

Óbudai Egyetem

Bánki Donát Gépész és Biztonságtechnikai Mérnöki Kar

Gépészeti Intézet



TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

TELEMANIPULÁCIÓS ROBOTKAR MEGTERVEZÉSE ÉS KIVITELEZÉSE

Szerző:

Herczeg Zoltán

mechatronika BSc. szak, III. évf.

Konzulens:

Dr. Nagy István

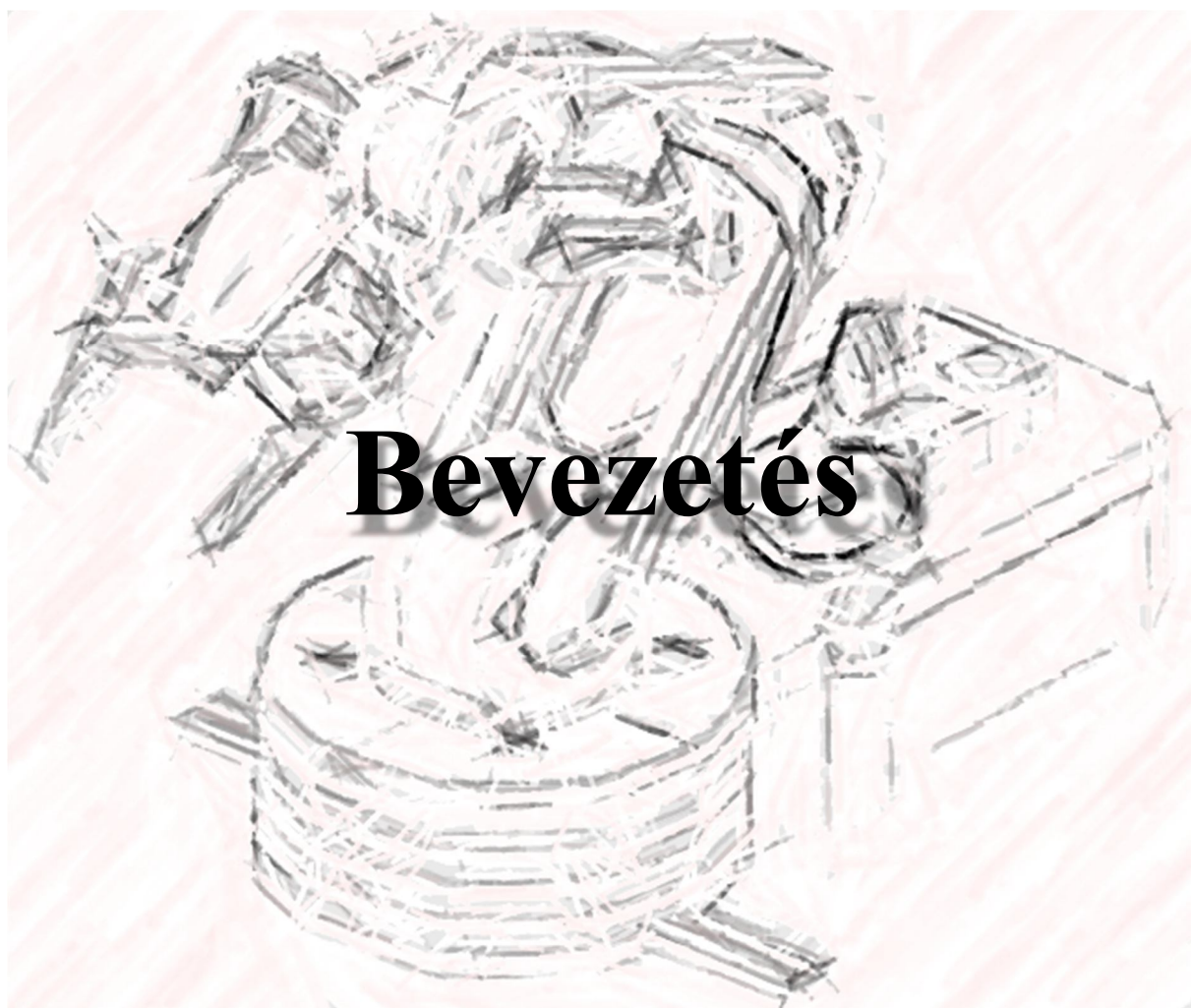
egyetemi docens

Kelt.: Budapest, 2014 November

Tartalomjegyzék

Bevezetés	4
Előszó	5
Robotokról röviden	6
Robotika alaptörvényei	6
A robotok csoportosítása	7
Ipari alkalmazás	8
Robotok szerepe a kutatásban	8
Robotok az egészségügyben	8
A telemanipulációs rendszerek kialakulásának története	10
Mit is takar a telemanipuláció	11
Tervezés	13
Célkitűzések és eszközválasztás	14
A készülék konstrukciója	15
OWI 535 Robotkar	17
StellarPad Tiva C Series TM4C123GH6PMI mikrokontroller	18
A fejlesztőkörnyezetről	19
ADXL335 gyorsulásérzékelő szenzor	20
A gyorsulásmérő jelének feldolgozása	21
Kivitelezés	25
A program folyamatábrája	26
Programom felépítése	27
Robottengelyeinek mozgatása, hozzá tartozó gesztusokkal	29
Saját tervezésű motorvezérlő elektronika	30
Célkitűzés összegzése	32
A rendszerem továbbfejlesztésének egy lehetősége	33
Mellékletek	34
A robotvezérlő elektronikai kapcsolása	35
Részlet a gyorsulásmérő szenzor adatlapjából	36
Teljes forráskód	37

Forrásmunkák.....	40
Felhasznált irodalom.....	41
Idézett forrásmunkák	32
Ábrajegyzék.....	43



Bevezetés

Kelt.: Budapest, 2014 November

Előszó

Az elmúlt 40 évben a rohamosan fejlődő technika hatása érzékelhető az ipar minden egyes szegletén. Ennek köszönhetően a modern technikai eszközök mára már annyira alapvető használati tárggyá váltak, hogy nélkülük el sem tudnánk képzelni a mindennapjainkat. Ilyeneknek tekinthető például a személyi számítógépek hálózata vagy a mobiltelefonok.

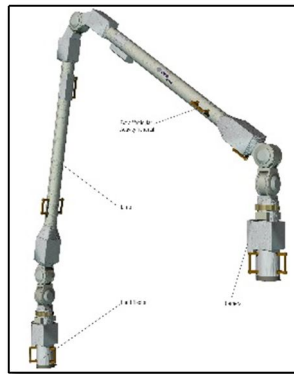
Az emberben mindig is ott motoszkált az a gondolat, vágy, hogy teremtsen, mint Isten, valami újat, amely a saját képmására hasonlít, ennek a folyamatnak köszönhetően jelentek meg a robotok. Minden olyan mechanikus rendszer, amely képes egy kitűzött feladat végrehajtására, tudományos szempontból robotnak tekinthető. A mai modern világban a robotok olyan fémkaroknak tekinthetők, amelyek képesek nagy munkapontossággal végrehajtani az adott munkát akár embertelen körülmények között is.

A TDK dolgozatom röviden bemutatja a robotok csoportosítását, majd a telemanipulációs rendszereket, annak felhasználási lehetőségeit, továbbá ennek kommunikációs csatornáit. A telemanipulációs rendszerem tervezését illetve kivitelezését a project megvalósításához alkalmazott eszközökön keresztül mutatom be.

A témaválasztás komoly érdeklődés miatt esett a robotokra, mivel már gyerekként is érdekelt voltam a technika ezen irányú fejlődése felé. A projektem olyan telemanipulációs rendszer kialakítása, amely képes az embertől távolabb eső környezetben beavatkozást végezni, mindezt az emberi kézfej mozgatásának függvényében. A project sikeres megvalósításához főképpen az Ipari robot rendszerek és az Elektronika tantárgyakon tanultakat használtam fel, azonban az ismereteket a mikrovezérlőről és annak programozásához szükséges tudást önfejlesztés során sajátítottam el.

Robotokról röviden

A robot egy olyan szerkezet, amely programozható és képes feladatok elvégzésére, lehet közvetlen emberi irányítás által (mint a Space Shuttle robotkarja), de számítógépre bízva önállóan is végezheti a feladatát. A robotok olyan munkát végeznek amely, az ember számára túl veszélyes, esetleg kivitelezhetetlen (például nukleáris hulladék megsemmisítése), túlságosan monoton vagy a pontosság nagymértékű prioritást élvez. Ez az oka a járműgyártás területén való alkalmazásnak. Robotokat harcászati célra is felhasználnak, bomba hatástalanítására és felderítésre.



1. ábra Space Shuttle robotkar; forrás: [1]

Robotika alaptörvényei

A XX. században Isaac Asimov megalkotta a robotika három törvényét:

1. „A robotnak nem szabad kárt okoznia emberi lényben vagy tétlenül tűrnie, hogy emberi lény bármilyen kárt szenvedjen.”¹
2. „A robot engedelmeskedni tartozik az emberi lények utasításainak, kivéve, ha ezek az utasítások az első törvény előírásaiba ütköznének.”¹
3. „A robot tartozik saját védelméről gondoskodni, amennyiben ez nem ütközik az első és második törvény előírásaiba.”¹

Robotika területén dolgozó tudósok és mérnökök, alkotják meg a mai modern robotokat, melyek követik eme három alaptörvényt.

¹ Isaac Asimov, Robottörténetek, Doubleday, Amerikai Egyesült Államok, 1941 október

A robotok csoportosítása

A robotok csoportosítása számos szempont alapján történhet: intelligencia szint, megjelenés vagy alkalmazási hely szerint. Akárhogyan is csoportosítjuk őket mindegyik szempont nagy változatosságot képes produkálni. Intelligencia szint szerint három generációt tudunk elhatárolni egymástól.

Az első generációs robotoknak a mozgáspályájukat egyértelműen meghatározza egy előre megírt program, amely többszöri ismétlésre képes. Munkadarabok mozgatására használja az ipar, ilyen volt az első ipari felhasználású robot is még a '60-as években.

A második generáció robotjai már különböző környezetet felmérő érzékelő rendszerekkel voltak felszerelve, amelyek segítségével már némi önállóságra is szert tettek, ami alatt azt értjük, hogy az előre megírt programban feladatelvégzés szempontjából némi rugalmasság is elérhető. A '70-es években jelent meg.

A harmadik generáció robotjai közé tartoznak azok, amelyek már kapcsolódnak a mesterséges intelligenciához. Ennek kivitelezése a környezetet vizsgáló szenzorok nélkül elképzelhetetlen. Az érzékelők információval látják el a robotvezérlőt, mely az érkező információ alapján és a megírt program függvényében a lehető legoptimálisabb önálló döntést hozza.

Ipari alkalmazás

J. F. Engelberg és G. Devol fejlesztették ki a Unimate-t, az első ipari felhasználású robotot. A General Motors alkalmazta a termelési részlegen, gépjárműgyártás automatizálásához. A robotkar olyan feladatot látott el, amely az ott dolgozó operátorok számára veszélyes feladatnak minősült. A Unimate sikerének köszönhetően a robotok fejlesztésének sebessége rögvest megnövekedett, ez vezetett ahhoz, hogy napjainkban megtalálható robotok 96%-át az iparban használják fel.



2. ábra Unimate az első ipari felhasználású robot; forrás: [2]

Ipari felhasználásban levő robotok közös tulajdonsága, hogy helyhez kötöttek, azaz nem képesek helyváltoztatásra, mivel többnyire egyetlen feladatot kell ellátniuk, a lehető legnagyobb precizitással és gyorsasággal. Ebből adódóan alkalmazhatóak az adagolástól egészen a gyártásig. A legfontosabb szerepet az autógyártásban töltik be (pl.: hegesztő-, festő-, szerelő-, anyagszállító robotok), azonban fontos szerepet töltenek be a mikroelektronikában is.

Robotok szerepe a kutatásban

Robotok ezen területen való felhasználása alapvetően olyan berendezéseket jelent, melyek adott környezetben képesek mozogni, ilyenek lehetnek a mobilrobotok. Megtalálhatóak levegőben, szárazföldön és vízben is. A legjelentősebb felhasználása katonai és űrkutatásban történik. Ilyenek például a pilóta nélküli repülőgépek, víz alatti robotok és a Mars járművek.

