

## OTDK – dolgozat

Méhes Jozef

Villamosmérnöki BSc. – 4. évfolyam

Széchenyi István Egyetem

Témavezető: Dr. Hidvégi Timót, egyetemi docens

2012

## Az intelligens Holdjáró

# 1. Tartalomjegyzék

|                                                     |    |
|-----------------------------------------------------|----|
| 2. KIVONAT                                          | 4  |
| 3. BEVEZETŐ                                         | 5  |
| 4. AZ ALVÁZ                                         | 6  |
| 5. JOYSTICK                                         | 8  |
| 6. RF ADÓ-VEVŐ MODUL                                | 12 |
| 6.1. RFM12B                                         | 12 |
| 6.1.1. Probléma a modullal                          | 13 |
| 6.2. RFM23B                                         | 15 |
| 7. TELJESÍTMÉNY-FOKOZAT                             | 18 |
| 7.1. Első és második változat                       | 18 |
| 7.2 Harmadik változat                               | 22 |
| 7.2.1. Ok a leszerelésre                            | 23 |
| 8. A HOLDJÁRÓ                                       | 24 |
| 8.1. Vezérlőelektronika                             | 24 |
| 8.2. Hibakeresés                                    | 26 |
| 8.2.1. SPI busz                                     | 27 |
| 8.2.2. SPI busz szűrése                             | 28 |
| 8.3. PWM                                            | 29 |
| 9. AZ IRÁNYÍTÁS                                     | 30 |
| 9.1. Botkormány                                     | 30 |
| 9.2. Adóállomás                                     | 32 |
| 9.3 Számítógépes program                            | 33 |
| 9.3.1. Az adóállomás és a kliens közti kommunikáció | 34 |
| 9.4. URC                                            | 36 |
| 9.4.1. Mikrovezérlő                                 | 38 |
| 9.4.2. RF modul                                     | 39 |
| 9.4.3. Kijelző                                      | 39 |
| 9.4.4. Kommunikáció a kontrollerrel                 | 40 |
| 9.4.5. Menü                                         | 41 |
| 9.4.6. Telemetry                                    | 43 |
| 9.5. Ultrahangos távolságmérő                       | 44 |
| 9.5.1. Automata üzemmód                             | 47 |
| 10. CMOS KAMERA, MINT A HOLDJÁRÓ SZEME              | 48 |
| 11. VÉGSZÓ                                          | 51 |
| 12. HIVATKOZÁSOK                                    | 52 |

## 2. Kivonat

A dolgozat egy mérnöki megoldásokat felvonultató rendszer tervezési és megvalósítási menetét mutatja be. A fő egység egy olyan négykerekű járgány, amely többféle irányítási móddal rendelkezik. Ismertetésre kerülnek a Holdjáró kezdeti formái, ezt követően pedig a jelenlegi felépítés. A járgány vezeték nélkül irányítható, illetve a parancsokat is ily módon kapja. Ezekért egy adóállomás és egy univerzális távirányító felel. Az adóállomás egy olyan áramkör, amelyre egy átalakított joystick kapcsolódik - ez az egyik módja a kézi irányításnak. Ez az áramkör USB-n keresztül számítógépre is csatlakoztatható, ezáltal biztosítja a parancsokkal történő irányítást, amelyeket egy saját fejlesztésű számítógépes programmal lehet kiadni. Az univerzális távirányító egy teljesen mobil egység, ami tulajdonképpen egy átalakított videojáték kontroller. Kijelzővel és saját áramforrással bír. Két miniatűr analóg botkormány teszi lehetővé a járgány tankszerű vezetését – ez a másik mód a kézi irányításra. Ez az eszköz a jövőben további saját készítésű modellek és robotok távirányításához is felhasználható majd. A kézi- és a parancsokkal történő irányításon túl a Holdjáró egy intelligens, automata üzemmóddal is rendelkezik. Ekkor a járgány egy saját fejlesztésű ultrahangos távolságmérő segítségével önmagát tudja irányítani úgy, hogy többé-kevésbé elkerüli az előtte álló akadályokat. A jövőbeli tervek közé tartozik egy vezeték nélküli kamerás rendszer is. Sajnos a motorok által keltett zaj jelenleg még zavarja a vezérlőelektronikát, ezért ez még javításra szorul.

### 3. Bevezető

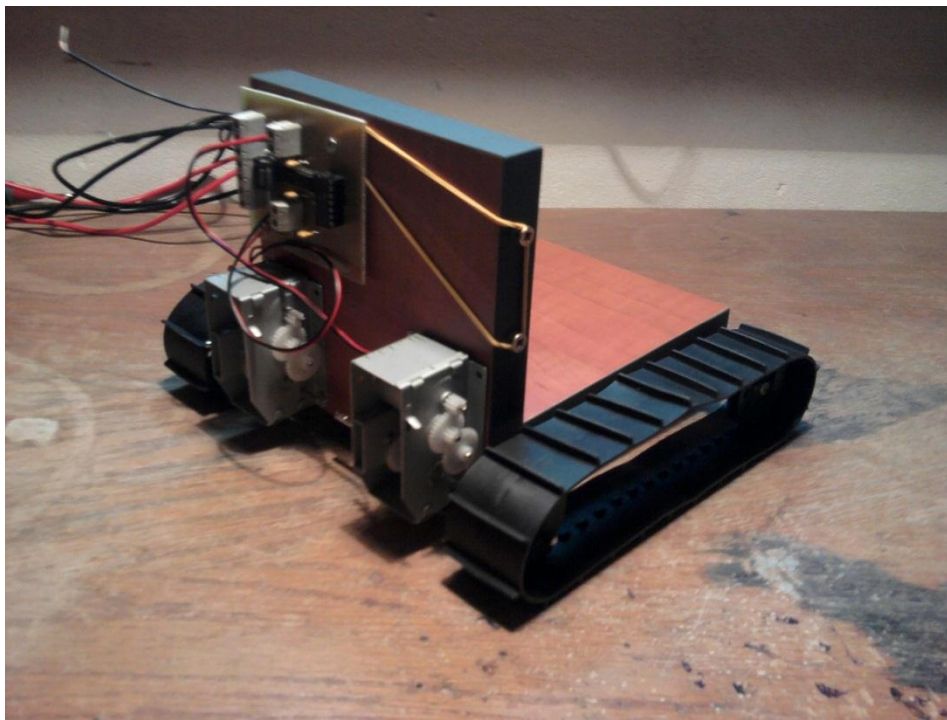
Mindig is érdekelt a modellezés és a műszaki megoldások gyakorlatba történő átültetése. Miután megkezdtem a tanulmányaimat a Széchenyi István Egyetem Villamosmérnöki BSc szakán, egy olyan témát kerestem, amelyben próbára tehetem az idővel gyarapodó szakmai tudásom, és azzal foglalkozhatom, ami ebben a szakmában a legjobban érdekel. Biztos voltam benne, hogy nem fog minden elsőre működni, de a kezdeti sikerélmények után már nem sajnáltam az időt és a pénzt, amit erre a projektre fordítottam.

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 és C32-es fordítóra, ugyanis a fejlesztők többsége ezt használja, és itt az egyetemen is ezt tanítják.

Miután megismerkedtem a 8 bites architektúra képességeivel, találtam egy olyan témát, amely többféle villamosmérnöki problémakört felölel, és amely nem túl nagy feladat egy mérnökhallgató számára. A nagy ötlet egy olyan elektromos meghajtású miniatűr jármű lett, amelyet vezetékek nélkül lehet irányítani, tud tájékozódni a térben, és autonóm módon önmagát is tudja „vezetni”. Nem egy hétköznapi autót akartam kicsiben megépíteni. A cél egy egyedi konstrukció kialakítása volt, de a hangsúly nem a gépészeti trükkökre és megoldásokra, hanem a vezérlésre és a kommunikációra irányult.

## 4. Az alváz

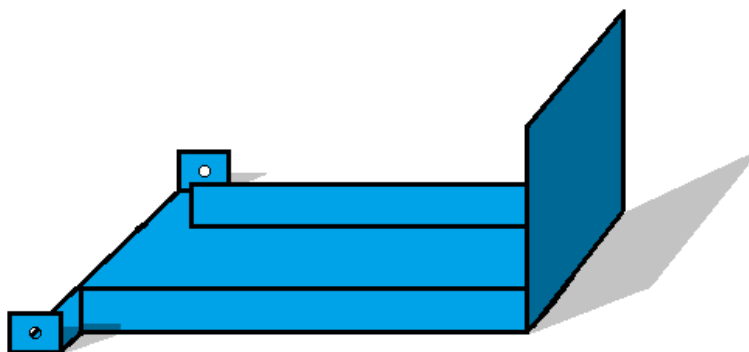
Az alváz az egyedüli elem, ami a legnagyobb vizuális fejlődésen ment keresztül. A kezdetek kezdetén az volt a koncepció, hogy a járgány rossz terepviszonyok esetén is tudjon közlekedni: sűrű szőnyeg, küszöb, esetleg kavicsos talaj. A megoldást kerekek és hernyótalpak (modellüzletben árusított gumi lánctalp) használatában láttam. Egyértelmű volt tehát, hogy a kerekek fixen lesznek rögzítve az alvázhoz, vagyis egyik kerék se fog télni. Törekedtem az egyszerűsége, ezért az alváz fából készült, nagyjából 20 x 20 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 vezérlőelektronika dugaszolós próbanyákon volt összeépítve, és saját feszültségforrással rendelkezett. A mikrovezérlő programja bug-os volt: néha lefagyott, de az is előfordult, hogy a motorok által keltett zaj hatására újraindult. Az irányítás itt még nem volt vezeték nélküli. Használt, régi gameport-os joystick tette lehetővé a vezérlést. A joystick analóg vezérlőjele ekkor még kis felbontással volt feldolgozva, ezért nem volt elég folytonos a járgány gyorsulása, ha megmozdítottuk a botkormányt. Azóta ez változott, bővebben a *Joystick* c. fejezetben.



1. ábra. A Holdjáró legelső változata

Hamar rájöttem, 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 problémát annak tulajdonítottam, 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 a relatíve sok vas miatt túl nehézre sikeredett, és így végeredményben a mozgás dinamikája semmit se javult. Ezután egy időre pihentettem a projektet, mással foglalkoztam.



2. ábra. A második generációs alváz

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

A jelenlegi 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.

## 5. 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 (3. ábra), szintúgy gameport-os darabot.



3. ábra. A jelenleg is használt joystick

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. A számítógépen belül ez a változó ellenállás egy ellenállás – frekvencia jelátalakítóra kerül (pl. NE555-tel felépített oszcillátor). A nyomógombok tűi a gameport-csatlakozóban alapállapotban lebegnek, vagy magas szinten vannak (típusfüggő), megnyomásuk esetén alacsony szintre váltanak [1].

Hogy egyszerűsítsem a joystick kezelését, bizonyos módosításokat hajtottam rajta végre . 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 $\Omega$  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 erősítő feszültségkövető áramkörökre. A biztonság kedvéért azért az ADC mintavételezési idejét a maximumra állítottam.

