

# TDK dolgozat

**Szerző: Lovas István**

**Óbudai Egyetem**

**Neumann János Informatikai Kar**

**Alkalmazott Informatikai Intézet**



**TUDOMÁNYOS DIÁKKÖRI DOLGOZAT**

# **GPS ALAPÚ KAMERA STABILIZÁTOR ROBOTREPÜLŐGÉPHEZ**

**Szerző:** **Lovas István**  
mérnök informatikus BSc. szak, IV. évf.

**Konzulensek:** **Dr. Molnár András**  
egyetemi docens

**Dr. Stojcsics Dániel**  
adjunktus

# Tartalomjegyzék

|       |                                                          |    |
|-------|----------------------------------------------------------|----|
| 1     | Bevezető.....                                            | 6  |
| 2     | Kamerarendszerek.....                                    | 8  |
| 2.1   | RQ-1 Predator's Skyball Gimbal .....                     | 8  |
| 2.2   | Alticam 400 kamerarendszer.....                          | 8  |
| 2.3   | Ultraforce 350 kamerarendszer .....                      | 9  |
| 2.4   | ELI UAV Gimbal .....                                     | 9  |
| 3     | OSD rendszerek.....                                      | 10 |
| 3.1   | Eagle Tree OSD Pro.....                                  | 10 |
| 3.2   | Rembizi OSD .....                                        | 11 |
| 3.3   | Video Experimenter feliratozó .....                      | 12 |
| 4     | Szenzorok.....                                           | 13 |
| 4.1   | Bevezetés.....                                           | 13 |
| 4.2   | Magasságmérés és a légnyomás összefüggése.....           | 14 |
| 4.3   | BMP-085 légnyomásmérő szenzor .....                      | 14 |
| 4.4   | Sparkfun Razor 9DOF modul .....                          | 17 |
| 4.5   | GPS.....                                                 | 18 |
| 4.6   | Kommunikációs protokollok.....                           | 20 |
| 4.6.1 | I2C busz.....                                            | 20 |
| 4.6.2 | SPI - Serial Peripheral Interface.....                   | 22 |
| 4.6.3 | UART – Universal Asynchronous Receiver/Transmitter ..... | 23 |
| 5     | Aktuátorok.....                                          | 25 |
| 5.1   | Szervomotorok .....                                      | 25 |
| 5.1.1 | Szervomotor felépítése.....                              | 25 |
| 6     | OSD.....                                                 | 28 |
| 6.1   | Bevezetés.....                                           | 28 |
| 6.2   | A PAL szabvány.....                                      | 28 |
| 6.3   | Szinkronjel leválasztás .....                            | 31 |
| 6.4   | Mikrokontroller által történő rajzolás.....              | 32 |
| 6.5   | A megvalósítás .....                                     | 32 |
| 6.5.1 | 4 bites D/A átalakító.....                               | 32 |
| 6.5.2 | A szinkronjel leválasztó áramkör .....                   | 34 |

|        |                                                                                            |    |
|--------|--------------------------------------------------------------------------------------------|----|
| 6.5.3  | Videojel előállítása .....                                                                 | 35 |
| 6.5.4  | Karakterkészlet, pixel szintű képernyőkezelés.....                                         | 39 |
| 6.5.5  | Kör és vonal rajzolása .....                                                               | 41 |
| 6.5.6  | Feliratozás külső videojelre.....                                                          | 42 |
| 6.6    | OSD paneltervezés .....                                                                    | 45 |
| 7      | A kamera stabilizátor .....                                                                | 48 |
| 7.1    | Bevezetés.....                                                                             | 48 |
| 7.2    | A gép és a célpont közötti távolság kiszámítása .....                                      | 48 |
| 7.2.1  | Haversine formula .....                                                                    | 49 |
| 7.2.2  | A célpont irányának a meghatározása .....                                                  | 49 |
| 7.2.3  | Bólintási szög meghatározása .....                                                         | 50 |
| 7.3    | A forgatás .....                                                                           | 51 |
| 7.3.1  | Földhöz rögzített koordinátarendszer .....                                                 | 51 |
| 7.3.2  | A test koordinátarendszere .....                                                           | 52 |
| 7.3.3  | A forgatáshoz szükséges szögek meghatározása .....                                         | 54 |
| 7.4    | A kamera elforgatásának bemutatása .....                                                   | 56 |
| 7.5    | A megvalósítás .....                                                                       | 59 |
| 7.5.1  | A rendszer főbb hardvermoduljai.....                                                       | 59 |
| 7.5.2  | A kamera felépítése .....                                                                  | 60 |
| 7.6    | A beágyazott rendszer általános szoftverdiagramja .....                                    | 63 |
| 7.6.1  | Konfigurálás és kalibrálás .....                                                           | 64 |
| 7.6.2  | UART konfigurálás .....                                                                    | 64 |
| 7.6.3  | I2C konfigurálás.....                                                                      | 64 |
| 7.6.4  | LED-ek és gomb.....                                                                        | 64 |
| 7.6.5  | TIMER konfigurálás .....                                                                   | 64 |
| 7.6.6  | Magasságmérő szenzor kalibrálása .....                                                     | 65 |
| 7.6.7  | BMP085 adatok lekérdezése és feldolgozása.....                                             | 67 |
| 7.6.8  | IMU adatok fogadása és feldolgozása.....                                                   | 67 |
| 7.6.9  | GPS adatok fogadása és feldolgozása .....                                                  | 68 |
| 7.6.10 | Magasságmérő modul, IMU, GPS teszt .....                                                   | 69 |
| 7.6.11 | Forgatáshoz szükséges adatok kiszámítása és vezérlőjelek előállításának<br>folyamata ..... | 72 |
| 7.7    | Kamera stabilizálás kefenélküli motorral ellátott gimbal segítségével .....                | 73 |

|       |                                                         |    |
|-------|---------------------------------------------------------|----|
| 8     | Célpont magasságának a pontos meghatározása .....       | 76 |
| 8.1   | Digitális domborzatmodell .....                         | 76 |
| 8.1.1 | Global 30 Arc-Second Elevation Data Set (GOTOP30) ..... | 77 |
| 8.1.2 | Shuttle Radar Topography Mission (SRTM) .....           | 77 |
| 8.1.3 | Magasság meghatározása a GRASS GRIS segítségével .....  | 78 |
| 9     | Tesztelés .....                                         | 80 |
| 10    | Továbbfejlesztési lehetőségek .....                     | 85 |
| 11    | Összefoglalás / Summary .....                           | 86 |
| 12    | Irodalomjegyzék .....                                   | 87 |
| 13    | Ábrajegyzék .....                                       | 89 |

# 1 Bevezető

Napjainkban egyre szélesebb körben alkalmaznak pilóta nélküli légi járműveket (UAV – Unmanned Aerial Vehicle). Az 1960-70-es években jelentek meg először, bár még csak kezdetleges formában. A katonaság főként felderítés illetve harcászati céllal alkalmazza őket. Segítségükkel olyan feladatokat végeznek el, melyek túl veszélyesek ahhoz, hogy emberek életét kockáztassák teljesítésük érdekében. Egyes bevetések olyan sok ideig tartanak, hogy egy pilóta számára rendkívül megterhelő. Továbbá egy felderítés vagy megfigyelés során a felhasznált eszközök száma kevés, így a pilóta és a kezelő személyzet önmagában is többszöröse lenne a hasznos tehernek, így gazdasági szempontból sem előnyös.

Ma már nem csak a katonaság, hanem más szervezetek is egyre szélesebb körben alkalmazzák őket pl. klímakutatók, természetvédelem, mezőgazdaság, régészek. Sokkal olcsóbb az üzemeltetési költségük, mégis ugyanazokat a feladatokat képesek ellátni, mint az ember által vezetett, kezelt gépek. 2013-ban, amikor a Duna kilépett medréből a Honvédelmi Minisztérium EI és Currus Zrt. által fejlesztett és gyártott pilóta nélküli felderítő repülőgépekkel végeztek élőképes légi felderítést. A fedélzeten elhelyezett kamera segítségével, figyelemmel tudták kísérni a kritikus partszakaszokat és a látottak alapján megfelelő intézkedéseket tudtak véghezvinni.

A TDK dolgozatom célja egy olyan önálló kameraplatform létrehozása, melyet robotrepülőgépekre lehet felszerelni. Segítségével a levegőből figyelhetünk meg egy adott objektumot vagy megadott GPS koordinátákkal kijelölt területet anélkül, hogy a kamerát emberi beavatkozással irányítanánk a célra. A platformon elhelyezett kamera élőképet sugároz a földi állomásra. A megfigyeléshez és stabilizáláshoz szükséges telemetriai adatokat és egyéb információkat egy OSD (On Screen Display) egység jeleníteni meg a kamera képén.

Manapság a legtöbb objektumkövető vagy pozíciókövető rendszer bonyolult képfeldolgozó algoritmusokon alapszik. Ezeknek az algoritmusoknak az alkalmazása ma már nem jelent gondot, mert a hozzá szükséges számítási kapacitás bárki számára könnyen elérhető. Azonban nem szabad elfelejtenünk azt, hogy a számítási kapacitással a felhasznált teljesítmény is növekszik, mely általában többlet súlyt is jelent. Természetesen léteznek olyan „mini” számítógépek is, melyeknek tömege néhány száz gramm, mégis a számítási kapacitása megfelelő lenne képfeldolgozásra. Ilyen pl. a Raspberry Pi mely egy 700Mhz-es processzorra épül, vagy pedig ennek egy módosított változata az úgynevezett Hackberry A10 mely már 1Ghz-es CPU-val és hozzá tartozó GPU-val lehet megvásárolni. Azonban ezeknek az

eszközöknek a hatékonysága még mindig nagyon kicsi, így komplexebb képfeldolgozási algoritmusok futtatására nem alkalmasak. Továbbá a feldolgozáshoz szükség van egy olyan kamerára is, mely megfelelő minőségű képet állít elő. Minél rosszabb a kép minősége, annál több szűrési eljárást kell elvégezni, ami további teljesítménytöbbletet von maga után.

A robotrepülőgépek testét úgy tervezik meg, hogy a lehető legkönnyebb legyen mégis strapabíró. Erre azért van szükség, mert minél nehezebb a gép, annál nagyobb teljesítményű propulzióra van szükség. Ha nagyobb teljesítményű propulzióra van szükség, az több súlyt is jelent, mivel nem csak a motor mérete növekszik, hanem a működtetéshez szükséges akkumulátorok száma is, és természetesen a költség is.

Egy 2-3kg-os robotrepülőgép hordozható hasznos terhe maximum 1-1,5kg. 12V-os feszültségről működik, melyet akkumulátorok biztosítanak. A rakodótér mérete ~15x15x10 cm. Belátható, hogy a paraméterek alapján a feladat kivitelezéséhez nem lehet bármilyen eszközt felhasználni, tehát a nagyobb számítási kapacitással rendelkező miniszámítógépek nem jöhetnek szóba.

A célom mégis egy olyan rendszer megalkotása, mellyel az égből repülés során egy földi objektumot meg lehet figyelni. A feladat kivitelezéséhez, egy a beágyazott rendszerek világában is elő szeretettel alkalmazott mikroprocesszort fogok alkalmazni. Ez egy ARM Cortex-M4-es 168MHz-es órajellel rendelkező mikrokontroller. Nem csak teljesítményben, hanem árban is a lehető legjobb a versenytársaihoz képest. A tápfeszültsége 3.3V, áramfelvétele maximális teljesítmény esetén is kevesebb, mint 100mA! Természetesen ez önmagában nem elég, további szenzorok kellenek a megfigyelés kivitelezéséhez. Azonban a jó hír az, hogy ezeknek a szenzoroknak a súlya pár gramm, az áramfelvétele pedig szinte elenyésző, mikroamperekben mérhető. Így a hasznos teher többi része megmarad a kamera mozgását segítő motorok, és egyéb eszközök számára.

A végső célom tehát, egy olyan önálló, független kameraplatform létrehozása, mely nem alkalmaz képfeldolgozást, így méretéből adódóan hasznos teherként bármilyen pilóta nélküli gépre felszerelhető.

Dolgozatommal már 2013-ban is indultam a Tudományos Diákköri Konferencián, de azóta további jelentős fejlesztések történtek és új eredmények születtek.

## 2 Kamerarendszerek

### 2.1 RQ-1 Predator's Skyball Gimbal

A Predator RQ-1 egy felderítési célra létrehozott UAV (Unmanned Aerial Vehicle). Úgy lett megtervezve, hogy akár 24 órán keresztül is képes az ellenséges terület megfigyelés alatt tartani. Színes orrkamerával lett ellátva a távoli navigálás céljából. Egy másik egység pedig a Predator fő "szemeként" funkcionál, azaz ezzel végzik a tényleges megfigyelést. A rendszeren belül helyet kapott egy infravörös kamera, mellyel éjjel gyenge fényviszonyok között is kiváló képet tudnak előállítani valós időben, illetve egy másik változtatható rekesznyílású színes kamera, mely a nappali megfigyelésre szolgál. Ellenséges helyzetben ez alapján gyors és megalapozott döntéseket tudnak hozni ahhoz, hogy a megfelelő helyre a megfelelő bevetésre váró csapatokat mozgósítsák. A legnagyobb előnye az, hogy a hagyományos felderítésekkel ellentétben, a pilóta nincs kitéve ellenséges környezetnek, mégis egy teljes vizuális leképezést kapnak a területről a kamera segítségével.

[1]



2.1. ábra: RQ-1 Predator's Skyball Gimbal

### 2.2 Alticam 400 kamerarendszer

Az Alticam 400-as sorozatú kamerát az Insitu ScanEagle UAV fejlesztette ki. A rendszerben helyet kapott egy IMU mely lehetővé teszi, hogy a platform orientációjától függetlenül a kamera mindig a célpontra mutasson. El lett látva egy szűrő rendszerrel, mely



2.2. ábra: Alticam 400

repülés során a gép rezgéseit kompenzálja. Képfeldolgozó algoritmusok segítségével a kijelölt objektumokat automatikusan követni tudja. Alapesetben egy 640x480-as felbontású színes CCD kamerát használnak, mely 25x optikai zoomal rendelkezik. Ezen kívül további infrakamerákkal is képes együtt működni a rendszer. [2]

### 2.3 Ultraforce 350 kamerarendszer

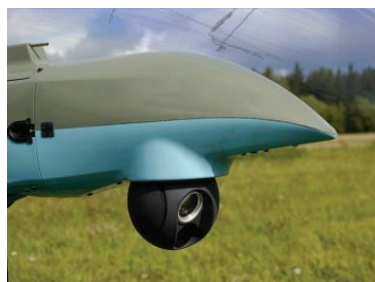


2.3. ábra: Ultraforce 350

Az Ultraforce 350 egy könnyű, nagyszilárdságú kompozit testű gimbal rendszer. A csökkentett súly miatt nagyobb repülési idő érhető el vele. 4 tengelyes aktív stabilizációval van ellátva, így kompenzálja a repülőgép mozgását és rezgéseit. A cél elemzésére 26x optikai zoommal rendelkező színes kamera áll rendelkezésre. A folyamatos megfigyelésen kívül további funkciókkal látták el. Képes meghatározni a cél távolságát és elhelyezkedését is. [3]

### 2.4 ELI UAV Gimbal

Az ELI UAV Gimbal egy kardáncsuklós, nagyteljesítményű stabilizált Pan-Tilt zoom kamerarendszer, mely az UAV robotpilóta és a földi állomás szoftverével áll kapcsolatban. A kamera giroszkóp és gyorsulásmérő által stabilizált, mely információkat a fedélzeti rendszertől kapja. A megfigyelés GPS koordináták alapján történik. A megfigyelni kívánt szélességi és hosszúsági adatokat a földi állomásról lehet továbbítani az UAV fedélzetén lévő rendszerhez. Egy nappali és egy infravörös kamera csatlakoztatható hozzá. A kamera felvételét vagy közvetlen egy videó felvevő egységgel rögzítik, vagy pedig nagyfrekvenciás adóegységgel sugározzák a földi állomásra. [4]



2.4. ábra: Eli UAV Gimbal

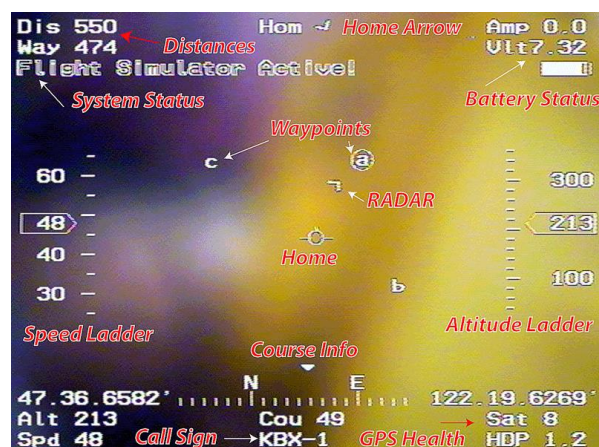
### 3 OSD rendszerek

#### 3.1 Eagle Tree OSD Pro



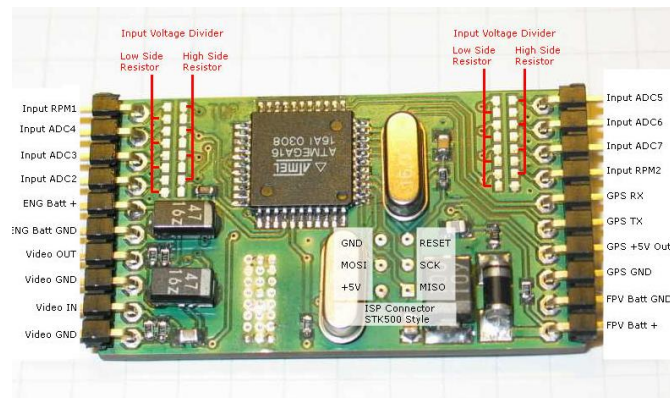
3.1. ábra: Eagle Tree OSD Pro

Az Eagle Tree OSD Pro system FPV (First Person View) repülések esetén előszeretettel alkalmazott rendszer. Segítségével az összes fontos információt meg lehet jeleníteni a gép állapotáról. Könnyen konfigurálható, a hozzá tartozó szoftverrel személyre szabása egyszerűen kivitelezhető. Számtalan érzékelő csatlakoztatható hozzá gyorscsatlakozók segítségével, így az összeállításához forrasztásra sincs szükség. Egy hangalapú riasztási rendszerrel is ellátták, mely jelzi, ha az akkumulátor töltöttségi szintje elérte azt a kritikus pontot, mely még ahhoz elegendő, hogy a gépet visszairányítsuk a kiinduló ponthoz. A kijelzőn a következő adatok megjelenítésére képes: hosszúsági és szélességi adatok, távolság az "otthontól", magasság, sebesség, akkumulátor töltöttség szintje, a szervók által felhasznált árammennyiség, hozzá csatlakoztatott szenzorok adatait, barometrikus magasság, hőmérséklet stb. A gyártó folyamatosan bővíti új szolgáltatásokkal a hozzá tartozó firmware-t, melynek a frissítése egyszerű. Egy USB kábel segítségével a számítógéphez csatlakoztatható és elvégezhető. További plusz kiegészítő, hogy földi állomás is kapható hozzá, mely segítségével adatokat lehet rögzíteni, megjeleníteni. [5]



3.2. ábra: Eagle OSD Pro megjelenített adatai

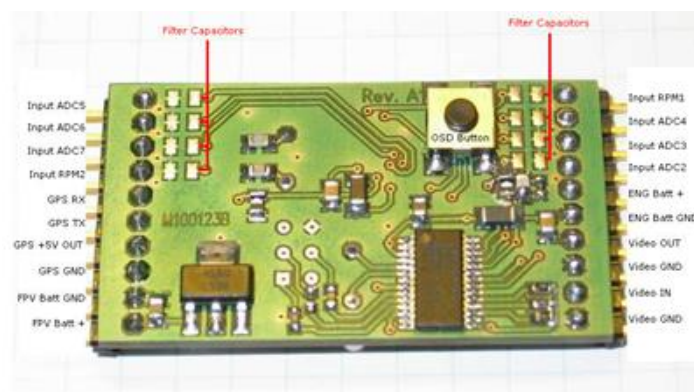
## 3.2 Rembizi OSD



3.3. ábra: Rembizi OSD 1

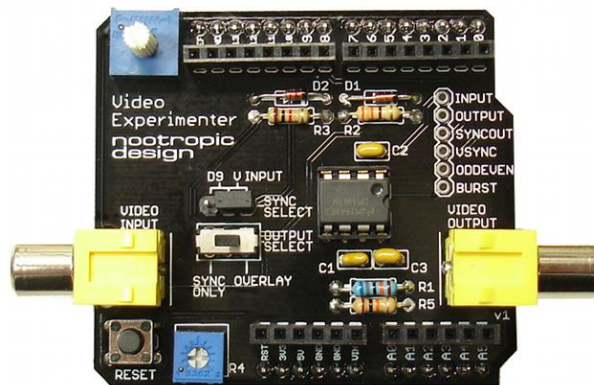
A Rembizi OSD egy amatőr fejlesztésű rendszer. A "szegény ember OSD"-jének is nevezik. Egy olyan eszköznek a megépítése volt a cél, hogy minél olcsóbban, de a lehető legtöbb funkció megjelenítésére legyen képes. Keletkezésének története némi szót hagy maga után. Két ember fejlesztette ki. A hardver tervezését és a mikro vezérlő programozását egy lengyel származású Rembizi névre hallgatató személy készítette el. A hozzá tartozó grafikus felületet pedig egy USA-ban élő Paul nevű fejlesztető valósította meg. A projekt során soha nem találkoztak, az együttműködést az internet tette lehetővé.

Tervezéskor ügyeltek arra, hogy mindenki saját kedve szerint válogathassa össze, és csatlakoztathassa a felhasználni kívánt eszközöket. Az eszközök kezelését egy Atmel ATmega16 típusú mikro vezérlő végzi. Feliratozáshoz a MAXIM által gyártott MAX7456 típusú feliratozó IC-t használja, mely megbirkózik a PAL és NTSC szabványú videó kompozit jelekkel is. Karakter szintű megjelenítésre képes. A feliratozás úgy történik, hogy az IC memóriáját SPI porton keresztül fel kell tölteni karakteres (ASCII) tartalommal, majd a kirajzolást ezek után az IC automatikusan elvégzi. [6]



3.4. ábra: Rembizi OSD 2

### 3.3 Video Experimenter feliratozó



3.5. ábra: Video Experimenter modul

Ez a projekt azok számára készült, akik szerettek volna a kompozit videojel feliratozásával kísérletezni. Egy nyílt forráskódú szoftverről és hardverről van szó. Tervezője közzé tette az interneten a hozzá tartozó kódrészleteket és kapcsolásokat is. Az Arduino Uno-hoz készült úgynevezett "pajzsként". A pajzs arra utal, hogy a nyáklap úgy lett megtervezve, hogy a rajta elhelyezett tüskesorok segítségével könnyedén össze lehessen kapcsolni őket, úgymond pajzsként lehet ráerősíteni. A videojelek feldolgozására, szeparálására egy LM1881-es szinkronjel leválasztó IC-t használ. Az összes szinkronjel számára külön kimenettel rendelkezik, mely megkönnyíti ezeknek a detektálását. A hozzá tartozó TVout könyvtár segítségével, egyszerűen lehet a feliratozást elvégezni. A panelon elhelyezett kapcsolók segítségével több üzemmód is kiválasztható. A külső videojel feliratozásán kívül a felhasználó saját maga is előállíthat kompozit jelet, csak megírt függvények közül a megfelelőeket kell alkalmaznia. A projekt tovább lett fejlesztve. Az egyszerű megjelenítésen kívül lehetőség van valós időben a kamera képének fekete-fehérré digitalizálása, különböző alakzat felismerése stb. [7]



3.6. ábra: Video Experimenter feliratozás



3.7. ábra: Video Experimenter - fekete-fehér kép

## 4 Szenzorok

### 4.1 Bevezetés

Mivel megfigyelést és stabilizálást képfeldolgozó algoritmusok nélkül szeretném megoldani, ezért több szenzort is alkalmaznom kell.

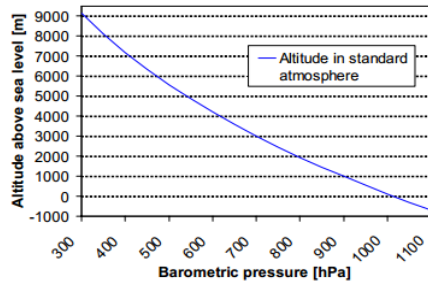
A következő adatokra lesz szükségem. Magasság, amit az adott légnyomásból lehet meghatározni, hőmérséklet, GPS adatok, giroszkóp és gyorsulásmérő adatok.

A szenzorok két fő osztályba sorolhatóak: analóg és digitális. Az analóg szenzorok a mért fizikai mennyiséggel arányos kimeneti jelet állítanak elő. Ez lehet feszültség vagy áram. Mivel a mikrokontrollerek csak digitális adatokat tudnak feldolgozni, ezért ezeket a villamos jeleket digitalizálni kell. Erre szolgálnak az úgynevezett A/D (analog-digital) konverterek. Az A/D átalakítók időben és amplitúdóban folytonos analóg jelből időben és amplitúdóban egyaránt diszkrét jelet állítanak elő. Az átalakításhoz a jelet először mintavételezni kell, utána diszkrét értékekhez hozzárendelni, majd pedig ezt valamilyen formában kódolni, ami leggyakrabban egy bináris kód. Shannon mintavételi törvénye értelmében, egy folytonos idejű jel tökéletesen visszaállítható mintáiból, ha a mintavételi frekvencia legalább kétszer akkora, mint a jel legmagasabb frekvenciájú komponense. A/D konvertálás esetén, ezt a számot általában 2 hatványaként szokták megadni. A jel feldolgozásához lehet építeni saját kezűleg A/D átalakítót, de léteznek kifejezetten erre a célra gyártott IC-k is. Manapság a legtöbb mikro vezérlő beépített A/D konverterrel van ellátva.

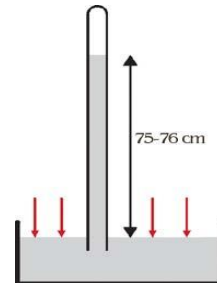
Az analóg szenzorokkal szemben a digitálisnak az az előnye, hogy a jelek diszkretizálásával nem a felhasználónak kell foglalkoznia, hanem egy jól használható, digitális jelet állít elő. Azonban ezek a szenzorok gyártótól, és modelltől függően különböző kommunikációs protokollokon keresztül kommunikálnak a külvilággal. A legelterjedtebb az USART (Universal Synchronous / Asynchronous Receiver Transmitter – Univerzális szinkron / aszinkron adó-vevő), SPI (Serial Peripheral Interface – Soros periféria interfész) és az I2C (Inter-Integrated Circuit). Ezeket a kommunikációs rendszereket később részletezem.

## 4.2 Magasságmérés és a légnyomás összefüggése

A légköri nyomás vagy légnyomás az adott területre ható nyomás, amit a levegő súlya okoz. A légnyomás először Torricelli olasz tudós mérte meg. Egyik végén zárt, kb. 1 m



4.1. ábra: Légnyomás exponenciális csökkenése a magasság függvényében



4.2. ábra: Légnyomás kísérlet

hosszú üvegcsövet megtöltött higannyal, majd a csövet nyitott végével lefelé, higanyt tartalmazó edénybe merítette. A függőleges csőben a külső higanyszinthez viszonyítva 76 cm magas higanyoszlop maradt. A csőben a higany felett légüres tér, úgynevezett Torricelli-űr keletkezett. A higanyoszlop súlyából származó nyomással a külső légnyomás tartott egyensúlyt. Innen adódik, hogy egy atmoszférikus nyomás 760higanymiliméter.

A tengerszint feletti magasság változásával a felszín felett lévő légoszlop magassága is változik. A magasság növekedésével a légköri nyomás exponenciálisan csökken.

