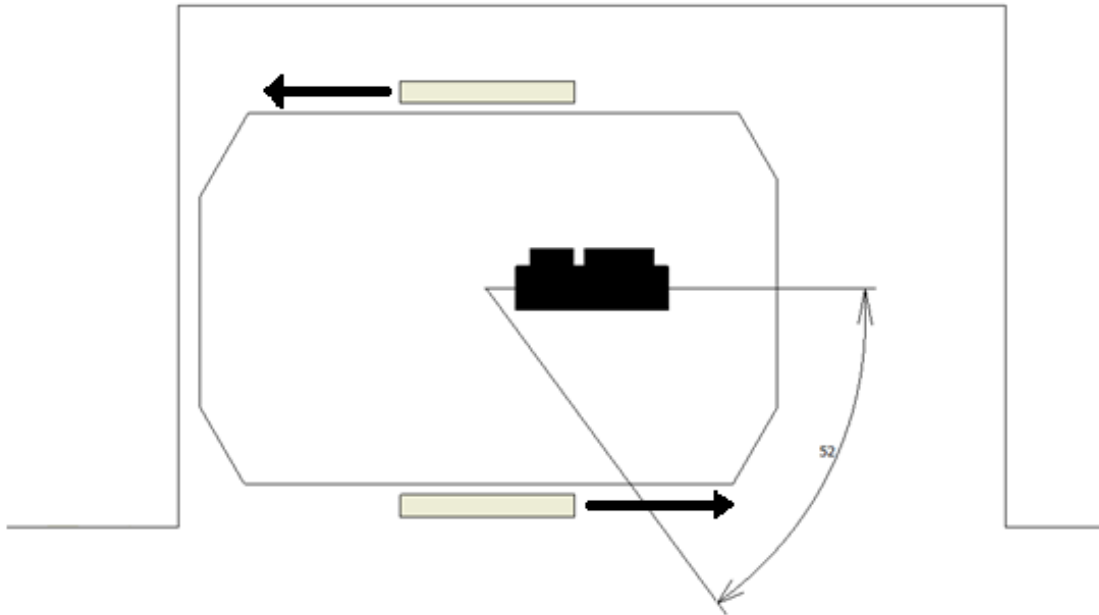
Mivel mindkét függvény ugyanaz, így egyik az egyik, másik a másik motor szabályzását ábrázolja.

Képletük:

$$PWM_{bal} = PWM_{jobb} = 0.0948x$$

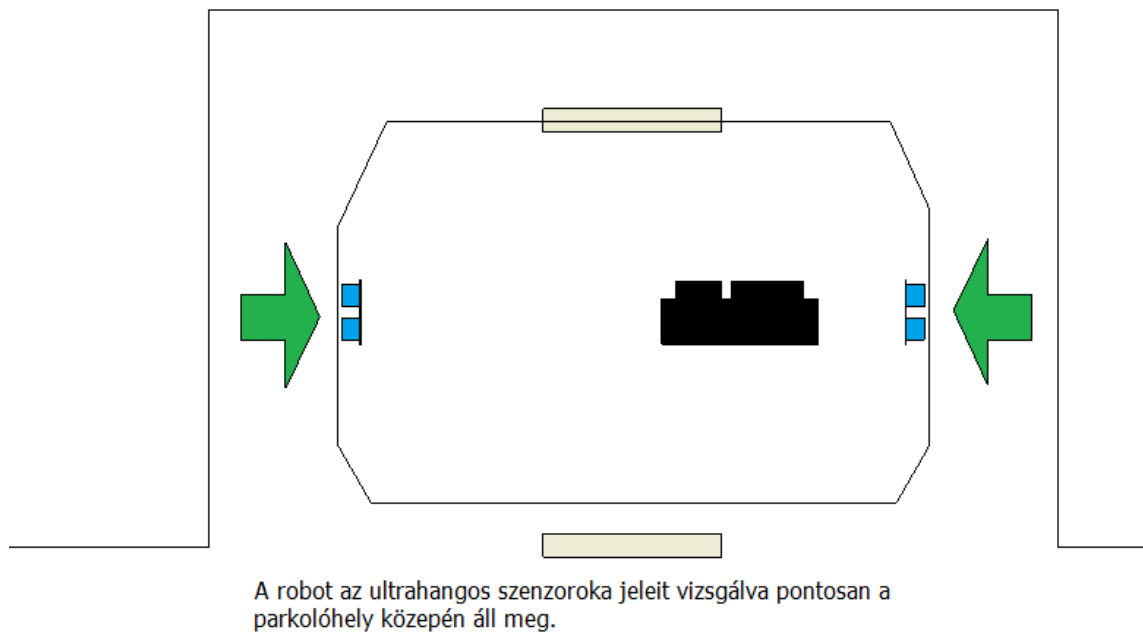
Mindkét keréknek pontosan ugyanannyi távolságot kell megtennie.

