

SZAKDOLGOZAT

Pap Zsolt

2022

Magyar Agrár- és Élettudományi Egyetem
Károly Róbert Campus

**Távirányítással vezérelhető
fegyverplatform tervezése és
kivitelezése**

Konzulens

Dr. Pántya Róbert

egyetemi adjunktus

Készítette

Pap Zsolt

Gazdaságinformatikus szak

levelező tagozat

2022

Tartalomjegyzék

1. BEVEZETÉS	5
2. IRODALMI ÁTTEKINTÉS	7
2.1 A robot	7
2.2 Az egykártyás számítógép és a mikrovezérlő kártyák (fejlesztői lapok).....	9
2.2.1 Egykártyás számítógép	9
2.2.2 Mikrovezérlő kártya	10
2.3 Vezérléshez használt mikrovezérlő kártya.....	12
2.3.1 NodeMCU-32s esp32.....	13
2.3.2 Kommunikációs lehetőségek.....	14
2.4 Irányítás, vezeték nélküli kommunikáció	15
2.5 Fejlesztői környezet, Arduino IDE	16
2.5.1 Az Arduino IDE bemutatása	16
2.5.2 Fontos kezdeti lépések.....	17
2.5.3 A programozás megkezdése.....	18
3. ANYAG ÉS MÓDSZERTAN	21
3.1 Airsoft P90, CM.060.....	21
3.2 KONECT 31 kg CORELESS HiVOLT WATERPROOF Digital servo (31kg-0.09s / 60 °).....	22
3.3 DC-DC konverter modul (step-down)	24
3.4 Relé modul.....	25
4. A PROJEKT FOLYAMATAI	26
4.1 Tervezés	26
4.1.1 A fegyvertorony kialakítása	26
4.1.2 Vezérlés tervezése.....	27
4.2 A vezérlőkód bemutatása.....	28
4.2.1 Setup és loop függvények	28
4.2.2 Kommunikáció, irányítás	30
4.3 Összeszerelés, tesztelés.....	32

5. KÖVETKEZTETÉSEK ÉS JAVASLATOK	35
5.1 Biztonsági fejlesztések.....	35
5.2 Mobilitás, hordozhatóság.....	36
5.3 Energetikai ellátásbiztonság.....	37
5.4 „SCI-FI” megoldások	37
6. ÖSSZEFOGLALÁS.....	38
7. IRODALOMJEGYZÉK.....	41
8. ÁBRAJEGYZÉK	44
9. TÁBLÁZATOK.....	45
10. MELLÉKLETEK.....	46
11. KÖSZÖNETNYILVÁNÍTÁS.....	48

1. BEVEZETÉS

Szakdolgozatom témája több indíttatásból is fakad. Elsősorban személyes jellegű motiváció, érdeklődés határozta meg választásomat, másodsorban munkahelyi feladatok, igények. Már tizenéves koromtól érdeklődök az informatika iránt. A technológiák fejlődése mind a hardverek, mind a szoftverek tekintetében kiemelkedő gyorsasággal megy végbe napjainkban. Ezt a sebességet nagyban elősegíti a hatékonyság növelése és az automatizálás bevezetése. Ezzel el is érkeztem ahhoz a ponthoz, ami a mai gondolkodásomat alapjaiban határozza meg. Ez pedig a hatékonyság folyamatos növelése és az automatizálás alkalmazása minden területen, ahol a lehetőségek megengedik. Olyan szerencsés helyzetben találtam magam a Szakdolgozat témaválasztás idején, hogy a listában megtaláltam egy olyan területet, ami igazán felkeltette az érdeklődésemet. Ez volt a „Robotok alkalmazása a gyakorlati életben.” témakör. Ekkor még nem volt teljes elképzelésem arról, hogy mit is fogok kezdeni ezzel a címmel, de azt tudtam, hogy ezt nekem találták ki. Ezzel közel egy időben a munkahelyemen elkezdtek olyan alkatrészek beérkezni a logisztikára, amik felébresztették a kíváncsiságomat. Értem ez alatt, hogy egy igényléshez több olyan eszköz és alkatrész érkezett, amik abba az irányba mutattak, hogy valamilyen robotizált céleszköz fejlesztése vette kezdetét. Kisebb kutakodás után megtaláltam a szervezeten belül, hogy hova tartoznak a megrendelések. Rövid megbeszélést követően be is kerültem a fejlesztői csapatba.

A feladatunk az volt, hogy a Magyar Honvédség (a továbbiakban: **MH**) kérésére fejlesszünk egy olyan eszközt, amivel különböző események alkalmával, látványos bemutatókat lehet megvalósítani. Az elsődleges brainstorming¹ megbeszélésekből kimaradtam, így az irány már adott volt, amikor becsatlakoztam a fejlesztésbe, azonban a későbbiekben közel a teljes megvalósítás az én feladatom lett. Természetesen nem kaphattam meg felügyelet nélkül a teljes projektet, hiszen a döntéshozóknak nem volt a birtokában a képességeim, tudásom szintje. Ezért mentorként mellém helyeztek egy vezető fejlesztőmérnököt.

¹ egy csoportos kreativitástechnika, amellyel egy konkrét probléma megoldására törekednek, összegyűjtve a tagjai által spontán ötletekkel. (<https://en.wikipedia.org/wiki/Brainstorming>)

A fejlesztendő eszköz méretére és működésére olyan behatárolások születtek, hogy:

1. használat közben helyhez kötöttnek, kompaktnak kell lennie.
2. logisztikai szempontból nézve ne igényeljen külön szállítójárművet az eszköz és a kiegészítők mozgatása, emellett könnyen telepíthető legyen.
3. vezérlés tekintetében egyszerűen kezelhetőnek kell lennie.

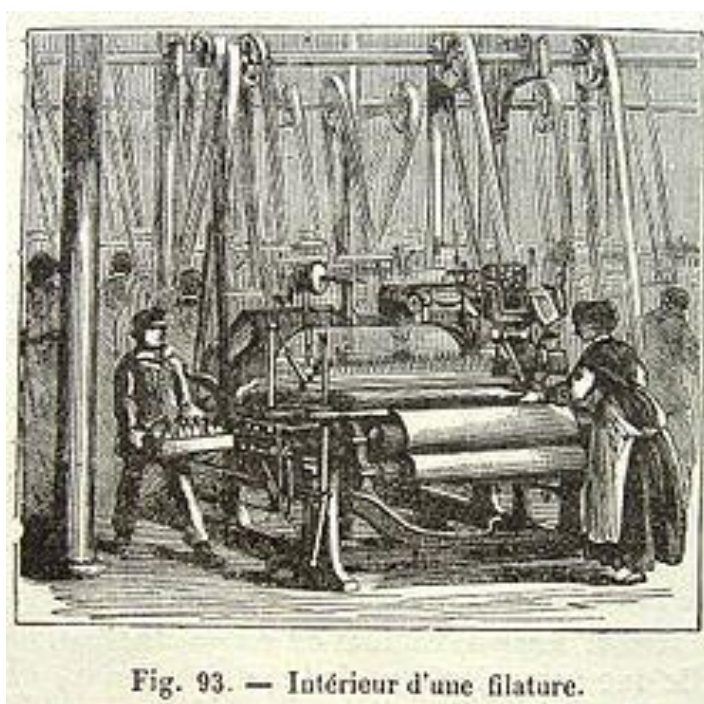
Ezen megkötések figyelembevételével született meg az elsődleges koncepció. A kérést előterjesztő MH és a vállalat profilját szem előtt tartva, megmaradt a „valamilyen” fegyverzet, mint kivitelezendő cél. A teljes projektre rövid, két hónapos határidő lett megszabva, emellett szűkös költségkeretet kaptam.

Szakdolgozatomban ennek az eszköznek a mechanikai és vezérlési tervezését, egyedi alkatrészeinek kialakítását, az összeállításának menetét és a működését szándékozom bemutatni.

2. IRODALMI ÁTTEKINTÉS

2.1 A robot

Felgyorsult világunkban a robotok fokozatosan egyre nagyobb szerepet töltenek be életünkben. Legyen szó akár polgári (űrkutatási-, ipari-, gyógyászati-, kutatási-, hobbi etc. felhasználás), akár harcászati felhasználásról. Elsődlegesen ipari területen történtek áttörések a robotok alkalmazásában. A 18. századi ipari forradalomnak egyfajta velejárója volt a fejlődés. Ekkor kezdtek megjelenni különböző automatizált eszközök az óragyártás és ezáltal a textilipar részéről.



1. ábra - Szövőgép a 19. században (WIKIPÉDIA, 2022)

A 20. század közepén leginkább az autóiparban indult be a robotok alkalmazása, elsősorban festő és hegesztő robotok formájában. Ezt követően a gyógyászatban, élelmiszeriparban és egyéb szektorokban is megindult a robotok térnyerése. (STRUIJK, 2014)

A „robot” szót Karel Čapek használta először az 1921-ben bemutatott R.U.R (*Rossum's universal robots*) című drámájában. A R.U.R. műben megalkotott robot idővel teljesen önálló lesz, és a megalkotója ellen fordul, majd rabszolgájává teszi az embert.

(WIKIPÉDIA, 2021) (SZABOLCSI & MENYHÁRT, 2015) (MENYHÁRT, 2017)

Ezt követően már más írók is elkezdtek használni a kifejezést, többek között a híres sci-fi író, Isaac Asimov.

1942-ben Asimov fektette le a robotika három alaptörvényét: (SZABOLCSI & MENYHÁRT, 2015)

1. A robotnak nem szabad kárt okoznia emberi lényben, vagy tétlenül tűnie, hogy emberi lény bármilyen kárt szenvedjen.
2. A robot engedelmeskedni tartozik az emberi lények utasításainak, kivéve, ha ezek az utasítások az első törvény előírásaiba ütköznének.
3. A robot tartozik saját védelméről gondoskodni, amennyiben ez nem ütközik az első vagy második törvény bármelyikének előírásaiba.

A **robot**, mint eszköz meghatározása az ISO² 8373:2021³-es számú szabvány alapján a robot: „Olyan előre programozott mechanikai szerkezet, amely különböző mértékig, de emberi beavatkozás nélkül, szenzorok vagy valamilyen állapot alapján képesek mozgásra, manipulálásra vagy pozicionálásra. Rendelkeznie kell valamilyen irányító berendezéssel, ami a robot viselkedését monitorozza és kontrollálja. A robot lehet helyhez kötött vagy mozgó eszköz.” (WIDSTRÖM, 2021)



2. ábra - Stanford Arm – 1969 (WIEDERHOLD, 2000)

² ISO: the International Organization for Standardization – Nemzetközi Szabványügyi Szervezet

³ ISO 8373:2021 - Robotics — Vocabulary

A robotok csoportosítása több szempont szerint is történhet. Az alábbi felsorolás ezek közül a legfontosabbakat tartalmazza (DR. SZABÓ, et al., 2014):

- Mobilitásuk alapján:
 - o mobil robotok
 - o helyhez kötött robotok
- Felhasználásuk alapján:
 - o anyagmozgató
 - o technológiai műveletet elvégző
 - o szociális
- Helyhez kötött robotok kinematikájuk alapján:
 - o párhuzamos kinematikájú
 - o láncolt kinematikájú
 - o vegyes kinematikájú
- Mobil robotok kinematikájuk alapján:
 - o differenciális
 - o holonomikus
 - o egyéb

2.2 Az egykártyás számítógép és a mikrovezérlő kártyák (fejlesztői lapok)

Dolgozatomban már említést tettem arról, hogy egy robotnak, vagy egy automatizált berendezésnek, rendelkeznie kell valamilyen irányító rendszerrel (control system). Ezeket a rendszereket meg lehet valósítani például egykártyás számítógéppel vagy mikrovezérlő kártyákkal is.

