

TDK dolgozat

L. Kiss Márton

Kézfejen elhelyezett távirányítóval vezérelhető légpárnás jármű megvalósítása

TDK dolgozat

L. Kiss Márton

Villamosmérnök BSc 3. évfolyam

Konzulens:

Dr. Vásárhelyi József

Egyetemi adjunktus



Miskolci Egyetem

Villamosmérnöki Intézet

Automatizálási és Infokommunikációs Intézeti Tanszék

Tartalomjegyzék

Bevezetés	- 5 -
Szakirodalom	- 6 -
1. Felhasznált eszközök	- 7 -
1.1. Hardverelemek	- 7 -
1.1.1. ZYBO.....	- 7 -
1.1.2. AP SoC	- 8 -
1.1.3. Pmod	- 16 -
1.2. STM32F3fejlesztőkártya.....	- 18 -
1.3. Szoftverelemek.....	- 21 -
1.3.1. Xilinx Platform Studio (XPS).....	- 21 -
1.3.2. Software Development Kit (SDK).....	- 23 -
1.3.3. PlanAhead.....	- 25 -
2. Légpárnás jármű	- 27 -
2.1. Követelmények	- 28 -
2.2. Felépítés	- 29 -
2.3. Légpárnás jármű elektronikus építő elemei	- 31 -
2.4. Perifériák implementálása XPS-ben	- 35 -
2.4.1. Univerzális, paraméterezhető PWM vezérlő	- 39 -
2.4.2. Léptetőmotor vezérlő	- 43 -
3.5. SDK szoftver	- 47 -
3.5.1. Perifériák konfigurálása.....	- 50 -
3.5.2. Megszakításos UART.....	- 51 -
3. Távvezérlő "Jedi kesztyű"	- 53 -
3.1. Követelmények	- 53 -

3.2. Felépítés	- 54 -
3.3. Elektronika	- 56 -
3.4. Szoftver	- 57 -
3.4.1. Menü rendszert szemléltető állapotgép.....	- 59 -
3.4.2. Felhasznált perifériák, és konfigurációjuk.....	- 60 -
Irodalomjegyzék	- 62 -
Ábrák jegyzéke	- 63 -
Rövidítések	- 64 -
Köszönetnyilvánítás.....	- 66 -

Bevezetés

A Miskolci Egyetem villamosmérnök hallgatói közül minden évben vannak olyan vállalkozó szellemű diákok, akik részt vesznek a Magyarok a Marson nemzetközi robotépítő versenyen. Én két éve veszek részt ezen a versenyen csapatommal az A-TEAM-el, amely már évek óta öregbíti az egyetemünk hírnevét ezen a versenyen.

Az idén rendezett Kilencedik Magyar Alkalmazott Mérnöki Tudományok Versenyére (röviden Magyarok a Marson) készült légpárnás jármű, vezérlésének felépítését és működését fogom ismertetni a dolgozatomban. A versenyfeladat egy légpárnás robot és a távvezérlő „Jedi kesztyű” megvalósítása volt. A vezérlő valós időben mozdulat felismerő szenzorok segítségével vezérli a légpárnás járművet. A járművön daru található, amely szintén a Jedi kesztyűvel irányítható. A daru különböző magasságú dobozokra helyez kockákat („Tatuin kocka”).

A távvezérlő egy kesztyű, amire egy giroszkóp szenzor lett elhelyezve. A giroszkóp érzékeli a kézfej mozgásait. A szenzor által szolgáltatott adatok bluetoothon keresztül jutnak el a légpárnás járműre. Az ujjak végén elhelyezett nyomógombok segítségével tudunk különböző menük és funkciók között választani. A Jedi kesztyű vezérlésére egy STM32F303 mikrovezérlős kártyát választottam, mert ez tartalmazza a szükséges giroszkóp szenzort. A kártyán egy 32 bites ARM architektúrájú mikrovezérlő található, ami támogatja a bluetooth modul által használt kommunikációs interfészt. Az ARM architektúrás mikrovezérlők a beágyazott rendszerekben, ezen belül is főleg az telefonokban elég elterjedt, így értékes szakmai tapasztalatnak ígérkezett.

A légpárnás járművet és a darut mozgató motorok (szervo motorok, egyenáramú szénkefés motorok, léptetőmotor) nagy száma miatt előnyösnek tartottam FPGA (programozható logikai áramkör) használatát, amelyben különálló egymástól függetlenül és párhuzamosan működő logikai egységek hozhatók létre. A vezérlésre egy ZYBO kártyát használtam, ami tartalmaz egy rendszert a chippen (FPGA + processzor) áramkört. A ZYBO-n lévő processzor és programozható logika egy sínrendszerrel van összekötve, ami lehetővé teszi, hogy a processzor perifériáit megírjuk meg VHDL hardver leíró programozási nyelven, így a légpárnás járművet működtető motorokat saját perifériák vezérlik. Ezen vezérlő perifériák tervező rendszerét, tervezési lépéseit és felépítésüket részletesen ismertetem majd dolgozatomban.

Szakirodalom

A „Jedi” kesztyű és a Légpárnás jármű vezérlő egységében egyaránt megtalálható az ARM architektúrájú processzor mag. De egyáltalán mi is az ARM?

Az ARM cég 1990-ben alakult az Acorn Computers, Apple Computer és VLSI Technology közös vállalkozásaként. Advanced RISC Machines Ltd. (ARM) processzor magok tervezésével foglalkozik. Közvetlen gyártással nem foglalkozik, a tervezett architektúrákat és processzorokat licenseli és ezeket árulja különböző chipgyártó cégeknek, mint például a STMicroelectronics és a Xilinx. Ezek a cégek perifériákat ültetnek a processzor mag köré és így adják el mikrokontrollerek vagy akár kész fejlesztő kártya formájában. A különböző processzorokat családokba rendezik tulajdonságaik alapján.

Mivel rengeteg ARM család létezik, ezért nem mutatom be mindegyiket, hanem csak az újabbakat, és amelyeket a feladat megvalósításában használt vezérlők is tartalmaznak:

- Cortex M család: alacsony fogyasztású, kevert jelű processzor mag. Háztartási elektronikában és egyéb operációs rendszer futtatását nem igénylő alkalmazásokban használják.
- Cortex A: beágyazott operációs rendszerekkel kompatibilis pl.: Android, Linux.
- Cortex R: real-time processzor család, autóiparba és egyéb olyan helyeken használják, ahol fontos a real-time kommunikáció.

Cortex M és a Cortex A családból származó processzorokat alkalmaztam a vezérlési feladatok megvalósítására, ezen belül is Cortex A9-et, amely ARMv7A architektúrájú, illetve Cortex M4-et amely, ARMv7E-M architektúrán alapul.

Ezek a processzorok RISC típusúak, azaz csökkentett utasítás készletűek.

Az ARM nem csak a processzormagokat, hanem a hozzá tartozó sínrendszereket is megtervezi. A leggyakrabban alkalmazott sínrendszer az AMBA, amire az a jellemző, hogy tartalmaz lassabban és gyorsabban működő vezetékek rendszereket. A legújabb verziója az AMBA 4.0. Az AXI sínrendszert is az ARM cég fejlesztése.

1. Felhasznált eszközök

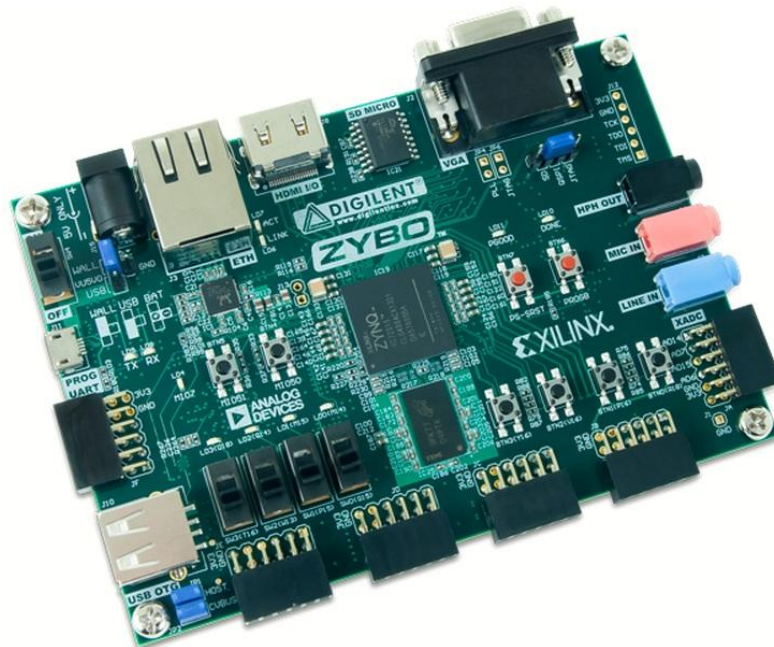
Ebben a fejezetben bemutatom a szoftver és hardver elemeket, amelyeket a légpárnás jármű és a „Jedi kesztyű” vezérlésének megvalósításakor felhasználtam.

1.1. Hardverelemek

Ez a fejezet a felhasznált hardver elemekről, ezek felépítéséről és tulajdonságairól szól. Két főbb hardver elem mutatok be az egyik a ZYBO kártya a másik pedig az STM 32F303.

1.1.1. ZYBO

A ZYBO (ZYnqBOard) funkcióban gazdag, alacsony szintű beágyazott rendszer fejlesztői platform kártya, aminek központi eleme a Xilinx Zynq-7000 család legkisebb tagja Z-7010 típusú áramkör. A Z-7010 a Xilinx All Programmable System-on-Chip(AP SoC) architektúrán alapul, amely két ARM Cortex-A9 processzormagot és Xilinx 7. sorozatú programozható logika magot (FPGA-t) tartalmaz egy integrált áramköri lapkán. A következő kép (1. ábra) a ZYBO kártyát ábrázolja.



1. ábra - ZYBO kártya

A ZYBO általános jellemzői:

- Az üzemeléshez 5 V feszültséget igényel és teljes kihasználtság esetén maximum 2,5 A áramot vesz fel a hálózathoz.
- Xilinx JTAG programozóval és USB-n keresztül is programozható
- Tartalmaz 6 darab Pmod csatlakozót, amely külső perifériák csatlakoztatására szolgál
- Analóg és digitális perifériákkal egyaránt el van látva: HDMI, Ethernet, Audio ki- és bemenet, 16-bit/pixel VGA port
- A kártyán található 512 MB DDR3-mas memória és memóriakártya olvasó, így alkalmas beágyazott operációs rendszerek futtatására

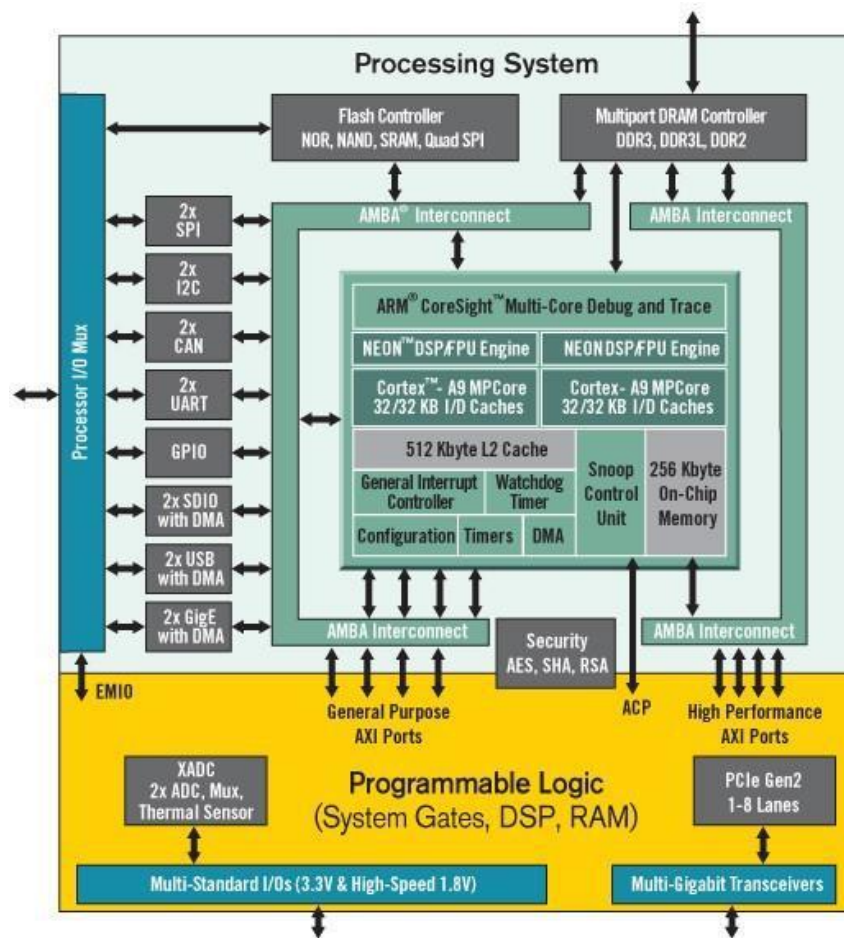
A Xilinx szoftverek támogatják a ZYBO kártyával történő fejlesztést. A légpárnás jármű szoftverének tervezése során három programot használtam. Az egyik az XPS, amely segítségével a PS és PL rendszerek hardver szinten konfigurálhatóak. A másik az SDK, ami arra szolgál, hogy a C nyelven megírt szoftvert lefordítsa és letöltse a processzorra és végül a PlanAhead-et, ami kiindulópontja a teljes beágyazott rendszer fejlesztésnek és ennek segítségével megnyithatóak a már említett szoftverek és átmenetet képez közöttük. A tervező rendszer működését az 1.2 Szoftverelemek című részben részletesen kifejtem.

1.1.2. AP SoC

SoC (System-on-Chip) - magyarul egylapkás rendszer - egy olyan integrált áramkör, aminek része egy vagy több processzormag és ehhez csatlakozó alrendszerek, perifériák. A SoC áramköröket gyakran használják beágyazott rendszerekben.

A Zynq AP SoC két különböző alrendszerre osztható: Processzor Rendszerre (PS – Processing System) és Programozható Logikára (PL – ProgrammableLogic). A PL ideális nagysebességű logikai vezérlő, míg a PS támogatja a szoftver rutinokat és operációs rendszereket. Ez azt jelenti, hogy bármilyen beágyazott rendszer megvalósítható, ha kihasználjuk a PS és PL közötti kapcsolatot. Ezt a kapcsolatot az AXI (Advanced eXtensible Interface) ipari szabványos sínrendszer biztosítja.

Az lenti ábrán (2. ábra) megfigyelhetők a különböző alrendszerek, a PS világoszöld, a PL pedig sárga háttérben látható.



2. ábra - Zynq AP SoC architektúra [1.]

A Zynq AP SoC architektúra PS alrendszere többek között a következő periféria elemeket tartalmazza:

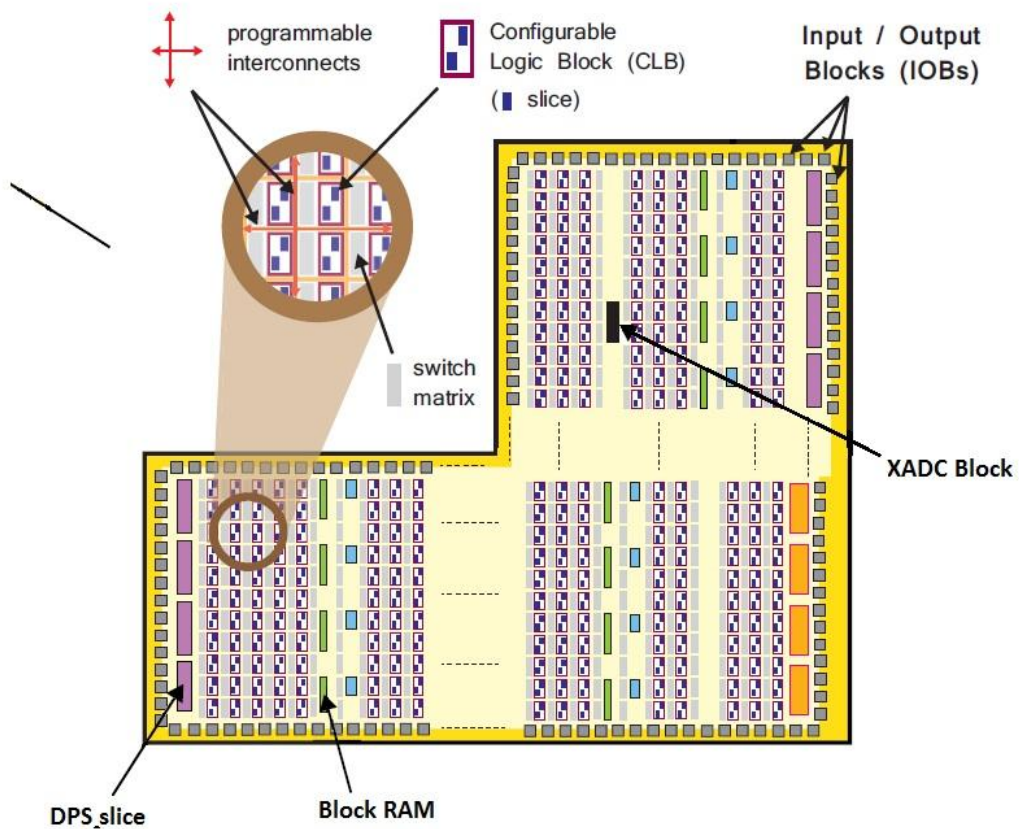
- 650Mhz kétmagos Cortex-A9 processzor
- DDR3 memória vezérlő interfész 8 DMA csatornával
- Nagy-sávszélességű periféria vezérlők: 1G Ethernet, USB 2.0, SDIO
- Alacsony-sávszélességű periféria vezérlők: SPI, UART, CAN, I²C

A PL egy Atrix 7 FPGA-val egyenértékű programozható logikai egység, topológiáját tekintve hierarchikus felépítésű.