Negyedik lépésként a robot 52 fokot fordul és így beáll a fallal párhuzamosan, hosszában; Ugyan azt a szabályzást használva, mint a második metódusban. Az inkrementális osztásszám is megegyezik, 86.



49. ábra Beparkolás negyedik lépése – elfordulás 52 fokkal a fallal párhuzamosan

Ötödik lépésként pontosan a parkolóhely közepén áll meg. Ehhez az első és a hátsó ultrahangos szenzort használja. Az előre és hátra haladáshoz a falkövető szabályzó algoritmusát kicsit átterveztem. Annyi módosítást végeztem rajta, hogy most nem az infravörös szenzor jele a hiba, hanem az ultrahangos szenzorok jele. A hibát úgy határozom meg, hogy a két ultrahangos szenzor jelét kivonom egymásból. Ha az így kapott érték pozitív, akkor a hátsó szenzorhoz van közelebb a robot, ilyenkor előrébb kell mennie, ha pedig az eredmény negatív, akkor az első szenzorhoz van közelebb, ilyenkor hátrébb kell tolatnia. A szabályzás a nulla értékre törekszik.

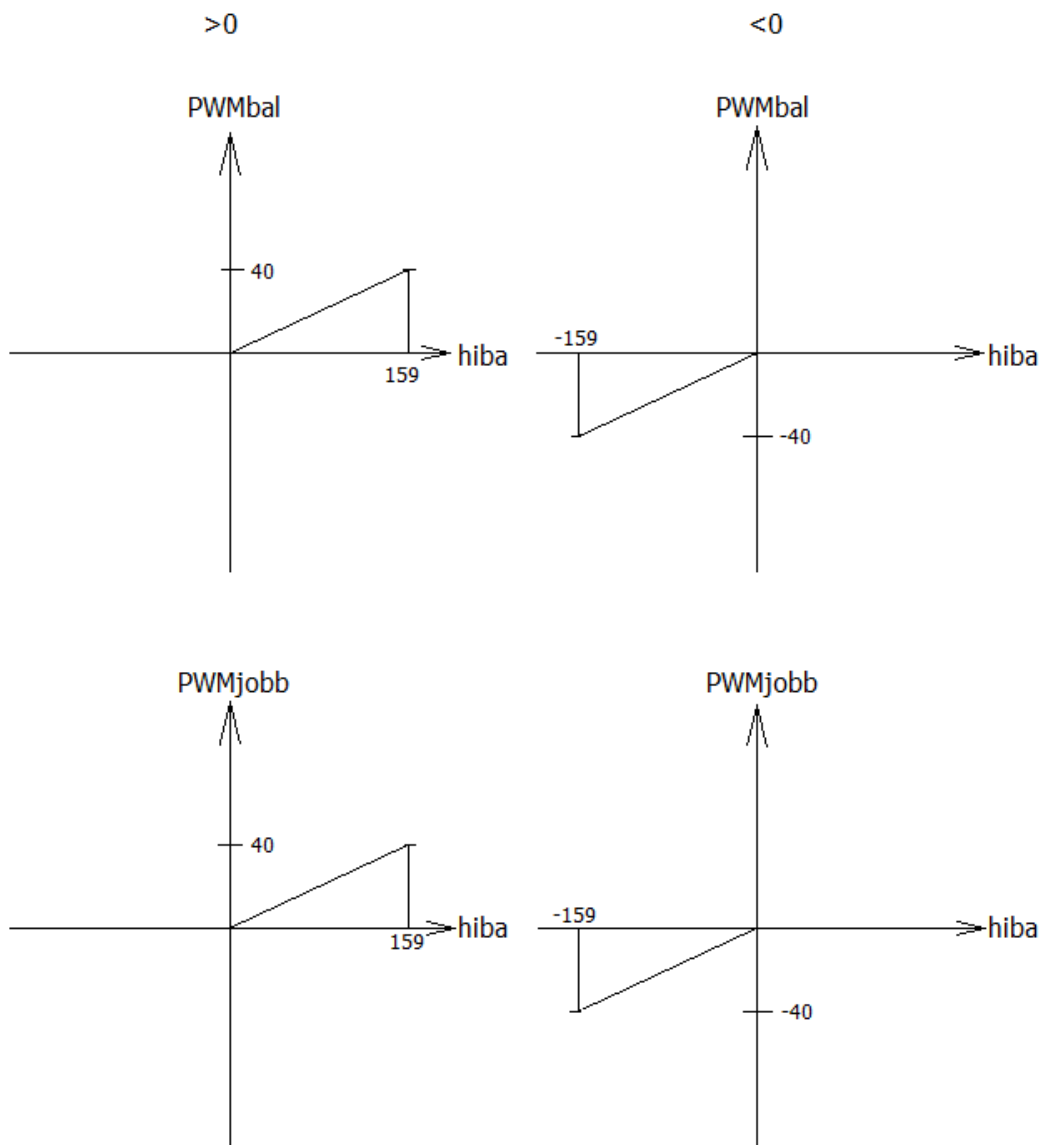


50. ábra Beparkolás ötödik lépése – a parkolóhely közepén történő megállás