2.2.1 Egykártyás számítógép

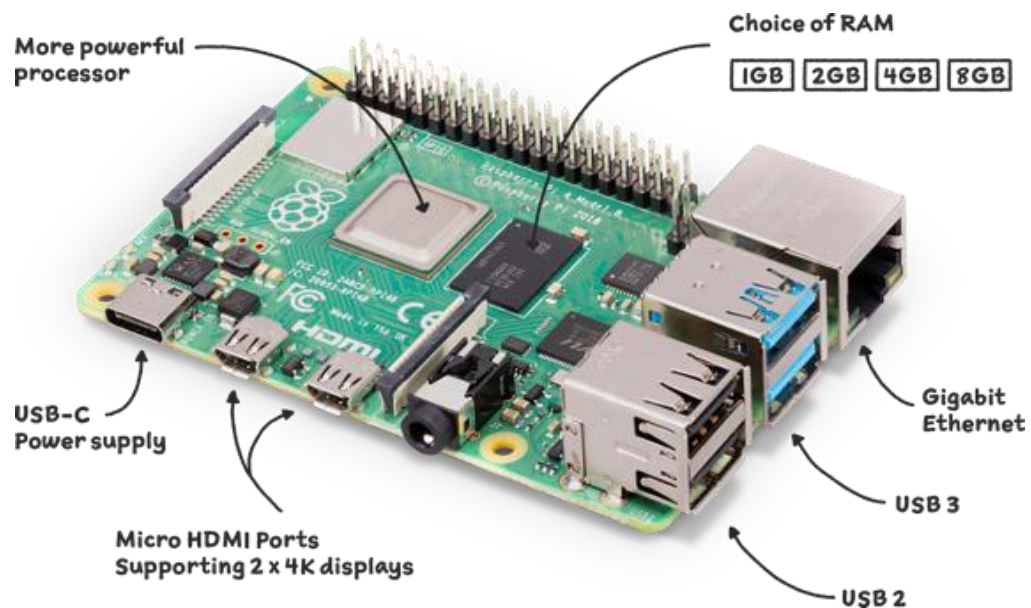
Az egykártyás számítógépek (SBC⁴) az 1970-es években jelentek meg. Elsősorban ipari környezetben történő munkavégzésre fejlesztették ki, azonban az 1980-as években a méltán népszerű Commodore 64 számítógép is ebbe a kategóriába sorolható. Napjainkban már széleskörű felhasználási lehetőséggel kecsegtet használatuk, legyen szó ipari vagy hobbi tevékenységről. Ezek a lehetőségek több mindenből adódnak. Kompakt méretük, alacsony fogyasztásuk és nem utolsósorban a megfizethetőségük, ami miatt

⁴ SBC - single-board computer

népszerűségnek örvendhetnek. Ezen tulajdonságokat kihasználva megvalósítható használatukkal:

- Okosotthon vezérlés
- Robot vezérlés
- Médiaszerver készítés, üzemelés
- NAS rendszer
- Kamerarendszer vezérlés
- Webszerver

A kártyán (alaplapon) megtalálhatóak azok az alkatrészek, chippek, csatlakozók, mint a személyi számítógépeknél azzal a különbséggel, hogy ebben az esetben a kártyára vannak rögzítve, ezért nem, vagy csak nehezen cserélhetőek elhasználódás esetén. Minden esetben fogunk találni mikroprocesszort (CPU), memória, memóriakártya foglalat, I/O csatlakozók, tápcsatlakozó. Hobbi szintű felhasználásra az egyik legnépszerűbb termékcsalád a Raspberry Pi™.



3. ábra - Raspberry Pi 4 (RASPBerry, 2022)

2.2.2 Mikrovezérlő kártya

Ezeket a fejlesztői lapokat (development board) felhasználhatjuk ipari környezetben, oktatási célra, hobbi robot építésénél, kutatási projektekben, vagy akár otthonunk eszközeinek „okosotthon” -ba való becsatornázására is. Ipari felhasználásuk korlátozott lehet, mert önmagukban, egyéb védelem nélkül nem állnak ellen a szélsőséges hőmérsékleti viszonyoknak, maró anyagoknak és a nedvességnek sem.

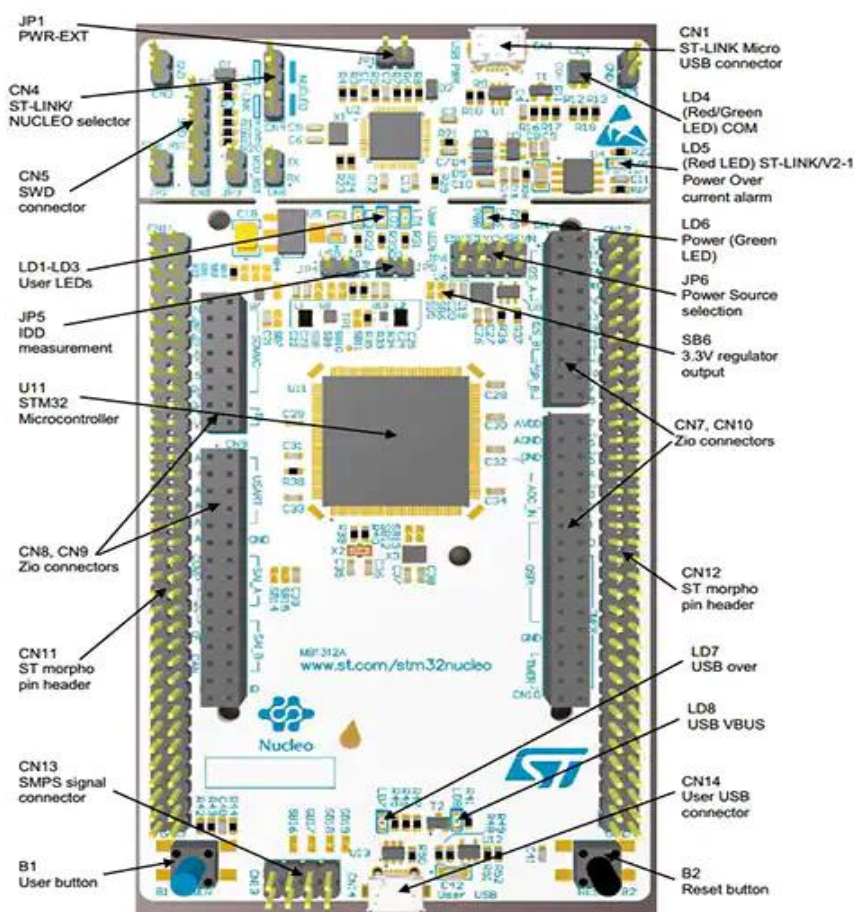
A nagyobb félvezetőgyártók a fejlesztői kártyákat elsődlegesen a mikrovezérlők támogatása érdekében készítik. A kártyák segítségével a fejlesztőmérnökök megismerhetik a kiszemelt mikrovezérlőt. Ezért ezeket úgy tervezik meg, hogy a felhasználó számára azonnal használhatóak legyenek a vezérlő program feltöltése után. Így a feladat megkezdése előtt nem kell időt áldozni a vezérlőegység hardverének megtervezésére.

A fejlesztői lapok minden esetben tartalmazznak:

- mikrokontrollert (MCU⁵)
- programozói interfészt (általában valamely típusú USB csatlakozó)
- tűfejes csatlakozókkal elérhetővé tett általános célú be- és kimenet (GPIO⁶)

A csatlakozók általában az alábbi interfészekkel rendelkeznek:

- Digitális és / vagy analóg kimenetek
- LAN, I²C, LCD, soros, PWM, CAN, SPI interfészek



4. ábra - NUCLEO-L4P5ZG board (GIOVINO, 2020)

⁵ MCU – **M**icro **C**ontrol **U**nit

⁶ GPIO - **g**eneral-**p**urpose **i**nput/**o**utput

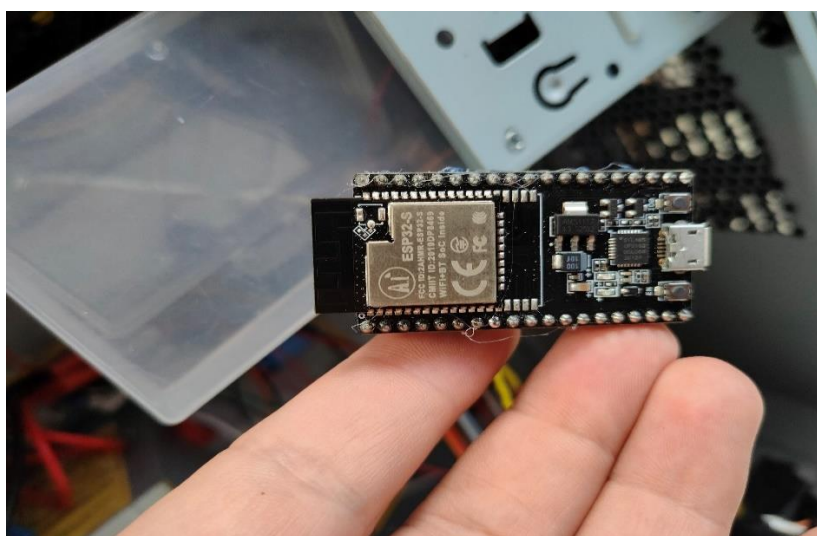
A GPIO tűskéken keresztül lehet vezérelni a csatlakoztatott eszközöket. A csatlakoztatható eszközök lehetnek:

- szenzorok (infra, ultrahang, légszennyezettség, alkohol, nyomás, széndioxid,)
- motorok (szervo, léptető, DC)
- kamerák
- LCD kijelző
- hálózati kommunikációs eszközök
- kapcsolók, gombok, relék

A felsorolásban a leggyakrabban használt eszközöket jelenítettem meg.

2.3 Vezérléshez használt mikrovezérlő kártya

A dolgozatom tárgyát képező fegyvertorony (RCWS⁷) elkészítéséhez szükségem volt egy fejlesztői kártyára, hogy a torony mozgását, a fegyver elsütését és az irányító eszközzel való kommunikációt megvalósíthassam. Korábban már dolgoztam hobbi szinten Arduino UNO kártyával, azonban vezetőim a költségek alacsonyabb szinten tartását várták el tőlem. Ez a 2022-es év elején történt, amikor a félvezetők hiánya nagyban befolyásolta az elérhető eszközök számát és minőségét. Ekkor lehetetlen volt beszerezni például Raspberry Pi termékcsalád kártyáit, az árak miatt az Arduino termékcsalád nem kerülhetett szóba. A probléma megoldását egy kollégámmal való beszélgetés oldotta meg, ő ajánlotta figyelmembe az Espressif cég termékeit, amik az ideális árkategóriába estek és raktárkészlet is biztosított volt. Így esett a választás a *NodeMCU-32s* esp32-es lapkára.

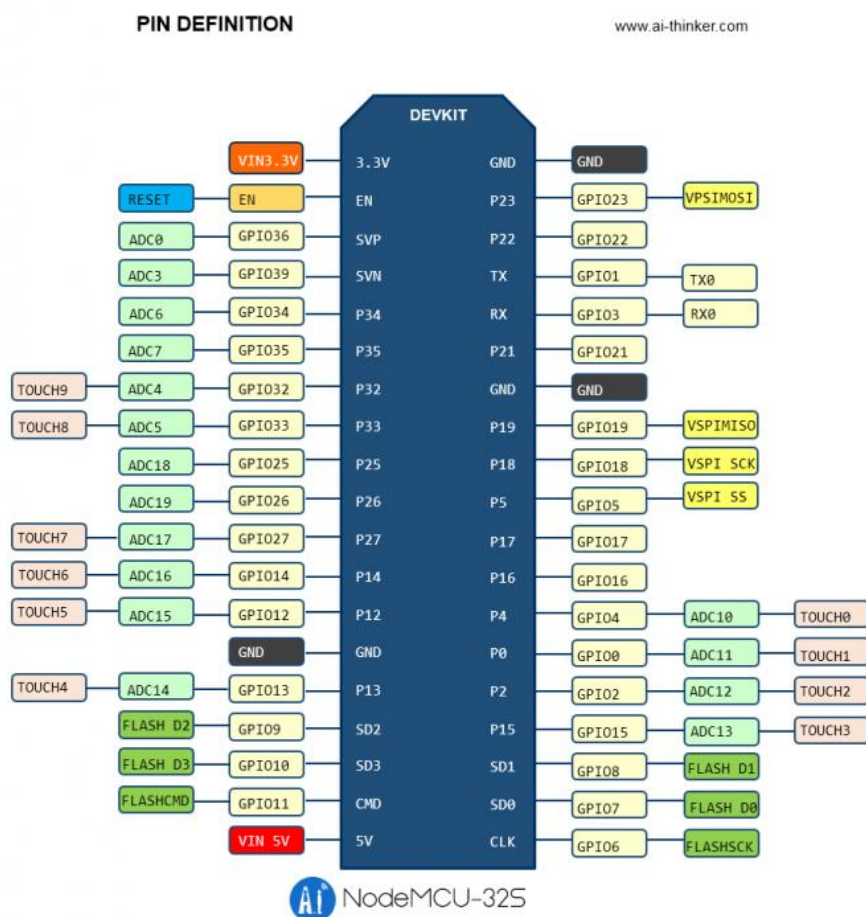


5. ábra - NodeMCU-32s esp32 (saját kép)

⁷ RCWS - Remote controlled weapon station

2.3.1 NodeMCU-32s esp32

A NodeMCU-32s esp32 az Espressif félvezető gyártó mikrovezérlő lapkáján alapul. Az ESP 32-S mikrovezérlő látja el az „agy” szerepét a kártyán. A lapka beépített wi-fi és bluetooth modullal rendelkezik. Vezeték nélküli hálózatok 802.11 b/g/n szabványait, illetve bluetooth 4.2 BR/EDR és Bluetooth LE⁸ protokollokat tudja használni. Rendelkezik USB interfésszel, melyen keresztül lehet a tápellátást biztosítani, vagy programot feltölteni. Ezek mellett található még UART, I²C, SPI interfész. A kártyán 38 db (2*19) PIN található. Ezeken a „tüskéken” keresztül lehet a tervezett egységek vezérléséhez szükséges vezetékes kapcsolatot létrehozni, illetve a tápellátás is biztosítható a megfelelő bemeneteken. A kártyán található két darab nyomógomb, az egyik a reset funkciót látja el, a másikat a felhasználó szabadon konfigurálhatja. (ESPRESSIF, 2022)



6. ábra - NodeMCU-32S lábkiosztás (AI-THINKER Technology Co., 2022)

⁸ BLE – Bluetooth Low Energy

A kártya főbb műszaki adatai:

Mikrokontroller	ESP32-WROOM-32
Üzemi feszültség	3.3V
Bemeneti feszültség (min.)	3.0V
Bemeneti feszültség (max.)	3.6V
Hőmérséklet tartomány (üzemi)	-40 ~ +85 °C
Egyenáram a 3,3 V-os kivezetésen	28 ~ 40 mA
Flash memória	32 Mbit
Órajel sebessége	40-MHz

1. táblázat - Nodemcu-32s Datasheet (AI-THINKER Technology Co., 2022)

2.3.2 Kommunikációs lehetőségek

A fejlesztői kártya képes vezetékes és vezeték nélküli kommunikációra, ezáltal egyszerre több típusú eszközzel is képes kapcsolatot teremteni. Vezeték nélküli kommunikációra a beépített wi-fi és bluetooth modulon keresztül kínál megoldást.