FPGA általános jellemzői:

- külön álló egymástól független logikai blokkok hozhatóak létre, amelyek párhuzamosan működnek
- szimmetrikus és hierarchikus felépítésű
- többszintű logikát valósítanak meg
- SRAM alapú

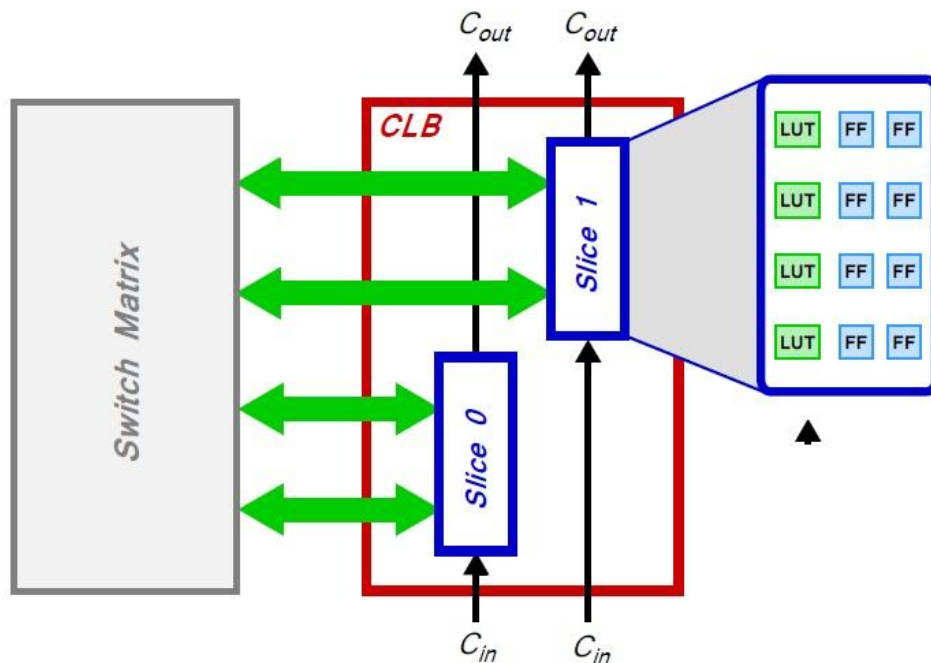
A kép (3. ábra) a PL architektúrát és építő elemeit szemlélteti



3. ábra - PL architektúra

PL építőelemei:

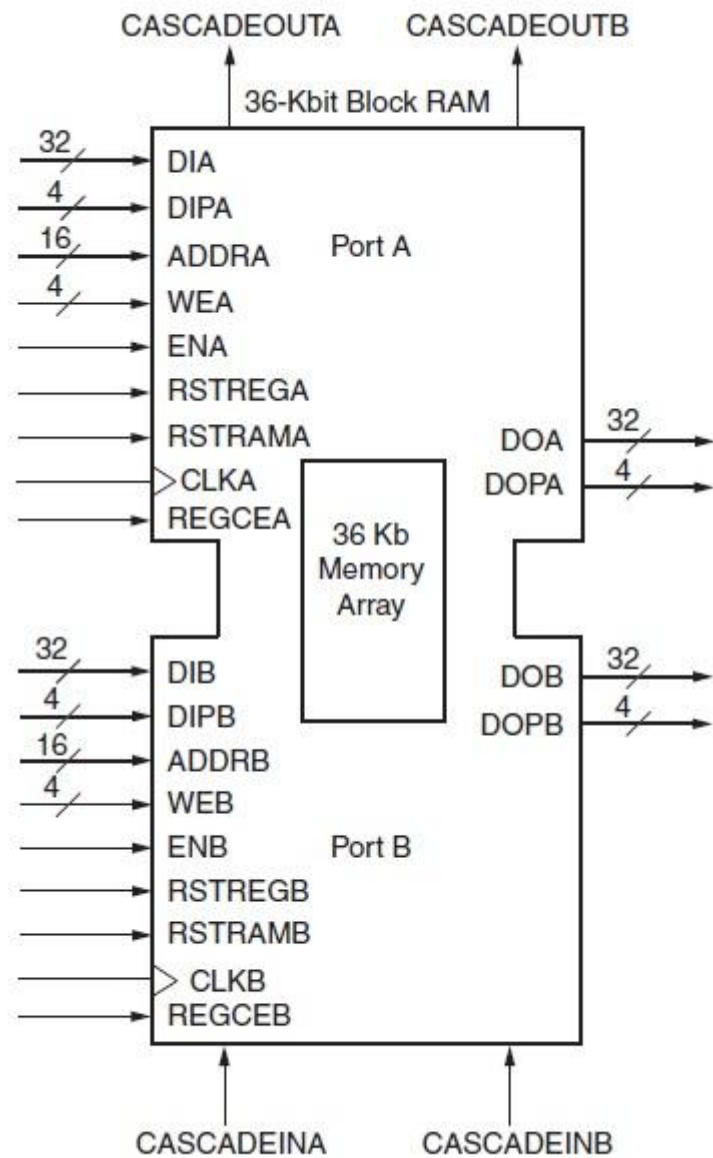
Configurable Logic Block (CLB) – programozható logikai blokk – a felhasználó által igényelt logikai függvényeket, funkciókat képes megvalósítani. Az egyes FPGA típusok között eltérő felépítésű lehet. Ahogy a következő képen (4. ábra - CLB) is látható egy programozható logikai blokk két szeletet – slice-t – tartalmaz, amelyekben különböző logikai elemek vannak integrálva. Az egyik leglényegesebb logikai elem a hat bemenetű Look-up-table (LUT), - keresőtábla – amely akár hat változós logikai függvényeket is képes megvalósítani és akár 16*1 bites memóriaként is használható. Ezen kívül tartalmaz még D típusú flip-flop-okat és Multiplexereket és egyéb helyi huzalozási erőforrásokat. Carry logika a jelátvitel és az összeadás - kivonás gyorsítását segíti elő. A következő képen (4. ábra) a CLB egyszerűsített tömbvázlata látható.



4. ábra - CLB

Blokk RAM egyenként 36Kb méretűek. Minden egyes BRAM két teljesen független porttal rendelkező 18Kb-es memória blokkból épül fel. Ezek külön és egyszerre is címezhetők, sőt több BRAM is összevonható (kaszkádozható) egy nagyobb memóriatömbbé. Beépítve tartalmaz FIFO vezérlő interfészt, ami képes szinkron és aszinkron működésre. Minden 18Kb-es memória blokk rendelkezik paritás bitek

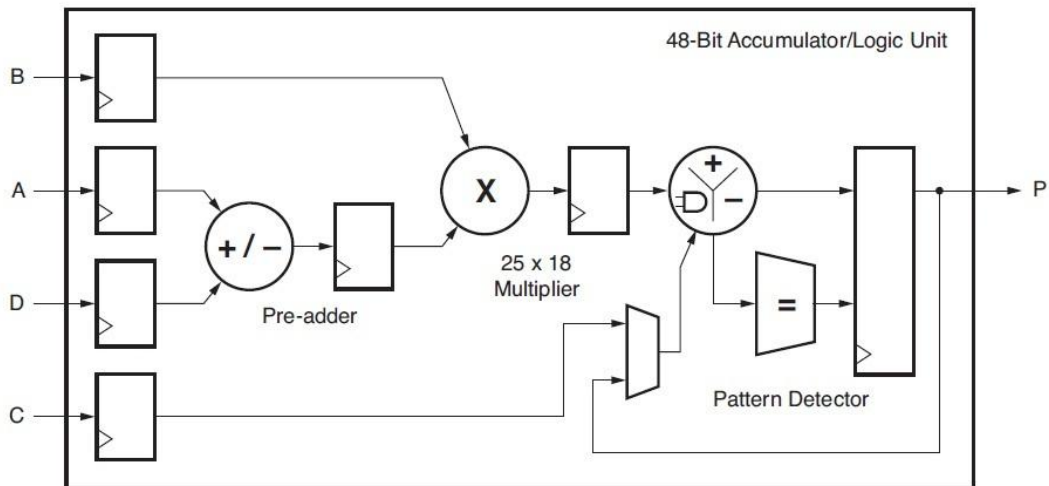
kezelésére kialakított ki és bemeneti portokkal (DIP, DOP). A következő kép (5. ábra) a Blokk RAM tömbvázletát, lábkivezetéseit szemlélteti



5. ábra - Dupla portos BRAM

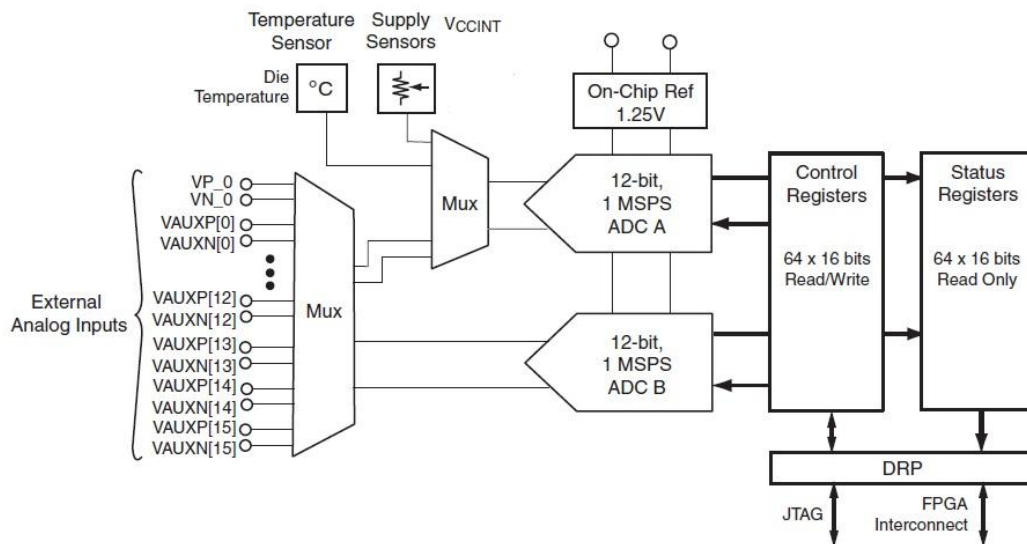
Órajel felügyeleti egység több különálló részből épül fel, az egyik kulcsfontosságú eleme a PLL (fázis zárt hurok) a másik pedig az MMCM (kevert jelű órajel feldolgozó egység), amelyek képesek az órajeleket jitter szűrt, leosztani és felszorozni is. Az átalakított jelek alacsony jitter hibát tartalmaznak. A jelszintek stabilitásáért integrált órajel bufferek (erősítők) felelnek.

DSP (digitális jel processzor) slices lebegőpontos alapműveletek gyors elvégzését segíti. A műveletek elvégzését a beépített aritmetikai és logikai egység (ALU), valamint a teljes összeadó, kivonó és szorzó egységek segítik elő. Ezek a DSP szeletek kaszkádozhatóak, ami azt jelenti, hogy egymás után kötve bonyolultabb kifejezések, egyenletek számítását gyorsítja meg. A DPS szelet hardver felépítését a következő ábra (6. ábra) mutatja be.



6. ábra - DPS blokkvázlat

Analóg Digitális átalakító (XADC) az elektronika analóg ágával való kapcsolatlétesítést segíti elő. A beérkező analóg jel - szűrés után - egy 12 bites és 1 MSPS közepes mintavételezési sebességű AD-n keresztül jut el az állapot regiszterbe (Status Registers), ahonnan az FPGA ki tudja olvasni az analóg mért értékeket. ADC-hez csatlakozik egy chipre integrált hőmérséklet és feszültség mérő szenzor, valamint egy 1,25V-os referencia feszültség, amely a mérések alapjául szolgál. A lent látható ábrán (7. ábra) a XADC tömbvázlata látható



7. ábra - ADC blokkvázlat

IOB (Input Output Block) a ki és bemeneti egység széleskörűen konfigurálható a különféle technológiai illesztésekhez 1,2 V-tól 3,3 V-ig és a kimeneti ellenállást is digitálisan állíthatjuk be kis teljesítményre vagy nagy sebességű működésre.

AXI ipari szabványos sínrendszer

AXI sínrendszer legújabb kiadott verziója az AXI4, amely kompatibilis az ARM cég által fejlesztett AMBA 3.0 sínrendszerrel. Sok perifériát és egyéb eszközt fejlesztettek ki e kapcsolatot kihasználva az iparban. Az AXI sínrendszer 3 féle busz protokoll szerint tud működni:

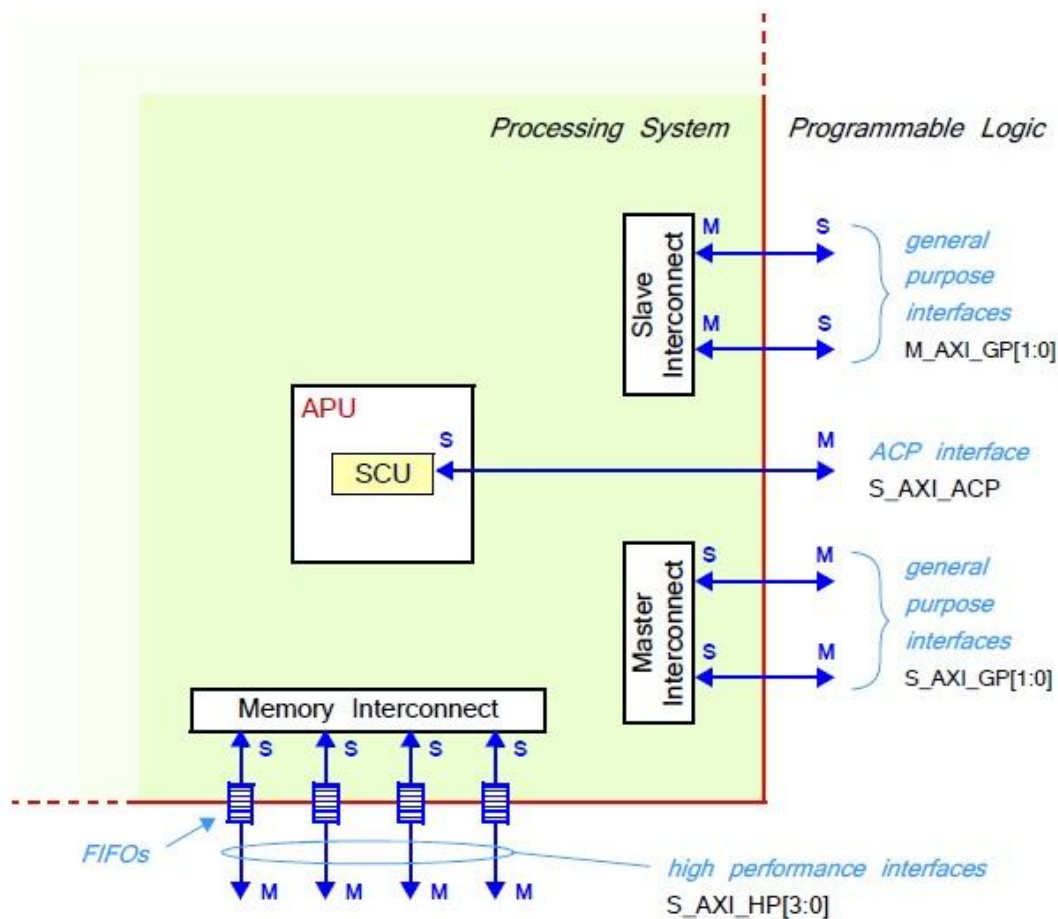
- AXI4-Lite: Regisztert tartalmazó protokoll, ahol a processzor tudja írni és olvasni a regisztereket. Ezt a típust használtam fel a légpárnás jármű motorvezérlő perifériáinak implementálása során.
- AXI4: Memóriával rendelkező protokoll, amely támogatja a csoportos memória hozzáférést. Ez azt jelenti, hogy a processzor egy memória rekeszt címez, ezután a szomszédos rekeszekből a memória szolgáltatja az adatokat.
- AXI4-Stream: Folyamatos adat továbbításra képes, nem rendelkezik belső regiszterekkel, memóriákkal.

A kapcsolat a PS és PL között kilenc AXI interfészen keresztül valósul meg, melyeknek mindegyike több vezetékből álló sínrendszer. Ezek a PL-t kötik össze a PS rendszerben

lévő AXI huzalozási erőforrásokkal azaz interconnectekkel. Itt fontos meghatározni az alábbi két fogalmat:

- **Interfész:** pont-pont összeköttetést jelent, amely csak huzal erőforrásként működik.
- **Interconnect:** magyarul huzalozási erőforrást jelent. Ez vezérli a különböző AXI interfészeket és kapcsolatot teremt az AMBA sínrendszerrel.

A következő képen (8. ábra) látható az AXI interfészek csatlakozása az AXI huzalozási erőforrásokhoz. Látható, hogy az ACP interfész nem az AXI huzalozási erőforráshoz, hanem az APU-n (Application Processing Unit) belül az SCU (Snoop Control Unit) központi cache memória elérést és sínrendszer felügyeletet végző egységbe csatlakozik. Az (8. ábra) ábra alatt lévő táblázat (1. táblázat) az AXI interfészeket tartalmazza rövid lírással.



8. ábra - AXI sínrendszer csatlakozása a PS rendszerbe

Interfész név	Interfész leírás	Mester oldal	Szolga oldal
M_AXI_GP0	Általános felhasználású	PS	PL
M_AXI_GP1		PS	PL
S_AXI_GP0	Általános felhasználású	PL	PS
S_AXI_GP1		PL	PS
S_AXI_ACP	Busz rendszeren történő adat áramlásról szolgálta adatokat a processzor felé	PL	PS
S_AXI_HP0	Nagy teljesítményű portok, támogatott a gyorsított memória elérés és úgynevezett AFI-val AXI Fifo Interfészsel bufferrel is rendelkezik.	PL	PS
S_AXI_HP1		PL	PS
S_AXI_HP2		PL	PS
S_AXI_HP3		PL	PS

1. táblázat - AXI port

Így épül fel az AXI ipari szabványos sínrendszer amely, kapcsolatot teremt a PS és PL rendszerek között. A segítségével megvalósítható kis, közepes és nagy sebességű kapcsolat, valamint a nagy teljesítményű portokon keresztül gyorsan, nagy mennyiségű adat továbbítható.

1.1.3. Pmod

A Pmod egy szabványos 2*6 lábbal rendelkező derékszögű csatlakozó, ahogy a (9. ábra) ábra is szemlélteti. A 6-os és 12-es lábak 3,3V-os tápfeszültséget, az 5-ös és 11-es lábak pedig a föld potenciált szolgáltatnak. A maradék 8 láb jelek csatlakozására szolgál.