A maximális hosszt, ahová még a robot beáll 450mm-re korlátoztam. Ebben az esetben a szenzorok által mérhető legnagyobb távolság 185 mm. Mivel az ultrahangos szenzorok mérési tartománya 20-400mm ezért a robot nem állhat a falhoz ennél közelebb, mert ekkor a kapott szenzorértékek teljesen hibásak lesznek. Ezért már a harmadik módszerben korlátoztam a robot mozgását. Így a 185 milliméterből kivonva a két szenzor méréstartományának alsó határát, 40-et, megkapjuk, hogy milyen értékek között tud pontosan középre beállítani a mobil robot. 145mm. Az ehhez az értékhez tartozó inkrementális osztásszám a következő:

$$\frac{145}{0.9139} \approx 159$$

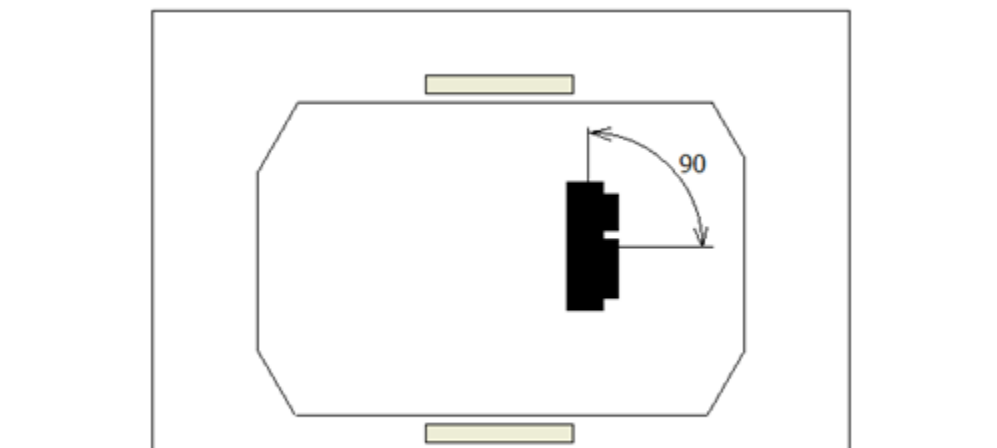
A robot maximálisan 159 osztásnyit mozoghat egyik irányban a 400mm hosszú parkolóhelyen. Ezen számolások alapján kapott két függvény:



51. ábra A parkolóhely pontosan közepén történő megálláshoz szükséges szabályzás függvényei

A függvények ismét szemléltetik a motorok irányait. A szabályzás a nulla ponthoz irányítják a rendszert.

