



Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Méréstechnika és Információs Rendszerek Tanszék

# Megbízható kommunikációs kapcsolattal rendelkező földi irányító állomás fejlesztése UAV-hoz

SZAKDOLGOZAT

*Készítette*  
Böjti Paszkál

*Konzulensek*  
Vörös András, Dr. Bartha Tamás

2013. október 28.

# Tartalomjegyzék

<b>Kivonat</b>	<b>3</b>
<b>Abstract</b>	<b>4</b>
<b>Bevezető</b>	<b>5</b>
<b>1. Háttérismertetés</b>	<b>6</b>
1.1. MTA-SZATKI ORCA . . . . .	6
1.2. Hordozható . . . . .	7
1.2.1. Komplex . . . . .	7
1.2.2. Kompatibilitás . . . . .	8
1.2.3. Megoldások . . . . .	9
<b>2. Tervezés</b>	<b>12</b>
<b>3. Saját implementáció</b>	<b>13</b>
3.1. RS232 . . . . .	13
3.2. Sebesség . . . . .	15
3.3. Irány . . . . .	16
3.4. GUI . . . . .	16
<b>4. Összefoglalás</b>	<b>17</b>
<b>Függelék</b>	<b>19</b>
F.1. Függelék1 . . . . .	19

## HALLGATÓI NYILATKOZAT

Alulírott *Böjti Paszkál*, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (Böjti Paszkál, Megbízható kommunikációs kapcsolattal rendelkező földi irányító állomás fejlesztése UAV-hoz, angol és magyar nyelvű tartalmi kivonat, 2013, Vörös András, Dr. Bartha Tamás) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hállózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervezet esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2013. október 28.

---

*Böjti Paszkál*  
hallgató

# Kivonat

Szakdolgozatom során egy biztonságkritikus robotrepülő földi állomásának grafikus felületet dolgoztam ki. Az MTA-SZTAKI Rendszer és Irányításelméleti Kutatólaboratóriumában kidolgozott szabályozó algoritmusok gyakorlatba való átültetésére egy pilóta nélküli járművet hoztak létre, mely a biztonságos üzemeltetés miatt, illetve az estelegesen előfordulható hibák ellen redundáns hardware elemekkel védekezik. Szakdolgozatom keretében feladtam ennek a repülőnek a földi állomásának a kialakítása volt, mely a redundánsan küldött rádiójelek feldolgozására és megfelelő megjelenítésre használ landó.

# **Abstract**

In English..

# Bevezető

Napjainkban egyre nagyobb teret hódít a pilóta nélküli légi járművek alkalmazása. Az 1960-as években a hadszíntéren jelentek meg először, ahol megfigyelésre, felderítésre, olyan feladatokra használták, ahol kockázatos lett volna emberi életet veszélyeztetni. Az utóbbi években praktikussága, alacsony üzemeltetési költségei miatt más területeken is hasznosnak bizonyult ez a technológia, pl. geológia mintázatok kutatása, mely az emberi perspektívából nehezen észlelhető, tűzoltósági alakulatok koordinálása, otthoni hobby felhasználás. [1] A pilóta nélküli légi jármű gondolata egészen a XX. század elejére nyúlik vissza, mikor az I. világháborúban egy olyan távirányítású repülőt alkottak, mely robbanószerrel a fedélzeten a célpontba csapódva okozott kárt. [2] Első nagyobb amerikai UAV projekt 1959-ben kezdődött, mikor aggódtak az ellenséges területre berepülő pilóták életéért. Később a vietnámi háborúban több mint 3000 küldetésben vett részt ilyen repülő és minden összes 554 veszett oda. A technológiai korlátok miatt a fő funkcionálitása video felvétel készítése egy meghatározott útvonalon (általában egyenes vonal, körökkel kiegészítve) repülve, majd a bázisra való visszaérkezés. A rádiótechnika fejlődése miatt egyre összetettebb feladatok elvégzésére lettek képesek. A nagyobb átviteli sebességek köszönhetően valós időben, monitoron keresztül kezelheti az operátor a távirányítású repülőt. A mai UAV-k több üzemmódot is támogatnak, egyik az előbb említett távirányítás, másik a fedélzeti intelligenciára hagyatkozó. Mind a hagyományos repülőiparban, mind ebben az érőben, szükséges és célszerű az emberi terhelés csökkentése, utasszállító gépek esetében is a robotpilóta elvégez minden olyan korrekciót, melyet azelőtt a pilóta folyamatos figyelésével, koncentrációjával lehetett elérni. A modern integrációknak köszönhetően, olyan fejlett feldolgozóegységgel dolgozhatjuk fel az adatokat, melyeknek nem jelentős a fogyasztása, nem foglalnak sok helyet. A szenzorokból érkező információkra, az algoritmusoknak köszönhetően úgy tud reagálni, hogy az nem veszélyezteti a repülő levegőben maradását. Ám hiába a fejlett hardware, a valóban automatikus üzemeltetés még mindig távoli cél, emberi beavatkozás minden is kelleni fog le-, felszállásoknál, illetve olyan helyzetekben melyre nincs előre felkészítve az intelligenciája.