## 4.3 BMP-085 légnyomásmérő szenzor



4.3. ábra: BMP-085 szenzor

A BOSCH által gyártott BMP-085 légnyomásmérő modul hőmérséklet kompenzált, egyedileg kalibrált precíziós mérőmodul. Segítségével a mért légköri nyomás alapján a tengerszint feletti magasságot is rendkívüli pontossággal meg lehet határozni. Ezt a modult úgy tervezték, hogy bármilyen mikrokontrollerhez közvetlenül csatlakoztatható legyen. I2C buszon keresztül kommunikál. 3,3V-os tápfeszültséggel működik. Nyomástartománya 300-1100hPa, azaz 9000m-es tengerszintfeletti magasság és -500m közötti tartományban használható. Több üzemmódban működtethető. Pontossága kis energiaigényű üzemmódban

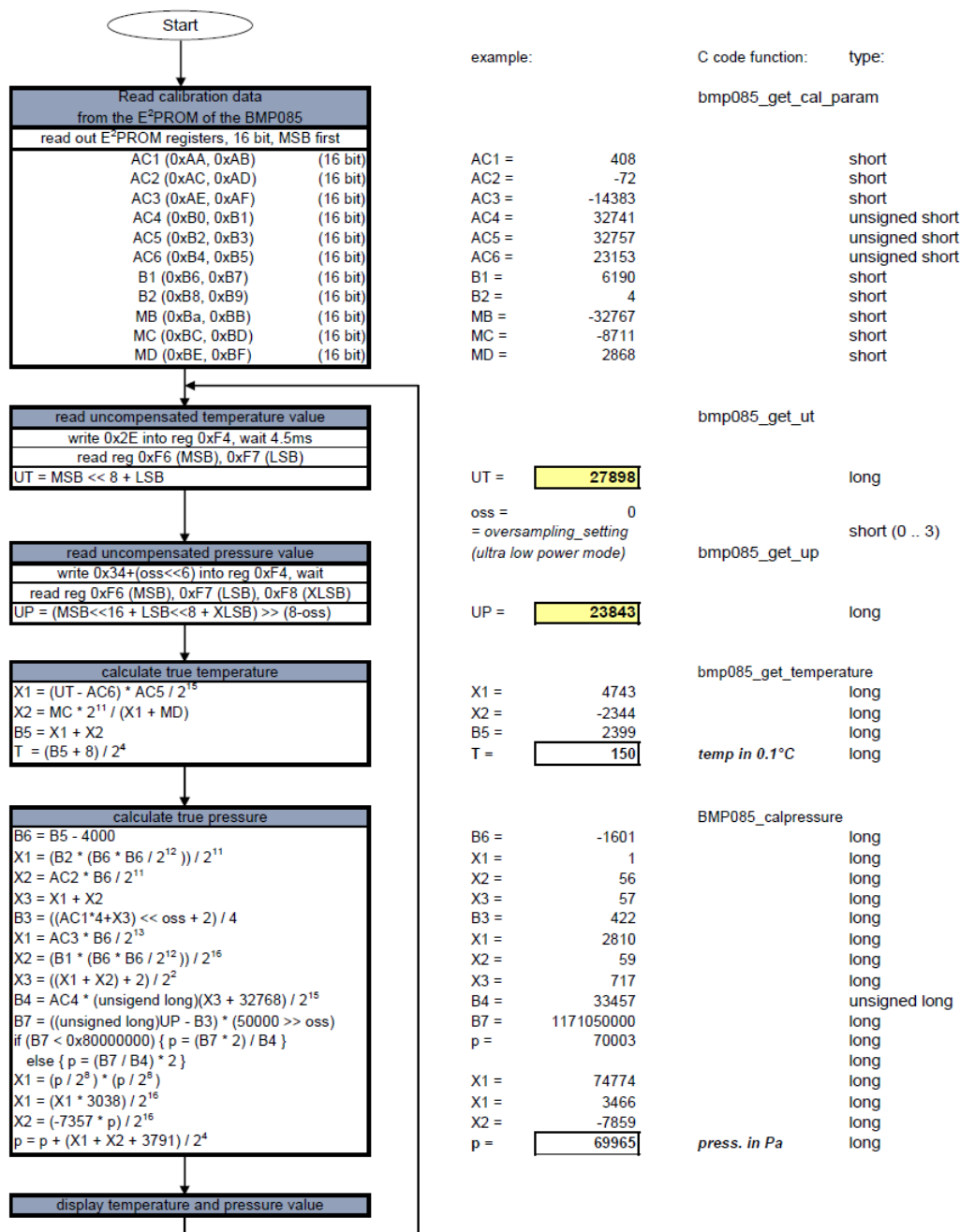
0.06hPa (0,5m), nagy pontosságú módban pedig 0,03hPa (0,25m). A válaszideje 4,5-től 25,2ms. Rendelkezik hőmérséklet kimenettel is. Mérete 20mm X 15mm-er.

A modul egy 176 bites EEPROM-al (Electrically Erasable Programmable Read-Only Memory) rendelkezik, mely tartalmazza a működéshez szükséges kalibrációs adatokat. 11 darab 16 bites részre van felosztva. [8]

|           | BMP085 reg adr |      |
|-----------|----------------|------|
| Parameter | MSB            | LSB  |
| AC1       | 0xAA           | 0xAB |
| AC2       | 0xAC           | 0xAD |
| AC3       | 0xAE           | 0xAF |
| AC4       | 0xB0           | 0xB1 |
| AC5       | 0xB2           | 0xB3 |
| AC6       | 0xB4           | 0xB5 |
| B1        | 0xB6           | 0xB7 |
| B2        | 0xB8           | 0xB9 |
| MB        | 0xBA           | 0xBB |
| MC        | 0xBC           | 0xBD |
| MD        | 0xBE           | 0xBF |

4.4. ábra: BMP-085 szenzor EEPR térképe

A „nyers” hőmérséklet és légnyomás adatok kiolvasásának a szekvencia diagramja a következő ábrán látható:



4.5. ábra: A „nyers” hőmérséklet és légnyomás adatok kiolvasásának a szekvencia diagramja

Először a kalibrációs adatokat kell kiolvasni. Ezek a 0xAA-tól a 0xBF regisztercímen találhatóak. Az egyes kalibrációs paraméterek 16 bitesek, 8 bites MSB és LSB-ből állnak össze. Ezt a kalibrációt csak egyszer kell elvégeznünk. Ezután következik a hőmérsékletmérés. Ahhoz, hogy a nyers adatokat előállítsa a szenzor, ezt valahogy jelezni kell

neki. A szenzor, a 0xF4-es regiszter címére írt adatból tudja eldönteni, hogy most hőmérsékletet vagy pedig légnyomás mérésére van igény. Hőmérséklet mérés esetén a 0x2E értéket kell erre a címre írni. Miután ez megtörtén 4,5ms-ig várni kell. A szenzor a nyers

| Measurement            | Control register value<br>(register address 0xF4) | Max. conversion time<br>[ms] |
|------------------------|---------------------------------------------------|------------------------------|
| Temperature            | 0x2E                                              | 4.5                          |
| Pressure<br>(osrs = 0) | 0x34                                              | 4.5                          |
| Pressure<br>(osrs = 1) | 0x74                                              | 7.5                          |
| Pressure<br>(osrs = 2) | 0xB4                                              | 13.5                         |
| Pressure<br>(osrs = 3) | 0xF4                                              | 25.5                         |

adatokat a 0xF6-0xF7-0xF8 (LSB-MSB-XLSB) regisztercímen állítja elő. Kiolvasás után, a gyártó által meghatározott összefüggések alapján lehet kiszámolni a valós hőmérsékleti adatokat. A légnyomás mérés folyamata hasonlóan néz ki, egy különbséggel. Mint említettem több üzemmódban működtethető. Minél pontosabban kell meghatározni a

légnyomást, annál több időt igényel az szenzor, az adatok előállításához. Ez 4,5ms-tól 25,5 ms-ig tarthat. Ezt fejlesztés során figyelembe kell venni, különben a szenzor téves adatokat fog szolgáltatni! Ennek érdekében ellátták egy hasznos funkcióval. Miután az adatok előálltak, ezt a szenzor az EOC (End Of Conversion) lábán logikai 1-nek megfelelő feszültségszinttel jelzi.

#### 4.4 Sparkfun Razor 9DOF modul

A repülőgép mozgását, orientációváltozását egy Razor 9DOF típusú szenzor fogja mérni. Azért ezt a modult választottam, mert nem csak gyorsulásmérőt és giroszkópot tartalmaz, hanem mágneses érzékelőt (íránytűt) is, illetve bootloader lehetőséggel is el van látva. Az adatokat egy ATmega328-as mikrokontroller állítja elő, melyet soros interfészen keresztül, UART-on továbbít. Több firmware is létezik hozzá, melyet szintén a soros porton



4.7. ábra: Sparkfun Razor 9DOF IMU

keresztül lehet az eszközre feltölteni. Bekapcsolás után, az 1-2 másodpercet igénybevevő kalibrálás után az eszköz használatra kész. A modulon található mikrokontrollernek nincs más feladata, minthogy a szenzorok által biztosított adatok alapján, az egyes tengelyek mentén való elfordulást meghatározza és azt soros porton továbbítja.

## 4.5 GPS

A megfigyeléshez és a stabilizáláshoz szükséges szélességi, hosszúsági és magassági adatokat egy U-blox gyártmányú GPS modul fogja szolgáltatni. A GPS adatokat, értékeket mondatokba szervezve karakterek formájában állítja elő. A legelterjedtebb formátum az NMEA szabvány. Az NMEA egy mozaikszó, amely a National Marine Electronics Association nevéhez fűződik.

Minden mondat elején szerepel 2 karakter, amely az üzenetet továbbító berendezés jellemzője, ez a GPS vevők esetén a „GP” karakter-kombináció. Az egyedileg meghatározott sorok az eszköz gyártójára vonatkozó információt tartalmaznak, ezek első karaktere „P” és a következő 3 karakter utal az eszközt gyártó cégre. Minden mondat kezdő karaktere a '\$' és utolsó karaktere az un. CR/LF (Carriage Return/ Line Feed), és a sorok maximális hossza 80 karaktert a befejező karakter nélkül. Az egy sorban szereplő, különböző üzenetek „,” karakterrel vannak elválasztva. Minden mondat végét egy un. 'checksum' karakter szerepel. Az NMEA üzenetek minden serial port esetében az RS232 protokollt használva elérhetőek. Az adattovábbítás sebesség általában 4800 bps (bit per second rate). Ez az érték azt jelenti, hogy a vevő 480 karaktert tud elküldeni egy másodperc alatt, ami 6 mondatnak felel meg. A mondatokat felépítő adatok 8 bitesek. Minden mondat egy sort jelent, és egyedi azonosítóval rendelkezik.

Nekem az úgynevezett „GGA”, „RMC” és „GSA” mondatra lesz szükségem. A „GGA” mondatból a tengerszint feletti magasság, földrajzi szélesség és hosszúságra, az „RMC” mondatból a sebesség és haladási irány, „GSA” mondatból pedig a 3D észlelés állapotára. A mondatok felépítése:

**\$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,\*47**

- **GGA** - GPS észlelési adatok
- **123519** - észlelési időpont (12:35:19 UTC)
- **4807.038,N** - földrajzi szélesség (48° 07.038', É)
- **01131.000,E** - földrajzi hosszúság (11° 31.000', K)
- **1** - Észlelés típusa (jóság):
  - o **0** - érvénytelen
  - o **1** - GPS észlelés (SPS)
  - o **2** - DGPS észlelés
  - o **3** - PPS észlelés

- **4** - RTK észlelés
- **5** - Float (lebegő) RTK
- **6** - értékelt (számítás nélkül)
- **7** - manuális bevitel
- **8** - szimuláció
- **08** - műholdak száma
- **0.9** - HDOP
- **545.4,M** - tengerszint feletti magasság [m]
- **46.9,M** - geoid-ellipszoid (WGS-84) távolság
- **(üres karakter)** - az utolsó DGPS frissítés óta eltelt idő [s]
- **(üres karakter)** - DGPS állomás ID (azonosító)
- **\*47** checksum adat (minden esetben \* az első karakter)

**\$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W\*6A**

- **RMC** - Javasolt minimális mondat C
- **123519** - észlelés ideje (12:35:19 UTC)
- **A** - Státusz (A=aktív vagy V=érvénytelen)
- **4807.038,N** - földrajzi szélesség (48° 07.038', É)
- **01131.000,E** - földrajzi hosszúság (11° 31.000', K)
- **022.4** - Sebesség (tengeri mérföld)
- **084.4** - Irányszög [°] (True)
- **230394** - Dátum – 1994.03.23
- **003.1,W** - mágneses tér változása
- **\*6A** - checksum

**\$GPGSA,A,3,04,05,,09,12,,,24,,,,,2.5,1.3,2.1\*39**

- **GSA** - műhold-státusz
- **A** - Automatikus 2D vagy 3D észlelés (M = manuális)
- **3** - 3D észlelés – lehetséges értékek:
  - 1 = nincs észlelés

- 2 = 2D észlelés
- 3 = 3D észlelés
- **04,05...** - az észlelésben részt vevő műholdak PRN adatai
- **2.5** - PDOP
- **1.3** - HDOP
- **2.1** - VDOP
- **\*39** – checksum

## 4.6 Kommunikációs protokollok

### 4.6.1 I2C busz

Mivel a BMP085-ös légnyomásmérő I2C buszon keresztül kommunikál, ezért erről is ejtek pár szót.

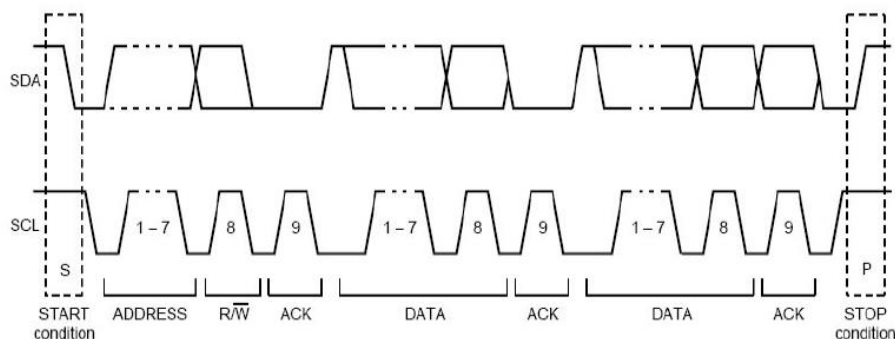
A Philips fejlesztette ki. Egy egyszerű kétirányú, kétvezetékes buszrendszer. A buszt Inter IC-nek, vagy csak röviden I2C-busznak nevezik. Minden I2C busszal kompatibilis eszköz tartalmaz egy on-chip interfészt, ami lehetővé teszi a több eszközzel való közvetlen kommunikációt. Csak két buszvezeték szükséges a működéshez, egy soros adatvonal (SDA) és egy soros órajel (SCL). Minden eszköz programból címezhető egy egyedi címmel. A kapcsolatban résztvevő egységek között master és slave kapcsolat áll fent. Soros, 8-bites kétirányú adatforgalommal rendelkezik, melynek sebessége normál üzemmódban 100kbit/s gyors üzemmódban pedig 400kbit/s. A chip szűrőkkel is el van látva, mely az adatvonalon lévő zavarokat szűri ki. Az egy buszra csatlakoztatható eszközök számát csak a busz kapacitása korlátozza, ami maximum 400 pF lehet. Az adatátvitel a master és a slave eszköz között a következő két módon történhet. [10]

1. Az **A** eszköz adatot akar küldeni **B**-nek:
  - **A** (master) megcímzi **B**-t (slave).
  - Az **A** adatokat küld **B**-nek (**B** a fogadó).
  - Az **A** befejezi az átvitelt.
2. Az **A** adatot akar fogadni **B**-től:
  - **A** (master) megcímzi **B**-t (slave).
  - **A** adatokat fogad a **B**-től.
  - **A** befejezi az átvitelt.

Mindkét esetben a master (A) generálja az időzítéseket és fejezi be az átvitelt. Azonban, előfordulhat az, hogy egy buszon egyszerre több master eszköz is jelen van. Ebben az esetben bekövetkezhet az az eset, hogy egyszerre több master is kezdeményez adatátvitelt. Erre egy arbitrációs eljárást dolgoztak ki. Ez az eljárás egy *ÉS* kapcsolaton alapszik. Ha kettő vagy több master akar adatot küldeni, akkor az első olyan master elveszti az arbitrációt, amelyik 1-et akar küldeni, miközben a többi eszköz 0-át.

Az SCL és az SDA vezeték egy felhúzó ellenálláson keresztül pozitív tápfeszültségre van kötve, azaz ha a busz szabad, akkor mind a két vezeték logikai magas szinten van. A logikai 0 és a logikai 1 feszültségszintek az eszközök kapcsolt tápfeszültségtől függ. Minden egyes bit átvitelhez egy órajel generálódik. Átvitelkor az adatnak stabilnak kell lennie, miközben az SCL magas állapotban van. Az adat változás csak akkor következhet be, ha az SCL logikai nulla értéket vesz fel. Minden adatátvitel egy START jellel kezdődik és egy STOP jellel fejeződik be. START esetén, az SCL logikai 1-es szinten van, miközben az SDA logikai 1-ből logikai 0-ra változik. A STOP bit is hasonló módon áll elő. Miközben az SCL logikai 1, addig az SDA logikai 0-ból logikai 1-re változik.

Az adatok átvitele 8 bitenként történik. Minden byte-ot egy nyugtázás követi. Az adatátvitel az MSB-vel kezdődik és az LSB-vel fejeződik be. (4.7-es ábra)



4.8. ábra: I2C adatátvitel

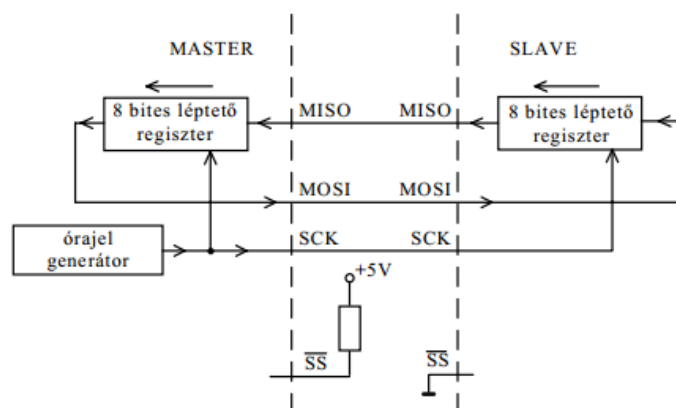
Ha a vevő nem képes fogadni a következő byte-ot, addig az órajel vezetéket (SCL) alacsony szinten tarthatja. Általában a megcímzett eszköz köteles minden egyes adatot ACK-val nyugtázni. Ha a slave nyugtázza a slave címet, de valamiért nem képes adatot fogadni, akkor az első byte átvitele után a slave azzal jelzi a hibát, hogy nem küld nyugtázást, így a master megszakítja az átvitelt. Ha a slave küld adatot a masternek, akkor a masternek kötelessége az utolsó byte előtt jeleznie a slave-nek, hogy a fogadott byte után nem kér többet. Ezután a slave szabaddá teszi az adatvezetéket, így a master egy STOP jelet generál.

Az eszközök címzése hét bites címekkel történik (4.7-es ábra). A START jel generálása után (S), a slave eszköz címe kerül kiküldésre. Ez hét bit hosszú, kiegészítve egy nyolcadik bittel. A nyolcadik bit egy jelző bit, ami az írás vagy olvasás szándékát határozza meg (R/W). Nulla írást az egyes pedig olvasást jelent. Kommunikáció közben, a master gondoskodik a STOP jelek generálásáról. Azonban, ha a master az egyik eszközzel befejezi a kommunikációt, viszont egy másik eszközzel folytatni szeretné a buszon, nem kell neki STOP jelet generálnia, hanem közvetlen egy START bit generálásával ezt jelezni tudja.

## 4.6.2 SPI - Serial Peripheral Interface

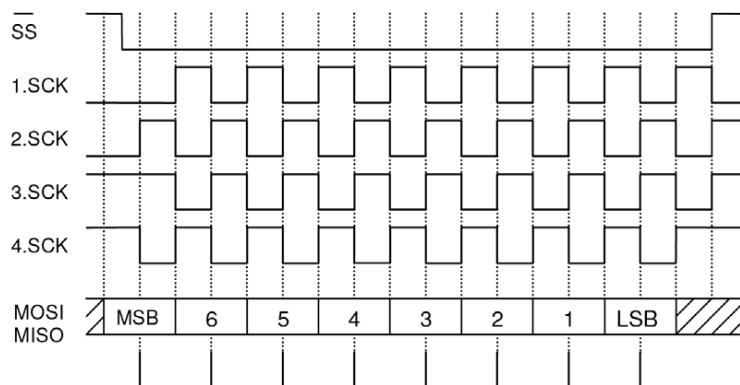
Az SPI busz egy nagysebességű kétirányú soros kommunikációt valósít meg két eszköz között. Az SPI alkalmas egy CPU és kiegészítő áramkörök összekapcsolására, illetve több processzor összekapcsolására is. Ezt a kapcsolatot a busz gyors sebessége teszi lehetővé. 2Mhz-től akár 4-5Mhz is elérhető. Szintén, mint az I2C esetében itt is az eszközök között slave és master viszony áll fent. Egy master-hez több slave eszköz kapcsolható, azonban azt az eszközt, mellyel kommunikálni szeretne, egy külön erre a célra kialakított Slave Select vezetéken keresztül hardveresen kell kiválasztani. Ennek a hátránya az, hogy minden egyes slave eszközhöz, külön adatvezeték tartozik, és ezeknek a vezetékeknek a kiválasztásához további kiegészítő hardverek szükségesek. A szabvány a busz kialakításához négy vezetékkel ír elő:

- MOSI (Master Out Slave In)
- MISO (Master In Slave Out)
- SCK (Serial Clock)
- CS (Slave Select).



4.9. ábra: SPI adatátvitel

Nyolc SCK órajel ciklus valósít meg egy adatátvitelt. Miközben a master kiküld egy adatot a slave-nek (MOSI), a slave is kiléptet egy másikat a master számára (MIO). Ezt a folyamatot egy közös órajel szinkronizálja. Az adatátvitel történhet az órajel felfutó és lefutó élére is. Az órajel jellemzőit úgy, mint polaritás (CPOL) illetve fázis (CPHA) szoftveres úton állíthatóak. Ezzel több vezérlési lehetőség között is választani lehet.



4.10. ábra: Adatátvitel módja

Az 4.10-es ábrán látható, hogy az első és a második állapotban az órajel alacsony, a harmadik és negyedik állapotban pedig magas. Az első és a negyedik esetben az adatátvitel felfutó élre történik, míg a második és a harmadik esetben lefutó élre.

Küldéskor a master a chip select vezetéket alacsony szintre húzza. Ezzel kiválasztja az eszközt, mellyel kommunikálni szeretne. Ezután az utasítás, amit végre szeretnénk hajtani az eszközzel, ezt követi az eszköz regiszter címe, majd pedig kiküldésre kerül az adat.

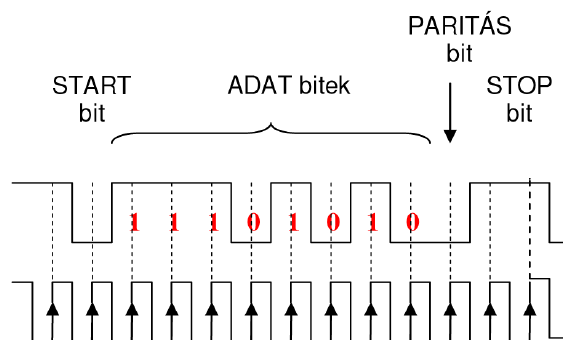
Az SPI egyik legnagyobb hátránya, ami egyben az előnye is az, hogy az I2C-vel szemben, itt nem történik nyugtázás miután kiküldésre került egy adat. Nincs visszajelzés. Viszont épp ezért a kommunikáció sokkal gyorsabban lezajlik. Akár 4-5Mhz-es működési sebességet is elérhetünk. Viszont a valódi előnyét a full duplex kommunikáció jelenti, amely során egy időben lehet küldeni és fogadni is adatokat.

Ez a működési sebesség azért fontos számomra, mert az OSD feliratozásnál ahhoz, hogy az adott sorban elegendő számú pixelt meg tudjak jeleníteni, egy gyors, könnyen kezelhető eszközre lesz szükségem. [11]

### 4.6.3 UART – Universal Asynchronous Receiver/Transmitter

Nagy távolságokra történő adattovábbítás esetén, szempont a kábelezés kialakítása. Az eddig tárgyalt buszoknál a szinkronizálást végző órajelet, egy külön vezetéssel juttatták el az eszközökhöz. Ez nagy távolságok esetén problémákat okozhat, így egy olyan kommunikációs szabványt próbáltak kitalálni, ami ezt a vezetéket megspórolja.

Az adó bármikor kezdeményezhet adatátvitelt. Ehhez azonban egy fontos feltételnek kell megfelelni. Az adónak, és a vevőnek is azonos sebességgel kell működni, különben az elküldött adat értelmezhetetlen lesz. A kommunikáció menetét a 4.11-es ábra szemlélteti.



4.11. ábra: UART kommunikáció menete

Egy START bittel kezdődik, amely egy magas alacsony állapot átmenet. Ezt követi az adat. Nyolcbites csomagokat használnak. A nyolc bitből hét az adatbit, a nyolcadik bit pedig a paritás bit. Az átvitel végét egy STOP bit zárja. A fogadó eszköz, az adatokat a vevő eszköz baud rate<sup>1</sup>-jének megfelelő frekvenciával mintavételezi, de egy fél órajel ciklussal eltolva. Tipikus baud rate érték a 9600bps. Ez azt jelenti, hogy egy másodperc alatt 9600 bitnyi adatot lehet továbbítani, azaz  $1/9600 = 104 \text{ us}$  idő kell egy bit továbbításához. A következő adattranszfer előtt és az utolsó átvitt adat végét jelző STOP bit után három órajel szünetnek kell eltelnie. [12]

1 A baud rate érték megadja az egy másodperc alatt elküldhető bitek számát. Mértékegysége: bps (bit per second).

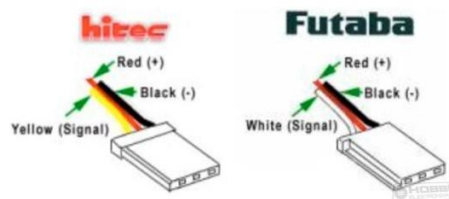
## 5 Aktuátorok

### 5.1 Szervomotorok

A modellezők körében elterjedt szervomotoroknak többféle változata is létezik. A hagyományos méretűtől a micro szervóig, „analógtól” a digitálisig. A kamerát stabilizáló rendszer egyik prototípusához ilyen beavatkozásszerveket fogok használni. A szervomotorok vezérlése úgynevezett PWM (Pulse Width Modulation - Impulzus szélesség modulált) jellel



5.1. ábra: Vitorlacsörlő szervó



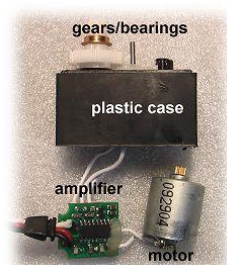
5.2. ábra: Szervomotor csatlakozók

történik, amelyek állandó periódusidejű jelek, ahol az átlagfeszültség beállítása a jel kitöltési tényezőjének változtatásával történik. A servo működtetése három vezetéken keresztül történik (5.1. ábra).

- Táp (piros). Általában 4,8V-6V-ig lehet. Nagyobb feszültség esetén növekszik a servo nyomatéka, de az élettartalma csökken.
- Föld (fekete).
- Vezérlés (sárga vagy fehér).

#### 5.1.1 Szervomotor felépítése

A modellezők által használt RC szervók rengeteg méretben, súlyban és formában elérhetőek, azonban a működési elvük ugyan az. [13]



5.3. ábra: Servó felépítése

A szervo feladata az, hogy a rajta elhelyezett kart, tárcsát a távirányítón található joystick illetve kormány pozíciójának megfelelő helyzetbe állítsa. A következő főbb elemekből áll:

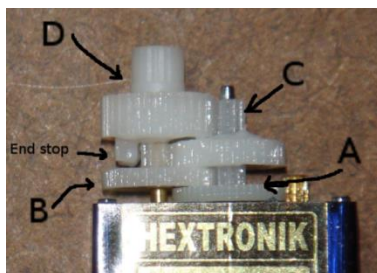
- mechanika, mely a fogaskerekeket jelenti.
- a motor, mely a mozgatót végzi.
- egy potméter, mely a szervo aktuális pozícióját méri.
- egy analóg vagy digitális erősítő, vezérlő.

A legtöbb szervo egy műanyag házba van beleépítve. Ez tartalmazza a fém vagy műanyag fogaskerekeket. A pontossága függ ezeknek a minőségétől. Általában műanyag, de a profibb kivitelezésű szervókban fémfogaskerekek találhatók attól függően, hogy milyen feladatot látnak el. Ahol nagyobb súlyt kell mozgatniuk ott általában fémből készült áttételeket használnak, illetve a hajtótengelyt külön fémcsapággal látják el. Kisebb modellek esetén, a műanyag is tökéletesen megfelel, mert nincs kitéve akkora erőhatásoknak. Háromféle motort használnak. A leggyakoribb a három vagy ötpólusú kefések motor. Előnye az ára, mert viszonylag olcsón hozzá lehet jutni, viszont „nehéz” fém forgórésze van, ezáltal lassabbak is. Ezzel ellentétben, léteznek úgynevezett magnélküli motorok is. Sokkal gyorsabbak, könnyebbek és nagyobb nyomaték leadására képesek. A harmadik típusú a kefe nélküli (brushless) motorok. Mint a neve is mondja, nem tartalmaznak kefét, így nem is fognak elkopni. Hihetetlenül gyorsak és erősek, azonban nagyon drágák.

A szervomotorok vezérlése úgynevezett PWM (Pulse Width Modulation - Impulzus szélesség modulált) jellel történik, amely állandó periódusidejű jel, ahol az átlagfeszültség beállítása a jel kitöltési tényezőjének változtatásával történik. A szervo házában elhelyezett vezérlő feladata az, hogy az impulzus szélességét megmérje és azt egy adott feszültséggé konvertálja. Gyárilag meghatározott a minimum és maximum kitöltési tényező. Ez az impulzus szélesség 1ms-tól 2ms-ig terjedhet. A középpontos 1,5ms körül van. Ezek az értékek természetesen gyártó szerint változhatnak. A szervo tengelyén egy potenciométert helyeztek el, mely a tengellyel együtt elforog, így a szervo helyzetét mérni lehet. Ezeknek a minőségétől függ a mozgás precizitása. A referencia feszültséget felhasználva, egy áramkör összehasonlítja a motor jelenlegi helyzetéhez tartozó feszültséget és az új pozícióhoz tartozó feszültséget és a különbség alapján meghatározza, hogy a motornak jobbra vagy pedig balra kell elfordulnia mindaddig míg a különbség nulla nem lesz. Általában a vezérlőimpulzust nem elég egyszer kiadni, mert nem biztos, hogy a motor ilyen rövid idő alatt képes beállni a kívánt

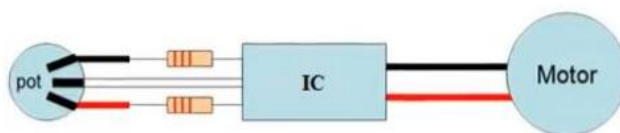
pozícióba, illetve impulzus híján a szervo elernyed. Ezért általában egy 50 vagy 60Hz-es jel formájában történik a vezérlés.

