

TMDK – dolgozat

Méhes Jozef

Villamosmérnöki (BSc) – 3. évfolyam

Széchenyi István Egyetem

Konzulens: Dr. Hidvégi Timót – egyetemi docens

2012

Holdjáró
Moonwalker

Kézirat lezárása: 2012. április 19.

TARTALOMJEGYZÉK

BEVEZETŐ	1
AZ ALVÁZ	2
JOYSTICK	5
RF ADÓ-VEVŐ MODUL	9
RFM12B	9
RFM23B	12
TELJESÍTMÉNY-FOKOZAT	16
A HOLDJÁRÓ	18
AZ ADÓÁLLOMÁS	21
JÖVŐBELI TERVEK	25
Automata üzemmód	25
CMOS kamera, mint a Holdjáró szeme	27
VÉGSZÓ	28
IRODALOMJEGYZÉK	29

BEVEZETŐ

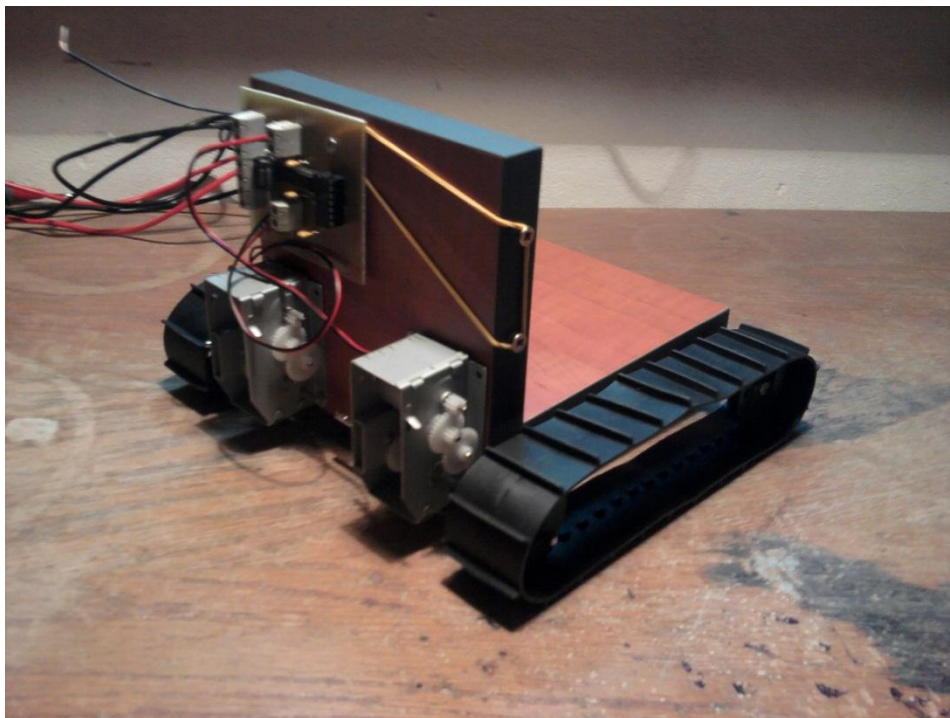
Mindig is szerettem otthon barkácsolni, ezért amikor megkezdtem a tanulmányaimat a Széchenyi István Egyetem Villamosmérnöki szakán, rögtön tudtam, hogy idővel bele fogok vágni egy komolyabb projektbe, amelynek a fejlesztése segít elmélyíteni bennem az előadásokon hallottakat, illetve amelyből valamelyest belekóstolhatok a mérnöki munka szépségeibe és nehézségeibe.

Érdeklődésem a beágyazott rendszerek és a távirányítás alkalmazási területei felé irányult. Ez idő tájt kezdtem el foglalkozni a Microchip vállalat PIC mikrovezérlőivel. A gyártó saját fordítója helyett a MikroElektronika vállalat Pascal fordítóját kezdtem el tanulmányozni és használni annak kiváló terméktámogatása és példaprogramjaik gazdag gyűjteménye miatt. Tanulásra kitűnő volt, mostanra viszont már áttértem a C18-as fordítóra, ugyanis az emberek többség ezt használja, és itt az egyetemen is ezt tanuljuk.

Miután megismerkedtem a 16F-es architektúra képességeivel, elkezdtem keresni egy olyan témát, ami érdekel, ami megvalósítható egy mikrovezérlővel, és amely nem mutat túl a képességeimen. Így jött az ötlet, hogy tervezni és építeni kellene egy olyan elektromotor meghajtású miniatűr járművet, amelyet vezeték nélkül lehet vezérelni, tud tájékozódni a térben, és autonóm módon önmagát is tudja irányítani. A név már csak hab a tortán - annyit elárulhatok, hogy az ötletet hozzá az egyik kedvenc énekesem adta...

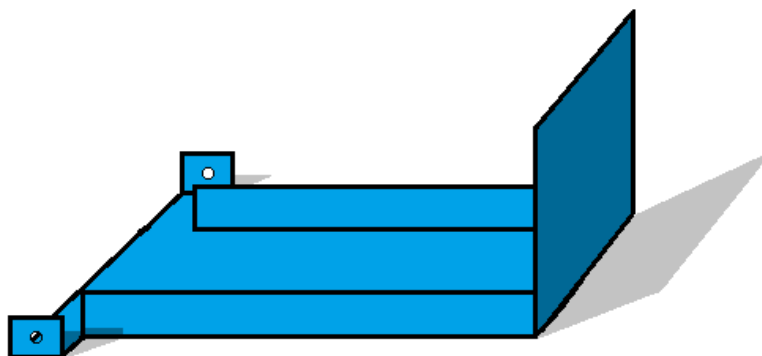
AZ ALVÁZ

Azt hiszem, elmondhatom, hogy az alváz az egyedüli elem, ami a legnagyobb fejlődésen ment keresztül. A kezdetek kezdetén az volt a koncepció, hogy a járgány rossz terepviszonyok esetén is tudjon közlekedni: sűrű szőnyeg, küszöb, esetleg kavicsos talaj. A megoldást kerekek és hernyótalpak (modellüzletben árusított gumi lánctalp) használatában láttam. Egyértelmű volt tehát, hogy a kerekek fixen lesznek rögzítve az alvázhoz, vagyis egyik kerék se fog tért, nem úgy, mint például egy személyautónál. Törekedtem az egyszerűsége, ezért az alváz fából készült, nagyjából 20x20 centiméteres alapterülettel. A járgány mindkét oldalán egy-egy hernyótalp volt, ezért a meghajtást két áttételes modellmotor biztosította. A hátsó oldalra volt felerősítve a teljesítmény-fokozat, melynek a tápellátását 6db sorba kötött, egyenként 1,2V-os NiMH akku biztosította. Maga a vezérlő elektronika dugaszolós próbanyákon volt összerakva, és saját tápfeszültség-forrással rendelkezett. A mikrovezérlő programja bug-os volt: néha lefagyott, de az is előfordult, hogy a motorok által keltett zaj hatására párszor újraindult. Az irányítás itt még nem volt vezeték nélküli. Használt, régi gameportos joystick tette lehetővé a vezérlést. A joystick jele ekkor még kis felbontással volt feldolgozva, ezért nem volt folyamatos a járgány gyorsulása, ha megmozdítottuk a botkormányt. Azóta ez változott, bővebben a *Joystick* c. fejezetben.



Hamar rá kellett jönnöm, hogy ez a konstrukció nem életképes, mivel a nagy súrlódás miatt a járgány mozgása elég lomha lett. Előre és hátra megfelelően haladt, de a menet közbeni kanyarodással nehézségei voltak, továbbá képtelen volt egy helyben elfordulni. A hernyótalp elég feszes volt, de néha megesett, hogy kanyarodáskor egyszerűen lefutott a kerékről.

A probléma gyökerét abban láttam, hogy a Holdjáró nem elég széles, vagyis túl közel vannak egymáshoz a hernyótalpak. Ha elképzelünk egy katonai tankot, az viszonylag keskeny a hosszához képest, de láncaljai nem a tank teljes hosszában érintkeznek a talajjal. Szóval a következő generációs Holdjáró immár szélesebb lett, és milliméteres vaslemezről készült az alváza. Még mindig két motor mozgatta a járgányt, viszont valamit nagyon elszámoltam, mivel túl nehéz lett, és így végeredményben a mozgás dinamikája semmit se javult. Ezután egy időre pihentettem a projektet, mással foglalkoztam.



Csaknem egy év telt el, mire újból előkerült a szekrény mélyéről a setesuta Holdjáró. Szerettem volna végre az elektronikával foglalkozni, de tudtam, hogy egy jármű – még ha csak miniatűr is – alapja a jó alváz.