# 1. fejezet

## Háttérismерet

### 1.1. MTA-SZATKI ORCA

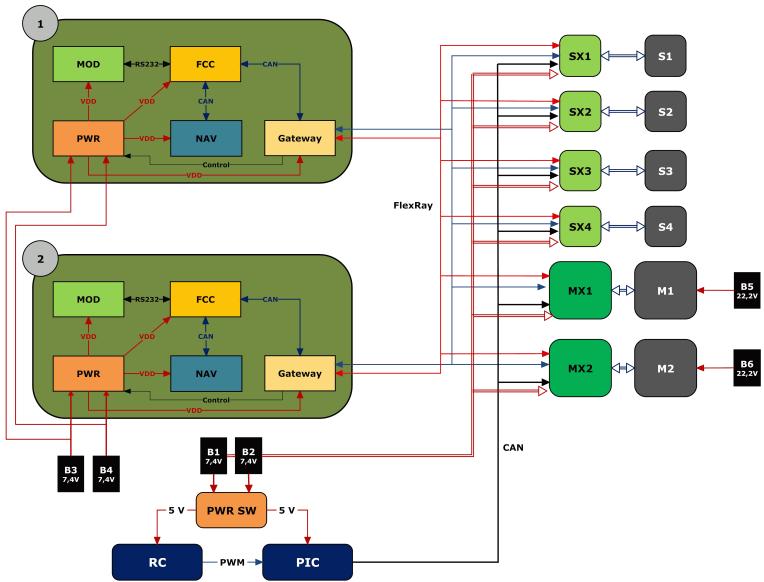


1.1. ábra

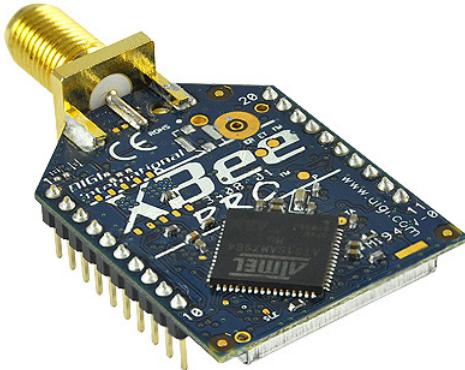
A repülőgépen redundánsan szerepel:

- motor
- akkumulátor
- központi számítógép

Látható, hogy központi számítógép kettőzött, párhuzamosan működnek, irányítják a servo motorokat. A szenzorokból érkező jelek egy Gateway-en keresztül az FCC-be jutnak, ahol a feldolgozás utána a szabályozó algoritmusok kiadják a megfelelő utasítást a kormányszerveknek. Ilyen utasítás lehet pl, a csűrőlapok adott fokban történő elmozdítása, motor fordulatszámának változtatása. A motorok, úgy mint a központi egység, külön tápellátást kapnak, ezzel is csökkentve a végzetes meghibásodás esélyét. Jelen esetben egyik legfontosabb elem a kommunikációért felelős modem, mely soros porton keresztül kapcsolódik az FCC-hez. A vevő oldalon hasonló modem, szintén soros porton küldi a földi állomásnak a vett jelet, a köztük lévő vezeték nélküli kommunikáció saját szabványa a cégnek. A választás XBee-PRO 868 típusú modemre esett, mely alacsony fogyasztása és nagy hatótávolsága miatt ideális egy ilyen környezetbe.



## 1.2. ábra



### 1.3. ábra

Egy UAV vezetéséhez elengedhetetlen egy bázis, ahonnan a földi személyzet irányítja, monitorozhatja a repülést. Általában több funkciót lát el:

- Küldetés tervezés: útvonal meghatározása, illetve a célpontok kijelölése
  - Adatok megjelenítése: megfelelő módon kijelezni a repülőgép állapotát, esetleges hibáit.

## 1.2. Hordozható

Egy egysége építették az akkumulátort, kijelzőt, kezelőszerveket, hordozhatósága miatt ez az elterjedtebb a kézzel indítható UAV-k esetében.

### 1.2.1. Komplex

Általában közel a hadszíntérhez üzemeltetik a repülőket, így a személyzetet, illetve a felszereléseket páncélozott járművel szállítják, védik. Pl. az MQ-1 Predator GCS egy utánfutóban



**1.4. ábra.** Hordozható GC

helyezkedik el, mely szünetmentes tápegységet biztosít a pilóta és segédszemélyzeti, adat-elemző/rögzítő és radar munkaállomásoknak. A kommunikációt UHF és VHF frekvencián bonyolítják közvetlen rálátás esetén, azon kívül műhold közbeiktatásával.



**1.5. ábra.** Utánfutós megoldás

### 1.2.2. Kompatibilitás

[4] Ahogy jöttek az újabb gépek, problémát okozott, hogy mindegyik földi állomása küllönbözőt egymástól, így 2008-ban szorgalmaztak egy univerzális GCS architektúrát, mely minden addigi és jövőbeli katonai UAV-vel ugyanúgy tud kommunikálni, adatokat szolgáltatni. Így született a Raytheon s Common Ground Control System, melynek fejlesztéséhez videojátékokból merítettek ötleteket. Csökkenteni próbálták a kiképzési időt, ne kelljen

több hónapos kurzusokon részt venni, hanem úgy, mint egy videojátéknál, kisebb gyakorlás után már tudja az ember, mit hol kell keresni, illetve úgy érezze, hogy valóban ő vezeti a repülőt.



**1.6. ábra. UGCS**

### 1.2.3. Megoldások

Számos megoldás született a földi állomás GUI<sup>1</sup>-jának kialakítására. Elsődleges követelmény, hogy az operátor mindenkor minden fontosabb információt láthassa, ehhez szoftverergonomiaiaknak kell megtervezni a műszerek, adatok elrendezését. Az alábbiakban összehasonlítással kerülnek a megjelenítési felületek.