Itt szeretnék külön kitérni a mozgástartományra. A legtöbb szervo mind két irányba 45°-ot képes elmozdulni. Természetesen vannak 60° illetve 90°-os mozgástartománnyal rendelkező motorok is, azonban ezek túl drágák. A kamera stabilizálásához egy 60°-os szervón kisebb mechanikai és elektronikai módosítást végeztem el, a 180°-os elfordulás érdekében.



5.4. ábra: Szervo mechanika

A tengelyen elhelyezett fogaskerék el van látva egy biztosító csapocskával (end stop), mely arra szolgál, hogy a gyárilag megadott szögnél ne tudjon jobban elfordulni. Ezt eltávolítottam, majd az elektronikára került a sor. A gyári potenciométer kb. 120-140°-os elfordulásra képes, ami a motorokban használt áttételek miatt bőven elegendő ahhoz, hogy 180°-ra módosítsuk. Mivel, a referenciafeszültséghez képest végzi a vezérlő a pozicionálást, ezért a következő lépéseket tettem.



5.5. ábra: Szervo elektronikai módosítása

Az IC és a potenciométer közé két ellenállást helyeztem el. Ezzel azt értem el, hogy az gyári paraméterek alapján, ugyan akkora elmozduláshoz kisebb feszültségváltozás tartozik. Azaz a tengelyt tovább kell forgatni, hogy elérje a referencia feszültséget. Ebben az esetben azonban azt tapasztaltam, hogy a megszokott 1-2ms-os PWM jeltartomány balra elcsúszott 0,3-0,4ms-al. De ezt, szoftveres úton le lehet kezelni.

## 6 OSD

### 6.1 Bevezetés

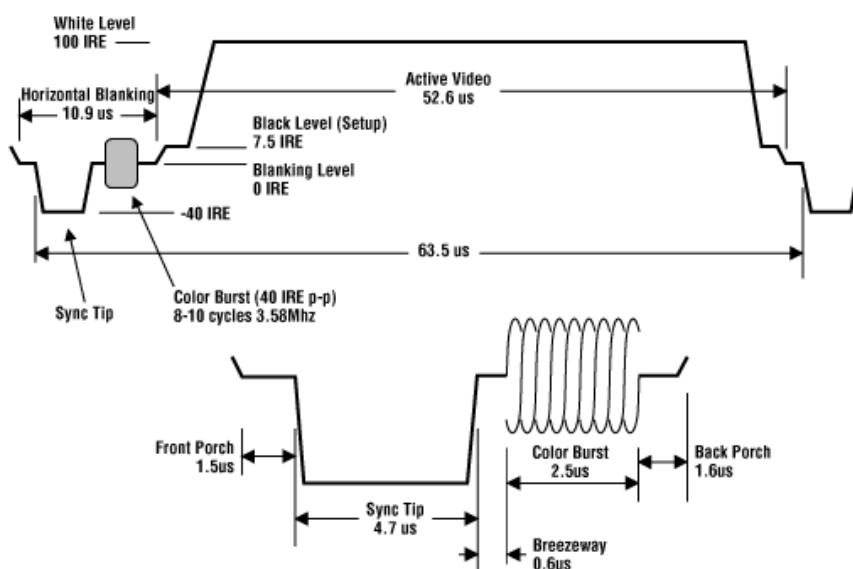
Az internet elterjedésével az ember számára egyre több minden vált elérhetővé. Így manapság egyre több lelkes hobbista áll neki saját maga elkészíteni az általa használt céleszközt. Számos nyílt forráskód és hozzá tartozó kapcsolási rajz található az interneten, illetve számos cél hardver is vásárolható. Én is már létező eszközöket vettem alapul és hoztam létre a saját elképzelésem szerint az OSD feliratozó modult.

### 6.2 A PAL szabvány

A kamera stabilizálásához és a megfigyeléshez szükséges telemetriai adatokat és egyéb információkat egy OSD (On Screen Display) egység fogja megjeleníteni a kamera képén.

A modellezők által előszeretettel használt kamerák nagy része PAL vagy NTSC kompozit videojelet állít elő. Az OSD feliratozást a kamera közvetlen kimeneti jelét felhasználva kell elvégezni, úgy hogy az eredménye egy szintén szabványos kompozit jel legyen. Az egyik legelterjedtebb szabvány a PAL (Phase Alternating Line, fázist váltó sorok). 1967-ben fogadták el és Európa nagy részében ez terjedt el. Magyarországon 1996 óta használják földfelszíni sugárzásnál. Célja az volt, hogy egy olyan összetett kompozit jelet hozzanak létre, ami magába foglalja a videojel összes paraméterét, melyet egy adott rádiófrekvenciás tartományban könnyen sugározni lehet. Tartalmazza a tényleges képi információt, illetve a megjelenítéshez szükséges vízszintes és függőleges szinkronjeleket. A képfrekvenciája 50Hz (25 képkocka / másodperc), ami a 230 voltos hálózati feszültség váltakozó frekvenciájához igazodik. A PAL szabvány szerint sugárzott kép 625 sorból áll (ebből azonban csak 575 sor alkotja magát a képet), illetve a maximális horizontális felbontás 768 képpont. Mint már említettem, a frekvenciája 50 Hz, azonban csak 25 képkocka/másodperc megjelenítésre képes. Ez az úgynevezett „interlaced” (sorváltó) üzemmód miatt van. A lényege, hogy a teljes kép nem egyszerre, hanem meghatározott részekben kerül megjelenítésre, de oly módon, hogy az emberi szem számára ez ne legyen észrevehető. Nem átlapolt (Non-Interlaced) esetén a sorok egymás után kerülnek kirajzolásra. Ezzel szemben interlaced módban először a páratlan, majd a páros sorszámú pixelsorok kirajzolása történik olyan sebességgel, hogy az egy összefüggő képnek látszódjon.

A jel a régi katódsugárcsöves TV-k működéséhez lett kialakítva. A sorokat egy elektronsugárral végigpásztázzuk, majd a sor végén ezt a sugarat a következő sor kezdőpozíciójához kell irányítani. A sugár visszatérítéséhez kell egy holtidő, ami alatt az elektronsugár kioltódik, mert különben téves képi információ fog megjelenni. Ez azonban nagyon pontos, precíz időzítéseket kíván. A következő ábrán egy PAL szabványú videojel egy sorának az ábráját láthatjuk.

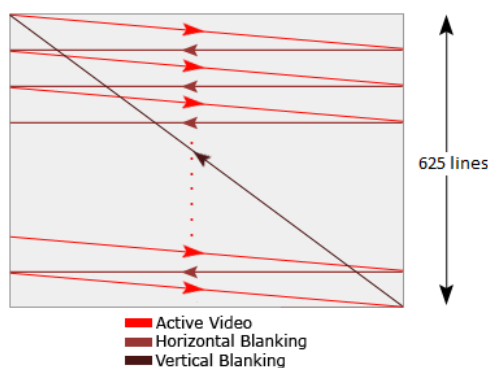


6.1. ábra: Sorszinkron jel

Minden sor, egy horizontális szinkronimpulzussal (H-Sync) kezdődik. Ez jelzi azt, hogy új sor következik, azaz az elektronsugarat vissza kell téríteni a sor elejére. Ez a jel PAL szabvány esetén  $4,7 \pm 0,1 \mu s$  hosszúságú. Ezt követi egy úgynevezett „Back porch”, ami a színes adás sugárzásához szükséges információkat tartalmazza (Color Burst). Időtartama  $5,6 \pm 0,1 \mu s$ . Ezután következik az a jeltartomány, ahol az adott sorhoz tartozó (luminance) fényesség információk jelennek meg. Amplitúdó modulált a jel, tehát a képi információt a jelszint határozza meg. A kompozit videó jel csúcstól csúcsig 1V-os feszültségszintet használ. 0V-tól 0,3V-ig a szinkronjel tartomány található. E fölött kezdődik a képi tartomány. 0,3V jelenti a teljesen fekete, az 1V pedig a teljesen fehér szint. Ez egy  $51,95 \mu s$ -os hosszúságú jel. Nem minden készülék képes megjeleníteni a teljes sort, ezért a tartomány közepén kijelöltek egy  $41,6 \mu s$ -os szakaszt, így ami ezen belül helyezkedik el, minden készüléken látható lesz. Az aktív jeltartomány után következik a Front Porch. Ez egy  $1,65 \pm 0,1 \mu s$  hosszú szakasz, ami időt biztosít a következő horizontális szinkronimpulzus előtt. [14]

Ahhoz, hogy a kirajzolt sorokat mozgóképnek lássa a szem, másodpercenként 25-ször kell frissíteni a teljes képernyőt. PAL szabvány esetén ez azt jelenti, hogy  $1 \text{ másodperc} / 25 \text{ képkocka} / 625 \text{ sor} = 64 \mu\text{s}$  idő jut egy sor kirajzolására.

Miután mind a 625 sort megjelenítésre került, az elektronsugarat vissza kell téríteni a képernyő bal felső sarkába, az első sor elejére.



6.2. ábra: Sor és képmegjelenítés menete

Erre szolgál a vertikális szinkronjel (V-Sync). Ez a művelet sokkal több időt vesz igénybe, mint a horizontális szinkronizáció. Amíg az elektronsugár visszatér a kezdő pozícióba, ki van kapcsolva. Ezt az időtartamot fel lehet használni további információk továbbítására (pl. teletext).

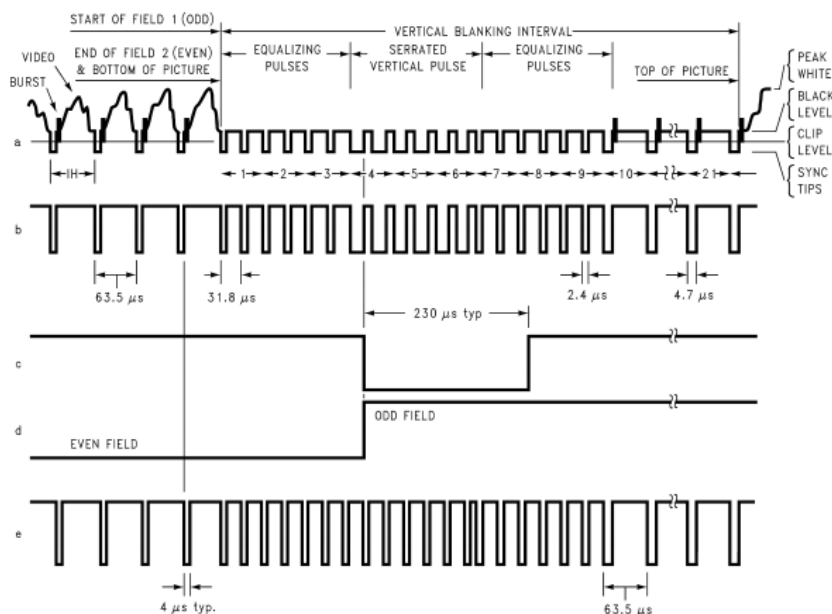


FIGURE 1. (a) Composite Video; (b) Composite Sync; (c) Vertical Output Pulse; (d) Odd/Even Field Index; (e) Burst Gate/Back Porch Clamp

6.3. ábra: LM1881 IC által leválasztott szinkronimpulzusok

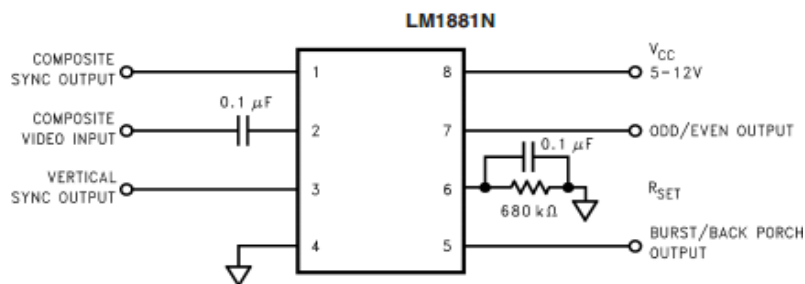
Az ábra (a) felső részén látható maga a kompozit videojel, az alsó (b) részén pedig a leválasztott szinkronimpulzusok. A horizontális szinkron impulzushossz  $63,5\mu\text{s}$ , míg a vertikális ennek a fele  $31,8\mu\text{s}$ . A képváltást leíró impulzussorozatot három részre bontották. Két kiegyenlítő impulzusra (pre- és post-equalizing pulses) és a tényleges képváltást leíró (vertical pulse) impulzus sorozatra. PAL szabványú non-interlaced üzemmód esetén ez hat darab 92,5%-os, öt darab 14,9%-os és szintén hat darab 92,5%-os kitöltésű tényezőjű jelből áll.

### 6.3 Szinkronjel leválasztás

A kamera által előállított video kompozit jel módosításához szükség van a szinkron jelekre. A sorok kirajzolásához nélkülözhetetlen a horizontális szinkronjel. Ennek ismeretében, a PAL szabványnak megfelelő vízszintes felbontást (725 pixel) kihasználásához, a pixelfrekvenciának megfelelő sebességgel kell a kirajzolást elkezdeni. Egy sorban  $\sim 52\mu\text{s}$  idő áll rendelkezésre a hasznos képi információ megjelenítésére. Ez azt jelenti, hogy  $52\mu\text{s} / 725\text{pixel} = 71\text{ns}$  idő jut egy pixelre, azaz a pixelfrekvencia  $1/71 \approx 14\text{MHz}$ . Ez a maximális felbontás kihasználásához szükséges.

A vertikális szinkronjel biztosítja azt, hogy a sorok ne csússzanak el egymáshoz képest. Továbbá interlace üzemmód esetén szükség lehet arra is, hogy az adott képkockán belül a páros vagy pedig a páratlan (fenti ábra, d) sorok vannak éppen megjelenítés alatt.

A szinkronjelek leválasztására több lehetőség is van, azonban léteznek kifejezetten erre a célra gyártott IC-k is. Ilyen a Texas Instruments LM 1881-es típusú Video Sync Separator IC-je. A bemenetére kötött videojelből, komparátorok és számlálók segítségével előállítja a H-sync, V-Sync, Odd/Even jeleket. A gyárilag beállított komparálási szinteknek megfelelően a képi információt, aminek a jelszintje  $0,3\text{V}$ -nál nagyobb egyszerűen levágja, a szinkronjelek pedig a megfelelő kimeneti lábakon megjelennek.



6.4. ábra: LM1881 gyári kapcsolási rajza

A mintakapcsolás alapján, az IC 5-12V-os tápfeszültséggel működik, és digitális jelet állít elő. Ez a kialakítás lehetővé teszi, hogy közvetlenül, bármilyen mikrokontrollerhez hozzá lehessen kapcsolni.

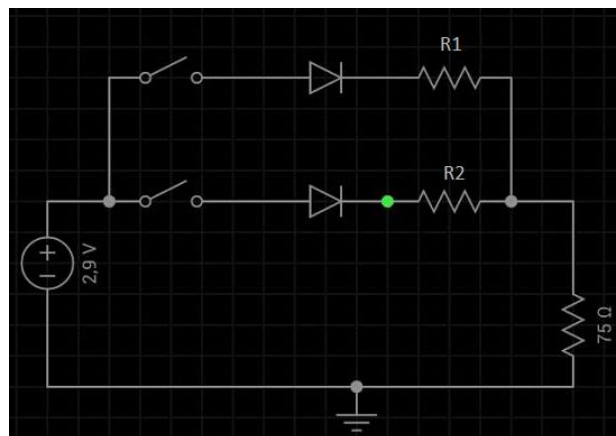
## 6.4 Mikrokontroller által történő rajzolás

Az egyes sorok kirajzolására  $\sim 52\mu s$  idő áll rendelkezésre. Ahhoz hogy a maximális soronkénti pixelszám kihasználásra kerüljön, az egyes pixeleket  $\sim 71ns$  alatt kell megjeleníteni. Ehhez, egy elég gyors működési sebességgel bíró eszközre lesz szükségem. Az általam használt mikrokontrollerbe különböző perifériák kaptak helyet. Ilyen az I2C, SPI, USART. Fent már részleteztem az egyes buszoknak a tulajdonságait. Ezek alapján könnyen belátható, hogy csak az SPI illetve az USART felel meg az elvárásaimnak. Az én választásom az SPI-ra esett a sebessége miatt. A működése egyszerű. A kiküldendő adatot az SPI DR-be (Data Register) kell másolni. Ez egy 8 bites shiftregiszter, melyből az órajelnek megfelelően lépnek ki az adatokat a fizikai lábra. Logikai egy jelenti majd a fehér pixel a logikai nulla pedig a fekete pixelt. További előnyt jelent az, hogy az STM processzor rendelkezik DMA-val (Direct Memory Access) is. Így elegendő a kiírandó bájtokat a memóriában előkészíteni. A DMA-nak meg lehet adni azt, hogy melyik címtől kezdve hány bájtot kell kiküldeni SPI porton keresztül. Ez azért fontos, mert így a kiküldés nem foglal CPU időt, tehát további számításokkal foglalkozhat.

## 6.5 A megvalósítás

### 6.5.1 4 bites D/A átalakító

Léteznek IC-be integrált digitál-analóg átalakítók, de a feladatot tökéletesen elvégzi egy ellenállásosztóból álló 4 bites konverter is. A kapcsolás a 6.5-ös ábrán látható.



6.5. ábra: Ellenállásosztó

A kapcsolók a mikrokontroller egyes lábainak felelnek meg. A lábakon logikai 1 esetén 3.3V logikai 0 esetén pedig 0 volt jelenik meg. Itt megjegyezném, hogy a mikroprocesszor gyári paraméterei alapján 3,3V-nak felel meg a logikai 1, viszont az áramkör megépítése után, abba a problémába ütköztem, hogy az ellenállásosztó által kiszámolt értékek és a ténylegesen mért értékek nem egyeztek meg. A probléma oka az volt, hogy 3,3V helyett csak 2,9V-os feszültségszint jelent meg a processzor lábán a várthoz képest. Így a kapcsolást már a helyes értékkel tudtam újra megtervezni. Az ábrán látható két dióda, a mikro vezérlő egyes lábainak a leválasztását biztosítják. Az 1N4148-as diódát választottam, mely kellőképp gyors a feladat elvégzéséhez. Ennek a diódának a nyitóirányú küszöbfeszültsége 0,6V. A képen látható 75ohmos ellenállás, a vevőkészülék hullámimpedancián történő lezárására szolgál.

- 0,3V-os kimeneti jel (fekete):

$$U_{ki} = 0,3V = 2,9V - 0,6V * \frac{75\Omega}{R1 + 75\Omega} \rightarrow R1 = 500\Omega$$

- 1V-os kimeneti jel (fehér):

$$U_{ki} = 1V = 2,9V - 0,6 * \frac{75\Omega}{\frac{R1 * R2}{R1 + R2} + 75\Omega} \rightarrow R2 = 300\Omega$$

A kapcsolás tesztelésére összeállítottam egy szimulációt, mellyel a helyes kimeneti értékeket ellenőrizni tudtam.



6.6. ábra: Ellenállásosztó szimuláció

A pirossal bekarikázott értékek a kimenetnek felelnek meg. Ha mind a két kapcsoló zárt állapotban van, akkor 978mV a kimeneti feszültség, ami a fehér színű pixelt jelent. Ha a fenti

kapcsoló zárva van az alsó pedig nyitva, akkor a fekete színű pixelnek megfelelő 318mV-os jel jelenik meg a kimeneten.

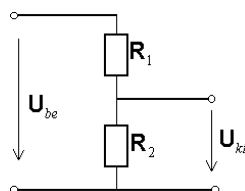
## 6.5.2 A szinkronjel leválasztó áramkör

A horizontális és vertikális szinkronjel leválasztásához egy erre gyártott LM1881-es típusú cél IC-t használok. A hozzá tartozó gyári kapcsolás úgy lett kialakítva, hogy közvetlen bármilyen mikrokontrollerhez hozzá lehessen csatlakoztatni. Azonban egyes mikroprocesszorok nem 5V-os feszültséggel működnek, hanem 3,3V-al. A szinkronszeperátor IC tápfeszültsége 5-12V-ig terjed, ami azt jelenti, hogy a kimenetek is ennek megfelelően fognak előállni. Az STM processzor bemeneteire maximum 3,3V-os feszültséget lehet kapcsolni. A gyártók azonban úgy tervezik meg őket, hogy ha esetleg az előírt jelszintnél magasabb feszültséget kötnek rá a bemeneti lábra, akkor azt egy bizonyos szintig tolerálja. Ez ebben az esetben 5V-ot jelent, ami megfelel a szinkronjel leválasztó IC kimeneti jelének. Azonban, hogy ez ne jelenthessen problémát az illesztést feszültségosztók segítségével valósítom meg.

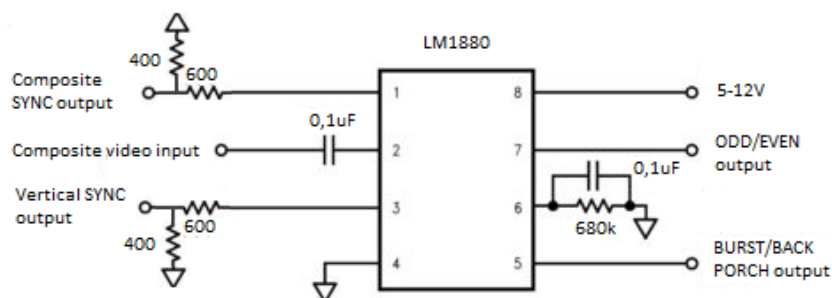
$$U_{ki} = U_{be} * \frac{R_2}{R_1 + R_2} = 5 * \frac{400\Omega}{400\Omega + 600\Omega} = 2V$$

(1)

Mivel az LM1881-es IC minimális tápfeszültsége 5V, ezért a feszültségosztó kimeneti értékét úgy választottam meg, hogy ha a későbbiekben instabilan működne a szinkronjelszeperátor, akkor a tápfeszültséget minden probléma nélkül tovább lehessen növelni 12V-ig. Ebben az esetben a feszültségosztó kimenetén, maximum 5 volt fog megjelenni.

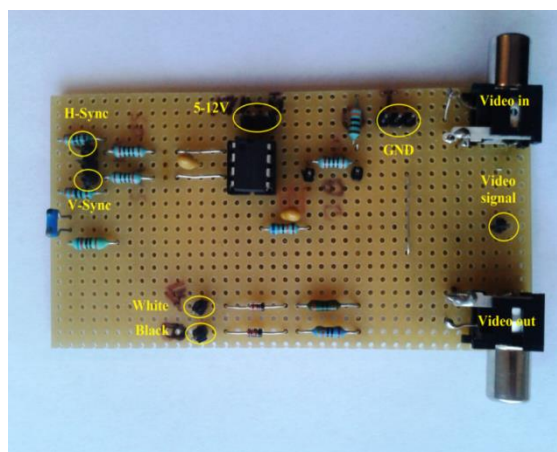


6.7. ábra: Feszültségosztó



6.8. ábra: LM1881 módosított kapcsolás

A megtervezett 4 bites A/D átalakító és a szinkronjel leválasztó IC kapcsolását, egy próbanyákon valósítottam meg, mely a 6.9-es ábrán látható.

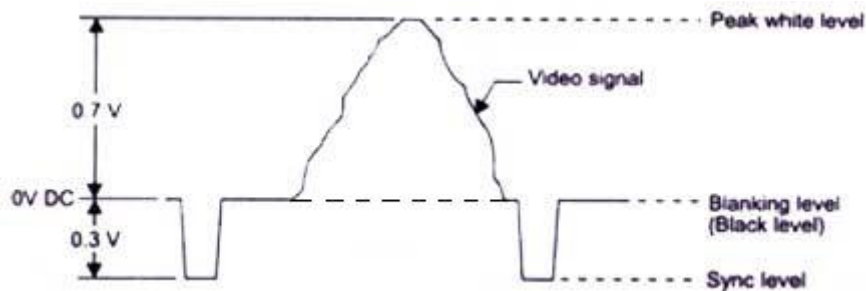


6.9. ábra: Szinkronjel leválasztó modul

A panelon helyet kapott két RCA csatlakozó. Van egy „Video in” és egy „Video out” kimenet. A közte lévő „Video signal” pin arra szolgált, hogy a módosított kompozit videojelet oscilloscope segítségével mérni lehessen. A „H-Sync” és a „V-Sync” kimeneteknél egy-egy feszültségosztó található.

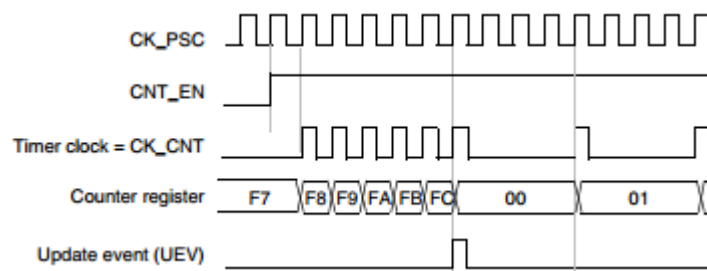
### 6.5.3 Videojel előállítása

A folyamatos információk megjelenítése érdekében biztosítani kell, hogy ha a kamera jele megszűnik továbbra is látni lehessen a telemetriai adatokat. Ehhez a mikrokontroller segítségével is elő kell tudni állítani a szükséges szinkronjeleket. Ha jobban megvizsgáljuk az LM 1881-es szeparátor által leválasztó jeleket (6.10-es ábra), észrevehetjük, hogy a horizontális szinkron impulzus egy PWM jelnek felel meg.



6.10. ábra: Horizontális szinkronjel

A PAL szabványnak megfelelően ez egy 64 $\mu$ s-hoggyúságú jel. 4,7 $\mu$ s-ig alacsony, 59,3 $\mu$ s-ig pedig magas állapotot vesz fel. Az STM mikroprocesszor több TIMER-el is rendelkezik, melyeket PWM üzemmódban is lehet használni. A jel előállításához a TIM4-es időzítőt használok. Két fontos regiszterére lesz szükségem. A processzor 168Mhz-es frekvencián működik, ami azt jelenti, hogy a TIM4 counter regisztere,  $1/168000000=5,95\text{ns}$ -alatt inkrementálódik eggyel. Azaz a  $64\mu\text{s}/5,95\text{ns}=10752$ . Az időzítő rendelkezik egy auto-relod regiszterrel, melynek az értéke megegyezik a jel periódusidejével. Inkrementáláskor, ha ennek a regiszter értékét eléri a counter regiszter, akkor lenullázódik és kezdődik előlről a számolás. A horizontális szinkronjelnak, 4,7 $\mu$ s-ig kell alacsonynak lennie. Ez azt jelenti, hogy egy  $59,3/64=92\%$  kitöltési tényezővel rendelkező PWM jelről van szó. Invertáló üzemmódban, alap állapotban a jel alacsony értéket vesz fel. Azt, hogy mikor váltson át magas állapotba a CCR1-es regiszter segítségével lehet megadni.  $4,7\mu\text{s}/5,95\text{ns}=789$ . Ha ezt az értéket eléri a számláló, a jel logikai 0-ból logikai 1-re vált át.

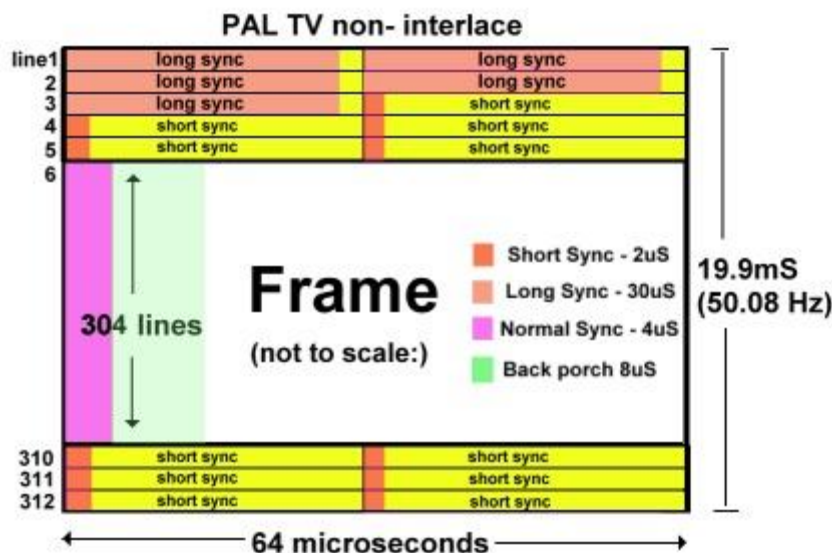


6.11. ábra: TIMER4 idődiagramja

Minden counter register resetelést egy update event megszakítás jelez (6.11-es ábra). Ez a következő célt fogja szolgálni. Minden horizontális szinkronimpulzus után, amikor a számlálás újra kezdődik, ki kell várni a 4,7  $\mu$ s-os új sort jelző impulzust, illetve az utána következő 5,6 $\mu$ s-os „Back porch-ot”. Mivel nem minden készülék képes megjeleníteni a teljes sort, ezért  $4,7+5,6=10,3\mu\text{s}$  helyett  $\sim 14\text{-}15\mu\text{s}$  ideig várok, a pixelek megjelenítésével. A PA7-es lábon előállított PWM jelet az általam összerakott panelen található „black” (fenti ábra) pinre

kötöm. Így amikor magas állapotban van, az ellenállás osztó kimenetén  $\sim 0,3V$ -os feszültség fog megjelenni, azaz a teljes képernyő fekete lesz.

