



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem  
Villamosmérnöki és Informatikai Kar  
Automatizálási és Alkalmazott Informatika Tanszék

# Deep Learning alapú P300 jelfeldolgozás beágyazott környezetben

**TDK dolgozat**

Készítette:

Südi István  
Torma Szabolcs

Konzulens:

dr. Szegletes Luca

2020

# Tartalomjegyzék

**Kivonat**

**Abstract**

<b>1. Bevezetés</b>	<b>1</b>
<b>2. Rendszerterv</b>	<b>3</b>
2.1. Piackutatás . . . . .	3
2.1.1. EEG mérő áramkörök megvalósításai . . . . .	3
2.1.1.1. OpenBCI Cython Biosensing Board . . . . .	3
2.1.1.2. HackEEG . . . . .	3
2.1.2. Elektroenkefalográfia mérésére alkalmas IC-k . . . . .	3
2.1.2.1. Texas Instruments ADS1299 . . . . .	3
2.1.2.2. Texas Instruments ADS1298 . . . . .	5
2.1.2.3. Intan Technologies RHD2216 . . . . .	5
2.1.3. Bluetooth modulok . . . . .	5
2.1.3.1. Michrochip RN42-I/RM630 . . . . .	5
2.1.3.2. ST SPBTLE-RF0TR . . . . .	6
2.2. Hardver felépítése . . . . .	6
2.2.1. Mikrovezérlő . . . . .	6
2.2.2. Analóg front end . . . . .	7
2.2.3. Adat továbbítás . . . . .	8
2.2.4. Energia ellátás . . . . .	9
2.2.5. Áramkör elrendezése . . . . .	11
2.3. Változtatások az áramkör második verzióján . . . . .	13
2.3.1. Analóg front end . . . . .	13
2.3.2. Bluetooth modul . . . . .	14
2.3.3. Referencia feszültség forrás . . . . .	14
2.4. Rendszer felépítése . . . . .	14
2.5. Jetson Developer Kit . . . . .	18
<b>3. Áttekintés az EEG jelek méréséről és a Deep Learning területéről</b>	<b>20</b>
3.1. EEG technológia . . . . .	20
3.2. P300 jelek . . . . .	21
3.3. Deep Learning röviden . . . . .	23
<b>4. BCI rendszer fejlesztése</b>	<b>27</b>
4.1. P300 betűző fejlesztése . . . . .	27
4.2. Dekóder fejlesztése . . . . .	30
4.2.1. EEGNet, a state-of-the-art DL architektúra . . . . .	30
4.2.2. Adathalmazok . . . . .	31

4.2.3. Transfer Learning . . . . .	32
4.2.4. Keretrendszer a modell tanításához és teszteléséhez . . . . .	34
<b>5. A fejlesztett rendszer tesztelése és eredmények</b>	<b>37</b>
5.1. Rendszerteszt . . . . .	37
5.2. Végleges eredmények . . . . .	37
<b>6. Konklúziók és tervek a rendszer továbbfejlesztésére</b>	<b>40</b>
6.1. Összefoglalás . . . . .	40
6.2. Továbbfejlesztési tervek . . . . .	40
<b>Köszönetnyilvánítás</b>	<b>42</b>
<b>Irodalomjegyzék</b>	<b>43</b>
<b>Függelék</b>	<b>46</b>
F.1. Az elvégzett munka felosztása . . . . .	46

# Kivonat

A különböző agy-számítógép interfész (BCI) eszközök fejlesztése és alkalmazása egyre nagyobb teret hódít magának. Ezen rendszerek nem csak kényelmi funkciókat képesek szolgálni, hanem ami még fontosabb, kommunikációjukban és/vagy mozgásukban korlátozott emberek életminőségének javítását teszik lehetővé. A BCI eszközök direkt kapcsolatot létesítenek a felhasználójuk és a jelfeldolgozó számítógép között, amelyhez agyi jeleket használnak fel. Az agyi jelek rögzítésére a ma ismert legkevésbé komplex és legolcsóbb eljárás az elektroencefalográfia (EEG). Az EEG jelek feldolgozása sok kihívást rejt magában, főképp a jel rendkívül alacsony jel-zaj viszonya miatt. Az EEG jelek komponensei közé tartoznak a P300 jelek, amelyeket külső inger válthat ki. A P300 jelek napjainkban a legalkalmasabb jelek közé tartoznak a BCI rendszerek fejlesztésének tekintetében.

Dolgozatunk során egy olyan BCI rendszert mutatunk be, amelyet magunk hoztunk létre, és amely képes EEG jelek feldolgozására, dekódolására. Az eszköz használata során a felhasználó az eszköz segítségével képes lesz egy adott szó lebetűzésére, tisztán egy képernyőn látott 6x6-os betű- és számmátrix (ú.n. P300 betűző) használatával. Jelrögzítésre egy saját fejlesztésű EEG eszközt használunk, míg a nyert adat feldolgozására egy korszerű hardvert, az NVIDIA Jetson Nano Developer Kit-et. A Jetson Nano rendszerünk magját adja, amely felhasználói felületet kezel, jelfeldolgozást végez, mi több, egy modern mély neurális háló architektúrát futtat. Az agyi jelek dekódolására az EEGNet 2-dimenziós konvolúciós neurális háló modellt alkamazzuk, amely nagy pontosságú P300 detekcióra képes. A modell teljesítőképességét és adaptálhatóságát különböző eljárások segítségével növeljük.

Eredményeinket előre feldolgozott, nem nyers EEG jeleket tartalmazó adathalmazokon nyert eredményekhez mérve validáljuk. A dolgozat során bemutatjuk, hogy milyen lehetőségeket és kihívásokat rejt magában egy state-of-the-art BCI eszköz fejlesztése.

# Abstract

The development and application of various brain-computer interface (BCI) tools is gaining ground. Not only can these systems provide convenience functions, but more importantly, they can improve the quality of life of people with limited communication and/or mobility. BCI devices establish a direct connection between their user and the signal processing computer using brain signals. The least complex and cheapest method known today for recording brain signals is electroencephalography (EEG). The processing of EEG signals presents many challenges, mainly due to the extremely low signal-to-noise ratio of the signal. EEG signals include P300 signal components that can be elicited by an external stimulus. P300 signals are among the most suitable signals for the development of BCI systems nowadays.

In the course of our thesis we present a BCI system, which we created ourselves and which is able to process and decode EEG signals. When using the device, the user will be able to spell a specific word by looking at an on-screen 6x6 alphanumeric matrix (so-called P300 speller). We use a self-developed EEG device for signal recording, while a novel hardware, the NVIDIA Jetson Nano Developer Kit, is used to process the obtained data. The Jetson Nano provides the core of our system, which manages a user interface, performs signal processing, what's more, it runs a modern deep neural network architecture. To decode brain signals, we use the EEGNet 2-dimensional convolutional neural network model, which is capable of high-precision P300 detection. We increase the performance and adaptability of the model using different procedures.

Our results are validated by measuring them on pre-processed data sets containing non-raw EEG signals. In this work, we present the opportunities and challenges of developing a state-of-the-art BCI tool.

# 1. fejezet

## Bevezetés

A különböző agy-számítógép interfész (BCI) eszközök fejlesztése nem új keletű terület, azonban az orvostudomány és a jelfeldolgozási algoritmusok fejlődése folyamatosan új lehetőségeket nyit a BCI eszközök alkalmazási területeinek, illetve módjainak bővítésére. A BCI eszközök segítséget nyújthatnak a mozgásukban korlátozott személyek számára rehabilitáción vagy végtagokat helyettesítő robotprotézisek irányításán keresztül a mozgás részleges, bizonyos esetekben teljes visszanyerésére. Az eszközök másik fő alkalmazási területe a kommunikáció, amelyre a jelenleg legelterjedtebb példa a P300 betűző. A BCI rendszereket általánosságban négy fő egység alkotja: a jelfelvevő, a jelet előfeldolgozó, a dekódoló és a rendszert irányító egység.

A jelfelvevő egységeket két típusra bontjuk az elektródák implantáltsága szerint: az invazív és a non-invazív megoldás. A legelterjedtebb BCI rendszerek napjainkban a non-invazívak, mivel kevésbé komplikáltak, kisebb költségekkel járnak és - jelenleg pár megoldás kivételével - sokkal hordozhatóbbak, mint az invazív megoldások. További szempont, hogy az invazív alkalmazáshoz nagy fokú orvosi szakértelem és önkéntes páciensek hajlandósága szükséges, mivel a rendszer elektródáit ilyenkor az agyszövetbe ültetik. BCI rendszerek alapjául több orvosi jelelőzítésre és képalkotásra alkalmas technológia - pl. ECoG, EEG, EOG, fMRI, fNIRS, MEG - szolgálhat, azonban a jelenlegi kutatások túlnyomó többsége az elektroencefalográfiát (EEG) hasznosítja. Az EEG alkalmazása többnyire non-invazív módon történik, továbbá az olcsó, illetve a mozgást kevésbé korlátozó technológiák közé tartozik. Használatával különböző típusú agyi jelek rögzíthetők - köztük a P300 -, amelyek különböző módon használhatók fel külső eszközök irányítására. Az előbb felsorolt előnyök miatt a dolgozat során megvalósított BCI rendszerünk jelelőző egységét egy saját fejlesztésű EEG eszköz adja.

A felvett jelek előfeldolgozását, dekódolását, illetve a rendszer irányítását az NVIDIA által fejlesztett Jetson Nano Developer Kit végzi. A Jetson Nano-n fut emellett a P300 betűző alkalmazásunk is, miként a betűző alkalmassága kommunikációs feladatok terén számos irodalmi forrásban bizonyított, továbbá a használata során keletkező P300 agyi jelek jól dekódolhatók konvolúciós neurális hálókkal. Az eszköz fő alkotóeleme egy képernyőn megjelenő 6x6-os betű- és számmátrix, amely segítségével megkönnyíthető a pl. bezárt-szindrómás emberek kommunikációja. A jelek dekódolására egy state-of-the-art deep learning model architektúrát választottunk.

A különböző mély tanulási algoritmusok sikeres alkalmazása több területen is - köztük gépi látás, természetes nyelvfeldolgozás - jelentősen elterjedt. Az orvostudomány területén PET és CT képek elemzésére egyre gyakoribb választás, azonban agyi jelek

feldolgozására, dekódolására még nem terjedt el olyan szinten, mint az előbb említett területeken. A kevesebb alkalmazás ellenére az eddig született irodalom alapján nagy potenciált hordoz magában a különböző mély tanulási megközelítések alkalmazása az agyi jelek feldolgozásának területén. Sok megközelítés használ rekurrens neurális hálókat (RNN), auto-encoder-eket (AE), azonban a jelenleg legsikeresebben teljesítő modellek a konvolúciós neurális hálókat (CNN) tartalmazzák. Ezek a hálóak rendkívül jól képesek megtanulni a kívánt jelek karakterisztikáját, mind idő-, mind frekvenciatartományban és nagy pontossággal tudják elkülöníteni a célkomponenseket a többi, nem relevánsaktól. Rendszerünkben a jel dekódolását az EEGNet végzi, amely egy modern, nagy pontosságú, beágyazott rendszerekben való alkalmazásra optimalizált architektúra.

A dolgozat felépítése a következők szerint alakul. A *fejezet 2*-ben a fejlesztett rendszer egészének felépítését ismertetjük. A *fejezet 3*-ban az EEG technológia, a rögzített jelek, illetve a P300 jelkomponens tulajdonságait ismertetjük, kitérve az alkalmazás szempontjából releváns előnyökre és nehézségekre. Az elméleti áttekintés után a *fejezet 4*-ben rendre a P300 betűző, illetve a dekóder fejlesztésének menetét és tanulságait dokumentáljuk. A *fejezet 5*-ben prezentáljuk az tesztelés folyamatát és annak eredményeit, összehasonlítva más, előre felvett és szűrt adathalmazokon szerzett eredményekkel. Végezetül *fejezet 6*-ban összefoglaljuk konklúzióinkat a rendszer fejlesztésével és annak eredményességével kapcsolatban, továbbá javaslatokat teszünk a rendszer továbbfejlesztésére.

## 2. fejezet

# Rendszerterv

### 2.1. Piackutatás

#### 2.1.1. EEG mérő áramkörök megvalósításai

##### 2.1.1.1. OpenBCI Cython Biosensing Board

Az OpenBCI Biosensing Board (2.1 ábra) 8 differenciális csatornán képes mintavételezni és a mért adatot Bluetooth-on továbbítani egy számítógépre, ahol az USB-s OpenBCI Dongle-t kell használni az adatok fogadására. A Daisy Biosensing Board segítségével a csatornák száma 16-ra bővíthető. Beépített gyorsulásmérővel rendelkezik. 3.3-12 V tápfeszültségről működtethető, ami nagyon széles tartomány. A PIC32MX250F128B mikrovezérlőt használja, amit az Arduino környezetből is lehet programozni, tehát azok számára is megfelelő lehet, akik csak éppen ismerkednek a beágyazott rendszerek programozásával. A Texas Instruments ADS1299 analóg-digitális konverterét alkalmazza a mintavételezésre. Az EEG elektródákat nem a hagyományos DIN 42802-1 szabványnak megfelelő csatlakozóval köti az áramkörhöz, hanem tűskesorral, ami problémát jelenthet, ha valaki más forrásból szeretne elektródát beszerezni és nem akarja átalakítani azokat. Aktív és passzív elektródákkal egyaránt üzemeltethető [22].

##### 2.1.1.2. HackEEG

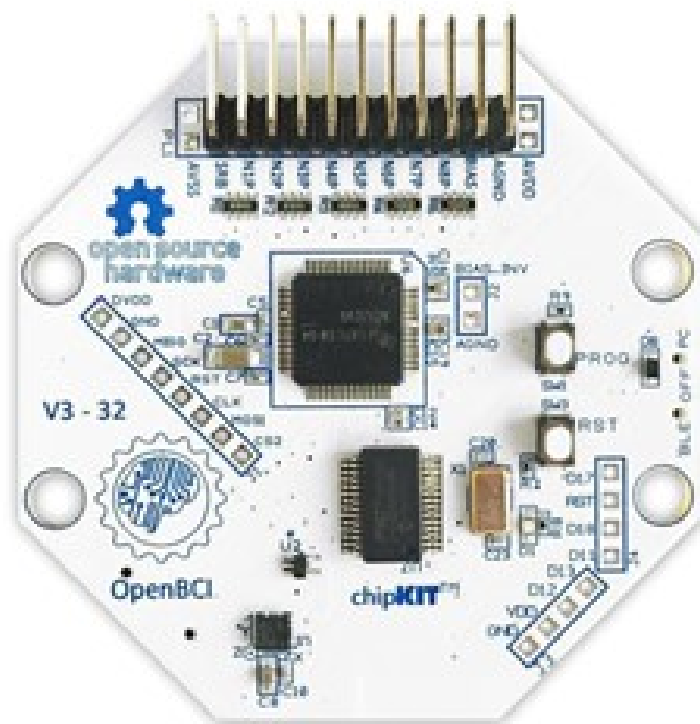
A HackEEG Adam Feuer közösségi finanszírozásból megvalósított projektje, aminek célja, hogy egy olcsó, sokak által elérhető eszközt adjon az emberek kezébe, amivel könnyen lehet EEG, EMG és EKG jeleket rögzíteni (2.2 ábra). Az áramkör egy Arduino Due bővíítőpanelként funkcionál. A Texas Instruments ADS1299 IC-jének segítségével 8 csatornán képes mérést végezni. Maximálisan 4 HackEEG áramkör köthető egymáshoz és ezzel 32-re növelhetjük a csatornák számát. Mivel a galvanikus leválasztás nincs a panelen, ezért csak akkumulátorról üzemeltethető. A termékét Adam teljesen nyíltan elérhetővé tette, tehát a fájlok birtokában bárki készíthet megának egy ilyen áramkört [5].