A mostani verzió tehát alumínium lemezből készült, alakjában megegyezett az előző változattal. Lényegi különbség a kettő között a 4 motoros meghajtás és a hernyótalpak kerekekre cserélése.

Rájöttem, hogy a hernyótalpas változat nem életképes, mivel ezek a kisteljesítményű motorok – még ha áttételesek is – nem rendelkeznek kellő nyomatékkal. Így hát a járgány immár műanyag kerekeken gurul, ezáltal sokkal mozgékonyabb lett. Jelenlegi állapotában minden kerékre jut egy motor.

JOYSTICK

A joysticknek (magyarul botkormány) a kézi irányításban van szerepe. Kezdetben, amikor még nem volt a távirányítás megvalósítva, a joystick közvetlenül a Holdjáróra csatlakozott, ma már az adóállomásra van kapcsolva. A botkormány bedöntésével és a rajta lévő egyik nyomógommbal lehet mozgásra bírni a járgányt. Menet közben tud kanyarodni, illetve nyomva a gombot álló helyzetben forogni.

Először az egyik barátom által felajánlott joysticket kezdtem el használni, viszont mostanra már beszereztem egy jobb állapotban lévő, szintúgy használt darabot.

Egy gameport csatlakozóval rendelkező joystickról azt kell tudni, hogy a számítógép a botkormány bedőlésének a szögét azokból az ellenállásokból határozza meg, amelyeket a belül lévő, a botkormány által elcsavart potenciométerek szolgáltatnak. Tehát az összes potenciométer középső lába tápra van kapcsolva, és az egyik szélső lábuk pedig ki van vezetve. Ha jól tudom, a számítógépen belül ez a változó ellenállás egy ellenállás – frekvencia jelátalakítóra kerül (pl. NE555-tel felépített oszcillátor). A nyomógombok túli a gameport csatlakozóban alapállapotban lebegnek vagy magas szinten vannak (típusfüggő), megnyomásuk esetén alacsony szintre váltanak.

Hogy egyszerűsítsem a joystick kezelését, bizonyos módosításokat hajtottam végre rajta. Az általam használt mikrovezérlő tartalmaz analóg-digitális átalakítót (ADC), ezért célszerű volt a joystick potenciométereit úgy átforrasztani, hogy azok feszültséget osszanak le a csavarodásuk függvényében. Mivel ezek a potenciométerek 100k Ω ellenállásúak, elképzelhetőnek tartottam, hogy nem fog megfelelően működni az átalakítás – nem fog tudni időben feltöltődni az ADC mintavételező és tartó kondenzátora, de szerencsére nem származott ebből semmiféle probléma, és így nem volt szükség műveleti erősítős feszültségkövető áramkörökre. A biztonság kedvéért azért az ADC mintavételezési idejét a maximumra állítottam.

A joystick kezeléséhez saját programkönyvtárat írtam, melyben makrókkal és definiálásokkal végezhethetjük el a különböző beállításokat:

```
#define joystick_x_ch      // pl. ADC_CH0 (adc.h csatorna makró)
#define joystick_x_tris    // a csat. irányát beállító reg. bitje
```

```

#define joystick_y_ch      // p1. ADC_CH1 (adc.h csatorna makró)
#define joystick_y_tris    // a csat. irányát beállító reg. bitje

#define use_analog_x       // az x - tekerő használata
#define analog_x_ch        // p1. ADC_CH2 (adc.h csatorna makró)
#define analog_x_tris      // a csat. irányát beállító reg. bitje

#define use_analog_y       // az y - tekerő használata
#define analog_y_ch        // p1. ADC_CH3 (adc.h csatorna makró)
#define analog_y_tris      // a csat. irányát beállító reg. bitje

#define use_button1        // B1 nyomógomb használata
#define button1            // B1 nyomógomb olvasó reg. bitje
#define button1_tris       // B1 nyomógomb irányát beáll. reg. bitje

#define use_button2        // B2 nyomógomb használata
#define button2            // B2 nyomógomb olvasó reg. bitje
#define button2_tris       // B2 nyomógomb irányát beáll. reg. bitje

#define use_button3        // B3 nyomógomb használata
#define button3            // B3 nyomógomb olvasó reg. bitje
#define button3_tris       // B3 nyomógomb irányát beáll. reg. bitje

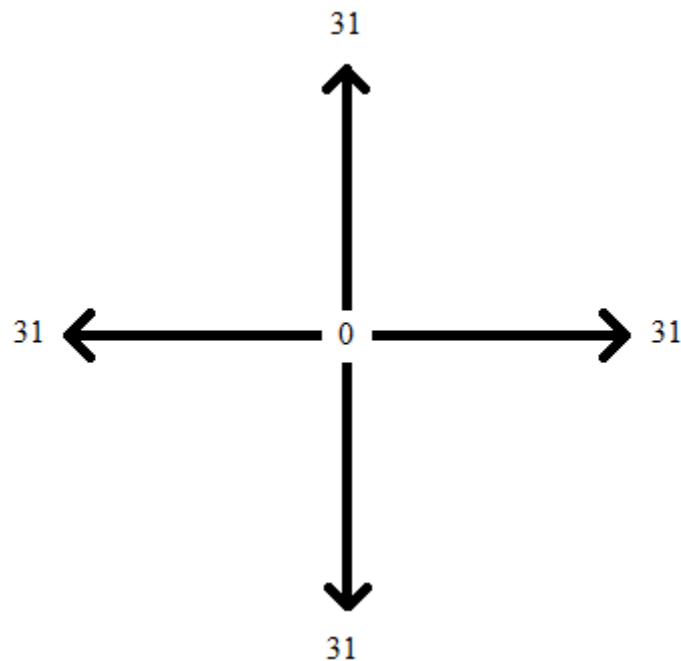
#define use_button4        // B4 nyomógomb használata
#define button4            // B4 nyomógomb olvasó reg. bitje
#define button4_tris       // B4 nyomógomb irányát beáll. reg. bitje

#define joystick_init_adc() // adc beállítási függvények
#define joystick_plug_detect // csatl. detekt. olvasó reg. bitje
#define joystick_plug_detect_tris // irányt beállító reg. bitje

```

Ezek a makrók egy *my_defs.h* header fájlban vannak elhelyezve, amely hozzá van adva a projekthez, így nincs szükség arra, hogy beállításkor a joystick programkönyvtárába belemódosítsunk.

A botkormány bedöntését két tengely mentén – X és Y – kapjuk meg. Ezek lesznek az X- és Y-irányú sebességek. Ezek a sebességek először 0-tól 9-ig terjedtek, de ma már 0-tól 31-ig, ahol a 0 értelemszerűen a zéró sebesség, 31 pedig a maximális sebesség. Minden egyes sebesség-érték egy kitöltési tényezőt képvisel, amelynek a motor-meghajtásnál van szerepe. Ezen sebesség-értékhez párosul még egy irány is – vagyis hogy merre van bedöntve a botkormány: X-tengely esetén jobbra vagy balra, Y-tengely esetén előre vagy hátra.



Mivel a joystick tartalmaz plusz két tekerő potenciométert is, ezért lehetőségünk van ezek állását is leolvastatni, de ez persze nem kötelező. Állásukat 0-tól 63-ig terjedő intervallumban kapjuk meg egész számként.

A nyomógombok közül is kiválaszthatjuk, hogy melyeket kívánjuk használni, így csak azok állapota fog ellenőrzésre kerülni.

Használható függvények:

```
unsigned char joystick_read();  
void joystick_init();
```

A *joystick_init()*; függvényt a joystick használata előtt kell meghívni. A *joystick_read()*; függvény elindít egy joystick szkennelést, és visszatérési értéként megadja, hogy a joystick csatlakozója be volt-e dugva az aljzatba. Csatlakozás hiányában a függvény nem végez se AD átalakítást, se nyomógomb tesztelést, rögtön visszatér nullás értékkel.

A botkormány, tekerők és nyomógombok állapota a joystick programkönyvtárban előre deklarált változókba kerül. Ezeket vizsgálhatjuk a *joystick_read()*; függvény lefutása után.

RF ADÓ-VEVŐ MODUL

RFM12B

Ahogy az előzőekben említettem, az irányítás kezdetben nem volt vezeték nélküli. Miután elkészült végre a végleges alváz, elkezdtem keresni olyan rádiós modulokat, amelyek megfelelnek nekem ehhez a projekthez. Mindenképp olyan típust szerettem volna alkalmazni, amely a rádiós kommunikáció lehető legnagyobb részét saját maga oldja meg.