Ezek után szükség van a vertikális szinkronimpulzusok előállítására is. Ehhez viszont nyomon kell követni, hogy éppen hányadik sor kirajzolásánál tart. Ehhez egy sorszámlálót vezettem be. [14]



6.12. ábra: Horizontális és vertikális szinkronjelek

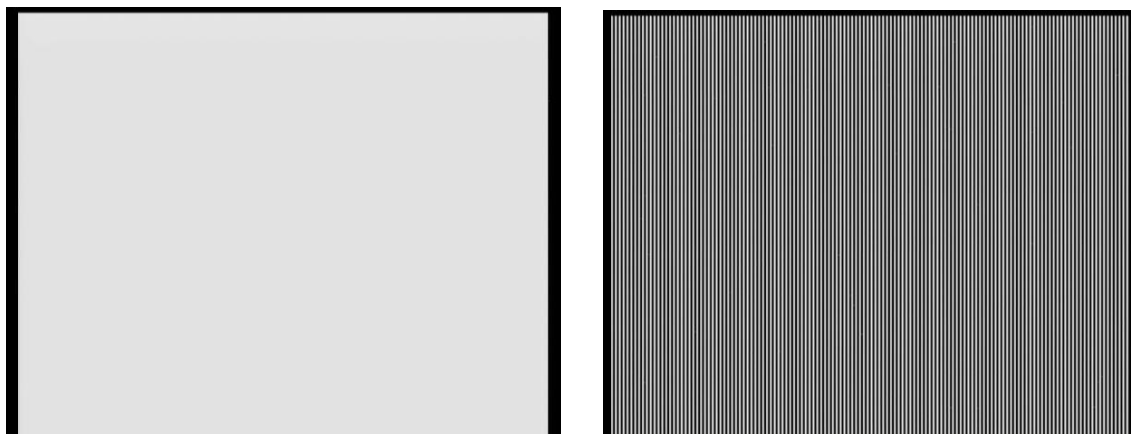
Non-interlace üzemmódban nincsenek megkülönböztetve a páratlan és páros sorok, a kompozit jel némileg leegyszerűsödik, viszont a felhasználható sorok száma is lecsökken. Ebben az esetben, a kompozit videojel 6 Pre-equalizing, 5 Vertical, 5 Post-equalizing pulzusból áll. A sorszámlálót fogom használni ezeknek a nyilvántartására. 0-4-ig egy „long sync” impulzus kell kiadni. Ennek a hossza 64us sorszinkron jelnek a fele, azonban a kitöltési tényezője nem változik. Így, a TIM4 periódus idejét és a kitöltési tényezőjének az értékét is a felére állítom át. Ezután következik 5 db. „short sync” impulzus. Ez a „long sync” impulzussal megegyező periódus idejű, azonban 90% helyett 10% kitöltésű. Minden egyes szinkronimpulzus kiküldésekkor a sorszámlálót növelem. Ha a számláló elérte a 10-et, következhet a sorok megjelenítése. Összesen 304 sor áll rendelkezésre. Az utolsó sornál a számláló értéke 314 lesz. Ezt további 6 „short sync” impulzus követi, majd a sorszámlálót lenullázom és kezdődik előlről a folyamat.

A kirajzoláshoz az SPI1-es portot vettem igénybe. Mivel a szabványnak nem megfelelő módon fogom használni, így a szokásos négy vezeték helyett csak egyet fogok konfigurálni. A processzor master módban fog működni, tehát csak a MOSI (Master Out Slave In) lábat fogom bekötni a „white” pinre (6.9-es ábra). A logikai 0 a fekete a logikai 1

pedig a fehér színnek felel meg. Magas állapot esetén a MOSI kimeneten 2,9V-os feszültség fog megjelenni, az ellenállás osztó keresztül áthaladva pedig ~780mV-os feszültség lesz belőle. Azonban az előírt szabványt nem teljesen lehet figyelmen kívül hagyni. A kommunikációhoz mindenképp ki kell választani a slave eszközt. Mivel itt nincs fogadó eszköz, ezért a processzor SS (Slave Select) lábát szoftveresen alacsony feszültség szintre kell lehúzni.

Ahhoz, hogy minél jobban ki lehessen használni az egy sorban megjelenítendő pixelek számát, az átviteli sebességet (Baud Rate) be kell állítani. Az STM processzor SPI egysége a fő órajelből származtatott frekvenciával működik. Ez az osztó 2-től 256-ig terjedhet. Jelen esetben én 8-al osztom le a fő órajelet, így egy teljes sor esetén 536 látható pixelt tudok generálni.

A kirajzolási folyamat a következő módon zajlik le. A TIMER4 előállítja a 64us-os PWM jelet. Minden jel végén keletkezik egy interrupt mely jelzi, hogy a sor végén van. Ezt az interruptot a TIM4\_IRQHandler() függvény kezeli le. Megszakításon belül a processzor áll, tehát idő szempontjából nem hasznos sok időt igénylő számítást végezni. Így egy újabb TIMER-t felhasználva oldottam meg az időzítést. A TIM4 által kiváltott megszakításban elindítottam a TIMER2-öt ha a sorszámláló értéke elérte a 10-et, kinullázom TIM\_IT\_Update flaget, hogy a processzort felszabadítsam. A TIMER2 úgy lett beállítva, hogy 13us idő elteltével szintén generáljon egy megszakítást, amin belül ténylegesen megtörténik a képi információk kiküldése. Az SPI 8 bites adatcsomagokkal kommunikál, tehát egyszerre 8 pixelnyi információt lehet kiküldeni. Elsőnek az MSB bit léptetődik ki az SPI data regiszteréből, így nem kell külön foglalkozni a pixelek helyes sorrendben való megjelenítésével. Az eredmény, a következő ábrán látható.

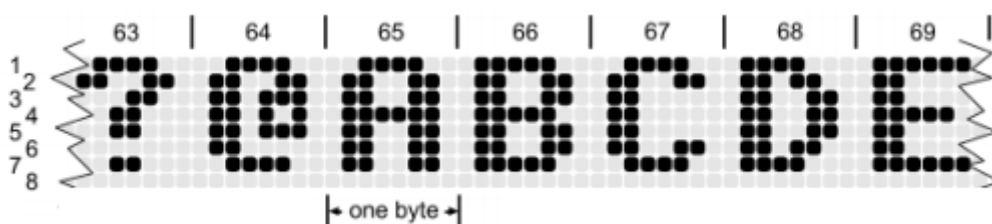


6.13. ábra: Fehér képernyő és fekete-fehér képernyő

Tesztelés során a következő hibával találkoztam. Ha az SPI port data regiszterébe felváltva 01010101-et írtam, a megjelenítés tökéletes volt. Azonban ha csupa fehér pixelt szerettem volna megjeleníteni, azaz 11111111-at írtam a DR regiszterbe, a kép szétszúszott, eltűntek a szinkron jelek. A probléma forrása a következő volt. Miután kiküldtem mind az 536 pixelnyi adatot a DR regiszter nem nullázódott ki. Így amikor a PWM jel alacsony állapotban volt, az SPI kimeneten akkor is megjelent a logikai magas érték, ami azt eredményezte, hogy a 0V-os horizontális jelre ráült a 780mV-os fehér pixelnek megfelelő érték. Azonban PAL szabvány szerint csak 0,3V-nál kisebb jelszint felel meg szinkronimpulzusnak. A megoldás az volt, hogy a sor végén a DR regisztert nullázni kell.

#### 6.5.4 Karakterkészlet, pixel szintű képernyőkezelés

Ahhoz hogy szöveges üzeneteket is meg tudjon jeleníteni a képernyőn, szükség lesz egy karakterkészletre, ami bináris formában tárolva van. Az Arduinohoz készült TvOut library-ben található Font6x8.h fájlt használom fel. A teljes ASCII tábla le van tárolva egy bináris tömbbe. Ennek megfelelően a mérete 8x256 byte.



6.14. ábra: Karakterkódolás

A 6.14-es ábra alapján, minden karakter 8x8=64 bitnyi helyet foglal le. Ennek a szervezési formának van egy nagy előnye. Kirajzoláskor mindig csak egy sort kerül megjelenítésre. Belátható, hogy így egy karakter megjelenítése viszonylag leegyszerűsödik. Tegyük fel, hogy az „A” karaktert akarom kirajzolni. Ekkor a hozzá tartozó ASCII kóddal és az éppen aktuális sorhoz tartozó értékkel és egy ofszettel, a 65. elem 0. sorát ki tudom címezni. Ez az „A” karakter első sorának a megjelenítéséhez tartozó értékeket fogja tartalmazni.

A legtöbb OSD feliratozóval, amit forgalomban lehet kapni, csak karakter szinten lehet feliratozni. Azonban néha szükséges lehet egyéb ábrák megjelenítésére is.



A fenti ábrán az Apache névre hallgató OSD feliratozó eszköz által megjelenített információk láthatóak. A telemetriai és egyéb információkon kívül található egy műhorizont és a kamerakép középpontját jelző kör. A szebb megjelenítés érdekében ehhez pixel szinten kell tudni kezelni a képernyőt.

Egy bitmap-et hoztam létre, ami egy 304 sorból és 69 oszlopból álló uint8\_t típusú tömb. Minden egyes bit egy pixelnek felel meg.

[illegible]

### 6.16. ábra: Bitmap

A tömböt csak bájtanként lehet kezelni, ezért a bitszintű címzést maszkolás segítségével oldottam meg. Tegyük fel, hogy a 0. sor 20 pixelét szeretném módosítani. Először egy egyszerű osztással meghatározom azt, hogy az adott pixel hányadik bájtban található.  $20/8=2$ . Ezt követően egy moduló osztással meghatároztam a bithez tartozó ofszet értéket, azaz a bájtban belüli helyét.  $20 \bmod 8 = 4-1=3$  (mivel az indexelés 0-tól kezdődik). A maszkoláshoz

létrehoztam egy `bit[]` nevű tömböt, melyben az egyes helyi értékekhez tartozó maszk található.

```
uint8_t bit[8] = { 0b10000000, 0b01000000, 0b00100000, 0b00010000,  
0b00001000, 0b00000100, 0b00000010, 0b00000001};
```

Miután rendelkezésre állt mind a két adat a bitszintű címzés leegyszerűsödik. A 0. sor 20. bitjének a módosítása a következőképpen fog kinézni:

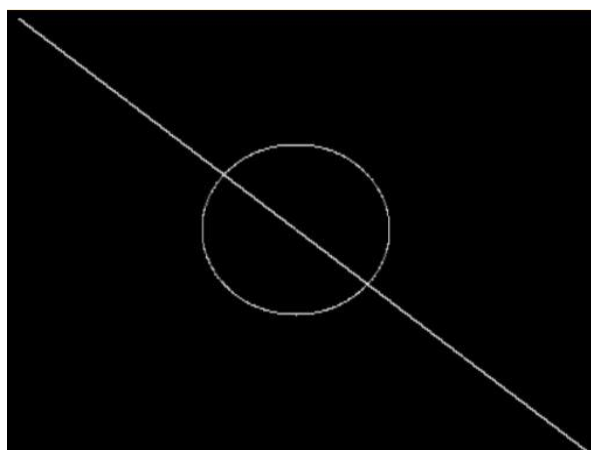
```
bitmap[0][2] |= bit[3];
```

Azaz a bitmap 0. sorában található 2. bájtját össze „OR-lom”<sup>2</sup> a `bit[3]` értékével. Így 0-ról 1-re változik meg a tartalma.

Az SPI porton való kiküldés inentől kezdve a bitmap alapján történik a sorszámláló segítségével. Azonban ügyelni kell rá, hogy a kirajzolás akkor kezdődik, ha a számláló eléri a 11. sort, azaz a tömb címzését a counter-10 értékkel kell kezdeni.

### 6.5.5 Kör és vonal rajzolása

A kör és a vonal rajzolásáért két függvény felel. Kör esetén paraméterként a középpontot és a sugarat kell megadni, vonal esetén pedig a kezdőpont és a végpont koordinátáját.



6.17. ábra: Kör és vonalrajzolás eredménye

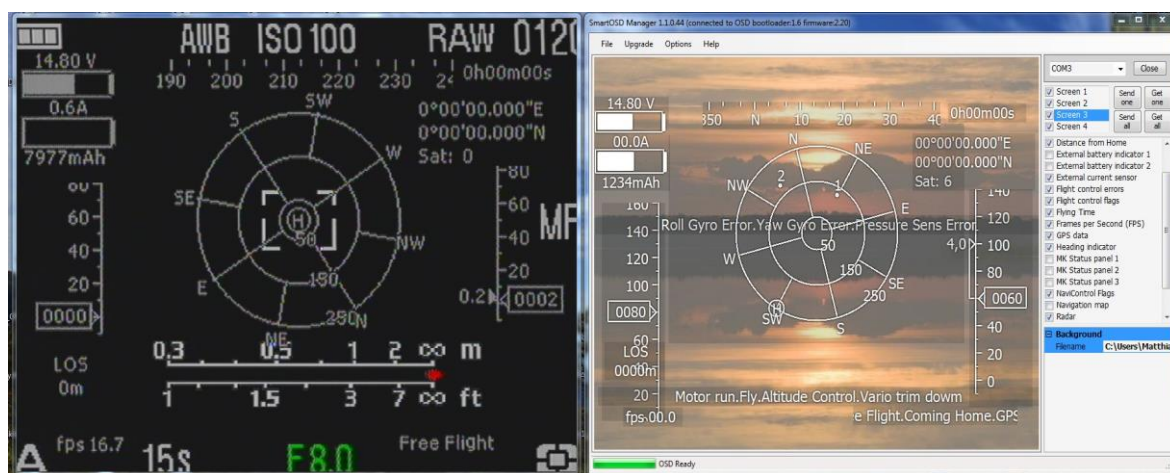
---

<sup>2</sup> Logikai VAGY művelet.

A 6.17-es ábrán látható a pixel szintű rajzolás eredménye. A felbontás miatt a kör némileg torzult, ellipszis formájú lett. Ezt természetesen szoftveres úton kompenzálni lehet.

Innentől kezdve a pixelszintű rajzolásnak köszönhetően, a szabványos karakteres megjelenítés mellett szinte bármilyen alakzatot ki lehet rajzoltatni.

Ilyen a pl. a műhorizont, mely FPV repülés során elengedhetetlen. Az egyes repülési adatokat akár kis diagramokon valós időben is meg lehet jeleníteni. Erre láthatunk példát, a Video Experiementer-nél a 3.6-os ábrán. A 3.7-es ábrán valós időben előállított binarizált képet (fekete-fehér kép) láthatunk. Ezt továbbgondolva, egyszerű térképek megjelenítésére is lehetőség van. A 6.18-as ábrán a Smart OSD névre hallgató modul kameraképe látható. Középen egy egyszerű radar kapott helyet, melyen a közelben található repülő tárgyak elhelyezkedését jeleníti meg.



6.18. ábra: Smart OSD

## 6.5.6 Feliratozás külső videojelre

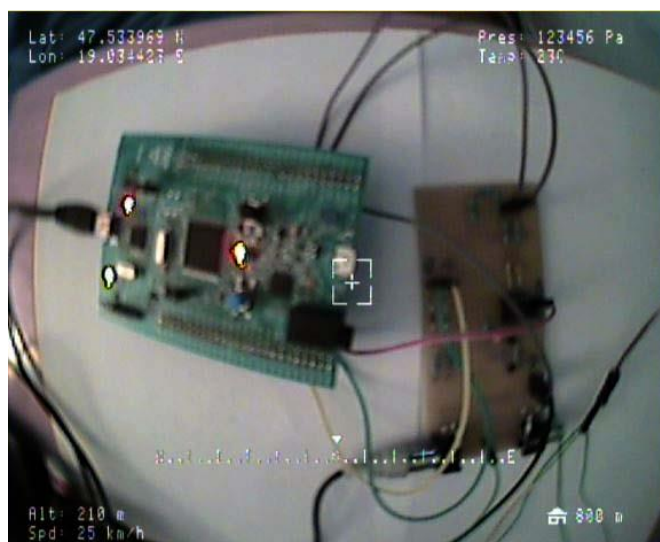
Miután a kompozit videojelet sikerült a mikroprocesszor által előállítanom és módosítanom a megjelenítendő információnak megfelelően, következhet a külső videojel feliratozása. Ehhez a próbanyákra épített LM1881-es szinkronszeparátor által leválasztott horizontális és vertikális szinkronjeleket fogom felhasználni. Az IC két kimenetét a mikroprocesszor két bemeneti lábára kötöttem. Mind a két láb úgy lett konfigurálva, hogy ha a bemeneten 0V-nál nagyobb feszültség jelenik meg, az egy interruptot generál a processzor számára. A komparálási szintet úgy állítottam be, hogy a H-Sync és a V-Sync felfutó élére történjen a megszakítás. Ez azért hasznos, mert így csak az 5,7us-os „Back porch” idejét kell kivárni és utána közvetlenül kezdődhet a feliratozás. A sorszámláló növelése a horizontális szinkron által kiváltott megszakítás esetén történik. Ebben az esetben csak a sorokat számolja,

mivel külön vált a horizontális és a vertikális impulzus. Miután a képernyő aljára ért, a V-Sync-hoz tartozó interrupton belül a sorszámológót kinullázza, és kezdődhet előlről a feliratozás.

Ezen kívül bevezettem egy trigger nevű változót is. Induláskor nem lehet tudni azt, hogy épp hányadik sornál tart a megjelenítés. Ebben az esetben néhány mikro szekundumig nem a sornak megfelelő adatok kerülnek kiküldésre, azaz elcsúszhat a kép. Ezért az első vertikális szinkronimpulzusig ennek a trigger változónak az értéke nulla lesz, azaz a kirajzolás le van tiltva. A külső jelre való feliratozás és rajzolás eredménye a következő ábrán látható.



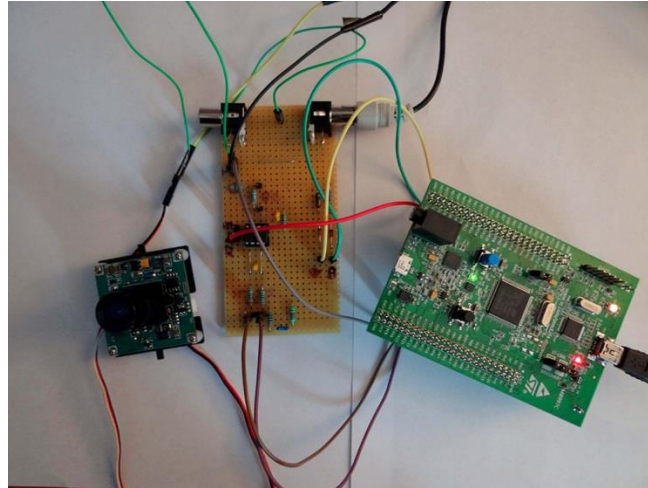
6.19. ábra: Feliratozás külső videojelre eredménye



6.20. ábra: Feliratozás külső videojelre eredménye (2)

A (6.19-es ábra) bal felső sarokban a hosszúsági és szélességi adatok, a jobb felső sarokban a légnyomás és a hőmérséklet kapott helyet. Középen egy kereszt látható, mely a kép közepét

jelzi illetve a megfigyelt földi objektumra fog irányulni. A bal alsó sarokban található a magasság és sebességi adat, középen egy haladási irányt jelző csúszka, jobb oldalon pedig a kiindulási ponttól való távolság.



**6.21. ábra: OSD modul, kamera, mikrokontroller**

A kamera jelét egy Boscam gyártmányú 5,8Ghz-es 200mW-os teljesítményű adó sugározza a földi állomás felé. Nyílt terepen 500m-től 1km-es távolságra tudja továbbítani a jelet. Beépített mikrofont is tartalmaz, így igény szerint hang továbbítására is lehetőség van.



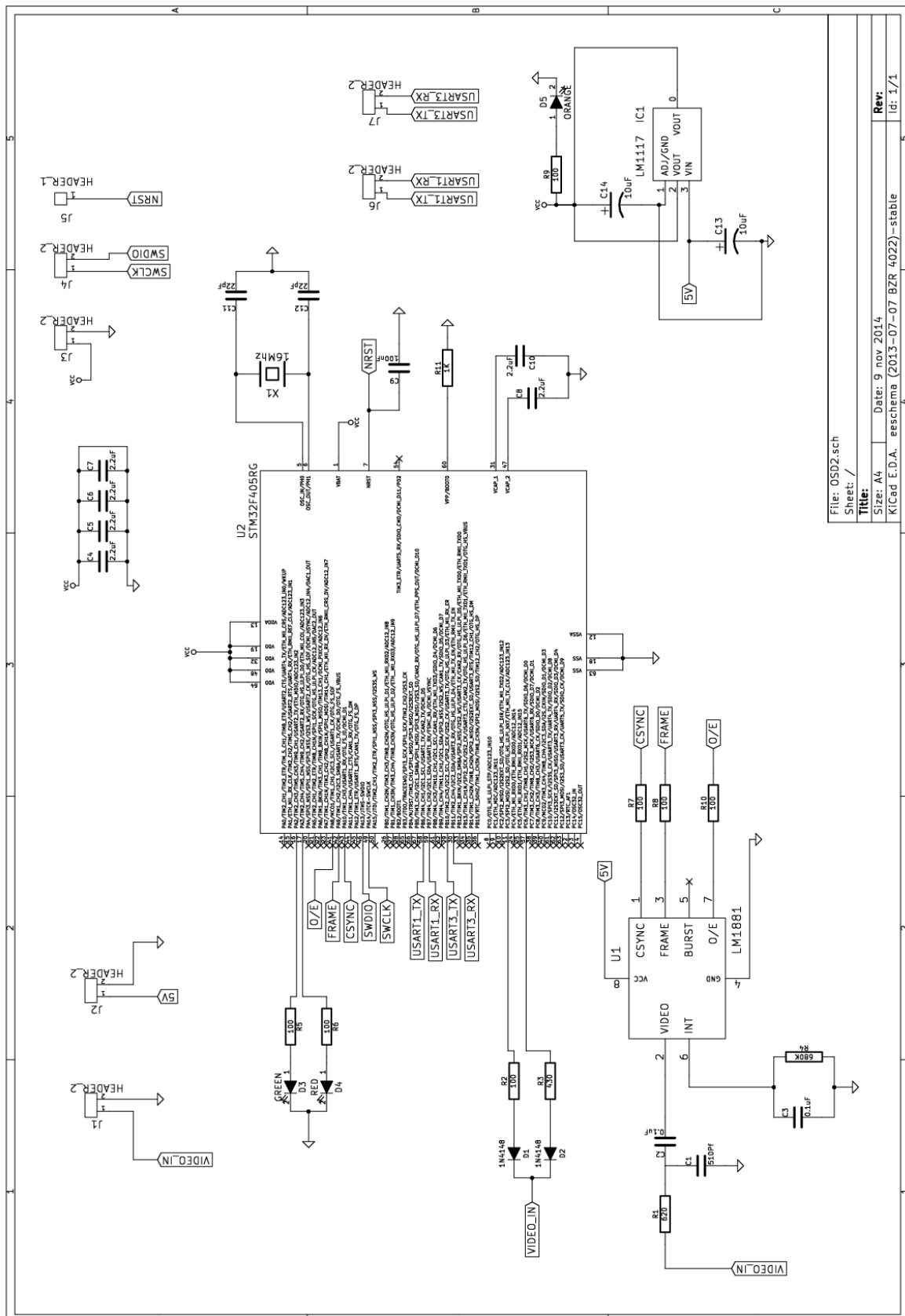
**6.22. ábra: Boscam 5.8Ghz-es 200mW-os adó vevő egység**

A választható csatornák száma nyolc. Tömege FPV (First Person View) repüléshez lett méretezve, 25 gramm. 7-12V-os tápfeszültséggel működik. Két cserélhető RP-SMA aljzattal ellátott bot antenna tartozik hozzá.

## 6.6 OSD paneltervezés

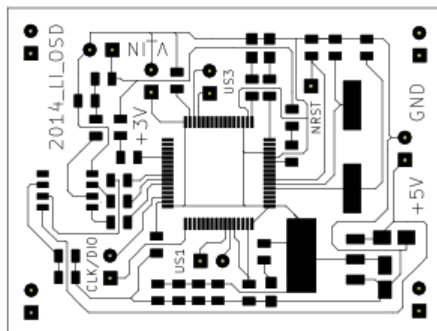
Mivel a próbapanelon megépített kapcsolás a tesztelések során működőképesnek bizonyult, nekiálltam egy saját OSD feliratozó modul tervezésének. A kapcsolási rajz a 6.22-es ábrán látható. A panelon helyet kapott, egy STM32f405RGT6W típusú 168Mhz-es ARM processzor. 1MB program memóriával és 192KB RAM-al rendelkezik. Ezen kívül elhelyezésre került a videó szinkronjelek leválasztásához szükséges szinkronjel szeparátor IC. A jelleválasztó és a processzor tápfeszültsége különbözik, ezért egy feszültségszabályzó IC-t is alkalmaznom kellett.

Mivel, egy különálló modult terveztem meg, ezért két UART port is kivezetésre került. Az egyik porton a feliratozni kívánt adatokat fogadja a processzor, a másik pedig tartalékként, illetve debuggoláshoz lesz fenntartva.



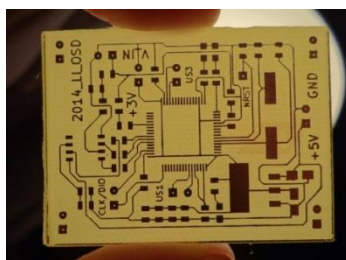
6.23. ábra: Sajáttervezésű OSD modul kapcsolási rajza

A 6.23-as ábrán a nyák terve látható. Tervezéskor a lehető legkisebb modult szerettem volna létrehozni, viszont úgy hogy az egyes alkatrészeket még kényelmesen fel lehessen forrasztani. Maga a nyák 5cm X 3,5cm. Az alkatrészek pedig SMD-0805 típusú tokozásúak. Természetesen léteznek ennél jóval kisebb alkatrészek is, azonban azoknak a forrasztása emberi kézzel már nagy kihívást jelent.



**6.24. ábra: Saját tervezésű OSD modul nyákterve**

A nyáklap transzferfóliás prototípusgyártás technológiával lett elkészítve. Az STM processzor LQFP-64-es tokozású, azaz 64 lábbal rendelkezik, melyek kb. 0.3mm-re vannak egymástól. Az alkalmazott PnP fóliának köszönhetően, ez a vonalvastagság még nem jelentett problémát. A nyákrajz tükörképe erre a fóliára lézernyomtatóval lett kinyomtatva. A lemezre, megtisztítás után a rajzolatot hőkezelés alkalmazásával vittem át. 200-230 fokos hőmérsékleten a fóliától a rajzolat elvált, és átragadt a lemez rézzel bevont oldalára. Ezek után TN 150-es gyorsmarató alkalmazásával marattam ki a nyákot. A 6.24-es ábrán a maratás utáni állapot, a 6.25-ös ábrán pedig a kész modul látható.



**6.25. ábra: Saját tervezésű OSD modul kimaratott nyáklemeze**



**6.26. ábra: Saját tervezésű OSD modul**

## 7 A kamera stabilizátor

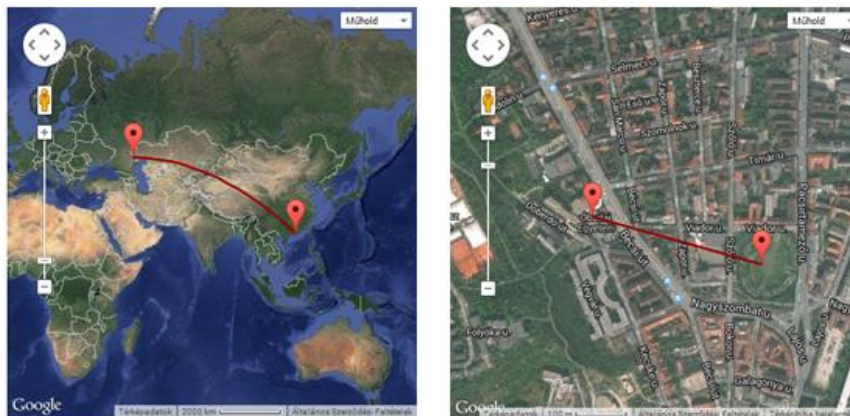
### 7.1 Bevezetés

Dolgozatom második felében a pozíciókövetéshez és a kamera stabilizálásához szükséges számításokat, feladatokat fogom részletezni.

Manapság a legtöbb objektumkövető, vagy pozíciókövető kamerarendszer képfeldolgozás alapján működik. Ehhez azonban nagy számítási kapacitás szükséges. A dolgozatom fő célja egy olyan rendszer létrehozása, melyhez nincs szükség képfeldolgozó algoritmusokra, mégis megvalósítható egy földi objektum megfigyelése a levegőből. Azonban, hogy ez kivitelezhető legyen más adatok alapján kell elvégezni a követést. A következő információkra van szükség. Pontos magassági információk, szélességi és hosszúsági adatok GPS alapján, illetve a repülőgép pozíciójára és orientációjára. A feladat a következő. A cél és a repülőgép közötti pontos távolság kiszámítása. A célpont irányának a meghatározása a repülőgéphez képest, illetve a kamera forgatásához szükséges szögek meghatározása. A következőkben ezeknek a kiszámítási módját írom le.

### 7.2 A gép és a célpont közötti távolság kiszámítása

Két GPS koordináta közötti távolság meghatározásánál számolni kell a Föld alakjából adódó problémákkal. Kis távolságok és sík terep esetén ez nem számottevő, azonban változó



7.1. ábra: Távolság mérés példa

terepviszonyoknál már befolyásolja a számítás eredményét. A 7.1-es ábrák jól reprezentálják az eltérést. Ezt figyelembe kell venni. Azonban, ezeknek a problémáknak a kiküszöbölésére számos formulát dolgoztak ki, melyek közül egyet mutatok be.