A joystick kezeléséhez saját programkönyvtárat írtam, melyben makrókkal és definiálásokkal végezhetjük 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      // pl. 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        // pl. 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        // pl. 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
```

```

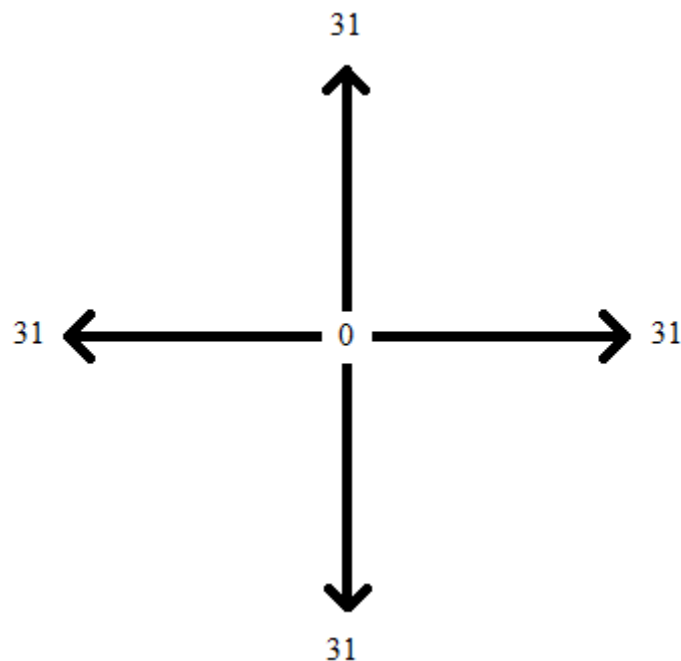
#define use_button3      // B3 nyomógomb használata
#define button3          // B3 nyomógomb olvasó reg. bitje
#define button3_tris     // B3 nyomógomb irányát beáll. reg. bitje

#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.



4. ábra. Bedőlési intervallumok

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 ahogy a 4. ábrán is

látható, ez ma már 0-tól 31-ig tart, ahol a 0 értelemszerűen a zero 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. A 31 definiált sebesség egészen folytonossá teszi a gyorsítást, szinte nem is lehet érzékelni, hogy diszkrét értékekkel dolgozunk.

Mivel a botkormány mechanikája már kissé elhasználódott, ezért nulla állapotban mind két tengelyen megfigyelhető egy kis holtjáték. Ezen kívül a joystick teljes bedőlésénél a kimeneti analóg jel sohasem éri az ADC szélső kimeneti értékeit (0 – 1023). Ezek a pontatlanságok szoftveres úton vannak korrigálva.

Ezekkel a konstans számokkal állíthatóak be az X- és Y-tengelyek intervallumhatárai:

```
const unsigned int joystick_x_intervals[4] = {947, 555, 438, 52};  
const unsigned int joystick_y_intervals[4] = {995, 596, 480, 112};
```

A 2. és a 3. szám között helyezkedik el a zero sebesség, az 1. és a 4. szám pedig a joystick végállást adja meg.

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

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.

## 6. RF Adó-Vevő modul

### 6.1. RFM12B

Ahogy az előzőekben említettem, az irányítás kezdetben nem volt vezeték nélküli. Miután elkészült végre a végleges alváz, elkezdtem keresni olyan rádiós modult, amely megfelelt nekem ehhez a projekthez. Mindenképp olyan típust szerettem volna választani, 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 utasításokat [2].



5. ábra. RFM12B [3]

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 már rutinosan használt nyelvre.

Miután elkészültem, tesztelés gyanánt összeállítottam egy adó- és egy vevőállomásból álló tesztkörnyezetet: 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 a következő volt:

- az egyes állomások felkonfigurálása adó/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 hőmérséklet kijelzése

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.

### **6.1.1. Probléma a modullal**

A Holdjáró 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 kefék DC motorokat használtam, a motorok által keltett elektromágneses zavar működés közben lefagyasztotta a Holdjáró elektronikáját. Kívülről

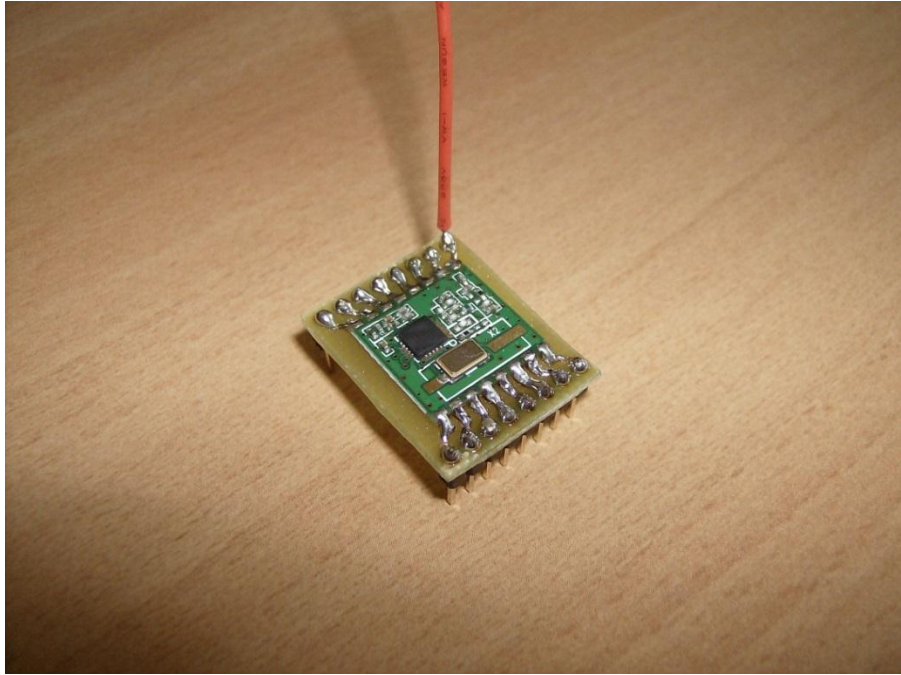
csak azt lehetett látni, hogy a mozgásban lévő járgány egyszer csak megáll, és nem reagál semmire. Többféle módszerrel próbálkoztam megoldani a problémát: A Holdjáró elektronikájának alapjául szolgáló mikrovezérlőt PIC18F4550-re cseréltem, de nem segített. Bekötöttem egy LED-et a kapcsolásba, melynek 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 van. Felcseréltem a Holdjáró 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<sub>pp</sub> 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 a felől érdeklődtek, hogy milyen cégtől írok, illetve mit gyártok. Tudattam velük, hogy diák vagyok, és egy egyetemi projekthez használom a rádiós moduljukat. Erre válasz bizony csak 1 – 2 héttel később érkezett, viszont az a férfi 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 ötlet híján félretettem egy kicsit a Holdjáró projektet...

## 6.2. RFM23B

Mivel a Holdjáró gyengepontját az adó-vevő modulban láttam, következő lépésben ezt cseréltem le – beszereztem két RFM23B modult (6. ábra).



6. ábra. Az új adó-vevő

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 [4].

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-<br>szó | Címszó<br>(Header) | Csomag-<br>hossz | Adat        | CRC              |
|--------------|------------------|--------------------|------------------|-------------|------------------|
| 1 – 512 bájt | 1 – 4 bájt       | 0 – 4 bájt         | 0 vagy 1<br>bájt | 1 – 64 bájt | 0 vagy<br>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 használatkor nálam sajnos valami oknál fogva egy idő után megállt 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 másodperces működés után a Holdjáró állva maradt és nem reagált semmire sem. Tehát a hibajelenség megmaradt, én viszont kifogytam az ötletekből.

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 olvasni 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, ezért ún. *bus transceiver* integrált áramköröket kezdtem el alkalmazni, és ezekkel oldottam meg a szintillesztést, hozzáteszem, mindkét irányban. A csere látszólag megoldotta a problémát, bár azóta egyszer már újra előfordult, hogy nem reagált az irányításra a Holdjáró.

A konklúzió tehát az volt, hogy a motorok által keltett zaj az SPI buszra ráülve valószínűleg néha módosította az ott közlekedő biteket, és így a csomagolvasási utasítások bizony néha – néha regiszter-írásokká változhattak. Nyilván árnyékolt kábel használata csökkentené a motorok behatását a buszra.

## 7. Teljesítmény-fokozat

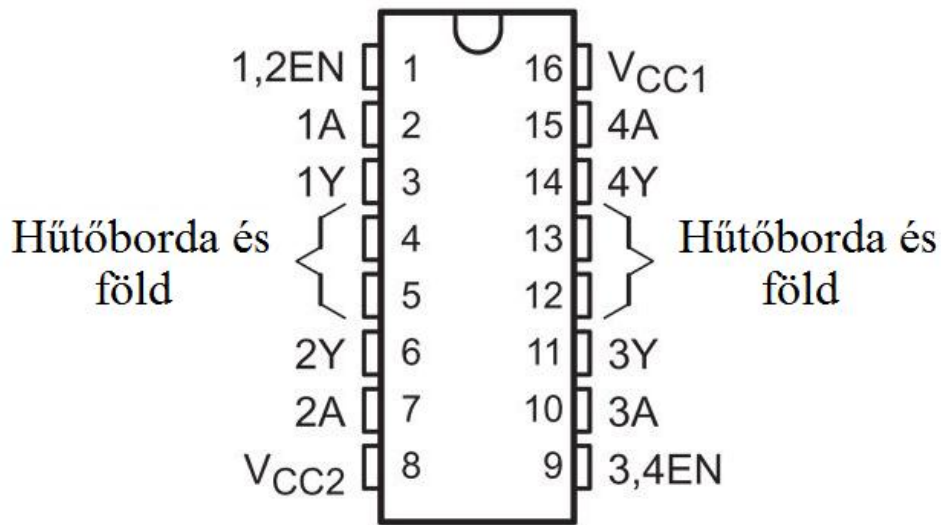
Ez az áramkör hajtja meg a motorokat a Holdjárón, ezért viszonylag nagy áramok leadására képes. A járgány hátoldalán kapott helyet. A táplálása 4 sorba kapcsolt 1,2V-os NiMH akku. Eddig összesen három verzió készült el belőle, jelenleg a második van fent a gépen.

Az első és a második változat csupán abban különbözik, hogy a másodikban optocsatolókat alkalmaztam galvanikus leválasztás céljából. Ezzel a kommunikációs probléma megjelenésekor próbálkoztam, sikertelenül.

### 7.1. Első és második változat

A kapcsolás alapját az SN754410 integrált áramkör adja, amely 4 teljesítmény fél hidat tartalmaz (7. ábra). 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.