Hatodik lépésként az infravörös szenzort visszafordítjuk a Home Pozícióba. Ehhez 90 fokkal kell elforgatni jobbra. Ez 50 lépés a léptető motornak.

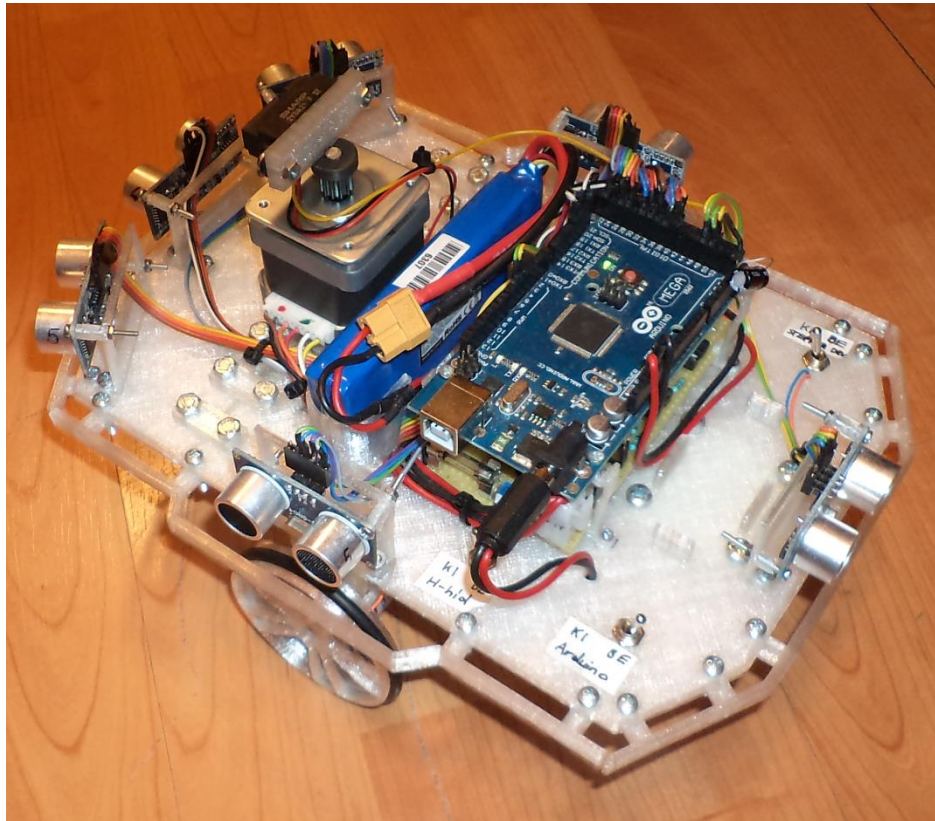


52. ábra Beparkolás hatodik lépése – ultrahangos szenzor elforgatása 90 fokkal

Ezzel a parkolás teljesen kész.

5. Eredmények kiértékelése

A mobilrobot tervezése és kivitelezése jól sikerült. A 3D-s nyomtatásnak hála olyan robotot sikerült építeni, amelyet elképzeltem. A szerkezet kellően stabil ahhoz, hogy további érzékelőkkel is el lehessen látni a robotot. Végző esetben akár hozzá is lehet toldani a vázszerkezethez, hogy még nagyobb területem legyen a különböző feladatok ellátására.