#### 2.1.2. Elektroenkefalográfia mérésére alkalmas IC-k

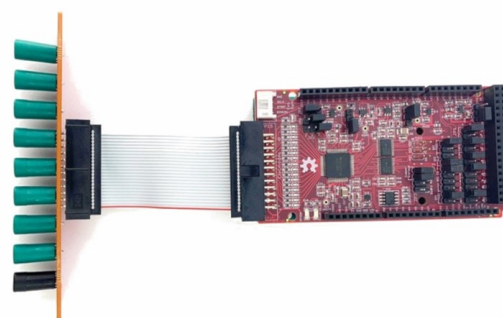
##### 2.1.2.1. Texas Instruments ADS1299

Az ADS1299 IC 4, 6, vagy 8 csatornás 24 bites analóg-digitális átalakító IC EEG és biopotenciál mérésére lett tervezve. Igény esetén sorba köthető több IC. Programozható erősítővel rendelkezik, ami 1 és 24 közötti előre meghatározott értékekre állítható be. A főbb alkalmazási területek EEG, magzati ECG és az alvás monitorozás. A mintavételezést előre meghatározott frekvenciákon végezhetjük 250 és 16000 minta / másodperccel és SPI segítségével olvashatjuk ki. Detektálhatjuk vele a leváló elektródákat. Belső multiplexere





2.1. ábra. OpenBCI Cyton Biosensing Board [22]



2.2. ábra. HackEEG áramkör szabványos elektróda bővítő panellel [5]

miatt kényelmesen üzemeltethető különböző módokban. A közös jeleket a megfelelő bemenetekre kapcsolhatjuk vele. Belső jelgenerátorával tesztelhetjük a kiolvasást, lemérhetjük az IC hőmérsékletét, valamint a tápfeszültségeket is a delta-sigma ADC-kre vezethetjük. A közös módosú elnyomása -110 dB. A TQFP tokozás relatíve könnyen forraszhatóvá teszi. A lábkiosztása miatt kompatibilis az ADS129X IC-vel [9]. Az egységára 70.39 USD. Beszerezhető az általam általában használt Farnell oldalról.

#### **2.1.2.2. Texas Instruments ADS1298**

Az ADS1298 egy 8 csatornás analóg-digitális átalakító biopotenciál mérésére. A programozható erősítője 1 és 12 között állítható fix értékekre. Főbb alkalmazási területei az ECG, EMG, EEG, beteg monitorozás és alvás monitorozás. A minimális mintavételezési frekvenciája megegyezik az ADS1299-ével, a maximális 32000 minta/másodperc, ami az ADS1299 maximumának kétszerese. Ez az IC is SPI kompatibilisan kommunikál a mikrovezérlővel. Az ADS1299-hez hasonlóan a belső multiplexerrel az ADC-k bemeneteire. A közös módosú elnyomása -115 dB, ami az ADS1299-nél 5 dB-lel jobb. A tokozás az előző pontban említett TQFP, valamint NFBGA verzióban is elérhető [8]. A BGA tokozás használatához előnyös, ha 2-nél több rétegű nyomtatott áramkört alkalmazunk. Ennek a verzióknak a használata akkor lenne előnyös, ha már a TQFP tokozásával sikerült egy működő prototípust építeni. Egy második verziónál a kisebb méret előnyös lehet, míg az elsőnél a hozzá nem férhető lábak, valamint a forrasztáskor fellépő nehézségek miatt nem javasolt. Az egységára 42.07 USD. Beszerezhető az általam általában használt Farnell oldalról.

#### **2.1.2.3. Intan Technologies RHD2216**

Az RHD2216-os IC 16 differenciális bemenettel rendelkezik, amiket 16-bit-es analóg-digitális átalakítókkal akár 30000 minta/másodperces frekvenciával mintavételezhetünk. Hagyományos SPI interfészen keresztül lehet a jeleket kiolvasni, ami CMOS vagy LVDS feszültség szintekkel működik. Ha az LVDS-en lábát a föld potenciáljára lehúzzuk, akkor az SPI LVDS módban fog működni, aminek alacsonyabb a feszültsége és differenciális. Az erősítő felső és alsó vágási frekvenciája is beállítható a regiszterértékek módosításával. Képes a felszerelt elektródák impedanciájának mérésére. Kiegészítő ADC bemenetekkel rendelkezik. EMG, EKG vagy EEG mérési területeken alkalmazzák [10]. Egy IC 260 USD-ért szerezhető be.

### **2.1.3. Bluetooth modulok**

Mivel a Bluetooth modulok között sokkal szélesebb a választék, ezért elsősorban azokra koncentrálok, amik a Farnell oldalon keresztül is megvásárolhatóak.

#### **2.1.3.1. Microchip RN42-I/RM630**

Az RN42-es modul a Bluetooth 2.1-es szabványnak megfelel. Két kivitelben érhető el, antennával és antenna nélkül (2.3 ábra). Az antennával rendelkező modell méretei 13.4x25.8x2.4 mm. UART-on és USB-n keresztül lehet vele kommunikálni. USB-n keresztül akár 3.0 Mbps adatátviteli sebességre is képes. Több változata elérhető, amik különböző Bluetooth profilokat támogatnak úgymint GAP, SDP, RFCOMM, L2CAP és DUN. Maximálisan 10 m-es távolságig használható. UART-on keresztül ASCII parancsokkal konfigurálható a működése [19].



2.3. ábra. RN42 és RN42N Bluetooth modulok [19]



2.4. ábra. ST SPBTLE-RF0TR Bluetooth modul [20]

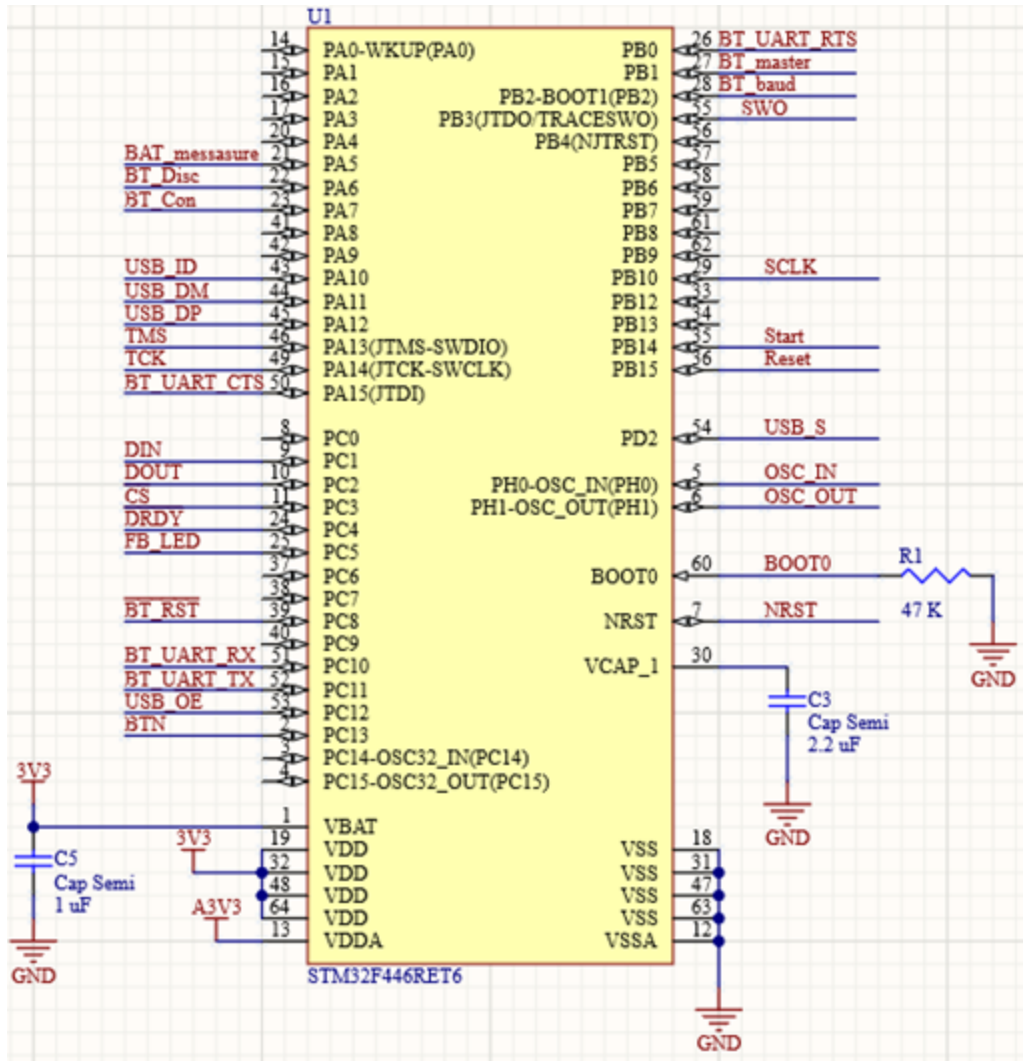
### 2.1.3.2. ST SPBTLE-RF0TR

Az ST Microelectronics SPBTLE-RF0TR Bluetooth modulja a 13.5x11.5x2.0 mm-es befoglaló méreteivel különösen alkalmas kis helyigényű áramkörök létrehozására (2.4 ábra). A Bluetooth 4.1-es szabványának megfelel a modul, ami kis fogyasztású alkalmazások esetén előnyös. SPI interfészen keresztül lehet kommunikálni vele. A legegyszerűbben a gyártó által biztosított függvényekkel használható. Az ST kiadott egy fejlesztői kártyát is ezzel a modullal X-NUCLEO-IDB05A1 néven. A BLE (Bluetooth Low Energy) protokollok közül a GAP, GATT, L2CAP, LL és RFPHY kerültek implementálásra a gyártó által [20].

## 2.2. Hardver felépítése

### 2.2.1. Mikrovezérlő

Az elektroencefalográfia (EEG) jelek méréséhez terveztem egy áramkört. Az áramkör alapját egy STM32F446RET6 mikrovezérlő képezi, ami kellően erős ahhoz, hogy a mérő IC-ről az adatokat begyűjtse, azokon kisebb előfeldolgozást végezzen és a kommunikációs feladatokat is ellássa (2.5 ábra). A mikrovezérlő kiválasztásánál fontos szempont volt, hogy ezzel az IC-vel már rendelkeztem tapasztalattal a programozás és az áramkör tervezés terén is. A mikrovezérlő programozására egy Nucleo-n található ST-Link V2-t fogok használni. Az áramkörhöz tartozik egy bekapcsolást jelző és egy általánosan használható LED, mely a hibakeresést jelentősen megkönnyíti. A mikrovezérlő egy gombbal resetelhető. A funkció váltásra és a tesztelésre egy további gomb is elhelyezésre került a nyomtatott áramkörön.

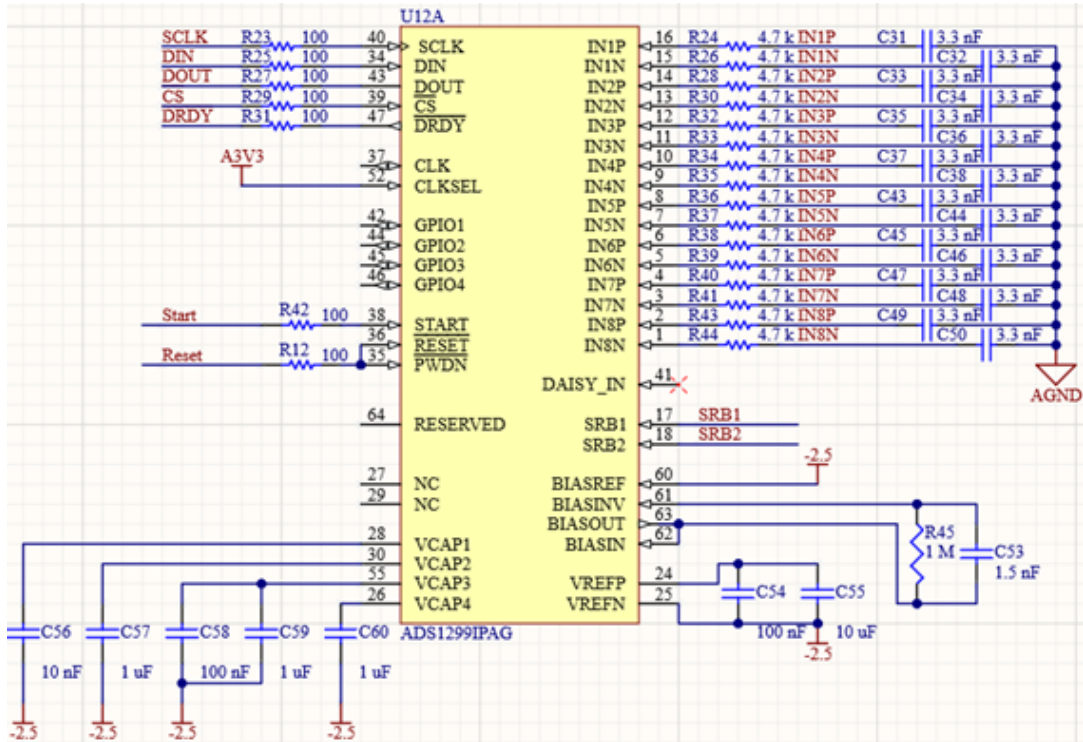


2.5. ábra. Mikrovezérlő bekötése

Az órajelet egy 8 MHz-es kristállyal állítom elő, mivel a belső RC oszcillátorok frekvencia stabilitása elmarad a külső kristályokétól.