#### MIT GCS

[5]Elsődleges megfontolás az volt, hogy a műszereket az értékeket linerisan jelenítsék meg, mivel az avionikában ez a kérdés már felmerült, így a valós műszerek alapján a lineárisat választották. Másodlagos kérdés a műszerek formája, itt megvalósítható lenne a csupán digitális értékek kiírása, de szintén a repülésből merítve a kör alakú egységeket választottak

#### Rodan

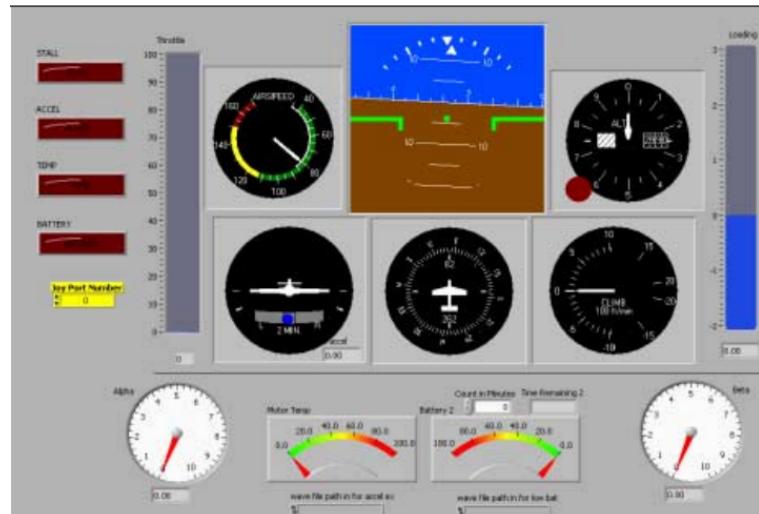
[6]Földi, légi, vízi járműveket is támogat, TCP/IP-n keresztül kommunikál, így videó átküldésére is van sávszélesség. SIL<sup>2</sup> és HIL<sup>3</sup> támogatás.

A GUI itt már összetettebb, a kamera képe foglalja el a képernyő nagyobbik részét, mellette egy térképen láthatjuk a jármű eddigi útvonalát. Egyetlen vizuális műszer a műhorizont, a többi érték csak számadattal van jelezve.

<sup>1</sup>Graphics User Interface, grafikus megjelenítés

<sup>2</sup>Software In the Loop

<sup>3</sup>Hardware In the Loop



**1.7. ábra.** Alap összéallítás



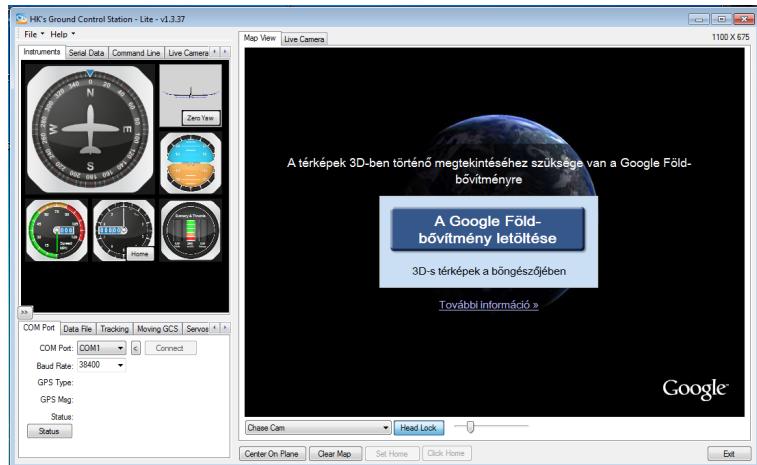
**1.8. ábra.** Rodan

## HappyKillmore

[7] Ez a GUI az ArduPlane nevű nyílt forráskódú, háziépítésű UAV-hoz készült. Itt is a térkép nézet dominál, oldalt mozgathatjuk számunkra megfelelő elrendezésbe a műszereket. Több különböző oldal közül választhatunk, ha pl. a bejövő adatokra vagyunk kiváncsiak vagy a soros port port számát szeretnénk beállítani. Lehetőségünk van exportálni is az adatokat.

## Arducopter

Az előzőhöz hasonlóan ez is az ArduCopterhez készült. Itt lehetőségünk van térképen előre kijelölni, milyen útvonalat járjon be felszállás után.