43. ábra A kész mobilrobot

Eddigi parkolási tesztjeim során a robot jól működött, de még sokat kell fejleszteni a működtető algoritmust, hogy pontosan be tudjon parkolni. Az infravörös szenzor meglehetősen érzékeny a különböző fényhatásokra, ezért sokszor előfordult, hogy rossz irányba kanyarodott már a parkolás első lépésekor.

Végeredményben kijelenthető, hogy a robotkonstrukció megfelel az elvárásoknak, de a programozás terén még sok fejlesztési lehetőség akad.

6. Összegzés és fejlesztési lehetőségek

A TDK dolgozatomban olyan mobilrobotot terveztem és kiviteleztem, amely bizonyos mértékig feltérképezi a környezetét, megtalálja a parkolóhelyet és beparkol. Az Inventorban tervezett és 3D-s nyomtatóval létrehozott váz, teherbírása és egyéb mechanikai tulajdonságai kiemelkedően jók. A parkolási algoritmus, ami jelenleg is fejlesztés alatt áll, kifejezetten differenciálhajtású robotok számára készült. A fejlesztés Arduino programban végeztem. A beparkoláshoz szükséges számításokat ezen dolgozat alapján bárki ki tudja számolni. A korlátok kibővíthetők, esetenként meg is szüntethetők. Tesztjeim alapján a konstrukció megállja a helyét a többi parkolási rendszerrel szemben.

Továbbra is használható négykerekűként, de akár továbbfejleszthető háromkerekűként is. Fejlesztési lehetőségek a feltérképezés során: komplett környezet feltérképezés, jobb szenzorok. A parkolásnál ne csak hosszában, hanem keresztben is be tudjon állni, emellett a valós életben előforduló szituációkat tudja kezelni.

Ezekén felül multi-ágensű környezetben a kommunikáció megvalósítása. Egy komplett parkolóház, vagy bevásárlóközpont parkolójának teljes körű automatizálása. Ezt több robot segítségével bemutatva.

7. Köszönetnyilvánítás

Külön szeretném megköszönni Dr. Nagy Istvánnak és Liszkai Bencének a sok segítségét, akik nélkül nem készülhetett volna el ez a dolgozat. Dr. Nagy István a dolgozat megírásában, a pályatervezésben segített. Liszkai Bence a 3D-s nyomtatásban, az Arduino mikrokontroller és programozási felület megismerésében, emellett a különböző szabályzások megírásában nyújtott segítséget.

8. Irodalomjegyzék:

- [1] Arduino Mega 2560: <http://arduino.cc/en/Main/arduinoBoardMega2560>
- [2] Arduino Mega 2560: <http://arduino.cc/en/Main/arduinoBoardMega2560>
- [3] ATmel Corporation, ATmega 2560 Datasheet: http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf, utolsó letöltés időpontja: 2014.04.15.
- [4] Tudomány és technika, 16.4 PWM üzemmód: <http://www.t-es-t.hu/elokep/pic/felhkk/kk/sz1604.htm>
- [5] Tudomány és technika, 16.4 PWM üzemmód: <http://www.t-es-t.hu/elokep/pic/felhkk/kk/sz1604.htm>
- [6] ZHENGKE, ZGA17RU DC Gear Motor: <http://wzh001.gotoip55.com/upload/file/ZGA12FT%20%26%20ZGA17RU.pdf>, utolsó letöltés időpontja: 2014.04.15.
- [7] ZHENGKE, ZGA17RU DC Gear Motor: <http://wzh001.gotoip55.com/upload/file/ZGA12FT%20%26%20ZGA17RU.pdf>, utolsó letöltés időpontja: 2014.04.15.
- [8] Allegro A4988 stepper motor driver datasheet: http://www.pololu.com/file/0J450/a4988_DMOS_microstepping_driver_with_translator.pdf, utolsó letöltés időpontja 2014.04.15.
- [9] Pololu A4988 stepper motor driver carrier with included header pins soldered: <http://www.pololu.com/picture/view/0J4578>
- [10] JAPAN SERVO CO. KH42JM2B140E stepper motor: <http://pdf1.alldatasheet.com/datasheet-pdf/view/116933/ETC1/KH42JM2-903/+0159W5VwSwJbLKYN0Tzppz.+/datasheet.pdf>, utolsó letöltés időpontja: 2014.04.15.