## SN754410



7. ábra. Kettős H-híd [5]

Az IC igazságtáblája:

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

H – magas szint

L – alacsony szint

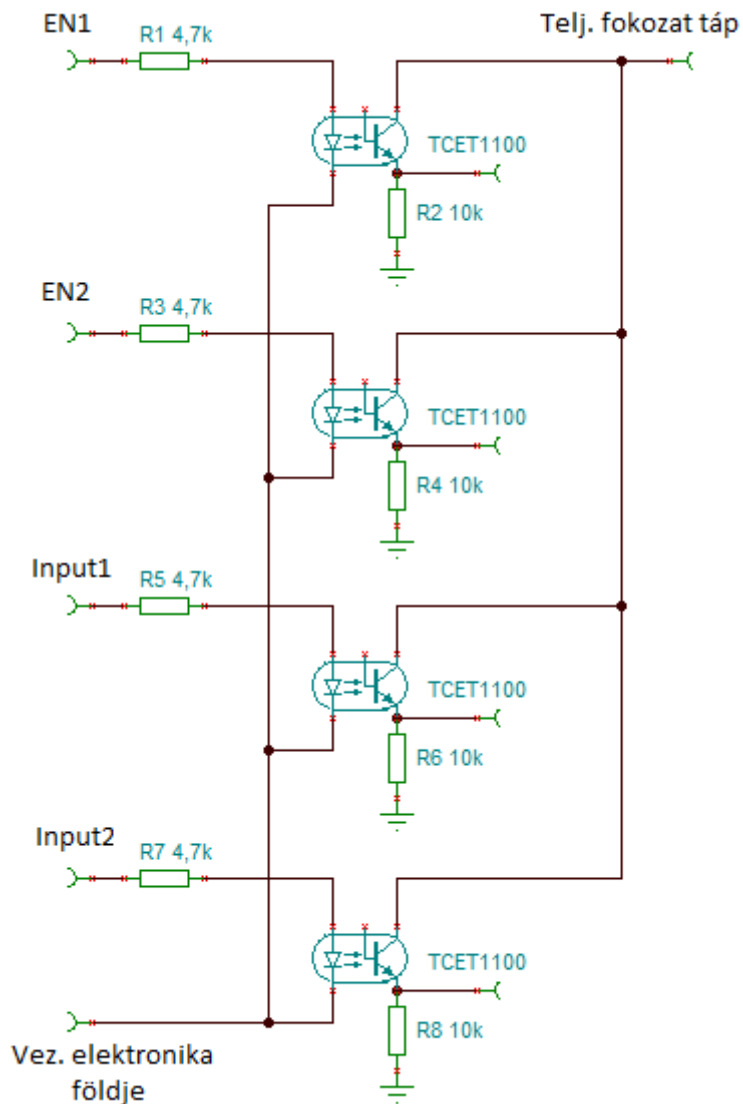
x – don't care

Z – nagy impedanciás állapot

Látható, hogy a teljes hidat 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 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:

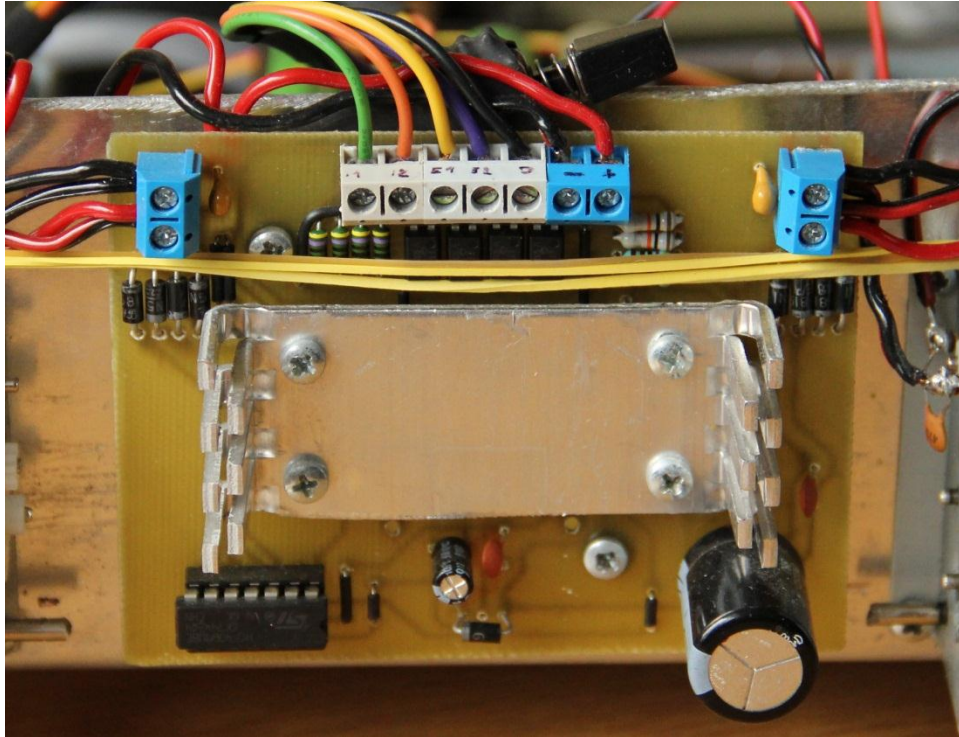
1. láb: A jobboldali motorok be- vagy kikapcsolása,
2. láb: A jobboldali motorok forgásiránya (előre vagy hátra),
3. láb: A baloldali motorok be- vagy kikapcsolása,
4. láb: A baloldali motorok forgásiránya (előre vagy hátra).

Az engedélyező lábakra kapcsolt impulzusszélesség modulált jellel tudja a mikrovezérlő a motorok sebességét szabályozni. A mikrovezérlő és a teljesítmény-fokozat közti galvanikus leválasztást az áramkör második változatában optocsatolók valósítják meg (8. ábra).



8. ábra. Optocsatolók elrendezése

Mivel a H-hidas integrált áramkör DIP tokozása nem teszi lehetővé az IC hűtőbordára való felszerelését, ezért a túlmelegedés ellen egy, az integrált áramkörhöz hozzászorított alumínium borda segítségével védekezek (9. ábra).

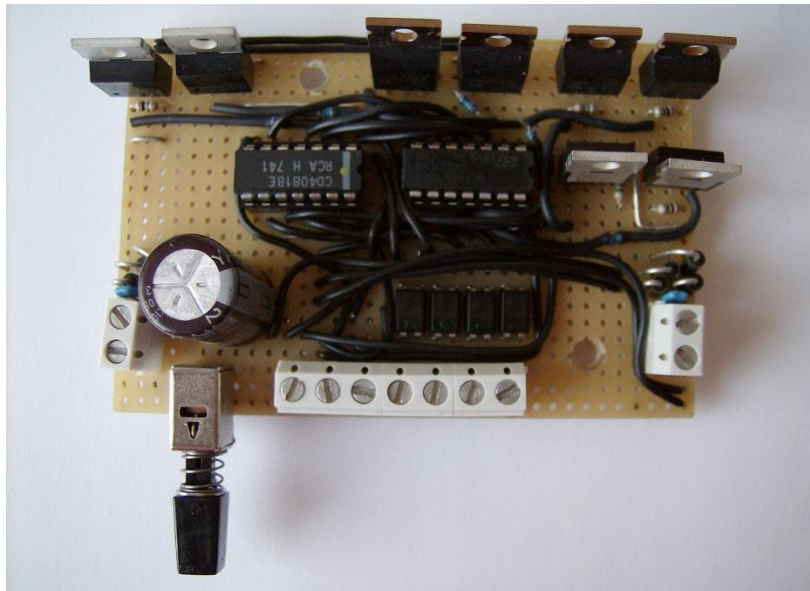


9. ábra. A teljesítmény-fokozat második változata

A teljesítmény-fokozat harmadik változatának megépítését az indokolta, hogy csökkentsem a hődisszipációt és növeljem a hatásfokot. Az IC-ben bipoláris tranzisztorokkal vannak kialakítva a H-hidak, ezért teljes nyitáskor is kb. 0,6V maradékfeszültség esik a nyitott tranzisztor CE átmenetén. Mivel nyitáskor a motorok árama egyszerre két fél híd ágon folyik keresztül, ezért egy hídon összesen kb. 1,2V feszültség esik, ami a 4,8V-os tápfeszültség mellett elég soknak számít. Az így kárba vesztett energiának a valóságban durván a kétszeresét kell venni, mivel menet közben általában a Holdjáró mindkét oldalán lévő motorok működnek.



diódák a FET-eket védik a motorok önindukciós feszültségeitől. Ahogy a 11. ábra is mutatja, ez az áramkör időhiány miatt csak csupalyuk próbapanelon lett összeforrasztva.



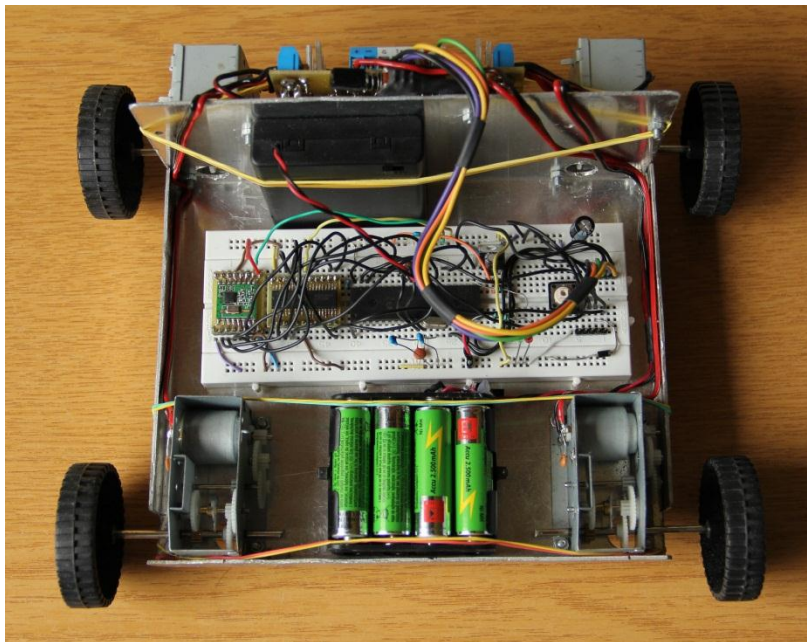
11. ábra. A harmadik teljesítmény-fokozat realizációja

### 7.2.1. Ok a leszerelésre

Az előzőekben már említettem, hogy bár jobb hatásfokkal működik a MOSFET-es áramkör, ennek ellenére egyelőre még sincs használatban. Ennek az az oka, hogy használatával a motorok magasabb csúcsfeszültségről járnak (amivel elviekben nem is lenne gond), ez viszont kihat a vezérlőelektronikára. Megépítése után ugyanazok a „tünetek” jelentkeztek, mint amelyek a 6.1.1. *Probléma a modullal* c. alfejezetben kerültek bemutatásra. Ez azt jelentette, hogy a motor – RF modul kapcsolatát még mindig nem sikerült a megfelelő módon rendbe hoznom. A megoldást a vezérlőelektronika árnyékolásában láttam, ehhez viszont először szükséges volt összeépíteni az áramkört egy nyáklemezen, majd ezután ezt egy fémdobozba szerelve a Holdjáróra felszerelni.



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. A nyomógomb a Bootloaderhez szükséges, mivel a mikrovezérlő szoftvere USB-n keresztül frissíthető, ezáltal nincs szükség PICKit 2 programozóra.

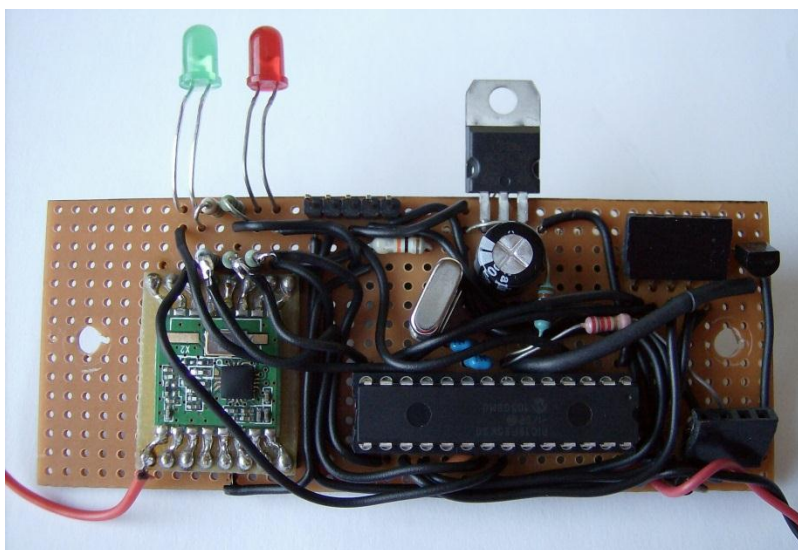


13. ábra. Holdjáró

A 13. ábrán látható a Holdjáró, ebben az állapotában a vezérlőelektronika még dugaszolós próbanyákon volt összeállítva. Mikor a legújabb teljesítmény-fokozat hatására bizonytalanná vált az áramkör működése, megépítettem egy módosított változatát alumínium dobozba szerelve. Az új verzió 3 dologban tért el a régitől:

1. Csupalyuk próbanyákon forrasztottam össze a vezérlőelektronikát, ezzel elejét vettem a kontakthibákból, silány illesztésekből származó üzemzavaroknak.
2. Mivel a nyák földelt fémdobozba került, ezért megakadályoztam a motorok káros hatását a mikrovezérlőre és a rádiós modulra.
3. A mikrovezérlőt egy olyan típusra cseréltem, amely 3,3V tápfeszültségről működik, ezáltal szükségtelenné vált a *bus transceiver* integrált áramkörök használata.