Végül egy népszerű modulra, a HOPE Microelektronics vállalat 868MHz-es RFM12B moduljára esett a választásom. Ez az adó-vevő half-duplex módon tud kommunikálni, maximum 256kbps sebességgel. Tartalmaz két 8 bites TX adatregisztert és egy 16 bites RX FIFO puffert. A tápfeszültsége 2,2V-tól 3,8V-ig terjed. Ezzel a modullal SPI buszon keresztül lehet kommunikálni, az adatlapja tartalmazza az ehhez szükséges parancsokat.



Ebből az adó-vevőből három darabot vásároltam, de mivel SMD változatról volt szó, nem tudtam közvetlenül a dugaszolós próbapanelra rákötni őket. Ezért terveztem és készítettem hozzájuk három adapter-nyákot, ezekre lettek ráforrasztva a modulok, és így már használni tudtam őket.

A cél itt is az volt, mint a joysticknél: megismerni az eszközt és programkönyvtárat írni hozzá. Rendelkezésemre állt a modul adatlapja és egy példaprogram, az adásra és a vételre vonatkozóan. Ez idő tájt még Pascalban programoztam, ezért a példaprogramban szereplő alapvető függvényeket a megértésük után C-ről át kellett fordítanom az általam használt nyelvre.

Miután elkészültem, tesztelés gyanánt összeállítottam egy adó és egy vevő állomásból álló próbakapcsolást: az adó másodpercenként 2 bájt hasznos adattal elküldte a vevőnek az aktuális hőmérsékletet. Szokáshoz híven nem működött rögtön elsőre az adatátvitel, de később szerencsére sikerült megjavítanom.

Az adás és a vétel menete:

- az egyes állomások felkonfigurálása adó vagy vételi üzemmódra,
- az adó oldalon egy függvénnyel egy tömb elemeinek az elküldése,
- a vételi oldalon a rádiós modul ellenőrzése – van-e vétel; ha igen, akkor a vett csomag bájtjainak kiolvasása a modulból.

A vételi puffer mérete nálam 8 bitre volt állítva, ezért ha érkezett egy bájt hasznos adat, azt még az előtt ki kellett olvasni, mielőtt egy újabb bájt érkezhett volna. Az egyszerűség kedvéért ezért fix számú bájtokból álltak a csomagok, amelyet az inicializáláskor kellett megadni.

Ahhoz, hogy a vevő modul szinkronizálódni tudjon az adóra, szükség volt 2 – 3 olyan bájttal kezdeni az adást, ami 1 és 0 váltakozásából állt. Ezután az adónak két speciális bájtot kellett küldenie – lényegében ez volt a cím –, ha a vevő ezeket vette, akkor a következő vett bájtok már belekerültek egyesével a vételi pufferbe. Miután az összes hasznos adat megérkezett, a FIFO puffer resetelésre került, innentől kezdve a vevő újra először a két cím bájtra várt.

A Holdjárom fejlesztésében most már ott járunk, hogy van egy készre szerelt alumínium alváz, forognak a kerekek, működik a joystick, össze van állítva a járgány és az adóállomás elektronikája, meg vannak írva a programok a mikrovezérlőkre. A rádiós adatátvitel során csak pár hasznos bájt + ellenőrző összeg van átküldve a távirányítás miatt, ezek is csak akkor, ha nem alaphelyzetben van a botkormány – vagyis zero sebesség esetén nincs küldés.

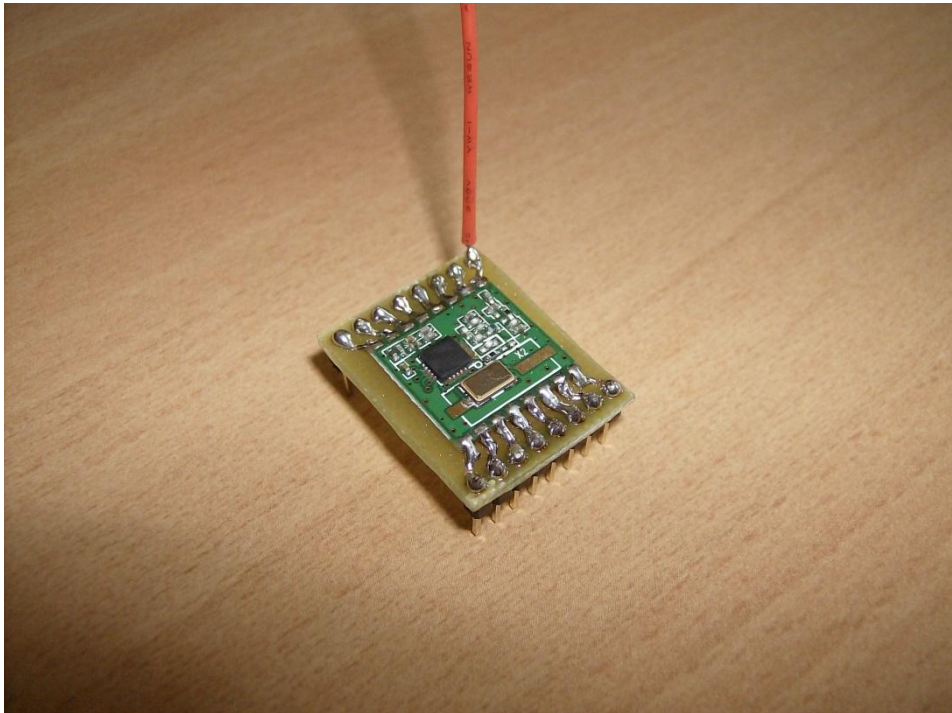
Elviekben a történet itt akár véget is érhetne, de sajnos egy nagyon kellemetlen probléma jelentkezett működés közben. Mivel kefések DC motorokat használtam, a motorok által keltett elektromágneses zavar működés közben lefagyasztotta a Holdjárom elektronikáját. Kívülről csak azt lehetett látni, hogy a mozgásban lévő járgány egyszer csak megáll, és nem reagál semmire. Többféle módszerrel próbálkoztam megoldani a problémát: A Holdjárom elektronikájának alapjául szolgáló mikrovezérlőt PIC18F4550-re cseréltem, de nem segített. Bekötöttem egy LED-et a kapcsolásba, melynek az volt a feladata, hogy világít, ha van az ellenőrző összeg alapján is megfelelő vétel. Ekkor feltűnt, hogy a hiba jelentkezéskor a LED már nem világít többé. Ebből nyilvánvalóvá vált számomra, hogy a gond a rádiós modullal lesz. Felcseréltem a Holdjárom és az adóállomás moduljait, de a hiba továbbra is fennállt.

A motorokat meghajtó H-hidas teljesítmény-fokozat vezérlése 4 biten keresztül történt párhuzamosan. Noha külön tápról járt a Holdjáró teljesítmény-fokozata és a mikrovezérlő elektronikája, mégis volt a két áramkör között galvanikus kapcsolat. A végleges megoldást e kapcsolat megszakításában láttam, ezért módosítottam a H-hidas kapcsoláson, és optocsatolókat építettem be (a motormeghajtóról bővebben a *Teljesítmény-fokozat* c. fejezetben). A probléma nem múlt el. Oszilloszkóppal vizsgálva a vezérlő elektronika tápját kb. 10 – 30 mV_{pp} nagyságú zaj volt megfigyelhető. Megpróbáltam többféle típusú és nagyságrendű kondenzátorral szűrni, de az se segített. Az volt a gyanúm, hogy a zaj nem a tápon keresztül jut be a rádiós modulba.

Felvettem a kapcsolatot e-mailben a modul gyártójával, és segítséget kértem. Rajzoltam egy vázlatot arról, hogy a Holdjárón belül mi hova van kötve, és csatoltam az egyik e-mailemhez. Ábrázoltam azt is, hogy nincs galvanikus kapcsolatban a motor és az adó-vevő modul. Pár nap után válaszoltak és afelől érdeklődtek, hogy milyen cégtől írok, illetve mit gyártok. Tudattam velük, hogy diák vagyok, és egy egyetemi projekthez használnám a rádiós moduljukat. Erre válasz bizony csak 1 – 2 héttel később érkezett, viszont az a fickó írt vissza, aki a modulhoz a példaprogramokat készítette, amelyeket én is anno alapul vettem. A tanácsai igazából nem segítettek, mivel azokban a táp szűrését és a külön tápról járatást szorgalmazta. Ekkor történt meg másodszor, hogy félretettem egy kicsit a Holdjáró projektet...

RFM23B

Mivel a Holdjáró gyengepontját az adó-vevő modulban láttam, következő lépésben ezt cseréltem le – beszereztem két RFM23B modult.