### 2.2.2. Analóg front end

Az EEG jeleket egy ADS1299IPAG ADC IC-vel fogom mérni, mert a lehetőségek közül ez tűnt a legkönnyebben használhatónak és a helyigénye is kisebb volt az alternatívákénál, valamint ezt EEG jelek mérésére tervezték. A mérő IC bekötését a 2.6 ábra és a 2.7 mutatja. A passzív alkatrészek értékét és a bekötését az IC adatlapján szereplő értékeknek megfelelően választottam ki. Az ADC bemenetek kivételt képeznek ez alól, mivel az aluláteresztő szűrőt az adott alkalmazáshoz kell illeszteni. Az értékek meghatározásakor a maximálisan átengedett frekvencián a veszteséget 1 % körüli értéken tartását tűztem ki célul, hogy a mért jeleket jelentősen ne nyomjam el. Lehetett volna még a jeleket mérni általános ADC-vel is, akkor a csatornákon a jelet műveleti erősítővel és analóg szűrő áramkörrel kellett volna előkészíteni a mintavételezésre. Az 50 Hz elnyomása, a magas frekvenciák szűrése és az értékes frekvencia tartomány kiemelése annyira sok áramköri elemet jelentett volna, ha mind a 16 csatornánk külön kéne megvalósítani, hogy ezt a lehetőséget inkább elvettem. A jelek egy DIN42802 szabványnak megfelelő csatlakozón jönnek be,

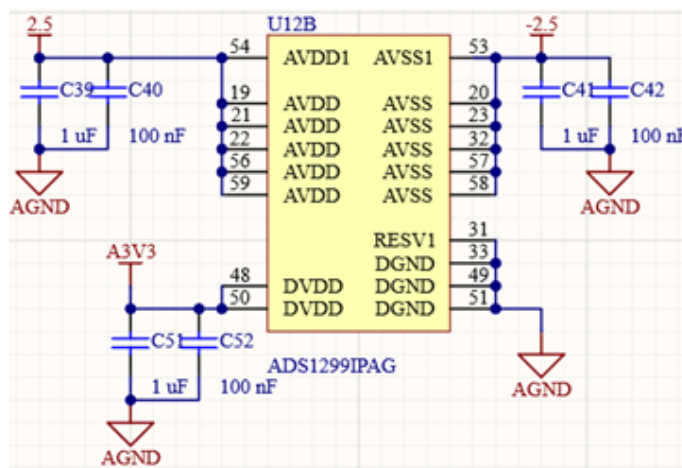


2.6. ábra. Analóg front end bekötése

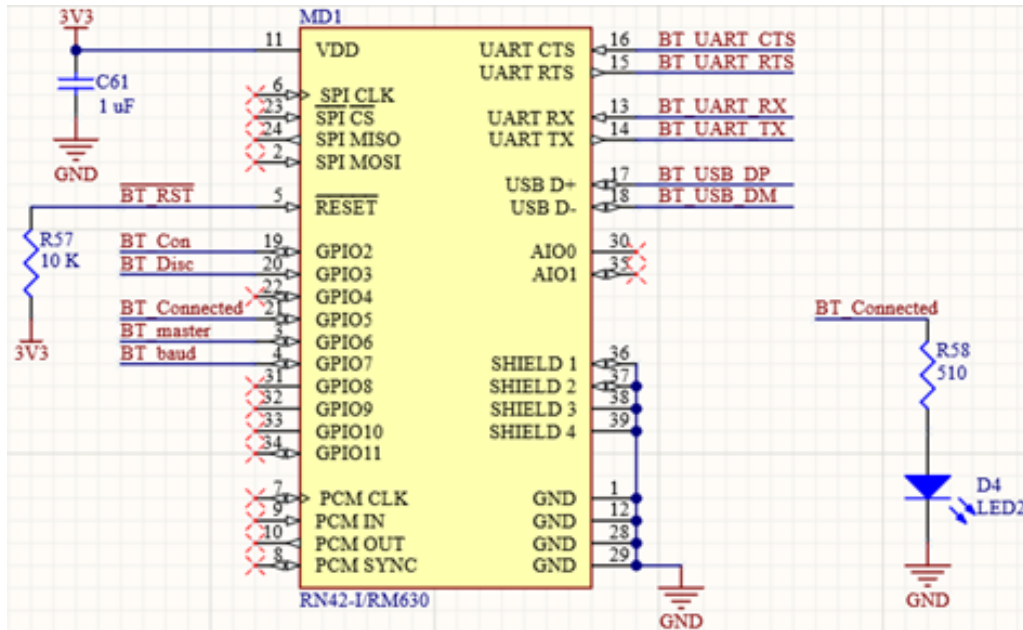
mert az EEG mérő elektródák ilyen csatlakozóval kompatibilisek. Minden csatornán TVS dióda van az esetleges ESD okozta károk elkerülésére. Magas feszültségű zavarok könnyen kerülhetnek az elektródákra, ha a mért személy műszálas ruházatot visel például. Ezért ez ellen mindenképpen védekezni kell. A mérő IC bipolárisan van táplálva az adatlapon feltüntetett módon.

### 2.2.3. Adat továbbítás

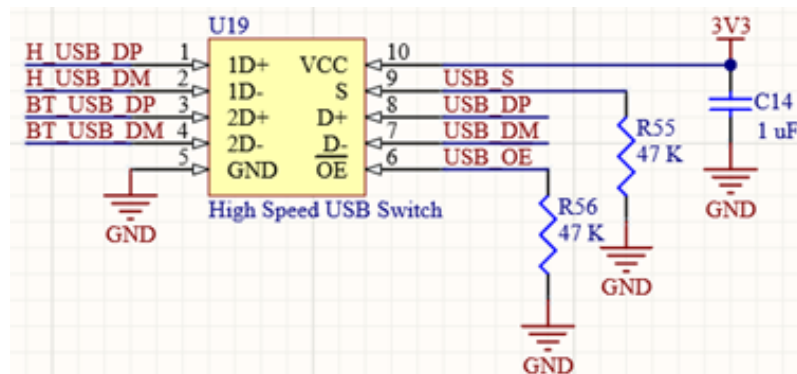
Az áramkör által mért értékeket Bluetooth-on keresztül lehet továbbítani normál üzemetelés során, amire a Microchip RN42-I/RM630 modul szolgál (2.8 ábra). Azért döntöttem mellette, mert a Bluetooth 2.1 még a könnyen használható verziók közé tartozik. A low



2.7. ábra. Analóg front end energia ellátása



2.8. ábra. Bluetooth modul



2.9. ábra. USB útválasztó

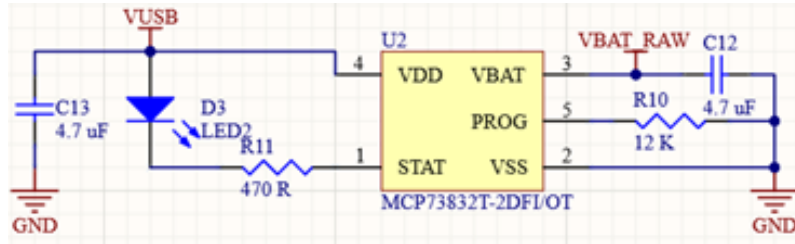
energy modulok kevesebb helyet foglalnak el az áramkörben, de a működtetésük sokkal több beállítást igényel és a hiba keresés is több időt vesz igénybe. Az általam választott eszköz 3Mbps sebességre is képes, ha USB-n keresztül kapja az adatokat és nem UART-on keresztül. Mivel az USB használatot sem szeretném ellehetetleníteni, ezért egy USB útválasztót is kellett az áramkörben használni (9.2.9 ábra). Az ábrán láthatóhoz hasonló IC-eket USB elosztókban szokták alkalmazni. A select láb segítségével tudom kiválasztani, hogy az USB csatlakozó vagy a Bluetooth modul legyen az STM mikrovezérlő USB kimenetével összekötve. Az USB bemenetre külön TVS diódás ESD védelmet helyeztem, hogy az utána következő alkatrészeket megvédhessem.

## 2.2.4. Energia ellátás

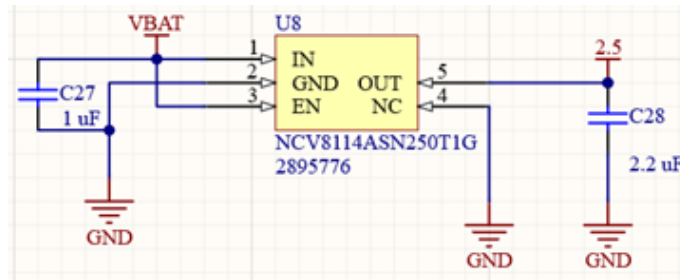
Az áramkört lítium-ion vagy lítium-polimer 1 cellás akkumulátorról lehet táplálni. A töltő áramkörrel ezt külső töltő nélkül lehet használni. A töltés állapotát egy visszajelző LED-ről lehet leolvasni (2.10 ábra).

Az akkumulátorról 2.5 és 3.3V-ot állítok elő LDO-val (11.2.11 ábra és 12.2.12 ábra). Az IC-eket az adatlapjaikon feltüntetett módon kötöttem be. A 3.3V-os LDO-ból kettő van az áramkörre építve, hogy az analóg front end-hez a lehető legkevesebb zaj menjen





2.10. ábra. Akkumulátor töltő IC

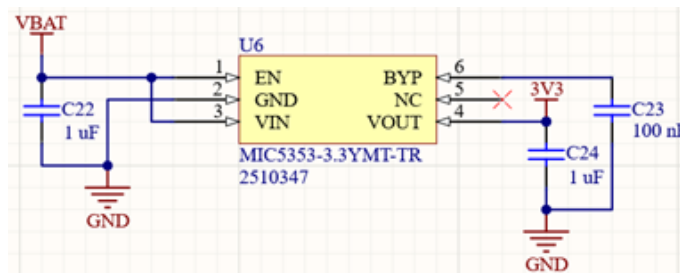


2.11. ábra. 2.5 V-os LDO

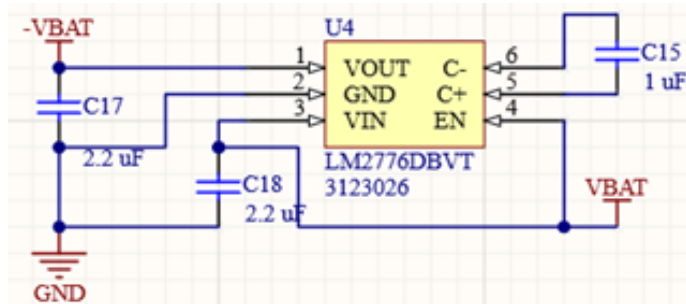
a többi digitális áramköri elemtől. Ezzel remélhetőleg a mért jelek minősége javul. A mérő IC-t tápláló 3.3V és a többi komponenst ellátó 3.3 V-os busz között van lehetőség egy ferrit maggal kapcsolatot teremteni, tehát tesztelhető a teljes áramkör egy 3.3 V-os LDO-ról is, ezzel a két felmerülő lehetőség összehasonlítható. Ez egy továbbfejlesztett változatnál jelenthetné azt, hogy egy LDO-val és a hozzá kapcsolódó passzív elemekkel kevesebb alkatrész is elláthatja ugyanazt a funkciót. Ugyanakkor meg kell jegyezni, hogy az LDO a hatásfokát tekintve rosszabb, mint a Buck vagy a boost konverterek, mert a terhelésen az állandó feszültséget a saját ellenállásának változtatásával éri el. Tehát jelentős lehet a rajta diszipálódó teljesítmény.

Az akkumulátor feszültségéből charge pump segítségével annak a -1 szeresét előállítom, hogy egy LDO-val -2.5V-ot tudjak készíteni (2.13 ábra és 2.14 ábra).

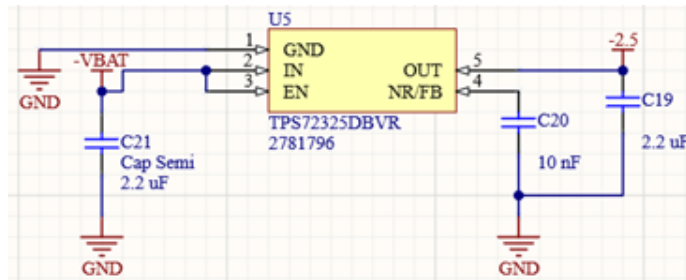
A 2.5 és -2.5V-tal lehet a mérő IC-t ellátni. Ezzel a megoldással nem kell egy drágább 2 cellás akkumulátort választani a drágább töltő IC-vel együtt és elkerülhető a boost konverter tekerése által keltett zaj is. Az akkumulátor csatlakozója után közvetlenül található egy fordított polaritás bekötése és zárlat elleni védelem (2.15 ábra). Zárlat esetén a PTC hőmérséklete és ellenállása megnő, így az utána elhelyezett alkatrészekeken kisebb feszültség esik. Az áramkör bekapcsolását a nyákra helyezett kacsolóval szeretném a tesztek alatt elvégezni, de ezt a műanyag védőházba helyezve már nem lehet elérni. Ezt azzal küszöbölnöm ki, hogy egy tűkesort is párhuzamosan bekötök a kapcsolóval. Erre a pontra lehet bekötni a burkoltra helyezett kapcsolót. Az akkumulátor feszültségét egy feszültség



2.12. ábra. 3.3V-os LDO



2.13. ábra. Charge pump bekötése



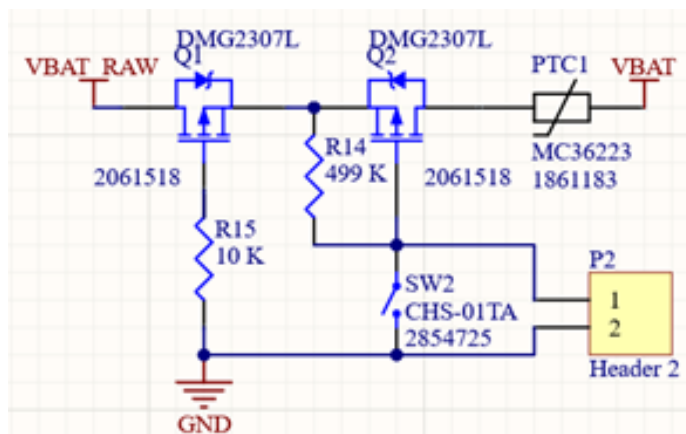
2.14. ábra. -2.5 V-os LDO

osztó ellenálláspárral a mikrovezérlő számára mérhető értékre csökkentem és az egyik lábán mintavételezem. Egy hosszabb mérési sorozatnál fontos lehet, hogy az akkumulátor mennyire merült le és mikor kell elkezdni tölteni, esetleg a mérést mindenképpen megszakítsuk egy adott érték elérésekor, hogy az akkumulátor semmiképpen ne károsodjon.

Az elterjedt elektronikai alkatrészeket a Farnell-től szereztem be. A szabványos (DIN 42802-1) EEG csatlakozókat Shenzhen Sinelink Electronics nevű cégtől vásároltam.