1.9. ábra. *HK*



1.10. ábra. *Arducopter*

## Viking

[8] Itt egyértelműen a térkép feletti pozíció van a hangsúly, jobb oldalt néhány műszer, illetve a fontosabb adatok számokkal



1.11. ábra. *Viking*

## **2. fejezet**

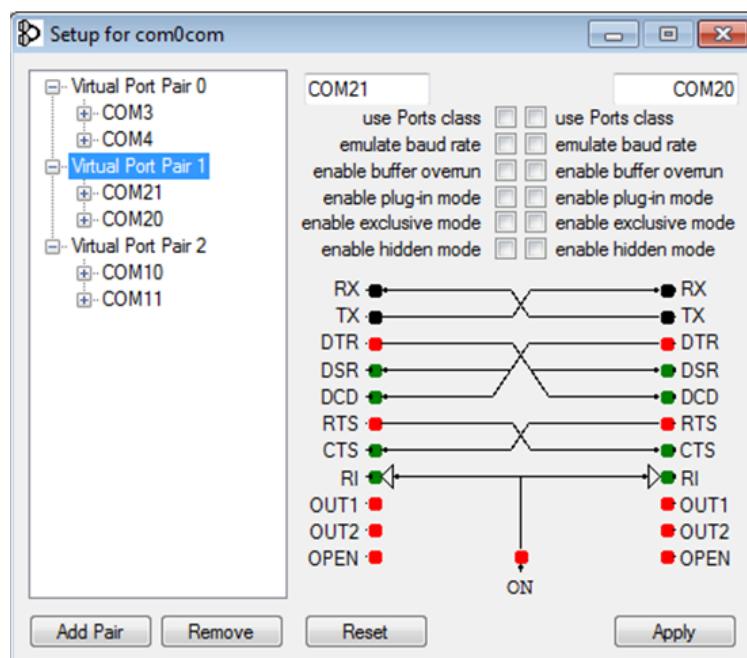
### **Tervezés**

### 3. fejezet

## Saját implementáció

### 3.1. RS232

A modemből érkező adatokat soros porton keresztül fogadja a program, a tesztkörnyezet felállításához HIL adatok szolgáltak. A küldött log fájlokat egy programmal beolvasom és egy [11]null-modem segítségével sorosporton keresztül küldöm a megfelelő portra.



3.1. ábra

Beállítottam 2 párt, COM20-COM21 és COM10-COM11 között, a pároson küldöm, páratlanon fogadom az üzeneteket.

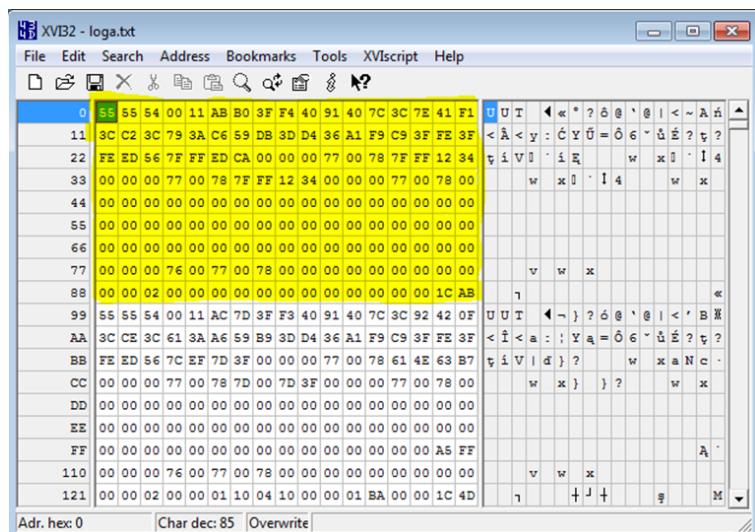
153 bájtos egy csomag, melyet egy UUT 3 bájtos fejléc és egy 2 bájtos checksum zár. A checksum a hasznos bájtok 16 bitre csonkolt összege. minden fogadott csomagnál, a feldolgozás előtt kiszámolom az összeget és ellenőrzöm, az egyezést, a rossz csomagok egyelőre eldobásra kerülnek.