A Microchip termékválasztója segítségével találtam rá a PIC18F25K20 mikrovezérlőre. Ez a kontroller maximum 64MHz-es órajelen tud járni, és a többi 8 bites vezérlőhöz hasonlóan 4 órajel alatt hajt végre egy utasítást. Rendelkezik ezen felül még négy Timer perifériával, két CCP modullal, hardveres SPI interfésszel és egy ADC-vel. Mivel a fémdobozban relatíve kevés hely volt, ezért szükséges volt a lehető legkisebb lábszámú kontroller mellett dönteni. Az általam választott típus 28 lábú, tehát minden tekintetben teljesen megfelelt az én elvárásaimnak [5].



14. ábra. Új vezérlőelektronika

Ez az áramkör szintén 48MHz-ről jár, mint az előző, így könnyen át lehetett ültetni az előző változat programját erre, szinte csak az egyes lábak portolását kellett beállítani. Mivel a két mikrovezérlő azonos core-al rendelkezik, ezért ezen kívül más változtatásra nem is volt szükség. Habár USB periféria híján le kellett mondanom a Bootloader funkcióról.

Miután a Holdjáróra felkerült az új vezérlőelektronika és a MOSFET-es teljesítmény-fokozat, ideje volt kipróbálni a konfigurációt. A motorok még szinte fel sem pörögtek, de a rádió-vétel újra megállt. Az új fejlesztések csődöt mondtak, az árnyékolás és többi trükk úgy látszik nem segített.

## 8.2. Hibakeresés

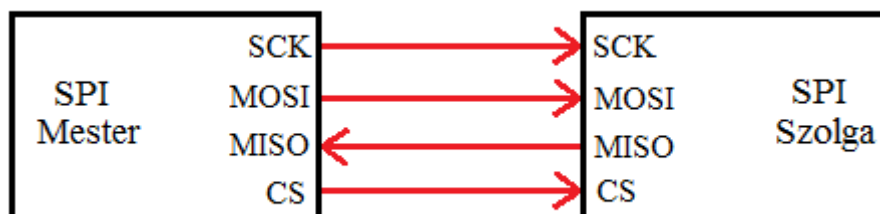
A teljesítmény-fokozat kikapcsolásával a vétel nem állt meg sose, ezért azt feltételeztem, hogy a program jó, tehát máshol kellett keresni a hibát, a hiba okát. Mivel az árnyékolás

teljesen beburkolta az új áramkört, ezért a vezérlőelektronikát a teljesítmény-fokozattal összekötő kábellel kezdtem el kísérletezgetni. Betekertem alufóliával, ennek hatására a rádió összeomlása csak a motor magasabb fordulatszámain volt megfigyelhető. Beszereztem gyári, alufóliával árnyékolt kábelt, amelynek az árnyékolása a telj. fokozat földjével került kapcsolatba. Az eredmény romlott, tehát ez nem volt járható út. Kipróbáltam többféle földelési módszert, de egyik se vezetett eredményre. Érdekes módon, ha kivettem ezt a kábelt és külsőleg kapcsoltam be a motorokat, akkor a kommunikáció nem állt le...

Kipróbáltam azt is, hogy hogy viselkedik az új vezérlőelektronika a régi teljesítmény-fokozattal. Az eredmény nem változott. A tanulság: Az új mikrovezérlős áramkör árnyékolva még rosszabbul, még érzékenyebben működik, mint a régi. Az utolsó javítási kísérlet magára az SPI buszra irányult.

### 8.2.1. SPI busz

A Serial Peripheral Interface egy soros, szinkron adatátviteli szabvány. Általában két résztvevője van: a mester és a szolga (15. ábra). Az adatcsere full-duplex módon, egyidőben történik. A kommunikációt teljes mértékben a mester irányítja.



15. ábra. SPI

A kommunikáció kezdetekor a mester a CS (Chip Select) lábát alacsony szintre állítja (aktív negált), ezzel jelzi a szolgának, hogy megkezdje a kommunikációt. A MOSI láb a mester adat kimenete, a MISO pedig az adat bemenete. Az egyes bitek átvitele két irányban történik szinkron módon. Erre szolgál az SCK órajel láb.

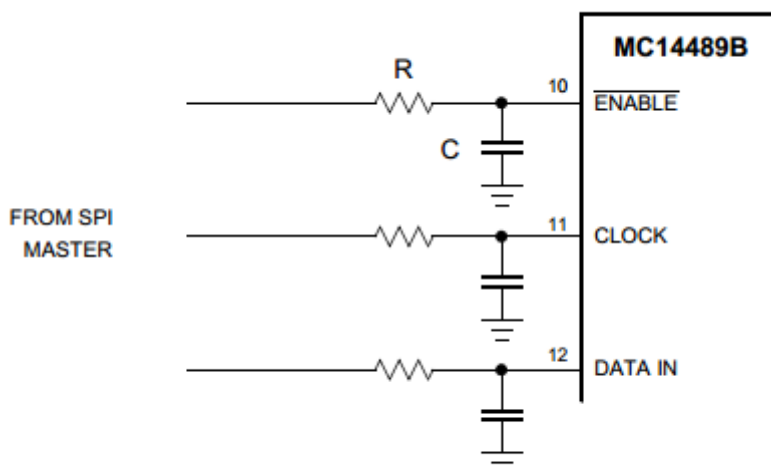
Az adatátvitel történhet az órajel fel-, vagy a lefutó élére is. Egyes alkalmazásokban az órajel tétlen állapota a logikai 1, de általában a logikai 0 a jellemző. Utóbbi beállítás jellemző a Holdjáróra is.

A Holdjáró SPI interfészénél megjelenik egy 5. láb is, ez az nIRQ. Ennek a feladata jelezni a mikrokontroller felé, hogy valamilyen esemény történt. Vételi üzemmódnál ez a láb alacsony szintre vált, ha sikeres csomagvétel történik. A mikrovezérlő pollingozással figyeli ezt a lábat, ami annyit tesz, hogy gyakorlatilag végtelen ciklusban olvassa ezen láb értékét.

### 8.2.2. SPI busz szűrése

A Holdjáró esetében az SPI busz frekvenciája kb. 150-200kHz-re tehető. Logikai analizátorral vizsgálva a menet közbeni kommunikációt arra jutottam, hogy összeomláskor egy-egy bit megváltozik az adatvonalakon, így feltételezhetően elállítódik a rádiós modul. A feltételezésemet az is bizonyította, hogy egy összeomlás utáni RF modul újrainicializáció újra működőképesé teszi az áramkört. A diagnózis tehát meglett, most már csak gyógyírt kellett találni rá.

Felmerült bennem, hogy a zavar valószínűleg elég gyors fel- illetve lefutású, ezért egy megfelelően méretezett RC aluláteresztő szűrővel talán kiszűrhető. A Motorola egy ajánlásában találtam a 16. ábrán látható rajzot.



16. ábra. SPI lezárás és leválasztás [6]

100kHz-es órajelű kommunikációhoz 100Ω ellenállás és 1nF-os kapacitás értéket ajánlottak. Ezekkel próbálkoztam én is, sajnos sikertelenül. A kondenzátorok értékével elmentem

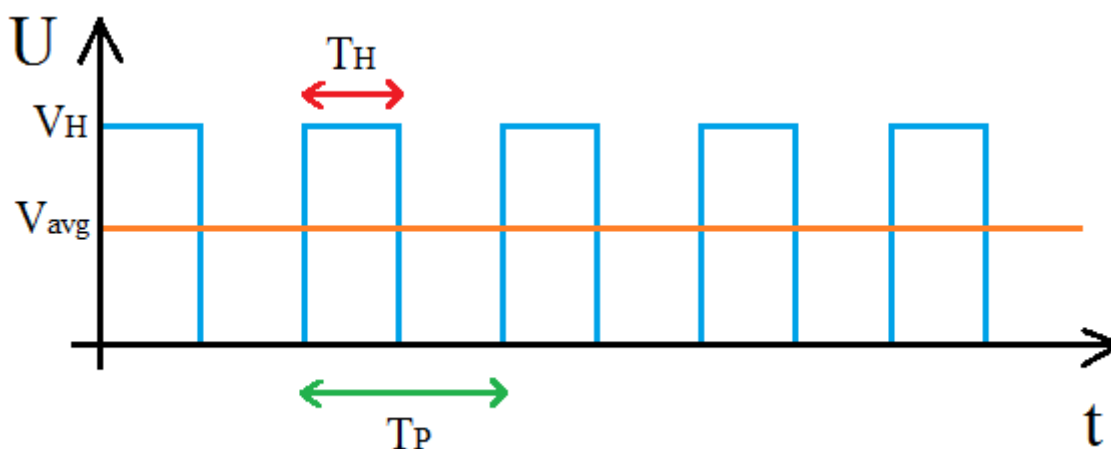
egészen 10nF-ig, de akkor már egyáltalán nem működött a kommunikáció – valószínűleg a szűrő már nagyon csillapította az órajel alapharmonikusát is.

Ötlet híján egyelőre a régi vezérlőelektronika és telj. fokozat páros van használatban.

### 8.3. PWM

A mikrovezérlő CCP moduljának egyik szolgáltatása a PWM mód (impulzusszélesség moduláció). Ez a periféria teszi lehetővé a motorok fokozatmentes – a valóságban diszkrét – gyorsítását azáltal, hogy a be- illetve kikapcsoláson kívül a motorok számos köztes sebességgel is képesek forogni.

A PWM moduláció a 17. ábrán látható. A kimeneti feszültség teljes mértékben megjelenik a fogyasztón, ezáltal minimalizálható a kapcsoló tranzisztorokon disszipálódó hőmennyiség. Annak érdekében, hogy a felvett teljesítmény mégis szabályozható legyen, szükséges a feszültség megfelelő gyakorisággal és szélességgel történő kapcsolgatása.



17. ábra. Impulzusszélesség moduláció

Kék színnel van jelölve kimeneti feszültség, narancssárgával az átlagfeszültség. A PWM jel fontos paramétere a frekvenciája, ami a másodpercenkénti impulzusszámot adja meg. Ennek a reciproka a periódusidő, amely  $T_P$ -vel van jelölve a fenti ábrán. Az impulzus szélességét  $T_H$  jelöli. A kimeneti feszültség és az átlagfeszültség közti összefüggést az alábbi képlet adja meg:

$$V_{avg} = V_h \cdot \frac{T_h}{T_p}$$

A  $\frac{T_h}{T_p}$  hányados adja a kitöltési tényezőt (D), amelyet százalékban fejezünk ki. Ez a képlet jól alkalmazható motorokhoz, viszont rezisztív terheléseknél nem.

Mivel a motor induktív fogyasztó, ezért rajta az áram nem tud ugrásszerűen változni. Mivel a belső feszültsége az áram idő szerinti deriváltjától (is) függ, ezért a feszültség sem tud gyorsan változni. Ebből az következik, hogy a motor nem tudja követni a PWM gyors változásait, ebből kifolyólag ő csak az átlagfeszültséget fogja látni. Másfelől közelítve, a motor – mint egy fojtótekercs – elnyomja a PWM spektrumában az AC komponenseket. Ami marad, az nem más, mint a DC átlagfeszültség.