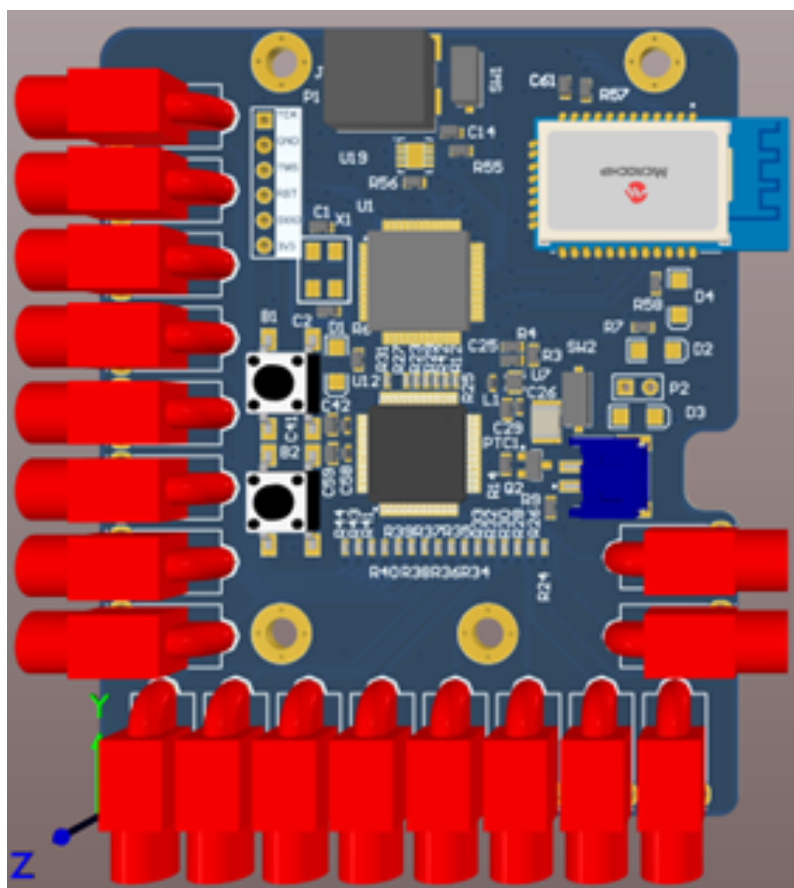
### 2.2.5. Áramkör elrendezése

Az áramkör felülnézeti képét a 3D-s modellről az 2.16 ábra mutatja, alulról a 2.17 ábra szemlélteti. Az áramkör befoglaló mérete a kilógó csatlakozókkal együtt 65 mm x 80 mm A tervezéshez az AUT által biztosított Altium Designer programot használtam. Ezt azért választottam, mert a RobonAUT verseny alatt szerzett áramkörtervezési tapasztalatom-



2.15. ábra. Áramkör védelem

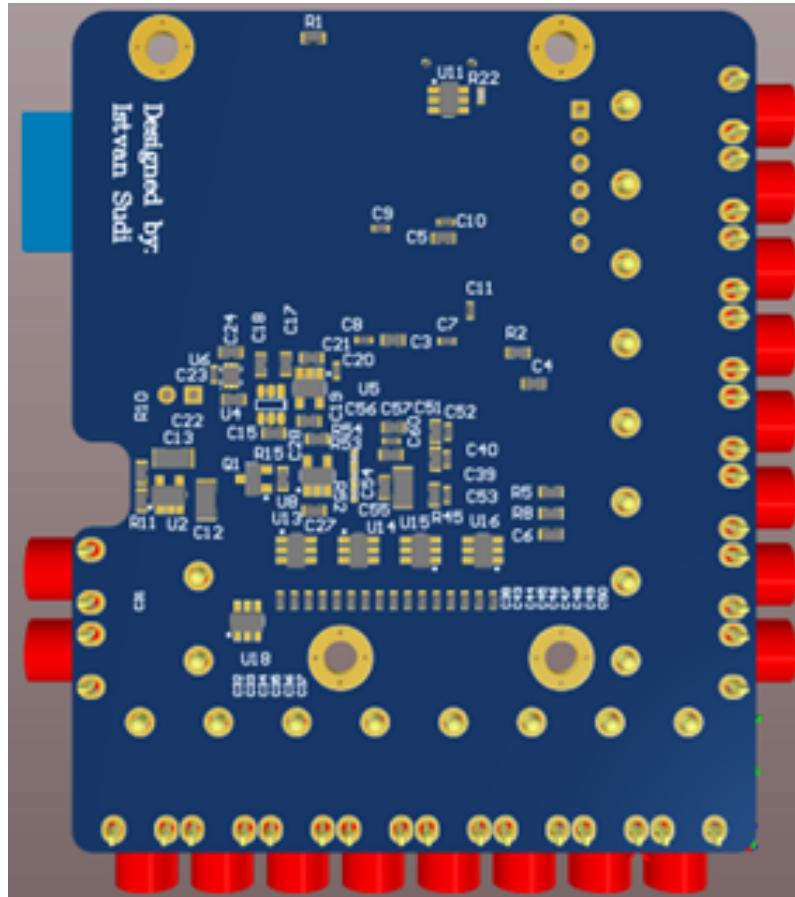




**2.16. ábra.** EEG jel mérő áramkör felső oldala

mal ebben a programban tudtam a leggyorsabban haladni. A régebben kipróbált ingyenes alternatívák nem rendelkeztek ennyi funkcióval, amik a fejlesztést meggyorsították.

Az 2.16 ábra tetején található az USB csatlakozó. Az adat vonalak a mikrovezérlő jobb oldalán találhatóak, így nem keresztelték más vonalak, valamint a töltéshez használt 5 V is kényelmesen elvezethető volt a Nyák szélén. Az USB adat vonalait meg kellett szakítani az USB elosztó IC-vel, hogy a Bluetooth modult is rá lehessen kötni. Az ábra jobb felső sarkában a Bluetooth modul helyezkedik el. A modul antennája körül az adatlapon meghatározottak szerint nem lehet vezető réteg és ajánlott az üvegszálát kivágni. Középen a halványzöld IC a mikrovezérlő. Ez a felső nyomógombbal újraindítható, az alatta lévő gomb szabadon felhasználható. Az USB csatlakozó bal alsó felénél lévő feliratozott csatlakozón keresztül programozható a mikrovezérlő. Az áramkör közepén a nagy fekete IC az analóg front end, amivel az EEG jeleket mérem. A nagy alkatrészűréség miatt a szűrő RC kör ellenállásai a felső, míg a kondenzátorok az alsó oldalon fértek csak el, amik a 2.17 ábra alsó felén egy sorban megtalálhatóak. A kondenzátorok fölé a TVS dióda blokkok kerültek, mivel ez volt a legközelebbi szabad tér a mérő IC körül. A felső oldalon a sötétkék csatlakozóba köthető az 1 cellás lítium-polimer vagy lítium-ion akkumulátor. A csatlakozó alatt az alsó oldalon elfért a teljes töltésvezérlő áramkör. Körülötte a bekapcsolás, fordított polaritás és zárlat elleni védelem komponensei foglalnak helyet. A töltés állapotát az akkumulátor csatlakozója feletti LED mutatja. Az alsó oldalra került a mikrovezérlő bootolását befolyásoló kapcsoló. Ezzel lehet beállítani, hogy a mikrovezérlő indulásakor honnan töltsse be a futtatandó programot. A felső oldalon az áramkör széle mentén látható piros csatlakozóba köthetők az EEG elektródák. Ezek szándékosan lógnak túl az áramkör szélén, mert így a burkolatból ki fognak látszódni annyira, hogy kényelmesen hozzájuk



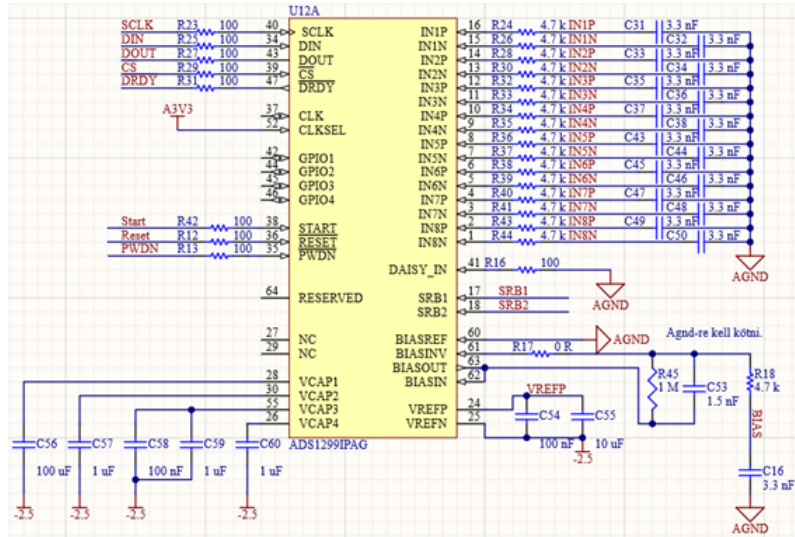
2.17. ábra. EEG jel mérő áramkör alsó oldala

lehesen féni. A sárga furatok átmérője akkora, hogy M3-as csavarok átférjenek rajtuk. A rögzítést nem sikerült szimmetrikusan az áramkör sarkaihoz elhelyezni, mert az EEG csatlakozók nagyon sok helyet elfoglalnak és az áramkör méretét a lehető legkisebbre szerettem volna elkészíteni. A kis mérethez részben a kényelmesebb használhatóság, részben az olcsóbb gyárthatóság miatt ragaszkodtam. A gyártási költségek csökkentését szolgálta még az, hogy az áramkör csak 2 vezető réteget használ. A tervezést megnehezítette a 2 réteg alkalmazása a 4 réteghez képest, de végül így is el lehetett rendezni az alkatrészeket.

## 2.3. Változtatások az áramkör második verzióján

### 2.3.1. Analóg front end

Az ADS1299-es mérő IC bekötésénél több hibát is találtam az első verzióban. Az áramkör jobban működtethető, ha a RESET és a PWDN lábak nincsenek összekötve, mivel ezeket külön is kell állítani. Meg kell jegyezni, hogy az alkalmazási példánál az adatlapon is összekötve szerepelnek ezek a lábak és parancsként is kiküldhető az SPI-on a reset. A VCAP1 nevű lábón lévő C56 kondenzátor értéke az első verzióban hibásan 10 nF lett, ami több nagyságrenddel eltér az adatlap alapján javasolt 100  $\mu$ F-tól. Mivel a kiszerezések méretei is jelentősen eltérnek, ezért ez volt az egyik indokom, hogy miért kellett második verziót gyártani az áramkörből. A BIASINV lábára lehet még egy csatlakozót forrasztani, amit az első verziónál nem tettem meg, de a másodikonál már fenntartom ennek is a lehetőségét (2.18 ábra).



2.18. ábra. Analóg front end bekötése a második verziónál

### 2.3.2. Bluetooth modul

Az áramkör első verziójánál még nem tűnt fontosnak, hogy a Bluetooth modul 22-es lábán elérhető factory reset funkciót igénybe vegyem. A programozás során rájöttem, hogy a mikrovezérlőn van még néhány elérhető lábam, amiket úgysem használok ki és a gyári állapot visszaállítása mégis hasznos funkció abban az esetben, ha valamit olyan értékre állítanék véletlenül be, ami miatt később nem lehetne a paramétereket módosítani (2.19 ábra).

### 2.3.3. Referencia feszültség forrás

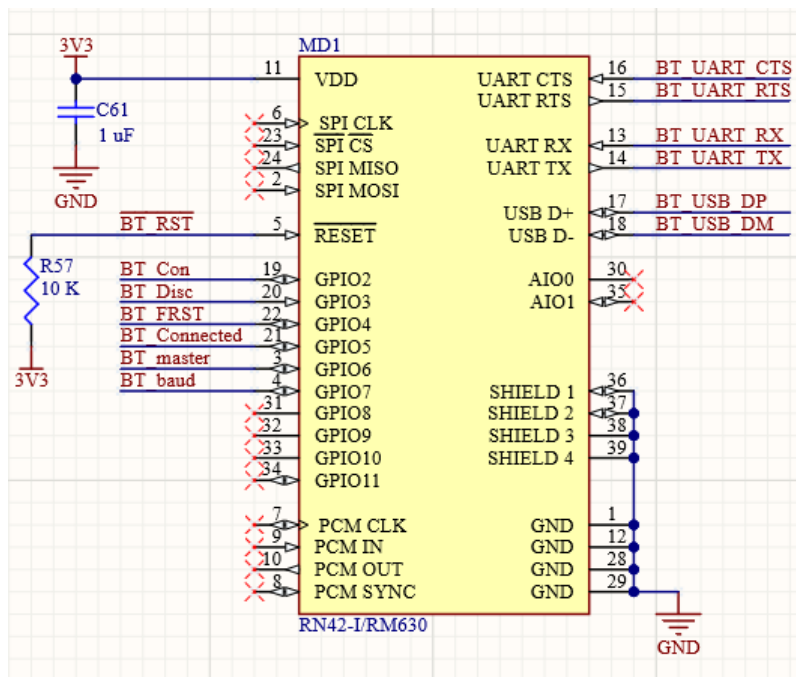
Az ADS1299 IC lehetőséget biztosít, hogy külső forrással biztosítsuk a referencia feszültséget, aminek nagyobb lehet a stabilitása. A referencia feszültség forrásokat tipikusan nagyon alacsony terhelhetőségűre készítik. A Maxim MAX6107EUR+T IC terhelhetősége 5 mA. A mérő IC adatlapja azt javasolja, hogy a referencia feszültség forrást az alábbi kapcsolással kössük az áramkörünkbe, amit a 2.20 ábra mutat.

A referencia feszültség forrásokat tipikusan nagyon alacsony árammal lehet csak terhelni. A referencia feszültség stabilitása miatt a mérő IC adatlapja javasolja, hogy kössünk be még egy műveleti erősítőt is a feszültségforrás után, hogy a zajokat jobban el lehessen nyomni. Ennek a bekötését a 2.21 ábra mutatja meg. A VREFP-re való rákötés előtt a kimentési feszültségen van egy 0 Ω-os ellenállás, így ezt leforrasztva továbbra is használható marad az ADS1299-es IC belső referencia feszültség forrása.

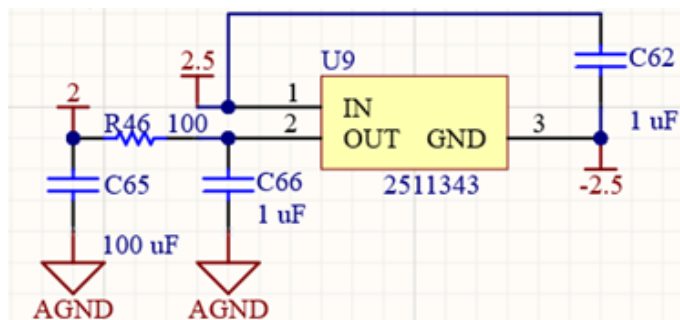
## 2.4. Rendszer felépítése

A 2.22 ábra mutatja be a teljes rendszer felépítését. A grafikus felhasználói felülettel rendelkező programban indíthatjuk el a sorok és oszlopok vagy a karakterek felvillantását. A felhasználói felület felvillanásai stimulánsként hatnak a mért személy agyára. A mért EEG jelet továbbítjuk az alkalmazásnak, ami ezt lementi.