- [11] JAPAN SERVO CO. KH42JM2B140E stepper motor
<http://pdf1.alldatasheet.com/datasheet-pdf/view/116933/ETC1/KH42JM2-903/+0159W5VwSwJbLKYN0Tzppz.+/datasheet.pdf>, utolsó letöltés időpontja: 2014.04.15.
- [12] Sharp 2Y0A21YK infravörös szenzor:
<http://pdf.datasheetarchive.com/indexerfiles/Datasheet-075/DSAE003120.pdf>,
utolsó letöltés 2014.04.15.
- [13] Society of robots: http://www.societyofrobots.com/sensors_sharpirrange.shtml
- [14] Sharp 2Y0A21YK infravörös szenzor:
<http://pdf.datasheetarchive.com/indexerfiles/Datasheet-075/DSAE003120.pdf>,
utolsó letöltés időpontja: 2014.04.15.
- [15] Fizikus, 201.02.05. WILL-I V2.0 (Robotika és AVR kezdőknek), Sharp infravörös távolságérzékelő szenzor:
http://www.hobbielektronika.hu/cikkek/will-i_v20_robotika_es_avr_kezdoknek.html?pg=6
- [16] Fizikus, 201.02.05. WILL-I V2.0 (Robotika és AVR kezdőknek), Sharp infravörös távolságérzékelő szenzor:
http://www.hobbielektronika.hu/cikkek/will-i_v20_robotika_es_avr_kezdoknek.html?pg=6
- [17] Fizikus, 201.02.05. WILL-I V2.0 (Robotika és AVR kezdőknek), Sharp infravörös távolságérzékelő szenzor:
http://www.hobbielektronika.hu/cikkek/will-i_v20_robotika_es_avr_kezdoknek.html?pg=6
- [18] Fizikus, 2011.12.07. SRF-04 Ultrahangos távolságmérő szenzor használata Arduino-val és AVR-rel, SRF-04 Ultrahangos távolságmérő szenzor:
http://www.hobbielektronika.hu/cikkek/srf-04_ultrahangos_tavolsagmero_szenzor_hasznalata_arduino-val_es_avr-rel.html?pg=2

- [19] Fizikus, 2011.12.07 SRF-04 Ultrahangos távolságmérő szenzor használata Arduino-val és AVR-rel, SRF-04 Ultrahangos távolságmérő szenzor:
http://www.hobbielektronika.hu/cikkek/srf-04_ultrahangos_tavolsagmero_szenzor_hasznalata_arduino-val_es_avr-rel.html?pg=2
- [20] Fizikus, 2011.12.07, SRF-04 Ultrahangos távolságmérő szenzor használata Arduino-val és AVR-rel, SRF-04 Ultrahangos távolságmérő szenzor,
http://www.hobbielektronika.hu/cikkek/srf-04_ultrahangos_tavolsagmero_szenzor_hasznalata_arduino-val_es_avr-rel.html?pg=2
- [21] HCSR-04ultrahangos szenzor: www.micropik.com/PDF/HCSR04.pdf, utolsó letöltés időpontja 2014.04.15.
- [22] Fizikus, 2011.12.07, SRF-04 Ultrahangos távolságmérő szenzor használata Arduino-val és AVR-rel, SRF-04 Ultrahangos távolságmérő szenzor,
http://www.hobbielektronika.hu/cikkek/srf-04_ultrahangos_tavolsagmero_szenzor_hasznalata_arduino-val_es_avr-rel.html?pg=2
- [23] Fizikus, 2011.12.07, SRF-04 Ultrahangos távolságmérő szenzor használata Arduino-val és AVR-rel, SRF-04 Ultrahangos távolságmérő szenzor,
http://www.hobbielektronika.hu/cikkek/srf-04_ultrahangos_tavolsagmero_szenzor_hasznalata_arduino-val_es_avr-rel.html?pg=2
- [24] beczl, 2010 02. 11. Litium-Polimer LiPo akkuk használata az airsoftba:
<http://6mm.hu/posts/lithium-polimer-lipo-akkuk-hasznalata-az-airsoftban>
- [25] ZIPPY Flightmax 1500mAh 3S1P20C akkumulátor:
https://www.hobbyking.com/hobbyking/store/_6307_ZIPPY_Flightmax_1500mAh_3S1P_20C.html
- [26] ATtiny, 2010.02.01, Inkrementális szögadó illesztése mikrokontrollerhez:
http://www.hobbielektronika.hu/cikkek/inkrementalis_szogado_illesztese_mikrokontrollehez.html?pg=2
- [27] ATtiny, 2010.02.01, Inkrementális szögadó illesztése mikrokontrollerhez:
http://www.hobbielektronika.hu/cikkek/inkrementalis_szogado_illesztese_mikrokontrollehez.html?pg=2