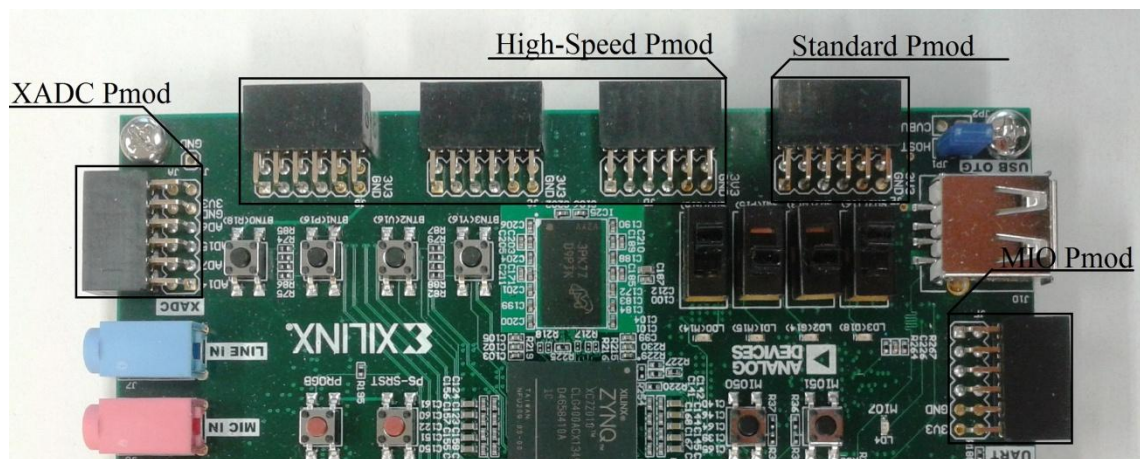


9. ábra - Pmod felépítése

Aszerint, hogy az architektúra mely fő részéhez vannak kötve a Pmod csatlakozók között megkülönböztetünk:

- Processzor rendszerhez csatlakozó:
 - MIO Pmod
- Programozható logikához csatlakozó:
 - Standard Pmod
 - High-speedPmod
 - XADC Pmod

A következő kép (10. ábra) a ZYBO kártyán integrált Pmod csatlakozat jelöltem meg típus szerint.

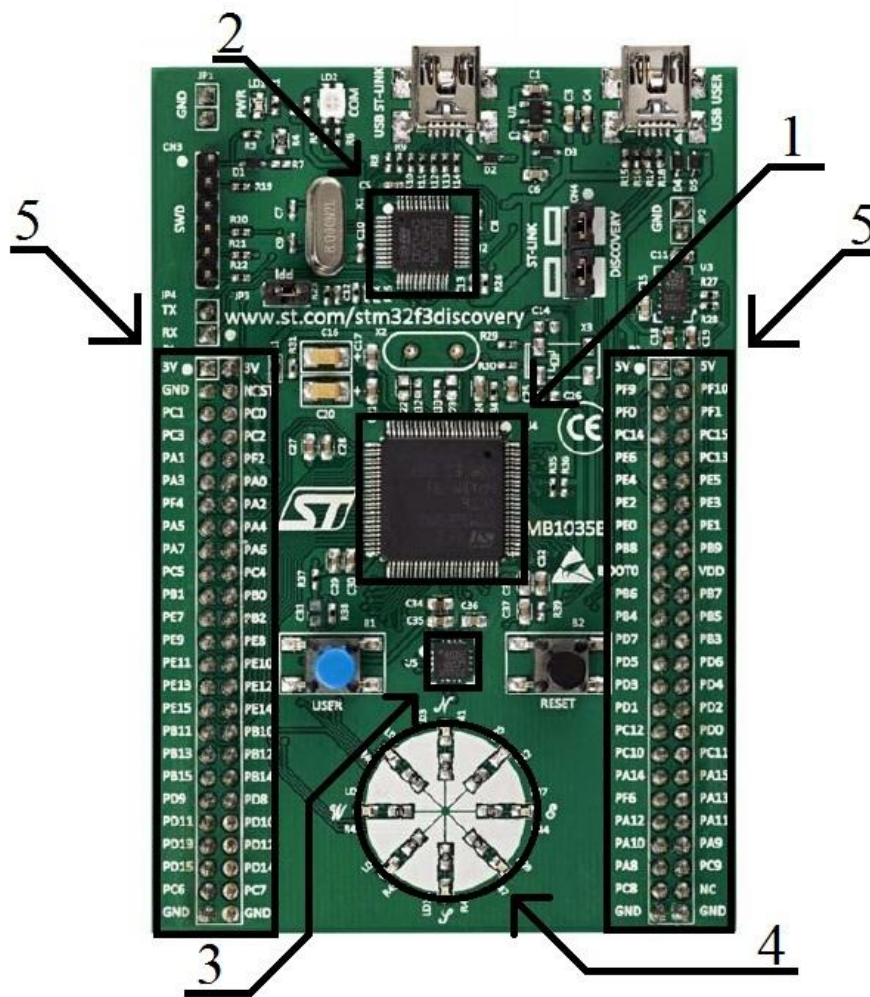


10. ábra - Pmodok a ZYBO kártyán

A programozható logikához csatlakozó Pmodok az IOB-k kivezetései. Az elektronikai egységek és egyéb külső perifériák illesztése a ZYBO-hoz ezeken a csatlakozókon keresztül történik.

1.2. STM32F3fejlesztőkártya

A következő ábrán (11. ábra) a „Jedi kesztyűn” használt mikrovezérlős kártya látható. A kártyán lévő főbb elemeket kiemeltem és számokkal jelöltem. Ebben a részben ezek a főbb hardverelemek kerülnek bemutatásra.



11. ábra - STM32F3 fejlesztőkártya

1. STM32F303VCT6 ARM Cortex M4 mikrovezérlő

A kártyán lévő mikrovezérlő ARM architektúrájú, amelyet a világon a legtöbb beágyazott rendszerben alkalmaznak. A Cortex M4 az áramköri családra utal, jellemzője, hogy kis fogyasztású, kompakt, egyszerűen programozható.

Általános jellemzők:

- Stop módban mindössze 5,7 μ A áramfelvétel
- 32 bites AHB sínrendszer
- 256 KbFlash memória
- 48 Kb SRAM
- 72 MHz órajel frekvencia
- 6 külső port

Főbb perifériák:

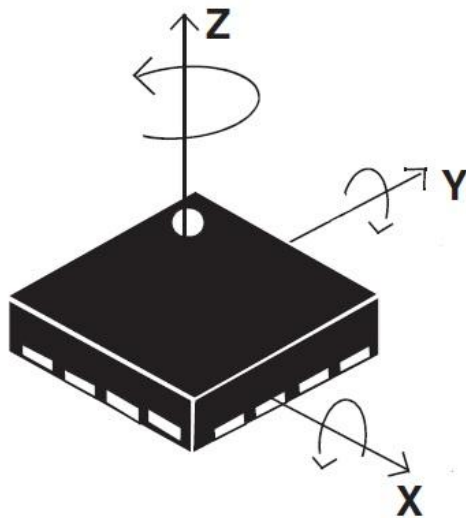
- 4 csatornás 12-bites Digitális Analóg Átalakító 5 – 18 MSPS mintavételezési sebességgel
- 13 Időzítő, egy 32 bites és tizenkét 16 bites
- Kommunikációs interfészek: UART, USART, SPI, I²C, CAN Bus
- Több szintű megszakítás vezérlő interfész

2. STM-LINK/V2 programozó, hibajavító interfész

A kártyán lévő perifériák közül ez az egyik legfontosabb. Ezen keresztül tudjuk programozni egy egyszerű USB kábel segítségével a kártyát. Lehetőségünk van szoftveres hibajavításra is, ami azt jelenti, hogy a mikrovezérlő szoftvere vizsgálható futás közben lépésről lépésre, megfigyelhetők belső változók illetve töréspontok helyezhetők el. Ez a hibajavítási lehetőség egyszerűsíti a szoftverek szemantikai hibáinak javítását, sőt segít megérteni a processzor működését is.

3. Giroszkóp

A „Jedi kesztyű” tervezésekor merült fel azaz ötlet, hogy olyan mikrovezérlős fejlesztő kártya legyen a vezérlő, amin giroszkóp szenzor található, ezért is esett a választás az STM32F303 kártyára. A következő ábra (12. ábra) a szenzor koordináta-rendszerét ábrázolja

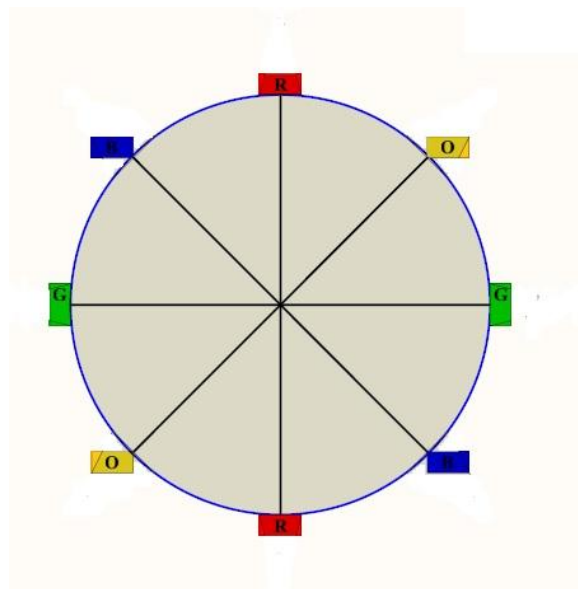


12. ábra - Szenzor koordináta rendszere

A szenzor típusa L3GD20, amely a 3 tengelyre eső vetület vektorok formájában ad értékeket az elmozdulásról. Ezt a nyers értéket matematikai számítások után tudjuk szög mértékben megadni.

4. LED-ek

A LED a távvezérlő „Jedi kesztyű” menüinek vizuális megkülönböztetésére lettek felhasználva. A képen (13. ábra) látható színben és elrendezésben helyezkednek el a kártyán.



13. ábra - LED-ek

5. A mikrovezérlő port kivezetései

A 7. ábrán látható port kivezetéseket az abc nagy betűivel jelölik: PA, PB, PC, PD, PE, PF. Minden port a PF kivételével 15 fizikai lábkievezéssel rendelkezik a kártya két megjelölt területén. Ezekre a kivezetésekre csatlakoznak a kesztyűn található nyomógombok és a bluetooth modul kommunikációs interfésze.

A port-lábakat különböző módokra lehet konfigurálni:

- Digitális bemenet
- Digitális kimenet
- Analóg mód
- Alternatív funkció

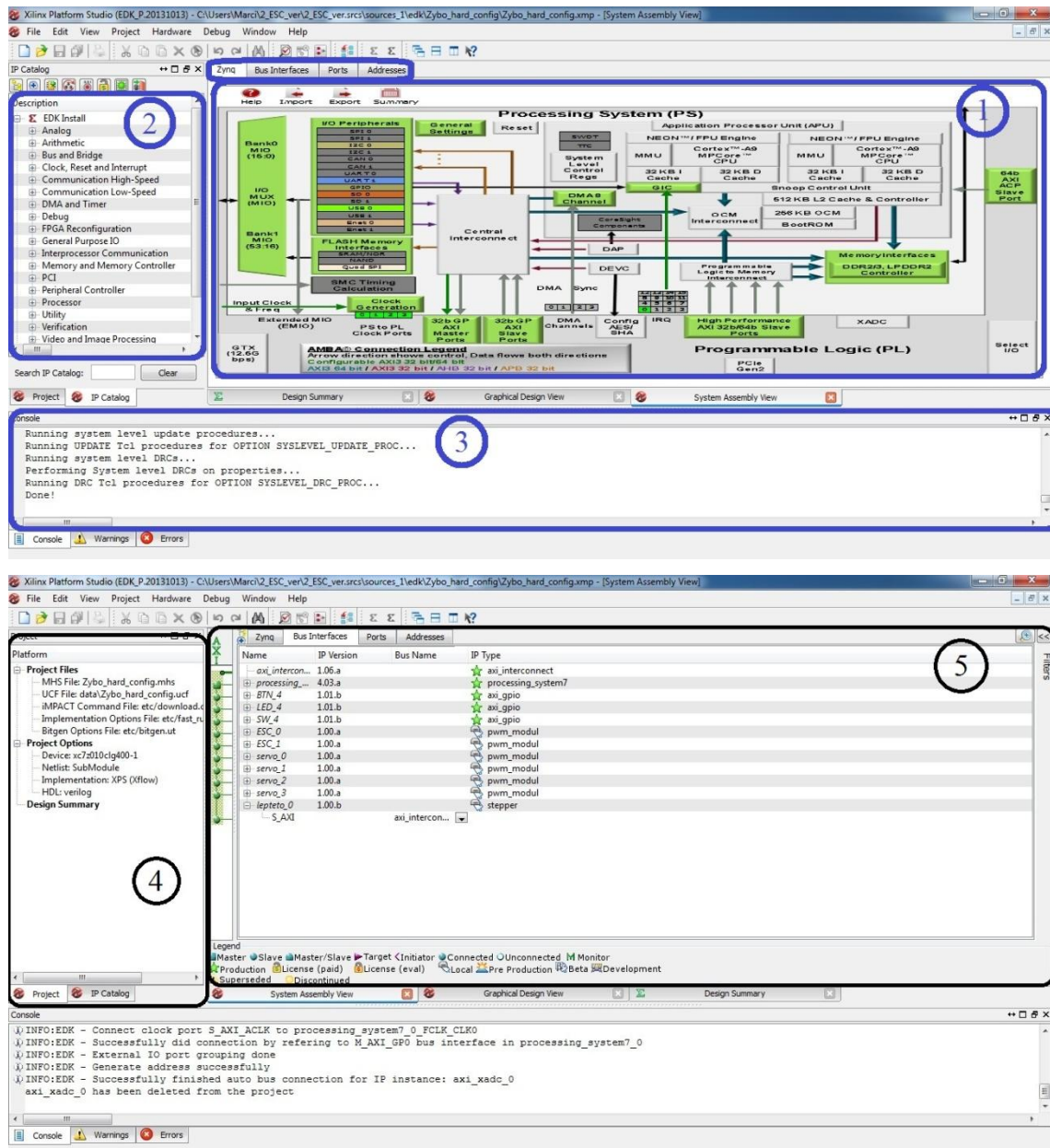
Ezek közül egyet emelnék ki, mégpedig az alternatív funkciót. Ez a mód azt jelenti, hogy bizonyos perifériák vezethetők ki a külső portok lábaira például: az UART Rx (Read Data – adat olvasás) és Tx (Transmitdata – Adat továbbítás) lábát ki lehet vezetni a külső portok lábainak valamelyikére. Sajnos nem vezethető ki periféria akármelyik port lábra, mert a mikrovezérlő tervezése során megszabták a perifériák kivezetéseit. A [4.] számú mellékletben található egy táblázat, ami tartalmazza a port lábak alternatív funkcióját.

1.3. Szoftverelemek

1.3.1. Xilinx Platform Studio (XPS)

Az XPS egy grafikus felülettel rendelkező beágyazott rendszer fejlesztő, ami magában foglal minden olyan eszközt, ami a Xilinx beágyazott hardver és szoftver fejlesztéshez szükséges.

A következő kép (14. ábra) a szoftver felépítését mutatja be. A képeken számokkal jelöltem a szoftver különböző részeit.



14. ábra – XPS ablak

A szoftver jelölt részei:

1. ZynqProcessing System View: a PS perifériák konfigurálására szolgál.
2. IP Catalog: Innen előre VHDL nyelven megírt perifériákat csatolhatunk a PS mellé.
3. Console: A program itt dokumentálja a lefutott folyamatokat.
4. Project InformationArea: A project beállításait és a project fájlokat tartalmazza.

5. Connectivity Panel: Beállíthatjuk, hogy a perifériák kivezetései hova csatlakozzanak az AXI sínrendszeren.

Beágyazott Rendszer fejlesztő főbb eszközei, funkciói:

- Base System Builder (BSB) segítségével konfigurálhatjuk a Zynq PS rendszert és MicroBlaze processzort is implementálhatunk.
 - A MicroBlaze egy VHDL nyelven megírt FPGA-n futó beágyazott processzor.
- Create or Import Peripheral funkció segítségével a saját perifériáinkat csatlakoztatjuk a PS sínrendszeréhez.
- ChipScope Pro analyzercore felhasználásával futás közben figyelhetünk jeleket az FPGA-n belül úgy, mintha egy oszcilloszkóp mérőszinórával be tudnánk nyúlni az FPGA integrált áramkörének a belsejébe.

Ezen funkciók közül a BSB-t és a Create or Import Peripheralt használtam a légpárnás szoftverének fejlesztése során.

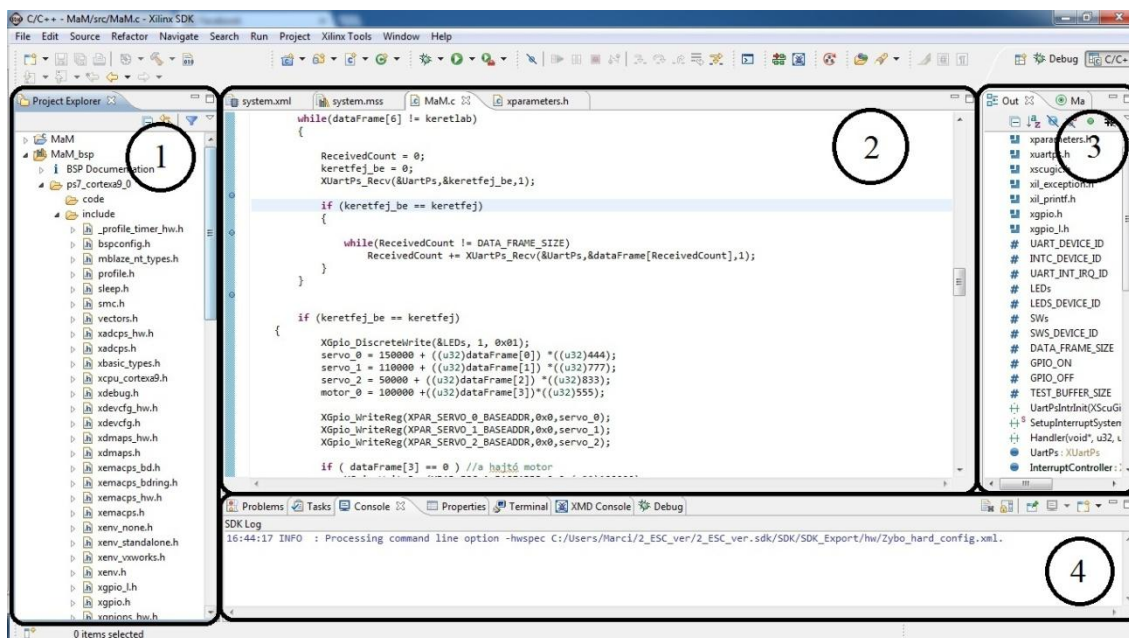
1.3.2. Software Development Kit (SDK)

Az SDK szoftver arra szolgál, hogy az XPS-ben konfigurált Zynq vagy MicroBlaze processzorra C vagy C++ nyelvű szoftvert tudjunk írni.