Vezeték nélküli kommunikációs lehetőségek:

Wi-fi:

- adatátviteli sebesség maximálisan: 150 Mbps,
- használt frekvencia 2,4 GHz.

Bluetooth:

- class-1, class-2, class-3 transzmitter-t (adó) tartalmaz (akár 100 m hatótáv)
- NZIF (Near Zero Intermediate frequency / Nulla közeli köztes frekvencia) vevője – 97 dBm⁹ érzékenységgel rendelkezik

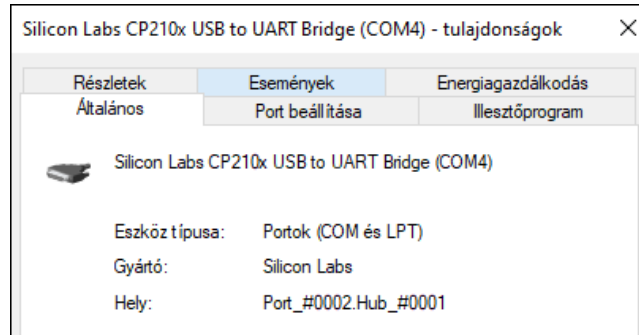
Vezetékes kommunikációs lehetőségek:

- UART (Universal asynchronous receiver transmittaer / univerzális aszinkron adóvevő), a soros és párhuzamos interfészek közötti kommunikáció fordítását végzi
- SPI (Serial Peripheral Interface / Soros periféria illesztő), a mikrokontroller és a perifériák (szenzor, shift regiszter stb.) közötti szinkronos kommunikációért felel. egy kontroller (Master) kommunikál egy vagy több perifériával (Slave)
- I²C (Inter-Integrated Circuit /Integrált Áramkörök közötti), egy vagy több kontroller (Master) kommunikál egy vagy több perifériával (Slave)

⁹ dBm - logaritmikus teljesítmény mértékegység / abszolút szint egység 1 mW (Milliwatt) teljesítményhez viszonyítva

- LED és motor PWM (**p**ulse-**w**idth **m**odulation / impulzusszélesség-moduláció), az analóg kimenő feszültségjelek, digitális impulzussorozat-jelekkel való helyettesítésére szolgál

A kártya USB kábelén keresztül kommunikál a számítógéppel USB COM porton keresztül.



7. ábra - NodeMCU-32s esp32 csatlakoztatás után (saját kép)

2.4 Irányítás, vezeték nélküli kommunikáció

Az általam készített eszköz irányítása egy Xbox[®] Series XS típusú kontrollerrel valósítható meg. A kontroller és a vezérlőegység is rendelkezik Bluetooth 4.0 és wi-fi vezeték nélkül kommunikációval. A párosítás és az azonosítás történhet a Bluetooth eszköz, jelen esetben a kontroller UUID (**U**niversally **U**nique **I**dentifier) azonosítója alapján.



8. ábra - Xbox Series XS controller (XBOX, 2022)

Az **UUID** azonosító egy 128-bites érték, ami a bluetooth eszközök által nyújtott szolgáltatások azonosítására szolgál. Az azonosítót sokszor karakterláncként szokták megjeleníteni az alábbi formátumban:

XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX

A Bluetooth Special Interest Group (SIG) számos szolgáltatás UUID azonosítóját előre meghatározza, alapvetően olyan szolgáltatásokét, amelyeknek gyakori a használata. Ezeket az azonosítókat Assigned Numbers-nek nevezi a csoport. Ezt a listát minden változás esetén azonnal frissítik, így folyamatosan naprakész információk alapján lehet fejleszteni. Azonban ezek az azonosítók „csak” 16-bitesek és a SIG alap UUID-nek nevezi. Az 16-bit adat a 128-bites UUID elején van megjelenítve. (BLUETOOTH, 2022) (SMITH, 2016)

Például a Battery service-hez a **0x180F** UUID tartozik. A 128-bites UUID:

0000180F-XXXX-XXXX-XXXX-XXXXXXXXXXXX

2.5 Fejlesztői környezet, Arduino IDE

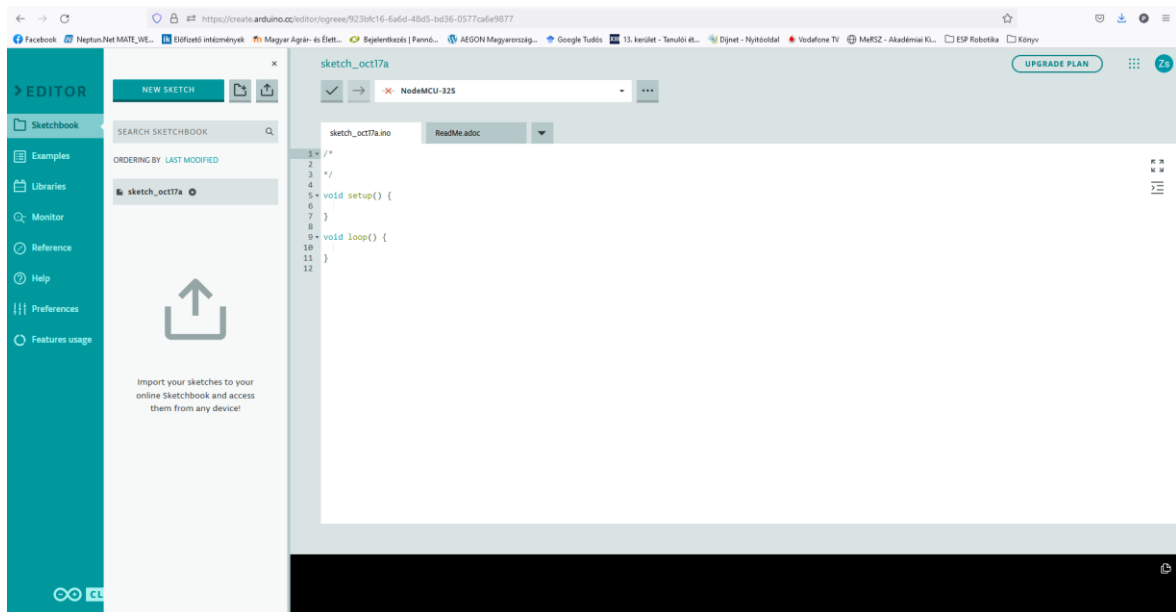
2.5.1 Az Arduino IDE bemutatása

A fejlesztői kártyánkat USB-n keresztül tudjuk felprogramozni, kommunikálni vele, valamint a fejlesztés alatt ezen keresztül tudjuk garantálni a működéséhez szüksége tápfeszültséget. A programok készítéséhez rendelkezésünkre áll egy fejlesztői környezet, amit elsődlegesen az Arduino kártyákhoz fejlesztettek ki. Ez az Arduino IDE¹⁰. A kártyák programozása a C/C++ programozási nyelveken alapszik. Az Arduino programozási nyelv három fő részre osztható (ARDUINO, 2022):

- függvényekre (functions)
- értékekre (values) / változók, konstansok
- adatszerkezetre (structure)

Az integrált fejlesztői környezetekre jellemző, hogy tartalmaznak valamilyen szövegszerkesztőt, a szükséges fordítót és hibakeresőt. (DR. NYAKÓNÉ DR. JUHÁSZ, et al., 2011) Az Arduino IDE emellett rendelkezik a mikrovezérlőtől érkező üzenetek megjelenítésére szolgáló eszközzel, a Serial monitorral és a Serial plotterrel. A programok elkészítése napjainkig csak a számítógépekre letölthető szoftverek használatával volt elérhető, azonban az www.arduino.cc weboldalon már elérhető egy online fejlesztői környezet, az Arduino Web Editor. Az oldal megnyitva az alábbi képernyő fogad minket.

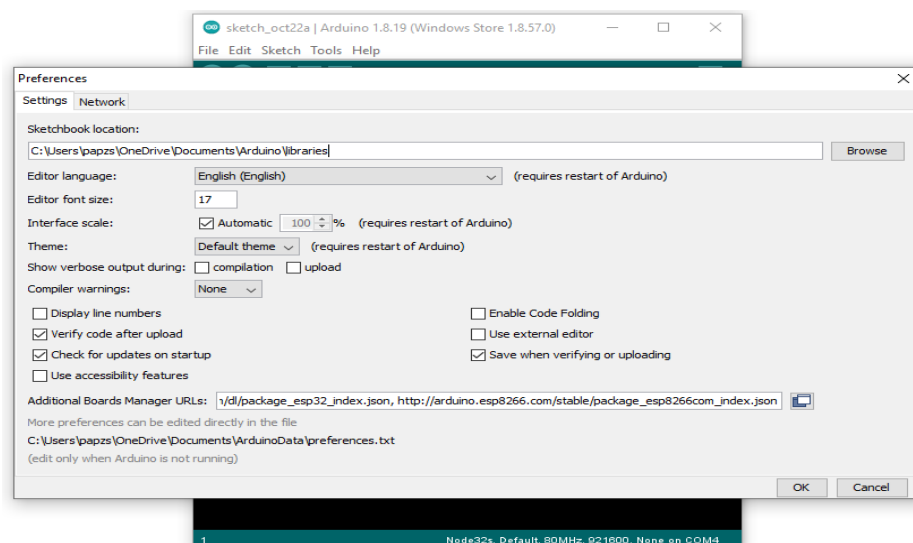
¹⁰ IDE - Integrated Development Environment / integrált fejlesztői környezet



9. ábra- Arduino Web Editor (saját kép)

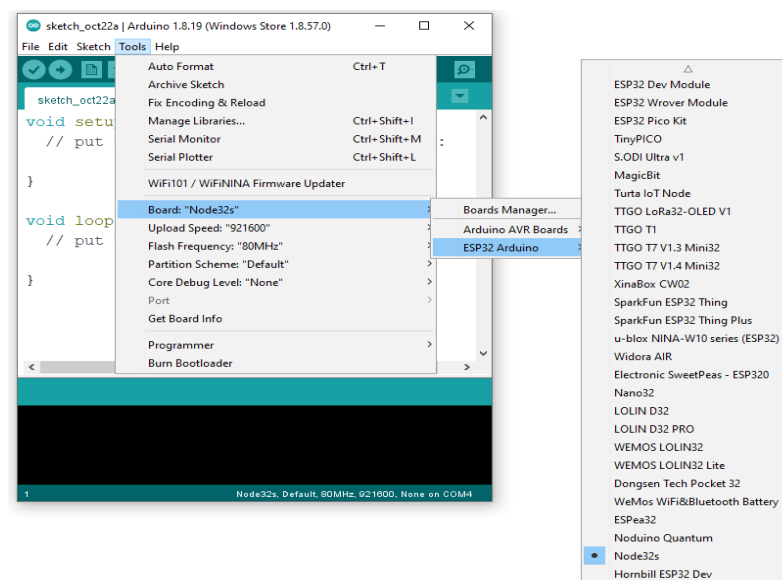
2.5.2 Fontos kezdeti lépések

Ez a felület sokban hasonlít az szoftveres verzióhoz, a továbbiakban azon keresztül mutatom be a fontosabb jellemzőit a szoftvernek. Az első fontos lépés a letöltés követően, hogy a környezetet alkalmassá kell tenni a fejlesztői lapunk kezelésére, mert alapesetben csak Arduino kártyák kezelésére van felkészítve. Ezt egyszerűen megtehetjük a File/Preferences menüpontban alatt, az „Additional Boards Manager URLs:” beviteli mezőben. A mezőbe azoknak csomagoknak (package) az elérését kell megadni, amik tartalmazzák a fejlesztői kártyánkhoz szükséges adatokat. Amennyiben több URL-t is szeretnénk megadni, úgy azokat vesszővel elválasztva megtehetjük. Ezzel a lépéssel felkészítettük a szoftvert a kártyánk kezelésére.