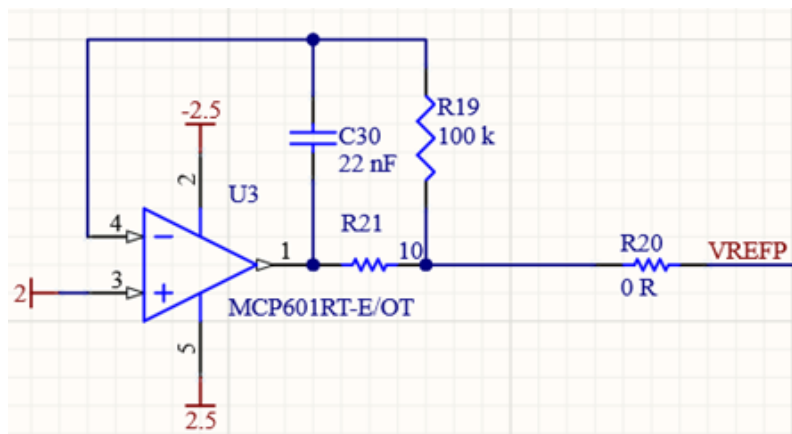
A beágyazott rendszerben az inicializálással kell kezdeni, mivel az összes használt periféria elvárt működéséhez be kell állítani a paramétereket (2.24 ábra). Az inicializáláshoz szükséges kód nagy részét az STM32CubeIDE program generáltattam le az STM32F446RE típusú mikrovezérlőre. Az ADS1299-es ADC IC-hez én írtam meg a kódot,



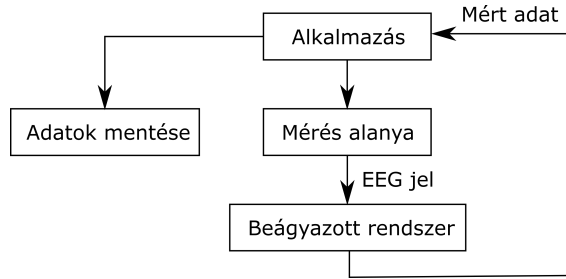
2.19. ábra. Analóg front end bekötése a második verziónál



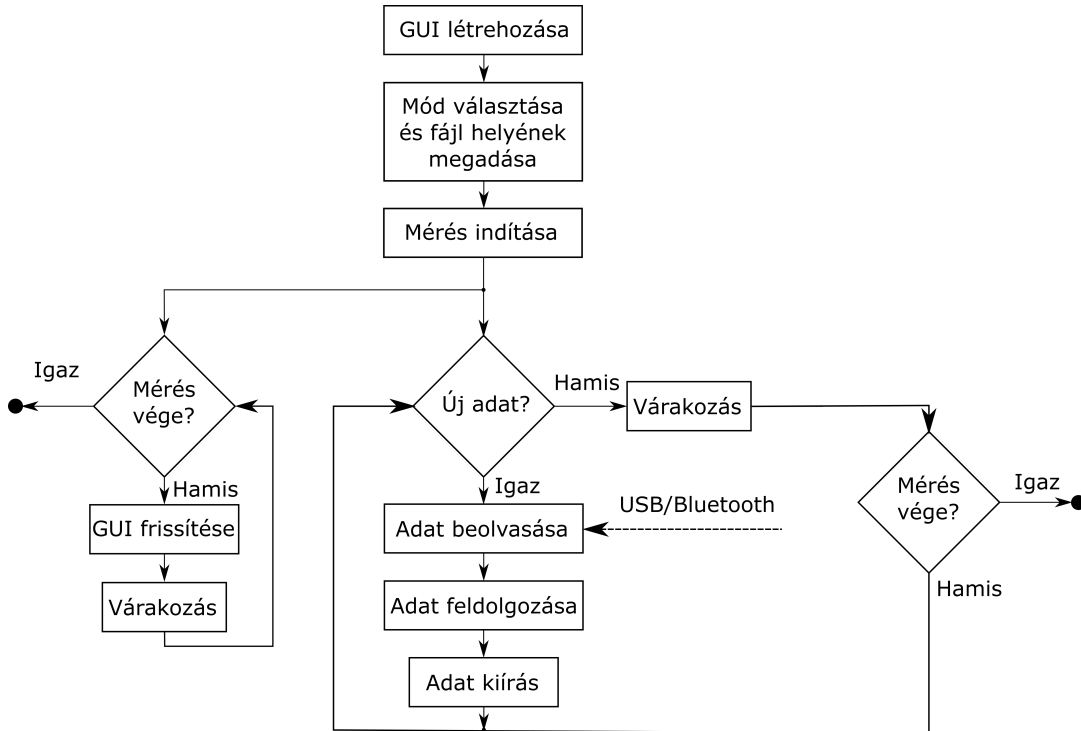
2.20. ábra. 4.5 V-os feszültség forrás



2.21. ábra. Műveleti erősítő

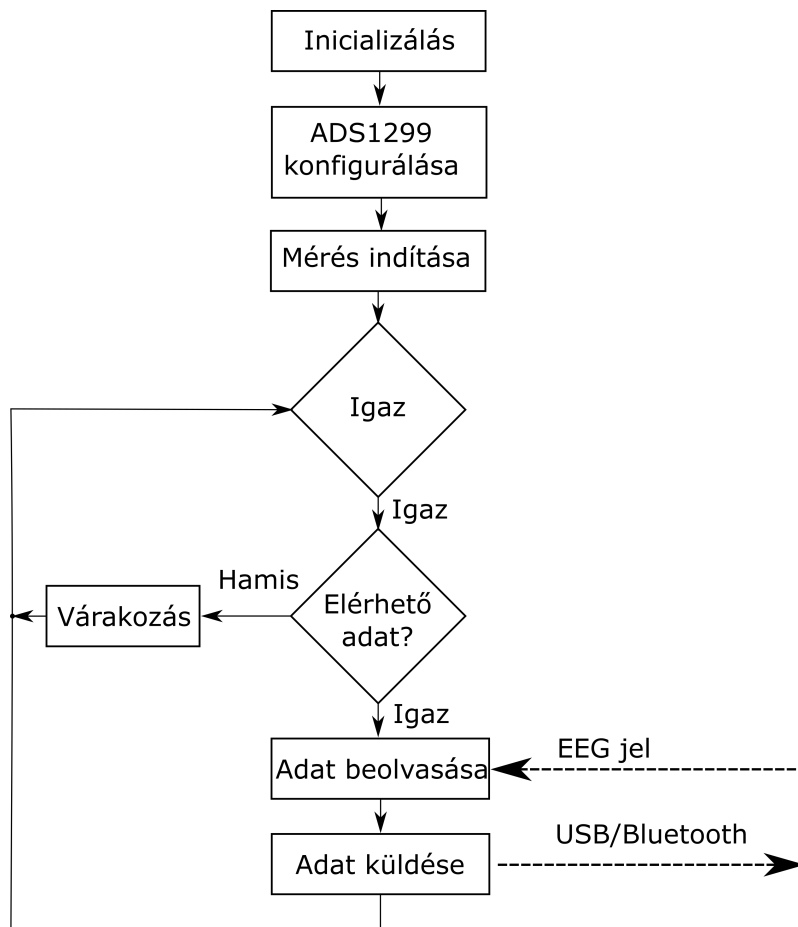


2.22. ábra. Rendszer Felépítése



2.23. ábra. Alkalmazás felépítése

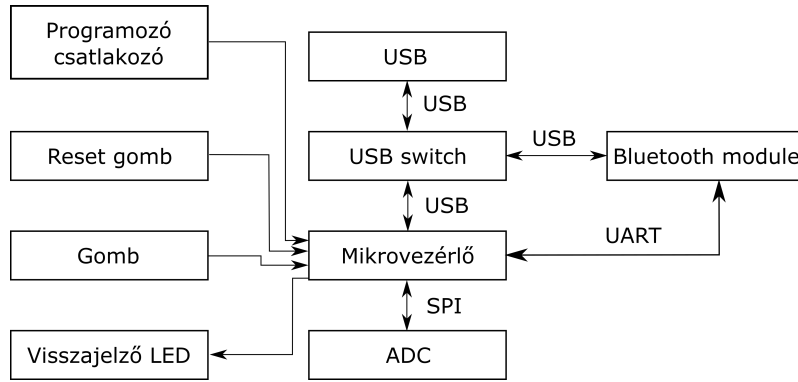
amivel a konfigurációt és az adatok beolvasását elvégeztem. A fejlesztés során a regiszterek írását követően vissza is olvastam azokat, majd a virtuális COM port-on kiküldtem, hogy könnyen és gyorsan ellenőrizhető legyen a regiszterek beállítása. Ezt követően az IC-nek SPI-on keresztül kiadhatjuk a mérés indítása parancsot. Ha a mérési értékek készen állnak a kiolvasásra, akkor az ADS1299 IC data ready lábát lehúzza. Ezt külső megszakításként kezeltem le. A mérési adatokat egy végtelen ciklusban olvasom ki az ADC-ről. Az adatokat USB-n vagy Bluetooth-on keresztül továbbítja a mikrovezérlő a Jetson Nano felé. A Jetson Nano-n futó alkalmazásnál első lépésben a GUI-t hozom létre és jelenítem meg (2.23 ábra). A felhasználó ezt követően kiválaszthatja a fájl mentésének helyét vagy a felülírti kívánt fájlt. Két különböző módban működtethető a GUI. Az RC (row-column) módban a sorokat, majd az oszlopokat villantom fel 0.15 s-ig. Ez a klasszikus P300 mérésnek felel meg. Az SC-t (single character) választva egyesével villannak fel a karakterek. A GUI frissítése időzített megszakítással történik. A soros port-ról beérkező adatokat 2 ms-onként ellenőrzöm le. Így a 4 ms-onként mért értékek egymásba folyása elkerülhető. Ha a beérkező bájtok száma nem felel meg vagy az első bájtnak nem 1 értékű, akkor a csomag hibás, ezért elvetésre kerül. A feldolgozás során a bájtokból hármassával int típusú változót kell készíteni az előjel megfelelő kiterjesztésével, majd a csv fájloknak megfelelően vesszővel



2.24. ábra. Beágyazott rendszer működése

el kell választani az értékeket. A GUI felső sorában a Stop gombra kattintva véget ér a mérés.

A 2.25 ábra mutatja a beágyazott eszközön elhelyezett alkatrészek összeköttetését és a közöttük zajló információáramlás irányát. Az egész rendszer az STM32F446 mikrovezérlő köré épül fel. A mérési adatokat SPI-on keresztül lehet beolvasni az ADC-ről és ezen keresztül lehet a regiszterek beállítását is induláskor elvégezni. A visszajelző LED szerepe elsősorban programozáskor jelentkezik. A program különböző állapotait lehet vele gyorsan megnézni, ami a fejlesztést gyorsítja. Eldönthető vele, hogy egy adott végtelenciklusban lefutott kódrész megfelelően fut-e, ha a LED-et a lefutás szerint kapcsolgatjuk és a LED árnyalatát figyeljük. Fejlesztés közben ezzel több hibát is megtaláltam. A gomb segítségével a kód írásakor megszakítással egy adott változó értékét változtattam meg. Ez nagyban megkönnyítette, hogy oszcilloszkóppal a megfelelő eseményhez tartozó hullámformát a kommunikációban megfigyelhessem. A kód futás újraindításához vagy megállításához a reset gombot alkalmaztam. A futást többször meg kellett állítanom. A 4 ms-onként kiküldött adattól számtalanszor lefagyott a soros porton érkező adatok megjelenítésére használt Hercules nevű program. Az újraprogramozáshoz és a hibakereséshez kiemelkedően hasznos volt a programozó csatlakozó. Mivel a Bluetooth module USB bemenetet alkalmazva gyorsabban is képes kommunikálni, mint az UART használatakor, ezért egy USB switch-et kellett alkalmazni, amivel az adat áramlásának irányát lehet kiválasztani, így az USB port is használható maradt.



2.25. ábra. Beágyazott rendszer felépítése



2.26. ábra. Jetson Nano Developer Kit [21]

## 2.5. Jetson Developer Kit

A mérési adatokat fel kell dolgozni egy beágyazott rendszeren. Erre a feladatra a Jetson Nano Developer Kit (2.26 ábra) mellett döntöttem a konzulensem javaslatára. Az Nvidia ezt a rendszert neurális hálózatok futtatására fejlesztette ki. Kellően erős képfeldolgozás-hoz is, ezért az ehhez képest jóval kisebb adatmennyiséget eredményező EEG méréseim kiértékelése sem jelent a rendszernek problémát. A beágyazott rendszer 4 GB RAM-mal rendelkezik. Egy Quad-core ARM A57 processzor maximum 1.47 GHz órajelen futtatja a programokat. Ezen kívül még fontos megemlíteni a 128-core Maxwell GPU-t is, mivel a neurális hálózatok tanítása és futtatása ezen történik. GPIO, I2C, I2S, SPI, valamint UART segítségével számtalan az egyszerűbb beágyazott rendszereknél használt eszközt lehet még a rendszerre kötni, de ezekre ebben a projektben nem lesz szükség. HDMI-n vagy display port-on keresztül lehetséges egy képernyőt kötni a rendszerre [21]. Sajnos a Jetson Nano nem támogatja az általános HDMI-VGA átalakítókat, ezért kénytelen voltam beszerezni egy új képernyőt is a rendszerhez.

A Jetson Nano Developer Kit alaphoz nem rendelkezik Bluetooth és Wifi modulokkal, de a M.2 Key E csatlakozójára lehet megfelelő modult találni. A választásom az Intel Dual





**2.27. ábra.** ábra Intel Dual Band Wireless-AC 8265 modul [11]

Band Wireless-AC 8265 modulra esett, mivel ezzel előttem már számtalan ember megoldotta ezt a problémát, így biztos lehettem benne, hogy nincsenek kompatibilitási gondok a két rendszer között. A Jetson Nano-ra a 2.27 ábra bal oldali kivitelében csatlakoztatható az áramkör. Két antenna jár az eszközhöz, amikkel a Wifi 802.11ac és a Bluetooth 4.2 szabványoknak megfelelő kommunikáció megvalósítható [11]. A rendszerhez illesztés probléma nélkül megtörtént. A Jetson Nano kártyát le kell csavarozni a fejlesztői kiegészítő áramkorról és az alatta található csatlakozóba illeszthető a modul. Összeillesztve a rendszert csak a kilógó antennák árulkodnak az új funkció hozzáadásáról. A Jetson operációs rendszere külön driver telepítése nélkül tudta kezelni az eszközt.

Telepítés közben feltűnővé vált, hogy a Fejlesztői kártya hűtőbordája nagyon fel tud melegedni. A hűtőbordán van 4 darab furat, amikhez rögzíthető egy 40x40 mm-es ventilátor. A ventilátor a modul bekapcsolása után a maximális fordulatszámmal üzemel. Így már nem sikerült érezhetően felmelegíteni a hűtőbordát. Kezdetben egy 16 GB-os Class 10 U1 SD kártyára telepítettem fel a rendszert, mert amikor először megnéztem a gyártó honlapját, akkor még ez volt az ajánlott méret. Az SD kártyán a telepítés után már csak 1 GB szabad hely maradt, ami arra sem volt elegendő, hogy a frissítéseket letöltssem a fogyó helyre való figyelmeztetések nélkül. Ezért váltottam egy 64 GB-os Sandisk Extreme Class 10 U3 SD kártyára. Ez a négyszeres méret mellett gyorsabb írási és olvasási sebességet garantál az előző kártyához képest. A U1 osztályú memória kártyák esetén a minimális írási sebesség 10 MB/s, míg az U3-as osztálynál ez az érték 30 MB/s.