Főbb funkciók:

- GNU szoftver támogatása
- C/C++ fordító MicroBlaze és ARM Cortex-A9 processzorokhoz
- Debugger MicroBlaze és ARM Cortex-A9 processzorokhoz
- XilinxTools: arra szolgál, hogy a processzor mellett az FPGA-ra is tudjunk szoftvert letölteni.

A következő képen (15. ábra) az SDK ablak rendszere látható, a főbb alkotóelemeket számokkal jelöltem.



15. ábra –SDK ablak

Az ablak rendszer főbb elemei:

1. Project Explorer: tartalmazza a project mappáit és a hozzá tartozó fájlokat, valamint lehetőség van úgy fájlokat hozzáadni (pl.: h vagy .c).
2. Project Editor: Ez a leggyakrabban használt ablak, amely a project fájlok szerkesztésére szolgál.
3. Codeoutline display: Ebben az ablakban megtalálhatóak az aktuálisan szerkesztett fájl függvényei és header fájljai linkelve, ami azt jelenti, hogy kattintásra megnyílik a kiválasztott függvény vagy fájl.
4. Problems, Console, Properties: Ez az ablak több részből áll. A Console-on követhető a folyamatok lefutása (pl.: fordítás), a Problems ablakban külön ki vannak listázva a fordítás során észlelt figyelmeztetések és hibák, a Properties ablakok (pl.: Terminal, XMD Console, Tasks) hasznos kiegészítói a programnak, amelyek megkönnyítik a tervező munkáját.

Az SDK programot az ARM Cortex A9 processzor szoftverének megírására, valamint a bináris fájlok letöltésére használtam.

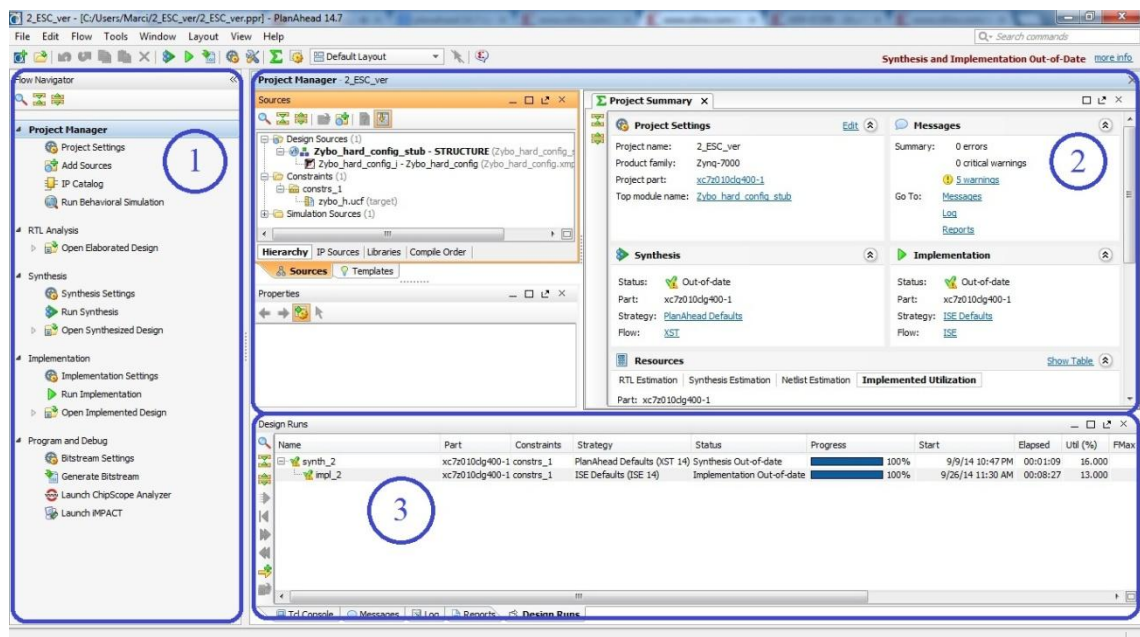
1.3.3. PlanAhead

A PlanAhead az utóbbi időbe a Xilinx beágyazott rendszer fejlesztőszoftver csomag központi egységévé vált. Meg tudja hívni az SDK-t és a XPS-t és kapcsolatot teremt a 2 program között, de ezen kívül számos funkcióval rendelkezik.

PlanAhead főbb funkciói:

- Synthesis – VHDL kódból huzalozási lista generálása
- Implementation – Hardver megtervezése a FPGA chipen
- GenerateBitstream – bináris fájlt generál, amit le lehet tölteni az FPGA SRAM programmemóriájába
- Huzalozási lista megtervezése
- Pmod és egyéb külső portra kötött lábak logikai szabványhoz történő konfigurálása
- Lehetőség van Chipen belül eldönteni és megtervezni a különböző alrendszerek elhelyezkedését, felépítését.

Következő képen (16. ábra) a PlanAhead fejlesztő szoftver kinézete látható, amelyen számokkal jelöltem a főbb részeket.



16. ábra - PlanAhead ablak

PlanAhead ablak részei:

1. Flow Navigator: Ebben az ablakban fordítási utasításokat adhat ki a felhasználó, valamint a projecthez hozzáadhat különböző komponenseket (VHDL file vagy beágyazott XPS project)
2. Project managerben láthatóak és szerkeszthetők a projecthez tartozó fájlok és állományok. A Summary ablakban fordítás utáni dokumentáció röviden olvasható.
3. Log ablakban végig követhető a project parancsok részletes lefutása, valamint rendelkezésre áll terminál ablak, amin keresztül scriptekkel irányítható a project lefutása.

A légpárnás jármű szoftverének fejlesztése során ez a program futtatta a projectet és segített meghívni az XPS-t és az SDK-t, valamint az FPGA programozásához állította elő a bináris állományt. Ezzel a programmal együtt lesz egész a beágyazott rendszer fejlesztő csomag, mert a XPS-ben konfigurált hardver lefordított bináris állományát és egyéb dokumentációit tovább adja az SDK-nak, ami projectet generál belőle majd letölti a bináris állományokat a fejlesztő kártyára. Véleményem szerint így rendezetten strukturáltan lehet felépíteni a beágyazott rendszerünket.

Minden egyes ablak kinagyítható és mozgatható, így a lehetőség nyílik több monitoros fejlesztésre, ami megkönnyíti az átláthatóságot. Ez a fejlesztő rendszer összes szoftverére igaz.

2. Légpárnás jármű

A következő képen (17. ábra) a légpárnás jármű látható, rajta a daruval.



17. ábra - Légpárnás jármű

2.1. Követelmények

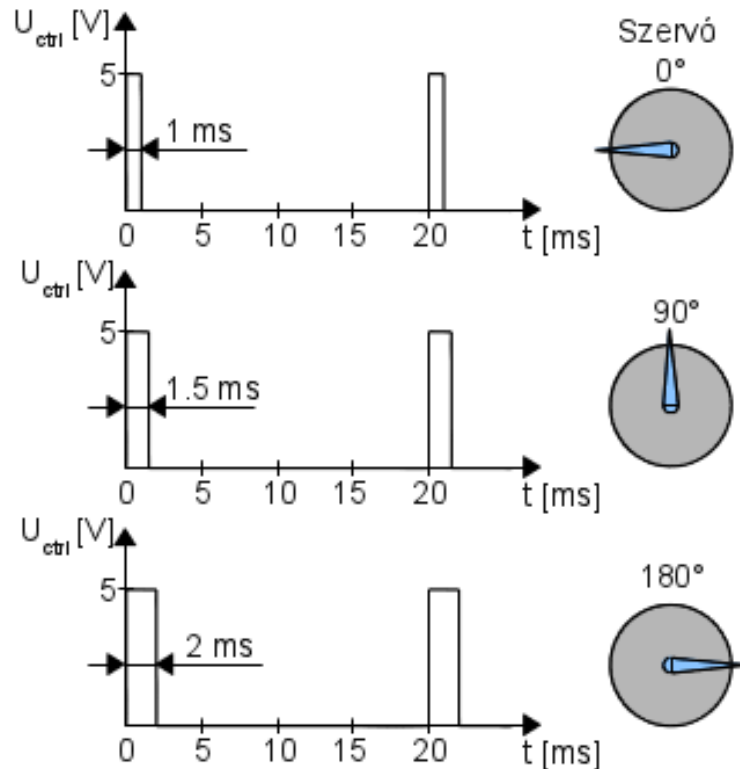
Légpárnás járművel szemben támasztott követelmények a 9. Magyarok a Marson verseny szabályzatában található a következőképpen:

- Légpárnásnak olyan jármű számít, ami egy vékony levegőrétegen siklik. Ez lehet a klasszikus szoknyás verzió vagy olyan is, ami új felfedezésű, de nem repülő eszköz (helikopter,quadrocopter).
- A robot bármikor érintkezhet a talajjal, megállhat, siklás közben súrlódhat, lényeg, hogy ne guruljon. A súrlódás nem használható kormányzásra (korcsolya szerű mechanika).
- A robot mérete maximálisan 50x40cm lehet és ebből a síkból egyedül a robotkar (1 darab / robot) nyúlhat ki legfeljebb **35cm**-re. A robot súlya ne legyen 3 kg-nál nagyobb.
- A robot minden olyan forgó alkatrészét védeni kell, ami sérülést okozhat egy másik robotban (fémháló). A nagy fordulatszámmal forgó légcsavar csak gyári műanyag lehet a balesetek elkerülése végett.
- Minden robotot kötelező azonnali vészmegállítási funkcióval ellátni.
- Bármilyen hardver szerelhető a robotra, ami az irányítást segíti pl. FPV kamera.
- Lehetőség van aktív védelemre, ami adott esetben azt jelenti, hogy a mozgathoz használt ventilátorok alkalmazhatóak az ellenfél mozgásának megnehezítéséhez. A robotkarral lehet védeni a letett kockát, de csak a célterület felett.
- A robotkar nem tartalmazhat extra felszerelést (penge, lángszóró, légfúvó) csak olyan eszközt, ami a kocka felvételéhez és lerakásához kell. Korlátozni más robotkar mozgását csak akadályozással lehet, nem lehet pl. lefogni. Kocka csak mechanikus módon távolítható el a célterületről. (nem lehet lefűjni)
- A robotot úgy kell felépíteni, hogy a legalacsonyabb (21cm) célterületen letett kockát se fújhassa le semmilyen körülmények között.
- Más csapat kockáját nem lehet tartósan felvenni, gyűjteni.

2.2. Felépítés

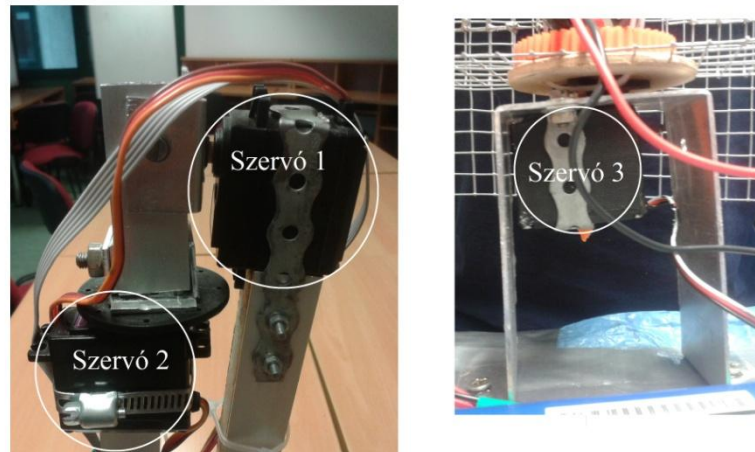
Ebben a részben röviden bemutatom a légpárnás jármű mozgást elősegítő motorok funkcióit elhelyezkedését és hogy hogyan illeszkednek a vezérlő ZYBO kártyához. A projektben az én feladatom az elektronika és a vezérlés megvalósítása volt, ezért nem ismertetem a légpárnás jármű vázának pontos felépítését.

A légpárnás járművön a hátsó hajtó motor és a daru horizontális és vertikális mozgásáért szervo motorok felelnek. A szervo motorok jellemzője, hogy általában csak 180 fokot tudnak fordulni, ezt a forgást egy külön a motorba épített szabályzó kontrolálja, de általában mechanikai végállás korlátok is be vannak építve. Három kivezetéssel rendelkezik: egy föld potenciál (fekete), egy általában 5 V-os feszültségű táp kivezetés (piros) és a szabályzóhoz csatlakozó vezérlő jelvezeték (fehér). Ez a fajta motor nem igényel illesztést a vezérlőhöz, a tápellátást adó lábakat a megfelelő potenciálra kell kötni, a vezérlő láb pedig közvetlenül a ZYBO kártyához csatlakozik. A következő ábra (18. ábra) a szervo motor mozgását szemlélteti különböző kitöltési tényezőjű vezérlő jelek rákapcsolása esetén.



18. ábra - Szervo motor vezérlő jelek

A képen (18. ábra) jól látható, hogy 20 ms-os periódus idővel rendelkező jelet kell előállítani, ahol a magas szintű állapot egy és két ms között változhat. A szervo motor ennek függvényében közel lineárisan mozog. A következő képen (19. ábra) a légpárnás járművön elhelyezett szervo motorok láthatóak.



19. ábra - Szervo motorok számozva

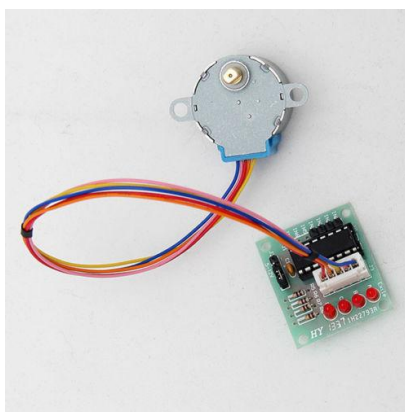
A légpárnáson elhelyezett szervo motorok és funkcióik:

- szervo motor 1: daru vertikális mozgatása
- szervo motor 2: daru horizontális mozgatása
- szervo motor 3: hajtó motor horizontális forgatása

A légpárnás jármű szoknyájának felfűzéséért valamint haladásáért egy-egy szénkefés motor felel. Ennek a motor típusnak mindössze csak kettő táp kivezetése van, amelyet a fordulatszám függvényében a rákapcsolt feszültség szintje vezérel. Ez a vezérlés úgy is megoldható, hogy konstans feszültség mellett nem folyamatos, hanem szaggatott jellel (PWM-vel) vezérelm, ekkor a rákapcsolt jel kitöltési tényezőjét változtatom. A kitöltési tényező és a jel magas szintű potenciáljának szorzata megadja, hogy mekkora egyenfeszültségű jelnek felel meg. A szaggatott jel frekvenciája 1,5kHz. A ZYBO-hoz történő illesztés 2 darab ESC-n keresztül történik, amelynek a vezérlése teljesen megegyezik a szervo motor vezérlésével. A ESC ennek a vezérlő jelnek a függvényében adja ki a kimenetén a 1,5 kHz –es PWM jelet a motornak.

A daru karján található unipoláris léptetőmotor, amely egy fogaskeréken keresztül hajt egy fogaslécet, ami a kar belsejében lévő dobókockákat nyomja ki. A léptetőmotor az egyik legkülönlegesebb szinkron motorfajta, mert két egymástól független

középmegcsapolással rendelkező tekercset tartalmaz. Digitális jellel vezérelhető úgy, hogy a tekercsek középmegcsapolását a földre kötjük, ezután minden egyes tekercs kivezetésre egymás után feszültséget kapcsolunk. A digitális jel tehát egy 4 bites vektor („0001”) amit körbe kell shiftelni. Minden átkapcsoláskor egy kicsit forog, „lép” a motor, ezért is nevezték el léptetőmotornak. Azért esett a választás léptetőmotorra, mert fontos volt a pontos pozícionálás, egyszerre csak egy kockát szabad kidobnia a kar belsejéből. A ZYBO kártyához való illesztés egy meghajtó áramkörön keresztül történik. A meghajtó áramkör és a léptetőmotor a következő képen látható (20. ábra).



20. ábra - Léptetőmotor vezérlővel

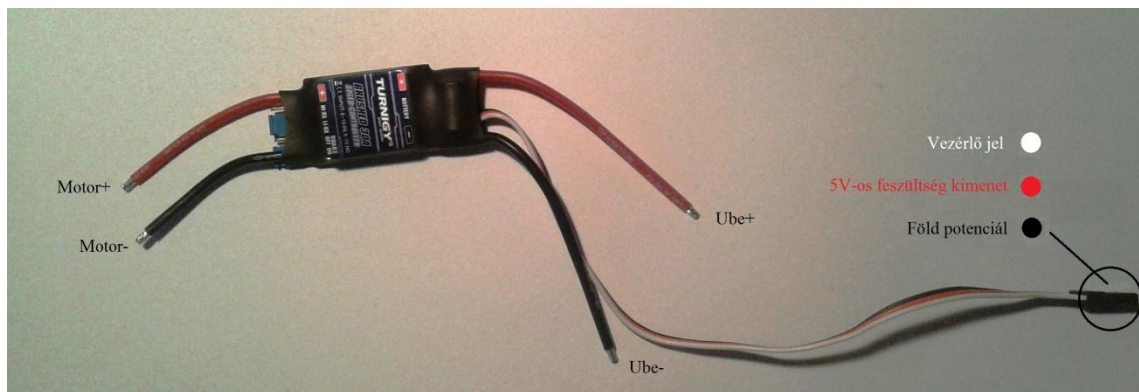
Ezek a légpárnás jármű főbb alkotó elemei, amelyeket vezérelni kell.

2.3. Légpárnás jármű elektronikus építő elemei

Az elektronika egyik legfontosabb része az **akkumulátor**, az eddigi évek Magyarok a Marson verseny tapasztalatai alapján a rossz akkumulátorválasztás az egyik legnagyobb hátrány. Fontos, hogy gyorsan lehessen tölteni és ne kelljen teljesen lemeríteni az újratöltéshez. Ezen kritériumok alapján a modellezésben gyakran használt Litium-Polimer akkumulátor a legalkalmasabb választás. Miután az akkumulátor típus kiválasztásra kerül, mérlegelni kell az áramfelvételt és ez alapján ki kell választani, hogy hány V-os és hány mAh-ás legyen az akkumulátor. A két szénkefés motor egyenként 6 – 6 A-t vesz fel, az egyéb motorok és elektronikai alkatrészekkel együtt összesen 14A az áramfelvétel, ha minden egyszerre működik. A két szénkefés motor 2 cellás – cellánként 3,7V-os – akkumulátorhoz kompatibilis, tehát az üzemeleti feszültsége 7,4V. Ezek alapján a kiválasztott akkumulátor 2 cellás, 5000mAh-ás és

legalább 3C-s, ami azt jelenti, hogy tartóan képes leadni az áram kapacitásának háromszorosát, ami 15A.