Rezisztív terhelésnél az AC komponensek is végeznek munkát, tehát ott az effektív feszültséggel kell számolni:

$$V_{eff} = V_h \cdot \sqrt{D}$$

## 9. Az irányítás

A Holdjáró manuálisan és utasításokkal irányítható, ezen felül mára már elkészült az automata üzemmód is. A kézi irányítás tovább osztható joystick általi vagy URC-s irányításra.

### 9.1. Botkormány

Joystick használatakor az adóállomás másodpercenként kb. 50-szer leolvassa a botkormány és a gombok állapotát, majd ezekből az információkból 3 bájtos csomagokat kreálva átküldi rádión 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, szükség volt egy olyan saját függvényre, 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};
```

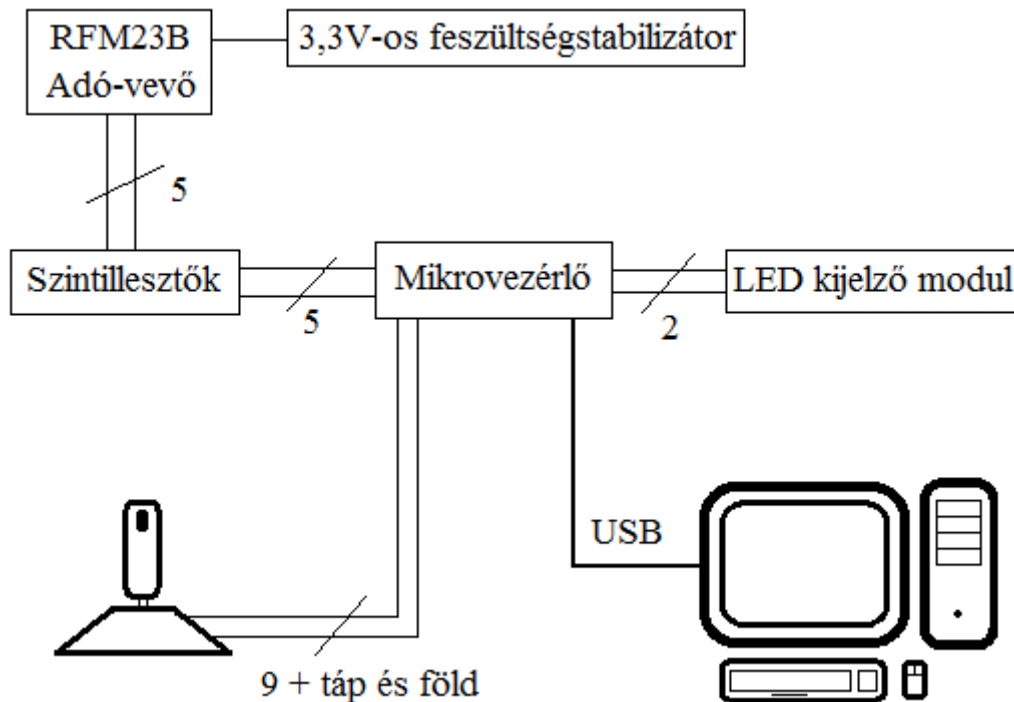
Mivel a mikrovezérlő PWM (impulzusszélesség moduláció) perifériája 10 bites, ezért 1024 féle kitöltési tényezőt lehet rajta beállítani, amelyek közül az 1023-as a 100%-os, a zéró pedig a 0%-os kitöltés. A tömb indexelése nullával kezdődik, és az egyes indexek egy konkrét sebességet jelölnek. Ha a Holdjáró 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 rádión vett csomag. A nyomógomb nyomva tartásakor 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.

## 9.2. Adóállomás

Az adóállomás felépítését a 18. ábra szemlélteti:



18. ábra. Az adóállomás felépítése

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. Feladata a joystick X- és Y-tengelyű sebességeinek a kijelzése.

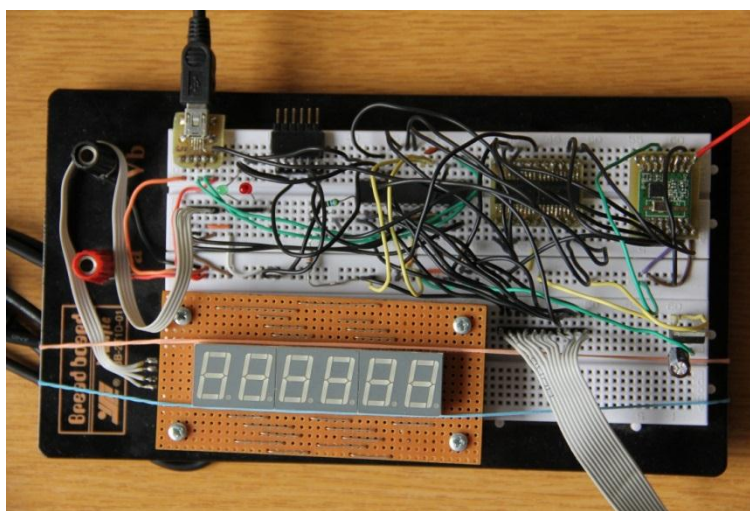
Az adóállomás működésében két üzemmód létezik:

- joystick általi irányítás,
- kliens programmal történő utasítás-kiadás.

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.

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 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.



19. ábra. Adóállomás jelenlegi realizációja

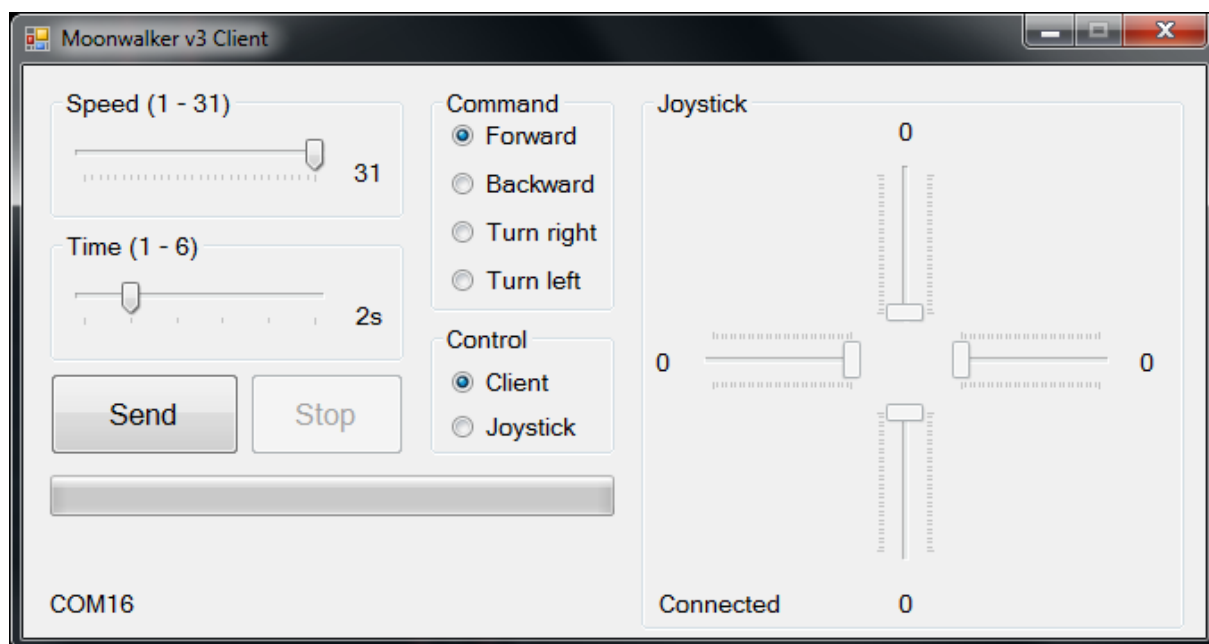
### 9.3. Számítógépes program

A kliens program segítségével négy mozgástípus közül választva egy megadott ideig (1–6 másodperc) mozgásra utasítható a Holdjáró. Lehetőségünk van arra is, hogy ezen idő letelte előtt megszakítsuk a műveletet (Stop).

#### Mozgástípusok:

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

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



20. ábra. Program

Az alkalmazás a Microsoft Visual Studio 2008 fejlesztőkörnyezetében készült, C# nyelven. A bal alsó sarokban jelenik meg az a virtuális port, ahol elérhető az adóállomás. Ezt kézzel módosítani nem lehet, a program a legutolsóként felismert portot kezdi el használni. Az utasítás kiadása után a progress bar mutatja a művelet előrehaladását. A sebesség és az idő egy csúszkán állítható. Az utasítás és vezérlés típusát radio gombokkal lehet kiválasztani. Joystick használata esetén, a jobb oldalon lévő csúszkák valós időben mutatják az X- és Y-sebességeket. Mindig csak azok a vezérlők érhetőek el, amelyek az adott üzemmódban szükségesek. A többi ilyenkor letiltásra kerül.

### 9.3.1. Az adóállomás és a kliens közti kommunikáció

| Irány              | Név  | Adat    |         |         |         |         |
|--------------------|------|---------|---------|---------|---------|---------|
|                    |      | 1. bájt | 2. bájt | 3. bájt | 4. bájt | 5. bájt |
| PC →<br>Adóállomás | MOVE | 0x41    | speed_x | speed_y | button  | time    |

|                    |                              |      |         |         |   |   |
|--------------------|------------------------------|------|---------|---------|---|---|
| Adóállomás<br>→ PC | MOVE_ACK                     | 0x42 | -       | -       | - | - |
| PC →<br>Adóállomás | STOP                         | 0x43 | -       | -       | - | - |
| Adóállomás<br>→ PC | STOP_ACK                     | 0x44 | -       | -       | - | - |
| PC →<br>Adóállomás | CLIENT<br>CONTROL            | 0x45 | -       | -       | - | - |
| PC →<br>Adóállomás | JOYSTICK<br>CONTROL          | 0x46 | -       | -       | - | - |
| Adóállomás<br>→ PC | JOYSTICK<br>DISCONNECTED     | 0x47 | -       | -       | - | - |
| Adóállomás<br>→ PC | JOYSTICK<br>CONNECTED        | 0x48 | -       | -       | - | - |
| Adóállomás<br>→ PC | JOYSTICK<br>SPEED STATE      | 0x49 | speed_x | speed_y | - | - |
| PC →<br>Adóállomás | NEED JOYSTICK<br>SPEED STATE | 0x4A | -       | -       | - | - |
| PC →<br>Adóállomás | STOP JOYSTICK<br>SPEED STATE | 0x4B | -       | -       | - | - |

21. ábra. A saját protokoll

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 (21. ábra). 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.

## 9.4. URC

Az URC (Universal Remote Controller – univerzális távirányító) segítségével valósul meg a második fajta kézi irányítás. Ez az eszköz egy teljesen egyedi, módosított videojáték kontroller (22. ábra).

A koncepció az volt, hogy legyen egy saját távirányítóm, amely teljesen mobil, saját feszültségforrással rendelkezzen, tudja kezelni az összes általam épített, és a jövőben megépülő modelleket, robotokat. Fontos elvárás volt a magam részéről az interaktivitás, különböző beállítások, kijelző és menük kezelése.

A célok kitűzése után keresni kezdtem egy olyan videojáték kontrollert, amelyhez található adatlap és leírás. A választásom a Sony Playstation, DualShock 2 kontrollerére esett. Ideális volt abból a szempontból is, hogy két analóg joystick-kel rendelkezett.



22. ábra. URC

Az interneten keresztül sikerült egy olcsó, használt darabhoz hozzájutnom. A legelső dolgom az volt, hogy megvizsgáljam, mennyi szabad hely van a belsejében. Sajnos a mérnökök elég jó munkát végeztek, szinte minden hely ki volt benne használva. A teljes hosszában végighúzódtott a controller nyáklemeze, rajta a vezérlő chip-pel. A nyomógombok is úgy voltak kialakítva, hogy a nyáklemezen foglaltak helyet, ezért muszáj volt szinte mindent meghagyni. Egyedülként csak a DC motorokat távolítottam el, amelyek a vibrálásért feleltek. Ezek a controller nyúlványaiban helyezkedtek el egy-egy röpsúllyal, így hát csak ez a minimális hely szabadult fel.