## 3. fejezet

# Áttekintés az EEG jelek méréséről és a Deep Learning területéről

A non-invazív módon alkalmazott elektroencefalográfia (EEG) az egyik legelterjedtebb agyi jeleket rögzítő technológia. [31][24] Orvosi területen gyakran alkalmazzák az agy működésének kutatására, továbbá a hétköznapi életben agyi aktivitások és betegségek vizsgálatára. Epilepszia, alvási rendellenességek, addiktív szer - pl. alkohol, drog - használata, érzelmi és fizikai állapot, mind megfigyelhető az EEG segítségével. [31] A BCI rendszerek jelfelvező egységének is gyakori választás, mivel olcsó és kevésbé komplikált a használata, továbbá jól használható valós-idejű rendszerekben a nagy pontosságú időfelbontása miatt. [29][31][24]

### 3.1. EEG technológia

EEG-vel történő jelrögzítés alatt olyan potenciálváltozások mérését értjük, amelyek - a főképpen - agykérgi neuronsejtek összességében keletkeznek. A neuronsejtek aktivációja során gerjesztő vagy gátló posztszinaptikus potenciálok keletkeznek. A mérhető potenciálkülönbségeket a posztszinaptikus potenciálok által - a neuron testek és a neuron ágak között - létrejövő elektromos dipólusok adják. Mivel az EEG elektródái a fejbőrön helyezkednek el, ezért a jelek amplitúdója a  $100 \mu V$ -ig terjedő tartományban mérhető. [29][18] Érezhető, hogy ebben a tartományban rendkívül kényes kérdés a mért jel jel-zaj viszonya, amely azonban az EEG eszköz többi alkatrészének helyes megválasztásával - a jó minőségű jelrögzítéshez szükséges - megfelelő értéken tartható.

Az agyi jelek rögzítése a fejbőrre helyezett elektródákon keresztül történik, ez a felhasználó, vagy páciens számára semmilyen fájdalommal nem jár, sőt még a haj levágása sem szükséges a procedúrához. Az elektródák helyén vagy után a mért jelek erősítve vannak, hogy feldolgozhatók legyenek, majd szűrésen, illetve analóg-digitális átalakításon esnek át. Több típusú rögzítésre alkalmas elektróda is létezik, általában aszerint csoportosítva, hogy kerül-e valamilyen gél vagy paszta az elektróda és a fejbőr közé vagy sem, illetve, hogy az erősítés az elektróda helyén van-e vagy nem. Rendre ezek az elektróda típusok a nedves, száraz, aktív és passzív. Az aktív és nedves elektródák általánosságban jobb jel-zaj viszonytal rendelkeznek, míg a száraz és passzív elektródák valamelyest kényelmesebbek (rendre pl. nincs hátramaradó gél, könnyebb elektródák). [29][31][24] Az elektródák pozícióinak meghatározására több rendszer is létezik, amelyek közül a legelterjedtebb a nemzetközileg standardizált 10-20 rendszer. A rendszer elnevezését azok a százalékosan arányos távolságok - 10% és 20% - adják, amelyek az elektródák helyeit jelölik a fülek és az orr között. Az elektródák elnevezését ebben a rendszerben az adott

elektróda szomszédságában elhelyezkedő agyi terület és az agyfélteke oldala határozza meg. [18][29][24] Az ajánlott távolság az elektródák között 1-2cm-es a SNR alacsony értéken tartása végett. [24] Az elektródák anyagát többnyire valamilyen arany vagy ezüst és ezüst-klorid ötvözet adja. [29]

Az EEG legnagyobb előnye a többi jelrögzítő, illetve képalkotó technológiával szemben, hogy rendkívül jó időbeli - többnyire milliszekundumos - felbontással rendelkezik, ezáltal nagy pontossággal megfigyelhetők az agyi jelek változásai. [31][24] Ennek az előnyös időfelbontásnak is köszönhető, hogy az EEG nagyon jól alkalmazható valós-idejű rendszerekben. Azonban, mivel a mérhető jelek csak több, többé-kevésbé egy időben aktiválódó neuronoterület potenciálváltozásainak együtteseként állnak elő, az EEG térbeli felbontása nem olyan pontos, mint pl. MRI, PET, vagy CT alkalmazása során. [18][24][2] A jelek forrásának pontosabb meghatározásához az elektródákat már az agyszövetbe kell helyezni, mint azt pl. elektrokortikográfia során teszik. [18][24] Többi kihívást jelentenek a zaj mellett megjelenő melléktermékek. A melléktermékek eredete lehet emberi, illetve technológiai. Emberi melléktermékek közé tartoznak a pislogás, izzadás, tudatlan megmozdulás, szívdobogás vagy a szemmozgás által keletkező nagy amplitúdójú zavaró jelkomponensek. A technológia melléktermékek eredete származhat impedancia ingadozásból, kontakthibából, kábelmozgásból vagy felszáradt gél miatt. [29]

Az EEG technológia alkalmazásával több típusú jel is mérhető. A három fő csoport, amelybe oszthatók az önkéntelen (Spontaneous EEG) és a kiváltott potenciálok (EP), illetve az eseményhez köthető szinkronizáció/deszinkronizáció (ERS/ERD). [2][18] Az önkéntelen vagy spontán jelek közé tartoznak az alvás során megfigyelhető, a képzelt motoros, vagy az érzelmi jelek, A kiváltott jelek valamilyen külső vagy belső stimulus hatására keletkeznek. A stimulus külső hatás esetén lehet pl. vizuális vagy audiális. Az általunk is hasznosított P300, egy olyan külső inger hatására keletkező jel, amely a kiváltott jelek eseményhez köthető potenciálok alcsoportjába tartozik. Az eseményhez köthető szinkronizáció/deszinkronizáció szintén külső vagy belső stimulus hatására jelenik meg. [2][31] A mért agyi jelek sávszűrésen esnek át, amely frekvenciasávja - a jel simítása érdekében - 1 és - a lassú frekvenciás melléktermékek kiszűrése miatti - 50Hz közé esik. A szűrési lépés után az analóg jeltől pár száz Hz-es frekvenciával mintavételeznek a digitalizálás során. [24][29]

## 3.2. P300 jelek

A P300 elterjedten használt jel, amely külső vizuális vagy audiális stimulus hatására váltódik ki az agyban. Kommunikációs, illetve irányító BCI rendszerek működtetéséhez gyakori választás a P300. A jelet legjobban amplitúdójának nagysága és amplitúdójának maximális pontjának késleltetésével lehet jellemezni. A P300 amplitúdója a 10-25 $\mu$ V-os tartományban mérhető. a jel csúcsa a külső stimulus megjelenésétől számítva körülbelül 300ms-al később jelenik meg, innen is ered a jel neve. [23] A jel főként az agy centrális frontális területéről indulva halad a centrális parietális terület felé, ezért ezeken a területeken - Fz, Cz, Pz a 10-20-as rendszernél ismertetett elnevezések szerint - a legfontosabb, hogy helyezkedjen elektróda a megfelelő mérhetőség érdekében. [23][2] Két komponensre osztható a jel; a külső inger észlelésével kapcsolatban álló, a frontális területen a stimulus után kb. 250ms-al megjelenő P3a és az információ feldolgozással és memória-folyamatokkal korreláló, a kb. 350ms-tól kezdődő P3b. A kiváltódó jel amplitúdóját és a késedelmét befolyásoló tényezők forrása lehet mind emberi, mind a stimulushoz köthető paraméterekből eredő, ezért a karakterisztika alanyról-alanyra és környezetről-környezetre változik,

megnehezítve annak feldolgozását. [23]

A jel kiváltásához külső inger szükséges, ezért a jel szándékos kiváltásához általában az alanynek valamilyen feladatot kell teljesítenie. Vizuális stimulus esetében ez a feladat lehet képek megkülönböztetése vagy felvillanó karakterek figyelése. Hangi inger esetén csörgő telefon vagy ismerős emberi hang figyelése lehet például feladat. A jel kiváltásához végzett feladatnak egyszerűnek és mentális kevésbé megterhelőnek kell lennie, ellenben a jel amplitúdója csökken, késleltetése pedig megnő. [23] A P300 betűző használata során a felhasználónak megfelelő szintű figyelemmel kell lennie a választott karakter felvillanására. Gyakran a betűző használata során a felhasználónak számolni kell a kívánt karakter felvillanásának számát. A felvillanások számolása fenntartja a figyelmet, illetve növeli az információ feldolgozásához szükséges agyi erőforrások és memóriaigény mértékét, amellet, hogy a számolási feladat általánosságban nem kíván különösebb mentális megerőltetést. [23][25] Az ingerek gyakorisága és előállítása meghatározott paradigma szerint történik.

A stimulusok előállításához leggyakrabban az "oddball" paradigma kerül alkalmazásra. A paradigma alkalmazása során kétféle inger kerül megkülönböztetésre; a céltstimulus és a háttérstimulus. A felhasználónak az agyában történő P300 kiváltásához a célingerre kell fókuszálnia, azzal kapcsolatban kell valamilyen egyszerű feladatot végeznie. A célinger más, a felhasználó számára nem releváns, a célingernél jelentősen sűrűbben előforduló háttéringerek között jelenik meg. A céltstimulus előfordulása történhet véletlenszerűen vagy pontosan meghatározott időközönként. A célingerek között eltelt időközök - azaz inter-stimulus interval (ISI) - hossza egyenes arányos - egy bizonyos mértékig, kb. 8 szekundumig - a kiváltott P300 amplitúdójával, azaz minél ritkábbak a céltstimulusok, annál nagyobb amplitúdójú jel áll elő. [23][25]

A P300 sikeres detektálására az irodalomban az eddigi leggyakoribb megoldás során több célinger által stimulált jelminták kerülnek feldolgozásra és átlagolásra. Erre a feladatra gyakran alkalmaznak lineáris diszkrimináns analízist (LDA), support vector machine-t (SVM), főkomponens analízist (PCA), illetve független komponens analízist (ICA). [2][27] Többszöri jelminta analízására azért van szükség, mert az előbb felsorolt metódusok megfelelő pontossággal csak így képesek elkülöníteni és detektálni a P300 jelkomponenst a mért jelekből. Ez a pontosság gyakran emelkedik 90% fölé, azonban az átlagolás miatt a rendszer, és így a kommunikáció sebességének korlátját a detektálás folyamatának hossza jelenti. [7][27] Azonban a feldolgozás és detekció folyamata gyorsítható egyetlen-próba detekcióval, illetve az egyre jobban elterjed Deep Learning algoritmusok alkalmazásával. Az egyetlen-próba detekció során a rendszer a bemenetére érkező adott hosszúságú - P300 esetén kb. 800-1000ms - adaton azonnali predikciót végez. A Deep Learning területébe tartozó modellek képesek rendkívül komplex problémák gyors és pontos megoldására, így ezek a modellek általánosságban jó választásnak bizonyulnak a P300 észlelésére. [31][27]

Széles a köre azoknak az alkalmazásoknak, amelyek P300 jelet használnak fel irányító agyi jelként. Az agyi jelek felhasználásának nagy előnye, hogy ugyanazon jel, több eltérő típusú feladat elvégzésére is alkalmas lehet. A P300 alkalmas kurzor mozgatásra, kerekesszék irányításra, illetve kommunikációra. A leghatékonyabb P300 alapú kommunikációs eszköz a P300 betűző. [2][25] A készülék általános esetben egy 6x6-os karakter mátrixok tartalmaz, amelyen az angol ABC betűi, illetve számok találhatóak 0-9-ig. A mátrix sorai és oszlopai véletlenszerűen villannak fel egymás után, miközben a felhasználó egy adott betűre vagy számra koncentrálna. Megadott számú villanási iteráció után a rendszer a képernyőn az általa prediktált karakter megjelenítésével ad visszajelzést a felhasználó számára. [25] A betűző használata relatíve egyszerű feladat, általánosságban mindössze 5 percnyi

kalibráció és gyakorlás szükséges a felhasználó által a rendszer megfelelő pontosságú működéséhez. [7]

### 3.3. Deep Learning röviden

A BCI rendszerek dekóder egységében futó jelfeldolgozó algoritmusokat hosszú ideig hagyományosnak mondható eljárások uralták, mint pl. LDA, ICA vagy PCA. Azonban a képfeldolgozás és a természetes nyelvfeldolgozás területén nagyon jól teljesítő gépi tanulási algoritmusok egyre nagyobb népszerűséget szereznek az agyi jelfeldolgozás területén is. A gépi tanulás egy szűkebb csoportját alkotják a mély tanulási algoritmusok. A mély tanuló - továbbiakban Deep Learning (DL) - algoritmusok végtől végig modellek, amelyeknek bemenet(ei), illetve kimenet(ei) vannak, a kettő között pedig különböző transzformációkon keresztüli adatfeldolgozás történik. A modellek alkalmazása során az adat a bemenettől, a hálóstruktúrán át halad a kimenet fele. A DL modelleket több neurális háló réteg sorozata alkotja. Ezen rétegeknek több típusa is létezik. Az általunk alkalmazott modell egy 2 dimenziós konvolúciós neurális rétegekből álló architektúra, így a továbbiakban mi ezen rétegekre és az ezen rétegekből álló modellekre szűkítjük le fogalmainkat. [15]

A konvolúciós rétegeket kiemelkedően alkalmasak lokális minták megtanulására és felismerésére. A konvolúciós rétegek transzformációt hajtanak végre a rajtuk átfolyó adaton. A réteg kimeneti adatai az ún. feature map-ek, amelyek a bemeneti adat által hordozott információk különböző leképzései.

$$\text{conv}(I, K)_{x,y} = \sum_{i=1}^{n_{\text{sor}}} \sum_{j=1}^{n_{\text{oszlop}}} \sum_{k=1}^{n_{\text{melyseg}}} K_{i,j,k} I_{x+i-1,y+j-1,k} \quad (3.1)$$