Ezekhez is készült adapter-nyák, amely segítségével a modul próbanyákhoz csatlakoztathatóvá vált. Ez a modul is hasonló paraméterekkel rendelkezik, mint az előző, de a csomagkezelése sokkal fejlettebb, és nagyobb az adóteljesítménye is. Ezen kívül tartalmaz három programozható GPIO lábat, ADC-t és digitális RSSI-t. Utóbbi (Received Signal Strength Indicator) azt jelenti, hogy a vevőből kiolvasható a vételi jel erőssége. A GPIO lábak több célt szolgálhatnak: egyrészt kivezethetjük rajtuk a státusz regiszter bizonyos bitjeit, másrészt használhatóak digitális kimenetként vagy bemenetként, harmadrészt pedig a beépített 8 bites ADC bemenetei is lehetnek. A modul hőmérséklet mérésére is alkalmas, ha az ADC-t a belső hőmérő-szenzorhoz rendeljük.

Mint már említettem, ez az adó-vevő lényegesen fejlettebb az előzőhöz képest. 64 bájtos TX és RX pufferral rendelkezik, így a vevő áramkörben lévő mikrovezérlőnek lényegesen egyszerűbb a dolga. Itt is be lehet állítani egy 4 – 4 bájt hosszú szinkron- és címszavat, ezáltal lehetőség adódik több vevőt használni egy frekvencián. Az adó modul automatikusan

összeállítja a kiküldésre szánt keretet, ami pár bájt hosszú, egyesek és nullák váltakozásából álló preambulummal kezdődik. Ezt követi a szinkron- és címszó, a csomag hossza, a hasznos adat, végül pedig a CRC ellenőrző összeg.

Preambulum	Szinkron- szó	Címszó (Header)	Csomag- hossz	Adat	CRC
1 – 512 bájt	1 – 4 bájt	0 – 4 bájt	0 vagy 1 bájt	1 – 64 bájt	0 vagy 2 bájt

Ehhez az adó-vevő modulhoz is írtam egy saját programkönyvtárat, amely szerencsére az első teszt során azonnal működött. Itt is makrókkal és definiálásokkal adunk meg bizonyos beállításokat, amelyeket a projekthez hozzáadott *my_defs.h* header fájlban kell elhelyeznünk:

```
#define rssi_auto_read    // RSSI érték autom. kiolvasás vétel után
#define burst_mode        // adat-bájtok egészben történő mozgatása

#define nirq               // rfm23b nIRQ láb
#define nsel               // rfm23b nSEL láb
#define sck                // rfm23b SCK láb
#define sdi                // rfm23b SDI láb
#define sdo                // rfm23b SDO láb

#define nirq_tris          // az adott láb iránybeállító
                           // regiszterének a megfelelő bitje

#define nsel_tris
#define sck_tris
#define sdi_tris
#define sdo_tris

#define delay1ms()         // pl. Delay1KTCYx(8);
#define delay10SCK()
#define gpio2              // autom. RSSI kiolvasásnál van szerepe
#define gpio2_tris
```

Lehetőségünk van hardveres vagy szoftveres SPI kommunikáció használatára. Alapesetben a szoftveres van alkalmazva, hardveres SPI-hoz definiálnunk kell az alábbi makrót:

```
#define hw_spi
```

A hardveres SPI előnye a nagyobb sebesség lenne, de használatakor nálam sajnos valami oknál fogva egy idő után bedöglött a kommunikáció. Internetes fórumokat olvasgatva utólag kiderült, hogy az általam használt PIC18F4550 mikrovezérlő SPI perifériája minden jel szerint bug-os, ezért én maradtam a jól bevált szoftveres változatnál.

Az adás és a vétel menete:

- egyes állomások felkonfigurálása, szinkron- és címszavak beállítása,
- üzemmód választás (TX, RX, Ready),
- adó oldalon küldéskor egy tömb kezdőcímének és az elküldendő bájtok számának a megadása, küldés elindítása,
- vételi oldalon az inicializálás után RX üzemmódba kapcsolás, vétel esetén a vett bájtok számának a kiolvasása a modul megfelelő regiszteréből, a csomag kiolvasása.

Most, hogy már rendelkezésemre állt az új rádiós modul egy működőképes programkönyvtárral, eljött az ideje annak, hogy beépítsem őket az adóállomásba és a Holdjáróba.

Élesztés után jött a hidegzuhany: pár másodperc működés után a Holdjáró állva maradt és nem reagált semmire sem. Tehát a hibajelenség megmaradt, én viszont kifogytam az ötletekből, és azt gondoltam, hogy ez sose fog rendesen működni.

Mikor újra elkezdtem foglalkozni a hibával, akkor először az adó-vevő modul belső regisztereit vizsgáltam meg. Ezen regiszterek segítségével lehet elvégezni a különböző átviteli, frekvencia, sebesség és egyéb beállításokat. Debuggolási célból módosítottam a Holdjáró programján, és így ezen túl sorosporton keresztül bármikor ki tudtam olvasatni egy megadott regiszter értékét. Ezután elindítottam a Holdjárót, majd megvártam, míg lefagy a rádiós kommunikáció. Néhány regiszter megvizsgálása után észrevettem, hogy azok értékei nem egyeznek meg azokkal, amelyeket inicializáláskor beléjük töltöttem. A rádiós modullal ezután már nem tudtam mit kezdeni, ezért más alkatrészt vettem górcső alá.

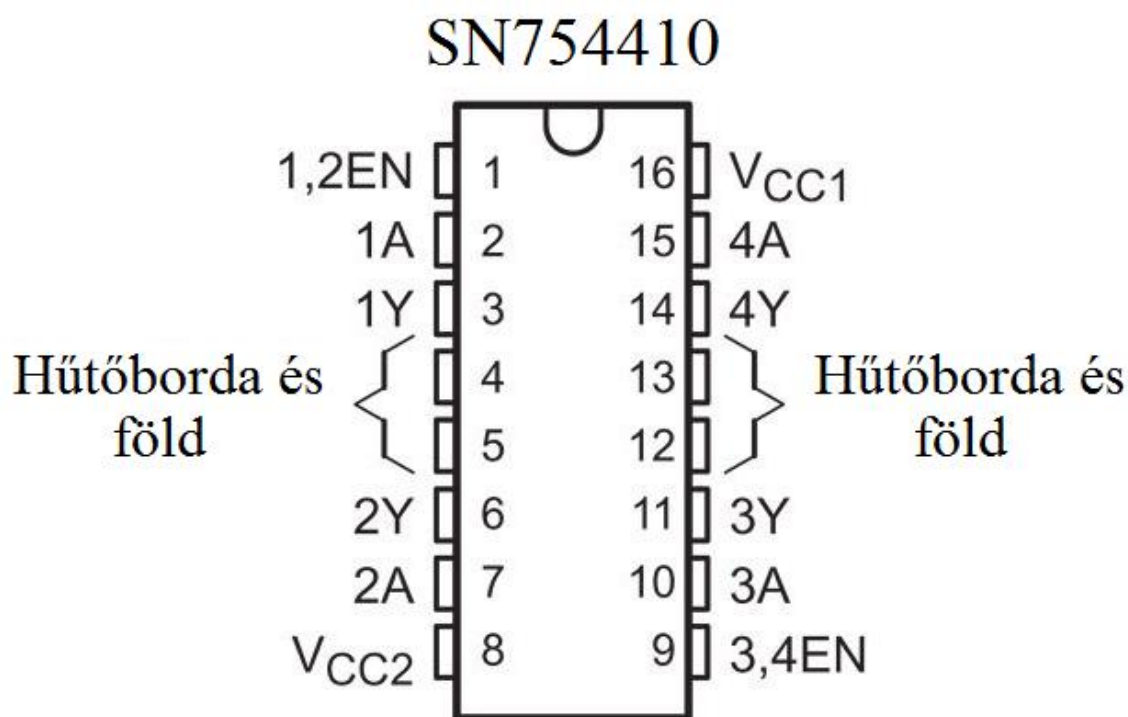
A mikrovezérlők és a rádiós modulok eltérő tápfeszültségről működnek, ezért az elejétől kezdve szükség volt szintillesztőkre az SPI buszon: $5V \rightarrow 3,3V$ irányban (rádiós modul felé) ellenállásos feszültség-osztókat alkalmaztam ($10k\Omega + 4,7k\Omega$). Visszafelé a jel közvetlenül ment a mikrovezérlőre, ugyanis a TTL jelszinteknél a logikai magas állapothoz elég a 2V feszültség is.