Fogadó oldalon a két sorosport aszinkron ír 1-1 byte tömböt, melyből egy dekódoló függvénytel nyerjük ki a sebesség, pozíció, irány, stb. adatokat.

```
file:///C:/Users/Pasquale/Desktop/sorossend/sorossend/bin/Debug/sorossend.EXE
Serial ports:
COM1
COM21
COM3
COM10
COM4
COM20
COM11

com10.          sending B0*153 bytes
1*153 bytes
2*153 bytes
3*153 bytes
4*153 bytes
5*153 bytes
6*153 bytes
7*153 bytes
8*153 bytes
9*153 bytes
10*153 bytes
11*153 bytes
12*153 bytes
13*153 bytes
14*153 bytes
15*153 bytes
16*153 bytes
```

### 3.2. ábra



### 3.3. ábra

```
public double[] Decode(byte[] array)
```

Ebben a függvényben ellenőrzöm, a checksum-ot, illetve a kezdő UUT bájt hármast. Mivel bájtosával lehet feldolgozni az adatokat, így pl. a 4 bájtos időbélyeget 4 db egymás után jövő bájtból kell összerakni:

```
uint ido = (uint)array[3]<<24 | (uint)array[4]<<16 |  
(uint)array[5]<<8 | (uint)array[6];
```

Ugyanígy folytatódik az adatok feldolgozása, az előre megadott protokoll szerint.

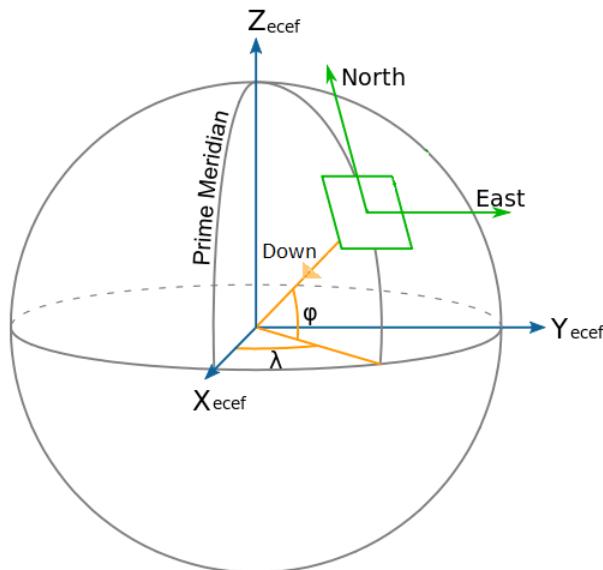
bájt index	leírás	típus	skálázás	offset
1	start	char(fix 'U')		
2	start	char(fix 'U')		
3	start	char(fix 'T')		
4	idő 1/4	unsigned int	10000	0
5	idő 2/4			
6	idő 3/4			
7	idő 4/4			
...				
26	északi irány 1/2	unsigned short	0x7FFF/400	200
28	keleti irány 1/2	unsigned short	0x7FFF/400	200
30	lefelé irány 1/2	unsigned short	0x7FFF/400	200
...				

Mivel a változások Hamming-távolsága<sup>1</sup> kicsi lenne, az eredeti számábrázoláson, így skálázással és offset képéssel megnöveljük. A sebesség adatoknál a visszakódolás:  $\text{eredeti} = (\text{nyersadat/skalazas}) - \text{offset}$  képlettel oldható meg.