A leképzéshez egy rétegben a 3.1 szerinti konolúció műveletét alkalmazva több ún. konvolúciós kernel fut végig a réteg bemeneti adatán. A kernelek különböző paraméterezésű szűrők, amelyek előállítják a kimeneten megjelenő más és más információt hordozó feature map-eket. Az adaton végigfutva a kernel alacsonyabb szintű információt, lokális mintákat nyer ki az adatból, így a réteg a bemenetére érkező adatot "építőelemeire" bontja. Képek esetén könnyebb elképzelni a folyamatot, ahol az előálló feature map-ek pl. éleket vagy sarkokat tartalmaznak. Jelen - agyi jelekre történő - alkalmazásunkban a kernelek különböző frekvenciákon nyernek ki lokális mintákat az adatból. A kernelek mérete és paraméterei állíthatók; míg az előbbi általában manuális úton történik állításra, az utóbbi az egész modell tanítása során került hangolásra az ún. backward propagation folyamata során. [16][17] A konvolúciós réteg bemeneti és kimeneti adat dimenziói közötti összefüggést a következő egyenlet adja meg:

$$N_{ki\_sorok} = \frac{N_{be\_sorok} - kernel_{sorok} + 2 * padding}{kernel_{lépés}} + 1$$

$$N_{ki\_oszlopok} = \frac{N_{be\_oszlopok} - kernel_{oszlopok} + 2 * padding}{kernel_{lépés}} + 1 \quad (3.2)$$

$$N_{ki\_mélység} = N_{feature\_map}$$

A neurális rétegek között más funkcionalitással rendelkező rétegek is helyet kapnak. Ezek a rétegek nem-linearitást visznek a modellbe, valamilyen további adatmanipulációt végeznek vagy a tanítás során látnak el a folyamatot segítő vagy robusztusságot növelő funkciót. A DL alap gondolata, hogy a modellek hasonlóan működjenek a valódi neurok alkotta agyi területekhez. Innen ered a struktúra is, miszerint a modell először az alacsonyabb

szintű információkat nyeri ki, majd azokra építkezve, egyre magasabb szintű információkkal bővíti ki tanult ismeretét. Emellett az egyes neuronok működését is másolja a modell. A neuronok az őket érő ingerek hatására aktiválódhatnak. Ez a folyamat gyakran nem írható le lineáris függvény segítségével, így szükséges valamilyen nem-linearitást vinni DL modell architektúrájába. Ezt a nem-linearitást adják az aktivációs függvények, amelyek szintén hozzájárulnak ahhoz, hogy a DL modellek olyan komplex problémák megoldására is képesek, amelyeket egyébként egyáltalán nem, vagy csak nagyon számításigényesen lehetne teljesíteni. Számos aktivációs függvény létezik, így egy adott függvény alkalmazása függ a megoldandó problémától, a használt adathalmaztól és annak típusától, illetve magától az architektúrától. A függvényválasztás egy nem triviális, nehéz feladat, mivel jelentősen befolyásolja a modell teljesítményét és a tanítás idejét. Gyakori probléma a tanítás során az eltűnő-gradiens problémája, amely gyakran az aktivációs függvény esetleg szaturáló hatása miatt jelentkezhet. [14][3] A rendszerünk dekóderében helyet kapó EEGNet az exponenciális-lineáris egység (ELU) aktivációs függvényt alkalmazza. A 3.3-mal megadott ELU függvény a 0-nál nagyobb és kisebb értelmezési tartományon másként viselkedik. Pozitív bemeneti érték esetén az ELU, mint identitásfüggvény viselkedik, így csökkenti az eltűnő-gradiens probléma előfordulásának valószínűségét. Ezzel szemben, negatív értékek esetén, egy  $\alpha$  paraméterrel megadott értéket közelítve szaturál a függvény. Ezzel a megoldással a függvény a bemeneti adat átlagát egy nullához közeli értékhez tolja, amely a tanítás sebességét gyorsítja, továbbá regularizációs hatással van a modellre. [3]

$$f(x) = \begin{cases} x & \text{ha } x > 0 \\ \alpha(\exp(x) - 1) & \text{ha } x \leq 0 \end{cases} \quad (3.3)$$

Klasszifikációs, illetve detekciós problémák esetén a DL modell utolsó rétege után egy szigmoid vagy egy softmax aktivációs függvény biztosítja, hogy a modell kimenetén az egyes megkülönböztetni kívánt osztályokhoz tartozó prediktált - 0 és 1 közötti - valószínűségi változók jelenjenek meg. Jelen alkalmazásunkban a jelek csak egy osztályhoz vannak hozzárendelve - tartalmaz P300 jeleket vagy sem -, ezért a 3.4 képpen megadott softmax függvényt célszerű az utolsó réteg után alkalmazni a kimeneti adatokon.

$$f_i(x) = \frac{\exp^{x_i}}{\sum_{j=1}^J \exp^{x_j}}, \text{ ahol } i = 1, 2, \dots, J \quad (3.4)$$

Egy DL modell kimenetén megjelenő információ jellege és dimenziója alkalmazásfüggő. Detekciós és klasszifikációs problémák során a kimeneten megjelenő adat 0 és 1 között értékeket tartalmaz, pontosan annyi, amennyi osztályt szeretnénk megkülönböztetni az adaton belül. Dolgozatunk során P300 detekciót végzünk, ami egy bináris klasszifikációs probléma - hiszen a modellnek azt kell megmondani, hogy a bemeneti jel tartalmaz-e P300 komponenset vagy sem -, így egy 2 elemű vektor. (Megjegyzés: az adat címkékre one-hot encoding algoritmus alkalmazunk, ezért a 0-ás és 1-es értékekhez [0, 1], illetve [1, 0] vektorok párosulnak. A fogalmak és az eljárás, illetve a tanítás bővebb kifejtésére a szakasz végén kerül sor.) Látható, hogy a nagydimenziójú bemeneti adat a kimeneten már - szegmentációs és lokalizációs problémáktól eltekintve - csak egy vektorként jelenik meg, amely az egyes osztályokhoz tartozó prediktált értékeket tartalmazza. A modellen belül a dimenzió csökkentését - az esetekben konvolúciós rétegeken felül - az ún. pooling rétegek végzik. A pooling réteg a bemeneti adatot egy meghatározott stratégia szerint leskálázza. A választott stratégia egy adott méretű ablakon - szintén kernelnek nevezik, azonban ezen kernel paraméterei nem kerülnek hangolásra a tanítási folyamat során - belül funkcionál. Gyakori választás stratégiaként a max-pooling, illetve az average-pooling, amelyeknél az előbbi esetben az ablakon belüli legnagyobb érték, míg utóbbi esetben az ablakon belüli

értékek átlaga kerül a réteg kimenetére. [6]

Gyakori tanítást segítő és teljesítményt javulást eredményező réteg az ún. batch-norm és a dropout. Mindkét réteg beépítésre került az EEGNet architektúrába. Egy neurális háló tanításának ideje rövidebb és eredményesebb, hogyha az adat normalizált és null-centrált. A modell egyes rétegei között az adatt eloszlása változhat. Ez problémát okoz, mivel az egyes rétegeknek külön-külön alkalmazkodniuk kell az új eloszláshoz, amitől a modell rendkívül érzékennyé válik a különböző paramétereinek változására, továbbá a teszt adat disztribúciójának akár részbeni eltérése a tanító adattól jelentős performancia esést okozhat. A batch-norm réteg az előbb említett problémákat küszöböli ki azzal, hogy megszünteti az adat belső kovariáns eltolódását. A lépés normalizálja a következő réteg bemenetére érkező adatot, így a következő réteg bemenetére érkező adatban nem fordul elő olyan a többitől jelentősen eltérő adat, amely hatására megjelenhet az eltűnő-gradiens problémája. A háló tanítási ideje ezáltal csökkenthető és a modell robusztussága, illetve általánosító képessége nő. [12] A dropout réteg a neurális háló tanítási folyamata során  $p$  valószínűséggel "dob ki" egységeket és hozzá tartozó kapcsolódási pontokat a modelltől, így megelőzve azt, hogy mindig csak adott egységek vegyenek részt a probléma megoldásában. A dropout használata segít megelőzni a túltanulás problémáját - amely a későbbiekben kifejtésre kerül - és alkalmazása után a modell jobb probléma általánosító képességgel rendelkezhet. [28]

Egy DL modell tanítása történhet felügyelt, nem felügyelt vagy részben felügyelt módon. A dekóder fejlesztése során az EEGNet modellt felügyelt módon tanítottuk. A felügyelt stratégia követése során az adathalmaz felcímkézésre kerül, azaz jelen esetben minden adott hosszúságú jelhez hozzárendeljük azt az információt, hogy az tartalmaz-e P300 komponenst vagy sem. A címkék lehetnek számok, betűk, vektorok, azonban tanítás előtt enkódnálni kell őket. Erre a célra népszerű választás a one-hot enkóder, amely tulajdonképpen egy bázist állít a címkék terében és az egyes normált, bázisvektorokat rendeli a hozzájuk tartozó címkékhez. A DL modell tanítására és tesztelésére használt címkézett adathalmaz a tanítás előtt tanító, validációs és teszt alhalmazokra kerül felbontásra. A modell tanítására a tanító adathalmazon kerül sor, míg a validációs halmazzal validálható a modell az egyes epoch-ok - tanítási iterációk - között. A teszt halmazt kizárólagosan a modell tesztelésére szabad használni. A tesztelés eredményei akkor tekinthetők érvényesnek, hogyha a modell számára a teszt halmazban található adatok ismeretlenek, azaz teszt halmazból adat sem a tanítási, sem a validációs halmazokban nem szerepel. A rendelkezésünkre álló adathalmazból tanító, validációs és teszt halmazokat a teljes mérethez viszonyítva rendre 70%, 15% és 15% aránnyal állítottunk elő.

Ahogy az emberek tapasztalatok útján, úgy a DL modellek is a vétett hibáik alapján képesek tanulni, ezért van szükség címkékre a felügyelt tanítás során. A modell a belső paramétereinek - pl. neuronok bemeneteihez tartozó súlyok, kernel paraméterek, stb. - állításával, finomhangolásával képes tanulni. A tanuláshoz szükség van egy olyan függvényre, amely képes megmondani, hogy mennyire volt jó vagy rossz a predikció. Ez a függvény a veszteségi függvény. A tanulási folyamat során a cél, hogy a modell úgy hangolja belső paramétereit, hogy minimalizálja a számára meghatározott veszteségi függvényt. Az EEGNet tanítására használt veszteségi függvény, a más architektúrákban is gyakran szereplő kategorikus kereszt-entrópia, amelyet a 3.5 ad meg, ahol  $l_i$  a bemeneti adathoz tartozó  $i$ . osztály címke,  $p_i$  pedig az  $i$ . prediktált címke. (Megjegyzés: A háló kimenetén mindegyik lehetséges osztály megjelenik egy ahhoz társított valószínűségi értékkel, amely azt mondja el, hogy a modell szerint mekkora valószínűséggel tartozik a bemeneti adat az

adott osztályhoz. A prediktált értékek a valóstól való eltérése kerül összegzésre.)

$$H_{veszteség} = - \sum_i^{N_{osztályok}} l_i \log(p_i) \quad (3.5)$$

Ahhoz, hogy a modell a kimenetén megjelenő predikciós hiba alapján tanulhasson, vissza kell teljeszteni a hibát a modell belső egységei számára. A legelterjedtebb megoldás az ún. backpropagation, amely gradiens alapú tanulást megvalósító algoritmus. Az algoritmus alapja, hogy a veszteségi függvény gradiensét terjeszti vissza a kimenete felől a modell bemenete fele haladva. Az algoritmus népszerűségének egyik legfőbb oka, hogy nem túl számításigényes, mivel alkalmazható a lánc-szabály a gradiens számolása közben. A gradiens terjesztése egy adott egységen keresztül a 3.6 szerint történik, ahol  $X_{n-1}, X_n$  rendre az egység bemeneti és kiemeneti mátrixa,  $W_n$  pedig a súlymátrix. [17]

$$\begin{aligned} \frac{\partial E^P}{\partial W_n} &= \frac{\partial F}{\partial W}(W_n, X_{n-1}) \frac{\partial E^P}{\partial X_n} \\ \frac{\partial E^P}{\partial X_{n-1}} &= \frac{\partial F}{\partial X}(W_n, X_{n-1}) \frac{\partial E^P}{\partial X_n} \end{aligned} \quad (3.6)$$

A modell tanulása során a belső paraméterei minden tanulási iterációban frissítésre kerülnek a backpropagation lépés során. A paraméterek frissítésének célja megtalálni azt a paraméterhalmazt, amely minimalizálja a veszteségi függvényt. A paraméterek frissítését az optimalizáló algoritmus határozza meg. Több típus létezik, azonban mindegyik célja, hogy a modell konvergáljon ahhoz az optimális paraméter halmazhoz, ahol a legkisebb a predikciós hiba. Minden iterációban az optimalizáló algoritmus egy adott konstans - az ún. learning rate - és a veszteségi függvény paraméterek szerinti gradiensét használja fel a paraméterek frissítéséhez. Az EEGNet, akárcsak számos másik modell tanítása során az Adam optimalizáló algoritmus került alkalmazásra. Az Adam gyorsan konvergál az optimum felé és robusztus a gradiens-zajokra vonatkozóan.

$$W_t = W_{t-1} - \alpha \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon} \quad (3.7)$$

A paraméterek frissítésének szabályát a 3.7 adja meg. Az  $\alpha$  adja meg a lépés nagyságát, azonban az Adam előnyei  $\widehat{m}_t$  és  $\widehat{v}_t$  értékekből származik. Ugyanis ez a két paraméter rendre a gradiens becslés átlagát és varianciáját adja meg, az előző időlépésbeni értékek alapján. [13]

## 4. fejezet

# BCI rendszer fejlesztése

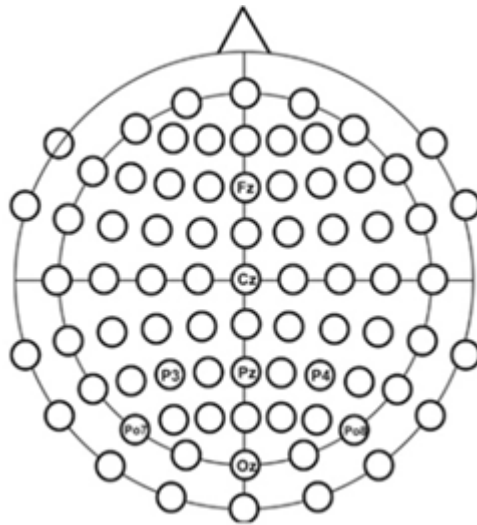
### 4.1. P300 betűző fejlesztése

Az EEG jelek mérésén alapuló rendszerek lehetővé teszik, hogy különböző eszközöket irányítsunk. A mozgásukban korlátozott embereknek lehetőséget biztosíthatunk, hogy képek legyenek számítógépen gépelni, a helyváltoztatásukhoz elengedhetetlen kerekesszéket irányítani. A gépelés lehetővé tételére két rendszer terjedt el a kutatásoknál. A P300 speller-ek táblázatba rendezik a begépelhető karaktereket. A lényeges eltérés a karakterek kiválasztásánál jelentkezik. Az RC speller esetén az oszlopokat, majd a sorokat villantják fel egymás után. A sor és az oszlop kitalálása után egyértelműen meghatározható a kívánt karakter. Az SC speller esetén a táblázatba rendezett karaktereket egymás után villantjuk fel [4]. Ennek a módszernek a hátránya, hogy több idő végig jutni a táblázaton az RC speller-hez képest. A P300 alapú EEG speller-ek esetén is az elterjedt nemzetközi 10-20 elektróda elhelyezési rendszert használják. A 4.1 ábra szemlélteti a legnépszerűbb elektróda elhelyezést. A méréshez 8 csatornát használnak fel. Ez megegyezik az általam tervezett rendszer maximális csatornaszámával.