ESC betűszó jelentése Electronic Speed Controller. Véleményem szerint az egyik leghasznosabb alkotó része a légpárnás járműnek. Képes egy szénkefés motort vezérelni, a bemenetére érkező szervo motor jelével megegyező jel alapján és 5V-os tápfeszültséget is szolgáltat, amelyről megtáplálható a ZYBO kártya és a többi elektromos alkotóelem. Az ESC áram terhelhetősége 30A. A légpárnás járművön is használt ESC a következő képen (21. ábra) látható, a kivezetések villamos funkciójának jelölésével.



21. ábra - ESC

Bluetooth modul

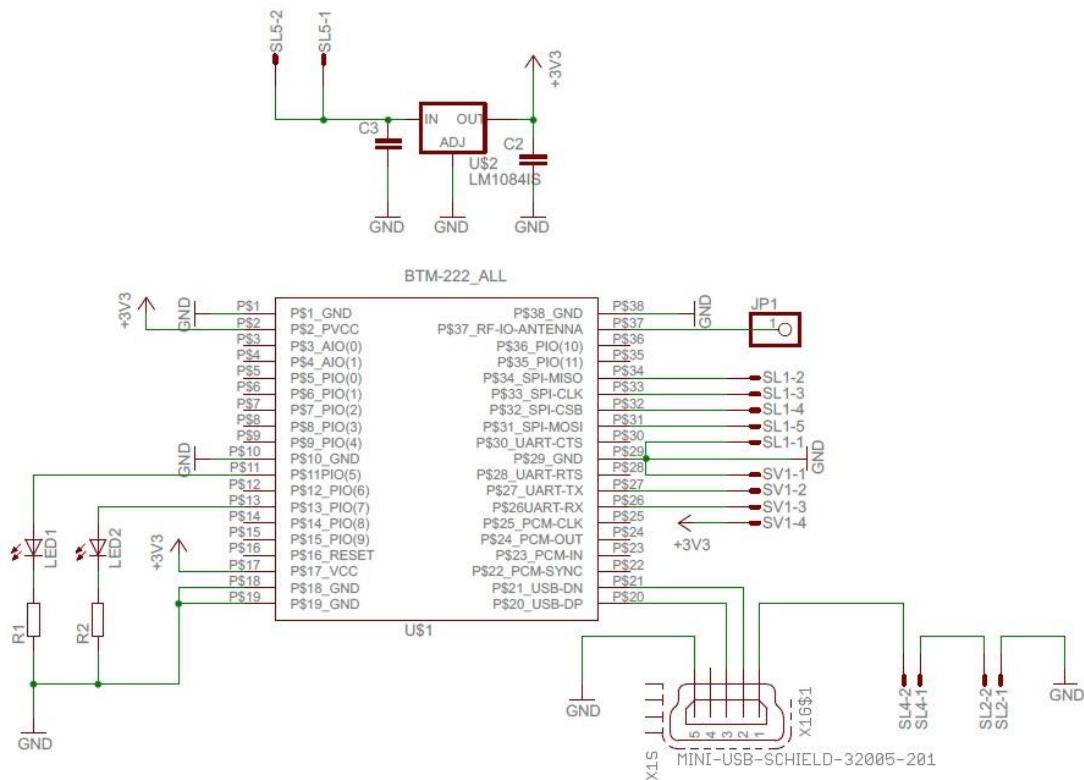
A Magyarok a Marson csapat évek óta használ megbízhatóan bluetooth-t kommunikációra. A választás ebben az esetben is a bluetooth-ra esett, pontosabban egy BTM222 bluetooth modulra. Integrált SPI és UART valamint USB kommunikációs interfészeket is tartalmaz. Ezek közül az UART aszinkron soros kommunikációs interfészt vettük igénybe. Ez a modul a soros UART portján keresztül konfigurálható.

UART soros port beállításai:

- baudrate 9600
- nyolc adatbit
- nincs paritás
- egy stoppbit

Beállításra került az is, hogy a két bluetooth modul automatikusan észlelje egymást, csatlakozzanak össze és várják az továbbításra szánt adatokat a vezérlő felől.

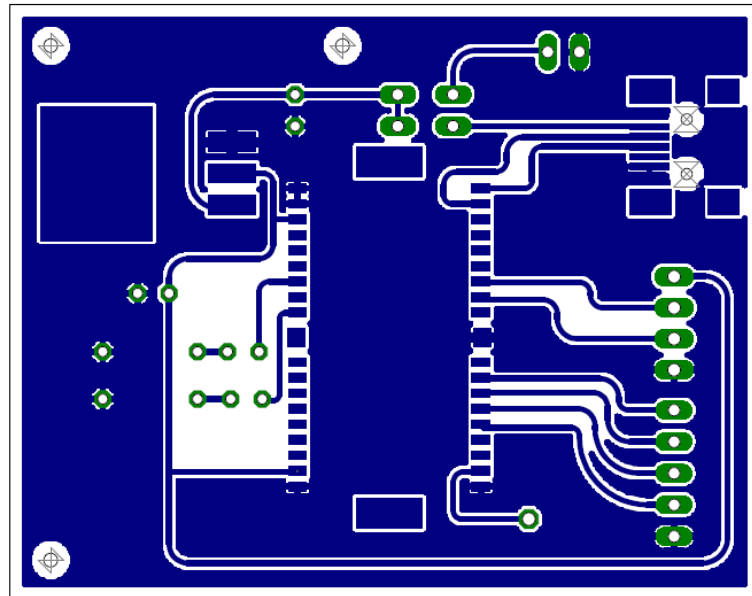
A következő Képen (22. ábra) a bluetooth modulhoz tervezett áramkör kapcsolási rajza látható. Tervezéskor az volt a cél, hogy egy univerzális panelt hozzak létre, amelyet nem csak e projekt keretein belül, hanem más alkalmazásra is fel lehet használni.



22. ábra - BTM 222 bluetooth modul kapcsolási rajz

A kapcsolási rajzon két integrált áramkör látható, az egyik a bluetooth modul, a másik LM1082IS lineáris feszültség stabilizátor, amely a bemenetére érkező 5V-os feszültségből 3.3V-ot állít elő a bluetooth modul számára. Az integrált áramkör előtt és után elhelyezkedő szűrő kondenzátorokat az integrált áramkör tervezési ajánlása alapján 10 μ F értékre méreteztem. A dielektrikum anyaga pedig tantal-pentoxid. A kommunikációs interfészek külső portokra lettek kivezelve. Látható még két LED is, amelyek arra szolgálnak, hogy jelezzenek a felhasználó számára, ha csatlakoztak a bluetooth-hoz (LED1 folyamatosan világít mikor csatlakozott, addig csak villog), valamint ha az érkező adatot (LED2 villog, ha érkezik adat) észlel. A LED-ekkel sorba kötött ellenállásokat 220 Ω -ra méreteztem.

Az itt látható kép (23. ábra) a bluetooth modul nyomtatott áramköri tervét ábrázolja:

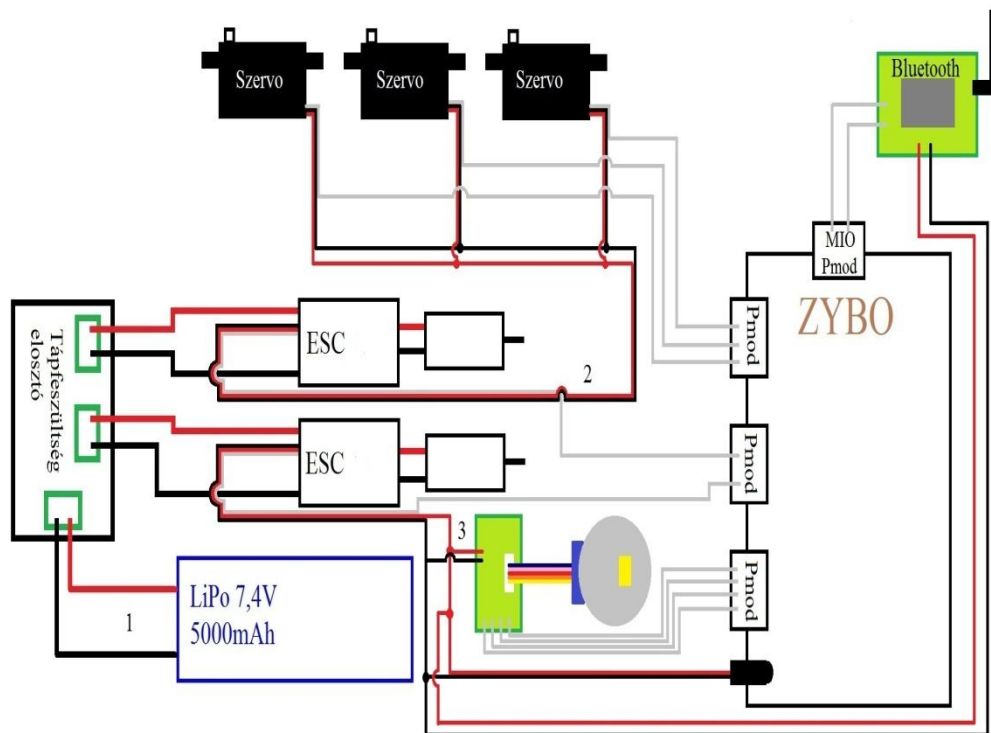


23. ábra - BTM 222 nyomtatott áramkör terv

A tervezési cél nem volt rossz, de sajnós azt nem mérlegeltem, hogy a kesztyűn nincs hely felesleges, nem használt kivezetésekre, így ebben az esetben nem előny az univerzitás. A kesztyűre jobb lett volna egy kisebb, minimális port kivezetéssel rendelkező nyomtatott áramközi panel.

A léptetőmotor meghajtó áramköre a motorokkal együtt alkot egy egységet. Az áramkör tartalmaz egy ULN2003 meghajtó integrált áramkört, ami a 4 vezérlő bemeneten kívül igényel még egy 5V-os tápfeszültséget is. A meghajtó integrált áramkörre azért van szükség, mert az ZYBO kártya kimenetére érkező TTL jelszint nem elég a motor meghajtásához, valamint az áramterhelhetősége sem megfelelő.

Az eddigiekben bemutattam a légpárnás jármű elektronikájának építő köveit. A következő képen (24. ábra) a teljes villamos hálózat látható:



24. ábra - Légpárnás jármű elektronikai blokkvázlata

A képen három tápfeszültség vonalat jelöltem:

1. Az akkumulátorból származó 7,4V feszültség a 2 ESC-t táplálja.
2. ESC 5V-os kimenete a három szervo motor hajtja.
3. ESC 5V-os kimenete a ZYBO kártyának a bluetooth panelnek és a léptetőmotor vezérlőnek biztosít stabil tápfeszültséget.

A szürke vezetékek jelvezetékeket jelölnek. Ezek egytől egyik a ZYBO kártya Pmod csatlakozóihoz futnak.

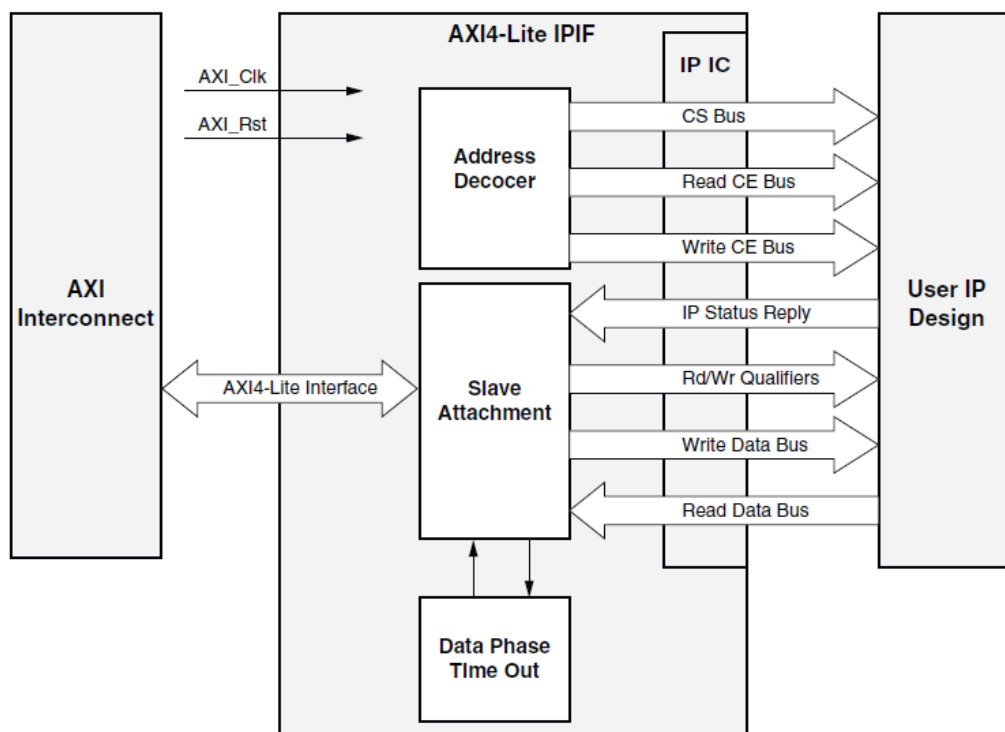
2.4. Perifériák implementálása XPS-ben

Az eddigiekben említettem a PS és PL rendszerek egy ipari AXI szabványos sínrendszerrel vannak összekötve, amely a PS oldalon csatlakozik az ARM processzor AMBA sínrendszeréhez. A PL oldalon viszont a felhasználónak kell létrehozni az AXI

sínrendszerhez csatlakozó logikai blokkokat, amelyekhez, mivel közvetve csatlakozva van az ARM buszrendszeréhez, egy memória címterület fog tartozni. Ezt a címterületet lehet írni és olvasni is. Így valósul meg a kapcsolat a PS és PL rendszer között.

A perifériák implementálása előtt létre kell hozni a PlanAhead projectet a felhasznált Xilinx – xc7z010clg400-1 - integrált áramkörhöz, és meg kell hívni belőle az XPS szoftvert. A XPS szoftverbe a teljes hardver konfigurálható, mind a Zynq PS rendszere, mind pedig a PL. A perifériákat a Create of Import Pheripheral funkcióval kell létrehozni, de hozzáadható a már előre a Xilinx cég által fejlesztett általános perifériák is. Fel is használtam az AXI GPIO perifériát, de ezen kívül számos hasznos elemet találhatunk például: AXI Timer/Counter, AXI CAN Controller, AXI Interrupt Controller, AXI XADC.

A periféria létrehozása során számos beállítási lehetőség van, többek között az is beállítható, hogy milyen típusú AXI protokollhoz csatlakozzon a periféria és hány regisztere legyen. Hozzáadhatunk külön memóriát és szimulációs állományt. A beállítások befejezése után a következő képen (25. ábra) látható entitású (IPIF, User_IP_Design) hardver tömbök jönnek létre:



25. ábra - Periféria entitás

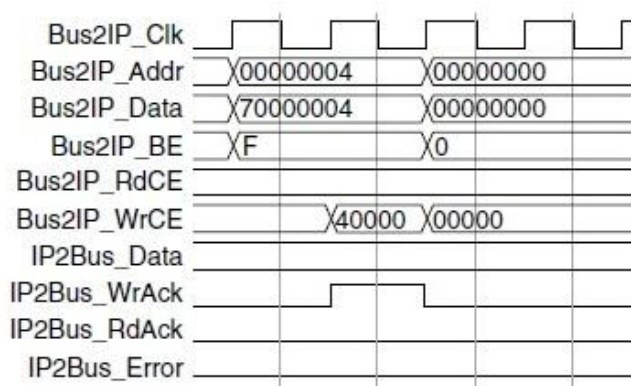
Ehhez tartozik egy-egy, a működést leíró VHDL állomány. Az IPIF tömb az AXI Interconnect és a saját periféria közötti kapcsolat biztosítja, ebbe az állományba nem lehet saját VHDL kódot írni. A User IP Design blokk a periféria működését leíró, regiszterkészletet tartalmazó állomány. Ebben az állományban már készen találjuk az ábrán jelölt összeköttetéseket és regisztereket leíró VHDL kódot, ehhez kell hozzá írni a periféria működését leíró VHDL kód részletet.

A generált periféria (User IP Design), generált portjainak funkciói:

- Bus2IP_Clk szinkron órajel a periféria számára. Alap állapotban 100MHz.
- Bus2IP_Resetn logikai nullára aktív reset.
- Bus2IP_Data adat vezeték regiszteríráshoz.
- Bus2IP_BE az adat vezetékeken érkező adatokról Bytonként tartalmaz információt. Byte sávokra osztja fel az adatot, és ha hasznos adat jön az adott sávban, akkor az értéke logikai egy.
- Bus2IP_RdCE regiszterírást engedélyező és egyben címző vezeték rendszer.
- Bus2IP_WrCE regiszter kiolvasást engedélyező és egyben címző vezeték rendszer.
- IP2Bus_Data adat vezeték regiszterolvasáshoz.
- IP2Bus_RdAck jelzi, ha kiolvasott adat van az IP2Bus_Data vezetéken.
- IP2Bus_WrAck jelz, ha adat érkezik a Bus2IP_Data vezetéken.
- IP2Bus_Errorba periféria rendellenes működése esetén logikai egy értékű.

A generált portokon kívül saját port is definiálható a perifériának, amelyeket akár külső Pmodra vagy a chipen belüli más perifériákhoz is csatlakoztatható.

A következő képen (26. ábra) a regiszter írás idődiagramja látható.



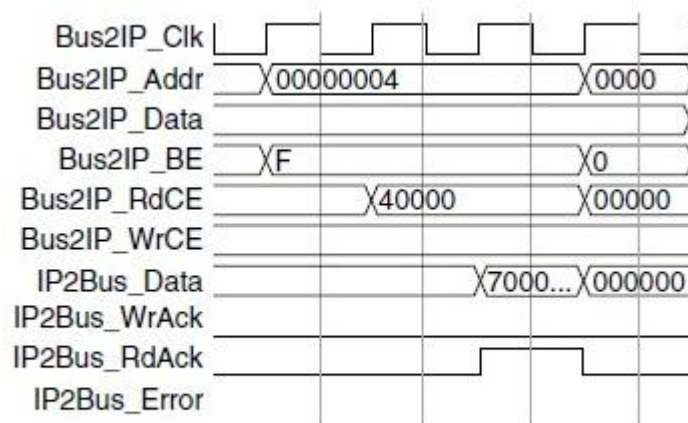
26. ábra – Regiszter írás idődiagram

A memória kezelés időigényét a memória órajel ciklusszámával szokták jellemezni. Látható, hogy a teljes folyamat lezajlása három órajel ciklust vesz igénybe. A regiszterek címzése és engedélyezése a Bus2IP_WrCE buszon keresztül történik, a következő táblázatba (2. táblázat) lefektetett szabályok szerint.