### 7.2.1 Haversine formula

A Haversine formula egy olyan egyenlet, amit előszeretettel alkalmaznak a navigációban. Egy úgynevezett nagy-kör (gömb felületén végighaladó) távolságot ad meg hosszúsági (longitude) és szélességi (latitude) adatokból. Azonban van egy hátránya. Nagy távolságok esetén ez sem ad 100%-os eredményt, viszont rövidtávra (2-3ezer km) elfogadható hibahatáron belüli adatot állít elő. [15]

Egyenlete:

$$a = \sin^2(\Delta\varphi/2) + \cos(\varphi_1) * \cos(\varphi_2) * \sin^2(\Delta\lambda/2)$$

$$c = 2 * \operatorname{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R * c$$

$$\text{ahol } \varphi \text{ a szélességi, } \lambda \text{ a hosszúsági fok, } R \text{ a föld sugara } (R = 6371\text{km})$$

(2)

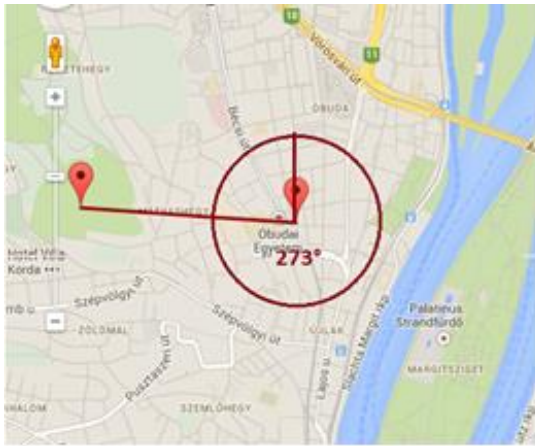
Utasszállító repülőgépek útvonaltervezésénél, vagy földközi hajóutak tervezésénél ez a formula természetesen nem alkalmazható, hiszen sokkal nagyobb távolságok is elképzelhetők 2-3 ezer km-nél. Léteznek sokkal komplikáltabb számolást igénylő módszerek is, mint pl. Vincenty's formula [20]. Azonban a megfigyelés pár száz méteren belül fog történni. Tehát a Haversine formula tökéletesen megfelel az elvárásaimnak.

### 7.2.2 A célpont irányának a meghatározása

A következő fontos dolog, amire szükség van, az a célpont helyzete. Repülés során a gép egy előre meghatározott pályát jár be. A legtöbb esetben a haladási irányt északhoz képest adják meg, ezért célszerűen én is ezt az eljárást fogom alkalmazni. A GPS által szolgáltatott adatok alapján ismert a gép pontos helye, a célpont szélességi és hosszúsági adatát pedig meg kell adni. Tehát, ez alapján kell kiszámolni az irányt. Természetesen itt sem lehet figyelmen kívül hagyni a Föld alakját. Ennek a kiszámítására szintén a Haversine formula egyik alakját fogom felhasználni, mely a következőképpen néz ki (3).

$$\theta = \operatorname{atan2}(\sin(\Delta\lambda) * \cos(\varphi_2), \cos(\varphi_1) * \sin(\varphi_2) - \sin(\varphi_1) * \cos(\varphi_2) * \cos(\Delta\lambda))$$

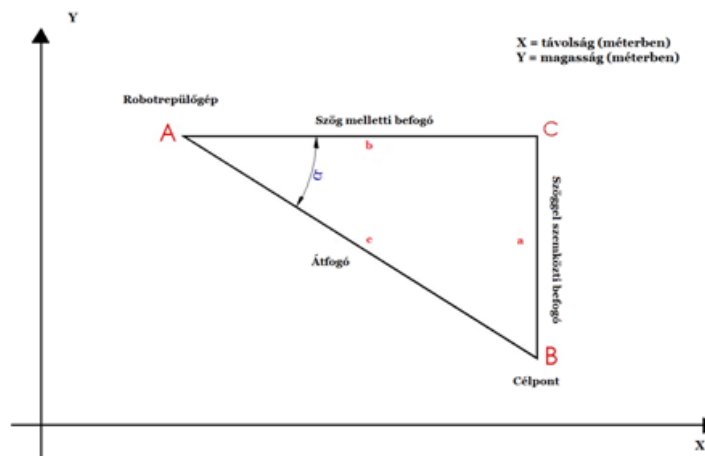
(3)



7.2. ábra: Célpont irányának meghatározása példa

### 7.2.3 Bólintási szög meghatározása

Ahhoz hogy a levegőből rálásson a kamera megfigyelni kívánt pontra a következőkre lesz szükség. A gép és a célpont közötti pontos távolságra (légvonalban) melyet a fentebb említett Haversine formulával lehet meghatározni, illetve a gép és a célpont magasságára.



7.3. ábra: Bólintási szög meghatározása

Az ismert adatokból a következő egyenlettel könnyedén meg lehet határozni az  $\alpha$  bólintási szöget:

$$\operatorname{tg} \alpha = \frac{\text{szögkel szemközti befogó}}{\text{szög melletti befogó}} = \frac{a}{b}$$

(4)

Repülés során előfordulhat, hogy a gép a célpont felett halad el, ekkor a fent említett számítási módszer nem ad eredményt, hiszen a két pont közötti távolság nulla, és az oszthatósági szabály miatt, az összefüggés nem alkalmazható. Ebben az esetben  $\alpha = 90^\circ$ .

### 7.3 A forgatás

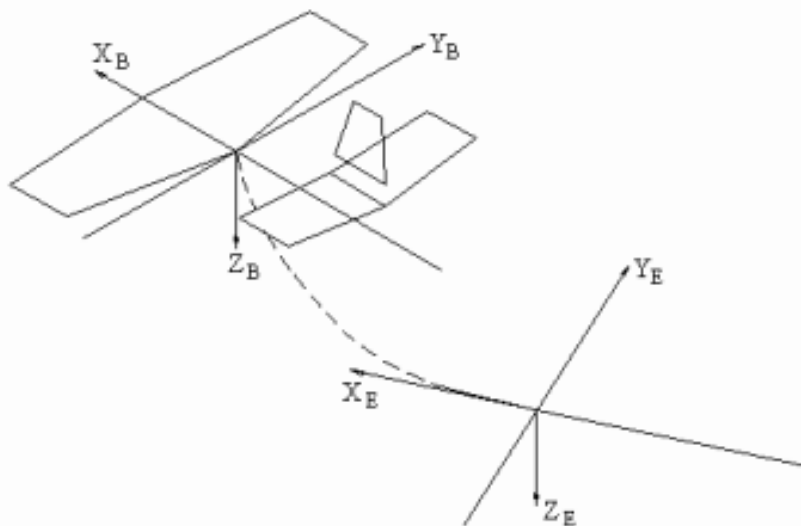
Az előzőek alapján meg tudjuk határozni azt, hogy a gép és a célpont között mekkora távolság van, az úgynevezett bólintási szöget, illetve képesek vagyunk megadni a megfigyelni kívánt pont irányát északhoz képest. Azonban szükség van még a forgatásához további két adatra.

A kamera egy olyan mozgatórendszerre lesz felrögzítve, mely az Y és Z tengely mentén tud elfordulni.

Azonban a tengelyek körüli elforgatáshoz, a szükséges szögek kiszámolásához további két koordinátarendszert kell bevezetni. Az úgynevezett test, illetve földhöz rögzített koordinátarendszert.

#### 7.3.1 Földhöz rögzített koordinátarendszer

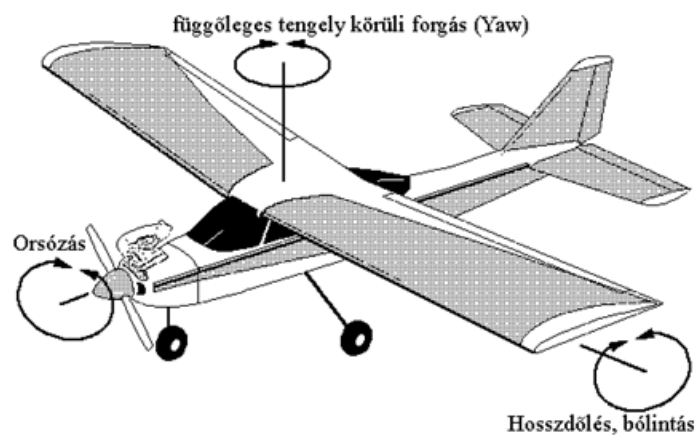
A földhöz rögzített koordinátarendszer egy háromdimenziós, derékszögű, Descartes féle koordinátarendszer. Általában az origója a földfelszínen helyezkedik el. Az  $X_E$  tengely az északi pólus felé, a  $Z_E$  tengely a Föld középpontja felé néz, az  $Y_E$  pedig ezekkel jobbrendszert alkot. (7.3-as ábra)



7.4. ábra: A Föld (E=Earth) és a test (B=Body) koordinátarendszere

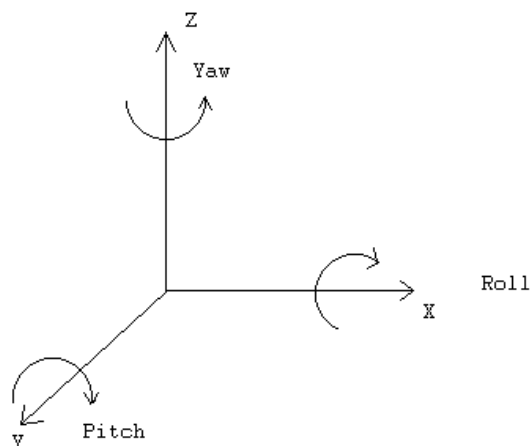
### 7.3.2 A test koordinátarendszere

A test koordinátarendszer kezdőpontját a repülőgép tömegközéppontjához szokás rögzíteni. Az  $X_B$  tengely a repülőgép orra felé mutat, az  $Y_B$  a jobbszárny felé, a  $Z_B$  pedig ezekkel jobbrendszert alkot, tehát lefelé mutat. Ideális helyzetben a géptest és a földhöz rögzített koordinátarendszer tengelyei párhuzamosak. Repülés során ez természetesen folyamatosan változik. A két koordinátarendszer tengelyei között az úgynevezett Euler-szögek teremtenek kapcsolatot. Az Euler szögek a következők: azimut vagy elfordulás (Yaw), bólintás (Pitch), orsózás (Roll).



7.5. ábra: Repülőgép mozgása

Ezeknek a szögeknek az ismeretében az egyik rendszerből a másik, három egymás utáni kötött sorrendű elforgatással megkapható. Ezt a transzformációt az egyes tengelyekhez tartozó forgatási mátrixokkal lehet véghezvinni. [16]



7.6. ábra: Yaw, Roll, Pitch fordulás

A „Roll” forgatást leíró mátrix, mely az X tengely körüli elfordulásért felel:

$$R(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (5)$$

A „Pitch” forgatást leíró mátrix, mely az Y tengely körüli elfordulásért felel:

$$R(\beta) = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (6)$$

A „Yaw” forgatást leíró mátrix, mely a Z tengely körüli elfordulásért felel:

$$R(\gamma) = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Belátható, hogy a három forgatási mátrix egy eredő forgatómátrixként felírható. Azonban ügyelni kell a helyes forgatási sorrendre. Először ugyanis a „Roll” forgatást kell elvégezni, és ennek eredményét beszorozni a transzformációknak megfelelően balról a „Pitch”, majd a „Yaw” transzformációk mátrixával.

$$R(\alpha, \beta, \gamma) = R(\gamma) * R(\beta) * R(\alpha) =$$

$$= \begin{bmatrix} \cos(\beta) * \cos(\alpha) & \sin(\gamma) * \sin(\beta) * \cos(\alpha) - \cos(\gamma) \sin(\alpha) & \cos(\gamma) * \sin(\beta) \cos(\alpha) + \sin(\gamma) \sin(\alpha) \\ \cos(\beta) \sin(\alpha) & \sin(\gamma) \sin(\alpha) \sin(\beta) + \cos(\gamma) \cos(\alpha) & \cos(\gamma) * \sin(\beta) * \sin(\alpha) - \sin(\gamma) \cos(\alpha) \\ -\sin(\alpha) & \sin(\gamma) \cos(\beta) & \cos(\gamma) \cos(\beta) \end{bmatrix} \quad (8)$$

### 7.3.3 A forgatáshoz szükséges szögek meghatározása

A forgatómátrixok ismeretében a kamera forgatása a következő módon fog végrehajtódni. Lesz egy  $\mathbf{T}$  vektor, mely a megfigyelni kívánt pontra fog nézni, illetve lesz egy  $\mathbf{C}$  vektor, mely a kamera orientációját fogja leírni.[16] [19]

Célpontra mutató vektor, a földi koordináarendszerben:

$$\mathbf{T}_E = \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix} \quad (9)$$

Az elforgatott célpontra mutató vektor, a földi koordináarendszerben:

$$\mathbf{T}_{E_{Rot}} = \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix} \quad (10)$$

Az elforgatott célpontra mutató vektor, a test koordináarendszerben:

$$\mathbf{T}_B = \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix} \quad (11)$$

A kamerát leíró vektor a test koordináarendszerben:

$$\mathbf{C}_B = \begin{bmatrix} C_X \\ C_Y \\ C_Z \end{bmatrix} \quad (12)$$

Forgatómátrix:

$$\mathbf{R} = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix} \quad (13)$$

Elsőnek a célpontra mutató  $\mathbf{T}$  vektort kell elforgatni a megadott irányba. Alaphelyzetben a  $\mathbf{T}$  vektor a földhöz rögzített koordináarendszer  $\mathbf{X}$  tengelyével párhuzamos, az iránya vele megegyezik, azaz észak felé néz és kiinduláskor  $\mathbf{T}_E = [1 \ 0 \ 0]$ . Legyen a

Haversine formula segítségével kiszámolt célpont északhoz képesti iránya  $\vartheta$  és az úgynevezett bólintási szöget  $\Omega$ . Ekkor az  $\mathbf{R}$  forgatómátrix segítségével, a célpontra mutató vektor kiszámolható a következő összefüggés segítségével:

$$\mathbf{T}_{E_{Rot}} = \mathbf{R}(\vartheta, \Omega, 0) * \mathbf{T}_E \quad (14)$$

A két koordinátarendszer között a kapcsolatot az  $\mathbf{R}$  forgatómátrix írja le. A fentebb említett módon, ha ismertek az Euler szögeket, akkor az egyik rendszerből a másik, három egymás utáni kötött sorrendű elforgatással megkapható. Azonban ebben az esetben ez leredukálható két forgatásra, mivel nem fontos az, hogy a kamera képe mindig vízszintes legyen. Azaz csak a  $\mathbf{Z}$  tengelyhez tartozó **Yaw** szögre és az  $\mathbf{Y}$  tengelyhez tartozó **Pitch** szögre lesz szükség. Ezeket a következő módon lehet meghatározni.

Elsőként a  $\mathbf{Z}$  tengely körüli elforgatást kell kiszámolni. Maga a forgatás mindig a testhez rögzített koordinátarendszer  $\mathbf{X}$  tengelyéhez képest fog megtörténni. Azaz a kamera orientációját leíró vektor rögzített, mindig  $\mathbf{C}_B = [1 \ 0 \ 0]$ , azaz a gép orra felé néz. Ezért a célpontra mutató vektort kell meghatározni a testhez rögzített koordinátarendszerben. Ehhez szükséges az  $\mathbf{R}$  mátrix inverze mert, az  $\mathbf{R}$  segítségével csak a test koordinátarendszerében ismert vektorokat lehet felírni a földhöz rögzített koordinátarendszerben. Mivel az  $\mathbf{R}$  mátrix ortogonális mátrix, azaz oszlopvektorai merőlegesek egymásra és hosszuk egységnyi ezért  $\mathbf{R}^{-1} = \mathbf{R}^T$ . Így:

$$\mathbf{R}^{-1} = \mathbf{R}^T = \begin{bmatrix} r_{xx} & r_{yx} & r_{zx} \\ r_{xy} & r_{yy} & r_{zy} \\ r_{xz} & r_{yz} & r_{zz} \end{bmatrix} \quad (15)$$

Az  $\mathbf{R}^T$  segítségével már fel lehet írni az elforgatott célpontra mutató vektort a testhez rögzített koordinátarendszerben.

$$\mathbf{T}_B = \mathbf{R}^{-1} * \mathbf{T}_{E_{Rot}} = \mathbf{R}^T * \mathbf{T}_{E_{Rot}} \quad (16)$$

Ezek után ismert a kamera orientációját leíró  $\mathbf{C}_B = [1 \ 0 \ 0]$  vektor és a testhez rögzített koordinátarendszerben felírt  $\mathbf{T}_B = [\mathbf{x} \ \mathbf{y} \ \mathbf{z}]$  vektor. Ahhoz, hogy meg lehessen határozni a **Yaw**

elfordulási szöget, szükség van a  $\mathbf{T}_B$  vektor  $\mathbf{X}$ ,  $\mathbf{Y}$  síkra vetített képére. Ehhez a  $\mathbf{T}_B$  vektor  $z$  koordinátáját ki kell nullázni. Így a  $\mathbf{Yaw}$  szög egyenlő lesz a két vektor skaláris szorzatával.

$$\cos(\theta) = \frac{\mathbf{T}_B * \mathbf{C}_B}{|\mathbf{T}_B| * |\mathbf{C}_B|} \quad (17)$$

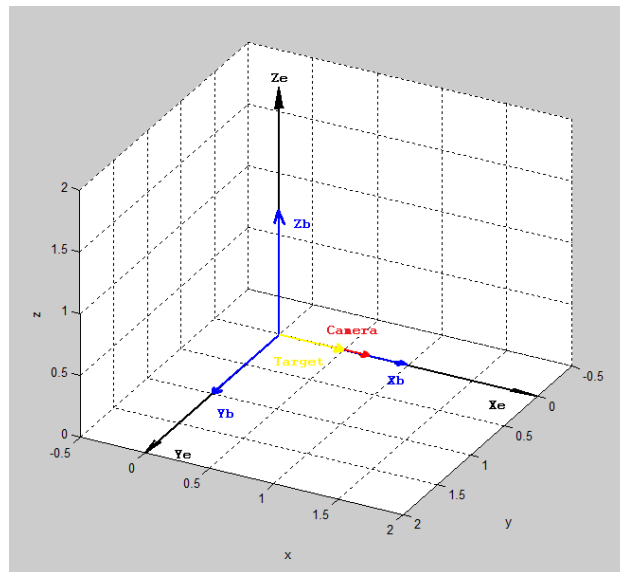
Azonban ha a gép, a célpont felett helyezkedik el, akkor a  $|\mathbf{C}_B| = 0$ . Ebben az esetben a kamerát nem kell elforgatni, hanem marad alap állapotban és csak a bólintási szög lesz egyenlő  $90^\circ$ -al.

A bólintási szöget egyébként, a  $\mathbf{T}_B = [\mathbf{x} \ \mathbf{y} \ \mathbf{z}]$  és a  $\mathbf{T}_B = [\mathbf{x} \ \mathbf{y} \ 0]$  vektorok skaláris szorzata fogja megadni.

## 7.4 A kamera elforgatásának bemutatása

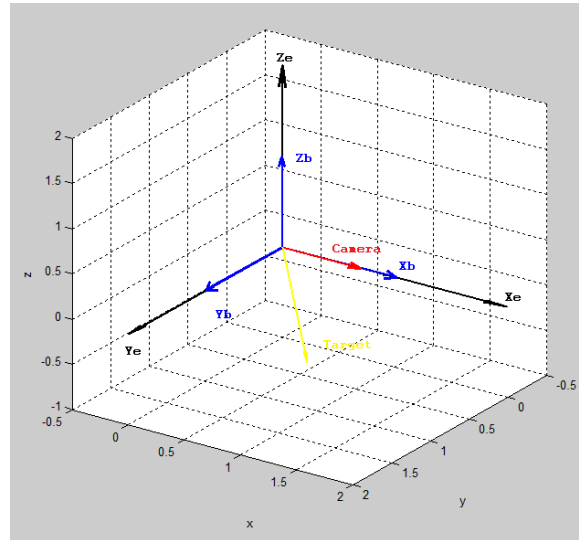
A következőekben néhány ábrán keresztül bemutatom a kamera forgatásának a menetét.

Alaphelyzetben a test és a földhöz rögzített koordinátarendszer tengelyei párhuzamosak. Tehát a gép orra észak felé ( $\mathbf{X}_b$ ), jobb szárnya kelet felé ( $\mathbf{Y}_b$ ) néz, az  $\mathbf{Z}_b$  tengely pedig függőleges helyzetben van. A kamera a gép aljára van rögzítve, így kiinduláskor a gép orra felé néz (Camera), illetve a célpontra mutató (Target) vektor is az  $\mathbf{X}_e$  tengellyel párhuzamos. (7.7-es ábra)



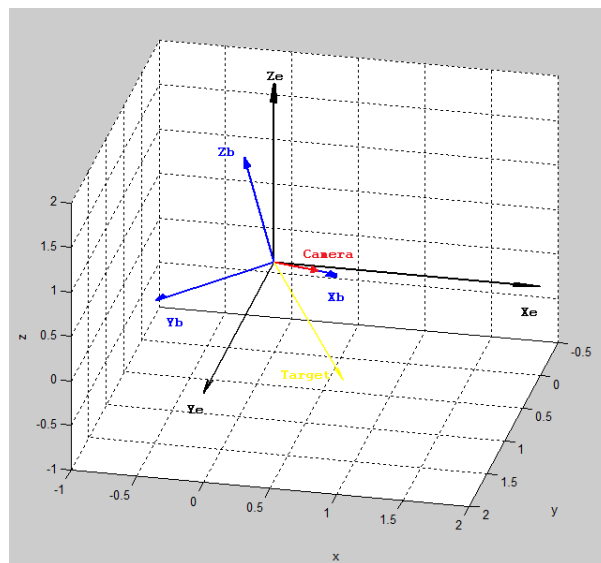
7.7. ábra: Kamera és repülőgép kiindulási helyzete

A Havershine formula segítségével kiszámolt irány, illetve távolság alapján, elsőnek a sárga színnel jelölt „**Target**” vektort kell elforgatni az **R** forgatómátrix segítségével úgy, hogy az a célpontra mutasson. A példában 45 fokos Yaw elfordulást és 45 fokos bólintást adtam meg. (7.6-os ábra).



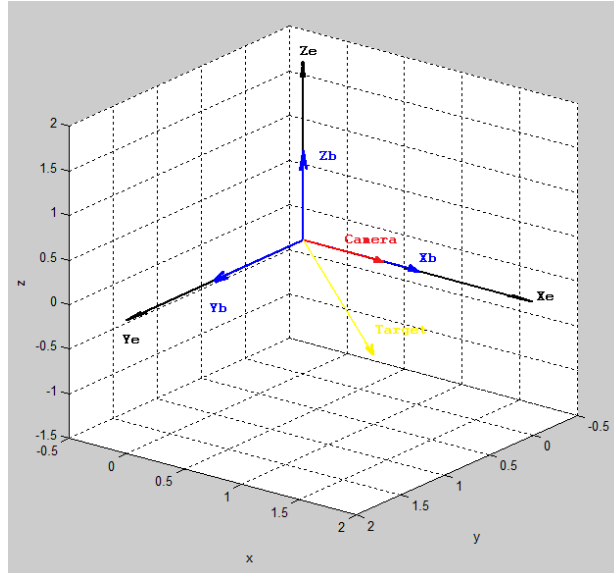
7.8. ábra: Célpont meghatározása

A repülés során a gép orientációját, az egyes tengelyek mentén való elfordulás szögét egy IMU (Inertial Measurement Unit) biztosítja. A yaw, pitch, roll szögek ismeretében az (14)-es mátrixegyenlet segítségével, a kék színnel ábrázolt test koordináta-rendszere elforog. Az elfordulás eredménye a 7.8-as ábrán látható. Megfigyelhető, hogy ebben az esetben, a piros színnel jelölt „**Camera**” vektor is a géppel együtt elfordult.



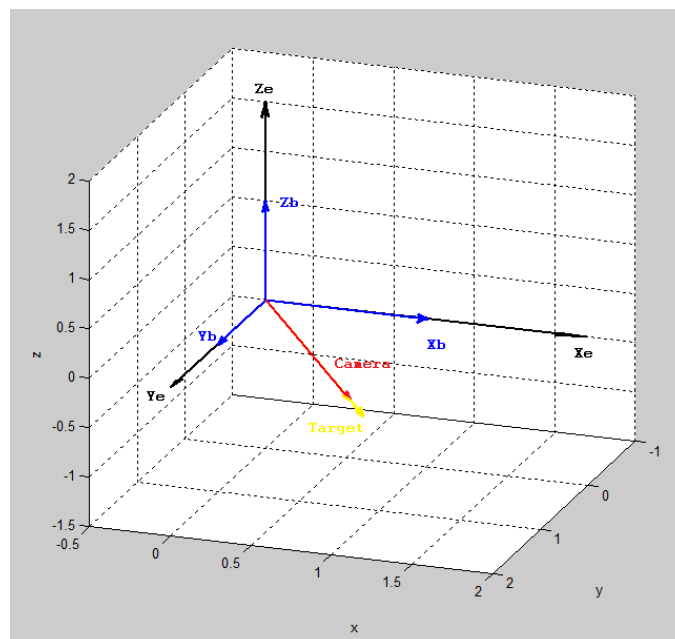
7.9. ábra: Gép koordináta-rendszerének az elforgatása

Azonban mint említettem, a könnyebb számolás érdekében, nem a földhöz rögzített koordinátarendszerben határozom meg a kamera orientációját leíró vektort, hanem fordítva, azaz a testhez rögzített koordinátarendszerben határozom meg a célra mutató vektort. Azaz a (13)  $\mathbf{R}$  forgatási mátrix inverzével kell beszorozni. Az eredmény a 7.9-es ábrán látható.



7.10. ábra: A célpontra mutató vektor meghatározása a testhez rögzített koordinátarendszerben.

Ebben az esetben, a „Target” vektor  $X_b$  és  $Y_b$  síkra vetített képe megegyezik a „Camera” vektorral, így csak a két vektor skaláris szorzatát kell kiszámolni, majd a kapott szöggel, a „Camera” vektort elforgatni, az  $Y_b$  tengely mentén. Az eredmény a következő ábrán látható.

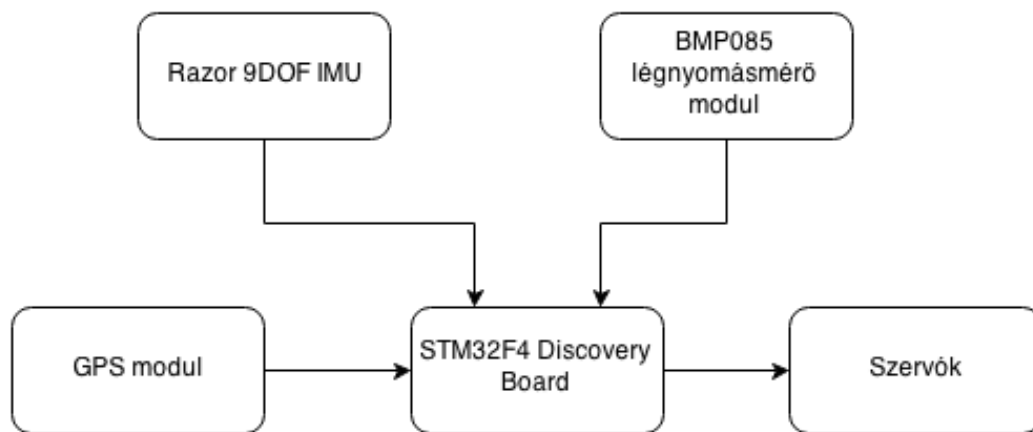


7.11. ábra: Kamera vektor elforgatása az  $Y_b$  tengely mentén

## 7.5 A megvalósítás

### 7.5.1 A rendszer főbb hardvermoduljai

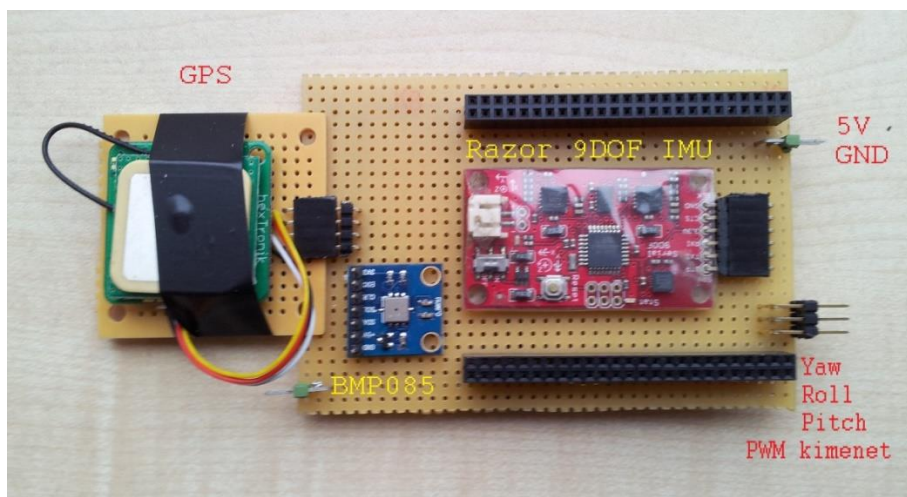
- STM32F4 Discovery Board
- BMP085 légnyomásmérő modul
- Sparkfun Razor 9DOF modul
- U-blox NEO-6M GPS modul
- Szervók
- Kamera



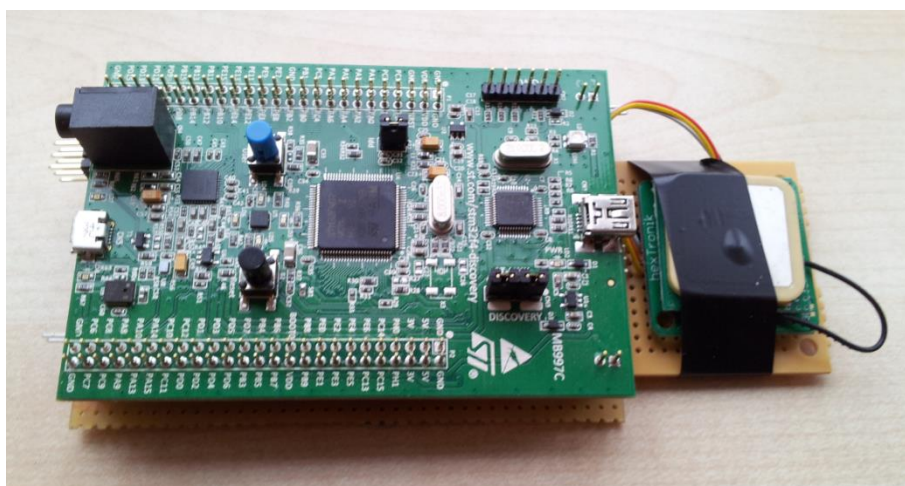
7.12. ábra: OSD és a kamerastabilizáló rendszerterve

A fenti ábrán látható a stabilizálást végző hardver blokkdiagramja. Középen az STM mikrokontroller, mely a számolásokat végzi, illetve ezek alapján előállítja a beavatkozó szervek számára a megfelelő PWM jeleket. Ehhez csatlakozik a GPS, IMU illetve a BMP085 szenzor.