- [28] ATtiny, 2010.02.01, Inkrementális szögadó illesztése mikrokontrollerhez:
[http://www.hobbielektronika.hu/cikkek/inkrementalis_szogado_illesztese_mikrokontrolle
rhez.html?pg=3](http://www.hobbielektronika.hu/cikkek/inkrementalis_szogado_illesztese_mikrokontrolle
rhez.html?pg=3)
- [29] ATtiny, 2010.02.01, Inkrementális szögadó illesztése mikrokontrollerhez:
[http://www.hobbielektronika.hu/cikkek/inkrementalis_szogado_illesztese_mikrokontrolle
rhez.html?pg=3](http://www.hobbielektronika.hu/cikkek/inkrementalis_szogado_illesztese_mikrokontrolle
rhez.html?pg=3)
- [30] ATtiny, 2010.02.01, Inkrementális szögadó illesztése mikrokontrollerhez:
[http://www.hobbielektronika.hu/cikkek/inkrementalis_szogado_illesztese_mikrokontrolle
rhez.html?pg=3](http://www.hobbielektronika.hu/cikkek/inkrementalis_szogado_illesztese_mikrokontrolle
rhez.html?pg=3)
- [31] ATtiny, 2010.02.01, Inkrementális szögadó illesztése mikrokontrollerhez:
[http://www.hobbielektronika.hu/cikkek/inkrementalis_szogado_illesztese_mikrokontrolle
rhez.html?pg=3](http://www.hobbielektronika.hu/cikkek/inkrementalis_szogado_illesztese_mikrokontrolle
rhez.html?pg=3)
- [32] ATtiny, 2010.02.01, Inkrementális szögadó illesztése mikrokontrollerhez:
[http://www.hobbielektronika.hu/cikkek/inkrementalis_szogado_illesztese_mikrokontrolle
rhez.html?pg=3](http://www.hobbielektronika.hu/cikkek/inkrementalis_szogado_illesztese_mikrokontrolle
rhez.html?pg=3)
- [33] HLC2701: www.bitsbox.co.uk/data/HLC2701.pdf, utolsó letöltés időpontja: 2014.04.15.
- [34] SEP8506-001: <http://www.datasheets360.com/pdf/-1134370170368341890>,
 utolsó letöltés: 2014.04.15
- [35] HLC2701: www.bitsbox.co.uk/data/HLC2701.pdf, utolsó letöltés időpontja 2014.04.15.
- [36] SEP8506-001: <http://www.datasheets360.com/pdf/-1134370170368341890>,
 utolsó letöltés: 2014.04.15.
- [37] J. Borenstein, H.R. Everett, L. Feng: Where am I? Sensors and Methods for Mobile Robot
 Localization, MIT, USA, 1996.?
- [38] M.D. Adams: Sensor modelling, Design and Data Processing for autonomous Navigation
 World Scientific Series, Vol. 13, ISBN 9810234961, 1999.
- [39] I. Nagy: Path Planning Algorithm, Based on User Defined Maximal Localization
 Periodica Polytechnica
 ISSN: 0324-6000 (paper),
 Vol.: 2005 49/1-2, pp.:43-57, Hungary 2005