10. ábra - Arduino IDE /Preferences/ (saját kép)

Következő lépésként ki kell választanunk a kezelendő kártyát. Ezt a Tools/Board menüpontban tudjuk megtenni. Figyelnünk kell arra, hogy pontosan válasszuk ki a listából a lapkánkat. Amennyiben nem találjuk a listában, vissza kell térnünk a korábban bemutatott lépéshez. A megfelelő kártya kiválasztása után lehetőségünk van a kártya feltöltési sebességének és a használandó órajel frekvenciájának beállítására.



11. ábra - A használni kívánt kártya kiválasztása (saját kép)

2.5.3 A programozás megkezdése

Amennyiben sikeresen be tudtuk állítani a lapkánkat, akkor meg is tudjuk kezdeni a programozást. Kezdeként egy vázlatkódot láthatunk, ami két függvényt tartalmaz. Az egyik a **setup** függvényt, ahova célszerű olyan függvényeket írni, amikkel az alapbeállításokat kezelhetjük. Ez a része a kódnak csak egyszer fut le a program indulásakor. A másik függvény a **loop** függvényt, ami gyakorlatilag egy végtelen ciklus. Az ebbe a részbe írt kódok folyamatosan ismétlődnek. A teljes programunk ezek alapján három részből áll:

1. a használni kívánt *könyvtárak* jelölése; változók megadása, saját függvények deklarálása
2. setup függvény
3. loop függvény

Itt tartom fontosnak megjegyezni, a könyvtárak használatát. Az Arduino IDE telepítésekor az általánosan használatos könyvtárak is telepítésre kerülnek, de ezek korlátozott programozási lehetőségeket biztosítanak. Magunk is megírhatnánk az eszközök működéséhez, kommunikációjához a szükséges kódokat, de ez már magas programozási

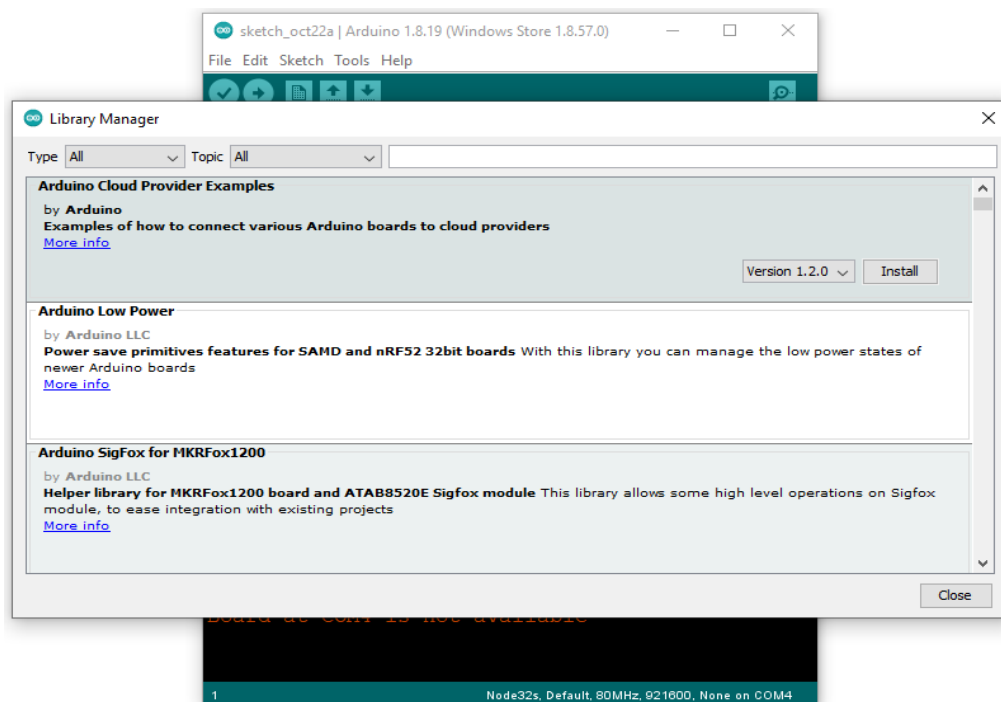
készségeket kíván. A programozó társadalom magasan képzett képviselői ezt megteszik helyettünk, nekünk már csak jól kell tudnunk felhasználni tudásukat. Erre szerencsére lehetőségünk van. A meglévő és a jövőbeni könyvtárakat a Tool/Manage Libraries menüpontban kezelhetjük. Lehetőségünk van újabb könyvtárak telepítésére, illetve a meglévő könyvtárak frissítésére. A „More info” linkre kattintva betekintést nyerhetünk a telepíteni kívánt könyvtár tartalmába!

A teljes lista szűrésére több lehetőségünk is rendelkezésre áll. A „Type” legördülő menüben az alábbiak szerint szűkíthetjük a listát:

- korábban már telepített
- Arduino vagy „Partner” által közzétett
- ajánlott, támogatott, visszahívott

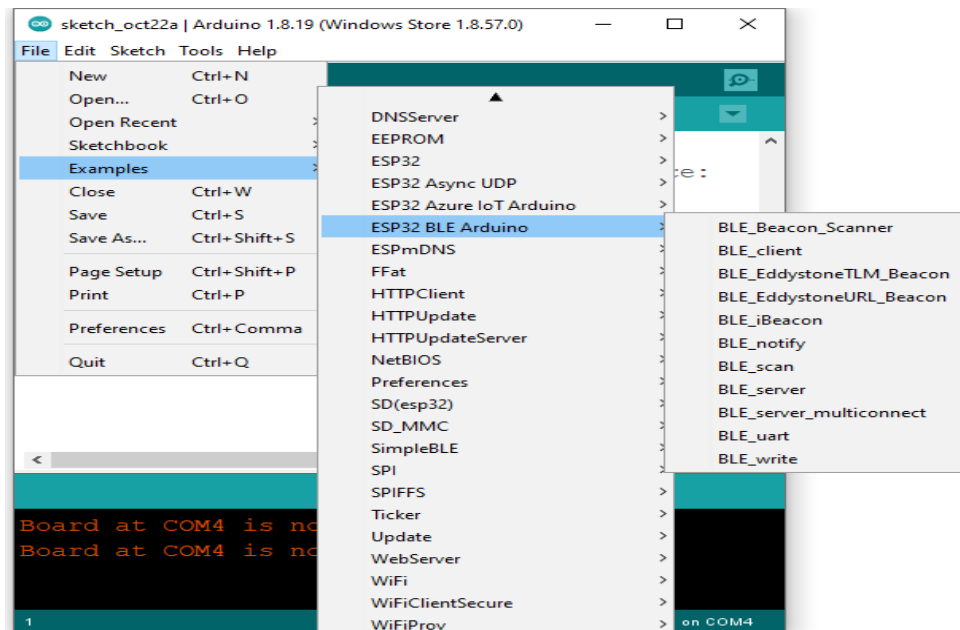
A „Topic” menüben felhasználási terület szerinti szűrést végezhetünk:

- kommunikáció
- Adatkezelés, adattárolás
- eszközkezelés
- kijelző
- szenzorok, jelek
- időzítés



12. ábra - Könyvtárak kezelése - Arduino IDE (saját kép)

A telepítést követően a könyvtárakat azonnal használatba is tudjuk venni. A telepítéssel kapunk még „bónuszként” egy bemutató vagy példakódot (Example). Ezek a példakódok megtalálhatóak és felhasználhatóak a File/Examples menüpont alatt. Ezeken a példakódokon keresztül könnyebben megérthetjük a működését a fejlesztői környezetnek, a programozási nyelv „speciális” használatának, perifériák kezelésének.



13. ábra - Példakód kiválasztása (saját kép)

Programunkban a könyvtárak használatát a `#include <LibraryFile.h>` *structure*-ral kezdhetjük meg. A programozás során használható függvényeket, változókat, szerkezeteket az dolgozatom 1. számú melléklete tartalmazza. Újabb könyvtárak letöltésével azonban rendelkezésünkre állnak újabb függvényeket, változókat, szerkezeteket, amelyeket felhasználhatunk.

3. ANYAG ÉS MÓDSZERTAN

Mint bevezetőmben említettem, néhány eszköz már rendelkezésre állt, mielőtt a projektet megkaptam. Ezen eszközök közé tartozott a fegyvertorony egyik fő építőeleme, maga a fegyver, illetve a mozgásért felelő motorok. A fegyver kiválasztásánál néhány fontosabb jellemzőt vettem figyelembe a döntéshozók.

1. az elsütés módja: elektronikus elsütő szerkezetet kellett választani, a vezérléssel való kommunikáció könnyítése végett
2. a tárcapacitás és az újra töltés módja: nagyobb tárcapacitás és gyors újra töltés kívánatos, a bemutatókon való várakozási idő csökkentése érdekében
3. kompakt kivitel, tartós felépítés: kisebb kiterjedésű fegyvert könnyebb kezelni szállítás és használat közben; az alkatrészek tartóssága fontos tényező a karbantartás, szervizelés tekintetében

3.1 Airsoft P90, CM.060



14. ábra - Airsoft P90, CM.060 (AIRSOFTPRO, 2022)

Alapvető információk

Lövés mód

Egyes és sorozatlövés

A funkció elve

Elektromos

A fegyver teljesítménye (+/- 10%)

A lövedék energiája

1,4 J

Kezdő sebesség

120 m/s (394 FPS)

Elméleti lőtávolság

54 m

A golyók megfelelő súlya

0,25g

Külső leírás

Hossz

510 mm

Tömeg

~2080 g

Test

Műanyag, fém alkatrész

Puskaagyak

Műanyag, rögzített

Belső leírás

Belső hordó	~265 mm x 6,08 mm
Hop-up	Műanyag, állítható
Gearbox	V6, Fém öntvény
Csapág	7 mm fém
Fogas kerék	Fémötvözet, 18:1
Dugattyú	Nejlón 1 fém foggal
Motor	Hosszú tengelyes standard motorok

Tölténytár

A tölténytár kapacitása	70
A tölténytár típusa	Rugóval tölt

Akkumulátor

11,1 V Li-Po ready	Nem
Bemeneti akkumulátor	Mini Block
Csatlakozó	Mini Tamiya
Az akkumulátor elhelyezése a fegyverben	A markolatban

3.2 KONECT 31 kg CORELESS HiVOLT WATERPROOF Digital servo (31kg-0.09s / 60 °)



15. ábra - KONECT 31 kg CORELESS HiVOLT WATERPROOF Digital servo (saját kép)

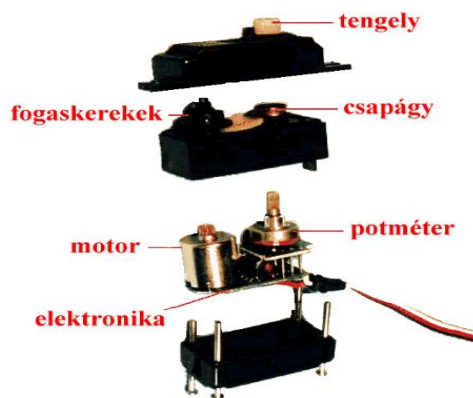
A feladat elvégzéséhez szervomotorokat használtam. Azonban elmondható, hogy a robotoknál felhasználható motorokra általánosan jellemző, hogy az adóból (lehet távirányító is) érkező, a vevő által (lehet a központi vezérlő) felfogott elektromos jelet alakítják át fizikai mozgássá. Legyen az servo-, DC vagy léptető motor.

Szervomotor:

A szervomotorok a beérkező jelet úgy alakítják fizikai mozgássá, távirányítás esetén, hogy az arányos legyen az adón végzett fizikai mozgással. Vagyis pontosan annyit mozduljon és abba az irányba a motor, amennyit és amilyen irányba a távirányító karja mozdul. Felépítése nem túl összetett. Működése a visszacsatolás elvén alapul.

Felépítése, alkatrészek:

- vezérlő elektronika (komparátor áramkör)
- egyenáramú motor
- fogaskerekek
- visszacsatolást végző alkatrész (potméter, optika, mágneses kódtárcsa)
- tengely



16. ábra - Szervomotor robbantott ábra
(HARSÁNYI & JUHÁSZ, 2014)

Fontos még megjegyezni, hogy a potméter közvetlen csatlakozik a tengelyhez, azonban nem képes teljes, 360°-os fordulatra. Ezért csak 0° és 180° között adhatunk meg elfordulási szöget. (HARSÁNYI & JUHÁSZ, 2014) (MODELL & HOBBY, 2018)

„Hobby” célú felhasználás esetén három méretet különböztethetünk meg.