### 3.2. Sebesség

A sebesség NED<sup>2</sup> koordinátarendszerben van megadva, mely a repülő középpontjából indul. Mivel kis magasságban repül a repülő, így síknak közelíthetjük a Föld felületét, ez számítások szempontjából előnyös, mivel könnyebb vele dolgozni. A sebesség számítási módja:  $\sqrt[2]{(V_E)^2 + (V_N)^2}$

Emelkedés:  $-V_D$



3.4. ábra. Koordináta rendszer

<sup>1</sup>Bináris számok XOR képzésével kapott 1-esek száma

<sup>2</sup>Nort East Down, Local Tangent Plane, helyi koordináta rendszer

### 3.3. Irány

Az irány meghatározásához a 4 negyedsíkot külön kellett választani:

```
if (Ecomp > 0 && Ncomp > 0)
{
    heading = Math.Atan(Ecomp * Ncomp);
}
else if (Ecomp > 0 && Ncomp < 0)
{
    heading = Math.Atan(Ecomp * Math.Abs(Ncomp)) + 90;
}
else if (Ecomp < 0 && Ncomp < 0)
{
    heading = Math.Atan(Math.Abs(Ecomp) * Math.Abs(Ncomp)) + 180;
}
else if (Ecomp < 0 && Ncomp > 0)
{
    heading = Math.Atan(Math.Abs(Ecomp) * Ncomp) + 270;
}
```

### 3.4. GUI

A grafikus felület kialakítása során figyelembe kell venni, hogy első ránézésre a legfontosabb adatok látszódjanak. Kettő fül közül az első oldalon a legfontosabb műszerek találhatóak, jobb oldalon egy Google Maps térkép. A térkép egy lokális cache-ből tölti be az előre letöltött térképszelvényeket, így a terepen lehetőség van offline módon is használni ezt a funkciót.

## **4. fejezet**

# **Összefoglalás**

asd

# Irodalomjegyzék

- [1] [http://en.wikipedia.org/wiki/Unmanned\\_aerial\\_vehicle](http://en.wikipedia.org/wiki/Unmanned_aerial_vehicle), 2013. május 4, 10:00
- [2] [http://en.wikipedia.org/wiki/Hewitt-Sperry\\_Automatic\\_Airplane](http://en.wikipedia.org/wiki/Hewitt-Sperry_Automatic_Airplane), 2013. május 4, 10:00
- [3] [http://en.wikipedia.org/wiki/Unmanned\\_combat\\_air\\_vehicle](http://en.wikipedia.org/wiki/Unmanned_combat_air_vehicle), 2013. május 4, 10:00
- [4] <http://www.defenseindustrydaily.com/uav-ground-control-solutions-06175/>, 2013. május 4, 10:00
- [5] [http://api.ning.com/files/ga5AVWy8xTu8cx9rHdzJ73Epc\\*dzb0uy\\*UPe0O8wFxogMF6WRNW8StK3xzkG2iBk16kNFpYTZe80NgM8kbXOiyxrEwUXIt/GroundControlStation.pdf](http://api.ning.com/files/ga5AVWy8xTu8cx9rHdzJ73Epc*dzb0uy*UPe0O8wFxogMF6WRNW8StK3xzkG2iBk16kNFpYTZe80NgM8kbXOiyxrEwUXIt/GroundControlStation.pdf), 2013. március 19., 15:00
- [6] <http://www.rodiangroup.com/uav-ground-control-station.html>, 2013. március 19., 15:00
- [7] <https://code.google.com/p/ardupilot-mega/wiki/HappyKillmore>, 2013. március 19., 16:00
- [8] [http://www.vikingaero.com/uav\\_ground\\_control\\_station.html](http://www.vikingaero.com/uav_ground_control_station.html), 2013. március 19., 16:00
- [9] [http://www\\_mono-project.com](http://www_mono-project.com), 2013. március 19., 17:00
- [10] <http://msdn.microsoft.com/en-us/library/system.io.ports.serialport.aspx>, 2013. március 20, 10:00
- [11] <http://com0com.sourceforge.net/>, 2013. május 4, 10:00

# Függelék

## F.1. Függelék1