Robotok az egészségügyben

Rendkívüli módon fejlődik a robotika ezen ágazata. Segítségükkel rengeteg műtét zajlott le sikeresen az Egyesült Államokban. 2003 óta da Vinci névre keresztelt robotot alkalmaznak sebészeti beavatkozásokra.



3. ábra *da Vinci sebészrobot; forrás: [3]*

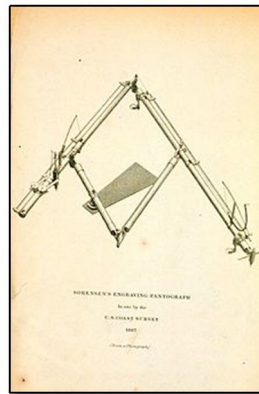
Ahogy korábban említettem, fontos a robotok magas fokú precizitása, ebből adódóan képesek a lehető legminimálisabb vágáson keresztül elvégezni a műtétet, ennek nagy előnye, hogy kisebb heg miatt a páciens hamarabb meggyógyulhat. Telemánipulációs kommunikációt alkalmazva lehetőség nyílik arra, hogy az orvos olyan páciensen hajtson végre beavatkozást, aki akár más kontinensen tartózkodik, tehát nem szükséges az orvosnak és betegének egy légtérben tartózkodnia. A jövő gyógyászatában felhasználhatunk nanorobotokat, amelyek képesek lesznek olyan betegségek legyőzésére, amiket hagyományos kezelésekkkel jelenleg elérhetetlenek. Ilyen kutatások zajlanak rákos sejtek legyőzése céljából. Orvosaink egyszer képesek lesznek, egy fecskendőnyi nanorobotot bejuttatni a szervezetünkbe, amelyek felkutatják és elpusztítják a rákos sejteket.



4. ábra *Nanorobot; forrás: [4]*

A telemanipulációs rendszerek kialakulásának története

Az első telemanipulációs rendszert 1945-ben Goertz fejlesztette ki, az emberi karok egyszerű meghosszabbításainak révén. Ez egy mechanikai pantográf (olyan mechanizmus melynek négy karja van, karjai egy paralelogrammát képeznek, melyeknek végeit csuklók alkotják) rendszer volt, az ember radioaktív anyagok mozgatására használta miközben biztonságos távolságra helyezkedett el a munkatértől és a veszélyes anyagtól. Az operátor megfogta és mozgatta a fogantyút, mint master eszközt, és a cellában lévő slave eszközt, mely megfogóként működött és tetszőlegesen tudta irányítani.



5. ábra Mechanikai pantográf; forrás: [5]

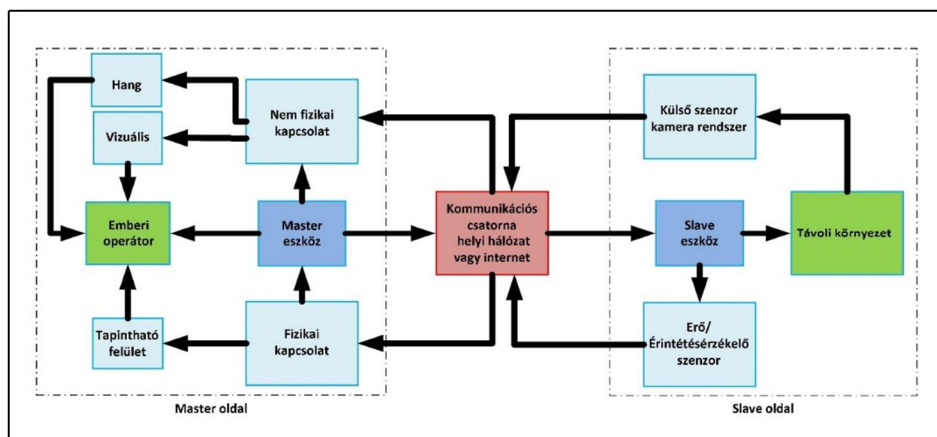
1945-ben megjelentek az elektronikus szervomechanizmusok, melyek helyettesítették a közvetlenül kapcsolódó mechanikus master-slave rendszerek kapcsolatait. Lényegében a slave és a manipulálandó környezet között kialakult erőhatásokról a master eszköz informálódott.

1964-ben Mosher kutatásai során kifejlesztett egy elektrohidraulikus kezekkel rendelkező modellt, ami kezenként 10 szabadságfokkal rendelkezett.

A távirányítású robotok iránti igények megnövekedése során a telemanipuláció egyre szélesebb körű felhasználása mutatkozik gyártásban, víz alatti kutatásban, veszélyes területeken való műveletvégzésnél például atomerőművekben, űrkutatásban.

Mit is takar a telemanipuláció

Amikor az operátornak el kell végeznie egy feladatot, amely veszélyes, esetleg ember számára megközelíthetetlen környezetben van, ahol az ember nem lehet jelen. A telemanipulátor használatával az operátor képes távoli munkavégzésre, ezáltal a hatásköre nagy mértékben bővül, ezt a bővítést master-slave eszközök kialakításával oldották meg. A rendszerünk két szoros kötelékű műveletvégzésre, folyamatra bontható. Az egyik a master és az operátor együttműködése, a másik slave azaz a robot manipulátora és az ott jelenlévő környezetnek a viszonya. A rendszer használata során, amikor az operátor oldalán kifejtett erőnek nincs kihatása a teljesítményre, olyankor az operátor helyzetének mérése és annak visszajelzése elegendő vizuális megfigyelővel, viszont annál az elvégzendő feladatnál, amikor a slave eszköz károsíthatja a manipulálandó környezetet, például alkatrész összeszerelése, fűrés, csavarás, az aktuális erő visszajelzése elengedhetetlen a hatékony munkavégzés miatt. Ezáltal az operátor teljes mértékben tisztában van a környezetben zajló folyamattal, mintha jelen lenne ott. A következő 6. ábra szemlélteti a telemanipulációs rendszerekben az információ áramlását, az operátor és az irányított távoli környezet között. Az ábra jól szemlélteti azon visszacsatolásokat, amelyek jelen vannak a rendszerben (hang, vizuális és tapinthatóság). Igaz az ember képes akár öt féle visszajelzést érzékelni a környezet felől, ezen telemanipulációs rendszerek csupán hárommal foglalkoznak.



6. ábra Master-slave kapcsolat blokkvázlata; forrás: szerző által készített ábra

Elsősorban az egészségügyben, mégpedig ahogy fentebb említettem, műtéteket végeznek telemanipulációval, ezek száma folyamatosan nő. Eme robotok felhasználása főképpen az emberi hiányosságokat igyekszenek kiküszöbölni, a kéz remegését esetleg pontatlanságát. Továbbá ha a sebész nem tud jelen lenni az operáció helyszínén, távvezérlést tudnak alkalmazni.

Maga a kommunikáció problémája igen nagy fontossággal bír, legfőképpen a kommunikációs csatorna kialakítása, valamint válaszidejének gyakorlatilag konvergálnia kell a 0 ms-hoz, ehhez viszont még zajlanak a megfelelő kommunikációs protokoll kutatások. Egy jelentős probléma lehet még a haptikus észlelés megoldása, azaz ha valamit megérintünk pusztán kézzel, akkor a tapintási receptoraink visszajelzést küldenek az agyunknak, mely szabályozza az ujjunk által kifejtett nyomást. Ez egy igen jelentős probléma, hiszen ha nem kapunk megfelelő visszajelzést a környezetben folytatott manipulációról, akkor könnyen kárt okozhatunk.



Tervezés

Kelt.: Budapest, 2014 November

Célkitűzések és eszközválasztás

Amikor a hírekben megláttam, hogy az Óbudai Egyetem birtokolja hazánk egyetlen sebész robotját, a fenn említett da Vinci-t, akkor utána olvastam, mivel vezérlik, ekkor bukkantam rá a telemanipulációra, mint az irányítás egy módjára. Minél többet olvastam a témával kapcsolatban, már tudatosult bennem, hogy ez tényleg egy olyan találmány, amely az emberiség segítségére van, és nagyszerű kutatások alappillére lehet.

A témával kapcsolatban kerestem különféle megoldásokat, amik megfelelően szemléltetik a telemanipulációt. Igen érdekes rendszerekre bukkantam, vannak remekül működő megoldások, például joysticktól, gombokon át, egészen a karra szerelt nyomásérzékelőig bezárólag.

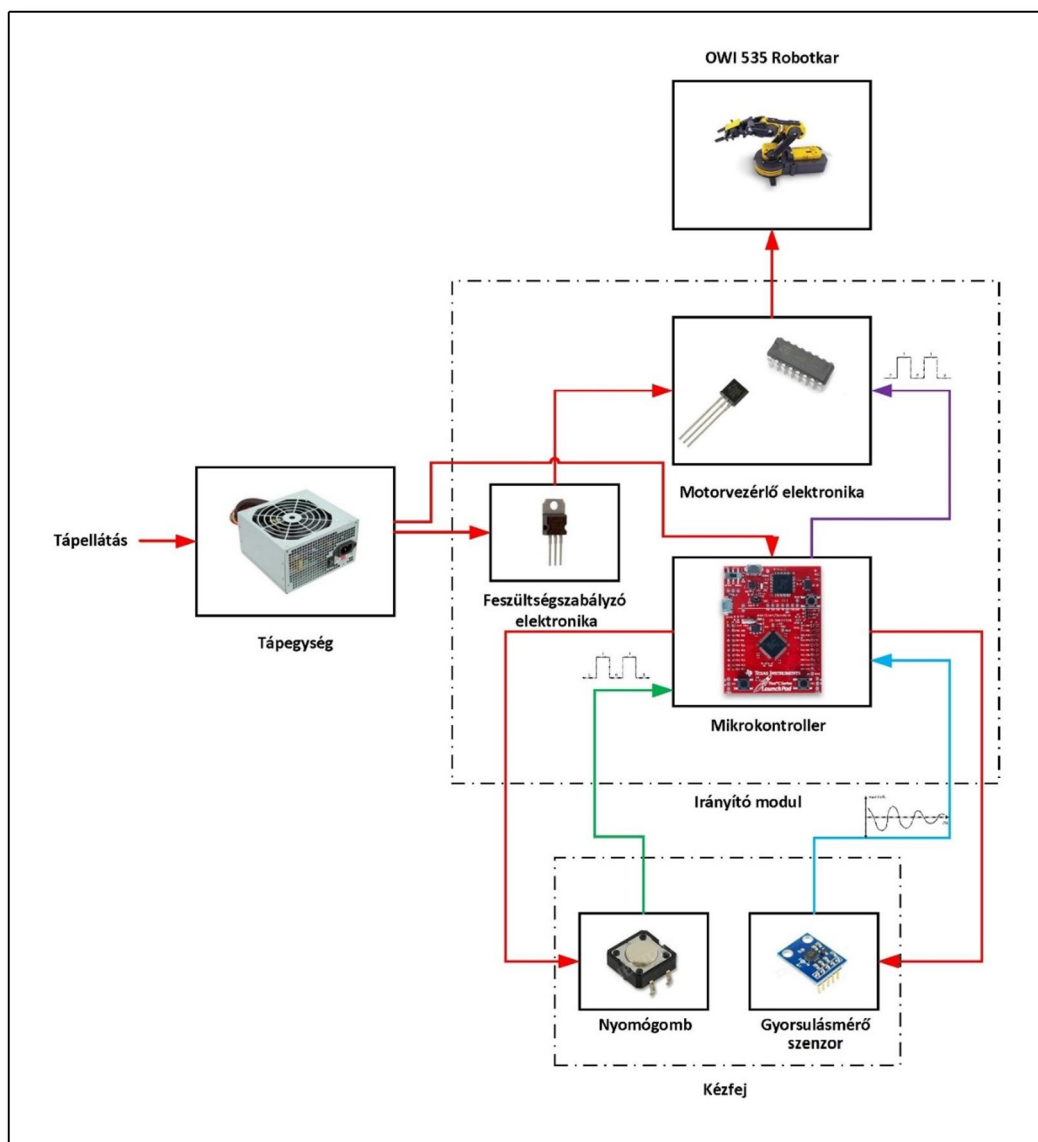
Általam elkészített rendszer megvalósítás is beléphet a telemanipulációs vezérlés lehetséges megoldásai közé.