- micro
- standard
- large scale (giant)

Műszaki paraméterek:

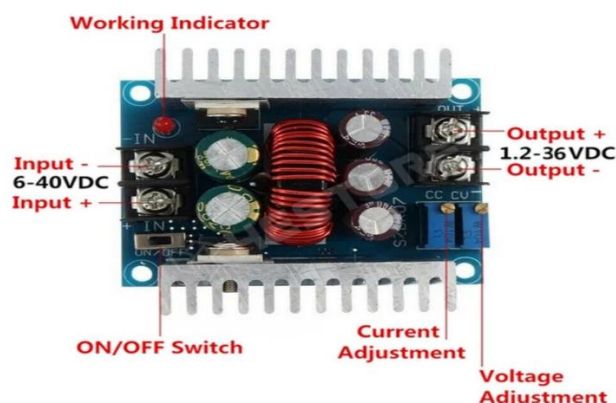
Típus	digitális
Motor típusa	CORELESS
Áttétel (fogaskerekek) anyaga	Steel & brass (acél és sárgaréz)
Vezérlés	impulzusszélesség szabályozás
Működési frekvencia	1520µs / 330hz
Üzemi feszültség	DC 6.0 - 8.4V
Üzemi hőmérséklet	-10 to +50 °C
Üzemi sebesség (6.0V) terhelés nélkül	0.11s / 60°
Üzemi sebesség (8.4V) terhelés nélkül	0.09s / 60°
Tolóerő / nyomaték (6.0V)	24.07kg.cm
Tolóerő / nyomaték (8.4V)	30.3kg.cm
Potentiometer	közvetlen vezérlés
Méret	40.0 x 20.2 x 40.2mm
Tömeg	58.0g

2. táblázat - KONECT 31 kg CORELESS HiVOLT WATERPROOF Digital servo datasheet

3.3 DC-DC konverter modul (step-down)

A szervomotorok működtetéséhez szükségem volt egy DC-DC konverterre. A konverter olyan elektronikus áramkör, amely egy bemeneti DC (direct current, egyenáram) feszültségből alakít ki egy másik kimeneti DC feszültséget. Egyik legfontosabb jellemzői a konvertereknek, hogy milyen irányban végzi az átalakítást. Ezek alapján megkülönböztetünk:

- Step-down (buck); A kimeneti feszültség alacsonyabb, mint a bemeneti feszültség, és azonos polaritású.
- Step-up (boost); A kimeneti feszültség nagyobb, mint a bemeneti feszültség, és azonos polaritású.
- SEPIC (single-ended primary-inductor converter); A kimeneti feszültség lehet alacsonyabb vagy magasabb, mint a bemeneti feszültség.



17. ábra - DC-DC konverter modul, step-down, 300W, 6-40V - 1.2-36V, 20A

Műszaki paraméterek:

Teljesítmény	300W
Kimeneti feszültség	1.2...36V DC
Bemeneti feszültség	6...40V DC
Kimeneti áram [max.]	20A
Kimeneti csatlakozó	csavaros szorítók
Üzemi hőmérséklet	-40...85°C
Kimeneti áram	15A
Elektromos csatlakozás	csavaros szorítók
Kimenetek száma	1
Hatásfok	95%
Tápegység fajtája	szabálvozható

táblázat - - DC-DC konverter modul, step-down datasheet

3.4 Relé modul

A fegyver elsütését egy relé modullal oldottam meg. Azért volt erre szükség, habár a vezérlő is képes lett volna közvetlenül kezelni, azonban ezzel a módszerrel ki tudtam simítani a feszültség hullámzásokat. A relé gyakorlatilag egy elektromosan működtetett „kapcsoló”, ami többfajta méretben, formában és teljesítményben megvásárolható. Működése arra irányul, hogy a tápfeszültséget vezérelve tudjuk kapcsolni. Az érintkezők típusa lehet:

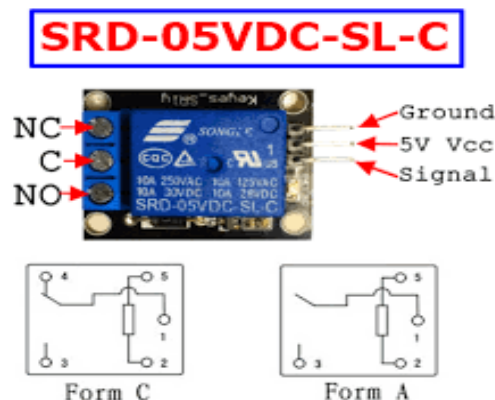
- NO (Normally Open); alaphelyzetben nyitott
- NC (Normally Closed); alaphelyzetben zárt

illetve,

- SPST - Egypólusú
- SPDT - Egypólusú váltó
- DPST – Kétpólusú
- DPDT – Kétpólusú váltó

ahol, SP = Single Pole, DP = Double Pole, ST = Single Throw, DT = Double Throw

Projektem során egy SPDT típusú relét használtam.



18. ábra - 1 relés modul (DATASHEETCAFE, 2021)

Maximális áramerősség	10A
Vezérlő feszültség	5V DC
Érintkező típusa	SPDT
Maximális kapcsolási feszültség	250V AC 30V DC

4. táblázat - 1 relés modul datasheet

4. A PROJEKT FOLYAMATAI

A dolgozatom témájául szolgáló fejlesztés a kezdetétől fogva projektként kellett kezelnem. Mindezt azért, mert az operatív feladataim mellett voltam kénytelen véghez vinni. Beosztásom szerint a „Logisztikai és anyaggazdálkodó csoport” csoportvezetője vagyok, ezért ezen feladatom mellett kellett időt szakítanom a fejlesztésre. Napi szinten egy-két órát, heti szinten néhány napot tudtam foglalkozni a projekttel. Ez valamelyest rá is nyomta a bélyegét a tervezési folyamatokra.

4.1 Tervezés

A klasszikus értelemben vett tervezésre, a rendelkezésemre álló idő rövideége és a ráfordítható idő csekély mennyisége miatt, nem volt lehetőségem. A szerkezet kialakításának és a rendszerfejlesztés lépéseit sem papír, sem elektronikus formában nem sikerült megvalósítanom. A projekt kezdetétől a befejezéséig lépésről lépésre haladtam.

4.1.1 A fegyvertorony kialakítása

Kiindulásnak adott volt a fegyver, amit be kellett építenem, valamint a forma is hamar körvonalazódott a gondolataimban, a katonai szervezeteknél rendszeresített eszközök alapján.



19. ábra - RCWS 2006-ból (DEFENSE, 2006)

Az fegyverrendszerekről készült képekről pontosan kiderült, hogy milyen egységeket kell kidolgoznom. Készítenem kellett,

1. egy alapzatot, ami stabilan tarja a tornyot és elfér benne a szervomotor
2. egy tornyot, ami az X tengelyen való elfordulásért felelős
3. egy olyan tartószerkezet, ami a csatlakozik a toronyhoz olyan módon, hogy az Y tengelyen való mozgást biztosítsa

Törekedtem a gyors megvalósításra, mert tudtam, hogy több időt kell szentelnem a vezérlés elkészítésére. Szerencsére az életem során szerzett tapasztalatok a gépjármű javítás (forgó-, billenő alkatrészek kezelése; elektromosság /DC/), házfelújítás (barkácsolás; erős- és gyengeáram), fémmegmunkálás területein óriási haszonnal voltak. Ezen tapasztalatok felhasználásával egy hét leforgása alatt elkészültem a szerkezet prototípusával. A későbbiekben már csak kisebb változtatásokra került sor.

4.1.2 Vezérlés tervezése

A projekt megkezdése előtt néhány évvel, már foglalkoztam szabadidőmben automatizálással, robotszerű eszközök készítésével. Ebből adódóan a fejlesztői környezetet ismertem, de sajnos megfakultak az emlékek. Első lépésként a szoftvert kellett megismernem és az általa nyújtott lehetőségeket. Az internet világában ez a feladat ijesztően könnyűnek bizonyult. Természetesen hatalmas segítségemre voltak az egyetemen szerzett programozói ismeretek, hogy meg is értem a kapott információkat. Az alapok újbóli elsajátítása után a motorok és az elsütő szerkezet (egy LED diódával helyettesítettem) programozását valósítottam meg. Az egységeket egy próbapanelen kötöttem össze a mikrovezérlővel a gyorsabb konfigurálhatóság megőrzése végett. Ezek után következhetett a kontroller integrálása a kódba. Több hetet vett igénybe a megfelelő könyvtár megtalálása, a könyvtár nyújtotta lehetőségek megismerése és helyes használata. Az idáig vezető úton rengeteg blog-ot, cikket, hozzászólást, GitHub projektet kellett átböngésznem.

A folyamat végén egy, az elvárásokat teljesítő, működő rendszer lett az eredmény, apró szépséghibával. A kontroller és a mikrovezérlő kapcsolódása, nem minden esetben jött létre. Ezért be kellett szerveznem egy olyan egységet, ami az éppen lejátszódó folyamatokról visszajelzést tud küldeni. Elégséges lehetett volna egyszerű LED-ek alkalmazása, de a problémák sokrétősége miatt ez kevésnek bizonyult volna. Ezért ennek megoldására egy LCD kijelzőt építettem a rendszerbe.

Kutatásom során több lehetőséget is megismertem, melyeket a leendő fejlesztéseim esetén alkalmazhatok. Ilyen lehetőséget kínál például az ingyenesen használható TinkerCad, GitHub webhelyek.

4.2 A vezérlőkód bemutatása

Dolgozatomban korábban már szót ejtettem a programkód két fontos részéről. A setup és a loop függvényekről. Ezek a kódrészek alapjaiban határozzák meg a kész eszköz működését. Ezek bemutatása után áttérek a kommunikációért felelős részhez, majd végezetül a motorok és az elsütő szerkezet vezérlését mutatom be.

A felhasznált, nem beépített könyvtárak:

```
#include <Arduino.h>
#include <ESP32Servo.h>

//***** LCD

#include <TFT_eSPI.h> // Graphics and font library for ILI9341 driver chip
#include <SPI.h>

//***** BLE és XBOX

#include <NimBLEDevice.h>
#include <XboxControllerNotificationParser.h>

//***** LCD

#define TFT_GREY 0x5AEB // New colour
TFT_eSPI tft = TFT_eSPI(); // Invoke library
```

4.2.1 Setup és loop függvények

A setup függvény alatt a kezdeti beállításokat végzem el:

```
void setup() {
    servoX.attach(14);
    servoY.attach(13);
    servoX.write(posX);
    servoY.write(posY);
    Serial.begin(115200);
    Serial.println("Starting NimBLE Client");

    setupLCD();
    tft.fillScreen(TFT_BLACK);
    tft.setCursor(5, 5, 2);
    tft.print("Start");
    myDelay(1000);
}
```

Két darab szervomotor csatlakoztatása a megadott lábhoz,
illetve a megadott kezdeti helyzet beállítása

A soros porton való kommunikáció sebességének beállítása

Szöveg kiírása / ellenőrzési funkció /

Az LCD kijelző alaphelyzetbe állítása

A loop függvény, a mikrovezérlő működése alatt, végtelen ciklusként folyamatosan lefut. Ebbe a kódrészbe kerültek a kontroller és a mikrovezérlő kommunikációjáért felelős függvények (ellenőrzés, csatlakozás), valamint az állapotok kiírása az LCD kijelzőre:

```
void loop() {
    if (connectedCall == true){
        connectedCall = false;
        Serial.println("Connected");
        tft.fillScreen(TFT_BLACK);
        tft.setCursor(5, 5, 2);
        tft.print("Connected");}
    if (disconnected == true){
        disconnected = false;
        Serial.println(" Disconnected");
        tft.fillScreen(TFT_BLACK);
        tft.setCursor(5, 5, 2);
        tft.print("Disconnected");
        myDelay(1001);}
    if (!connected) {
        if (advDevice != nullptr) {
            myDelay(500);
            if (connectToServer(advDevice)) {
                Serial.println("Success! we should now be getting notifications");
            } else {Serial.println("Failed to connect");}
            advDevice = nullptr;
        } else if (!scanning) {startScan();}
    }
    myDelay(2);
}
```

A myDelay() függvény deklarációjára azért volt szükségem, mert a beépített Delay() függvény megállítja (Pause) a futó programot. Ezek a megállások összességében befolyásolták az elsütési folyamatot. A Pause effektust egy While ciklussal oldottam meg, így a program nem akad meg, csak az adott „szálon” várakozik.

```
void myDelay(unsigned long waitTime){
    unsigned long startTime = millis();
    unsigned long actualTime = startTime;
    while ((actualTime - startTime) < waitTime)
    {actualTime = millis();}
}
```

4.2.2 Kommunikáció, irányítás

A kommunikációért felelő programkódot, a „printOnSerialWithNimBLE” Example segítségével valósítottam meg. A példából emeltem ki a szükséges elemeket. A kívánatos működéshez tanulmányoznom és értelmezni kellett a teljes működési folyamatokat. A kódból csak a fontosabb részeket mutatom be.