Próbaképp csökkenteni kezdtem ezen ellenállások értékeit, és azt vettem észre, hogy ritkábban marad abba a vétel. Mivel a végtelenségig nem csökkenthettem az ellenállásokat, ún. *bus transceiver* integrált áramköröket kezdtem el alkalmazni, és ezekkel oldottam meg a szintillesztést, hozzáteszem, mindkét irányban. A csere végül megoldotta a problémát, bár azóta egyszer már újra előfordult, hogy nem reagált az irányításra a Holdjáró. A konklúzió tehát az, hogy a motorok által keltett zaj az SPI buszra ráülve valószínűleg néha módosította az ott közlekedő bájtokat, és így a csomagolvasási utasítások bizony néha – néha regiszter-írásokká változhattak. Nyilván árnyékolt kábel használata csökkentené a motorok behatását a buszra.

TELJESÍTMÉNY-FOKOZAT

Ez az egyetlen áramkör a Holdjárón, ami házilag készített lemaratott nyákon van realizálva, és nem próbapanelon van összerakva. Ő hajtja meg közvetlenül a motorokat, ezért viszonylag nagy áramok leadására képes. A járgány hátoldalán kapott helyet.

A kapcsolás alapját az SN754410 integrált áramkör adja, amely 4 teljesítmény fél hidat tartalmaz. Két fél híd egy egész hidat alkot, amely két párhuzamosan kapcsolt motor energiaellátását valósítja meg. Ilyen párhuzamosan kapcsolt motor-pár van a Holdjáró jobb- és baloldalán.



Az IC igazságtáblája:

Bemenetek		Kimenet
A	EN	Y
H	H	H
L	H	L
x	L	Z

H – magas szint

L – alacsony szint

x – don't care

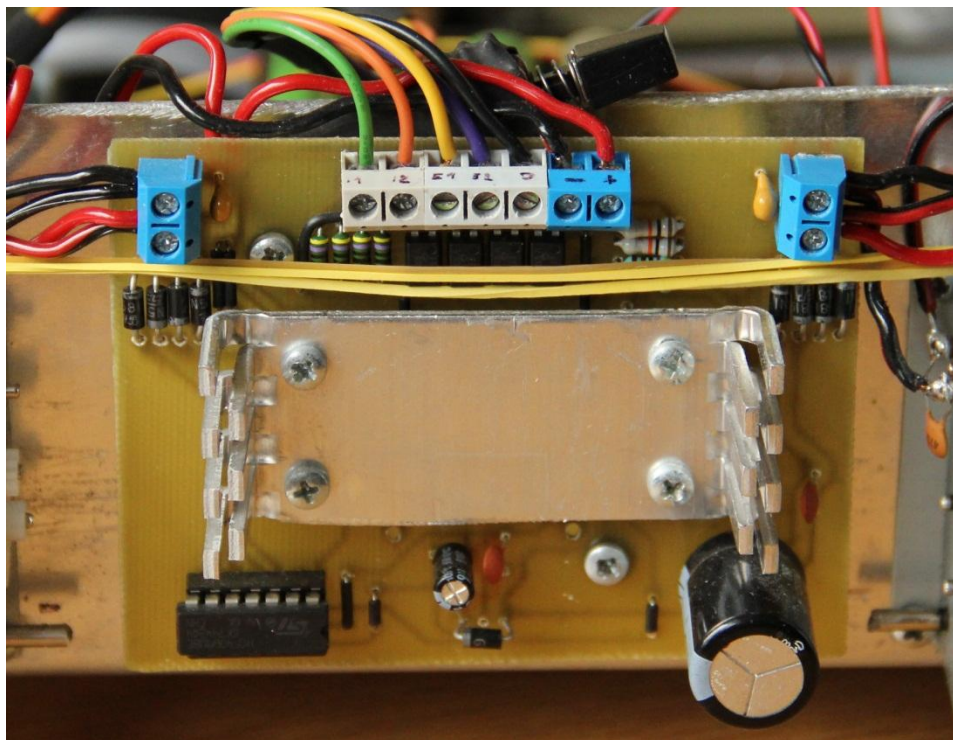
Z – nagy impedanciás állapot

Látható, hogy az egy egészt alkotó két fél híd vezérlése egy engedélyező lábbal és két bemenettel történik. Ezek alapján a két H-hidat legalább hat lábon keresztül lehetne csak vezérelni, de ennek ellenére megoldható az egész úgy négy lábbal is. Az én megoldásomban az egy egész hídhoz tartozó két bemenet felváltva kapja egy vezérlőjel ponált és invertált értékét, így az irányítás az alábbiak szerint valósul meg:

1. láb: a jobboldali motorok be- vagy kikapcsolása,
2. láb: a jobboldali motorok forgásiránya (előre vagy hátra),
3. láb: a baloldali motorok be- vagy kikapcsolása,
4. láb: a baloldali motorok forgásiránya (előre vagy hátra).

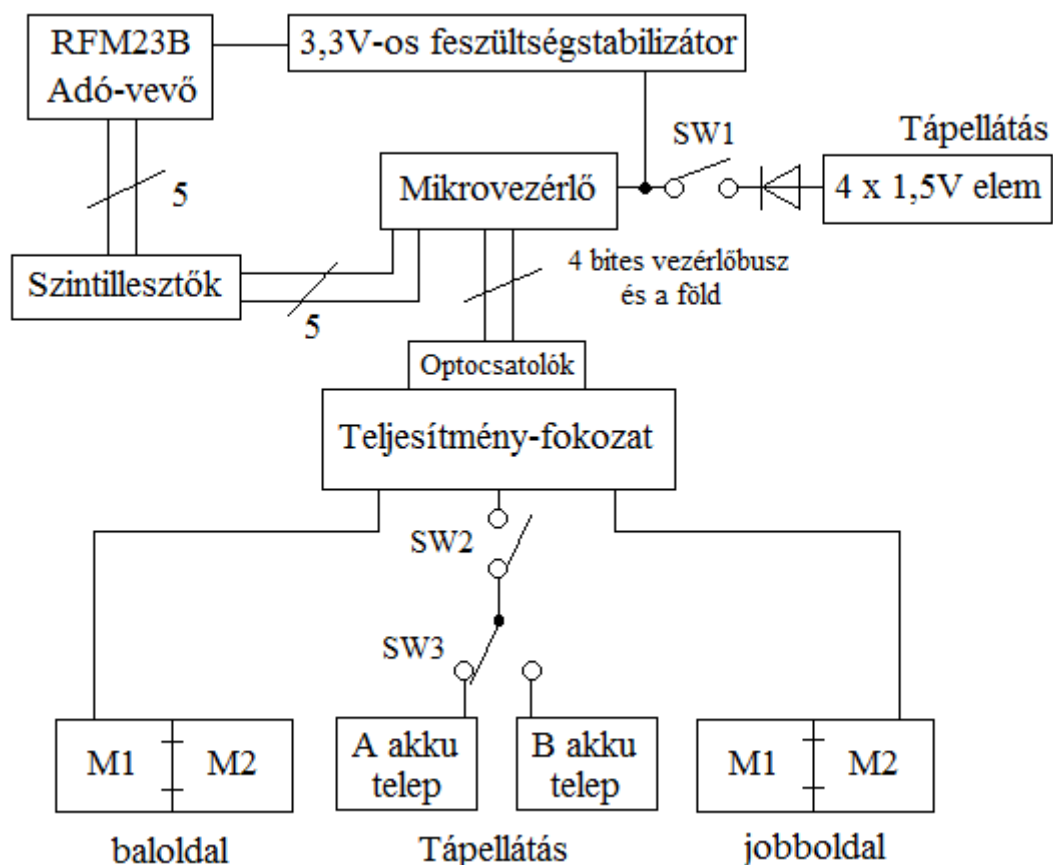
Az engedélyező lábakra kapcsolt impulzusszélesség modulált jellel tudja a mikrovezérlő a motorok sebességét szabályozni. A mikrovezérlő és a teljesítmény-fokozat közti galvanikus leválasztást optocsatolók valósítják meg.

Mivel a H-hidas integrált áramkör tokozása nem teszi lehetővé hűtőbordára való felszerelését, ezért a túlmelegedés ellen egy, az IC-hez hozzáerősített alumínium borda segítségével védekeznek.