Összesen három különálló, elektronikai egység került bele utólag a controller házába:

- mikrovezérlő,
- RFM23B adó-vevő modul,
- LCD kijelző.

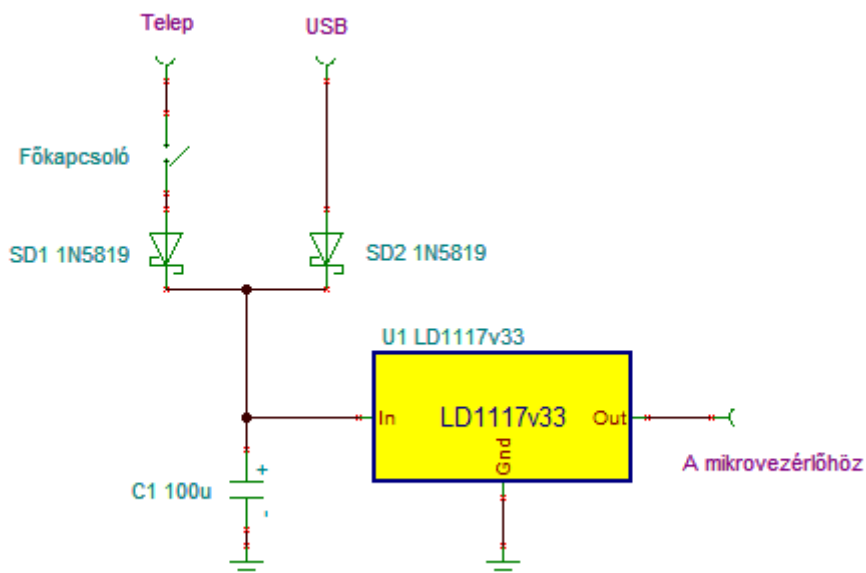
A negyedik maga a controller elektronikája. Szerencsére az összes szolga egység SPI interfésszel rendelkezett, ezért lábhiány miatt ezek közös buszra kerültek. A kommunikációt teljes egészében a mikrovezérlő irányította.

### 9.4.1. Mikrovezérlő

A helyhiány miatt szükséges volt olyan mikrokontroller használata, amely a lehető legkevesebb lábbal rendelkezik, de még el tudja látni a feladatát. A választásom a jól ismert PIC18F14K50-re esett, mivel az egyetemi előadásokon is ezzel a típussal foglalkoztunk részletesebben.

A távirányító csak újraprogramozással tehető kompatibilissé a jövőbeli eszközökkel, ezért elengedhetetlen volt USB-s Bootloader alkalmazása. Ahogy a 22. ábrán is látható, a kontroller baloldali részén kapott helyet az USB aljzat. Pár centiméterrel feljebb egy apró lyuk látható, amely mögött egy mikrokapcsoló helyezkedik el. A nyomógomb nyomva tartásával és a távirányító bekapcsolásával helyezhető az eszköz Bootloader üzemmódba.

Ahogy a mikrovezérlő, úgy az összes elektronikai egység is 3,3V-os tápfeszültségről működik. A táplálás történhet egy 4 elemből álló telepből (4 x 1,5V), vagy az USB portról. Kezdetben DC-DC konverterrel szerettem volna csökkenteni a feszültséget, de az egyszerűségekre törekedve – és helyhiány miatt – erről letettem. A feszültség csökkentését és stabilizálását egy LD1117v33 integrált áramkör végzi. Annak érdekében, hogy az USB portba ne folyhasson be áram a telepből, vagy adott esetben az USB ne kezdje el tölteni a telepet, szükséges volt védő Schottky diódák használata (23. ábra).



23. ábra. Stabilizátor

Mivel a távirányító működés közben általában telepről működik, ezért szükséges volt a telep feszültségének a monitorozása. A mikrovezérlő ADC-je tökéletes volt erre a feladatra. Fontos mennyiség a rendszerfeszültség, aminek az értéke 3,3V. Ez viszont csak akkor biztosítható, ha a feszültség-stabilizátor bemenetére legalább 4,3V kerül. Ezért a rendszerfeszültségen kívül a telep feszültsége az, ami kijelzésre kerül a kijelzőn.

Az ADC referencia-feszültségét a mikrovezérlő belső, 1,024V-os feszültséggenerátora biztosítja. Hogy ne legyen szükség hosszú matematikai számításokra, ezért a mérni kívánt pontok feszültsége egy 1:10-es osztóra kerül, ezért az AD átalakítás eredménye már rögtön  $\text{Volt} \cdot 10^{-2}$  –ben értendő. Tehát az 500-as mérési érték a valóságban 5V-ot jelent. A Schottky dióda feszültségesése figyelembe lett véve.

#### **9.4.2. RF modul**

Az univerzális távirányító adó-vevője az RFM23B modul furatszerelt változata. Jelenleg csak adó üzemmódban működik. A controllerre külső, SMA csatlakozású botantenna került. A kevés hely miatt nem került fel nyákra, a tűskéire szalagkábel lett forrasztva.

#### **9.4.3. Kijelző**

Kijelzőre azért volt szükség, hogy a különböző beállításokat menüből lehessen megadni, illetve a későbbiekben bővülő irányítandó eszközök közötti váltás egyszerűvé váljon. Pár éve sikerült egy Nokia 3310-es mobiltelefon LCD kijelzőjét működésre bírni, így hát nem volt szükség hosszas válogatásra.

A kijelző a közös SPI buszra csatlakozik, felbontása 84 x 48 pixel, monokróm. A megjelenítési terület 6 vízszintes sávra bomlik, mindegyik sáv 8 pixel magas, ezáltal a sávok egyes oszlopai 1 bájtal írhatóak le, eszerint is történik a rajzolás – 8 pixelenként. Az adatlap tanulmányozása után saját programkönyvtárat írtam a kijelzőhöz, így a szoftveres feladat megoldódott.

Mivel a controlleren nem volt elég nagy, lapos terület, ezért a kijelzőnek műanyagból házat készítettem. Az eredeti háttérvilágítás teljesítmény LED-jei a mobiltelefon nyakján maradtak, így 4 nagyfényerejű LED-ből, fehér, csillogós papírból és szivacsból összeraktam egy zöld színű háttérvilágítást (24. ábra). A szivacs itt a fény derítéséért szolgál.

A háttérvilágítás vezérlése PWM-mel történik, így a változó telepfeszültség mellett is a fényerő egy bizonyos hibahatáron belül konstansnak mondható.



24. ábra. Háttérvilágítás

#### 9.4.4. Kommunikáció a kontrollerrel

A kontroller vezérlőchip-je állandóan olvassa a gombok és joystick-ok állapotát, így amikor a mikrovezérlő ki akarja ezt belőle olvasni, akkor nem kell várnia. Kétféle üzemmódban használhatóak a nyomógombok:

- digitális
- analóg

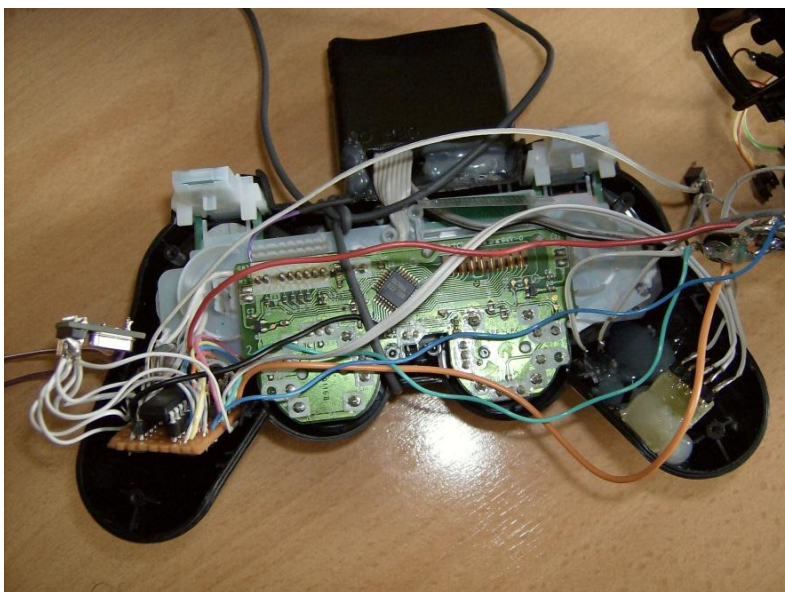
Digitális üzemmódban egy adott nyomógomb állapota 1 vagy 0 lehet – meg van nyomva, nincs megnyomva. Analóg módban a Select-en és a Start-on kívül a gombok megnyomásának nem csak a két véglete, hanem 256 számú erőssége-mértéke olvasható ki. Az én alkalmazásomban csak a joystick-ok analógok, a nyomógombok digitálisak.

Mind a két joystick az X és az Y tengely mentén is bedönthető, az egyes tengelyek pozíciói 8 bites felbontással vannak meghatározva. A két joystick között lévő piros, gyári LED valaha az üzemmódokat jelezte ki, most az átvitelkor világít. Analóg üzemmódba váltásra nincs lehetőség, egyébként teljesen felesleges is lenne.

Használt kontrollerről lévén szó a két joystick mozgása kissé bizonytalan volt kezdetben, érezni lehetett rajtuk, hogy a rúgóik nem húzzák vissza őket teljesen alapállapotba, ha

elengedjük őket. A mechanika megolajozása némiképp javított a helyzeten, ennek ellenére szükség volt szoftveres korrigálásra. Ez a korrigálás azt jelenti, hogy bizonyos határértékeket megadva nem csak akkor tekintjük középső állásban az egyik joystick adott tengelyét, ha 127-es számot olvasunk ki a vezérlőchip-ből. A középső állás egy nagyobb, kb. 10 – 30-as nagyságú intervallumot jelöl a 0-255 értékkészlet közepén. Ez persze mindegyik tengelynél más és más.

Hogy nagyobb legyen a vezetési élmény, a Holdjáró URC általi irányításánál már nem 31 sebesség létezik, hanem 50. Ez lehetővé teszi a még finomabb gyorsítást, nagyobb ívben történő térítést. Lényeges különbség, hogy a járgány két oldalának az irányítása most már teljesen elkülönül – a jobboldali joystick a jobboldali motorok, a baloldali joystick a baloldali motorok sebességét állítja be. Ebből kifolyólag az X-tengelyű joystick mozgások a Holdjáró irányításában nem játszanak szerepet.