Új eszköz csatlakoztatása:

```
bool connectToServer(NimBLEAdvertisedDevice* advDevice) {
    NimBLEClient* pClient = nullptr;
    tft.fillScreen(TFT_BLACK);
    tft.setCursor(5, 5, 2);
    tft.print("Connecting");
    if (!pClient) {
        pClient = NimBLEDevice::createClient();
        pClient->setClientCallbacks(&clientCB, false);
        pClient->setConnectTimeout(10);
        pClient->connect(advDevice, false);}
    return true;
}
```

Kapcsolódás után, szolgáltatások lekérdezése

```
bool afterConnect(NimBLEClient* pClient) {
    tft.setCursor(5, 35, 2);
    tft.print("Service search");
    for (auto pService : *pClient->getServices(true)) {
        auto sUuid = pService->getUUID();
        if (!sUuid.equals(uuidServiceHid)) {
            continue;} // skip
        for (auto pChara : *pService->getCharacteristics(true)) {
            charaRead(pChara);
            charaSubscribeNotification(pChara);}
    }
    tft.setCursor(5, 65, 2);
    tft.print("Service OK");
    return true;
}
```

Kommunikáció és irányítás:

```
void notifyCB(NimBLERemoteCharacteristic* pRemoteCharacteristic, uint8_t* pData, size_t length, bool isNotify) {
```

```
    xboxNotif.update(pData, length);
```

Ez a sor felel a kapcsolat alatti adacseréért

```
    static unsigned long firedAt = 0;
```

```
    lX = xboxNotif.joyLHori;
```

```
    rY = xboxNotif.joyRVert;
```

```
    trig = xboxNotif.trigRT;
```

A kontrolleren lévő joystick karok mozgását és az elsütéshez használt gomb megnyomását rendelem változókba

```
    if (lX >= 0 && lX <= 30000 && posX > RotateMin) {
```

```
        posX--;
```

```
        servoX.write(posX);
```

```
        myDelay(100);
```

```
    }
```

```
    if (lX >= 35000 && lX <= 65535 && posX < RotateMax) {
```

```
        posX += 1;
```

```
        servoX.write(posX);
```

```
        myDelay(100);
```

```
    }
```

```
    if (rY >= 0 && rY <= 30000 && posY > RotateMin) {
```

```
        posY --;
```

```
        servoY.write(posY);
```

```
        myDelay(100);
```

```
    }
```

```
    if (rY >= 35000 && rY <= 65535 && posY < RotateMax) {
```

```
        posY++;
```

```
        servoY.write(posY);
```

```
        myDelay(100);
```

```
    }
```

```
    if (trig == 1023) {
```

```
        if ( millis() - firedAt > fireinterval) {
```

```
            digitalWrite(21, HIGH);
```

```
            myDelay(130);
```

```
            digitalWrite(21, LOW);
```

```
            firedAt = millis(); }
```

```
    }
```

```
}
```

Ebben a kódrészletben a horizontális, X tengelyen való elfordulást valósítom meg. Az elágazások feltételrendszerében a joystick elmozdulását vizsgálom, teljesülés esetén a torony mozgatásáért felelő szervomotort fordítom a megadott irányba

Ebben a kódrészletben a vertikális, Y tengelyen való elfordulást valósítom meg. Az elágazások feltételrendszerében a joystick elmozdulását vizsgálom, teljesülés esetén a fegyvertartó sín mozgatásáért felelő szervomotort billentem a megadott irányba

A fegyver elsütését végrehajtó kódrész. Az elsütőszerkezet sajátos viselkedése miatt kellett relét alkalmaznom. A digitalWrite() függvénnyel a relét vezérlem.

A feltételekben szereplő értékeket mintavételezés útján határoztam meg. Úgy állítottam be ezeket, hogy az apró, véletlen mozgásokat ki tudjam szűrni.

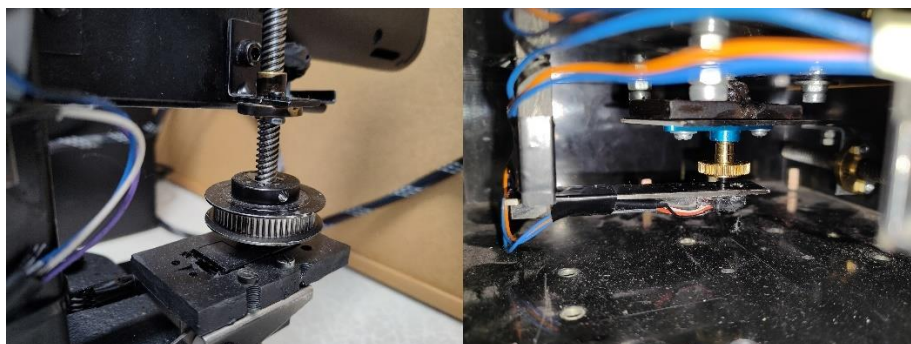
4.3 Összeszerelés, tesztelés

Az összeszerelést akkor kezdtem el, amikor úgy gondoltam, hogy a vezérlőprogram megfelelően működik. A próbapanelen össze voltak kötve a szervomotorok a mikrokontrollerrel és az XBOX kontrollerrel rendben tudtam mozgatni őket, valamint működött az elsütő szerkezet (LED) is. Első lépésként a tápellátást kellett biztosítanom az eszközökhöz. Ezt egy számítógépházzal és a hozzá tartozó tápegységgel oldottam meg. Végül kettő darab tápegység került beszerelésre, mert a fegyver elsütése nagy áramfelvétellel jár, az eszköz többi részét pedig biztonságosan el szerettem volna szeparálni a felmerülő áramingadozástól.



20. ábra - Tápellátás és csatlakozók (saját kép)

Következő lépésként a szervomotorokat illesztettem a helyükre, a tápellátást még a fejlesztői kártya biztosította ekkor feljűk. Az első használat alkalmával meg is mutatkozott a mélyebb tervezés hiánya. A motorok nem bírták el a mozgatáskor rájuk ható erőket (egy motor menthetetlen lett). „Egy lépés előre, két lépés hátra”, gondoltam magamban. A mentormommal közösen, illetve egy-egy érdeklődő kolléga kíséretében, megoldást kellett találni a problémára.



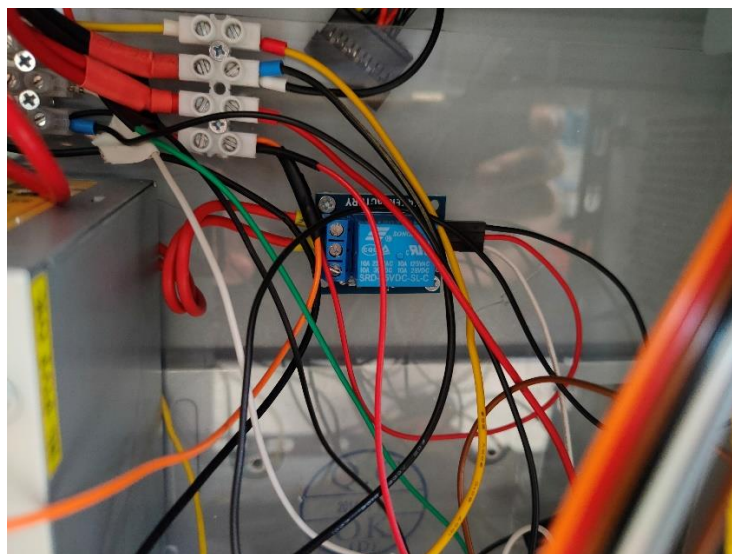
21. ábra - Orsós emelő használata (saját kép)

A következő problémát a szervomotor sajátossága okozta. Mint dolgozatomban korábban írtam, a szervomotorok folyamatos forgásra nem képesek a potméter elhelyezkedése miatt. Ezért a potmétert kiszereltük a motorokból és közvetlenül a fordulási tengelyekhez rögzítettük őket, a vezetékezést megtartva a motor elektronikájával. A motorokat pedig az orsókhöz rögzítettük, már szabad, folyamatos forgásra készen.



22. ábra - A potméter rögzítési megoldása (saját kép)

Ebben a fázisban a fegyver mozgatása megoldódott. Következő lépés volt az elsütés implementálása. Eddig még csak a LED-et vezéltem. Itt szembesültem a következő problémával. Az vezérlő kártyára közvetlenül csatlakoztatva a fegyver elektronikája nem működött megfelelően. A miértjére sajnos nem tudtunk rájönni, ezért került beépítésre a relé.



23. ábra - Elsütésért felelő relé (saját kép)

A relé közbeiktatásával megoldódott az elsütésnél tapasztalható probléma.

Az összeszerelés közben a tesztelés folyamatosan zajlott az újabb részegységek beiktatásával összhangban, valamint a programkódot is ezzel párhuzamosan alakítani kellett a változtatásokhoz. Ezekkel a felmerülő problémákkal együtt is határidőre sikerült a projektet befejezni, az elkészült eszközzel a feletteseim meg voltak elégedve.

Az első bemutatót követően a visszajelzések pozitívak voltak a felhasználók részéről. Feletteseim fel is vetették, hogy kezdjek el gondolkozni esetleges fejlesztés terveken.



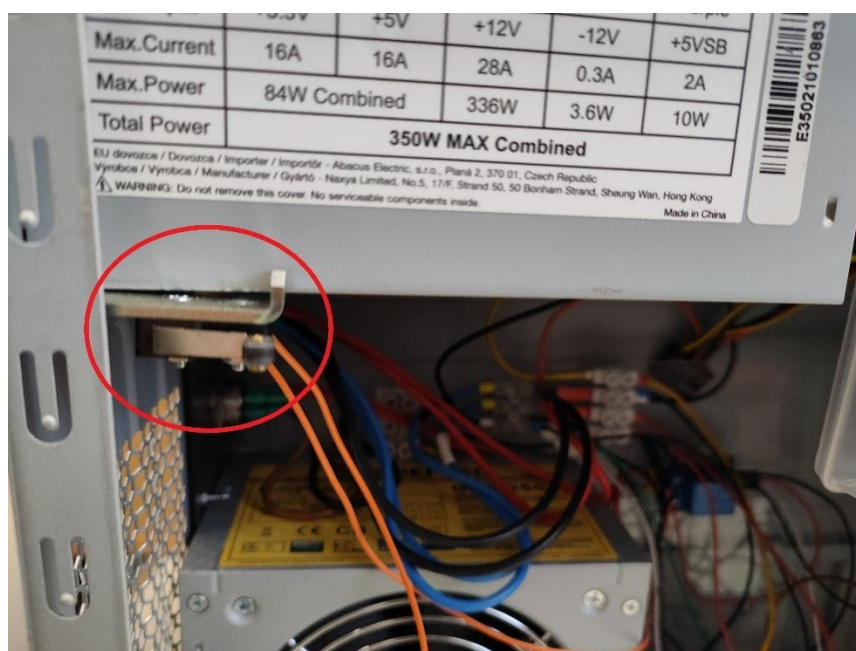
24. ábra - Az elkészült szerkezet (saját kép)

5. KÖVETKEZTETÉSEK ÉS JAVASLATOK

Az első bemutató után levontuk a megfelelő konzekvenciákat. A fegyvertorony működése megfelelő, azonban további fejlesztéseket muszáj lesz eszközölnünk.

5.1 Biztonsági fejlesztések

Jelenleg csak szabotázs elleni védelem van beépítve a számítógépházba. Ezzel a védelemmel a vezérlést szakítjuk meg.



25. ábra - Szabotázs elleni védelem (saját kép)

A fegyver használata közben azonban túl nagy a balesetveszély a nemtörődöm vagy figyelmetlen szemlélődők irányába. Ezen védelem kialakítására több megoldás is létezhet, talán a legegyszerűbb kivitelezés szempontjából, és nem is költséges, valamilyen mozgásérzékelő beiktatása a rendszerbe. A jól elhelyezett és beállított mozgásérzékelővel megszakítható az elsütési mechanizmus.



26. ábra - Mozgásérzékelő modul (CONRAD, 2022)

Ezek a PIR szenzorok¹¹ 120°-os látószöggel és késleltetéssel rendelkeznek, így kettő darab rendszerbe integrálásával kiküszöbölhetőek a véletlen balesetek. A szenzorok jelet tudnak továbbítani a vezérlő egységnek, hogy mozgást érzékeltek, ezzel a fegyvert elsütését annyi ideig meg lehet gátolni, hogy helyszínen lévő kezelő személyzet a rendszert leállíthassa, amennyiben szükséges, vagy engedélyezze a további használatot.