A HOLDJÁRÓ

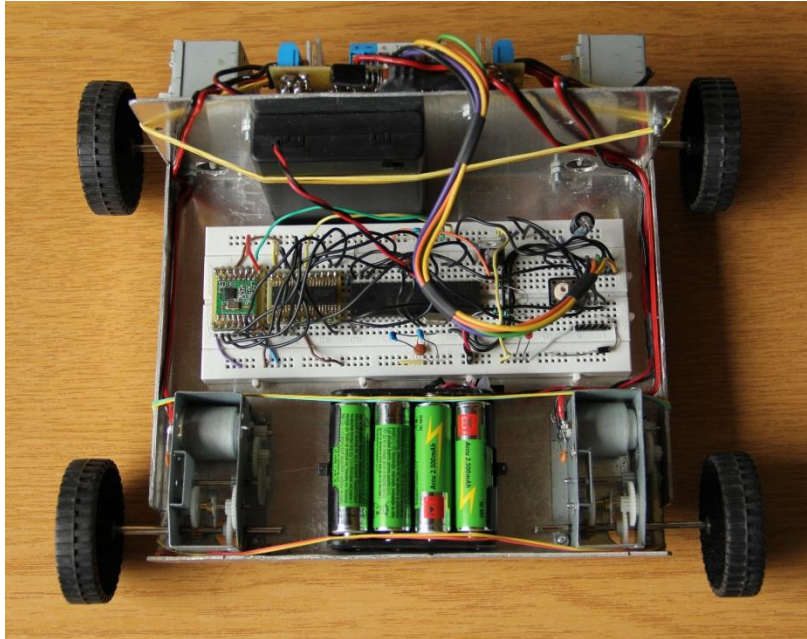
Ebben a fejezetben a Holdjáró felépítését és működését fogom kitárgyalni az előzőekben ismertetett részegységek segítségével.



A fenti ábrán látható a járgány sematikus rajza. Az egész kapcsolás alapját a 8 bites PIC18F4550 mikrovezérlő képezi, amely több feladatot is ellát. Bekapcsolás után inicializálja a rádiós modult vételi üzemmódra és a motorok sebességét zéróra állítja, ezáltal az egész járgány alapállapotba kerül. Az egy oldalon lévő motorok párhuzamosan vannak kapcsolva, majd ezek a teljesítmény-fokozat egy-egy egész hídjára csatlakoznak. A motorokon lévő parallel kerámia kondenzátorok a nagyfeszültségű tüskék ellen védenek. A teljesítmény-fokozat tápellátása az SW2 kapcsolón keresztül történik – vagyis ez az egyik főkapcsoló, az SW3 váltókapcsoló segítségével a telep-forrást választhatjuk ki. Az információátvitelt a mikrovezérlő és kettős H-híd között optocsatolók valósítják meg, amelyek galvanikus leválasztást is biztosítanak. A mikrovezérlő egy 5 bit széles buszon keresztül kommunikál a rádiós modullal, a kettőjük kapcsolata viszont nem közvetlen – szintillesztők valósítják meg a

különböző értékű tápfeszültségekből adódó jelszint különbségek transzformálását. Az SW1 kapcsoló a második főkapcsoló, rajta keresztül kap tápot a mikrovezérlő, a feszültségstabilizátor és a szintillesztő – lévén, hogy az is aktív eszköz. Az ábrán látható dióda a fordított tápfeszültségtől védi az elektronikát.

Mivel a Holdjáró még mindig fejlesztés alatt áll, ezért a vezérlő elektronikája is még csak dugaszolós próbapanelon van megvalósítva. A jövőbeli tervekről a későbbiekben lesz szó.



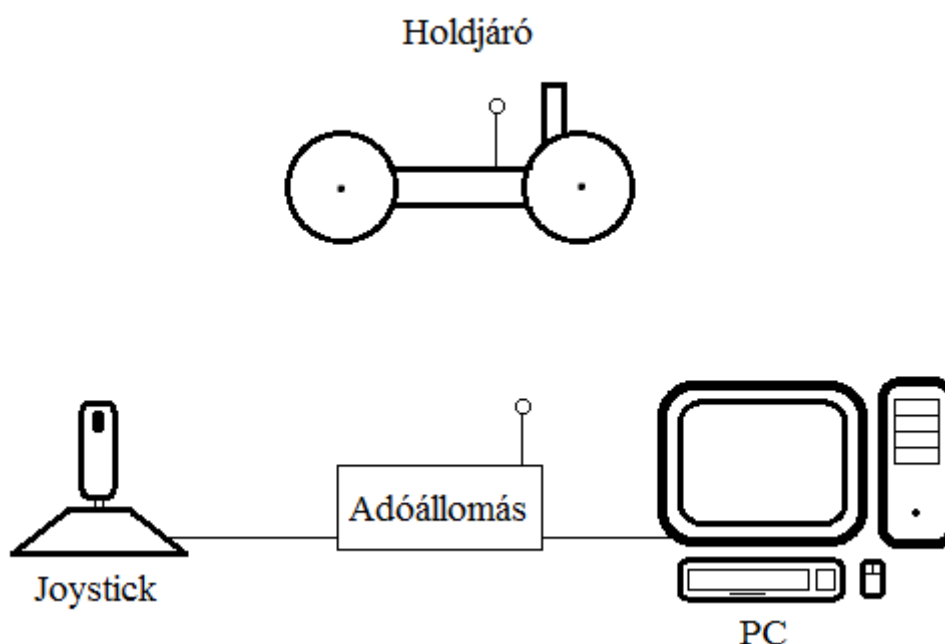
A járgány jelenleg csak manuálisan irányítható, az automata üzemmód még csak terv. A kézi irányítás tovább osztható joystick általi vagy kliens programos irányításra. Joystick használatkor az adóállomás másodpercenként kb. 100-szor leolvassa a botkormány és a gombok állapotát, majd ezekből az információkból 3 bájtos csomagokat kreálva átküldi rádióan a Holdjárónak a mozgásra vonatkozó paramétereket. Ezek a paraméterek az X- és Y-tengelyű sebességek, illetve a B1-es nyomógomb állapota. Ahhoz, hogy a Holdjáró ezekből az információkból megfelelően be tudja állítani a jobb- és baloldali motorok sebességét és irányát, írnom kellett egy saját függvényt, ami közvetlenül kezeli a kitöltési tényezőket illetve a forgási irányokat. Nélkülözhetetlen annak a tömbnek a megléte, amely tartalmazza az egyes sebességekhez tartozó kitöltési tényezőket:

```
const unsigned int speed[32] = {0, 265, 290, 315, 340, 365, 390, 415, 440, 465, 490, 515, 540, 565, 590, 615, 640, 665, 690, 715, 740, 765, 790, 815, 840, 865, 890, 915, 940, 962, 990, 1023};
```

Mivel a mikrovezérlő PWM (impulzusszélesség moduláció) perifériája 10 bites, ezért 1024 féle kitöltési tényezőt lehet rajta beállítani, melyek közül az 1023-as a 100%-os, a zero pedig a 0%-os kitöltés. A tömb indexelése nullával kezdődik, és az egyes indexek egy konkrét sebességet jelölnek. Ha a Holdjáró pl. egy 5-ös sebességű Y-tengelyű, és egy 0-ás sebességű X-tengelyű mozgásra való utasítást kap az adótól, akkor a motorok 35%-os ($365/1023$) kitöltési tényezővel lesznek meghajtva az egyik, vagy a másik irányban attól függően, hogy erre vonatkozólag mit tartalmazott a rádió vett csomag. A nyomógomb megnyomásakor a Holdjáró egy helyben történő forgást végez a botkormány eltolásának függvényében.

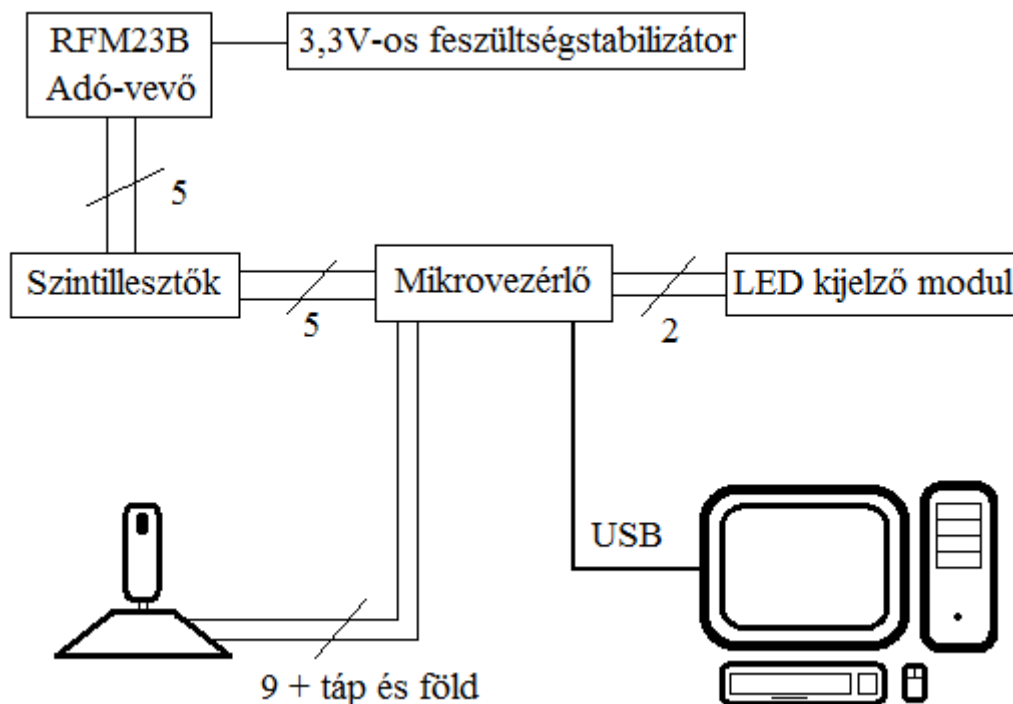
Az utolsó vétel után a járgány ha egy meghatározott ideig nem kap csomagot (kb. 100ms), akkor megállítja a motorokat. Ezzel a megoldással van megvalósítva az, hogy nem kell az adóállomásnak akkor is adnia, ha nincs mozgás.