25. ábra. Forrasztás és ragasztás után

#### 9.4.5. Menü

Ahogy az előzőekben már utaltam rá, a menü a könnyű kezelés elősegítésére jött létre. Navigáláshoz a kontroller Fel és Le gombjait, kiválasztáshoz a Start gombot lehet használni. A főmenüben jelenleg 3 opció található:

- Moonwalker
- Service menu

- Reset

A Reset a mikrovezérlő újraindítására szolgál. Azért került ide bele, hogy fejlesztéskor ne kelljen minden újraprogramozásnál kihúzni és bedugni az USB kábelt. Ehelyett a mikrokapcsoló nyomva tartásával és reseteléssel egyszerűen Bootloader üzemmódba kapcsolható a mikrovezérlő.

A Service menu volt a legelső opció, ami meg lett valósítva. Kiválasztásával a 26. ábrán látható kép fogad minket. Felhasználása a gombok és a joystick-ok ellenőrzésére irányul.



26. ábra. Service menu

Az RJX, RJY, LJX és LJY rövidítések az egyes joystick-tengelyeket szimbolizálják. Mellettük láthatjuk az analóg értékeiket. Az alattuk lévő sok nulla a nyomógombok állapotát jelöli. Ha valamelyik gombot megnyomjuk, akkor a hozzá tartozó nulla egyesre vált, és a középső piros LED világítani kezd. Az alsó sorban lévő B/S jelöli a telep, illetve a rendszerfeszültségeket. A jobb felső sarokban látható a Service menu-ből való kilépés nyomógomb kombinációja.

A menü mindegyik részében lehetőségünk van a háttérvilágítás kikapcsolására a Háromszög gomb megnyomásával.

A Holdjáró irányítását a főmenü Moonwalker opciójának a kiválasztásával lehet elérni. A jövőbeli eszközök irányítása szintén majd a főmenüben lesz indítható.



27. ábra. Moonwalker ablak

A 27. ábrán látható az a kép, ami a Holdjáró irányításakor látható. Az alsó sorban itt is a különböző tápfeszültségek jelennek meg. A jobb felső sarok a kilépési kombinációt mutatja. Alapesetben az adatküldés nem aktív, tehát hiába mozgatjuk a joystick-ot, nem történik átvitel. Az adás engedélyezése a bal felső sarokban látható Fel + X gombok együttes megnyomásával engedélyezhető. Újboli megnyomásuk tiltást eredményez. A kijelző középső részén két szám látható, ezek mutatják az éppen aktuális jobb- és baloldali sebességeket. Ha a sebesség nem nulla, akkor a szám mellett egy nyíl is megjelenik, amely az irányra utal. Adás csak akkor történik, ha sebesség(ek) nem zérusak. Ebben az esetben a középső piros LED is világít.

#### 9.4.6. Telemetry

Az URC-vel kapcsolatos jövőbeli tervek közé tartozik egy primitív telemetriai rendszer kiépítése, amely abból állna, hogy a Holdjáró távirányítására használt egyirányú kommunikáció (szimplex) kétirányúvá (half-duplex) lenne alakítva.

Az elgondolás a következő:

1. megmaradna a jelenlegi egyirányú üzemmód, de egy nyomógommbal kétirányúvá kapcsolható lenne a kommunikáció,

2. kétirányú üzemmódban az adatátvitel állandó lenne az URC részéről, nem csak a joystick mozgása közben történne adás,
3. az URC minden adás után vételi üzemmódba kapcsolna, és egy megadott ideig várna a Holdjáró válaszára,
4. a Holdjáró érzékelné a csomagokból, hogy kétirányú a kommunikáció, ezért telemetriás adatokat gyűjtene, majd ezeket válaszként elküldené a távirányítónak, ezután visszakapcsolna vételi üzemmódba,
5. az URC megjelenítené a telemetriás adatokat, és kijelezné a kapcsolat állapotát is (Connected-Disconnected)

Telemetriás adatoknak a következőket tudom elképzelni:

- hőmérséklet (a Holdjáró rádiós modulja rendelkezik belső hőmérővel),
- a telep és az akku feszültsége,
- vételi jelerősség,
- a Holdjáró előtt lévő távolság (szonár)

A telep feszültségének a mérése nem jelentene problémát, viszont a galvanikus leválasztás miatt az akku-feszültség meghatározása csak optocsatolón keresztül történhetne. A vételi jelerősség mérését a rádiós modul végzi, értékét minden vétel után dBm-ben ki lehet olvasni egy adott regiszterből.

## 9.5. Ultrahangos távolságmérő

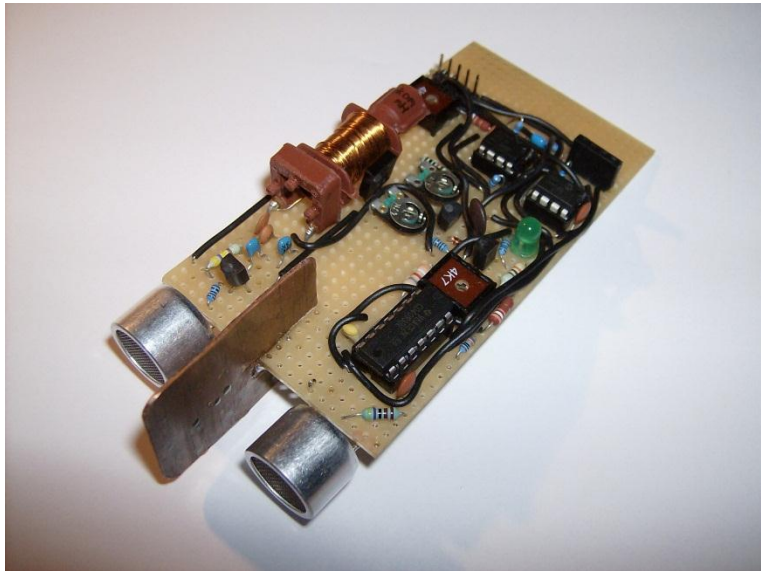
A ultrahangos távolságmérő (szonár) egy olyan sajátkészítésű érzékelő, amely ultrahang segítségével, a visszavert hanghullámokból képes a távolságot meghatározni (28. ábra).

A Holdjáró projekt mellett kezdtem el fejleszteni és tárgyak érzékelésére lett kitalálva 2,5 méteres távolságon belül. Ez az eszköz kb. egy hónap alatt készült el, és az 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 egy modul, tehát csak mérésre használható, nem tartalmaz semmiféle kijelzőt.

A modul sematikus felépítése az alábbi:

- mikrovezérlő,

- 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 csúcsegyenirányítja a vett jelet.



28. ábra. Szonár realizációja

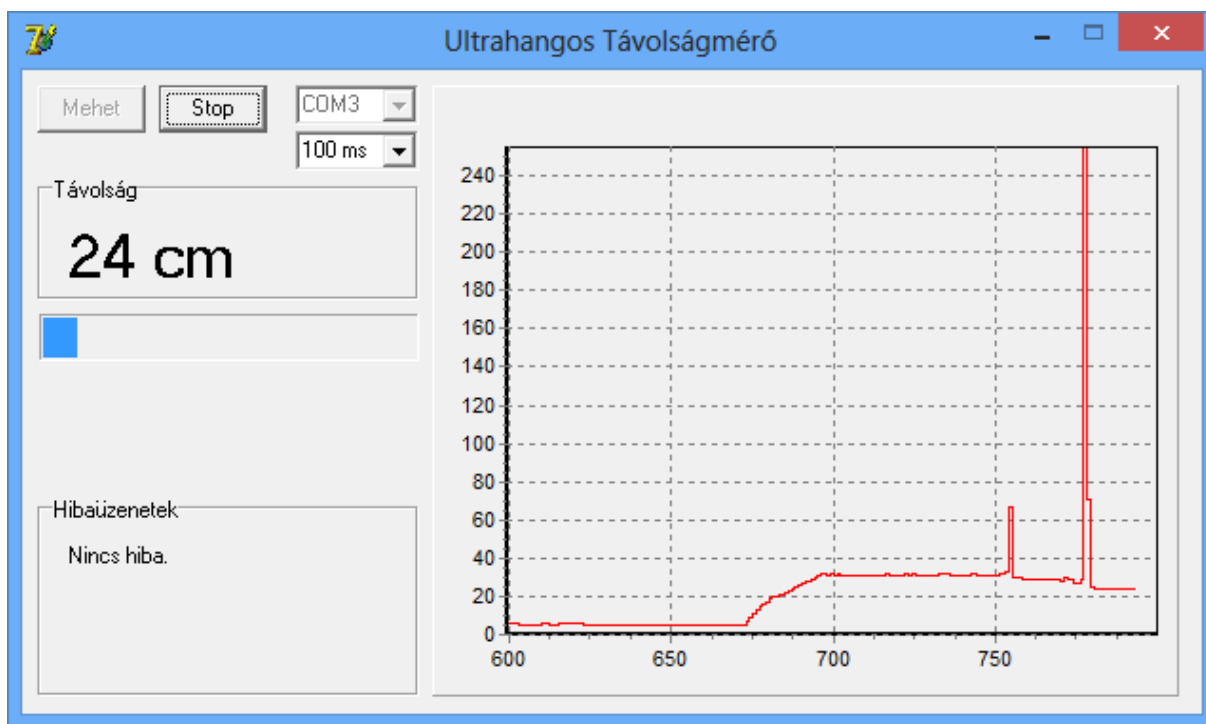
Két vezetéken keresztül, sorosan lehet kommunikálni vele, 600 bps sebességgel. A sebesség azért ilyen alacsony, mivel az egység alapját adó PIC12F675 mikrovezérlő a belső 4MHz-es RC oszcillátoráról jár, ezért csak kis bitsebesség mellett biztosítható a stabil kommunikáció. Az egyik vezeték a mérés elindítására szolgál, ez a modult használó mester mikrovezérlő egy kimenetére csatlakozik, a másik vezetéken a mért érték kerül visszaküldésre

.

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 nullaértékű 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 trükkal megspóroljuk azt, hogy a mikrovezérlőben bonyolult, művelet igényes lebegőpontos számításokat használjunk.

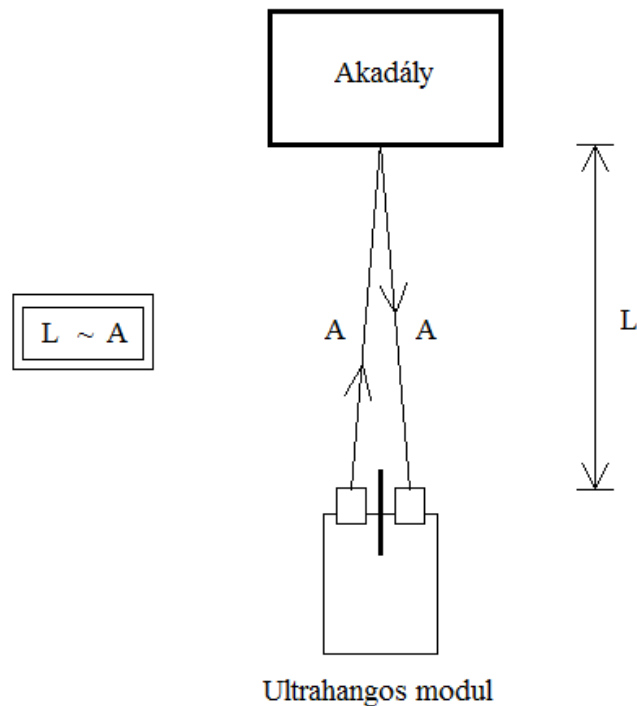
Mivel az eszközzel sorosan lehet kommunikálni, ezért számítógéphez is csatlakoztatható RS232-n keresztül. Tesztelés céljából egy monitorozó program is készült hozzá, ahol valós időben lehet figyelni a mérési értékeket (29. ábra).



29. ábra. A monitorozó program

A programban először ki kell választani a soros portot, ahol az eszköz található, ezután már indítható is a mérés. A távolság numerikusan és egy progress bar-on kerül kijelzésre. A mérési gyakoriság 100ms – 1000ms intervallumban állítható. A jobboldalon lévő görbén nyomon követhetőek az előző mérési eredmények.