A legbiztosabb megoldás egy kamera által rögzített felvétel realtime¹² elemzése. Az elemzés eredményeként a használat korlátozása. Ez természetesen nagyobb anyagi ráfordítást és szakmai hozzáértést követel meg.

5.2 Mobilitás, hordozhatóság

Jelenlegi állapotában az eszköz mozgatása több emberi erőt vagy hosszabb időtartamot követel. Ezt nem a súlya, hanem a fogáspontok kialakítása határozza meg. A vezérlést tartalmazó házra egyszerű fogantyú felszerelése megoldást kínálhat. A fegyver köré tervezett szállító doboz fogantyúval, szintén működőképes lehet. A professzionális, ezáltal költségesebb lehetőség, egy katonai szállítódoboz beszerzése és átalakítása a kívánalmak szerint.



27. ábra - Katonai szállítódoboz (MITCHELL'S, 2022)

¹¹ PIR szenzor - Passive Infrared Sensor

¹² realtime- valós idejű

5.3 Energetikai ellátásbiztonság

Az eszköz jelenlegi állapotában csak 230V-os hálózati energiaellátás esetén működőképes. Ezt nem minden esetben és körülmények között lehet biztosítani számára. Előfordulhat, hogy a kitelepülés helyszínén az áramellátás csak korlátozott mennyiségben elérhető. Ezért a rendszert fel lehet (kell) készíteni akkumulátoros üzemmódra. Legegyszerűbb megoldás erre a problémára egy szünetmentes tápegység. Ezzel néhány órára megoldható a tápellátás.

Esetlegesen az AC-DC tápegységek kihagyásával, de a lehetőség meghagyásával, szimplán akkumulátorok használatára lehet hagyatkozni, ezzel az üzemidőt ki lehet tolni.

5.4 „SCI-FI” megoldások

A következő lehetőség amire gondoltam, az a fegyver 3D-s szemüveggel való irányítása (munkahelyemen folynak ilyen témában fejlesztések). Ekkor a szemüveg viselőjének fejmozgását követné le a fegyver mozgása. A fegyver célzását egy kamera képe biztosítaná, amely képet a 3D szemüvegre lehetne továbbítani. Az ott látott kép alapján a felhasználó, a feje mozgásával vehetné célba az eltalálni kívánt objektumot.

A kollégák részéről érkeztek még érdekes ötletek (ekkor a jelenleg is zajló háború még csak kezdeti szakaszában volt):

- drónra szerelt fegyver
- drónra szerelt fegyver, kombinálva 3D-s szemüveggel
- mozgó platformra szerelt lövegtorony

De ezeket az ötleteket azonnal elvettem, mert az projekt kezdetekor kikötöttük a helyhez kötöttséget!

6. ÖSSZEFOGLALÁS

Dolgozatom témájának kiválasztása személyes indíttatásból és munkahelyi feladatszabásból eredt. Az, hogy a projektet megvalósíthattam és az a tudat, hogy a dolgozatomban erről beszámolhatok, nagyon felemelő érzés volt. A projekt elkészítése közben számos olyan tudásra tettem szert, amit a későbbiekben is nyugodt szívvel fel tudok használni, akár hasonló projekt kivitelezésénél, akár az otthonom automatizálásának esetén. A dolgozatom írásakor is számos új és fontos információ birtokába jutottam. Találkoztam más megközelítésekkel, amiket felhasználva, talán még jobb munkát tudtam volna letenni a feletteseim elé. Gondolok itt elsősorban a különböző tervező szoftverekre.

Első lépéskén a feladat megismeréséhez és kivitelezéséhez szükséges forrásokat néztem át. Egy rövid történelmi áttekintést tettem, körbejárva dolgozatom elsődleges célját, a robotok alkalmazhatóságát, eredetét. Felelevenítettem a robotika három alaptörvényét, a robot, mint szabvány meghatározását, valamint a robotok csoportosítási lehetőségeit. Ezt követően röviden tanulmányoztam, hogy mik is azok az egykártyás számítógépek és a mikrovezérlő kártyák, elsősorban az általam is használt kártya tulajdonságait vizsgáltam. A már meglévő alkatrészek tulajdonságait vizsgálva, értem ez alatt a legfontosabb műszaki sajátosságokat, a tápellátások típusát, az áramfelvételek egyediségét, közös nevezőt kellett találnom arra, hogy ezeket egy eszközzé szerelhessem össze. Fel kellett mérnem, akkor még az adott alkatrészekkel kapcsolatban tapasztalatlanul, hogy szükséges-e további eszközök beszerzése vagy megoldható a feladat az adott anyagok felhasználásával. Ezen kérdéseknek a megválaszolása a projekt előrehaladtával sor került, elsősorban a folyamatos teszteléseknek köszönhetően. A fejlesztői környezet megismerése ugyan ilyen fontos volt a feladat megkezdésének pillanatában.

Az Arduino IDE fejlesztői környezet nem volt teljesen ismeretlen számomra, azonban fontosnak tartottam mélyebb megismerését, amit a vezérlőprogram készítése során meg is tettem. Kulcsfontosságúak azok a kezdeti lépések, melyeket az Arduino IDE első elindításakor, valamint az újabb fejlesztői kártyák használatakor meg kellett tennem. Be kellett állítanom a megfelelő kártyát, amely nem feltétlenül áll rendelkezésünkre alapfeltételként. Meg kellett ismernem a program felépítését, működését. Az Arduino IDE környezetben használható programozói nyelv, amely a C/C++, hasonló az általam is tanult C#-hoz. A vezérlőprogram írásakor előtérbe került számomra a GitHub rejtette azon lehetőségek, amikről eddig nem volt tudomásom, amelyek használhatósága kimagasló. Az Arduino IDE felépítéséből adódik a különböző Library-k használata, de meg kellett látnom

a mögöttük meghúzódó tudást, szakértelmet. Sok megoldás, függvény első ránézésre értelmetlennek tűnt, de használat közben kiderült a használhatóságuk. Minden ilyen esetben addig kutattam, míg közel teljes mértékben megértettem működését. A legtöbbször teljesen a „gyökerekig” le kellett ásnom, mert ott rejtett a megoldás.

Összességében az eszköz megtervezésénél a programozása számomra nagyobb figyelmet igényelt. Az előfeltételek adottak voltak az egyetemi tanulmányaim által, de a terület egyedisége mégis elbizonytalanított a projekt elején. Meg kellett ismernem több speciális függvényt, hogy alkalmazni tudjam őket a programkódomban. A szervomotorok, relék, LCD kijelző, mind más-más megközelítést igényeltek. A megfelelő alkalmazás kutatása közben találkoztam egyéb, a későbbi fejlesztések során felhasználható alkatrészekkel és ezek működésével. Gondolok itt különféle szenzorokra, a felhasználttól eltérően működő motorra vagy vezérlőegységre.

A vezérlőprogram írására, az eszközt mozgató mechanizmus közel kész állapota után került sor. A rendelkezésre álló szervomotorok vezérlésével kezdtem. Több fórumon is kutakodtam, hogy a helyes használatot elsajátítsam. Amikor ezt az állapotot elértem, a motorokat beépítettem az eddig elkészült szerkezetbe. A fegyver elsütését, tápellátását gondoltam következő lépésként megoldani. Ehhez a fejlesztő kollégáimtól egy labortápot kértem kölcsön, hogy a fegyver alapvető tápellátását, egy NiMH akkumulátort kiválthassak. Erre azért volt szükség, mert a bemutatókat hosszabb időtávra terveztük, ezért nem láttuk jó megoldásnak a kis teljesítményű akkumulátorok használatát. A labortáp használata hasznosnak bizonyult, mert kiderült, hogy a fegyver elsütés közben tüskét produkál áramfelvétel közben.

Ekkor már a fegyvertorony és a tápellátást biztosító számítógépház teljesen elkészült, beszerelésre kerültek a motorok, a mikrovezérlő, relé, DC-DC konverter. Utoljára hagytam az irányítás kezelését. Ez a szakasz bizonyult a végén a legnehezebbnek. Úgy gondoltam, hogy néhány kódot kell csak beillesztenem a már meglévő kódba. Utólag kiderült, hogy könnyelmű gondolat volt. Hosszas kutatói munkát kellett ráfordítani arra, hogy a bluetooth kommunikációs csatornát megfelelően tudjam használni a fejlesztői kártya és az Xbox kontroller között. Meg kellett oldani a csatlakozás során felmerülő anomáliákat, valamint meg kellett értenem a kontroller által küldött jelek feldolgozhatóságát. A csatlakozási „érdekességeket” megérteni nem sikerült, azonban kezelhető keretek közé szorítottam őket. A kontroller által küldött adatokat sikerült dekódolnom, ezek kezelését egy egyszerű feltételrendszeren belül oldottam meg.

A kitűzött céljaimat végezetül sikerült elérnem, habár sok nehézségbe ütköztem a feladat elvégése során. Úgy gondoltam, hogy a fizikai megvalósítás egyszerű lesz a korábban megszerzett tapasztalataim alapján, de utólag kiderült, hogy nem szabad semmit sem félvállról venni. Problémába ütköztem a szervomotorok elhelyezésénél, illetve a fegyver elsütésénél is. Szerencsére ezek megoldására nem kellett a rendelkezésre álló időből sokat áldozni, még ha elsőre hatalmas problémáknak látszódtak is.

A teljes összeillesztést a további projektjeimnél, már tervezett módon, lépésről lépésre, vagy alkatrésztől alkatrészig szükséges végrehajtanom. Ilyen módon elkerülhetővé válhatnak a nagyobb, nehezen orvosolható problémák. Ennél a projektnél végül szerencsésnek mondhatom magamat, mert könnyen sikerült orvosolni a felmerülő problémákat.

Végezetül megfogadom egy bölcs egyetemi oktatóm tanácsát, és folyamatosan alkalmazni fogom életemben a **kaizen** filozófiáját.

7. IRODALOMJEGYZÉK

AIRSOFTPRO, 2022. *AirsoftPro*. [Online]

Available at: <https://airsoftpro.hu/hu/airsoft-fegyverek/elektromos-hosszu-aeg/aug-p90/cm060-detail>

[Hozzáférés dátuma: szeptember 2022].

AI-THINKER Technology Co., L., 2022. *Ai-Thinker Technology*. [Online]

Available at: https://docs.ai-thinker.com/_media/esp32/docs/nodemcu-32s_product_specification.pdf

[Hozzáférés dátuma: 2. október 2022].

ARDUINO, 2022. *Arduino.cc*. [Online]

Available at: <https://www.arduino.cc/en/software>

[Hozzáférés dátuma: 11. május 2022].

BLUETOOTH, S., 2022. *Bluetooth®*. [Online]

Available at: https://www.bluetooth.com/wp-content/uploads/2022/10/assigned_numbers_release.pdf

[Hozzáférés dátuma: 6. október 2022].

CONRAD, E. M. K., 2022. *Conrad*. [Online]

Available at: https://www.conrad.hu/p/makerfactory-mf-6402129-pir-mozgaserzekelo-1-db-2134043?gclid=CjwKCAjwzNOaBhAcEiwAD7Tb6MrzWPe8uh4JwGClXg_J4eXp3YcqgxmQqQaV1uW-yRoCOvy4Y53rElxoCapkQAvD_BwE

[Hozzáférés dátuma: 10. október 2022].

DATASHEETCAFE, 2021. *DatasheetCafe*. [Online]

Available at: <http://www.datasheetcafe.com/srd-05vdc-sl-c-datasheet-pdf/>

[Hozzáférés dátuma: október 2022].

DEFENSE, U., 2006. *Defense Update*. [Online]

Available at: https://defense-update.com/20060726_eurosatory06-rcws.html

[Hozzáférés dátuma: 6. október 2022].

DR. NYAKÓNE DR. JUHÁSZ, K., Dr. Terdik, G., Biró, P. & Dr. Kátai, Z., 2011.

Bevezetés az informatikába. Debrecen: Debreceni Egyetem, Informatikai Kar.

DR. SZABÓ, Z., Budai, C., Dr. Kovács, L. & Dr. Lipovszki, G., 2014.

Robotmechanizmusok. Budapest: BME MOGI.

ESPRESSIF, S., 2022. *Espressif*. [Online]
 Available at: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
 [Hozzáférés dátuma: 2. október 2022].

GIOVINO, B., 2020. *Digi-Key electronics*. [Online]
 Available at: <https://www.digikey.be/nl/articles/using-microcontroller-development-boards-as-single-board-computers>
 [Hozzáférés dátuma: szeptember 2022].