Az egyes szenzorokat egy próbapanelen helyeztem el (7.13-as ábra), melyet úgynevezett shield-ként (pajzsként) az STM boardhoz lehet csatlakoztatni (7.14-es ábra).



7.13. ábra: Próbapanelen elhelyezett szenzorok (shield)



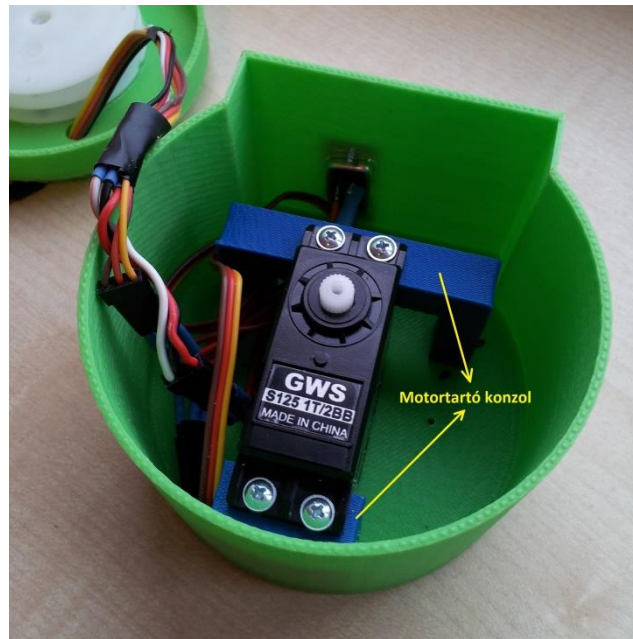
7.14. ábra: Az STM32F4-es Discovery board és a hozzá csatlakoztatott shield

## 7.5.2 A kamera felépítése

A repülés során az élőképet egy NTSC szabványú CCD kamera fogja közvetíteni. Ezt a kamerát oly módon kell elhelyezni, hogy szabadon el tudjon forogni a függőleges tengelye körül jobbra és balra is  $180^\circ$ -ot illetve be lehessen billenteni lefele  $90^\circ$ -ot. Ehhez egy CAD szoftver segítségével terveztem egy dómot, ami biztosítja mindezt. Az egyes alkatrészeket egy 3D nyomtató segítségével hoztam létre. Ez a technológia lehetővé tette azt, hogy könnyű, de egyben strapabíró anyagból tudjam megvalósítani az elemeket, ami elengedhetetlen szempont a robotrepülőgépek esetén.

Négy főbb részből áll. Van egy alsó része, amiben helyet kapott a függőleges körüli elfordulásért felelős szervomotor (7.15-ös ábra). Két konzol segítségével rögzítettem hozzá a

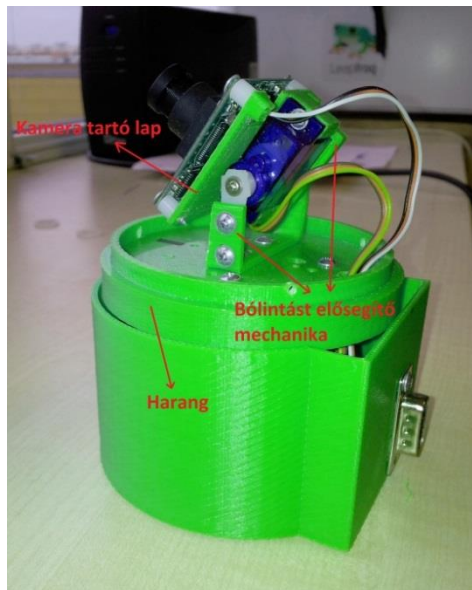
motort a dóm aljához. Úgy helyeztem el, hogy a szervomotor tengelye pontosan a dóm alsórészének a közepére essen.



7.15. ábra: Dóm kamera alsó része

Erre azért volt szükség, mert közvetlenül a motorhoz lesz rögzítve a dóm felső része, mely három részből áll. Van egy harang, ami a motorhoz lesz rögzítve. Erre fog rákerülni a bólintáshoz lehetővé tévő mechanika, illetve maga a kupola, ami a kamerát fogja védeni a külső behatásoktól.

A bólintásért is egy szervomotor felel, mely 90°-os elfordulásra képes. Ahhoz, hogy bólintás a lehető legkisebb helyen végbemehessen, a kamerát egy kis konzol segítségével közvetlenül a szervomotor házához rögzítettem (7.16-os ábra). Maga a forgástengely pedig a motor tengelye lett.



7.16. ábra: Kamera és a hozzátartozó mozgó mechanika

Mivel ezek a beavatkozó szervek általában egy erőkarral rendelkeznek, ezért a másik oldalát is alá kellett támasztani, mert mozgás közben a kamera túlságosan berezonált.

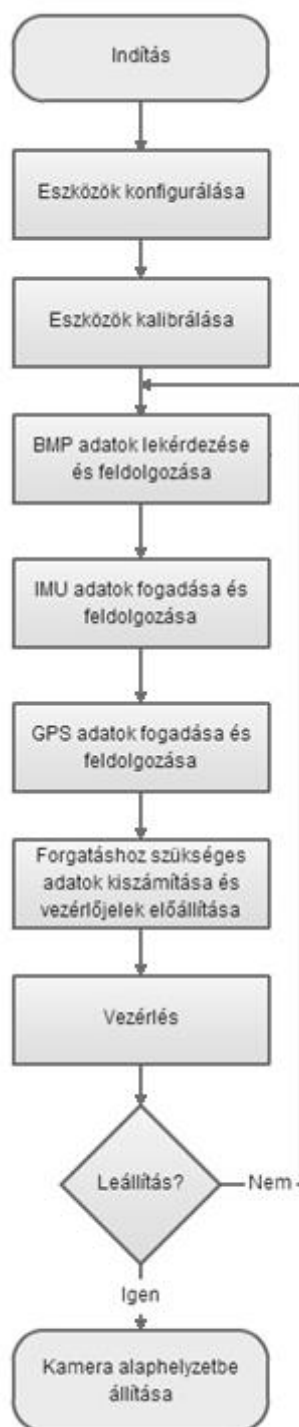
A védelmet a külső behatásoktól a kamera számára maga a kupola jelenti. A kupolán egy kémlelő nyílás található, mely közvetlen rálátást biztosít a megfigyelni kívánt földi pontra. Teljes egészében a kamera dóm a 7.17-es ábrán látható.



7.17. ábra: Dóm kamera

## 7.6 A beágyazott rendszer általános szoftverdiagramja

A 7.18-as ábrán, a stabilizálást végző beágyazott rendszer általános szoftverdiagramja látható.



7.18. ábra: Fő szoftvermodul általános diagramja

### **7.6.1 Konfigurálás és kalibrálás**

Első lépésként az egyes perifériákat kell konfigurálni. Szükség lesz két darab UART-ra, egy TIMER-re, melynek segítségével a PWM jeleket állítja elő a processzor, egy bemenetre melyhez a boardon található gomb tartozik, három általános kimenetre, melyhez szintén a boardon található LED-ek vannak csatlakoztatva, illetve egy I2C interfészre is.

### **7.6.2 UART konfigurálás**

Két soros interfészre lesz szükség, mert az IMU is és a GPS is soros porton keresztül továbbítja az adatokat. Manapság a mikrokontrollereken ezeknek az interfészeknek a száma jelentős. Az STM32F4 Discovery boardon egy STM32F407VGT6-os processzor található, melyen összesen hat darab USART van. Ezek közül én két darabot fogok felhasználni, az USART2-őt melynek a TX kimenete a PA2-es lábra az RX bemenete pedig a PA3-as lábra van kivezetve, illetve az USART3-at, melynek a TX kimenete a PB10-es az RX bemenete pedig a PB 11-es lábra van kivezetve.

Az kommunikáció 8 bit hosszú csomagokkal történik, mely tartalmaz egy darab STOP bitet. A GPS 38400 baudrate-el, a Razor IMU pedig 57600-as baudrate-el kommunikál.

### **7.6.3 I2C konfigurálás**

Az I2C a BMP085-ös légnyomásmérő szenzorhoz kell. Az I2C1-es perifériát fogom erre felhasználni. Két kivezetésre van szükségem. Az órajelet továbbító SCL-re mely a PB6-os lábon található illetve a soros adatvonalra, SDA-ra mely a PB7-es lábon található. A légnyomásmérő szenzor adatlapja alapján akár 400khz-es frekvenciával is tud üzemelni a szenzor, de én ezt 100khz-re állítottam be.

### **7.6.4 LED-ek és gomb**

Az LED-eket állapotjelzőkként fogom alkalmazni. Az egyes funkciók működését a villogó LED-ek fogják jelezni, mert magát a villogtatást a főcikluson belül fogom véghezvinni, ami azt fogja jelezni, hogy a program fut. A gomb segítségével pedig módokat lehet majd váltani.

### **7.6.5 TIMER konfigurálás**

A szervomotorok vezérléséhez PWM jelet kell előállítani. A beavatkozó jelet egy TIMER segítségével fogom generálni. Minden TIMER-hez összesen négy csatorna tartozik. Ebből én kettőt használok fel, mivel két motort kell vezérelni. A TIM4-es periféria egyes és kettős csatornája a PD12-es és PD13-as lábon található. A PWM előállításához további beállítást kell elvégezni. Először is PWM módba kell állítani a perifériát. Ezután meg kell adni azt a periódusidőt és kitöltési tényezőt, amellyel a megfelelő jel előáll. Ebben az esetben az órajelet le kell osztani 12-vel. A periódus értéket 65535-re kell beállítani. A servók 1000us-os kitöltési tényezővel rendelkező jel esetén a baloldali végállásba áll, 1500us-os jel esetén alaphelyzetbe, középre áll, 2000us-os kitöltési tényező esetén pedig a jobboldali végállásba áll. Ehhez, az egyes csatornákhöz tartozó CCR (capture/compare register) regisztereket kell módosítani. Azaz, 6500-tól 13000-ig vehetnek fel értéket. Induláskor mind a két servo alaphelyzetben van. A módosított bólintásért felelős servo 1 fokkal történő forgatásához  $(13000-6500)/90=72,22$ -vel kell megnövelni vagy csökkenteni az adott CCR regiszter értékét. A vitorlacsörlő esetén ez változik, mivel 4x360 fokos szervóról van szó. Ebben az esetben  $(13000-6500)/(4*360)=4,5$ -el kell növelni vagy csökkenteni.

### 7.6.6 Magasságmérő szenzor kalibrálása



7.19. ábra: Magasságmérés folyamatmodellje

A BMP085-ös szenzor nem rendelkezik bootloaderrel, azaz a felhasználónak kell gondoskodnia a kalibrálásról, illetve a nyers adatok kiolvasásáról.

A kalibrálás nem közvetlenül a modulon belül történik, hanem az általa mért adatok és a gyártó által megadott összefüggések segítségével lehet kiszámítani a valós értékeket.

Első induláskor a szenzorban található EEPROM tartalmát kell kiolvasni, mely a kalibrációs adatokat tartalmazza. Ezt mindig csak egyszer, a bekapcsolás után kell elvégezni. Ha ez megtörtént, be kell állítani a mérési pontosságot. Ez azért fontos, mert a pontosságtól függ az, hogy a szenzornak mennyi idő kell az adatok előállításához. Ez azt jelenti, hogy van egy holt idő, amikor nem lehet a szenzorból adatot kiolvasni. Ez a várakozási idő hőmérséklet esetén 4,5ms-ot, légnyomás esetén pedig 4,5-től 25ms-ot jelenthet. Ennek a betartása elengedhetetlen, mert különben fars értékeket kapnánk eredményül. Azonban a gyártó ennek a kiküszöbölésére egy plusz lábbal látta el a szenzort. Ez az EOC (End of conversion) láb arra szolgál, hogy ha a szenzor készen áll az olvasásra, akkor ezt egy logikai 1-nek megfelelő feszültségszinttel jelzi, azaz megszakítást lehet vele generálni a processzor számára.

### 7.6.7 BMP085 adatok lekérdezése és feldolgozása

A magassági adatok lekérdezése az egyik legfontosabb feladat. Négy főbb rutin tartozik hozzá. Az első a `BMP085_Init()`; mely induláskor az I2C perifériát állítja be. Ez a metódus induláskor a fő ciklus előtt fut le. Ezután következik a `BMP085_Calibration()`; Ez a függvény valósítja meg a kalibrációs adatok kiolvasását a szenzor EEPROM-ból. Az egyes adatokat több változóban tárolom le. A kalibrálás után a szenzor már működőképes. A hőmérséklet és légnyomás értékek kiolvasásához két különálló függvény tartozik, mert az egyes értékek a szenzor memóriájában különálló helyen található, illetve ahhoz hogy a nyers értékekből valós adatot tudjak előállítani más számítási módszert kell alkalmazni, amit már a 13. oldalon részleteztem. A két függvény `Get_Temperature()`; és `Get_Pressure()`; Mind a két függvény egy float értéket ad vissza eredményül.

Ahhoz, hogy a kamerát a megfelelő sebességgel és pontossággal lehessen a célponton tartani, a magassági adatoknak mindig a lehető legfrissebbnek kell lenniük. Éppen ezért 100Hz-es frekvenciával kérem le az adatokat, amibe bőven belefér a szenzorból való kiolvasáshoz szükséges időkorlát miatt fennálló várakozási idő.

### 7.6.8 IMU adatok fogadása és feldolgozása

A Razor 9DOF IMU szolgáltatja a három legfontosabb információt a gép orientációjáról. Az adatok UART-on keresztül biztosítja egy adott frekvenciával, egy előre definiált mondat formájában. A mondat felépítése a következő. „**\$ANG, Roll, Pitch, Yaw**”, ahol a Roll, Pitch, Yaw egész szám és fokban értendő.

Ha az RX lábon beérkezik egy adat, az az USART DR (data regiszter) regiszterébe kerül. Ezek után két lehetőség van. Vagy folyamatosan vizsgálni kell ennek a regiszternek a tartalmát és ha adat kerül bele, akkor vagy ki kell olvasni, vagy egy megszakítást kell hozzárendelni. Én az utóbbit alkalmaztam, mert lényegesen leegyszerűsíti a feladatot. Ha adat érkezik, akkor az USART periféria USART\_IT\_RXNE-bitje bebillen és egy megszakítást generál, a megszakításon belül pedig megtörténik a kiolvasás egy körbufferbe.

A magassági adathoz képest, a gép orientációját leíró szögeket sokkal magasabb frekvenciával kell mintavételezni, különben a kamera stabilizálásához előállított beavatkozó jel késni fog, így csökkenne a pontosság, a kamera nem a célpontra nézne. Természetesen ez a mintavételezés nem történhet gyorsabban, mint ahogy az IMU az adatokat küldi.

Az UART konfigurálásért az `Init_Razor_Usart()`; metódus felel.

## 7.6.9 GPS adatok fogadása és feldolgozása

A kamera pontos pozíciójának a nyomon követésére egy U-blox Neo-6M típusú GPS modult használok fel. A kommunikáció szintén soros porton (UART ) keresztül valósul meg és szintén csak egyirányú, azaz a GPS modul ad, az STM32F4 board pedig fogad. Az adatok fogadása ugyan úgy történik, ahogy az IMU-nál.

Azonban a GPS adatok NMEA mondat formájában érkeznek. '\$' jellel kezdődnek, és egy „checksum”-al végződnek, mely egy \*-ból és utána álló két hexadecimális számból áll.

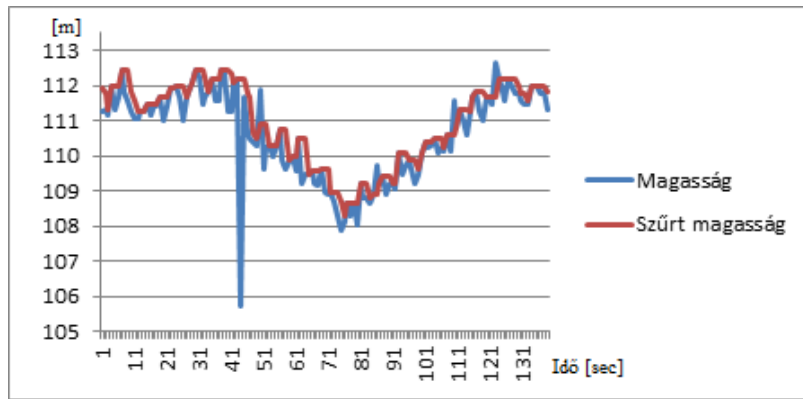
A GPS mondatok feldolgozásához egy struktúrát hoztam létre „gps” néven. Ebben a struktúrában a számomra fontos adatok típusai szerepelnek.

```
struct gps{  
    float   lon;  
    float   lat;  
    int     valid;  
    int     satellites;  
    float   altitude;  
    float   speed;  
    float   bearing;  
    float   time;  
    int     3dfix;  
    int     checksum;  
};
```

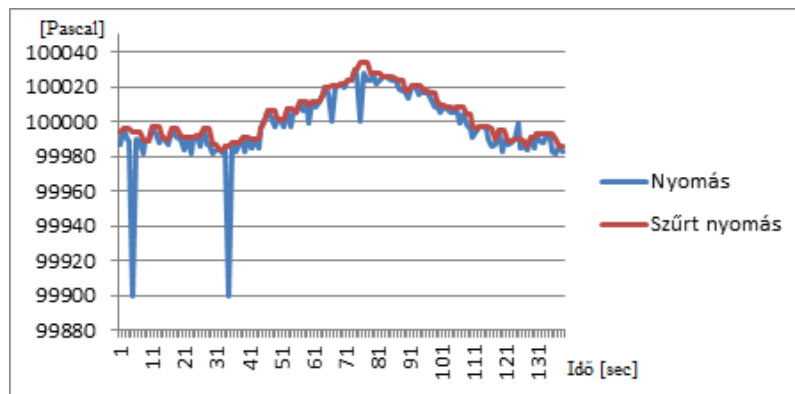
Első lépésként a mondat helyességét kell leellenőrizni, tehát hogy a fogadott „checksum” érték és az általam előállított „checksum” érték megegyezik e. A köbbufferban megkeresem a mondat kezdetét jelző „\$” jelet, majd pedig egy XOR művelettel előállítom a checksum-ot. Ha az általam előállított checksum megegyezik a GPS mondatban található számmal, akkor a mondat helyes, következhet a feldolgozás. Ezt egy GPS\_Parse() nevű függvény meghívásával viszem végbe. A függvényen belül megvizsgálom, hogy \$GPGGA , \$GPRMC vagy pedig GPGSA mondatról van e szó. \$GPGGA esetében kiolvasásra kerül a *longitude*, *latitude*, *valid*, *satellites*, *altitude*, *time* értékek. \$GPRMC esetén a *speed* és *beraing*, \$GPGSA-nál pedig a *3dfix*, ami azt jelzi, hogy a pontos 3D helymeghatározás él.

### 7.6.10 Magasságmérő modul, IMU, GPS teszt

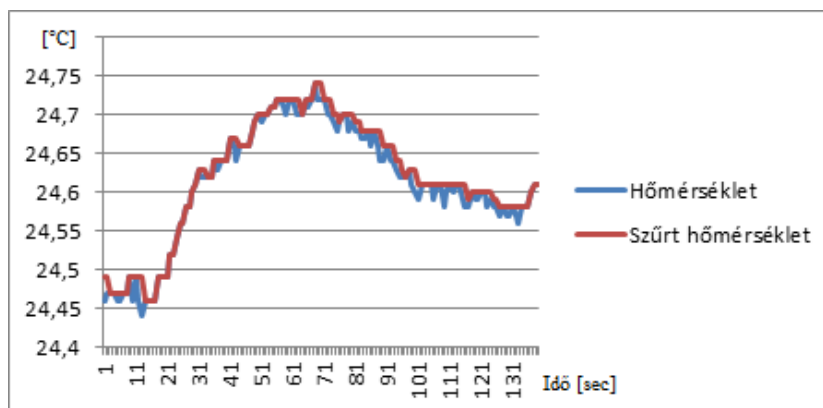
Miután sikeresen csatlakoztattam az egyes modulokat az STM32F4 boardhoz, tesztelés céljából néhány mérést végeztem el. A mérések eredményei a következő diagramokon láthatóak.



7.20. ábra: Magasság



7.21. ábra: Légnyomás



7.22. ábra: Hőmérséklet

A kék színnel jelölt vonaldiagrammok a nyers mérések eredményeit jelzi. Látható, hogy az egyes adatok zajosnak bizonyultak. Ezek a kiugrások az elforgatási szögek kiszámításánál téves adathoz vezethetnek, mely a kamera instabil működését idézné elő. Tehát szűrést kell alkalmaznunk. Számos módszer létezik a hibák kiküszöbölésére, azonban én egy úgynevezett mediánszűrőt alkalmaztam. Ez a szűrő egy szűrőablakot használ. Az adatok beérkezési sorrendben az ablakon áthaladnak. Az ablakban található értékek rendezése után, a középső elem fogja adni a szűrt értéket. Belátható, hogy rendezés során az irreálisan nagy, illetve kis értékek a sor elejére illetve végére kerülnek, így ezek az adatok nem fognak jelentéssel bírni.

A szűrés menete:

1. Adat beléptetése a csúszó ablakba.
2. Az ablakban lévő elemek sorba rendezése.
3. A középső elem kiválasztása.

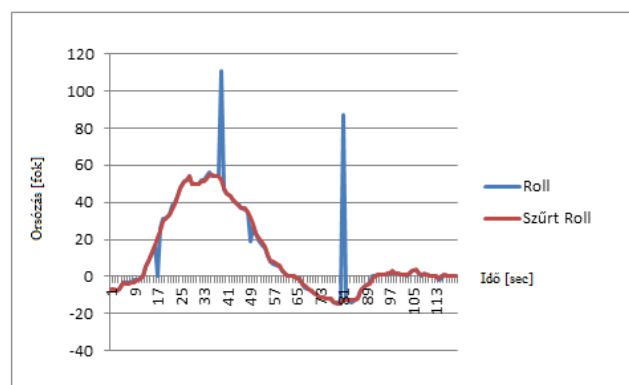
Az adatok jobbról kerülnek beléptetésre az ablakba, csúsztatás során pedig a baloldali elem lép ki. A csúszó ablak méretét nem érdemes túl nagyra választani, különben a hasznos adatot is kiszűrjük, kis méret esetén pedig nem elég hatásos. Célszerű egy köztes méretet meghatározni. Én jelen esetben 5 eleműre állítottam.

Példa:

Adatok: [10, 12, 40, 15, 0, 13, 16]

1. Ablak: [10,12,40,15,0] = [0,10,12,15,40] = 12
2. Ablak: [12,40,15,0,13] = [0,12,13,15,40] = 13
3. Ablak: [40,15,0,13,16] = [0,13,15,16,40] = 15

A példa jól szemlélteti, hogy a legkisebb és a legnagyobb elem nem jut „szóhoz”, el van nyomva. Az a BMP085 szenzor szűrt adatai az 7.15, 7.16, 7.17-es ábrán bordó színnel láthatóak, az IMU nyers és szűrt adatai pedig a 7.18, 7.19, 7.20-as ábrán.



7.23. ábra: Szűrt és szűretlen Roll adatok



### 7.6.11 Forgatáshoz szükséges adatok kiszámítása és vezérlőjelek előállításának folyamata

A program legnagyobb részét a forgatáshoz szükséges adatok kiszámítása teszi ki. Ez eléggé időigényes feladat, ezért a főciklusban kapott helyet. Erre azért volt szükség, mert ha megszakításokkal oldottam volna meg, akkor túlságosan sok ideig le lett volna foglalva a processzor, így az UART-on érkező GPS és IMU adatok kezelése is körülményesebb lett volna.

A forgatáshoz három rutin tartozik. A `Rotate_vector()`; , `InvRotate_vector()`; és a `Get_rotate()`;

A `Rotate_vetor()`; paraméterként egy vektort és három szöget vár. A vektor a célpontra mutató vektort írja le. Alaphelyzetben ennek az értéke  $t=[1\ 0\ 0]$ . A három szög rendre a Yaw, Roll, Pitch szögek. A függvényen belül, a 49. oldalon található (14)-es összefüggés megvalósítása található, illetve definiálva van a 48. oldalon található (13) **R** forgató mátrix.

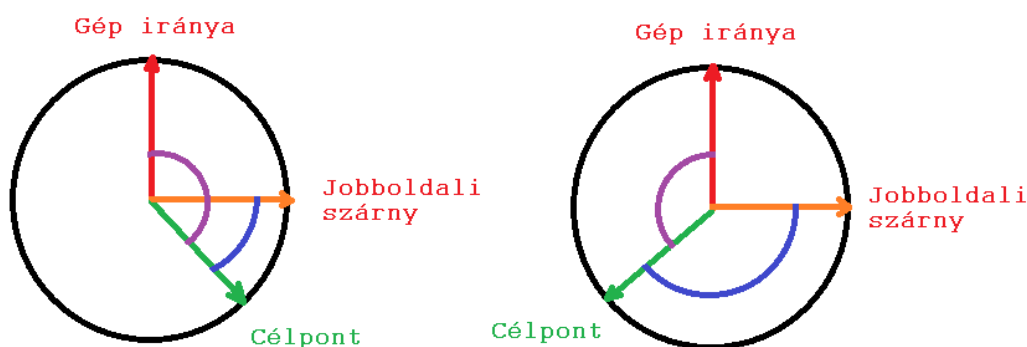
Elsőként a `Rotate_vector()`; fut le, azaz kiszámolásra kerül a célpontra mutató vektor. Ehhez azonban szükség van a célpont északhoz viszonyított irányára illetve a bólintási szögre. Az irány meghatározásáért a `Get_Bearing()`; függvény a felelős. Paraméterként a repülőgép és a célpont GPS koordinátáját kell átadni, majd egy float értékkel tér vissza. A bólintás meghatározását a `Get_Pitch()`; függvény végzi el. Paraméterként meg kell adni a célpont és a repülőgép magasságát, GPS koordinátáját. A GPS koordinátára azért van szükség, mert ezek alapján történik a két pont közötti távolság meghatározása.

Az első lépés tehát, a célpontra mutató vektor elforgatása az **R** forgatómátrix segítségével a földi koordinátarendszerben. A további forgatás már ezen a vektoron fog végrehajtódni, de nem az **R**, hanem az  $\mathbf{R}^{-1}$  mátrix segítségével. Ezt az `InvRotate_vector()`; függvény fogja elvégezni. A függvényen belül szintén definiálva van az  $\mathbf{R}^{-1}$  mátrix, illetve a hozzá tartozó számítások.

A célpontra mutató vektor elforgatása után, már csak a Yaw illetve Pitch szögeket kell meghatározni. Ezt a `Get_Rotate()`; függvény fogja megvalósítani. A függvényen belül meghatározásra kerül a célpontra mutató vektor X, Y tengely által kifeszített síkra vetített képe. Itt kell figyelni arra, hogy ha célpont felett van pontosan a gép, akkor ennek a levetített vektornak a hossza nulla lesz. Két vektor skaláris szorzatánál ez hibát eredményezne nullával

való osztás miatt. Ezért ebben az esetben a számítás félbeszakad, az előző Yaw érték megmarad, a bólintás pedig konstans  $90^\circ$  lesz.

A függőleges tengely körüli elforduláshoz kiszámolt szög azonban nem egyértelmű, mivel a két vektor által bezárt szög kiszámolásához használt skaláris szorzat, mindig a kisebb szöggel tér vissza, azaz  $180^\circ$ -nál nem lesz nagyobb. Ez azért probléma, mert nem lehet eldönteni így, hogy a kamerát jobbra vagy pedig balra kell fordítani.



7.27. ábra: Célpont lehetséges elhelyezkedése

A 7.27-es ábra szemlélteti a két esetet. A baloldali ábrán a célpont a kör jobboldali felében helyezkedik el, a jobboldali ábrán pedig a baloldali félkörben. Mind a két esetben ugyan azt a Yaw szöget kapnánk eredményül, de nem tudnánk eldönteni, hogy balra, avagy jobbra kell elfordulni. Ezért bevezettem egy újabb vektort, ami a gép jobboldali szárnyát írja le. Így, ha a jobboldali szárny és a célpont közötti szög nagyobb, mint  $90^\circ$  akkor a kamerának balra kell elfordulnia.

A forgatáshoz szükséges adatok rendelkezésre állása után már csak a megfelelő beavatkozó jeleket kell előállítani a PWM kimeneteken. Ehhez nem kell mást tenni, mint hogy a TIMER egyes csatornáihhoz tartozó CCR regiszterek értékét be kell állítani.