Ez a mérőmodul az RC – oszcillátorok, és a kétkapszulas rendszer alkalmazása miatt picit pontatlan, főleg kis távolságok esetén, de lévén, hogy egy hobbi eszközről van szó, nekem tökéletesen megfelel (30. ábra). A feladatát az automata üzemmódban megfelelően el tudja látni, bár nem ártana, ha több lenne belőle.



30. ábra. Az ultrahang útja oda és vissza

### 9.5.1. Automata üzemmód

A kézi irányítás mellett ez a mód lehetővé teszi a Holdjárónak, hogy önmagát „vezesse”. A cél az akadályok kikerülése. Az üzemmód aktiválását az URC/Moonwalker részében történő Kör gomb megnyomása teszi lehetővé.

A szonár 5-255 cm-es méréshatára 3 intervallumra van felosztva. Ezekből a tartományokból kikerülő mérési eredményeknek megfelelően avatkozik be a vezérlőelektronika a motorok vezérlésébe.

Ha a távolság:

$\geq 20\text{cm}$ , akkor a Holdjáró előre mozog kb. 80%-os sebességgel,

$\geq 12\text{cm}$  és  $< 20\text{cm}$ , akkor a Holdjáró megáll, és véletlenszerű irányba 100%-os sebességgel forogni kezd 1-3 másodpercig,

$< 12\text{cm}$ , akkor a Holdjáró 100%-os sebességgel tolatni kezd.

Az automata üzemmódban a szonár mindig mér, tehát forgás közben is figyelve van a távolság. Sajnos a mikrovezérlőben alkalmazott pszeudo-random generátor néha eléggé

kiszámítható, így a fordulások irányában megfigyelhető néha egy bizonyos séma. Ez főképp a szoba sarkában történő forgásoknál mutatkozik meg a leginkább.

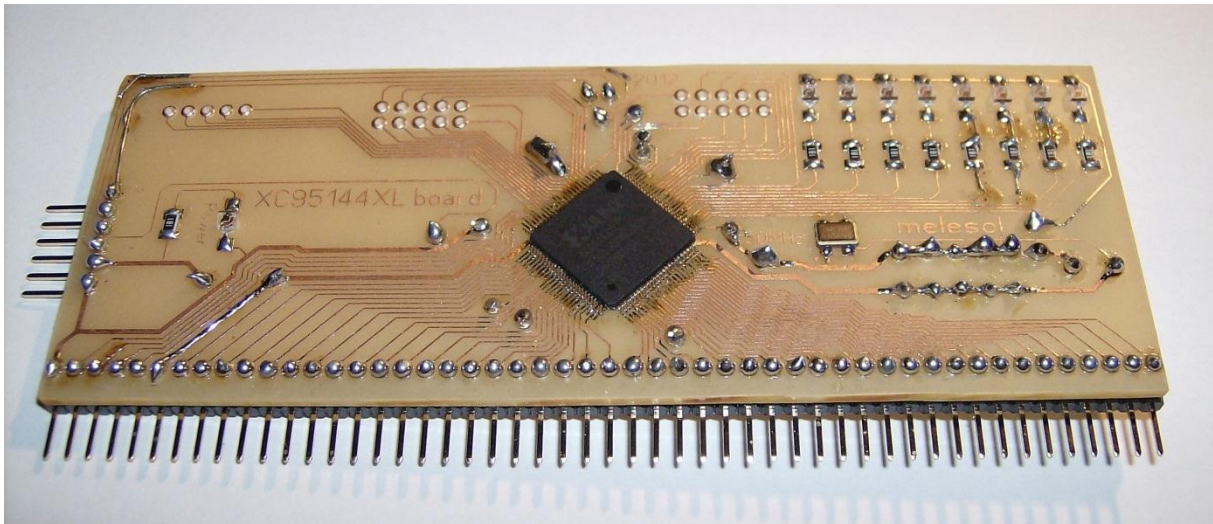
Automata üzemmódban a Holdjáró mind a két LED-je folyamatosan világít. Ha valami oknál fogva a vezérlőelektronika kérésére nem érkezik válasz a szonártól, akkor automatikusan a kézi üzemmód aktivizálódik. Mivel a szonár a Holdjáró középső részére van felszerelve, ezért előfordulhat, hogy nem érzékel egy olyan akadályt, ami pl. a jobb első kerék előtt van. Ezt a hiányosságot csak több szonárral, vagy egy szonár + több ultrahangos kapszulával lehetne megoldani. Egyelőre ez nem rendelkezik prioritással a fejlesztési listán.

## **10. 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 is használt rádiós modulok biztosítani tudnak. Nem kizárt WiFi modulok használata sem, de egyelőre a legfontosabb feladat egy adott kamera képének a kiolvasása és kijelzőn történő megjelenítése.

Az általam választott CMOS kamera a Toshiba által gyártott TCM8240MD. Ez a típus 1,3 Mp felbontású, de választhatóak alacsonyabb felbontású üzemmódok is. Választható 7,5 és 15 fps képfrekvencia. Két interfésszel rendelkezik: I<sup>2</sup>C és 8 bites párhuzamos adatbusz VBLK, HBLK és DCLK vezérlőbitekkel [7].

Az I<sup>2</sup>C a különböző beállítások megadására való, a párhuzamos porton pedig megkapjuk a képinformációt. Mivel a kamera nem rendelkezik belső pufferral, ezért a képinformációk küldését a kamera órajelezi a DCLK lábán. A gyors adatsebesség miatt a teszteléshez és az élesztéshez nem volt elég a mikrovezérlő sebessége, ezért CPLD-vel kezdtem el próbálkozni. A CPLD egy programozható logika, ahol a hardvert lehet „programozni”. Az XC95144XL típus, amit választottam, 100MHz-es órajelen is képes üzemelni. Mivel SMD alkatrészről van szó, ezért készítettem hozzá egy fejlesztő panelt, amit könnyen egy dugaszolós próbanyákhoz tudok csatlakoztatni (31. ábra). Tíz LED, és egy 50MHz-es oszcillátor is került a panelra.

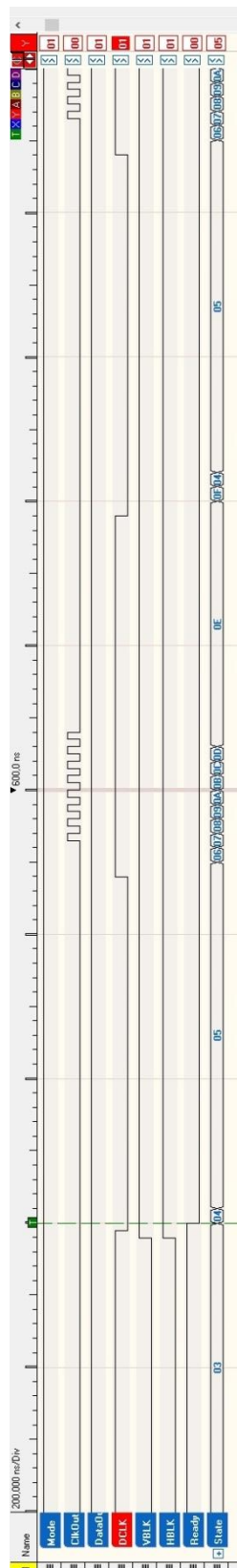


31. ábra. Saját CPLD panel

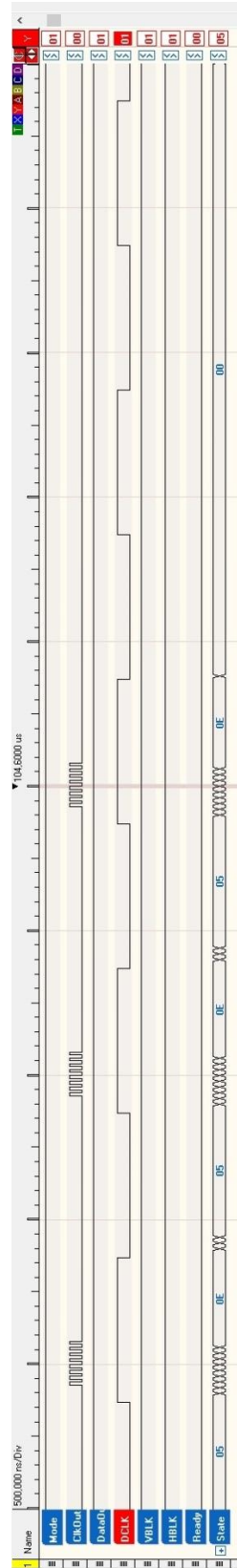
A CPLD hardverleírása VHDL-ben lett fejlesztve. A próbaalkalmazás szerint egy mikrovezérlő inicializálja a kamerát az I<sup>2</sup>C buszán keresztül, majd pedig letörli a bontásból származó 176 x 132 felbontású LCD kijelzőt. Ezután átadja a kamerát és a kijelzőt a CPLD-nek, ami lényegében párhuzamos-soros átalakítást végez: minden 8 bites adatbeolvasás után továbbítja a kijelzőnek a képinformációkat sorosan. Szerencsére a kamera és a kijelző teljesen kompatibilis volt egymással, nem kellett a CPLD-nek az adatokat tárolnia, vagy a sorrendjüket felcserélnie.

A VHDL kód szimulációja nem mutatott hibát, de a valóságban nem működött az áramkör. Módosítottam a CPLD kódján, hogy a kamerának csak a vezérlőbitjeit használja fel, a képinformációt ne, így elvileg fehérre kellett volna festenie a kijelzőt, ehelyett csak az alsó sor pár képpontja lett fehér.

Logikai analizátorral vizsgálva az áramkört úgy tűnik, hogy kezdetben jól működik a CPLD (32. ábra), de valami oknál fogva a ClkOut és a DataOut (kijelző soros interfésze) egy idő után minden ok nélkül mindig megáll (33. ábra). Egyelőre nem találtam megoldást a hibára, ezért fel fogom venni a kapcsolatot a CPLD-t gyártó Xilinx-szel, és tőlük kérek majd segítséget.



32. ábra.



33. ábra.

## 11. Végszó

A Holdjáró projektben ez idáig már sok minden megvalósult. Úgy érzem, ez idő alatt sok új ismeretet szereztem e témában mind a mikrovezérlők, mind a CPLD-k területén. Nem mondom, hogy egy eddig teljesen ismeretlen témába vágtam volna bele, de azt hiszem, hogy ez a kis mini jármű a maga nemében mégis csak valamelyest egyedülálló. Látok még benne fantáziát, ezért azt hiszem, hogy a későbbiekben tovább fogom folytatni a fejlesztését. Holott a mostani méretével és tudásával gyakorlati haszna nincs, de egy nagyobb és erősebb változatnak szerintem az ipartól kezdve az úrig biztosan van létjogosultsága és piaca.

## 12. Hivatkozások

- [1] [http://pinouts.ru/Inputs/GameportPC\\_pinout.shtml](http://pinouts.ru/Inputs/GameportPC_pinout.shtml), 2012
- [2] <http://www.kosmodrom.com.ua/pdf/RFM12B.pdf>, 2012
- [3] <http://dlnmh9ip6v2uc.cloudfront.net/images/products/09582-01.jpg>, 2012
- [4] [http://www.hoperf.com/upload/rf/RFM22B\\_23B.pdf](http://www.hoperf.com/upload/rf/RFM22B_23B.pdf), 2012
- [5] <http://erelement.com/images/motors/sn754410-sheet.jpg>, 2012
- [5] <http://ww1.microchip.com/downloads/en/DeviceDoc/41303G.pdf>, 2012
- [6] [http://www.freescale.com/files/timing\\_interconnect\\_access/doc/inactive/EB393.pdf](http://www.freescale.com/files/timing_interconnect_access/doc/inactive/EB393.pdf), 2012
- [7] <http://www.robotshop.com/content/PDF/sfe-cmos-camera-1300-1040-manual.pdf>, 2012