Memória Címzés		
Bus2IP_WrCE/Bus2IP_RdCE	Memória címek	eltolás
"1000"	Periféria kezdőcíme	0x0
"0100"	Periféria kezdőcíme	0x4
"0010"	Periféria kezdőcíme	0x8
"0001"	Periféria kezdőcíme	0xC

2. táblázat - Periféria regisztereinek címzése

A következő ábrán (27. ábra) a memória olvasás idődiagramja látható.

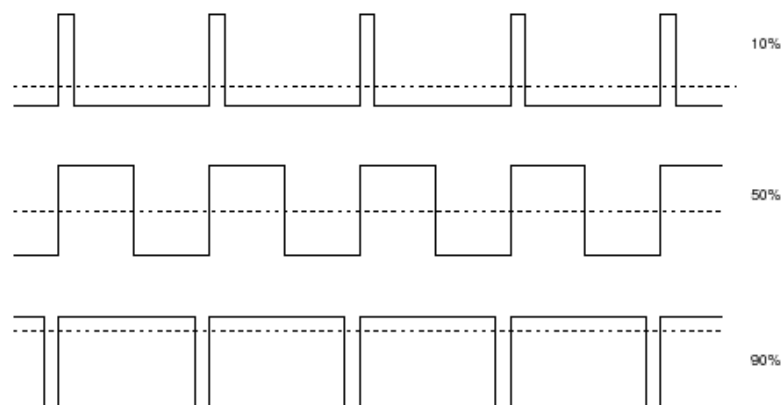


27. ábra - Regiszter olvasás idődiagram

A memóriaolvasás egy órajel lefutással több idő, mint az írás, tehát négy órajel. A regiszterek engedélyezése a regiszterírásnál és olvasásnál teljesen azonos folyamat, így az írásra is teljesülnek az előző táblázatban lefektetett szabályok.

2.4.1. Univerzális, paraméterezhető PWM vezérlő

Mielőtt ismertetném, a periféria működését definiálom, hogy mi is az a PWM. A PWM szó jelentése Pulse-Width Modulation, a saját nyelvünkre fordítva pedig impulzusszélesség moduláció. Digitális modulációs eljárás, amelyet a teljesítmény elektronikai kapcsolóelemek (például: tranzisztor) fejlődésével elkezdtek motorok és egyéb feszültséggel vezérelhető eszközök szabályozására használni. A következő képen (28. ábra) egy 10%-os egy 50%-os és egy 90%-os kitöltési tényezőjű jel látható.



28. ábra - PWM jelalakok

A szaggatott vonalak a PWM jelek egyenáramú megfelelőjét jelölik. Ez úgy számítható ki, hogy a kitöltési tényezőt megszorozzuk a magas szintű feszültség potenciállal.

A szervo motorok és szénkefés motorok vezérlésének megvalósításához volt szükség PWM vezérlőre. A szénkefés motorok indítási árama nagy, ezzel csökkenti az akkumulátor élettartamát és egyéb áramköri zavarokat is okozhat. Ha PWM jellel vezéreljük úgy, hogy a kitöltési tényezőt folyamatosan növeljük, míg el nem éri a 100%-ot, akkor tompíthatjuk az induló áram csúcsokat. A szénkefés motorok fordulatszáma arányos a PWM kitöltési tényezőjével, így nem csak az indítási áramok csökkenthetők, hanem a fordulatszám is jól szabályozható.

A szervo motorok számára meghatározott alakú és frekvenciájú jelet kell szolgáltatni, amit már ismertettem az előzőekben.

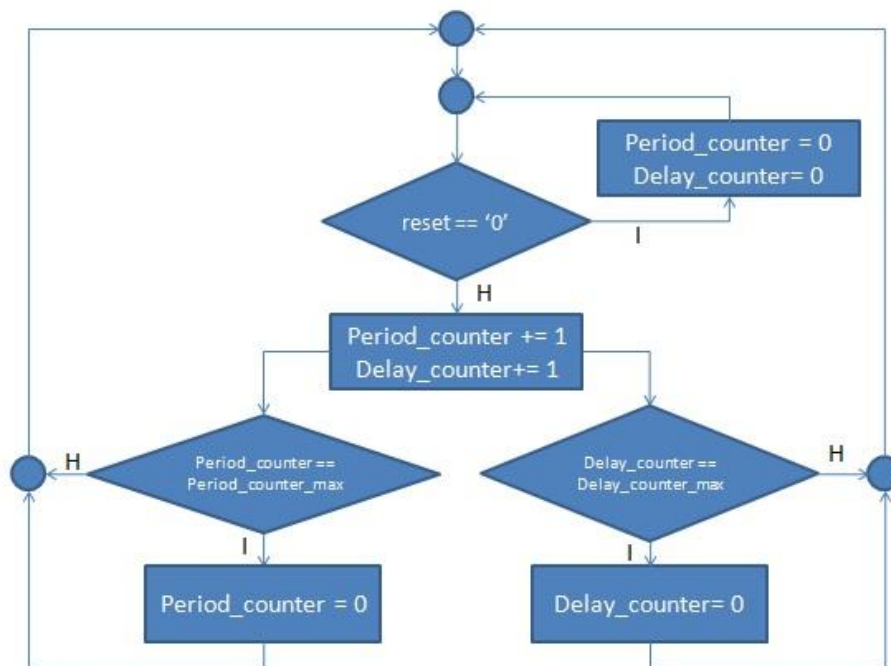
E két vezérlési feladat megvalósításához jól konfigurálható vezérlő egységre volt szükség. A PWM vezérlő periféria a regisztereken keresztül konfigurálható. A periféria 6 regiszterrel rendelkezik, a következő táblázat (3. táblázat) szemlélteti ezek funkcióit és címzési szabályait.

Regiszter funkció	Memóriacímzés	
	Kezdőcím	eltolás
periféria órajel	periféria kezdőcím	0x0
pwm órajel	periféria kezdőcím	0x4
pwm kitöltési tényező	periféria kezdőcím	0x8
pwm késleltetés	periféria kezdőcím	0xC
minimális kitöltési tényező	periféria kezdőcím	0x10
maximális kitöltési tényező	periféria kezdőcím	0x14

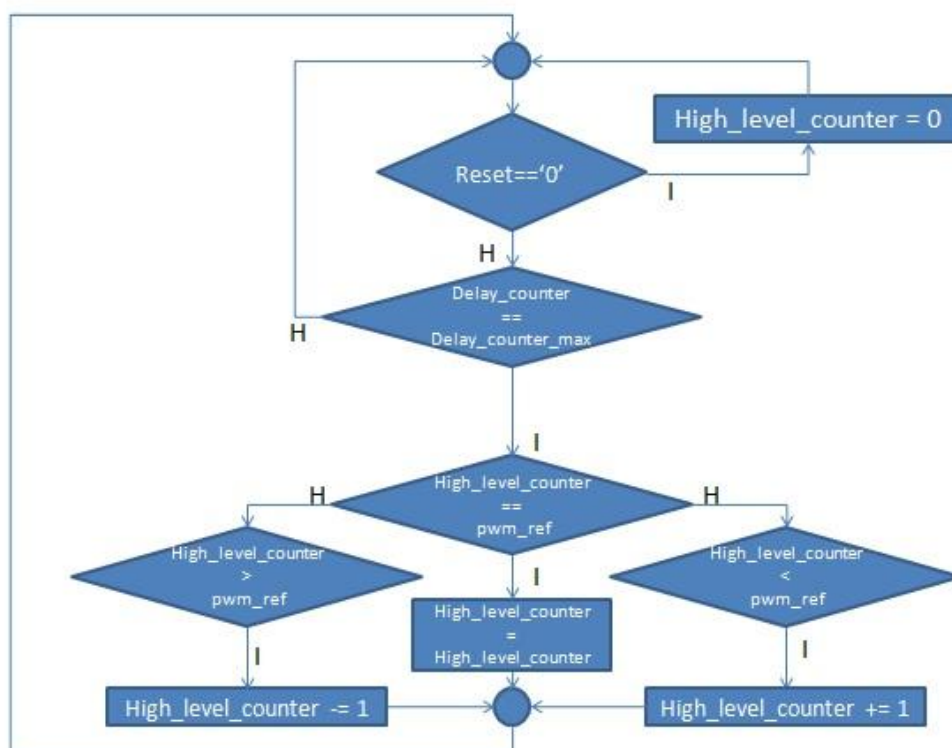
3. táblázat - PWM periféria regiszterei

A periféria lehetővé teszi, hogy széles frekvencia tartományban elő lehessen állítani jeleket. Minél nagyobb frekvenciájú a generált jel, annál jobban romlik a kitöltési tényező beállíthatósága. Ez azért van, mert maximum annyi részre bontható a jel, mint az órajel frekvencia/jelfrekvencia hányados.

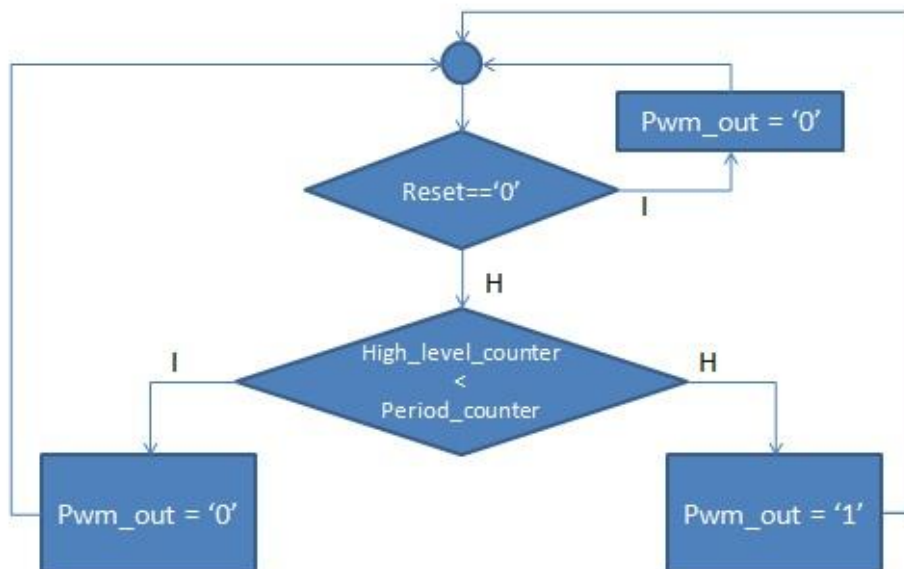
A következő ábrák (29. ábra , 30. ábra , 31. ábra) a VHDL kód processeinek a folyamatábrái. A VHDL process egy olyan folyamat, amely folyamatosan fut a felprogramozástól kezdve, mint egy processzor. A különböző processek párhuzamosan és szinkronba működnek. Ha be lehetne nézni az FPGA chip belsejébe, valószínűleg az lehetne látni, hogy ezek a processek jól elkülönült blokkok formájában helyezkednének el. Ez a párhuzamos tömbösített működés az egyik nagy előnye az FPGA-nak a processzorokkal szemben.



29. ábra - PWM modul process 1



30. ábra - PWM modul process 2



31. ábra - PWM modul process 3

A modulációt végrehajtó kód rész megírásakor azon gondolkodtam, hogy milyen digitális áramkörrel vagy áramkörökkel lehetne ezt a feladatokat egyszerűen megvalósítani. Végül arra jutottam, hogy két különbözően paraméterezett számláló áramkörre lesz szükség.

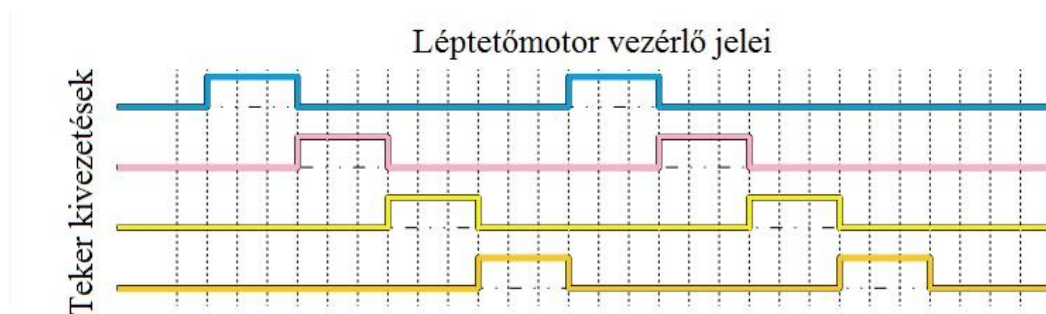
Az egyik számláló, amelyet az (29. ábra - PWM modul process 1) ábra ír le, a generálandó jel periódus idejének megfelelő idő alatt csordul túl, ezt az elkövetkezendőkben periódus számlálónak fogom hívni.

A másik számláló ugyanabban a tartományban számol, viszont ez mindig csak a periódus számláló maximum szorozva a kitöltési tényező értékig. A szorzat változásának esetén értelemszerűen a számláló követni foglya, amíg el nem érte a megfelelő értéket. Ezt a (30. ábra) ábrán látható folyamat írja le.

A harmadik folyamat (31. ábra) e két számláló értékének összehasonlításából digitális jelszintet állít elő a kimeneti portra.

2.4.2. Léptetőmotor vezérlő

Az előző fejezetekben kifejtettem, hogy miért lett a légpárnás járműbe beleépítve a léptető motor. A következő ábra (32. ábra) szemlélteti a léptetőmotor működéséhez szükséges jeleket, amelyeket tekercseinek a kivezetéseire kell adni.



32. ábra – Léptetőmotor vezérlő jelei

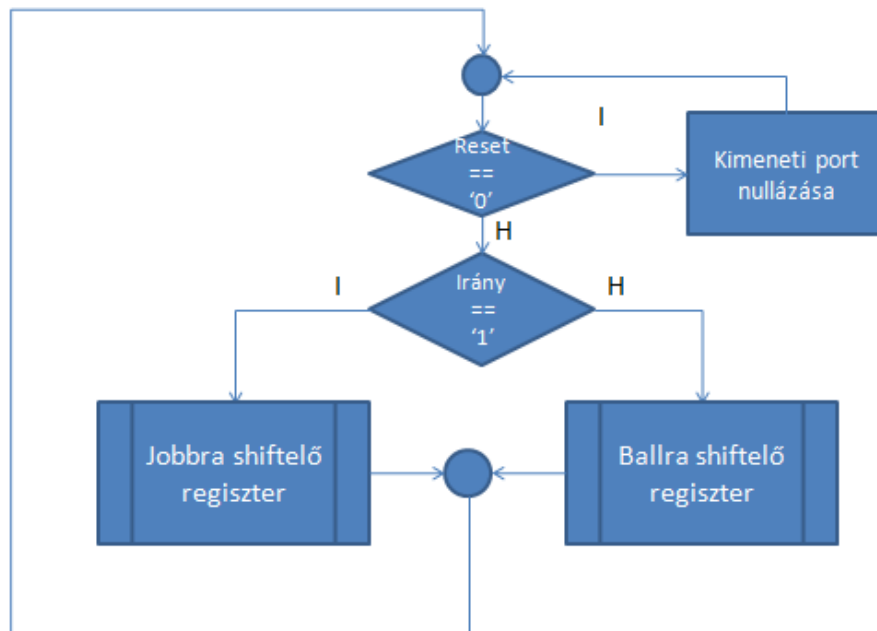
Véleményem szerint ezeket a jeleket egy sima processzoros mikrovezérlő segítségével nehézkes lenne előállítani, ezért a léptetőmotorokhoz gyártanak különböző vezérlő áramköröket, amelyek segítenek előállítani a vezérlő jeleket. Viszont ha programozható logika rendelkezésre áll, egészen más a helyzet.

Jobban megnézve a jeleket, rá lehet jönni, hogy ez egy nagyon egyszerű jel. Vajon milyen digitális áramkör valósít meg hasonló jelet?

A válasz a körbe shiftelő négybites shift regiszter, ami egy nagyon egyszerű digitális áramkör. Mindössze ennyit kell tenni, hogy minden regiszter bitet küldő portra kell kivezetni. Ezt a digitális áramkört egyszerű megvalósítani programozható logikai vezérlőre.

Így már nem kell külön áramkört vásárolni, amely előállítja a kellő jeleket, hanem csak a szintillesztésről kell gondoskodni a ZYBO portja és a léptetőmotor tekercs kivezetései között.

A következő ábrán (33. ábra) a periféria működését leíró folyamat (process) leírása látható:



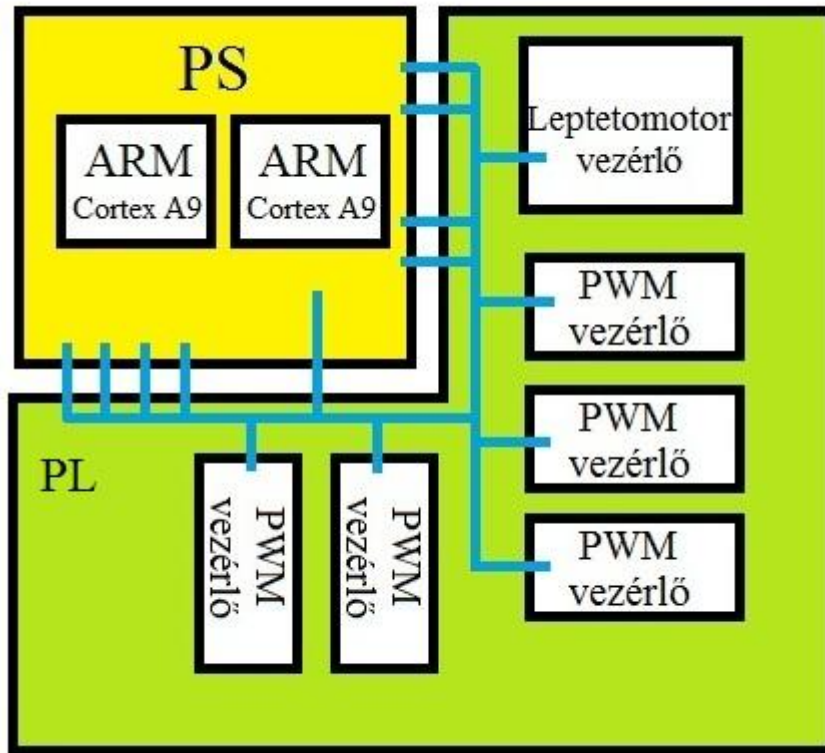
33. ábra - Léptetőmotor process

Ahogy a PWM vezérlőnél, ennél is alapkövetelmény a jó paramétereizhetőség. Négy regiszterrel rendelkezik a periféria, amelyek a következő táblázatban vannak felsorolva. A táblázat (4. táblázat) tartalmazza a memória címzés szabályait is.