A telemanipuláció egy lehetséges megvalósítását mutatom be, öttengelyes OWI 535 robotkar működtetése közben. Két gyorsulásmérővel érzékeli a master oldalt, mely a jobb és a bal kézfejemben lévő pántra van rögzítve. Ezzel tudok négy tengelyt mozgatni, továbbá a tenyereimben nyomógombok vannak, amely a robot manipulátorát képesek nyitni illetve zárni.

Az analóg kimenetű szenzorok jelei a kézfej döntésétől változnak, amelyek az irányító modulon keresztül kerülnek a mikrokontrollerbe. Az analóg jel ottani feldolgozása következik, amely a jel függvényében megfelelő digitális kimenetet hoz létre, és továbbítja a motorvezérlő kapcsoló pinjeire. Ennek hatására a kiépített motorvezérlő elektronika megfelelő feszültséget kapcsol a robot motorjaihoz, így megkezdődik a robot vezérlése, tehát létrejön a master-slave kapcsolat. Master oldalon ember által irányított gyorsulásmérő, slave oldalon a távoli környezetet manipuláló megfogó helyezkedik el.

A készülék konstrukciója

- OWI 535 Robotkar
- StellarPad Tiva C Series TM4C123GH6PMI mikrokontroller
- ADXL335 gyorsulásérzékelő szenzor
- Nyomógomb
- Motorvezérlő elektronika (L293D, tranzisztoros H-híd)
- Feszültség szabályzó elektronika (potenciométerrel szabályozható a robot tengelyeinek a sebessége, feszültségnöveléssel illetve csökkentéssel)
- Külső tápegység



7. ábra A project blokkvázlata; forrás: szerző által készített ábra

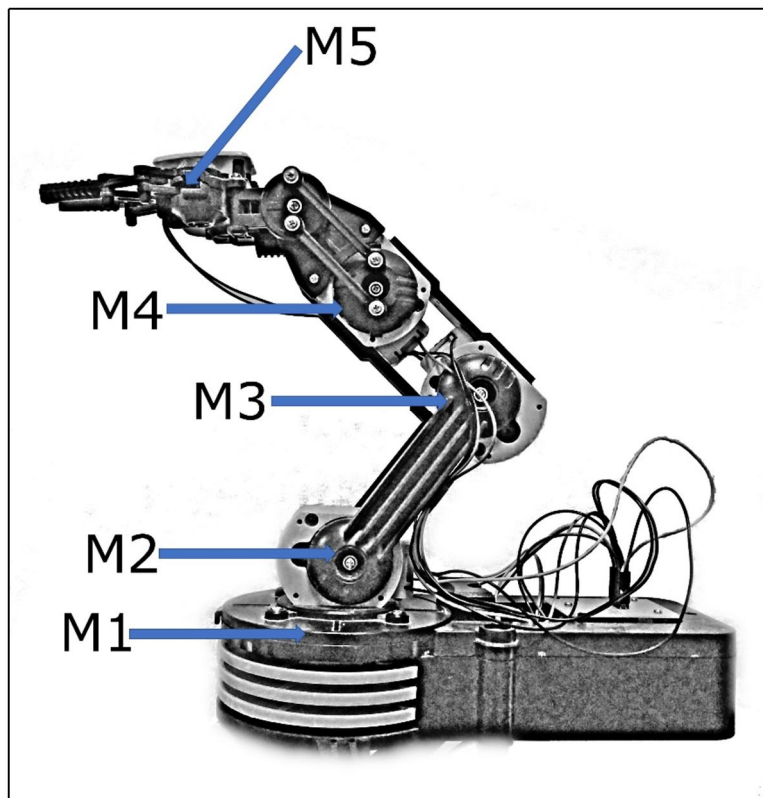
A blokkvázlatban a tápellátás piros, a StellarPad mikrokontroller által kiadott digitális jel lila, a bemenő jelek közül a gomb által generált HIGH/LOW feszültségszintből adódó digitális jel zöld, a gyorsulásérzékelő kimeneti analóg jele pedig kék színnel van jelölve.

A tápegységen, robotkaron és a kézfejen lévő gombokon, valamint szenzorokon kívül valamennyi alkotóelem egy fémdobozban helyezkedik el, az ábrán irányító modulként van megnevezve.

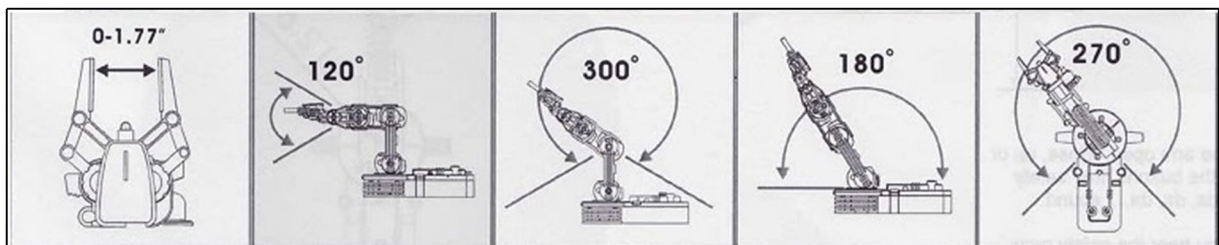
Az irányítómodul az egésznek a központja, ki és bemenetek ezen belül kerülnek feldolgozásra, kiértékelésre, valamint a vezérlő jel generálásáért felelős. Ennek elektronikája teljes egészében saját tervezés és kivitelezés.

A mikrokontroller fő feladata, hogy a beérkező analóg jelek függvényében valamint a digitális jelből megfelelő digitális kimenetet hozzon létre. Robotkar adott volt, amely tulajdonképpen egy gyermekeknek való játék, távirányítóval ellátva. Eltávolítottam róla mindent, ami számomra felesleges a projecthez.

A csatlakozást tekintve soros portos kommunikáció van a kézfej és az irányító modul között. A robot és a modul között viszont VGA kábel összeköttetést használtam fel.

OWI 535 Robotkar

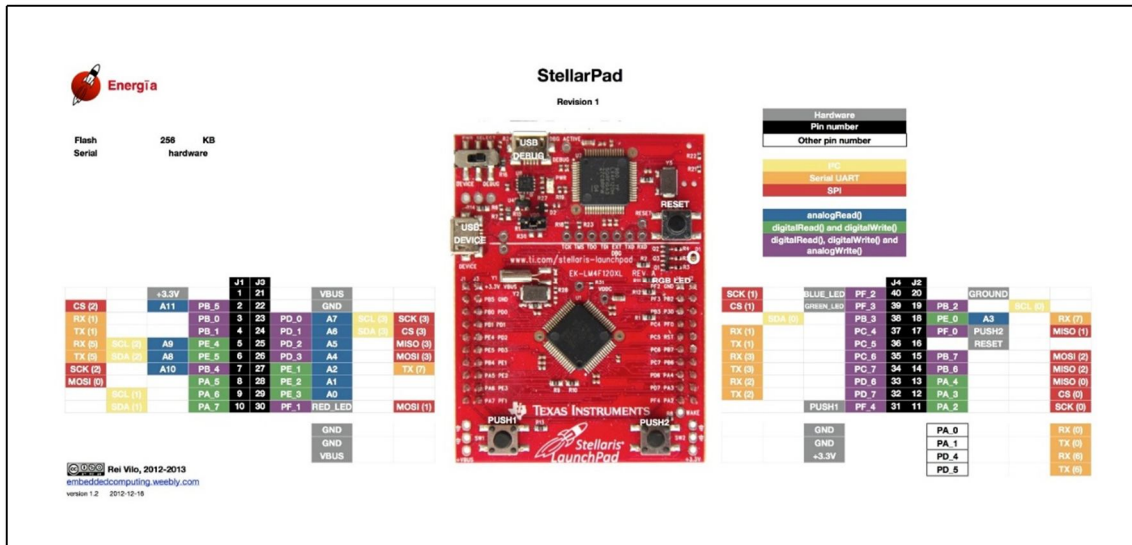
8. ábra OWI 535 robotkar; forrás: szerző által készített ábra



9. ábra A robotkar megengedhető mozgástere; forrás: [6]

Öt motorral és öt ízülettel rendelkező robotkar, a tengelyek hajtóműve összetett fogaskerekekkel van kialakítva, amely biztosítja a hosszú élettartamot, valamint szerepet játszik abban, hogy ne sérüljön meg a robotkar, ha a robot megközelíti a maximálisan megengedhető szélsőértéket. Azonban ha mégis így történne, akkor a felszerelésből hallható egy jelzés, amely informálja az operátort arról, hogy a maximumhoz vezérlete a robotot, így elkerülhető a fogaskerekek sérülése, esetleges törése.

StellarPad Tiva C Series TM4C123GH6PMI mikrokontroller



10. ábra A mikrokontroller lábkiosztása; forrás: [7]

A StellarPad népszerű azok körében, akik hobbi szinten foglalkoznak mikrokontrollerekkel és azok fejlesztői környezetével, mivel használata roppant egyszerű. Az interneten rengeteg forrás található angol és magyar nyelven egyaránt.

A mikrokontroller vagy mikrovezérlő egy lapkára integrált, többnyire vezérlési folyamatokra kialakított mikroszámítógép. Ezen a vezérlőegységen egy mikroprocesszor helyezkedik el, amelyhez a lapkán kialakított perifériás csatlakozási lehetőség is található. A forrásigénye alacsony, így költséghatékonyan és gond nélkül kezel egyszerűbb, kis számítás igénylő feladatokat. A projekt megvalósítása során arra törekedtem, hogy olyan alkatrészeket használjak, ami mind a méretigényt, mind a költséget minimalizálja.

A mikrokontroller programozásához szükségünk van az Energia nevű programra, amit a Texas Instruments honlapjáról szabadon letölthetünk.

A fejlesztőkörnyezetről

A program elindításakor az alábbi képernyő fogad, ekkor megkezdhetjük programunk írását. Az elkészített Energia programokat Sketch-nek (vázlatnak) nevezi.



11. ábra A mikrokontroller lábkiosztása; forrás: szerző által készített ábra

Minden program három részre bontható:

- Deklaráció (változók megadása)
- Setup (általános beállítások, egyes pinek felhasználásának módjának beállítása, például: digitalWrite, analogRead, ez a programrészlet egyszer fut le az indításkor)
- Loop (főprogram, ami ciklusosan fut)

Deklaráció

Az Energiában használt változókat és azok típusát itt tudjuk megadni, tetszés szerint átnevezhetjük a mikrokontrollernek bármelyik pinjét, és későbbi felhasználása során az általunk megadott névvel hivatkozhatunk rá.

Setup()

Ez a rész a mikrokontroller indításakor egyszer fut le, még a programunk előtt. Itt megadhatjuk az utasításokat, általános beállításokat, hogy melyik láb legyen ki-, illetve bemenet, továbbá ha soros kommunikációt használunk, akkor annak sebességét is ezen belül kell lerögzítenünk. Alapbeállításokban minden láb bemenetként működik, ezért nem szükséges ezeket definiálni, elég csak a kimenetekkel foglalkozni.

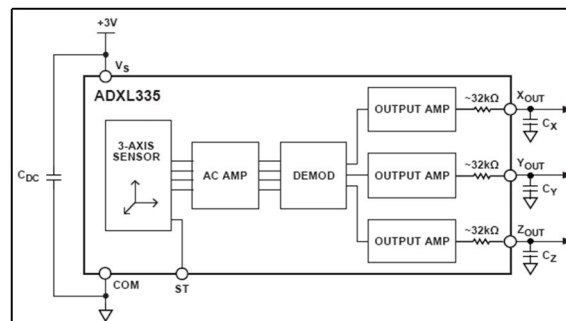
Loop()

Ez a rész tulajdonképpen a főprogram. Ennek a függvénynek az indulása közvetlenül a setup rész után történik, és végtelen ciklusként ismétlődik, amíg a mikrokontroller be van kapcsolva, tehát ha a függvény a programunk legutolsó részéhez ér, akkor visszaugrik a Loop elejére és onnan folytatja.