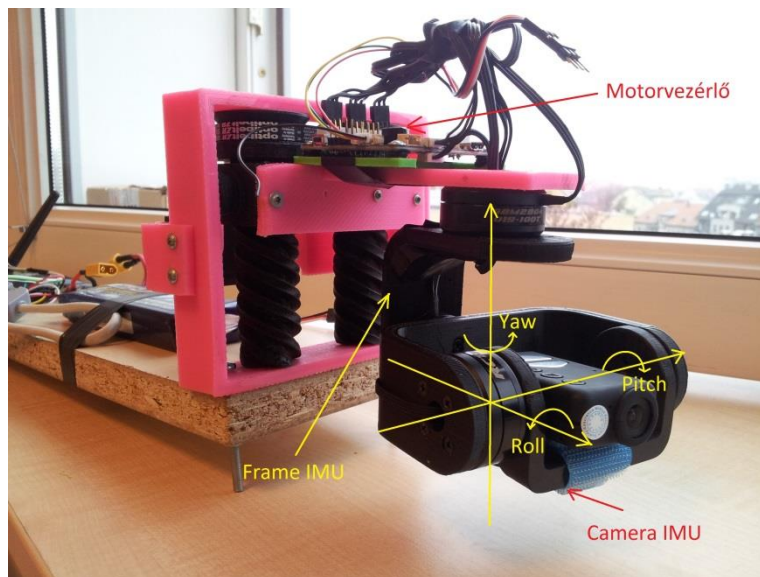
## 7.7 Kamera stabilizálás kefenélküli motorral ellátott gimbal segítségével

Manapság egyre szélesebb körben elterjedtek az úgynevezett kefenélküli motorral ellátott kamerastabilizáló gimbalok. Az egyre olcsóbb áraknak köszönhetően, egyre több modellrepülőre és quadcopterre szerelik fel. Segítségével bemozdulás és rezgésmentes videók készíthetők. Létezik kettő illetve háromtengelyes változat. Külön motorvezérlő elektronikával vannak ellátva, melynek köszönhetően a stabilizálás egyénileg, profi módon megoldott.

Miután az általam tervezett kétszervós kamera dóm elkészült, úgy döntöttem, hogy én is kihasználom a technika által nyújtott előnyöket.

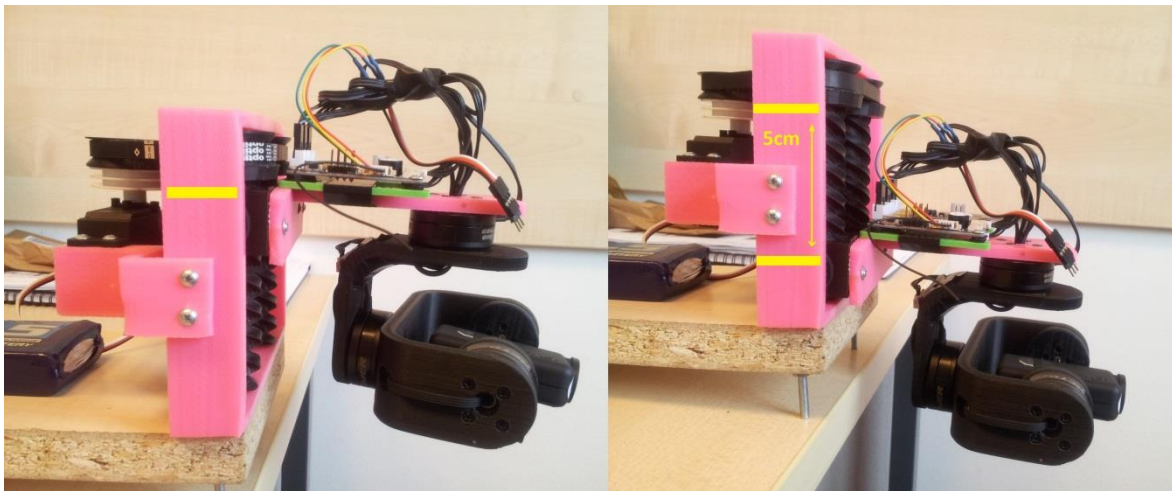
A háromtengelyes változat mellett döntöttem. Egy Alexmos típusú 32bit-es processzorral rendelkező vezérlőt választottam. Mind a három tengelyhez egy-egy BLDC (kefenélküli) motor tartozik. Ezek nagynyomatékú, lassú, de precíz mozgásra képesek. A tengely és a forgórész között nincs mechanikus áttétel, azaz a szervomotorokkal ellentétben nem kell számolni kotyogással. Mivel a motorok állásáról nincs közvetlen visszacsatolás, ezért a stabilizálás giroszkópok és gyorsulásmérő szenzorok segítségével valósították meg. Ehhez a vezérlőhöz két darab ilyen szenzort lehet csatlakoztatni. Az egyik szenzor a kamera tartóján foglal helyet, a másik pedig a gimballal tartó frame-en. Ennek köszönhetően nem csak a kamera mozgásából adódó változások alapján stabilizálja a kamerát a vezérlő, hanem a frame mozgását is belekalkulálja.

A boltokban kapható legtöbb gimbal fémről készült, melynek súlya emiatt jelentős. Ezért újból a 3D nyomtatás technológiát hívta segítségül, és egyénileg lett megtervezve az összes alkatrész. Mivel robotrepülőgépre lesz felszerelve, úgy döntöttem, hogy egy kieresztő mechanikával is ellátom. Természetesen ez többlet súlyt jelent, azonban felszállás és leszállás esetén a kamera a gép törzsében fog elhelyezkedni, azaz védve lesz a külső behatásoktól.



7.28. ábra: Gimbal

A 7.28-as ábrán látható maga a gimbal, illetve a kiengedő mechanika. A kamera kiengedését két tengely teszi lehetővé, melyet egy kocsi köt össze. Erre a kocsira van felrögzítve a gimbal. A tengelyeket egy szervomotor forgatja el egy bordásszíj segítségével, melynek hatására a kocsi mozgásba lendül. Ennek köszönhetően a géptestből közel 5cm-re lehet a kamerát kiengedni, ami bőven elegendő helyet biztosít a forgatáshoz. (7.29-es ábra)



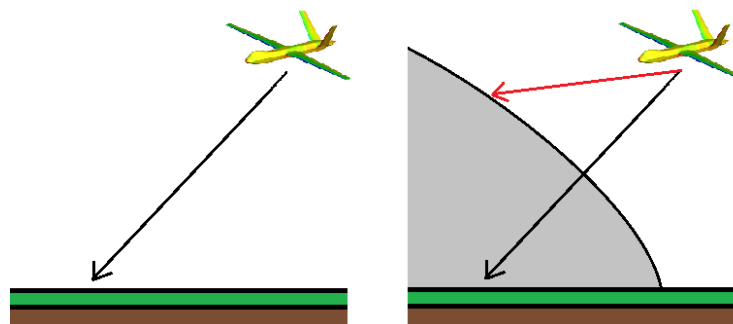
7.29. ábra: A gimbal behúzott illetve kieresztett állapota

A 7.29-es ábra baloldalán a kamera behúzott, a jobb oldalon pedig kiengedett állapotban látható.

A forgatáshoz szükséges számolások némiképpen módosultak. Ugyanis a kamera mindig vízszintes állapotban van, tehát nincs szükség forgatómátrixokra. Maga a gimbal egy csuklóként funkcionál, így a Yaw motor pl. 45 fokos elfordítása a kamera 45 fokos elfordulását jelenti orientációtól függetlenül. Ebben az esetben elegendő meghatározni, a repülőgép irányából és a cél irányából a szükséges elfordulási szöget. Továbbá a bólintási szög meghatározása is leegyszerűsödött, mivel mindig a vízszinteshez képest kell a kamerát bedönteni. Ezt a szöget pedig a magassági és távolsági adatokból könnyedén kiszámolható.

## 8 Célpont magasságának a pontos meghatározása

A célpont pontos magasságának a meghatározása elengedhetetlen a számítások során. Azonban a domborzati viszonyokkal számolni kell. A következő ábra (8.1) szemlélteti a lehetséges eseteket.



8.1. ábra: Célpont magasságának a meghatározásával járó probléma

Megfigyelés sík terep esetén, a forgatáshoz szükséges szögeket minden probléma nélkül GPS koordináták alapján meg lehet határozni. Azonban ha a pont nem síkságon, hanem egy domb oldalán helyezkedik el, ez a módszer már nem alkalmazható. Számításba kell venni a ponthoz tartozó magasságot is.

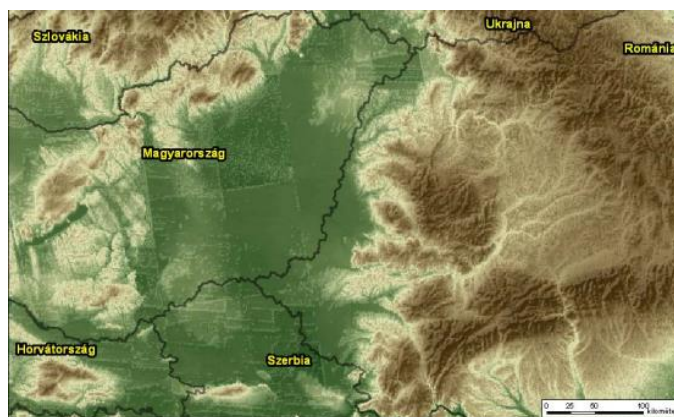
Ennek meghatározására az úgynevezett digitális domborzatmodellt fogom felhasználni.

### 8.1 Digitális domborzatmodell

A digitális terepmodell egy olyan adatbázis, mely egy topográfiai felület bizonyos pontjainak az X, Y, Z térbeli koordinátáit tartalmazza. Ha ez a felület egy földfelszín, akkor domborzatmodellről (DDM - Digitális domborzatmodell) beszélünk. A DDM a vizsgált terület magassági adatait tartalmazza számszerű formában. Ha ez a modell kiegészül a különböző felszíni objektumok, növényzet, épületek magassági adataival akkor digitális terepmodellről beszélünk (DTM). Nekem a DDM-re lesz szükségem. A domborzat leképezéséhez, digitalizálásához több technológiát is alkalmaznak. Ez történhet hagyományos geodéziai módszerekkel, globális helymeghatározó (GPS) rendszerek alkalmazásával, mikrohullámú távérzékeléssel, lézeres letapogatással stb. Ezeknek a mérési eredményeknek egy része elérhető az interneten, illetve megvásárolható több cégtől is, mint pl. a Földmérési és Távérzékelési Intézet (FÖMI). Globális lefedettségben két nyilvános adatbázis létezik. Az 1km-es felbontású GOTOP30 illetve a 90m-es felbontású SRTM (Shuttle Radar Topography Mission). [18]

### 8.1.1 Global 30 Arc-Second Elevation Data Set (GOTOP30)

A GOTOP30 digitális modell teljes globális lefedettségű, 30 szögmásodperces horizontális és méteres vertikális felbontással rendelkezik. Több magassági adatbázis felhasználásával jött létre. A Kárpát-medencét lefedő GOTOP30 adatok alapján létrehozott modellt, ha jobban szemügyre vesszük, látható hogy több topográfiai térképszelvényből lett összeállítva (8.2-es ábra). Nem a kellő gondossággal lett elkészítve, így kisebb léptékű modellezési feladatok esetén több hibával is számolni kell.



8.2. ábra: GOTOP30 adatok alapján előállított modell

### 8.1.2 Shuttle Radar Topography Mission (SRTM)

Ez a legegységesebben előállított, nagyfelbontású topográfiai adatbázis. Az adatokat egy radarrendszer segítségével állították elő. Egy nemzetközi projekt keretében valósult meg az NGA (National Geospatial-Intelligence) és a NASA közreműködésével. Az USA területén 1 szögmásodperces felbontásban az USA-n kívüli területeken pedig 3 szögmásodperces felbontásban tették közzé az SRTM adatokat. A vertikális felbontása 1 méter. Az adatok hibaelemzését a NASA Jet Propulsion Laboratory-ban végezték el. Eddig négy verziót publikáltak, minél későbbi verzió annál kevesebb hibát tartalmaz. A legjobb SRTM V4-es adatbázis elérhető a <http://srtm.csi.cgiar.org> oldalon. Én is az itt található modellt fogom felhasználni. Az oldal a Consultative Group for International Agriculture Research (CGIAR) társaságé. Az oldalon megtalálható a teljes globális adatbázis. Egy rácsháló segítségével manuálisan ki tudjuk választani a nekünk szükséges részeket, melyeket ezek után le is tölthetünk tömörített egy állomány formájában.

1. Select Server: ☒ CGIAR-CSI (USA) ☐ HarvestChoice (USA) ☐ JRC (IT) ☐ King's College (UK) ☐ TeleScience (USA)

2. Data selection method: ☒ Multiple Selection ☐ Enable Mouse Drag ☐ Input Coordinates

Many tiles can be selected at random locations. These selected tiles are listed in the results page for download.

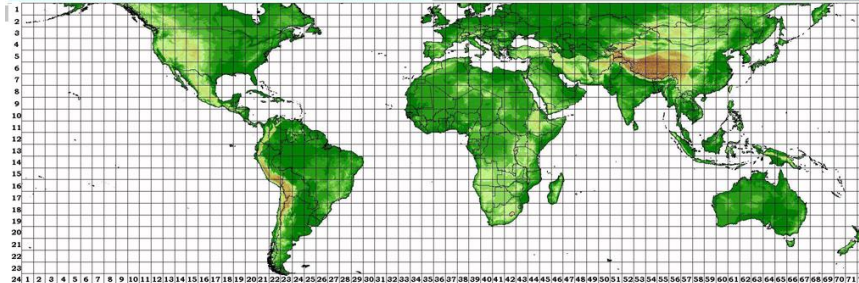
☐ Decimal Degrees (e 34.5, -100.5) ☒ Degrees: Minutes: Seconds (e 34 30 00 N, 100 30 00 W)

Longitude - min:  max:  Longitude - min:   East  max:   East

Latitude - min:  max:  Latitude - min:   North  max:   North

Longitude: 141.80 Latitude: -53.86 Tile X: 65 Tile Y: 23

3. Select File Format: ☒ GeoTiff ☐ ArcInfo ASCII

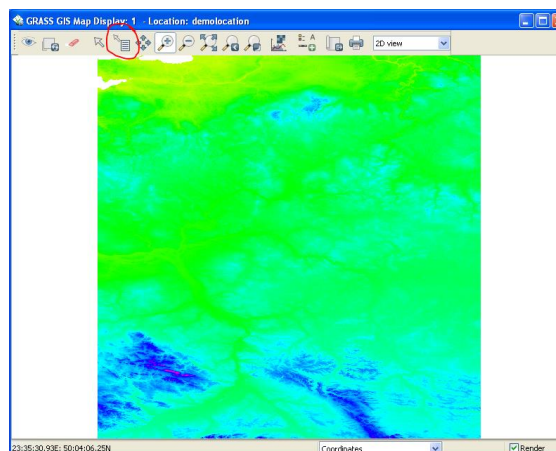


8.3. ábra: <http://srtm.csi.cgiar.org> oldalon található kezelőfelület

A zip állomány tartalmazza a kiválasztott térképszelvény „.tiff” kiterjesztésű raszteres képét. A fájlok megnyitására több program is rendelkezésre áll. Én az ingyenesen elérhető GRASS GIS névre hallgató térinformatikai célú programcsomagot használtam fel.

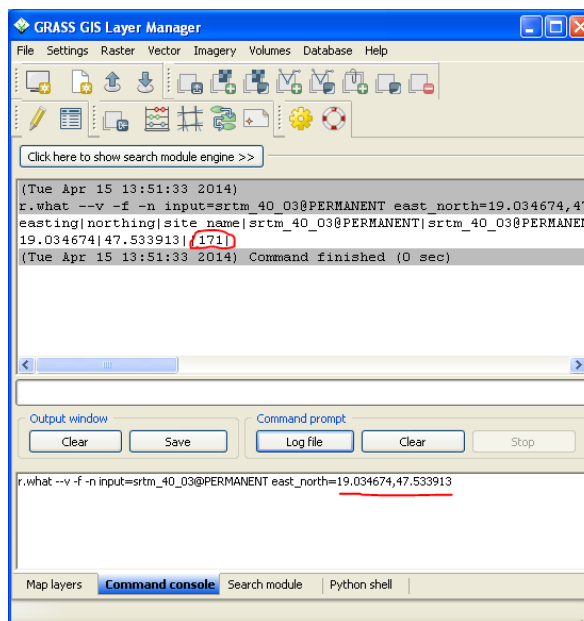
### 8.1.3 Magasság meghatározása a GRASS GIS segítségével

A magasság meghatározása nem igényel nagy feladatot. A program elindítása után két ablak jelenik meg. Az egyik az úgynevezett GIS Manager, a másik pedig a display, ahol a térkép fog megjelenni. Miután letöltöttük a CGIAR oldaláról a nekünk szükséges állományokat, a „File” menüben az „Import raster data” segítségével tudjuk importálni. Ha a betöltés befejeződött, a „Display” képernyőn megjelenik a beimportált térképrészlet. A 8.4-es ábrán Magyarország nyugati fele látható.



8.4. ábra: Magyarország nyugati részének raszteres képe

A magassági adatokat egyszerűen meg lehet határozni. A 8.4-es ábrán pirossal bekeretezett nyilat kiválasztva, ha a térkép bármelyik pontjára való kattintással, a hozzá tartozó szélességi, hosszúsági adatok és a hozzá tartozó magasság „GIS Manager” console ablakában megjelenítésre kerül. A térképen csak a domborzati modell látható, így manuálisan kell a szükséges pontot megtalálni és kijelölni, ami szinte lehetetlen. Azonban lehetőség van kód alapján is lekérdezni az adatokat.



8.5. ábra: GRASS GIS console ablaka

A „Manager” ablak alsó részén lehet a parancsokat kiadni, felül pedig a „console” rész található, ahol a kiértékelt adatok jelennek meg (8.5-ös ábra). Az ismert GPS koordináták alapján, a következő kódrészlettel lehet a magassági adatot lekérdezni:

```
r.what --v -f -n input=srtm_40_03@PERMANENT east_north=19.034674,47.533913
```

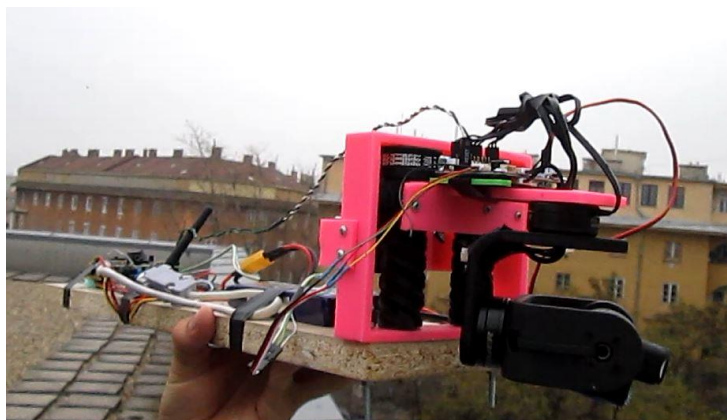
A szélességi és hosszúsági adatok a példában az Óbudai Egyetem Neumann János Informatikai karának a GPS koordinátái. Ez alapján, a hozzátartozó magassági adat 171m (fenti ábra, bekarikázva).

## 9 Tesztelés

Tesztelés céljából először egy mérőpadra (9.1-es ábra) szereltem fel az egyes eszközöket. Mind a két esetben, az általam írt algoritmusokat használtam fel.



9.1. ábra: A dóm kamera tesztelése



9.2. ábra: Gimbal tesztelése

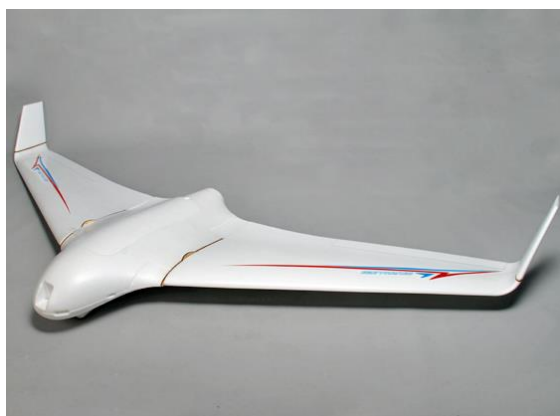
A dóm kamera esetén a CCD kamera biztosította az élőképet, a gimbalnál egy Mobius HD kamera.

Tesztelés során a következőket tapasztaltam. Az általam használt Sparkfun Razor 9DOF típusú IMU, a következő hibajelenségeket produkálta.

Az egyik probléma az, hogy az IMU-n található mágneses érzékelő (iránytű) nagyon érzékeny a fém jelenlétére, illetve az egyes motorok által keltett mágneses térre. Ezért ügyelni kellett arra, hogy az IMU-t úgy helyezzük el, hogy az a lehető legtávolabb essen ezektől a viselkedést befolyásoló eszközöktől. A másik probléma, a szenzor lassúsága. Ebből adódóan, néhány milliszekundumos késést tapasztaltam. Ez a forgatás során azt eredményezte, hogy a kamera forgatása a géptest orientációjának változásához képest késett, azaz nem volt pontos. Ezeket a problémákat egy jobb minőségű IMU segítségével ki lehet küszöbölni.

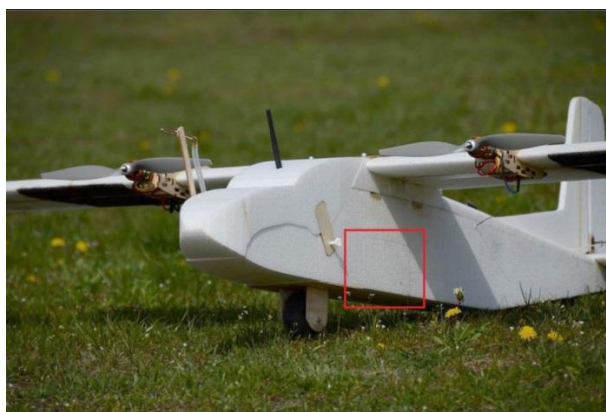
A gimbal rendszer a dóm kamerához képest sokkal nagyobb stabilitással rendelkezik. Azonban itt is jelentkeztek az IMU lassúságából adódó problémák. A gimbal esetén a kamera stabilizálását a motorvezérlő végzi a rajta elhelyezett két IMU segítségével. Az egyes tengely stabilizálásáért felelős kefenélküli motorokat PID szabályzó segítségével vezérli. Némi késés itt is érezhető volt, azonban ez a PID megfelelő hangolásával kiküszöbölhető.

A gimbalos rendszer egy X8 Skywalker típusú géptesthez készült (9.3-as ábra). Ennek a modellnek a bel tere a megszokottnál jóval nagyobb. Emiatt a kiengedő mechanika gond nélkül elfér a törzsben.



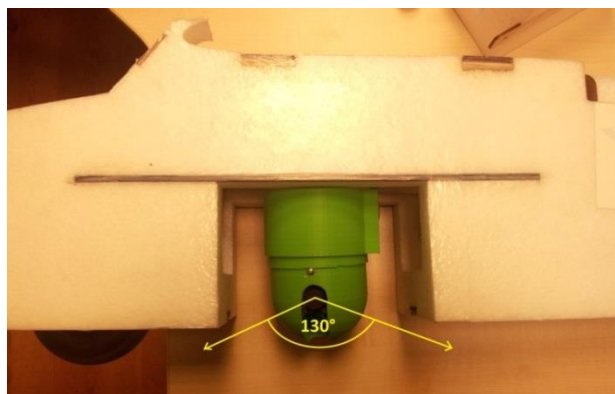
**9.3. ábra: X8 Skywalker**

A dóm kamerát a 9.4-es ábrán látható robotrepülőgépre lesz felszerelve. A gép teste EPP (expandált polipropilén) habból készült mely merev, de mégis könnyű szerkezetet alkot.



**9.4. ábra: Óbudai Egyetem által kifejlesztett pilóta nélküli robotrepülőgép**

A repülőgép rakterét a 9.4-es ábrán piros négyzet jelzi mely 14x20x10 cm. Mivel ez egy igen nagy nyílást jelent, ezért a testet belülről rétegelt lemezből készült merevítéssel látták el, melyre a kamerát is fel lehet rögzíteni. A 9.5-ös ábrán jól látszik a kamera elhelyezése. Látható, hogy szinte a gép törzsében helyezkedik el.



9.5. ábra: Dóm kamera elhelyezése

Emiatt lefelé a látószög lecsökkent, kb. 130°-ra. Ez azonban nem befolyásolja a célpontra való rálátást, ugyanis a megfigyelés minimum 60,70 méteres magasságon történik, aminek köszönhetően a rálátás igen nagy területet lefed. Továbbá, általában a megfigyelés közben a gép a célpont körül köröz. A 9.5-ös ábrán látható, hogy ebben az esetben a kamera nem kerül kitakarásba. Viszont az elhelyezésének köszönhetően védve van a külső behatásoktól, mely felszálláskor illetve leszálláskor érheti a kamerát.

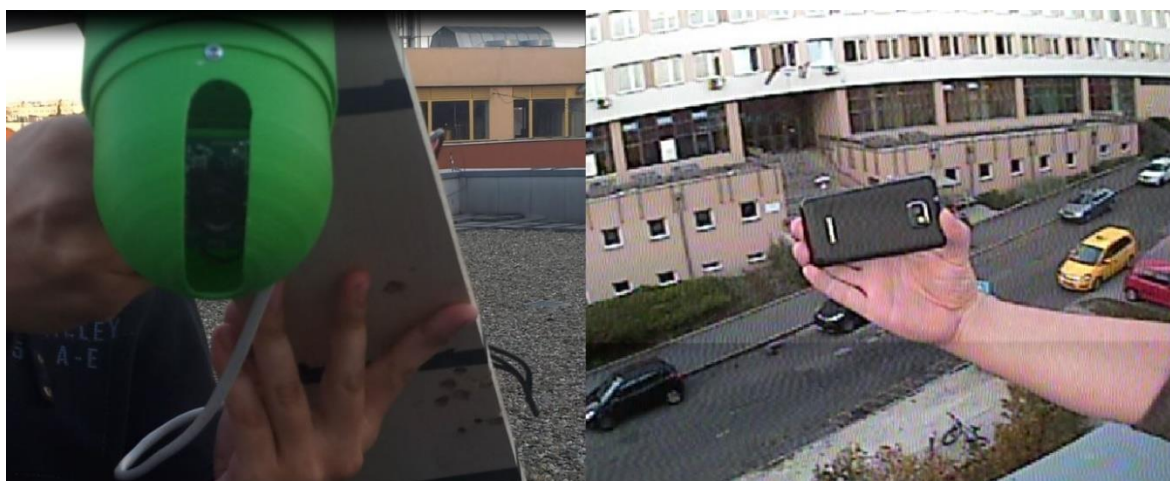
| Eszközök                                  | Mért tömeg (g) | ~Áramfelvétel (mA) |
|-------------------------------------------|----------------|--------------------|
| Felhasznált beavatkozó szervek:           |                |                    |
| Pitch forgatásért felelő szervomotor      | 13             | 95                 |
| Yaw forgatásért felelő szervomotor        | 55             | 150                |
| Yaw forgatásért felelő kefenélküli motor  | 41,42          | 210                |
| Pict forgatásért felelő kefenélküli motor | 41,42          | 210                |
| Yaw forgatásért felelő kefenélküli motor  | 41,42          | 210                |
| Szenzorok:                                |                |                    |
| STM32F4 Discovery board                   | 36             | 87                 |
| Sparkfun Razor 9DOF                       | 3              | 0,02               |
| BMP-085 légnyomásmérő                     | 1              | 0,013              |
| U-blox Neo-6M GPS                         | 15             | 0,02               |
| CCD kamera                                | 25             | 26                 |
| OSD feliratozó                            | 22             | 8                  |
| Boscam AV jeladó + antenna                | 34             | 115                |
| Dóm kamerarendszer                        | 190            | ~481               |
| Gimbal kamerarendszer+kieresztő mechanika | 528            | ~866               |

1. táblázat: Egyes eszközök tömege és áramfelvétele

A következő képeken a dómkamera tesztelése közben rögzített videókból kiragadott néhány részlet látható. A célpont ebben az esetben az épület bejárata volt.



9.6. ábra: Dóm kamera tesztkép 1



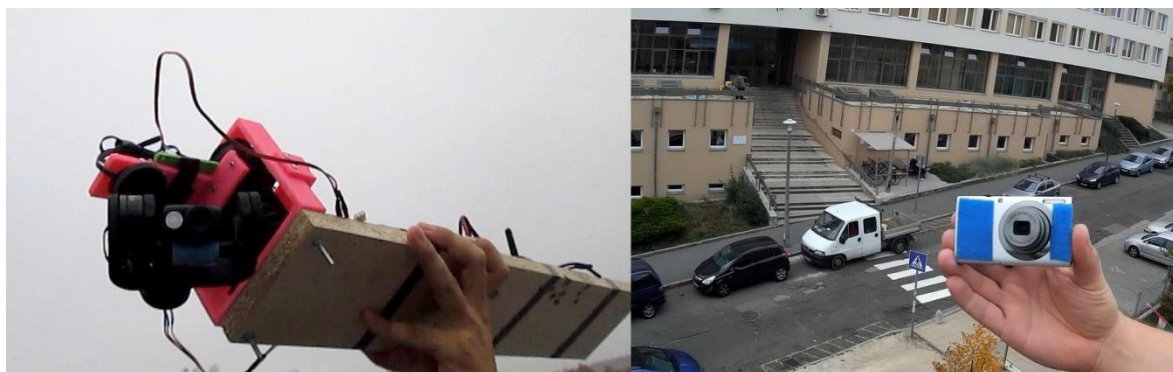
9.7. ábra: Dóm kamera tesztkép 2



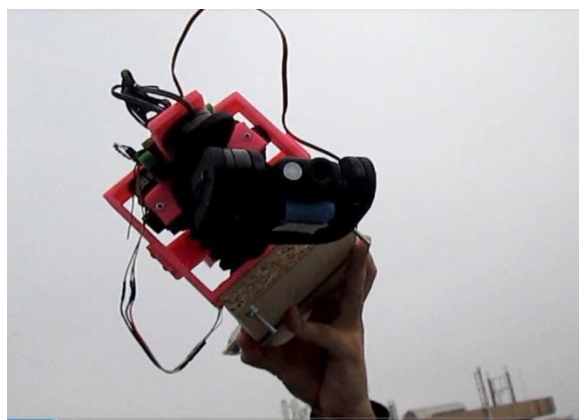
9.8. ábra: Dóm kamera tesztkép 3

A dóm kamera esetén a forgatás csak két tengely mentén valósul meg. A 9.8-as ábrán látható, hogy a kamera továbbra is a bejáratra néz, azonban a kép el van dőlve.