A járgány számítógéppel is irányítható egy kliens program segítségével, amely C# nyelven íródott. Ebben a manuális módban a számítógép USB-n keresztül csatlakozik az adóállomáshoz. A két eszköz között Master – Master kommunikációs csatorna van megvalósítva, vagyis mindkét fél küldhet adatot a másiknak előzetes engedély nélkül. A számítógépes program közvetve valamilyen mozgásra irányuló utasítást adhat a Holdjárónak, amelyet a járgánynak egy bizonyos ideig végre kell hajtania, de lehetőség van menet közben e művelet megszakítására is.



AZ ADÓÁLLOMÁS

Az adóállomás feladata a távirányításban mutatkozik meg. Felépítését az alábbi ábra szemlélteti:



A mikrovezérlő típusa PIC18F2550, tápellátását az USB busz vagy egy külső 5V-os feszültségforrás látja el. A joystick és a számítógép megléte nem feltétlenül szükséges az áramkör működéséhez, de ezek hiányában nincs lehetőségünk a Holdjáró irányításához. A LED kijelző modul egy sajátfejlesztésű 6 számjegyes kijelző, mellyel két lábon keresztül sorosan kommunikál a mikrovezérlő. Csak a fejlesztés végéig marad az áramkörben.

Az adóállomás működésében két üzemmód létezik:

- joystick általi irányítás,
- kliens programmal történő irányítás.

Bekapcsolás után az alapértelmezett irányítási üzemmód a joystick-es irányítás. Ebben az esetben a mikrovezérlő folyamatosan leolvassa a joystick állapotát, és az eredményeket rádióan átküldi a Holdjárónak. Ha nincs csatlakoztatva a joystick, akkor nincs leolvasás, se rádiós

küldés. Az X- és Y-tengelyű sebességek az irányokat is beleértve a LED kijelzőn kerülnek megjelenítésre.

Ha csatlakoztatjuk az áramkört a számítógéphez, akkor lehetőségünk nyílik kliens programmal történő irányításra. A csatlakozás után, ha a Windows felismerte az eszközt és telepítette a megfelelő drivert, egy piros LED villogása jelzi a számunkra, hogy felépült a kapcsolat. Az USB kommunikációhoz az egyszerűség kedvéért az USB CDC osztályát használtam, mert így egyszerűbbé vált a kliens oldali programozás, ugyanis a csatlakoztatott eszköz így virtuális soros portként jelenik meg a számítógépen. A kommunikációban résztvevő bármelyik fél kezdeményezhet adat küldést, ehhez nincs szükség előzetes engedélyre.

A kliens program segítségével négy mozgástípus közül választva egy megadott ideig (1 – 6 másodperc) gurulásra tudjuk kényszeríteni a Holdjárót. Lehetőségünk van arra is, hogy az idő letelte előtt megszakítsuk e műveletet.

Mozgástípusok:

- előre gurulás,
- hátra gurulás,
- egy helyben történő jobbra fordulás,
- egy helyben történő balra fordulás.

A kliens használatakor lehet üzemmódot váltani, ha joystick általi vezérlésre kapcsolunk, akkor az X- és Y-tengelyű sebességek a programban ábrázolásra és kijelzésre kerülnek.

Az adóállomás és a kliens közti kommunikáció:

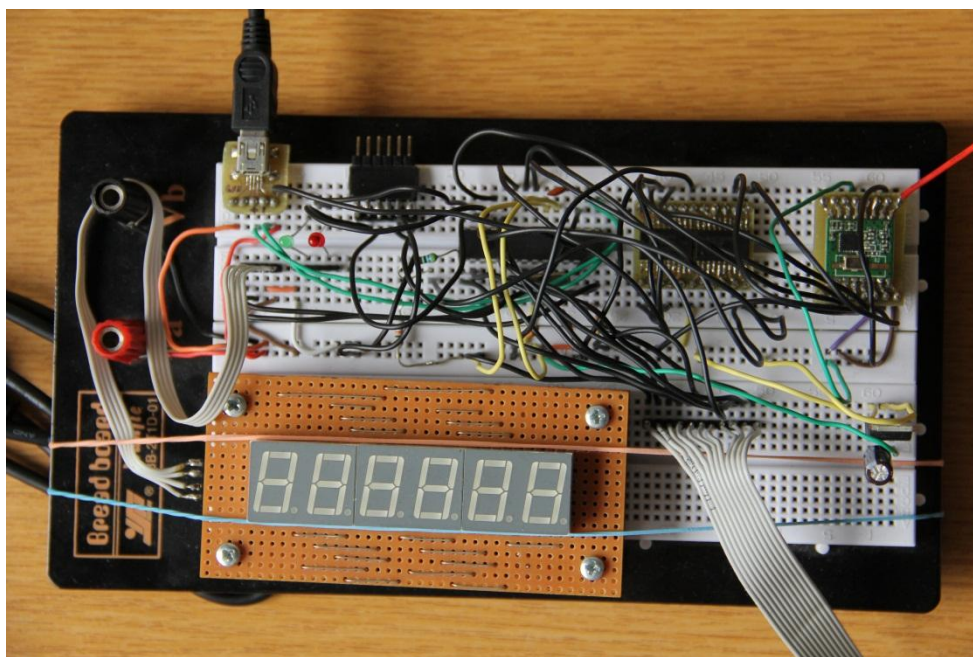
Irány	Név	Adat				
		1. bájt	2. bájt	3. bájt	4. bájt	5. bájt
PC → Adóállomás	MOVE	0x41	speed_x	speed_y	button	time
Adóállomás → PC	MOVE_ACK	0x42	-	-	-	-
PC → Adóállomás	STOP	0x43	-	-	-	-
Adóállomás → PC	STOP_ACK	0x44	-	-	-	-
PC → Adóállomás	CLIENT CONTROL	0x45	-	-	-	-
PC → Adóállomás	JOYSTICK CONTROL	0x46	-	-	-	-
Adóállomás → PC	JOYSTICK DISCONNECTED	0x47	-	-	-	-
Adóállomás → PC	JOYSTICK CONNECTED	0x48	-	-	-	-
Adóállomás → PC	JOYSTICK SPEED STATE	0x49	speed_x	speed_y	-	-
PC → Adóállomás	NEED JOYSTICK SPEED STATE	0x4A	-	-	-	-
PC → Adóállomás	STOP JOYSTICK SPEED STATE	0x4B	-	-	-	-

Az adóállomás felé irányuló utasítások közül a *MOVE* és a *STOP* parancs az, amely közvetlenül a Holdjárónak szól, a többi gyakorlatilag csak szolgálati közlemény az adóállomás felé. Az *_ACK* végződésű helyzetjelentés egyfajta nyugtázást jelent az utasítás vételére vonatkozóan.

Ha a kliens programban kiválasztjuk valamelyik irányítási módot, akkor erről a program értesíti az adóállomást, amely rögzíti az irányítás új forrását (kliens vagy joystick), és válaszul

visszaküldi a joystick csatlakozási állapotát. Ha joystick általi vezérlésre váltanánk, de a joystick nincs csatlakoztatva az adóállomáshoz, akkor az adóállomás kliens módban marad, és erről értesíti a számítógépes programot a *JOYSTICK DISCONNECTED* nevű helyzetjelentéssel. A kliens futása alatt, ha bármikor leválasztjuk a joysticket, az adóállomás jelez a programnak. A *JOYSTICK SPEED STATE* helyzetjelentés küldözgetése az adóállomás részéről arra szolgál, hogy a kliens a joystick-üzemmódban kiírni és ábrázolni tudja az X- és Y-tengelyű sebességeket. A *NEED JOYSTICK SPEED STATE* és a *STOP JOYSTICK SPEED STATE* utasítások az előző sebesség-helyzetjelentések kérvényezésére és megállítására szolgálnak.