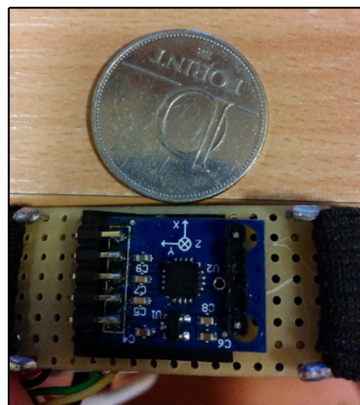
ADXL335 gyorsulásérzékelő szenzor

Kicsi, alacsony energiafogyasztású (csak 320 uA) 3-tengelyes gyorsulásmérő. Tökéletes választás mikrokontrollerekhez, mivel rendelkezik egy 3,3V-os szabályzóval, ezáltal a mikrokontroller 5V-os kimenetéről is táplálhatjuk. Jelenleg a legújabb és legjobb analóg gyorsulásmérő eszköz, kiváló példája a MEMS-nek (mikroelektromechanikai rendszerek, amelynek méretei 20 mikrométer és 1 milliméter közé eső eszközök), rendkívül alacsony zajszinttel rendelkezik.

A Földhöz képest méri a helyzetét, azaz a sebesség változását. Analóg kimenettel rendelkezik, ezáltal (x, y, z) kezelése roppant egyszerű, csupán a mikrokontroller, analóg jel fogadására alkalmas pinsorára kell csatlakoztatni (StellarPad részénél látható ábrán kékkel vannak jelölve az erre alkalmas lábak), majd a megfelelő programot létrehozni.



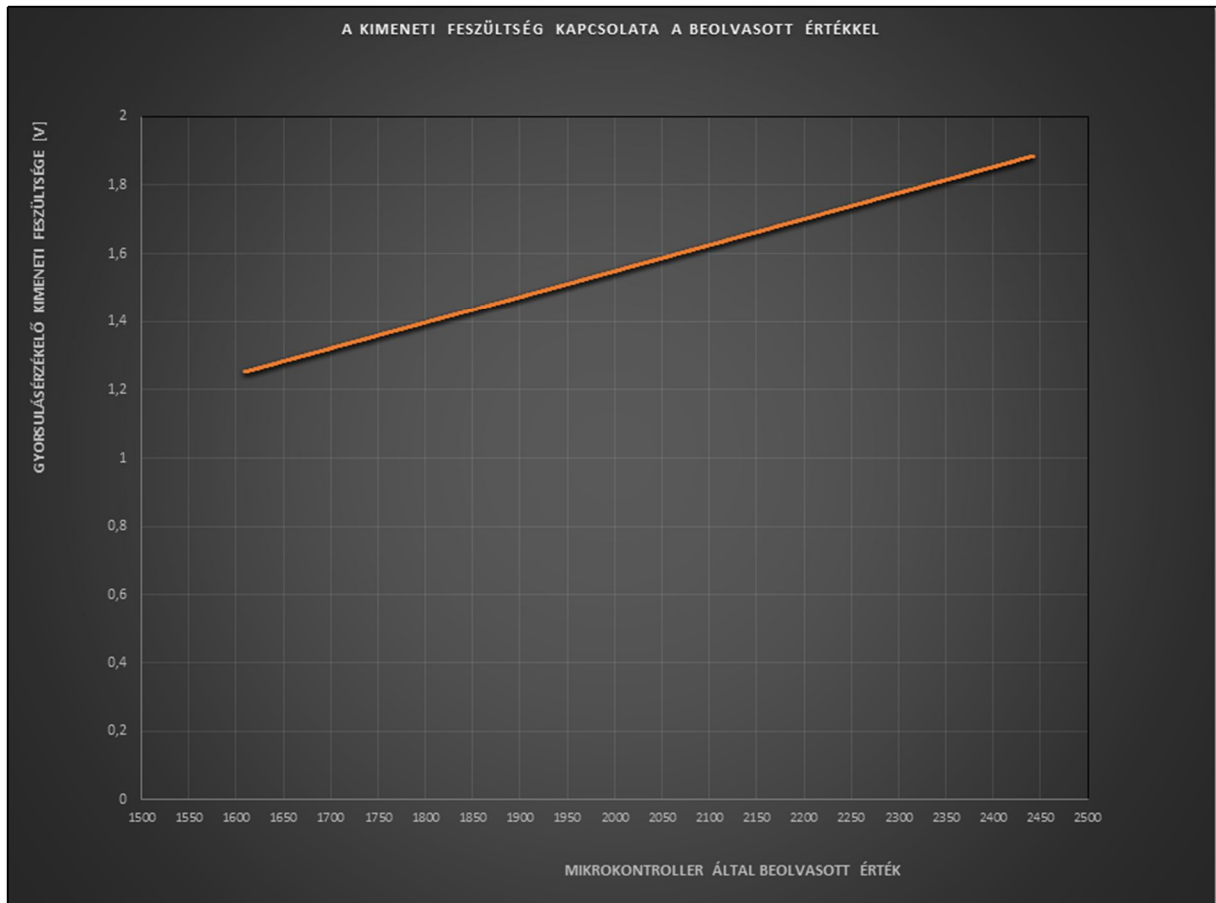
12. ábra ADXL335 blokkvázlata; forrás: [8]



13. ábra ADXL335; forrás: szerző által készített ábra

A gyorsulásmérő jelének feldolgozása

A gyorsulásmérő által generált analóg jeleket (x_1, y_1, x_2, y_2), mikrokontrollerrel beolvastatom, amely 1609 és 2442 közötti számként jelenik meg. A programomban ezekre a számokra készítettem különféle elágazásokat, mely bemeneti értékek között mit is csináljon a rendszer. A gyorsulásmérőnek 3 különféle állapotot kell létrehoznia a robotkaron. Egyszer kell, amikor plusz irányba forgatja, mínusz irányba forgatja illetve vízszintes tartandó gyorsulásmérőnek kimeneti jele semleges kell, hogy legyen, tehát a robotkarnak mozdulatlanak kell lennie.



14. ábra A mikrokontroller által beolvasott érték és a hozzá tartozó ADXL335-en mérhető kimeneti feszültség diagramja; forrás: szerző által készített ábra

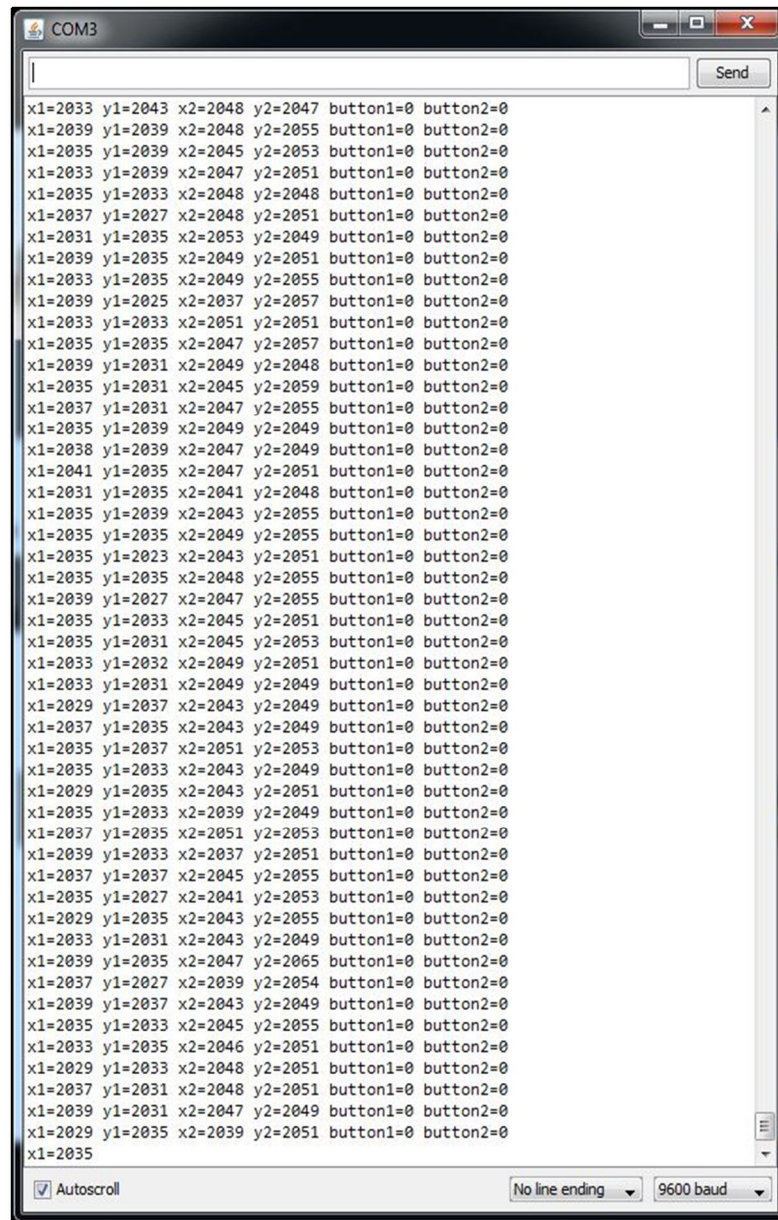
A fenti excel diagram bemutatja, hogy a gyorsulásmérő által gerjesztett feszültségekhez mely mikrokontroller által beolvasott érték tartozik, amelyet úgy határoztam meg, hogy a két szélsőértéknél feljegyeztem a mikrokontroller által kijelzett értéket, amihez megmértem az öt létrehozó gyorsulásmérőnek a kimeneti feszültségét (1609-nél 1,225V, 2442-nél 1,885V). Így kaptam egy maximum és egy minimum értéket.

Egyszerűen kivontam a maximum értékekből a minimum értékeket és képeztem a köztük lévő különbséget (2442-1609=833 és 1,885V-1,225V=0,66V). A feszültségek különbségét

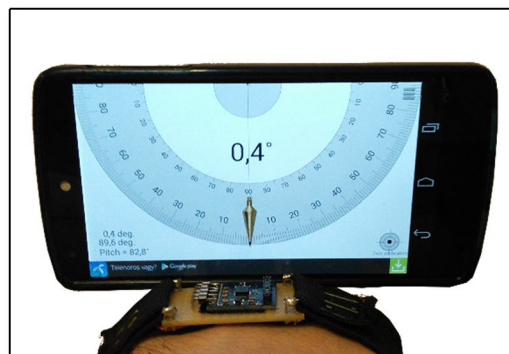
elosztottam a mikrokontroller értékének különbségével, így megkaptam mennyi a feszültségváltozás mialatt a mikrokontrollernek az analóg értéke egyet nő/csökken ($0,66V/833=0,0007923169V$). Ezután egy excel táblázatban 1609-től egészen 2442-ig kilistáztam a számokat egy megfelelő összegző függvénnyel és minden egyes számhoz hozzárendeltem a gyorsulásmérő kimeneti feszültségét. Így létrehoztam egy olyan diagramot, amelyen keresztül könnyedén tudunk tájékozódni a mikrokontroller által beolvasott érték alapján a gyorsulásmérő kimeneti feszültségére.

A gyorsulásérzékelő katalógusában a x, y kimeneteinek minimum és maximum értékei 1,35V illetve 1,65V-ként vannak dokumentálva (2. melléklet), viszont többszöri mérés alapján 1,225V és 1,885V-ként jelenik meg, ez alapján készítettem a diagramot.

Ellenőrzésképpen, amikor a kézfejemen lévő gyorsulásmérőt vízszintesen tarom, akkor a mikrokontroller által kijelzett számok alapján leolvasható a szenzor kimeneti feszültsége, hogy valóban a számításaimnak megfelelő értékű feszültségek generálódnak-e. Mivel multiméterrel többszöri mérést végeztem, ezek eredményét a következő ábrák (15., 16., 17. ábra) igazolják.