Regiszter funkció	Memóriacímzés	
	Kezdőcím	eltolás
periféria órajel	periféria kezdőcím	0x0
shifregiszter órajele	periféria kezdőcím	0x4
lépésszám	periféria kezdőcím	0x8
shiftelés iránya	periféria kezdőcím	0xC

4. táblázat - Léptetőmotor vezérlő periféria

Végeredményben a konfigurált beágyazott rendszer terv felépítését a következő ábra (34. ábra) szemlélteti:



34. ábra - Perifériák tömbvázlata

A kék vonal az AXI sínrendszert jelöli, zöld háttéren a PL sárgán pedig a PS alrendszerek láthatóak. A PL-en belüli bekeretezett részek a perifériákat jelölik. Ahhoz, hogy az összes motort vezérelni lehessen ennyi perifériára volt szükség. Ahogy a képen is látható azonos típusú perifériákat többször is lehet használni egy beágyazott rendszer tervén belül.

A létrehozott perifériák név és vezérelt elem szerinti elrendezéséről a következő táblázat (5. táblázat) ad tájékoztatást:

Periféria neve	Vezérelt elem	Periféria típusa	Pmod port	Pmod láb	Periféria kezdő memória cím
ESC_0	Szoknyát felfújó motor	PWM	JB	7	0x7E800000
ESC_1	Meghajtó motor	PWM	JB	8	0x7E820000
servo_0	Daru horizontális mozgását végző szervo motor	PWM	JB	1	0x6CE00000
servo_1	Daru vertikális mozgását végző szervo motor	PWM	JB	2	0x6CE20000
servo_2	Meghajtó motort forgató szervo motor	PWM	JB	3	0x6CE40000
lepteto_0	Kocka kitoló léptető motor	Léptetőmotor vezérlő	JC	1,2,3,4	0x76600000

5. táblázat - perifériák lábkiosztása és memóriacímzése

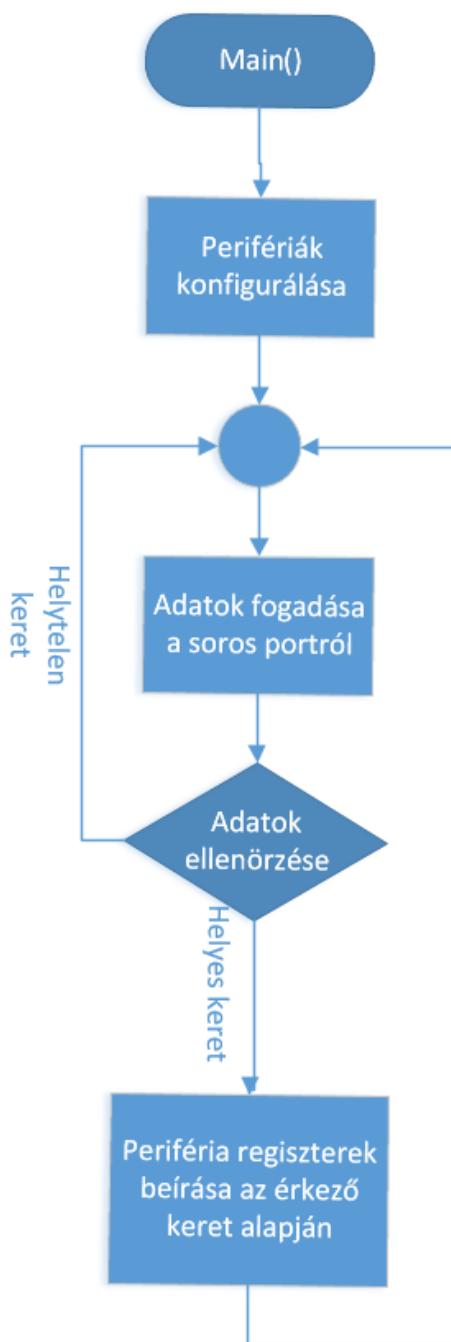
A perifériák a megnevezéseiket a vezérelt elemek után kapták. A táblázat tartalmazza azt is hogy, melyik periféria melyik Pmod csatlakozásra van kivezetve port és láb szerint.

A perifériák implementálása után nincs más dolgunk, mint bezárni az XPS programot és a PlanAhead-ba lefordítani az FPGA szoftverét, amelyet a perifériák kódja alkot. Ennek a végeredménye egy bináris fájl.

3.5. SDK szoftver

Az SDK fejlesztő környezetben kell a PS rendszer szoftverét fejleszteni, lefordítani, majd letölteni, miután XPS-ben elvégeztük a teljes hardver konfigurációt. Mikor az SDK-t meghívja a PlanAhead, exportálja a teljes hardver dokumentációt és az FPGA bináris lefordított szoftverét. Miután megnyílt az SDK, létre kell hozni egy projektet, ami tartalmazza a szükséges fájlokat a szoftverfejlesztéshez. Ezeket a fájlokat az SDK automatikusan legenerálja az exportált hardver dokumentáció alapján. Az ARM processzorhoz pedig a Xilinx cég által kiadott állományokat használhatjuk, amiket szintén automatikusan importál a projekthez az SDK.

A következő képen látható a szoftver folyamat (35. ábra) ábrája.



35. ábra - SDK szoftver folyamatára

Látható, hogy elég egyszerű a szoftver működése. A szoftver először konfigurálja a perifériákat, amivel majd külön fejezet foglalkozik. Majd végtelen ciklusba lépve

elkezd adatokat fogadni a soros UART interfésztől, amelyeket ellenőriz, és ha megfelelő, akkor felhasználja ezeket. Egy 9*8bit-es bináris egész számokat tartalmazó adatkeretet vár, amely meghatározza az összes motor állapotát. Az adatkeretet azért nevezik keretnek, mert az első bit minden esetben az adat bitek kezdetét jelzi, az utolsó pedig az végét, így egy keretbe foglalja az adatokat. Ez a fajta adat továbbítás azért ideális, mert a fogadó oldalon jól be lehet azonosítani az adatokat, ezzel kiszűrve a hamis adatok okozta üzemelési zavarokat.

A program a fogadott keret adatai alapján írja a perifériák regisztereit. A következő táblázatban (6. táblázat) összefoglalom, hogy az érkező keret adatait melyik periféria regiszterébe írja be a szoftver.

Adat sorszáma a keretben	Periféria neve	Regiszter megnevezése
Keretfej		
2	servo_0	Kitöltési tényező
3	servo_1	Kitöltési tényező
4	servo_2	Kitöltési tényező
5	ESC_1	Kitöltési tényező
6	ESC_2	Kitöltési tényező
7	lepteto_0	lépésszám
Keretláb		

6. táblázat - Fogadott adatkeret

3.5.1. Perifériák konfigurálása

Ebben a részben bemutatom, hogy milyen értékeket és miért kell beállítani a perifériáknak a megfelelő működéshez. A következő táblázat (7. táblázat) a PWM vezérlő konfigurált adatait tartalmazza regiszterenként.

Periféria megnevezése	Regiszterek					
	periféria frekvencia	PWM frekvencia	Kitöltési tényező	Minimális kitöltési tényező	Maximális kitöltési tényező	PWM számláló Késleltetés (órajelciklus)
ESC_0	100MHz	50Hz	5%	5%	10%	500
ESC_1	100MHz	50Hz	5%	5%	10%	500
servo_0	100MHz	50Hz	11,5%	7,5%	11,5%	0
servo_1	100MHz	50Hz	9%	5,5%	12,5%	0
servo_2	100MHz	50Hz	6,75%	2,5%	10%	0

7. táblázat – PWM perifériák regisztereinek beállításai

Látható, hogy a frekvenciákat beállító regiszterek értékei megegyeznek az összes perifériánál. Fontos megjegyezni, hogy a periféria frekvencia nem beállítás, csupán segéd adat. A periférián belüli változók számításához van rá szükség. A többi regiszter viszont befolyásolja a periféria működését. A kitöltési tényezők ebben a konfigurációban a motorok alapállását jelölik. A szervo motorok alapállása nem minden esetben egyeznek meg a középállással, azaz 90°-al. Ez azért van, mert például a darut mozgató szervók alap állapotba úgy állítják be a darut, hogy a karja függőlegesen álljon a tartó rúd mellett. A másik szembe ötlő dolog, hogy a szervo motorokat vezérlő jelek kitöltési tényező intervalluma nem felel meg a szabványos jelformának, amelynek 5% és 10%-os kitöltési tényező között kellene változnia. Ez két dolog miatt van:

- A szervo motorok nem teljesen ideálisak, így kilépve a vezérlő jel intervallumából tovább forognak.
- A szervo motorok mozgási intervallumát a motorok által mozgatott eszköz mozgástere határolja be (pl.: a daru kar vertikális mozgása nem kell, hogy 180°

legyen, mert így a kart függőlegesen felfelé nézne, és saját magára dobna ki a kockát).

A következő perifériát már jóval egyszerűbb konfigurálni, ahogy a táblázat (8. táblázat) is szemlélteti.

Periféria megnevezése	Regiszterek			
	periféria frekvencia	shift frekvencia	irány	lépésszám
lepteto_0	100MHz	250Hz	JOBB	0

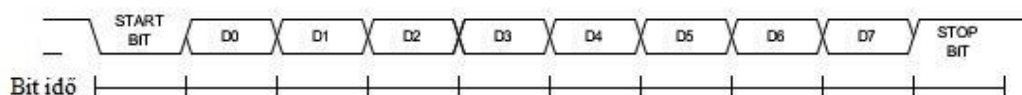
8. táblázat - Léptetőmotor vezérlő periféria regisztereinek beállításai

A periféria órajele ebben az esetben is csak belső változók kiszámításához szükséges. A többi regiszter határozza meg a periféria működését. A shift frekvencia a kimeneti jel frekvenciáját állítja be, az irány pedig azt, hogy merre shifteljen a belső shift regiszter. Végül a lépésszám pedig a shiftelések számát szabja meg. Egy kocka kidobásához 1000 lépést kell tennie jobbra a léptetőmotornak.

A perifériák konfigurálása sok kísérletezés árán alakult ki a bemutatott formában.

3.5.2.Megszakításos UART

A fejezet címében két fontos definiálandó szó is található. Az egyik az UART amely, Universal asynchronous receiver/transmitter rövidítése. Ez magyarul azt jelenti, hogy univerzális aszinkron adó/vevő. Az UART fizikailag két port kivezetést tartalmaz, amelyen ahogy a neve is elárulja, ad és vesz adatokat. Az univerzális tulajdonsága abban rejlik, hogy a felhasználó számára gazdag beállítási lehetőségeket biztosít, a várandó vagy továbbítandó jel alakjáról és gyorsaságáról. Az egyik legfontosabb beállítási lehetőség a baud rate, amely reciprokra egy bit időnek. A bit idő egy bit elküldésének az ideje. A következő ábra (36. ábra) szemlélteti egy 8 bites jel küldését:



36. ábra - UART soros jel

Látható, hogy az adatbitek egy START után kezdődnek és egy STOP bittel végződnek. A START és STOP bitek lehetnek különböző hosszúságúak is, amelyet a bitidő többszörösével adnak meg, például 1, 1.5, 2.

Az ZYBO, a STM32F303 fejlesztőkártya és a két bluetooth modul UART beállítása megegyezik:

- baudrate 9600
- nyolc adatbit
- nincs paritás
- 1 stopbit
- megszakítás engedélyezve

Ezeket a beállításokat a ZYBO kártyán és az STM32F303 fejlesztő kártyán szoftveres úton kell beállítani.

Az másik ismeretlen fogalom a megszakításos. Vajon mit jelent az, hogy megszakításos?

Minden processzorban található megszakítás vezérlő, amelynek többek között az a feladata, hogy a megszakítás vezérlőbe érkező megszakításkéréseket továbbítsa a processzor felé, ami ennek hatására felfüggeszti a program futását és kilép egy szubrutinba (függvénybe) ami egyszer lefut, majd a program onnan folytatódik ahol éppen megszakadt.

Adatok érkezése esetén az UART interfész jelez a megszakítás vezérlőnek, amely felfüggeszti a program futását és átlép egy szubrutinba. A szubrutin, lefutás közben az érkező adatot vizsgálja, és hogyha az adat keretnek megfelelő adatokat észlel, elmenti.

3. Távvezérlő ”Jedi kesztyű”

3.1. Követelmények

Az irányítás valós időben egy kézfejre (csuklótól az ujjhegyig) szerelt szenzorok segítségével történhet vagy a kéz mozgását feldolgozó rendszer segítségével. Erre a célra kifejlesztett gyári eszköz (kinect, leapmotion, vr kesztyű) nem használható, de a működési elv után építhető egy eszköz.

- Nincs lehetőség a kéz ‘gyári’ adottságait megnövelni (kitömött lapátkesztyű, dróttal meghosszabbított ujjak..)
- a kesztyű nem tartalmazhat potenciométert, de az ellenállás változásán alapuló szenzor minden további nélkül beépíthető. A mechanikus kapcsolók minden formája engedélyezett.
- A kesztyűnek nem kell tartalmaznia a teljes hardvert, csak az érzékelőket.

3.2. Felépítés

A vezérlő kesztyű felépítése jóval egyszerűbb, mint a légpárnás járműé. Az alapja egy sima fekete kesztyű. Erre lettek felragasztva a működéséhez szükséges elemek. A következő képen (37. ábra) a kesztyű felső – kézfej felőli - oldala látható, amelyen az STM32F3 és az akkumulátor helyezkedik el:



37. ábra - Vezérlő kesztyű I.

A következő képen (38. ábra) látható a vezérlő kesztyű hátoldala. Az újak végén találhatóak a nyomógombok, amelyeket pirossal be is karikáztam. A gombok számozása a mutató újtól indul egytől négyig.



38. ábra Vezérlő kesztyű II.

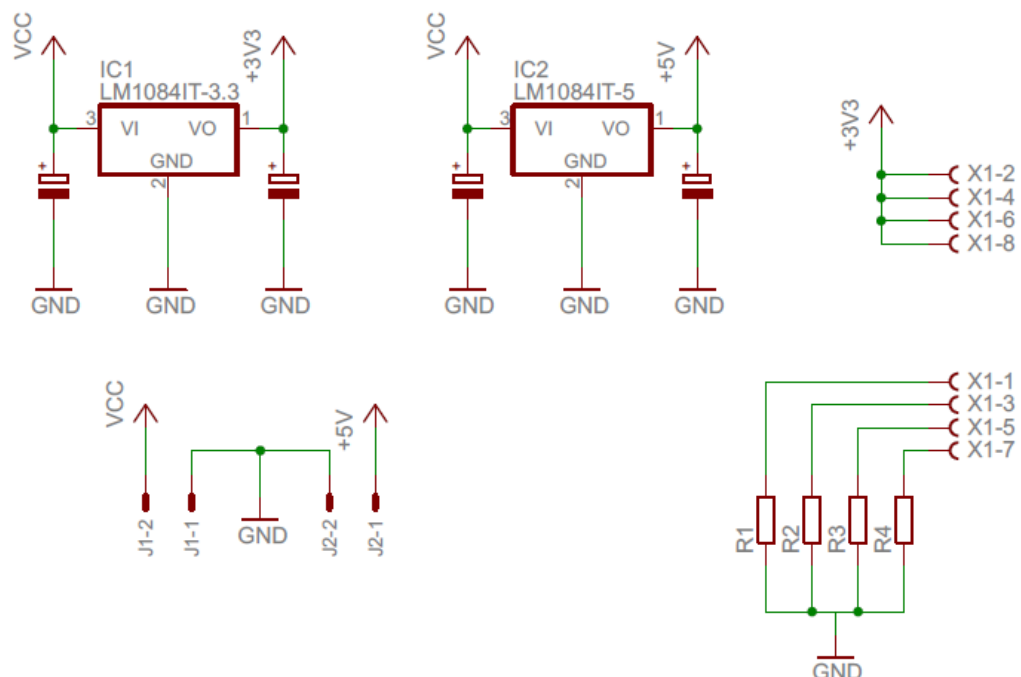
Ezekkel a gombokkal lehet váltani a különböző menü funkciók között, amelyet a szoftver részben mutatok be. A gombok illesztéséhez szükség van egyéb elektronikai elemekre is. A gombok lenyomása esetén a kártyán lévő bemenetet logikai magas szintre emelik. Mivel a nyomógombok nem ideálisak, egy lenyomás esetén akár többször is kapcsolhatnak, ezért gondoskodni kellett pergésmentesítésről. A pergésmentesítés szoftveresen lett megvalósítva.

3.3. Elektronika

Az akkumulátor egy 2 cellás 7,4V-os és 800mAh-s és lítium-polimer típusú. A vezérlő kesztyű átlagos áramfelvétele 150mA, így legalább 3-4 óráig bírja az akkumulátorral a kesztyű az akkumulátor újratöltése nélkül.

A távvezérlő kesztyű elektronikája tartalmaz egy, már a Légpárnás jármű felépítésénél bemutatott modul áramkört.

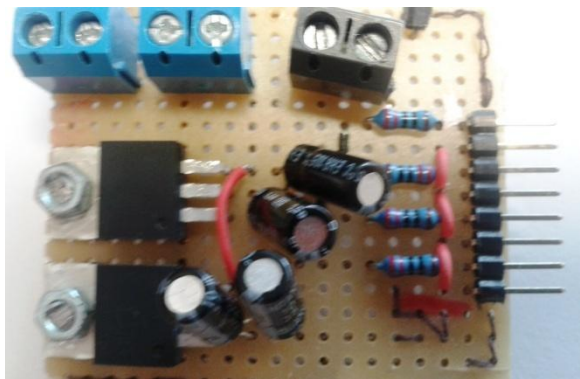
Ezen kívül már csak egy tápegység panelt tartalmaz, amelynek a kapcsolási rajza a következő képen (39. ábra) látható.