Elindítás után és bezárás előtt a program, ha lehet, joystick-üzemmódba próbálja kapcsolni az adóállomást.

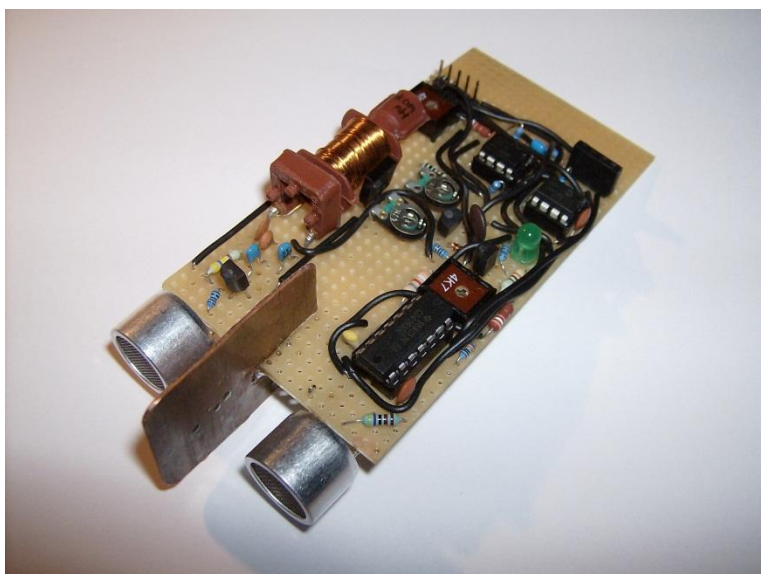


JÖVŐBELI TERVEK

Automata üzemmód

Ez a funkció idő hiányában sajnos még nem került megvalósításra, de mindenképp szeretném implementálni a Holdjáróban. Ezen üzemmód lényege abban áll, hogy a járgány véletlenszerű mozgásokat hajt végre a térben, eközben ultrahangos szenzorok segítségével próbálja az akadályokat elkerülni.

A kezdetek-kezdetén a Holdjáró projekt mellett kezdtem el fejleszteni egy ultrahangos távolságmérő modult is, amely tárgyak érzékelésére volt kitalálva 2,5 méteres távolságon belül. Ez az eszköz kb. egy hónap alatt készült el, és a felállított elvárásokat a tesztelés alatt sikeresen teljesítette. Hatótávolsága 5-től egészen 255 cm-ig terjed, bár két méter felett már csak keményebb tárgyakat tud érzékelni. Az eszköz a szó szoros értelmében véve is egy modul, tehát csak mérésre használható, nem tartalmaz semmiféle kijelzőt.

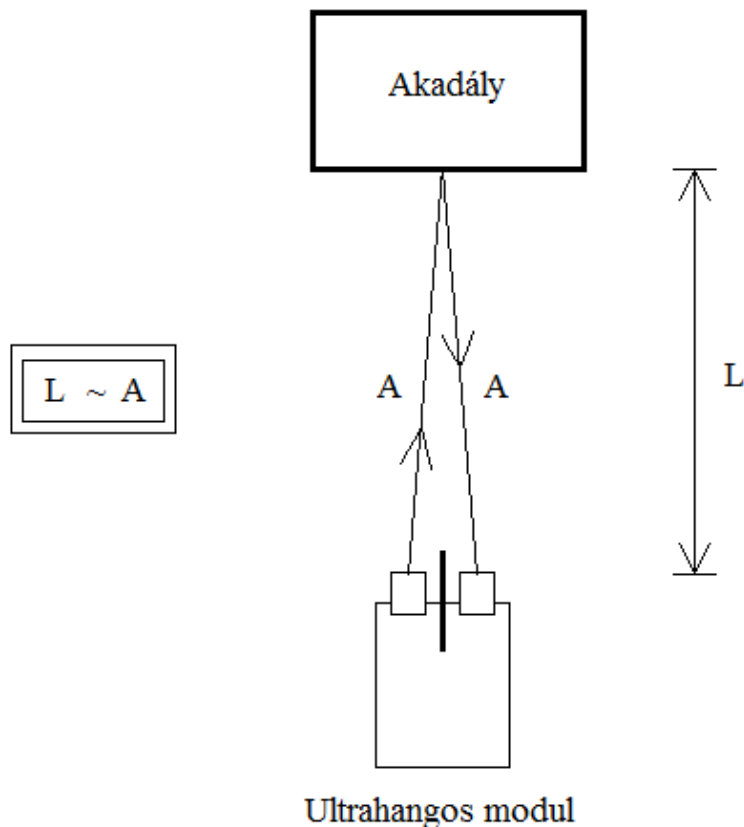


A modul logikai felépítése az alábbi:

- mikrovezérlő: PIC12F675,
- NAND kapukból felépített 40 kHz-es négyszögjel generátor az adó kapszulához,
- a belső számláló külső órajel generátora – kb. 17 kHz,
- vevő áramkör – erre csatlakozik a vevő kapszula; ez szűri, erősíti majd csúcseyenirányítja a vett jelet.

Két vezetéken keresztül, sorosan lehet kommunikálni a modullal 600 bps sebességgel. Az egyik vezeték a mérés elindítására szolgál, ez a modult használó mikrovezérlő egy kimenetére csatlakozik, a másik vezetéken a mért érték van visszaküldve.

A mérés elindítása után a modulban lévő mikrovezérlő beindít egy 40 kHz frekvenciájú négyszögjel generátort, amely néhány periódusidő erejéig meghajt egy ultrahang adó kapszulát. Ezután rögtön elindul egy 8 bites számláló a mikrovezérlőn belül (Timer0), és a modul elkezd várni a visszhang érkezését. Ha a visszhang nem érkezik meg az időzítő túlsordulásáig, akkor a modul azt feltételezi, hogy a potenciális tárgy méréshatáron túl van, ezért ezt jelezve a modult használó mikrovezérlőnek egy nulla értékű bájtot küld vissza. Ha időn belül érkezik meg a visszhang, akkor a számláló aktuális értéke kimentődik, majd végül ez az egész szám kerül visszaküldésre mérési eredményként. A számláló órajelét egy külső, NE555-tel felépített oszcillátor biztosítja. Ezen oszcillátor frekvenciája egy trimmer potenciométerrel szabályozható, értéke úgy van beállítva, hogy az órajel periódusideje megegyezzen azzal az idővel, ami alatt a hang levegőben megtesz 2 cm-t. Ezzel a trükkel megspóroljuk azt, hogy a mikrovezérlőben bonyolult, művelet igényes lebegőpontos számolásokat használjunk.



Ez a mérőmodul RC – oszcillátorok alkalmazása miatt picit pontatlan, főleg kis távolságok esetén, de lévén, hogy egy hobbi eszközről van szó, nekem tökéletesen megfelel. Feladatát a jövőben remélem megfelelően el fogja tudni látni.

CMOS kamera, mint a Holdjáró szeme

Szóba került egy olyan video rendszer kiépítésének az ötlete, amely lehetővé tenné egy, a Holdjáróra felszerelt kamera képének a továbbítását a számítógépes kliens program felé. Ennek igazából csak akkor lenne értelme, ha a video anyagot is vezeték nélkül lehetne továbbítani. Ehhez viszont nagyobb sávszélességre lenne szükség, mint amelyet a jelenleg használt rádiós modulok biztosítani tudnak. Egyelőre nem kizárt WiFi modulok használata sem, de egyelőre ezen új funkció megvalósítása még csak egy kósza ötlet.

VÉGSZÓ

Örülök, hogy belekezdtem a Holdjáró fejlesztésébe, úgy érzem ez idő alatt sok új ismeretet szereztem e témában. Nem mondom, hogy egy eddig ismeretlen témába vágtam volna bele, de azt hiszem, hogy ez a projekt a maga nemében mégis csak valamelyest egyedülálló. Látok még benne fantáziát, ezért azt hiszem, hogy tovább fogok még rajta dolgozni. Ez a munka valószínűleg végig fogja kísérni az egyetemen töltött tanulmányaimat is.

IRODALOMJEGYZÉK

Kép az RFM12B modulról: <http://dlmh9ip6v2uc.cloudfront.net/images/products/09582-01.jpg>
(letöltve: 2012. április 15.)

Kép az SN754410 integrált áramkör lábkiosztásáról:
<http://erelement.com/images/motors/sn754410-sheet.jpg>
(letöltve: 2012. április 16.)