15. ábra A mikrokontroller által beolvasott gyorsulásmérő szenzor értékek közel vízszintes helyzetben; forrás: szerző által készített ábra

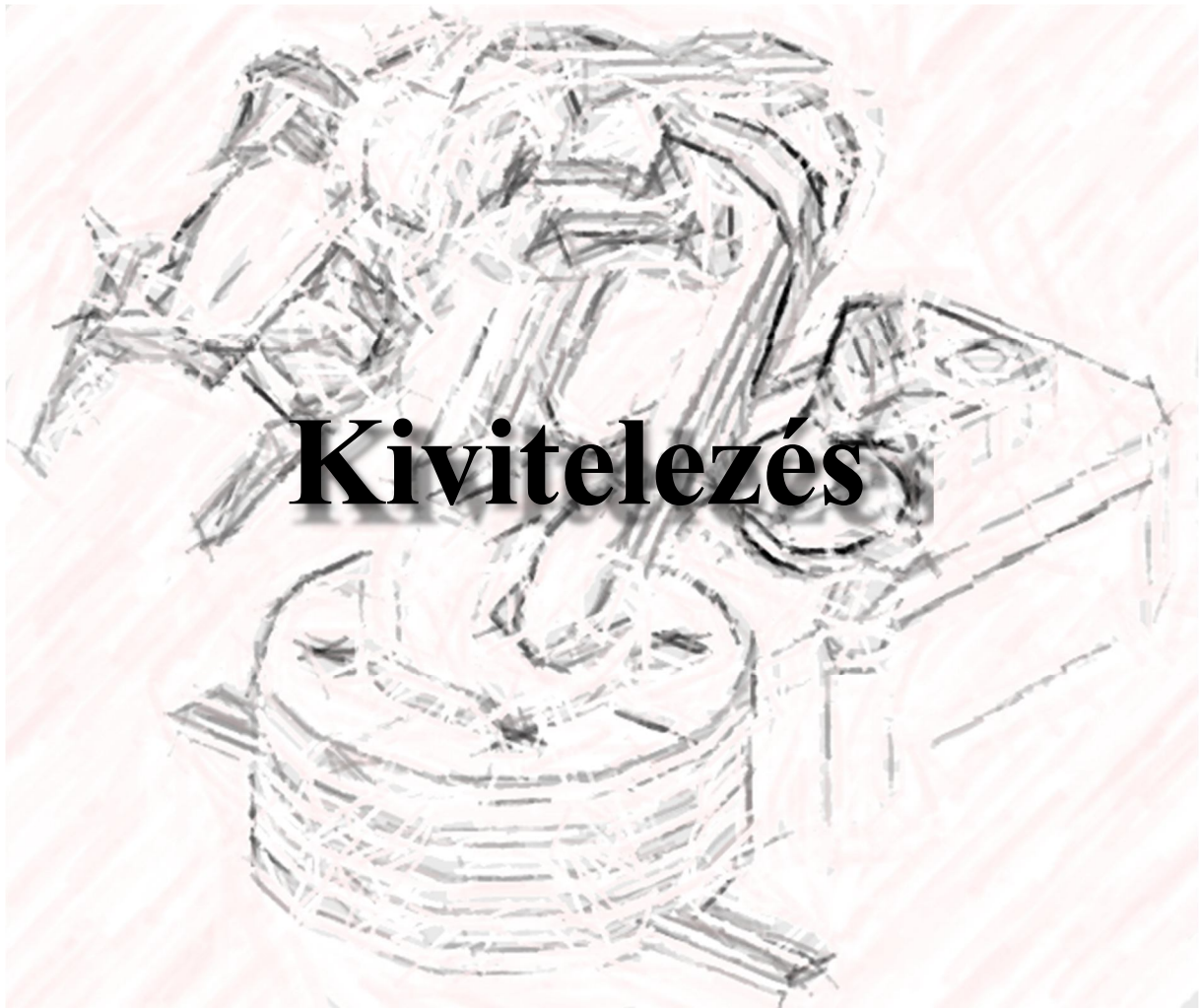


16. ábra Gyorsulásmérő vízszinteség igazolása végett; forrás: szerző által készített ábra

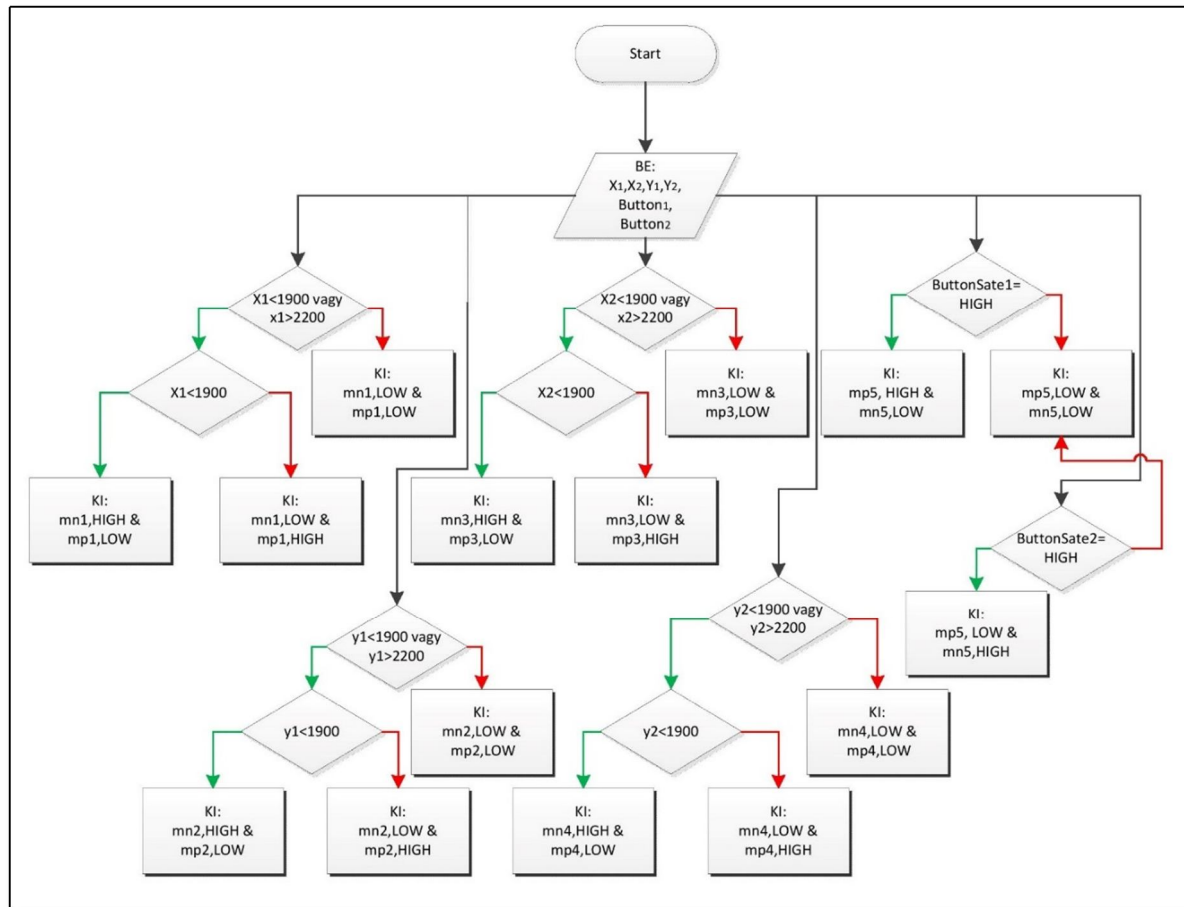


*17. ábra Gyorsulásmérő kimeneti feszültségének mérési eredménye vízszintes helyzetben;
forrás: szerző által készített ábra*

Tehát jól látható mikor vízszintesen tartom a szenzort, 2038 körüli értékek jelennek meg, a mikrokontroller programjában (15. ábra), továbbá 1,608V a multiméter által kijelzett feszültség. Ezek alapján, ha visszatekintünk az általam készített diagramra (14. ábra), akkor kis százaléknyi eltérést vehetünk észre a mért és számított értékek között..



A program folyamatábrája



18. ábra Folyamatábra; forrás: szerző által készített ábra

A folyamatábrán zölddel jelöltem az igaz, pirossal a hamis elágazásokat.

Programom felépítése

Az első részben a projecthez használandó változókat deklaráltam. Célszerű volt itt elneveztem a mikrokontroller egyes pineit, ezáltal későbbiekben elég lesz „mp1” - re hivatkozni ennek köszönhetően átláthatóbbá válik a program.

Deklaráció egy részlete a következőképpen épül fel:

...

```
const int mp1 = PB_0; // első motor
```

```
const int mn1 = PB_1;
```

```
const int mp2 = PE_4; // második motor
```

```
const int mn2 = PE_5;
```

...

A kódrészletben látható mp1 és mn1 ezek, azok a deklarált pinek, amiken keresztül ha digitális jelet adok ki, (minden tengelymozgatáshoz két-két kimenet tartozik tehát mp1, mn1, mp2, mn2... egészen mp5, mn5-ig) akkor a robot valamely tengelye megmozdul, mp=Motor Pozitív forgásirány létrehozására jó, mn=Motor Negatív, ellenkező forgásirány létrehozására alkalmas kimenet.

A következő részlet a void setup(), itt a korábban deklarált pineknek a funkcióját tudom beállítani, digitális be/ki és analóg be/ki, tulajdonképpen itt határozható meg, hogy milyen funkciót lásson el a program során adott mikrokontroller láb.

...

```
pinMode(mp1,OUTPUT);
```

```
pinMode(mn1,OUTPUT);
```

```
pinMode(mp2,OUTPUT);
```

```
pinMode(mn2,OUTPUT);
```

...

```
pinMode(buttonPin1, INPUT_PULLUP);
```

```
pinMode(buttonPin2, INPUT_PULLUP);
```

...

Az utolsó részletben három feladatot kell ellátnia, a programomnak. Elsőként a gyorsulásszenzorból beolvasott analóg jel kezelése, azaz számszerűsíteni kell, különben nem tudom felhasználni (ahogy korábban említettem a dolgozatban a bejövő jel 1609 és 2442 közötti számként jelenik meg, ez a két szám a gyorsulásmérő két szélső helyzetében van).

Továbbá ezen értékek függvényében három elágazásra van szükség, kimenet kiválthat pozitív mozgást, negatív mozgás vagy egyik sem tehát a robot nem mozdul. Valamint ebben a részben vizsgálom a nyomógomb kimenetét és továbbítom mp5 és mn5-re.

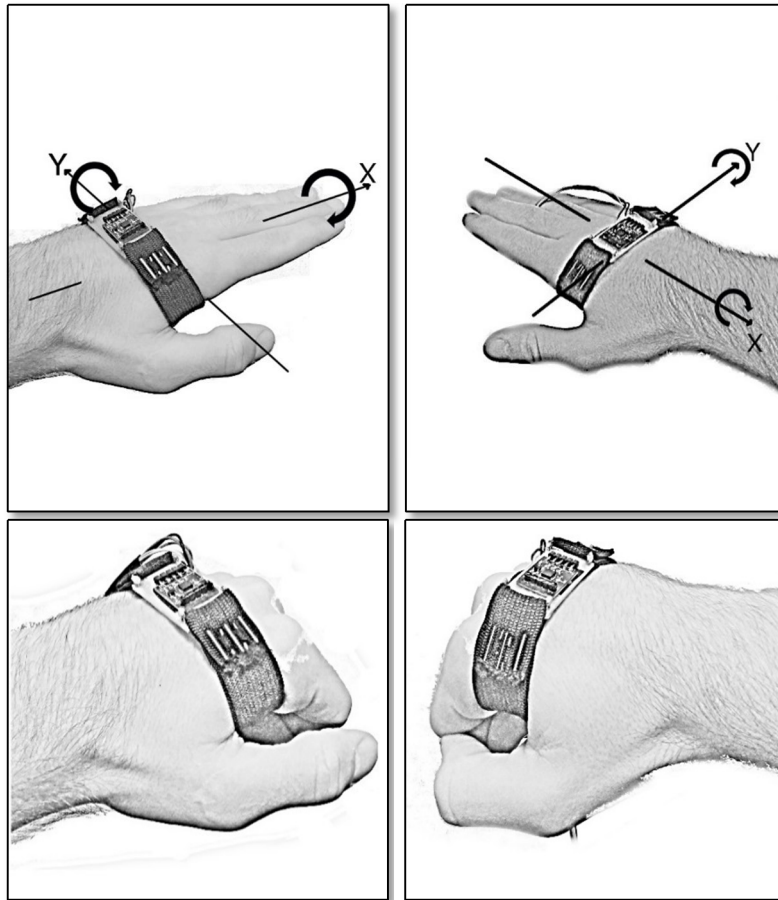
A következő táblázatban az előbb említett, három feladatra egy-egy kiragadt részlet található. A folyamatábráról (18. ábra) látszódik a program teljes kialakítása és annak lépésről-lépésre való futása.