Ez a dőlés a gimbalos rendszernél nem észrevehető, mert ott mind a három tengely stabilizálva van, emiatt a kamera mindig vízszintes helyzetben áll.



9.9. ábra: Gimbal tesztkép 1.



9.10. ábra: Gimbal tesztkép 2.



9.11. ábra: Gimbal tesztkép 3.



9.12. ábra: Tesztrepülés OSD nélkül 1.



9.13. ábra: Tesztrepülés OSD nélkül 2.

## 10 Továbbfejlesztési lehetőségek

A továbbfejlesztés első lépéseként az OSD feliratozó minőségét szeretném javítani. Nappali fényviszonyok között a fehér feliratok beleolvadnak a kamera képébe. Ezért a megjelenített karaktereket, alakzatokat fekete kontúrvonallal szeretném ellátni.

A jelenlegi rendszerhez egyelőre még nincs kifejlesztve a földi állomás. Így a megfigyelni kívánt GPS koordinátát felszállás előtt előre meg kell adni. Azaz, a következő lépés a földi állomáshoz tartozó alkalmazás kifejlesztése.

Továbbá egy olyan funkcióval szeretném kiegészíteni a rendszert melynek segítségével a kamerát emberi beavatkozással is el lehet forgatni. Ez azt a célt szolgálná, hogy repülés közben a megfigyelni kívánt pozíciót manuálisan is ki lehessen választani, melynek a GPS koordinátáját szintén az OSD feliratozó modul megjelenítené a kamera képén. Ennek köszönhetően, pl. ha egy adott objektumot keresése a cél, de nem ismert a pontos helyzete, viszont a kamera képén ez megjelenik, akkor egyszerűen le lehet olvasni az GPS koordinátáját.

Továbbá, a dóm kamera mozgását végző szervomotorokat, kefenélküli motorokra szeretném lecserélni, melyet saját vezérlővel látnám el. A mérési eredmények alapján, a gimbalos rendszer fogyasztása kétszer nagyobb, azonban, a motorok számát 2-re csökkentve, illetve kisebb méretű és tömegű motorok alkalmazásával, sokkal precízebb, finomabb mozgásra képes rendszert lehet megvalósítani

## 11 Összefoglalás

A TDK dolgozatom célja, hogy bemutassa egy olyan robotrepülőgépre szerelhető platformfüggetlen kamerarendszer megtervezésének és elkészítésének folyamatát, amely képes egy fix földi objektum megfigyelésére. A jelenleg piacon lévő rendszerek nagy része képfeldolgozó algoritmusokat használ, azonban a dolgozatom elsődleges szempontja a kis méret és teljesítményigény volt. Így a képfeldolgozás helyett más megközelítést választottam: Egy alacsony fogyasztású mikrokontrollert és további szenzorokat használtam fel, mint pl. GPS, IMU. Tervezés során törekedtem a lehető legkisebb tömegű és méretű kamerarendszer elkészítésére, melyet a 3D nyomtatási technológiának köszönhetően sikerült elérnem. Képfeldolgozó algoritmusok helyett a szenzorok által biztosított adatokra támaszkodtam, és a követést az általam kidolgozott számítási módszerrel sikerült véghezvinnem. A rendszerben helyet kapott egy saját tervezésű OSD modul is, mellyel pixelszinten lehet feliratozni a kamera képét. A szenzorok minőségéből adódó hibák ellenére a teszt során meglepően jó eredményt sikerült elérnem. Két prototípust készítettem el: Egy kéttengelyes rendszert, amelynek a fő célja az volt, hogy minél kevesebb alkatrészből álljon, illetve egy háromtengelyes változatot, ahol a kamera képének a vízszintesen tartása és stabilizálása volt a cél. Ehhez több beavatkozó szervet és plusz vezérlőelektronikát kellett felhasználnom, ami többsúlyt eredményezett, azonban a stabilizálás minősége nagyságrendekkel jobb lett.

## 12 Irodalomjegyzék

- [1] The UAV Predator  
(<http://www.hooked-on-rc-airplanes.com/uav-predator.html>), utoljára megtekintve 2014.03.10
- [2] Alticam 400  
(<http://defense-update.com/products/a/alticam400.htm>), utoljára megtekintve 2014.03.10
- [3] FLIR Systems, Inc. Ultraforce-350HD  
(<http://gs.flir.com/surveillance-products/ultraforce-350hd>), utoljára megtekintve 2014.03.11
- [4] ELI Airborne Solutions, ELI UAV Gimbal  
(<http://www.uav.ee/products/1/uav---gimbal>), utoljára megtekintve 2014.03.08
- [5] Eagle Tree Video On-Screen Display Pro  
(<http://www.espritmodel.com/video-onscreen-display-pro-osd-pro.aspx>), utoljára megtekintve 2014.02.10
- [6] Rembizi OSD  
(<http://www.mictronics.de/projects/remzibi-osd/>), utoljára megtekintve 2014.02.05
- [7] Nootropic design, Video Experimenter  
(<http://nootropicdesign.com/ve/index.html>), utoljára megtekintve 2014.02.05
- [8] Bosch Sensortec GmbH, BMP-085 sensor datasheet, 2009, Germany  
([http://www.adafruit.com/datasheets/BMP085\\_DataSheet\\_Rev.1.0\\_01July2008.pdf](http://www.adafruit.com/datasheets/BMP085_DataSheet_Rev.1.0_01July2008.pdf)), utoljára megtekintve 2014.01.25
- [9] Inversense inc.: MPU60xx datasheet, 2009, USA  
(<http://www.invensense.com/mems/gyro/documents/PS-MPU-6000A-00v3.4.pdf>), utoljára megtekintve 2014.02.01
- [10] NXP Semiconductors, I2C-bus specification and user manual, 2014  
([http://www.nxp.com/documents/user\\_manual/UM10204.pdf](http://www.nxp.com/documents/user_manual/UM10204.pdf)), utoljára megtekintve 2014.02.25
- [11] STMicroelectronic, ST SPI protocol, utoljára megtekintve 2014.02.05
- [12] Universal asynchronous receiver/transmitter (UART) datasheet  
([http://www.nxp.com/documents/data\\_sheet/SCC2691.pdf](http://www.nxp.com/documents/data_sheet/SCC2691.pdf)), utoljára megtekintve 2014.02.05

- [13] Xinis, Hogyan működnek a szervók?  
([http://kecskemetircmodell.blog.hu/2011/03/22/hogyan\\_mukodnek\\_a\\_szervok](http://kecskemetircmodell.blog.hu/2011/03/22/hogyan_mukodnek_a_szervok)),  
utoljára megtekintve 2014.03.01
- [14] Maxim Integrated, Video Basics, 2002  
(<http://pdfserv.maximintegrated.com/en/an/AN734.pdf>), utoljára megtekintve  
2014.02.26
- [15] Calculate distance, bearing and more between Latitude/Longitude points  
(<http://www.movable-type.co.uk/scripts/latlong.html>), utoljára megtekintve  
2014.04.03
- [16] Gregory G. Slabaugh, Computing Euler angles from a rotation matrix  
(<http://scholar.google.com/>), utoljára megtekintve 2014.04.03
- [17] James Diebel: Representing Attitude: Euler Angles, Unit Quaternions, and Rotation  
Vectors, Stanford University, Stanford, California 2006 (<http://scholar.google.com/>),  
utoljára megtekintve 2014.04.05
- [18] Bódis Katalin: Digitális domborzatmodellek és alkalmazási lehetőségeik az árvízi  
kockázatkezelésben című Phd. értekezés, Szegedi Tudományegyetem 2008, utoljára  
megtekintve 2014.04.05
- [19] William Premerlani and Paul Bizard: Direction Cosine Matrix IMU: Theory, 2009,  
utoljára megtekintve 2014.02.05
- [20] Bessel, F. W. (2010). "The calculation of longitude and latitude from geodesic  
measurements (1825)", utoljára megtekintve 2014.03.01

## 13 Ábrajegyzék

|                                                                                                                                                                                 |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1. ábra: RQ-1 Predator's Skyball Gimbal .....                                                                                                                                 | 8  |
| Forrás: <a href="http://www.hooked-on-rc-airplanes.com/uav-predator.html">http://www.hooked-on-rc-airplanes.com/uav-predator.html</a>                                           |    |
| 2.2. ábra: Alticam 400 .....                                                                                                                                                    | 8  |
| Forrás: <a href="http://defense-update.com/products/a/alticam400.htm">http://defense-update.com/products/a/alticam400.htm</a>                                                   |    |
| 2.3. ábra: Ultraforce 350.....                                                                                                                                                  | 9  |
| Forrás: <a href="http://gs.flir.com/surveillance-products/ultraforce-350hd">http://gs.flir.com/surveillance-products/ultraforce-350hd</a>                                       |    |
| 2.4. ábra: Eli UAV Gimbal.....                                                                                                                                                  | 9  |
| Forrás: <a href="http://www.uav.ee/products/1/uav---gimbal">http://www.uav.ee/products/1/uav---gimbal</a>                                                                       |    |
| 3.1. ábra: Eagle Tree OSD Pro .....                                                                                                                                             | 10 |
| Forrás: <a href="http://www.eagletreesystems.com/">http://www.eagletreesystems.com/</a>                                                                                         |    |
| 3.2. ábra: Eagle OSD Pro megjelenített adatai .....                                                                                                                             | 10 |
| Forrás: <a href="http://www.eagletreesystems.com/">http://www.eagletreesystems.com/</a>                                                                                         |    |
| 3.3. ábra: Rembizi OSD 1.....                                                                                                                                                   | 11 |
| Forrás: <a href="http://www.rcgroups.com/forums/showthread.php?t=921467&amp;highlight=remzibi">http://www.rcgroups.com/forums/showthread.php?t=921467&amp;highlight=remzibi</a> |    |
| 3.4. ábra: Rembizi OSD 2.....                                                                                                                                                   | 11 |
| Forrás: <a href="http://www.rcgroups.com/forums/showthread.php?t=921467&amp;highlight=remzibi">http://www.rcgroups.com/forums/showthread.php?t=921467&amp;highlight=remzibi</a> |    |
| 3.5. ábra: Video Experiementer modul.....                                                                                                                                       | 12 |
| Forrás: <a href="http://nootropicdesign.com/ve/">http://nootropicdesign.com/ve/</a>                                                                                             |    |
| 3.6. ábra: Video Experiementer feliratozás .....                                                                                                                                | 12 |
| Forrás: <a href="http://nootropicdesign.com/ve/">http://nootropicdesign.com/ve/</a>                                                                                             |    |
| 3.7. ábra: Video Experiementer fekete-fehér kép.....                                                                                                                            | 12 |
| Forrás: <a href="http://nootropicdesign.com/ve/">http://nootropicdesign.com/ve/</a>                                                                                             |    |
| 4.1. ábra: Légnyomás kísérlet.....                                                                                                                                              | 14 |
| Forrás: Saját ábra                                                                                                                                                              |    |
| 4.2. ábra: Légnyomás exponenciális csökkenése a magasság függvényében .....                                                                                                     | 14 |
| Forrás: <a href="http://www.adafruit.com/datasheets/BMP085_DataSheet_Rev.1.0_01July2008.pdf">http://www.adafruit.com/datasheets/BMP085_DataSheet_Rev.1.0_01July2008.pdf</a>     |    |
| 4.3. ábra: BMP-085 szenzor .....                                                                                                                                                | 14 |
| Forrás: <a href="http://www.adafruit.com/datasheets/BMP085_DataSheet_Rev.1.0_01July2008.pdf">http://www.adafruit.com/datasheets/BMP085_DataSheet_Rev.1.0_01July2008.pdf</a>     |    |

|                                                                                                                                                                                                     |    |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 4.4. ábra: BMP-085 szenzor EEPR térképe.....                                                                                                                                                        | 15 |
| Forrás: <a href="http://www.adafruit.com/datasheets/BMP085_DataSheet_Rev.1.0_01July2008.pdf">http://www.adafruit.com/datasheets/BMP085_DataSheet_Rev.1.0_01July2008.pdf</a>                         |    |
| 4.5. ábra: A „nyers” hőmérséklet és légnyomás adatok kiolvasásának a szekvencia diagramja .....                                                                                                     | 16 |
| Forrás: <a href="http://www.adafruit.com/datasheets/BMP085_DataSheet_Rev.1.0_01July2008.pdf">http://www.adafruit.com/datasheets/BMP085_DataSheet_Rev.1.0_01July2008.pdf</a>                         |    |
| 4.6. ábra: BMP-085 érzékenysége a táblázata .....                                                                                                                                                   | 17 |
| Forrás: <a href="http://www.adafruit.com/datasheets/BMP085_DataSheet_Rev.1.0_01July2008.pdf">http://www.adafruit.com/datasheets/BMP085_DataSheet_Rev.1.0_01July2008.pdf</a>                         |    |
| 4.7. ábra: Sparkfun Razor 9DOF IMU .....                                                                                                                                                            | 17 |
| Forrás: <a href="https://www.sparkfun.com/products/10736">https://www.sparkfun.com/products/10736</a>                                                                                               |    |
| 4.8. ábra: I2C adatátvitel.....                                                                                                                                                                     | 21 |
| Forrás: <a href="http://elegantcircuits.com/2014/08/15/ssd1306-to-atmega328p-using-i2c-protocol/">http://elegantcircuits.com/2014/08/15/ssd1306-to-atmega328p-using-i2c-protocol/</a>               |    |
| 4.9. ábra: SPI adatátvitel.....                                                                                                                                                                     | 22 |
| Forrás: <a href="http://www.hobbielektronika.hu/cikkek/kommunikacio_alapjai_-_soros_adatvitel.html?pg=4">http://www.hobbielektronika.hu/cikkek/kommunikacio_alapjai_-_soros_adatvitel.html?pg=4</a> |    |
| 4.10. ábra: Adatátvitel módja .....                                                                                                                                                                 | 23 |
| Forrás: <a href="http://www.hobbielektronika.hu/cikkek/kommunikacio_alapjai_-_soros_adatvitel.html?pg=4">http://www.hobbielektronika.hu/cikkek/kommunikacio_alapjai_-_soros_adatvitel.html?pg=4</a> |    |
| 4.11. ábra: UART kommunikáció menete.....                                                                                                                                                           | 24 |
| Forrás: <a href="http://www.hobbielektronika.hu/cikkek/egyszeru_soros_kommunikacio_avr-rel_uart.html">http://www.hobbielektronika.hu/cikkek/egyszeru_soros_kommunikacio_avr-rel_uart.html</a>       |    |
| 5.1. ábra: Vitorlacsörlő szerv.....                                                                                                                                                                 | 25 |
| Forrás: <a href="http://hobbyking.com">http://hobbyking.com</a>                                                                                                                                     |    |
| 5.2. ábra: Szervomotor csatlakozók.....                                                                                                                                                             | 25 |
| Forrás: <a href="http://kecskemetircmodell.blog.hu/2011/03/22/hogyan_mukodnek_a_szervok">http://kecskemetircmodell.blog.hu/2011/03/22/hogyan_mukodnek_a_szervok</a>                                 |    |
| 5.3. ábra: Szervo felépítése.....                                                                                                                                                                   | 25 |
| Forrás: <a href="http://kecskemetircmodell.blog.hu/2011/03/22/hogyan_mukodnek_a_szervok">http://kecskemetircmodell.blog.hu/2011/03/22/hogyan_mukodnek_a_szervok</a>                                 |    |
| 5.4. ábra: Szervo mechanika.....                                                                                                                                                                    | 27 |
| Forrás: <a href="http://kecskemetircmodell.blog.hu/2011/03/22/hogyan_mukodnek_a_szervok">http://kecskemetircmodell.blog.hu/2011/03/22/hogyan_mukodnek_a_szervok</a>                                 |    |
| 5.5. ábra: Szervo elektronikai módosítás.....                                                                                                                                                       | 27 |
| Forrás: <a href="http://kecskemetircmodell.blog.hu/2011/03/22/hogyan_mukodnek_a_szervok">http://kecskemetircmodell.blog.hu/2011/03/22/hogyan_mukodnek_a_szervok</a>                                 |    |
| 6.1. ábra: Sorszinkron jel.....                                                                                                                                                                     | 29 |
| Forrás: <a href="http://www.maximintegrated.com/en/app-notes/index.mvp/id/734">http://www.maximintegrated.com/en/app-notes/index.mvp/id/734</a>                                                     |    |
| 6.2. ábra: Sor és képmegjelenítés menete.....                                                                                                                                                       | 30 |

|                                                                                                                                                                                                      |    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Forrás: <a href="http://www.maximintegrated.com/en/app-notes/index.mvp/id/734">http://www.maximintegrated.com/en/app-notes/index.mvp/id/734</a>                                                      |    |
| 6.3. ábra: LM1881 IC által leválasztott szinkronimpulzusok.....                                                                                                                                      | 30 |
| Forrás: <a href="http://www.ti.com/lit/ds/symlink/lm1881.pdf">http://www.ti.com/lit/ds/symlink/lm1881.pdf</a>                                                                                        |    |
| 6.4. ábra: LM1881 gyári kapcsolási rajza.....                                                                                                                                                        | 31 |
| Forrás: <a href="http://www.ti.com/lit/ds/symlink/lm1881.pdf">http://www.ti.com/lit/ds/symlink/lm1881.pdf</a>                                                                                        |    |
| 6.5. ábra: Ellenállásosztó .....                                                                                                                                                                     | 32 |
| Forrás: Saját ábra                                                                                                                                                                                   |    |
| 6.6. ábra: Ellenállásosztó szimuláció.....                                                                                                                                                           | 33 |
| Forrás: Saját ábra                                                                                                                                                                                   |    |
| 6.7. ábra: Feszültségosztó .....                                                                                                                                                                     | 34 |
| Forrás: Saját ábra                                                                                                                                                                                   |    |
| 6.8. ábra: LM1881 módosított kapcsolás .....                                                                                                                                                         | 35 |
| Forrás: <a href="http://www.ti.com/lit/ds/symlink/lm1881.pdf">http://www.ti.com/lit/ds/symlink/lm1881.pdf</a>                                                                                        |    |
| 6.9. ábra: Szinkronjel leválasztó modul.....                                                                                                                                                         | 35 |
| Forrás: Saját ábra                                                                                                                                                                                   |    |
| 6.10. ábra: Horizontális szinkronjel.....                                                                                                                                                            | 36 |
| Forrás: <a href="http://www.maximintegrated.com/en/app-notes/index.mvp/id/734">http://www.maximintegrated.com/en/app-notes/index.mvp/id/734</a>                                                      |    |
| 6.11. ábra: TIMER4 idődiagramja.....                                                                                                                                                                 | 36 |
| Forrás: <a href="http://www.st.com/web/en/resource/technical/document/reference_manual/DM00031020.pdf">http://www.st.com/web/en/resource/technical/<br/>document/reference_manual/DM00031020.pdf</a> |    |
| 6.12. ábra: Horizontális és vertikális szinkronjelek.....                                                                                                                                            | 37 |
| Forrás: <a href="http://www.maximintegrated.com/en/app-notes/index.mvp/id/734">http://www.maximintegrated.com/en/app-notes/index.mvp/id/734</a>                                                      |    |
| 6.13. ábra: Fehér képernyő és fekete-fehér képernyő.....                                                                                                                                             | 38 |
| Forrás: Saját ábra                                                                                                                                                                                   |    |
| 6.14. ábra: Karakterkódolás.....                                                                                                                                                                     | 39 |
| Forrás: <a href="http://www.batsocks.co.uk/readme/art_SerialVideo_4.htm%29">http://www.batsocks.co.uk/readme/art_SerialVideo_4.htm%29</a>                                                            |    |
| 6.15. ábra: OSD kijelző minta .....                                                                                                                                                                  | 40 |
| Forrás: <a href="http://www.rcgroups.com/forums/showthread.php?t=1505348">http://www.rcgroups.com/forums/showthread.php?t=1505348</a>                                                                |    |
| 6.16. ábra: Bitmap .....                                                                                                                                                                             | 40 |
| Forrás: Saját ábra                                                                                                                                                                                   |    |
| 6.17. ábra: Kör és vonalrajzolás eredménye.....                                                                                                                                                      | 41 |
| Forrás: Saját ábra                                                                                                                                                                                   |    |
| 6.18. ábra: Kör és vonalrajzolás eredménye.....                                                                                                                                                      | 42 |

Forrás: <http://www.mkstation.de/osd-k-82.html>

6.19. ábra: Feliratozás külső videojelre eredménye .....43

Forrás: Saját ábra

6.20. ábra: Feliratozás külső videojelre eredménye (2).....43

Forrás: Saját ábra

6.21. ábra: OSD modul, kamera, mikrokontroller .....44

Forrás: Saját ábra

6.22. ábra: Boascam 5.8Ghz-es 200mW-os adó vevő egység.....44

Forrás: Saját ábra

6.23. ábra: Sajáttervezésű OSD modul kapcsolási rajza .....46

Forrás: Saját ábra

6.24. ábra: Saját tervezésű OSD modul nyákterve .....47

Forrás: Saját ábra

6.25. ábra: Saját tervezésű OSD modul kimaratott nyáklemeze .....47

Forrás: Saját ábra

6.26. ábra: Saját tervezésű OSD modul.....47

Forrás: Saját ábra

7.1. ábra: Távolság mérés példa .....48

Forrás: Saját ábra

7.2. ábra: Célpont irányának meghatározása példa .....50

Forrás: Saját ábra

7.3. ábra: Bólintási szög meghatározása.....50

Forrás: Saját ábra

7.4. ábra: A Föld (E=Earth) és a test (B=Body) koordinátarendszere.....51

7.5. ábra: Repülőgép mozgása.....52

Forrás: <http://users.atw.hu/smodell/info/figurak/1alapok.htm>

7.6. ábra: Yaw, Roll, Pitch fordulás .....52

Forrás: Saját ábra

7.7. ábra: Kamera és repülőgép kiindulási helyzete .....56

Forrás: Saját ábra

7.8. ábra: Célpont meghatározása.....57

Forrás: Saját ábra

7.9. ábra: Gép koordinátarendszerének az elforgatása .....57

|                                                                                                    |    |
|----------------------------------------------------------------------------------------------------|----|
| Forrás: Saját ábra                                                                                 |    |
| 7.10. ábra: A célpontra mutató vektor meghatározása a testhez rögzített koordinátarendszerben..... | 58 |
| Forrás: Saját ábra                                                                                 |    |
| 7.11. ábra: Kamera vektor elforgatása az Yb tengely mentén.....                                    | 58 |
| Forrás: Saját ábra                                                                                 |    |
| 7.12. ábra: OSD és a kamerastabilizáló rendszerterve .....                                         | 59 |
| Forrás: Saját ábra                                                                                 |    |
| 7.13. ábra: Próbapanelen elhelyezett szenzorok (shield).....                                       | 60 |
| Forrás: Saját ábra                                                                                 |    |
| 7.14. ábra: Az STM32F4-es Discovery board és a hozzá csatlakoztatott shield.....                   | 60 |
| Forrás: Saját ábra                                                                                 |    |
| 7.15. ábra: Dóm kamera alsó része.....                                                             | 61 |
| Forrás: Saját ábra                                                                                 |    |
| 7.16. ábra: Kamera és a hozzátartozó mozgató mechanika.....                                        | 62 |
| Forrás: Saját ábra                                                                                 |    |
| 7.17. ábra: Dóm kamera .....                                                                       | 62 |
| Forrás: Saját ábra                                                                                 |    |
| 7.18. ábra: Fő szoftvermodul általános diagramja.....                                              | 63 |
| Forrás: Saját ábra                                                                                 |    |
| 7.19. ábra: Magasságmérés folyamatmodellje .....                                                   | 65 |
| Forrás: Saját ábra                                                                                 |    |
| 7.20. ábra: Magasság .....                                                                         | 69 |
| Forrás: Saját ábra                                                                                 |    |
| 7.21. ábra: Légnyomás.....                                                                         | 69 |
| Forrás: Saját ábra                                                                                 |    |
| 7.22. ábra: Hőmérséklet.....                                                                       | 69 |
| Forrás: Saját ábra                                                                                 |    |
| 7.23. ábra: Szűrt és szűretlen Roll adatok .....                                                   | 70 |
| Forrás: Saját ábra                                                                                 |    |
| 7.24. ábra: Szűrt és szűretlen Pitch adatok .....                                                  | 71 |
| Forrás: Saját ábra                                                                                 |    |
| 7.25. ábra: Szűrt és szűretlen Yaw adatok.....                                                     | 71 |

|                                                                                                                                                                                                                           |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Forrás: Saját ábra                                                                                                                                                                                                        |    |
| 7.26. ábra: GPS teszt eredménye .....                                                                                                                                                                                     | 71 |
| Forrás: Saját ábra                                                                                                                                                                                                        |    |
| 7.27. ábra: Célpont lehetséges elhelyezkedése .....                                                                                                                                                                       | 73 |
| Forrás: Saját ábra                                                                                                                                                                                                        |    |
| 7.28. ábra: Gimbal .....                                                                                                                                                                                                  | 74 |
| Forrás: Saját ábra                                                                                                                                                                                                        |    |
| 7.29. ábra: A gimbal behúzott illetve kieresztett állapota .....                                                                                                                                                          | 75 |
| Forrás: Saját ábra                                                                                                                                                                                                        |    |
| 8.1. ábra: Célpont magasságának a meghatározásával járó probléma.....                                                                                                                                                     | 76 |
| Forrás: Saját ábra                                                                                                                                                                                                        |    |
| 8.2. ábra: GOTOP30 adatok alapján előállított modell .....                                                                                                                                                                | 77 |
| Forrás: <a href="http://www.geo.u-szeged.hu/~laci/ab-Geoinfo-tananyag/ch10.html">http://www.geo.u-szeged.hu/~laci/ab-Geoinfo-tananyag/ch10.html</a>                                                                       |    |
| 8.3. ábra: <a href="http://srtm.csi.cgiar.org">http://srtm.csi.cgiar.org</a> oldalon található kezelőfelület .....                                                                                                        | 78 |
| Forrás: Saját ábra                                                                                                                                                                                                        |    |
| 8.4. ábra: Magyarország nyugati részének raszteres képe.....                                                                                                                                                              | 78 |
| Forrás: Saját ábra                                                                                                                                                                                                        |    |
| 8.5. ábra: GRASS GIS console ablaka .....                                                                                                                                                                                 | 79 |
| Forrás: Saját ábra                                                                                                                                                                                                        |    |
| 9.1. ábra: A dóm kamera tesztelése .....                                                                                                                                                                                  | 80 |
| Forrás: Saját ábra                                                                                                                                                                                                        |    |
| 9.2. ábra: Gimbal tesztelése.....                                                                                                                                                                                         | 80 |
| Forrás: Saját ábra                                                                                                                                                                                                        |    |
| 9.3. ábra: X8 Skywalker .....                                                                                                                                                                                             | 81 |
| Forrás: <a href="https://www.hobbyking.com/hobbyking/store/__27132__Skywalker_X_8_FP_V_UAV_Flying_Wing_2120mm.html">https://www.hobbyking.com/hobbyking/store/__27132__Skywalker_X_8_FP_V_UAV_Flying_Wing_2120mm.html</a> |    |
| 9.4. ábra: Óbudai Egyetem által kifejlesztett pilóta nélküli robotrepülőgép.....                                                                                                                                          | 81 |
| Forrás: Saját ábra                                                                                                                                                                                                        |    |
| 9.5. ábra: Dóm kamera elhelyezése .....                                                                                                                                                                                   | 82 |
| Forrás: Saját ábra                                                                                                                                                                                                        |    |
| 9.6. ábra: Dóm kamera tesztkép 1 .....                                                                                                                                                                                    | 83 |
| Forrás: Saját ábra                                                                                                                                                                                                        |    |
| 9.7. ábra: Dóm kamera tesztkép 2 .....                                                                                                                                                                                    | 83 |

|                                                           |    |
|-----------------------------------------------------------|----|
| Forrás: Saját ábra                                        |    |
| 9.8. ábra: Dóm kamera tesztkép 3 .....                    | 83 |
| Forrás: Saját ábra                                        |    |
| 9.9. ábra: Gimbal tesztkép 1. ....                        | 84 |
| Forrás: Saját ábra                                        |    |
| 9.10. ábra: Gimbal tesztkép 2. ....                       | 84 |
| Forrás: Saját ábra                                        |    |
| 9.11. ábra: Gimbal tesztkép 3. ....                       | 84 |
| Forrás: Saját ábra                                        |    |
| 9.12. ábra: Gimbal tesztkép 3. ....                       | 84 |
| Forrás: Saját ábra                                        |    |
| 9.13. ábra: Tesztrepülés OSD nélkül 1. ....               | 84 |
| Forrás: Saját ábra                                        |    |
| 1. táblázat: Egyes eszközök tömege és áramfelvétele ..... | 82 |