HARSÁNYI, R. & JUHÁSZ, M. A., 2014. *Fizikai számítástechnika: elektronikai alapok és Arduino programozás*. első szerk. Budapest: Typotex Kiadó.

MENYHÁRT, J., 2017. *Felszíni villamos hajtású járművek és robotok (UGV) akkumulátorparaméter eltéréseinek vizsgálata Fuzzy logika és Support Vector Machine módszerekkel - Doktori (PhD) értekezés*, Budapest: Biztonságtudományi Doktori Iskola.

MITCHELL'S, 2022. *Mitchell's*. [Online]
 Available at: <https://www.mitchellsadventure.com/military/ammo-boxes/army-footlocker--plastic/ebxg019.aspx>
 [Hozzáférés dátuma: 10. október 2022].

MODELL & HOBBY, K., 2018. *BLOG.modell.hu*. [Online]
 Available at: http://blog.modell.hu/rc-szervok-es-azok-mukodese/?gclid=Cj0KCQjwhsmaBhCvARIsAlbEbH5_iOWP3w_UbMx7WE--mkNH2midSO1TK6YkRgwz9-IaqDRZTgJs7IgaAoSSEALw_wcB
 [Hozzáférés dátuma: 10. április 2022].

RASPBERRY, P., 2022. *Raspberry Pi*. [Online]
 Available at: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>
 [Hozzáférés dátuma: szeptember 2022].

SMITH, D., 2016. *Wires Are Obsolete*. [Online]
 Available at: <https://wiresareobsolete.com/2016/01/bluetooth-uuids-and-cross-platform-advertisements/>
 [Hozzáférés dátuma: október 2022].

STRUIJK, B., 2014. *Influence of the new trends in the economics on the military and industrial robot system design philosophy – Doctoral Dissertation*, Budapest: National University of Public Service, Faculty of Military Sciences and Officers' Training, PhD Institute in Military Technology.

SZABOLCSI, R. & MENYHÁRT, J., 2015. Loads Affecting UGVs' Technical Status. *Review of the Air Force Academy*, 3(30), pp. 15-20..

WIDSTRÖM, M. K., 2021. *ISO.org*. [Online]

Available at: <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-3:v1:en>

[Hozzáférés dátuma: szeptember 2022].

WIEDERHOLD, G., 2000. *Robots and their Arms; floor1: Mobile Robot developed at Stanford*. [Online]

Available at: <http://infolab.stanford.edu/pub/voy/museum/pictures/display/1-Robot.htm>

[Hozzáférés dátuma: 18 szeptember 2022].

WIKIPÉDIA, 2021. *Wikipédia - Karel Čapek*. [Online]

Available at: https://hu.wikipedia.org/wiki/Karel_%C4%8Capek

[Hozzáférés dátuma: 14 szeptember 2022].

WIKIPÉDIA, 2022. *Wikipédia - Szövegép*. [Online]

Available at: <https://hu.wikipedia.org/wiki/Sz%C3%B6v%C5%91g%C3%A9p>

[Hozzáférés dátuma: 14 szeptember 2022].

XBOX, 2022. *Microsoft Xbox*. [Online]

Available at: <https://www.xbox.com/hu-hu/accessories/controllers/xbox-wireless-controller#white>

[Hozzáférés dátuma: 28 szeptember 2022].

8. ÁBRAJEGYZÉK

1. ábra - Szövegép a 19. században (WIKIPÉDIA, 2022)	7
2. ábra - Stanford Arm – 1969 (WIEDERHOLD, 2000).....	8
3. ábra - Raspberry Pi 4 (RASPBERRY, 2022)	10
4. ábra - NUCLEO-L4P5ZG board (GIOVINO, 2020).....	11
5. ábra - NodeMCU-32s esp32 (saját kép)	12
6. ábra - NodeMCU-32S lábkiosztás (AI-THINKER Technology Co., 2022)	13
7. ábra - NodeMCU-32s esp32 csatlakoztatás után (saját kép)	15
8. ábra - Xbox Series XS controller (XBOX, 2022).....	15
9. ábra- Arduino Web Editor (saját kép)	17
10. ábra - Arduino IDE /Preferences/ (saját kép)	17
11. ábra - A használni kívánt kártya kiválasztása (saját kép).....	18
12. ábra - Könyvtárak kezelése - Arduino IDE (saját kép)	19
13. ábra - Példakód kiválasztása (saját kép)	20
14. ábra - Airsoft P90, CM.060 (AIRSOFTPRO, 2022).....	21
15. ábra - KONECT 31 kg CORELESS HiVOLT WATERPROOF Digital servo (saját kép)	22
16. ábra - Szervomotor robbantott ábra (HARSÁNYI & JUHÁSZ, 2014).....	23
17. ábra - DC-DC konverter modul, step-down, 300W, 6-40V - 1.2-36V, 20A.....	24
18. ábra - 1 relés modul (DATASHEETCAFE, 2021).....	25
19. ábra - RCWS 2006-ból (DEFENSE, 2006).....	26
20. ábra - Tápellátás és csatlakozók (saját kép).....	32
21. ábra - Orsós emelő használata (saját kép)	32
22. ábra - A potméter rögzítési megoldása (saját kép)	33
23. ábra - Elsütésért felelő relé (saját kép)	33
24. ábra - Az elkészült szerkezet (saját kép)	34
25. ábra - Szabotázs elleni védelem (saját kép).....	35
26. ábra - Mozgásérzékelő modul (CONRAD, 2022).....	35
27. ábra - Katonai szállítódoboz (MITCHELL'S, 2022)	36

9. TÁBLÁZATOK

1. táblázat - Nodemcu-32s Datasheet (AI-THINKER Technology Co., 2022).....	14
2. táblázat - KONECT 31 kg CORELESS HiVOLT WATERPROOF Digital servo datasheet	23
3. táblázat - - DC-DC konverter modul, step-down datasheet.....	24
4. táblázat - 1 relés modul datasheet.....	25

NYILATKOZAT

a szakdolgozat nyilvános hozzáféréséről és eredetiségéről

A hallgató neve: Pap Zsolt
A Hallgató Neptun kódja: NBD4E4
A dolgozat címe: Távirányítással vezérelhető fegyverplatform tervezése és kivitelezése
A megjelenés éve: 2022
A tanszék neve: Alkalmazott Informatikai Tanszék

Kijelentem, hogy az általam benyújtott szakdolgozat egyéni, eredeti jellegű, saját szellemi alkotásom. Azon részeket, melyeket más szerzők munkájából vettem át, egyértelműen megjelöltem, s az irodalomjegyzékben szerepeltettem.

Ha a fenti nyilatkozattal valótlan állítottam, tudomásul veszem, hogy a Záróvizsga-bizottság a záróvizsgából kizár és a záróvizsgát csak új dolgozat készítése után tehetek.

A leadott dolgozat, mely PDF dokumentum, szerkesztését nem, megtekintését és nyomtatását engedélyezem.

Tudomásul veszem, hogy az általam készített dolgozatra, mint szellemi alkotás felhasználására, hasznosítására a Magyar Agrár- és Élettudományi Egyetem mindenkori szellemi tulajdonkezelési szabályzatában megfogalmazottak érvényesek.

Tudomásul veszem, hogy dolgozatom elektronikus változata feltöltésre kerül a Magyar Agrár- és Élettudományi Egyetem könyvtári repozitori rendszerébe.

Kelt: 2022 év október hó 27 nap


Hallgató aláírása

KONZULTÁCIÓS NYILATKOZAT

A **Pap Zsolt** (név) (hallgató Neptun azonosítója: **NBD4E4**) konzulenseként nyilatkozom arról, hogy a szakdolgozatot¹ áttekintettem, a hallgatót az irodalmi források korrekt kezelésének követelményeiről, jogi és etikai szabályairól tájékoztattam.

A szakdolgozatot a záróvizsgán történő védelemre javaslom / nem javaslom².

A dolgozat állam- vagy szolgálati titkot tartalmaz: igen nem*³

Kelt: **2022. év október hó 26. nap**


Belső konzulens

¹ A megfelelő dolgozattípus meghagyása mellett a többi típus törlendő.

² A megfelelő aláhúzendő.

³ A megfelelő aláhúzendő.

10. MELLÉKLETEK

1. számú melléklet – Arduino Language Reference

Language Reference

Arduino programming language can be divided in three main parts: functions, values (variables and constants), and structure.

Functions

For controlling the Arduino board and performing computations.

Digital I/O

`digitalRead()`
`digitalWrite()`
`pinMode()`

Analog I/O

`analogRead()`
`analogReference()`
`analogWrite()`

Zero, Due & MKR Family

`analogReadResolution()`
`analogWriteResolution()`

Advanced I/O

`noTone()`
`pulseIn()`
`pulseInLong()`
`shiftIn()`
`shiftOut()`
`tone()`

Time

`delay()`
`delayMicroseconds()`
`micros()`
`millis()`

Math

`abs()`
`constrain()`
`map()`
`max()`
`min()`
`pow()`
`sq()`
`sqrt()`

Trigonometry

`cos()`
`sin()`
`tan()`

Characters

`isAlpha()`
`isAlphaNumeric()`
`isAscii()`
`isControl()`
`isDigit()`
`isGraph()`
`isHexadecimalDigit()`
`isLowerCase()`
`isPrintable()`
`isPunct()`
`isSpace()`
`isUpperCase()`
`isWhitespace()`

Random Numbers

`random()`
`randomSeed()`

Bits and Bytes

`bit()`
`bitClear()`
`bitRead()`
`bitSet()`
`bitWrite()`
`highByte()`
`lowByte()`

External Interrupts

`attachInterrupt()`
`detachInterrupt()`

Interrupts

`interrupts()`
`noInterrupts()`

Communication

`Serial`
`SPI`
`Stream`
`Wire`

USB

`Keyboard`
`Mouse`

Variables

Arduino data types and constants.

Constants

HIGH | LOW
INPUT | OUTPUT | INPUT_PULLUP
LED_BUILTIN
true | false
Floating Point Constants
Integer Constants

Conversion

(unsigned int)
(unsigned long)
byte()
char()
float()
int()
long()
word()

Data Types

array
bool
boolean
byte
char
double
float
int
long
short
size_t
string
String()
unsigned char
unsigned int
unsigned long
void
word

Variable Scope & Qualifiers

const
scope
static
volatile

Utilities

PROGMEM
sizeof()

Structure

The elements of Arduino (C++) code.

Sketch

loop()
setup()

Control Structure

break
continue
do...while
else
for
goto
if
return
switch...case
while

Further Syntax

#define (define)
#include (include)
/* */ (block comment)
// (single line comment)
; (semicolon)
{ } (curly braces)

Arithmetic Operators

% (remainder)
* (multiplication)
+ (addition)
- (subtraction)
/ (division)
= (assignment operator)

Comparison Operators

!= (not equal to)
< (less than)
<= (less than or equal to)
== (equal to)
> (greater than)
>= (greater than or equal to)

Boolean Operators

! (logical not)
&& (logical and)
|| (logical or)

Pointer Access Operators

& (reference operator)
* (dereference operator)

Bitwise Operators

& (bitwise and)
<< (bitshift left)
>> (bitshift right)
^ (bitwise xor)
| (bitwise or)
~ (bitwise not)

Compound Operators

%= (compound remainder)
&= (compound bitwise and)
*= (compound multiplication)
++ (increment)
+= (compound addition)
-- (decrement)
-= (compound subtraction)
/= (compound division)
^= (compound bitwise xor)
|= (compound bitwise or)

(ARDUINO, 2022)

11. KÖSZÖNETNYILVÁNÍTÁS

Mindenekelőtt nővéremet kívánom köszönettel illetni az élete során tanúsított kitartásáért, tanulni akarásáért, erejéért, mellyel örök életemre a példaképemmé vált. Ezek a tulajdonságai alapjaiban határozták meg gondolkozásomat a tanulás, munka, család iránt!

Természetesen szeretném megköszönni dr. Pántya Róbert tanárúrnak az évek során tanúsított elhivatottságát a Tanítás iránt. Az érdekfeszítő órákat, amikor a programozás rejtelmeibe, útvesztőibe kalauzolt minket. Bemutatta ennek a témának az érdekességeit, fontosságát, hovatovább szépségeit. Külön köszönöm a konzulensi szerep elvállalását és a segítségeket, iránymutatást, hogy dolgozatomat biztonságban befejezhessem.

Külön köszönettel tartozom szüleimnek, akik szeretettel, bátorítással támogattak tanulmányaim során. No meg persze étellel is, hogy biztosan egyek, akkor is, ha nem látják. Továbbá köszönettel tartozom családom többi tagjának, kislányomnak és húgomnak, akik minden alkalommal ki tudtak mozdítani fásultságomból, ezzel segítve céljaim elérését.

Utoljára, de nem utolsó sorban a szaktársaimnak mondok köszönetet, az összetartásért, a rengeteg segítségért, amit tanulmányaink során végig fenntartottunk.