A gyorsulásmérő szenzor analóg jelének beolvasása a következőképpen alakul	A gyorsulásmérő jelének függvényében kiadott digitális jel gerjesztése	A gomb digitális jelének beolvasása és annak függvényében kialakított digitális jel gerjesztése
<pre>int x1 = (analogRead(x1pin)); Serial.print (" "); Serial.print("x1="); Serial.print(x1); Serial.print ("\t");</pre>	<pre>if (x1 < 1900 x1 > 2200) { if(x1 < 1900) { digitalWrite(mn1,HIGH); digitalWrite(mp1,LOW); } if(x1 > 2200) { digitalWrite(mn1,LOW); digitalWrite(mp1,HIGH); } } else { digitalWrite(mn1,LOW); digitalWrite(mp1,LOW); }</pre>	<pre>buttonState1=digitalRead(buttonPin1); if (buttonState1 == HIGH) { digitalWrite(mp5, HIGH); digitalWrite(mn5, LOW); } else { digitalWrite(mp5, LOW); }</pre>

1. táblázat A program főbb folyamatainak részletei; forrás: szerző által készített táblázat

Robottengelyek mozgatása, hozzá tartozó gesztusokkal

A gyorsulásmérők kézfejen helyezkednek el, a gombok pedig a tenyérben tehát összeszorított marok esetében záródik a kapcsolat.



19. ábra Master oldalon alkalmazható kézmozdulatok; forrás: szerző által készített ábra

Az alapállapot mikor teljesen vízszintesen van a kézfej, nyitott tenyérrel, amint megdöntöm, a kézfejemet gyorsulásmérő értékei azonnal változnak, és mikrokontroller válaszol a master oldalon okozott erőhatásra, a robot rögtön reagál a távoli környezetben.

A robot tengelyek működését a következő táblázat szemlélteti, milyen gesztusra mit válaszol a robot.

	Bal kézfej	Jobb kézfej
Y tengely körül forgatva így, X érték változik	M1 irányítása	M3 irányítása
X tengely körül forgatva így, Y érték változik	M2 irányítása	M4 irányítása

2. táblázat Kézfejmozgások és robot motormozgások párosítása; forrás: szerző által készített táblázat

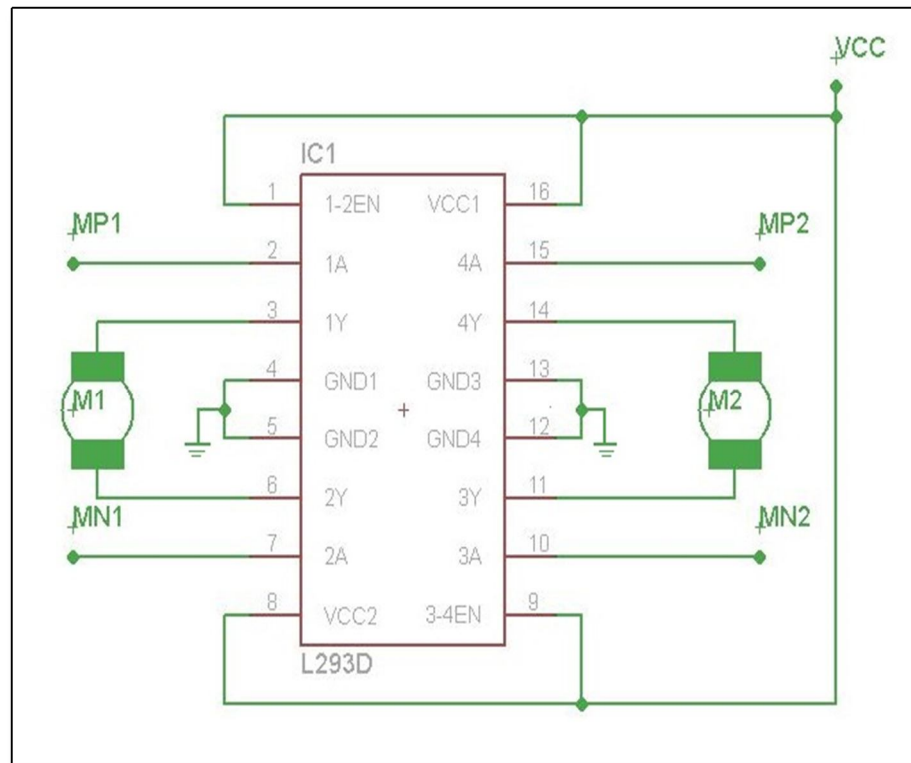
Ezek együttes kombinációjára is képes, ha az operátor kellő figyelemmel kíséri a környezetben létrejövő mozgást. A megfogó összeszorítása, a jobb tenyérben lévő gomb megnyomásakor indul meg, amint elérjük a kívánt állapotot, a gomb elengedésével az összezáró mozgás megáll. Így hát, ha kinyitni szeretnénk a megfogót, a bal tenyérben lévő gomb megnyomásával kinyithatjuk és elengedi a munkadarabot.

Saját tervezésű motorvezérlő elektronika

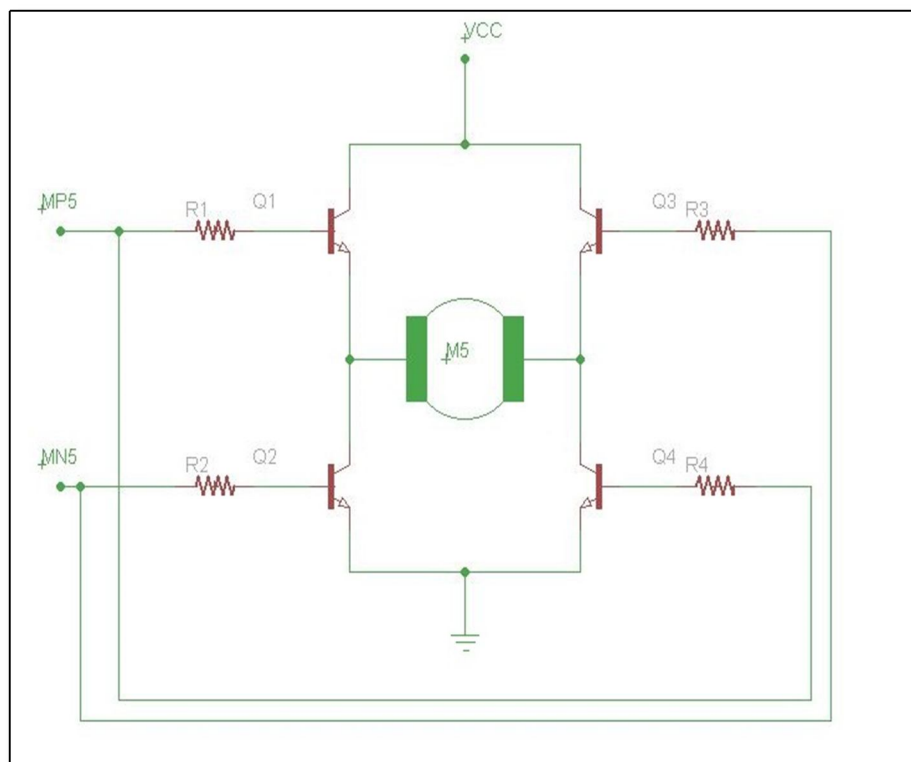
Tervezéskor sok mindent figyelembe kellett vennem, a robotkar motorjainak az áramfelvételétől egészen a költséghatékonyságig, így igyekeztem a legegyszerűbb, de mégis a leghatékonyabb motorvezérlő elektronikát kialakítani.

Az elektronikai kapcsolás kétféle motorvezérlést alkalmaz L239D IC–vel vezérlem M1, M2, M3, M4 tengelyek motorjait. Melyeknek tudom egy potenciométerrel szabályozni a tápfeszültségüket, további sebességbeli finomhangolást érhetek el. Megtalálható a tranzisztoros H-hídkapcsolással kialakított motorvezérlő egység is, melynek a kimeneti feszültségét másik potenciométerrel tudom szabályozni, amely vezérli az M5 manipulátort.

Különböző motorvezérlőkre azért van szükség, mert mind a főtengelyeknek és a manipulátornak más-más sebességgel kell működniük.



20. ábra L293D motorvezérlő IC lábkiosztása; forrás: szerző által készített ábra



21. ábra Tranzisztorokkal kiépített motorvezérlő (H-híd); forrás: szerző által készített ábra

L293D motorvezérlő IC egyszerre két motort képes vezérelni, ha MP1-re 1-et az MN1-re 0-át kapcsolunk elinul, majd a jelek értékének felcserélése után az M1 motorunk forgásiránya megváltozik, szimmetrikussága miatt ugyanez vonatkozik az M2-es motor vezérlésére.

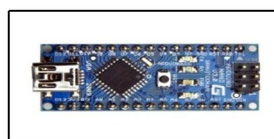
H-híd kapcsolás esetén MP5-re adunk 1-et és az MN5-re 0-át, akkor Q1 és Q4-es tranzisztorokon keresztül tápláljuk a megfogót, MP5 vezérlése esetén a Q2 és Q3-as tranzisztorok nyitnak ki és rajtuk keresztül tápláljuk. A teljes kapcsolás 1. mellékletben megtalálható.

Célkitűzés összegzése

Az úgy gondolom összegzésképpen elmondható, azon elképzelésem, hogy létrehozok egy telemanipulációval működő rendszert, teljes mértékben kivitelezésre került. Úgy gondolom, hogy a kivitelezett rendszeremet, lehetne használni telemanipuláció bemutatására, szemléltetésére, oktatásban való felhasználásra. A hallgatók programozhatják tetszés szerint a mikrokontrollert így akár más perifériás eszköz csatlakoztatása után irányíthatják is. Felméréseket lehetne végezni azzal kapcsolatban, hogy az ember öregedése során, miképpen hanyatlik tanulékonyága, koncentrációs, koordinációs készsége. A vizsgált célszemély részesülne egy rövid bemutatóval, hogy melyik kézmozdulattal milyen mozgást képes a robottal végeztetni. Majd átadni neki az irányítás lehetőségét, és vizsgálni egy egyszerű, ide-oda rakosgatós feladat keretein belül miképpen képes kivitelezni az elvárt műveletet, mialatt vizsgálni lehetne a páciens agyműködését.

A rendszerem továbbfejlesztése

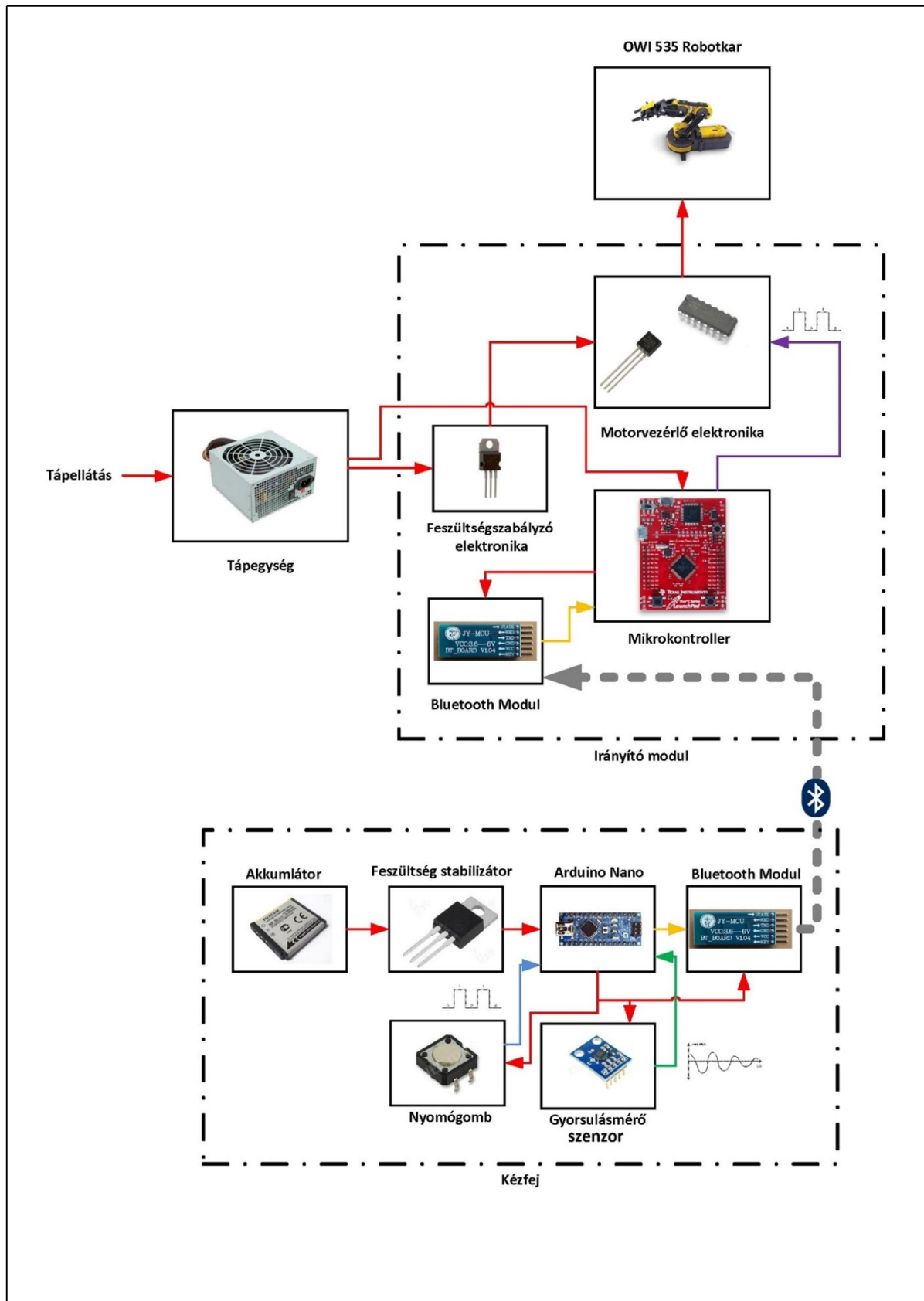
Ebben a kivitelezésben már vezeték nélkül kommunikálna a gyorsulásérzékelő szenzor és a mikrokontroller. Ehhez szükségem lenne két Arduino Nano mikrokontrollerre, amik feldolgoznák majd Bluetooth Modemmel továbbítanák a fő mikrovezérlőhöz a gyorsulásérzékelő analóg jeleit.