39. ábra - kesztyű tápegység kapcsolási rajz

A két integrált áramkör lineáris feszültség stabilizátor. Az egyik az LM1084IT-3.3, erre azért szükséges, mert a gombok magas szintű jelének előállításához szükség van egy referencia feszültségre. A másik integrált áramkör az LM1084IT-5, a STM32F303 fejlesztő kártya 5V-os tápfeszültségét állítja elő. Az integrált áramkör mellett elhelyezkedő szűrő kondenzátorokat 10 μ F értékre méreteztem az integrált áramkörök tervezési ajánlása alapján. A X1 portra csatlakoznak a gombok. Ahogy látható, a gomb egyik végét 3.3V-os referencia feszültség szintre van kötve, a másik láb pedig egy ellenálláson keresztül a föld potenciálra kerül. A kártyára az ellenállás és a nyomógomb közös pontját kötöttem, így ha a nyomógombot lenyomom, a kártya lába a kapcsolón keresztül 3.3V-os potenciálra kerül, viszont zárt állapotban az ellenálláson keresztül föld potenciálra esik. A lehúzó ellenállások értékeit 4,7k Ω -ra méreteztem.

Ez a kapcsolás egy próba nyomtatott áramkörön lett megépítve. A próba nyomtatott áramkör csak lyukakat és forszemeket tartalmaz. Az összeköttetéseket vezetékezéssel lehet megvalósítani. A kész áramkör a következő képen (40. ábra) látható:



40. ábra - Kesztyű tápegység NYÁK

3.4. Szoftver

A szoftver alapját az STM32F303 kártyához a gyártó cég által kiadott példaprogram adta, amely tartalmazza a giroszkóp adatainak vételezését és a megkapott nyers adatok átszámítását elfordulási szögre. A saját munka a gombok illesztése valamint a soros adatátvitel megvalósítása volt.

A szoftver fejlesztéséhez a CoCoX nevű fejlesztő szoftvert használtam. Az ARM programozásához a CMSIS könyvtárat vettem igénybe. Ez a Cortex áramkör családhoz kifejlesztett könyvtár. A rendszer alapbeállításának folyamatát segíti elő.

Háromféleképpen, azaz három menü szerint működik a szoftver. Az első menü a daru irányítását szolgálja. A második a légpárnás jármű irányítására, végül a harmadik menü a vészstop, ami leállít minden motort a légpárnás járművön. Minden menü folyamat lefutása közben kiolvasásra kerülnek a gyorsulásmérő adatai, és ezen adatok alapján feltöltésre kerül egy adat keret, amit a soros UART kommunikációs interfészen keresztül továbbküldünk a bluetooth felé. Minden egyes menüben különböző kap értéket a keret.

Keret adatai:

1. Keretfej
2. Egyes szervó motor állapota
3. Kettes szervó motor állapota
4. Hármasszervó motor állapota
5. Meghajtó motor fordulatszáma
6. Felfújó motor fordulatszáma
7. Kibekötés engedélyezése
8. Keretláb

A keretfej és a keretláb a keret azonosítására szolgáló konstans adat. A többi paraméter lehet változó (gyorsulásmérőből származó adat) vagy konstans (pl.: ha éppen egy motor ki van kapcsolva) értékű.

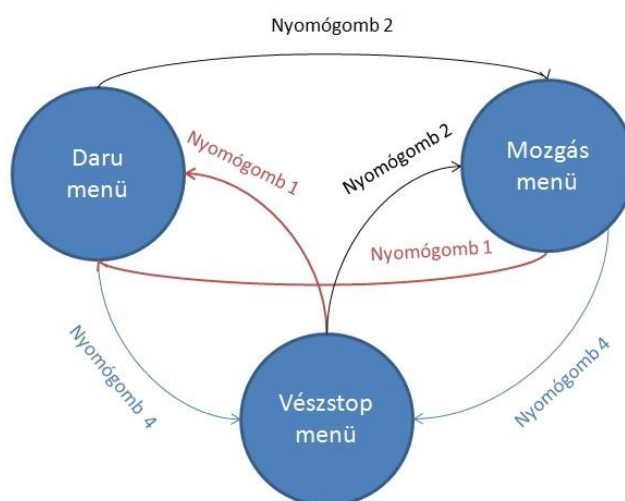
Az adatkeretek feltöltésének szabályait a különböző menükben a következő táblázat (9. táblázat) szemlélteti

Keretek adatainak értékei		
Daru menü	Mozgás Menü	Vészstop menü
1. konstans	1. konstans	1. konstans
2. giroszkóp x	2. konstans alapállás	2. konstans alapállás
3. giroszkóp y	3. konstans alapállás	3. konstans alapállás
4. konstans középállás	4. giroszkóp x	4. konstans középállás
5. konstans állj	5. giroszkóp y	5. konstans állj
6. konstans állj	6. konstans maximális	6. konstans állj
7. gombnyomásra változó	fordult	7. konstans nincs
8. konstans	7. konstans nincs	kilökés
	8. konstans	8. konstans

9. táblázat - Adatkeret elemeinek feltöltése, a különböző menükben

3.4.1. Menü rendszert szemléltető állapotgép

A Távvezérlő kesztyű működését állapotgéppel modellezve a lehet a legjobban szemléltetni (41. ábra). Az állapotváltozások a nyomógombok lenyomásakor felfutó élre történnek.

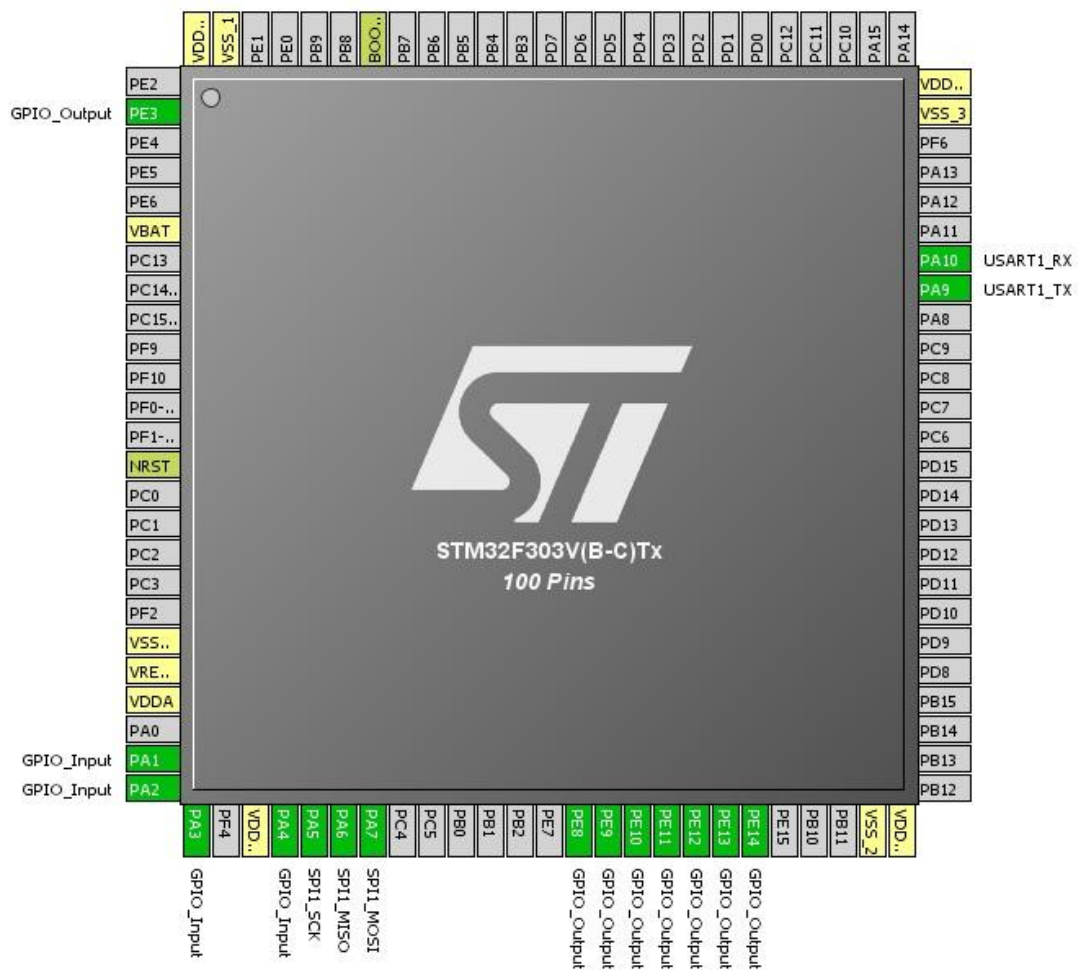


41. ábra - Menüválasztó állapotgép

Az első nyomógomb lenyomása hatására minden esetben az Daru menübe lép a program, ehhez hasonlóan a második nyomógomb lenyomása esetén a mozgás menübe lép. A negyedik gombot hívhatnánk vészstop gombnak is, mert a vészstop menübe lépteti a programot.

3.4.2. Felhasznált perifériák, és konfigurációjuk

A perifériák konfigurálásához felhasználtam a MicroExplorer programot, amely grafikus felületével segíti a konfigurálást. A következő képen (42. ábra) látható a grafikus felület, amelyen felhasznált lábak ki vannak emelve.



42. ábra - STM32F303 konfiguráció (MicroExplorer)

IO port kivezetések és funkcióik:

- GPIO bemenet (PA1:PA4) az új végi gombok csatlakozása
- USART 1 (PA9:PA10) Read Data és Transmit Data lábak kivezetései. A bluetooth kommunikációs interfészhez csatlakoznak.

- SPI 1 (PA5:PA7) Serial Clock és MISO, MOSI porton keresztül érkeznek a giroszkóp adatai
- GPIO (PE8:PE14) kimenet a LED-eket vezérlő lábak.

A GPIO bemenetek, amelyek a menüváltásokat segítik elő, megszakításos üzemmódban működnek. Ha a bemenet jelváltozást (felfutó élet) észlel, akkor egyből küld egy megszakításkérélmeket a processzor megszakítás vezérlőjének. Ennek hatására a futó program kilép, és végrehajt egy szubrutint. A szubrutinban szoftveres pergésmentesítés végrehajtása történik először, majd új értéket kap a menü váltásért felelő változó. Minden egyes gomb külön megírt szubrutint tartalmaz a szoftverben, és külön vezetéken keresztül csatlakozik a megszakítás vezérlőbe

Az USART beállításai megegyeznek a ZYBO-n és az bluetooth modulokon integrált UART interfészével. Az USART névben lévő plusz egy S-betű arra utal, hogy igény szerint képes szinkron kommunikációra is, amely plusz egy órajel lábkivezetést jelent a két adatvezeték mellett. Ez a lehetőség nem lett kihasználva. Az USART úgy lett konfigurálva, hogy ezt az órajel lábat ne használja.

Irodalomjegyzék

- [1.] ZYBO ReferenceManual, Digilent 2014. Online: http://www.digilentinc.com/Data/Products/ZYBO/ZYBO_RM_B_V6.pdf
- [2.] Crockett L.H.,Eliott R. A., Enderwitz M. A, Stewart R. W.,*The ZynqBook*,,Department of Electronic and ElectricalEngineering, University of Strathclyde, Glasgow, Scotland, ISBN 978-0-9929787-0-9, 2014, pp.460.
- [3.] 7 Series FPGA Overwiev, Xilinx 2014. Online: http://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf
- [4.] UM1570 Usermanual, © 2013STMicroelectronics. Online: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00063382.pdf
- [5.] 7 Series FPGAs Configurable Logic Block, Xilinx 2014
- [6.] 7 Series FPGAs Memory Resources, Xilinx 2014
- [7.] 7 Series FPGAs DSP48E1 Slice, Xilinx 2014
- [8.] 7 Series FPGAs Clocking Resources, Xilinx 2014
- [9.] 7 Series FPGAs and Zynq-7000 All Programmable SoC XADC Dual 12-Bit 1 MSPS Analog-to-Digital Converter, Xilinx 2014
- [10.] Scherer Balazs, Csordas Peter Nagyteljesitmenyő mikrovezérők Bevezetés: A mikrovezérők fejlődése előadás anyaga, 2011 BME
- [11.] Introduction to Embedded System Design using Zynq, Xilinx 2012

Ábrák jegyzéke

1. ÁBRA - ZYBO KÁRTYA	- 7 -
2. ÁBRA - ZYNQ AP SOC ARCHITEKTÚRA [1.]	- 9 -
3. ÁBRA - PL ARCHITEKTÚRA	- 10 -
4. ÁBRA - CLB	- 11 -
5. ÁBRA - DUPLA PORTOS BRAM	- 12 -
6. ÁBRA - DPS BLOKKVÁZLAT	- 13 -
7. ÁBRA - ADC BLOKKVÁZLAT	- 14 -
8. ÁBRA - AXI SÍNRENDSZER CSATLAKOZÁSA A PS RENDSZERBE	- 15 -
9. ÁBRA - PMOD FELÉPÍTÉSE	- 16 -
10. ÁBRA - PMODOK A ZYBO KÁRTYÁN	- 17 -
11. ÁBRA - STM32F3 FEJLESZTŐKÁRTYA	- 18 -
12. ÁBRA - SENZOR KOORDINÁTA RENDSZERE	- 20 -
13. ÁBRA - LED-EK	- 20 -
14. ÁBRA - XPS ABLAK	- 22 -
15. ÁBRA - SDK ABLAK	- 24 -
16. ÁBRA - PLANÁHEAD ABLAK	- 25 -
17. ÁBRA - LÉGPÁRNÁS JÁRMŰ	- 27 -
18. ÁBRA - SZERVO MOTOR VEZÉRLŐ JELEK	- 29 -
19. ÁBRA - SZERVO MOTOROK SZÁMOZVA	- 30 -
20. ÁBRA - LÉPTETŐMOTOR VEZÉRLŐVEL	- 31 -
21. ÁBRA - ESC	- 32 -
22. ÁBRA - BTM 222 BLUETOOTH MODUL KAPCSOLÁSI RAJZ	- 33 -
23. ÁBRA - BTM 222 NYOMTATOTT ÁRAMKÖR TERV	- 34 -
24. ÁBRA - LÉGPÁRNÁS JÁRMŰ ELEKTRONIKAI BLOKKVÁZLATA	- 35 -
25. ÁBRA - PERIFÉRIA ENTITÁS	- 36 -
26. ÁBRA - REGISZTER ÍRÁS IDŐDIAGRAM	- 37 -
27. ÁBRA - REGISZTER OLVASÁS IDŐDIAGRAM	- 38 -
28. ÁBRA - PWM JELALAKOK	- 39 -
29. ÁBRA - PWM MODUL PROCESS 1	- 41 -
30. ÁBRA - PWM MODUL PROCESS 2	- 41 -
31. ÁBRA - PWM MODUL PROCESS 3	- 42 -
32. ÁBRA - LÉPTETŐMOTOR VEZÉRLŐ JELEI	- 43 -
33. ÁBRA - LÉPTETŐMOTOR PROCESS	- 44 -
34. ÁBRA - PERIFÉRIÁK TÖMBVÁZLATA	- 45 -
35. ÁBRA - SDK SZOFTVER FOLYAMATÁBRA	- 48 -

36. ÁBRA - UART SOROS JEL.....	- 52 -
37. ÁBRA - VEZÉRLŐ KESZTYŰ I.	- 54 -
38. ÁBRA VEZÉRLŐ KESZTYŰ II.	- 55 -
39. ÁBRA - KESZTYŰ TÁPEGYSÉG KAPCSOLÁSI RAJZ	- 56 -
40. ÁBRA - KESZTYŰ TÁPEGYSÉG NYÁK	- 57 -
41. ÁBRA - MENÜVÁLASZTÓ ÁLLAPOTGÉP.....	- 59 -
42. ÁBRA - STM32F303 KONFIGURÁCIÓ (MICROEXPLORER)	- 60 -

Táblázatok jegyzéke

1. TÁBLÁZAT - AXI PORT.....	- 16 -
2. TÁBLÁZAT - PERIFÉRIA REGISZTEREINEK CÍMZÉSE	- 38 -
3. TÁBLÁZAT - PWM PERIFÉRIA REGISZTEREI	- 40 -
4. TÁBLÁZAT - LÉPTETŐMOTOR VEZÉRLŐ PERIFÉRIA.....	- 44 -
5. TÁBLÁZAT - PERIFÉRIÁK LÁBKIOSZTÁSA ÉS MEMÓRIACÍMZÉSE	- 46 -
6. TÁBLÁZAT - FOGYADOTT ADATKERET	- 49 -
7. TÁBLÁZAT – PWM PERIFÉRIÁK REGISZTEREINEK BEÁLLÍTÁSAI	- 50 -
8. TÁBLÁZAT - LÉPTETŐMOTOR VEZÉRLŐ PERIFÉRIA REGISZTEREINEK BEÁLLÍTÁSAI.....	- 51 -
9. TÁBLÁZAT - ADATKERET ELEMINEK FELTÖLTÉSE, A KÜLÖNBÖZŐ MENÜKBEN	- 59 -

Rövidítések

1. ZYBO – ZynqBoard
2. VHDL - VHSIC Hardware DescriptionLanguage
3. AP SoC - AllProgrammable System on Chip
4. XADC – XilinxAnalog Digital Converter
5. ARM – Advanced RiscMachine
6. APU – Application Processing Unit
7. SPU – Snoop Processing Unit
8. UART - Universal Asynchronous Receiver/Transmitter
9. USART Universal Synchron Asynchronous Receivert/Transmitter
10. SDK – Software Development Kit
11. XPS – Xilinx Platform Studio
12. FPGA – Field Programable Gate Array
13. ARM – Advanced RISC Machine

14. RISC – Reduced Instruction Set Computing
15. AMBA – Advanced Microcontroller Bus Architecture
16. AXI – Advanced eXtensible Interface
17. PL – Programable Logic
18. PS – Processing System
19. USB – Universal Serial Bus
20. SDIO – Secure Digital Input Output
21. SPI - Serial Peripheral Interface
22. CAN – Controller Area Network
23. CLB – Configurable Logical Block
24. LUT – LookUp Table
25. BRAM – Block RAM
26. FIFPO – First In First Out
27. PLL – Pulse Lock Loop
28. MMCM – Mix Mode Clock Manager
29. DPS – Digital Signal Processors
30. ALU – Arithmetic Logic Unit
31. XADC – Xilinx Analog Digital Converter
32. MSPS – Mega Sample Per Second
33. IOB – Input Output Block
34. ACP – Accelerator Coherency Port
35. GP – General Purpose
36. HP – High Performance
37. MIO – Master Input Output
38. PWM – Pulse-Width Modulation
39. TTL – Transistor Transistor Logic

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani azoknak az embereknek, akiknek köszönhető, hogy ez a projekt munka létrejöhett.

- Konzulensemnek Dr. Vásárhelyi József egyetemi adjunktusnak.
- Sipos Attilának a Magyarok a Marson versenyek főszervezőjének
- Bartók Csabának és Szabó István egykori A-TEAM csapattagoknak, akik támogatásukkal segítették elő a csapat munkáját
- A 2014-es A-TEAM csapat tagjainak a projektben való munkájukért