22. ábra Arduino Nano; forrás: [9]



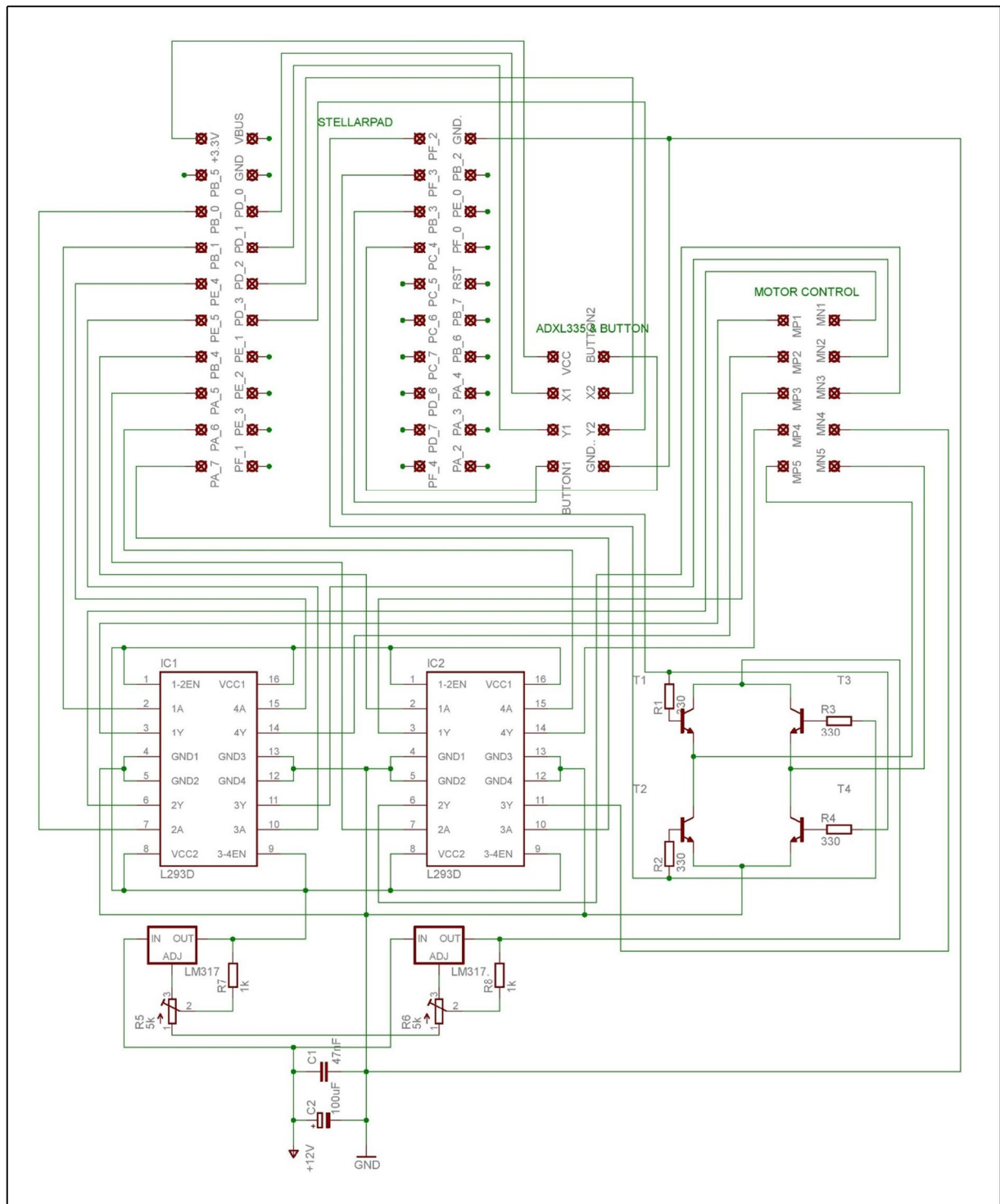
23. ábra Bluetooth Modem; forrás: [10]



24. ábra A továbbfejlesztett project blokkvázlata; forrás: szerző által készített ábra



Kelt.: Budapest, 2014 November



1. melléklet A robotvezérlő elektronikai kapcsolási rajza; forrás: szerző által készített melléklet

ADXL335

SPECIFICATIONS

$T_A = 25^\circ\text{C}$, $V_S = 3\text{ V}$, $C_X = C_Y = C_Z = 0.1\text{ }\mu\text{F}$, acceleration = 0 g, unless otherwise noted. All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

Table 1.

Parameter	Conditions	Min	Typ	Max	Unit
SENSOR INPUT	Each axis				
Measurement Range		± 3	± 3.6		g
Nonlinearity	% of full scale		± 0.3		%
Package Alignment Error			± 1		Degrees
Interaxis Alignment Error			± 0.1		Degrees
Cross-Axis Sensitivity ¹			± 1		%
SENSITIVITY (RATIOMETRIC) ²	Each axis				
Sensitivity at X_{OUT} , Y_{OUT} , Z_{OUT}	$V_S = 3\text{ V}$	270		3	mV/g
Sensitivity Change Due to Temperature ³	$V_S = 3\text{ V}$		± 0.01		%/ $^\circ\text{C}$
ZERO g BIAS LEVEL (RATIOMETRIC)					
0 g Voltage at X_{OUT} , Y_{OUT}	$V_S = 3\text{ V}$	1.35	1.5	1.65	V
0 g Voltage at Z_{OUT}	$V_S = 3\text{ V}$		1.5	1.8	V
0 g Offset vs. Temperature			± 1		mg/ $^\circ\text{C}$
NOISE PERFORMANCE					
Noise Density X_{OUT} , Y_{OUT}			150		$\mu\text{g}/\sqrt{\text{Hz}}$ rms
Noise Density Z_{OUT}			300		$\mu\text{g}/\sqrt{\text{Hz}}$ rms
FREQUENCY RESPONSE ⁴					
Bandwidth X_{OUT} , Y_{OUT} ⁵	No external filter		1600		Hz
Bandwidth Z_{OUT} ⁵	No external filter		550		Hz
R_{FILT} Tolerance			$32 \pm 15\%$		k Ω
Sensor Resonant Frequency			5.5		kHz
SELF-TEST ⁶					
Logic Input Low			+0.6		V
Logic Input High			+2.4		V
ST Actuation Current			+60		μA
Output Change at X_{OUT}	Self-Test 0 to Self-Test 1	-150	-325	-600	mV
Output Change at Y_{OUT}	Self-Test 0 to Self-Test 1	+150	+325	+600	mV
Output Change at Z_{OUT}	Self-Test 0 to Self-Test 1		5	+1000	mV
OUTPUT AMPLIFIER					
Output Swing Low	No load		0.1		V
Output Swing High	No load		2.8		V
POWER SUPPLY					
Operating Voltage Range		1.8		3.6	V
Supply Current	$V_S = 3\text{ V}$		350		μA
Turn-On Time ⁷	No external filter		1		ms
TEMPERATURE					
Operating Temperature Range		-40		+85	$^\circ\text{C}$

¹ Defined as coupling between any two axes.

² Sensitivity is essentially ratiometric to V_S .

³ Defined as the output change from ambient-to-maximum temperature or ambient-to-minimum temperature.

⁴ Actual frequency response controlled by user-supplied external filter capacitors (C_X , C_Y , C_Z).

⁵ bandwidth = 0.5 Hz.
 $C_X = C_Y = 0.003\text{ }\mu\text{F}$, bandwidth = 1.6 kHz. For $C_Z = 0.01\text{ }\mu\text{F}$, bandwidth = 500 Hz. For C_X , C_Y , $C_Z = 10\text{ }\mu\text{F}$, bandwidth = 0.5 Hz.

⁶ Self-test response changes cubically with V_S .

⁷ Turn-on time is dependent on C_X , C_Y , C_Z and is approximately $160 \times C_X$ or C_Y or $C_Z + 1\text{ ms}$, where C_X , C_Y , C_Z are in microfarads (μF).

2. melléklet Részlet a gyorsulásmérő szenzor adatlapjából; forrás: [11]

```
const int mp1 = PB_0; // első motor
const int mn1 = PB_1;
const int mp2 = PE_4; // második motor
const int mn2 = PE_5;
const int mp3 = PB_4; // harmadik motor
const int mn3 = PA_5;
const int mp4 = PA_6; // negyedik motor
const int mn4 = PA_7;
const int mp5 = PF_2; // ötödik motor
const int mn5 = PF_3;
const int x1pin = PD_0; // első gyorsulásmérő
const int y1pin = PD_1;
const int x2pin = PD_2; // második gyorsulásmérő
const int y2pin = PD_3;

const int buttonPin1 = PB_3; // gombok
const int buttonPin2 = PC_4;

int buttonState1 = 0;
int buttonState2 = 0;
int gombv1 = 0;
int gombv2 = 0;

void setup()
{
    Serial.begin(9600);

    pinMode(mp1, OUTPUT);
    pinMode(mn1, OUTPUT);
    pinMode(mp2, OUTPUT);
    pinMode(mn2, OUTPUT);
    pinMode(mp3, OUTPUT);
    pinMode(mn3, OUTPUT);
    pinMode(mp4, OUTPUT);
    pinMode(mn4, OUTPUT);
    pinMode(mp5, OUTPUT);
    pinMode(mn5, OUTPUT);

    pinMode(buttonPin1, INPUT_PULLUP);
    pinMode(buttonPin2, INPUT_PULLUP);
}

void loop()
{
    Serial.print("\n");

    int x1 = (analogRead(x1pin)); //gyorsulásmérő x1 értékének be
    Serial.print("x1=");
    Serial.print(x1);
    Serial.print("\t");
    int y1 = (analogRead(y1pin)); //gyorsulásmérő y1 értékének be
    Serial.print("y1=");
    Serial.print(y1);
    Serial.print("\t");

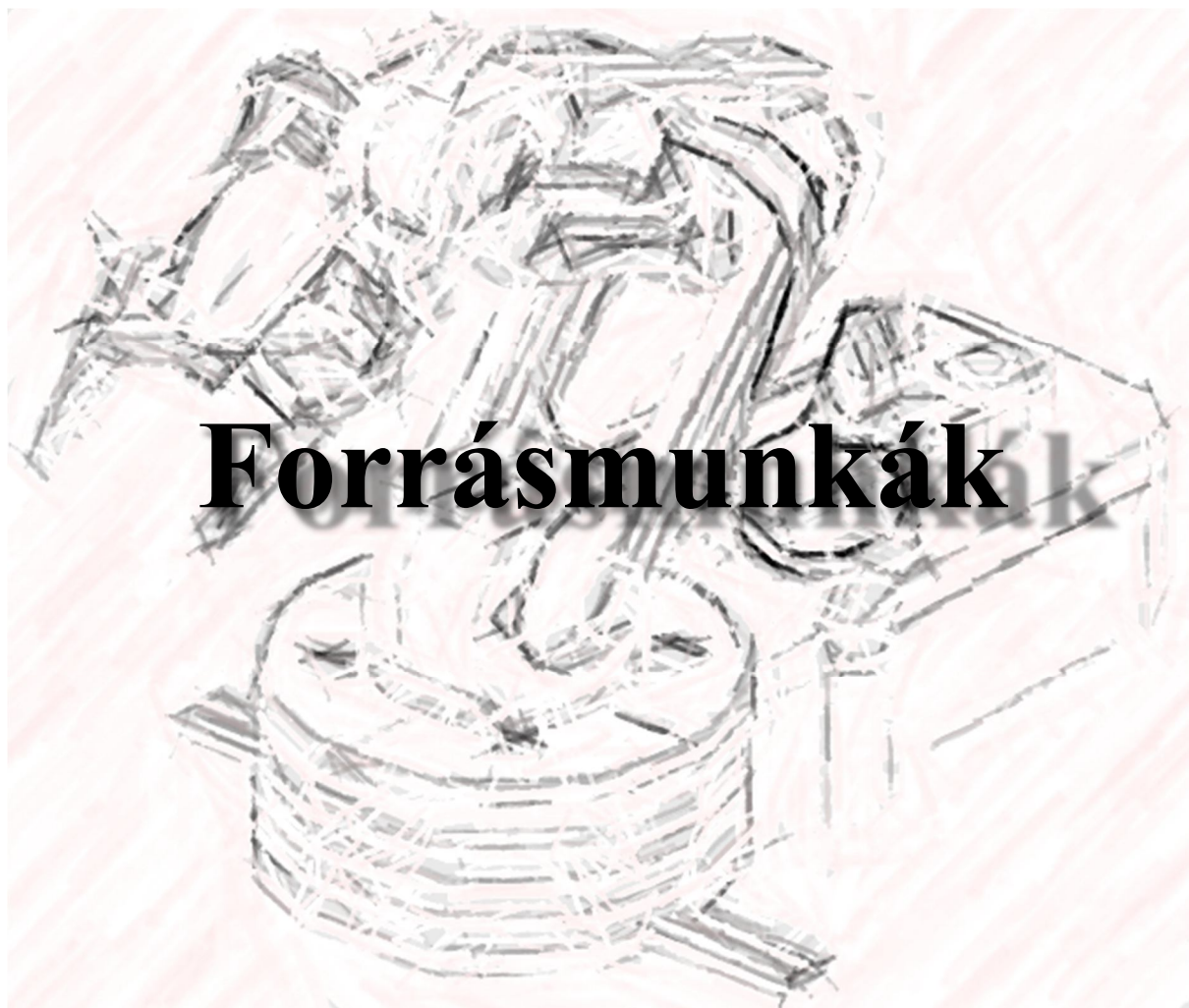
    int x2 = (analogRead(x2pin)); //gyorsulásmérő x2 értékének b
    Serial.print("x2=");
    Serial.print(x2);
    Serial.print("\t");

    int y2 = (analogRead(y2pin)); //gyorsulásmérő y2 értékének b
    Serial.print("y2=");
    Serial.print(y2);
    Serial.print("\t");
```

```
//M1 irányítása//////////  
  
if (x1 < 1900 || x1 > 2200)  
{  
    if(x1 < 1900)  
    {  
        digitalWrite(mn1,HIGH);  
        digitalWrite(mp1,LOW);  
    }  
  
    if(x1 > 2200)  
    {  
        digitalWrite(mn1,LOW);  
        digitalWrite(mp1,HIGH);  
    }  
}  
else  
{  
    digitalWrite(mn1,LOW);  
    digitalWrite(mp1,LOW);  
}  
  
//M2 irányítása//////////  
  
if (y1 < 1900 || y1 > 2200)  
{  
    if(y1 < 1900)  
    {  
        digitalWrite(mn2,HIGH);  
        digitalWrite(mp2,LOW);  
    }  
  
    if(y1 > 2200)  
    {  
        digitalWrite(mn2,LOW);  
        digitalWrite(mp2,HIGH);  
    }  
}  
else  
{  
    digitalWrite(mn2,LOW);  
    digitalWrite(mp2,LOW);  
}  
  
//M3 irányítása//////////  
  
if (x2 < 1900 || x2 > 2200)  
{  
    if(x2 < 1900)  
    {  
        digitalWrite(mn3,HIGH);  
        digitalWrite(mp3,LOW);  
    }  
  
    if(x2 > 2200)  
    {  
        digitalWrite(mn3,LOW);  
        digitalWrite(mp3,HIGH);  
    }  
}  
else  
{  
    digitalWrite(mn3,LOW);  
    digitalWrite(mp3,LOW);  
}  
}
```

```
//M4 irányítása//////////  
  
if (y2 < 1900 || y2 > 2200)  
{  
    if(y2 < 1900)  
    {  
        digitalWrite(mn4,HIGH);  
        digitalWrite(mp4,LOW);  
    }  
  
    if(y2 > 2200)  
    {  
        digitalWrite(mn4,LOW);  
        digitalWrite(mp4,HIGH);  
    }  
}  
else  
{  
    digitalWrite(mn4,LOW);  
    digitalWrite(mp4,LOW);  
}  
  
//M5 irányítása//////////  
  
Serial.print("button1=");  
Serial.print(buttonState1);  
Serial.print (" ");  
  
buttonState1 = digitalRead(buttonPin1);  
  
if (buttonState1 == HIGH)  
{  
    digitalWrite(mp5, HIGH);  
    digitalWrite(mn5, LOW);  
}  
else  
{  
    digitalWrite(mp5, LOW);  
}  
  
Serial.print("button2=");  
Serial.print(buttonState2);  
  
buttonState2 = digitalRead(buttonPin2);  
  
if (buttonState2 == HIGH)  
{  
    digitalWrite(mn5, HIGH);  
    digitalWrite(mp5, LOW);  
}  
else  
{  
    digitalWrite(mn5, LOW);  
}  
}
```

3. melléklet Teljes forráskód; forrás: szerző által készített melléklet



Kelt.: Budapest, 2014 November

Felhasznált irodalom

- Wikipedia → Robotok alkalmazása
- Fantazmo.hu → Amit tudni akartál a robotokról
- Nanotechnologia.blog.hu → Nanorobotok a gyógyításban
- Moodle.autolab.uni-pannon.hu → Telemanipulációs rendszerekről
- Robotshop.com → OWI 535 Robotic Arm Edge
- Ti.com → Tiva™ C Series LaunchPad Evaluation Kit datasheet
- Energia.nu → Fejlesztői környezet használata
- Dr. Schuster György → C programozási nyelv
- Sparkfun.com → ADXL335 datasheet

Idézett forrásmunkák

- [1] [Online]. Available:
http://www.esa.int/var/esa/storage/images/esa_multimedia/images/2006/03/artist_s_impression_of_the_european_robotic_arm_era/9782059-3-eng-GB/Artist_s_impression_of_the_European_Robotic_Arm_ERA.jpg.
[Hozzáférés dátuma: 13 11 2014].
- [2] [Online]. Available:
https://saintpaul-a-gymnasiou-2012-13.wikispaces.com/file/view/unimate_2000.jpg/409347722/329x226/unimate_2000.jpg.
[Hozzáférés dátuma: 13 11 2014].
- [3] [Online]. Available:
<http://www.winthrop.org/departments/institutes/family/ob-gyn/robotic-surgery/daVinci-Si-HD-Surgical-System/davinci-robot.jpg>
[Hozzáférés dátuma: 13 11 2014].
- [4] [Online]. Available:
http://nanotechnologia.blog.hu/2009/09/23/a_verunkben_van_nanorobotok_a_gyogytasban [Hozzáférés dátuma: 13 11 2014].
- [5] [Online]. Available:
<http://www.mimicsoda.hu/Files/Image/kempelen2.jpg>.
[Hozzáférés dátuma: 13 11 2014].
- [6] [Online]. Available:
http://2.bp.blogspot.com/-rHA7WN_7W9s/T8fUbVrImvI/AAAAAAAAAAU/e1v4zZtRfPQ/s1600/Nuevo+Imagen+de+mapa+de+bitt.bmp.
[Hozzáférés dátuma: 13 11 2014].
- [7] [Online]. Available:
http://embeddedcomputing.weebly.com/uploads/1/1/6/2/11624344/3560417_orig.jpg.
[Hozzáférés dátuma: 13 11 2014].
- [8] [Online]. Available:
http://www.analog.com/static/imported-files/images/functional_block_diagrams/ADXL335_fbl.png.
[Hozzáférés dátuma: 13 04 2014].
- [9] [Online]. Available:
http://arduino.cc/en/uploads/Main/ArduinoNanoFront_3_sm.jpg.
[Hozzáférés dátuma: 13 04 2014].
- [10] [Online]. Available:
http://www.arduino-hacks.com/wp-content/uploads/2014/08/DSC_0543.jpg.
[Hozzáférés dátuma: 13 04 2014].
- [11] [Online]. Available:
http://www.analog.com/static/imported-files/data_sheets/ADXL335.pdf.
[Hozzáférés dátuma: 13 04 2014].

Ábrajegyzék

1. ábra	<i>Space Shuttle robotkar; forrás: [1]</i>	6
2. ábra	<i>Unimate az első ipari felhasználású robot; forrás: [2]</i>	8
3. ábra	<i>da Vinci sebészrobot; forrás: [3]</i>	9
4. ábra	<i>Nanorobot; forrás: [4]</i>	9
5. ábra	<i>Mechanikai pantográf; forrás: [5]</i>	10
6. ábra	<i>Master-slave kapcsolat blokkvázlata; forrás: szerző által készített ábra</i>	11
7. ábra	<i>A project blokkvázlata; forrás: szerző által készített ábra</i>	15
8. ábra	<i>OWI 535 robotkar; forrás: szerző által készített ábra</i>	17
9. ábra	<i>A robotkar megengedhető mozgástere; forrás: [6]</i>	17
10. ábra	<i>A mikrokontroller lábkiosztása; forrás: [7]</i>	18
11. ábra	<i>A mikrokontroller lábkiosztása; forrás: szerző által készített ábra</i>	19
12. ábra	<i>ADXL335 blokkvázlata; forrás: [8]</i>	20
13. ábra	<i>ADXL335; forrás: szerző által készített ábra</i>	20
14. ábra	<i>A mikrokontroller által beolvasott érték és a hozzá tartozó ADXL335-en mérhető kimeneti feszültség diagramja; forrás: szerző által készített ábra</i>	21
15. ábra	<i>A mikrokontroller által beolvasott gyorsulásmérő szenzor értékek közel vízszintes helyzetben; forrás: szerző által készített ábra</i>	23
16. ábra	<i>A gyorsulásmérő vízszinteség igazolása végett; forrás: szerző által készített ábra</i>	23
17. ábra	<i>A gyorsulásmérő kimeneti feszültségének mérési eredménye; forrás: szerző által készített ábra</i>	24
18. ábra	<i>Folyamatábra; forrás: szerző által készített ábra</i>	26
19. ábra	<i>Master oldalon alkalmazható kézmozdulatok; forrás: szerző által készített ábra</i>	29
20. ábra	<i>L293D motorvezérlő IC lábkiosztása; forrás: szerző által készített ábra</i>	31
21. ábra	<i>Tranzisztorokkal kiépített motorvezérlő (H-híd; forrás: szerző által készített ábra</i>	31
22. ábra	<i>Arduino Nano; forrás: [9]</i>	32
23. ábra	<i>Bluetooth Modem; forrás: [10]</i>	32
24. ábra	<i>A továbbfejlesztett project blokkvázlata; forrás: szerző által készített ábra</i>	33