A GUI fejlesztése során a karakter mátrix kialakításnál a szakirodalomban ismertett kinézetre támaszkodtam. A működésben próbáltam a legjobban megközelíteni a már bevált elrendezésnél alkalmazott felvillanási és a villanások közötti időket [4]. A 4.2 és a 4.3 ábrák mutatják a szakirodalomban megtalálható és az általam fejlesztett grafikus felhasználói felületeket.

A GUI fejlesztéshez a Python nyelvet választottam annak ellenére, hogy jelentős tapasztalatom nincsen vele. Elsősorban az az indok szólt mellette, hogy gyorsan lehet benne haladni, valamint az elkészült program minimális plusz befektetett munkával képes lesz Windows és Linux operációs rendszereken futni. Az USB, Bluetooth eszközök elnevezésében, valamint a mappák elérési útvonalában tapasztalható a legnagyobb különbség az általam használt könyvtárak esetében. A többi kódrésznel nem kellett külön ügyelni a futtató operációs rendszerre. A GUI esetében azért fontos, hogy több rendszeren is jól fusson, mert így a tesztelés könnyebben kivitelezhető a fejlesztés során és később a mérés több eszközön is elvégezhető. A GUI fejlesztésének legnagyobb részét a Windows-t használó laptopomon végeztem, ezért ez egy kényelmi szempont is volt, hogy ne az általam a programozás megkezdése előtt ismeretlen Jetson Nano-n kelljen a kódom nagy részét megírnom. A fejlesztés kezdeti szakaszában, a Jetson Nano megvásárlása előtt, a kódomat egy Raspberry Pi 3B-n teszteltem, mivel ezen is Linux alapú operációs rendszer fut és ez volt az egyetlen ilyen elérhető beágyazott rendszer a tulajdonomban. Azt feltételeztem, hogy a két rendszer kellően hasonló ahhoz, hogy az egyiket tesztelt kód a másik rendszeren is futni fog. A GUI fejlesztése előtt több keretrendszernek is utánanéztem. Többek között a Kivy, PyQt és a TKinter rendszerek voltak a legszimpatikusabbak. A PyQt esetén

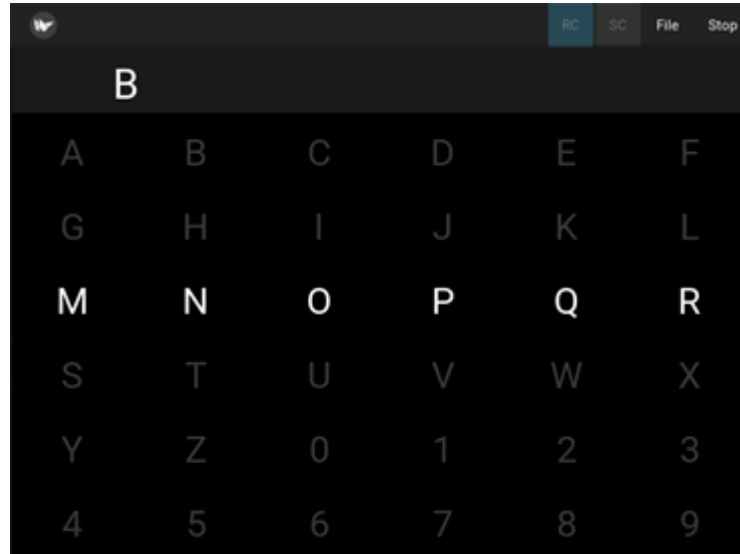




4.1. ábra. Tipikus elektróda elhelyezés a P300 rendszereknél[4]

A	B	C	D	E	F
G	H	I	J	K	L
M	N	O	P	Q	R
S	T	U	V	W	X
Y	Z	0	1	2	3
4	5	6	7	8	9

4.2. ábra. Christopher Guger Et Al. által alkalmazott Speller??



4.3. ábra. Általam fejlesztett Speller GUI

néhány elem nem elérhető az ingyenes verzióban. Mivel még nem használtam soha, ezért nem voltam teljesen biztos benne, hogy az összes olyan funkciója elérhető ingyen, amire a fejlesztés során szükségem lehet, ezért nem ezt választottam. A TKinter-ben és a Kivy-ben készült projektekre rákeresve azt tapasztaltam, hogy a TKinter esetén a programok sokkal inkább a régebben használt egyszerű kinézetűek. A Kivy esetén már a keretrendszer elsajátítását bemutató példánál is egy letisztult, egyszerű, de mégis modern kinézetű felületeket lehetett találni. Mivel nem értek a grafikai tervezéshez, de szeretném, hogy az általam létrehozott program jól nézzen ki, ezért a Kivy mellett döntöttem. A grafikus felhasználói felületem felső sorában a mérés elindítása előtt választhatjuk ki, hogy egy karakter vagy egy egész sor majd oszlop villanjon fel (25. ábra). Alapértelmezetten a sorok és oszlopok villogtatása van kiválasztva. Ezekről balra lehet megadni a mérési fájl mentési helyét. A fájl neve az EEG-vel kezdődik és az aktuális dátumot tartalmazza percre pontosan vagy ha nem csak egy mappát választottunk ki, hanem egy fájlt, akkor az felülírható. Így könnyebb azt biztosítani, hogy a hosszú monoton mérések közben véletlenül ne írjuk felül a már meglévő méréseinket. Az utolsó gombra kattintva elindíthatjuk és leállíthatjuk a mérést. A második sorban az a karakter van megjelenítve, amire gondolni kell a mérés közben. Alatta található a karakter mátrix. Ezeket villantom fel 0.15 s-ra. A mérési adatok után a fájlba még két szám következik. Az első szám a felvillantott sor vagy oszlop azonosítója. A második érték 1, ha a karakter a felvillantott sorban vagy oszlopban van, egyébként 0. Az áramköröm a mért adatokat USB-n és Bluetooth-on keresztül képes átküldeni egy másik rendszerre. A Raspberry-n végzett tesztek alapján egyértelmű lett, hogy az USB jóval megbízhatóbb a Bluetooth-nál. A beérkező byte-okból hármasával int típusú számokat kell visszaállítani és ezeket egy csv fájlba menteni. A folyamat leglassabb része a mérési eredmények string-gé történő összefűzése volt. Erre a problémára a Python-ban találtam több módszert, ami gyorsabb kódot eredményezett, de az eredménnyel nem voltam teljesen megelégedve. Végül rátaláltam a C wrapper függvényekre. A Python-ban van lehetőség C-ben írt függvényeket becsomagolni és Python függvényekként meghívni. A C-ben lehetséges előre lefoglalni a karakterláncnak egy tömböt és azt feltölteni a kívánt adattal, míg a Python a string összefűzéskor mindig új területet foglal le a memóriából, ami lassúvá teszi a kód futását. Ezen kívül a byte-ok összefűzése int típusú változóvá shift-eléssel és bitenkénti logikai művelettel közelebb áll a C nyelvhez, mint a Python-hoz, ezért ezt is el tudom végezni ott. A végeredmény egy olyan Python függvény lett, aminek

átadom a beérkező Byte-okat és az aktuálisan világító sorról vagy oszlopról néhány adatot és az azt a string-et adja vissza, amit ki kell írni a fájlba.

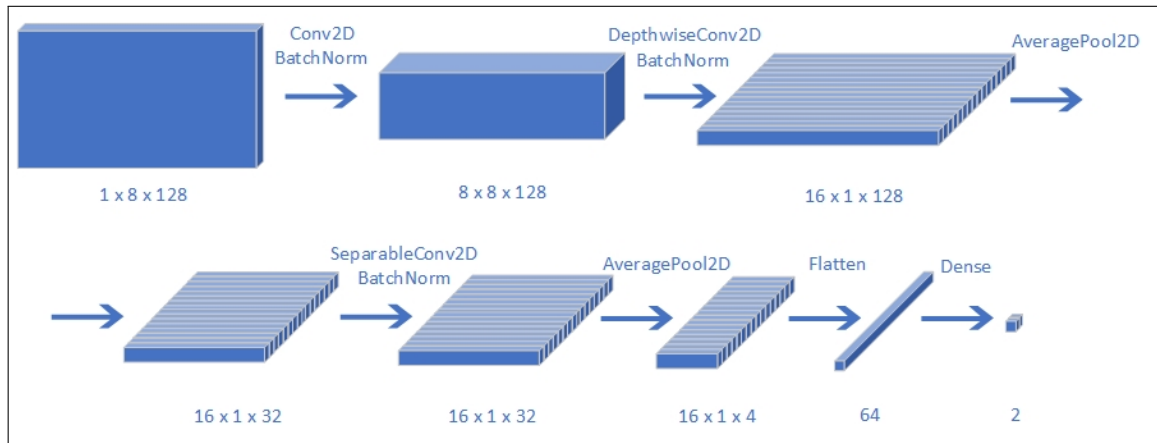
## 4.2. Dekóder fejlesztése

Rendszerünk dekóder egységeként az EEGNet-et választottuk, amely egy state-of-the-art mély tanuló algoritmus. A modellt 2 dimenziós konvolúciós neurális hálórétegek alkotják. Nehézség a feladat során, hogy jelen rendszerünk 8 jel elektródát tartalmaz, amely kis mennyiségű adatot és korlátozott kontextust jelent a P300 jelre és megjelenésének körülményeire vonatkozóan. Azért, hogy a modellt felkészítsük az előbb felsorolt problémákra, olyan különböző eljárásokat - mint pl. dropout, transfer learning - alkalmaztunk, amelyek más mély tanulási alkalmazási területeken is megjelennek. A BCI rendszerek legnagyobb kihívásai közé - a pontos dekódolás és gyors működés mellett - a kalibrálhatóság, illetve a hordozhatóság tartozik. Aképpen, hogy ezen szempontok szerint is megvizsgálhassuk rendszerünket, alany adaton belüli, illetve kereszt-adathalmaz validációt is végeztünk.

### 4.2.1. EEGNet, a state-of-the-art DL architektúra

Jellemző, főleg a képfeldolgozás területén, hogy a megalkotott DL (Deep Learning) architektúrák több száz réteget tartalmaznak és paraméterszámuk  $10^6$ ,  $10^7$ , sőt  $10^8$  nagyságrendben számolhatók. Tanításuk rengeteg adatot és időt igényel. Ezzel szemben az agyi jelek feldolgozására szánt DL modellek jóval kisebb méretűek. Tulajdonképpen nincs is lehetőség olyan méretű architektúrák megépítésére, mint a képfeldolgozás területén, mivel rendkívül korlátozott mennyiségben áll rendelkezésre szabadon elérhető adathalmaz. Nincs lehetőség olyan architektúrák megfelelő betanítására, amelyeknek rétegszáma  $10^2$ -es nagyságrendben van. BCI rendszerek fejlesztése során azonban előfordul még egy korlátozó tényező. Míg a képfeldolgozó algoritmusokat hasznosító alkalmazásokat futtató hardverektől megszoktuk, hogy viszonylag nagy mérettel és fogyasztással rendelkeznek, addig a BCI rendszerekben a minél kisebb méret és a minél kevesebb energiafogyasztás elérése fontos szempont a rendszer használhatóságának tekintetében. Egy BCI rendszertől a hétköznapiakban elvárás, hogy minél kevésbé korlátozza felhasználóját, illetve minél hosszabb időtartamban legyen használható, hiszen gyakran az adott eszköz lehet az egyetlen kapcsolata - pl. kommunikációs csatorna - a felhasználónak a külvilág felé. A nehézség a következő; az agyi jeldekódernek gyorsnak és pontosnak kell lennie, amely szempontok szimultán teljesülése már magukban hordozzák a kihívást, azonban emellett, viszonylag alacsony hardverigénnyel kell, hogy rendelkezzen és előnyös, hogyha ugyanazon rendszer több típusú probléma megoldására is képes. Jelen tudomásunk szerint az EEGNet az a 2D konvolúciós DL modell, amely képes az előbb említett szempontok szimultán teljesítésére, megfelelő szinten.

Az EEGNet architektúrája két fő blokkra bontható. Az első blokk egy sima 2D és egy 2D depthwise (mélység menti) konvolúciós réteget tartalmaz, továbbá mindkét konvolúciós után egy-egy batch-norm réteg, míg a blokk végén egy ELU aktivációs függvényt tartalmazó és egy 2D átlagoló pooling réteg helyezkedik el. A sima konvolúciós réteg által előállított feature map-ek különböző frekvenciákon sávszűrt variánsai a bemeneti jelnek. A depthwise réteg egy speciális konvolúciós réteg, amelynek feladata frekvencia-specifikus térbeli szűrők kinyerése. A réteg különlegessége abban rejlik, hogy a konvolúció műveletét az egyes feature map-eken külön-külön végzi el. A megoldás előnye, hogy az architektúra paraméterszáma így kevesebb, mintha sima konvolúciós réteget tartalmazna a depthwise helyett. A kevesebb paraméter gyorsabb taníthatóságot és kis méretű adathalmazokhoz jobb kompatibilitást jelent. Emellett a modell mentett



**4.4. ábra.** EEGNet architektúra, amely bemenetére 8 csatornán rögzített 128 Hz-en mintavételezett jelet vár, kimenetén pedig egy *one – hot* enkódolt (0: [0 1], 1: [1 0]) osztály predikció jelenik meg

súlyfájlja, amely a végső dekóder hardverre kerül, kisebb helyigényű. A második blokk szintén egy speciális konvolúciós réteget tartalmaz, amely után egy batch-norm, egy ELU-t tartalmazó és egy 2D átlagoló pooling réteg van. A speciális konvolúciós réteg egy 2D separable réteg, amely hasonlóan a depthwise-hoz, szintén külön-külön dolgozza fel a bemeneti feature map-eket, azonban a feldolgozás után még egy pontonkénti konvolúciót végez el. A pontonkénti konvolúció során egy  $1 \times 1$ -es kernel halad végig egyszerre az összes feature map-en, így az eredmény a feature map-ek pontonkénti szummáinak variációi. Ezzel a lépéssel az EEGNet az tanulja meg, hogy milyen súlyozással kombinálja össze az előálló feature map-eket. Ez azért is előnyös, mert a különböző agyi jelek, így az előálló különböző feature map-ek más frekvenciákon, illetve időbeli skálán hordozzák a bemeneti jelből kinyert információt. Mindkét blokk végén még egy dropout kapott helyett a modell regularizálása és az túltanulás elkerülése céljából. A háló kimenetén egy softmax aktivációs függvény transzformálja a prediktált eredményt valószínűségi értékekké.

#### 4.2.2. Adathalmazok

Az EEGNet tanításához és validálásához több adathalmazt is felhasználtunk. Mindegyik EEG technológiával került rögzítésre P300 jel vizsgálatának céljából, azonban a csatornaszám, a digitalizáláshoz választott mintavételi frekvencia és a sávszűrés határfrekvenciái eltérőek. Mindhárom esetben eltérő feladatot kellett a pácienseknek végrehajtaniuk a P300 kiváltása érdekében, azonban az oddball paradigma szerint történtek a kiváltó események irányítása. Az adathalmazok feldolgozáson estek át, amely során sávszűrést és egyes esetekben újra-mintavételezést alkalmaztunk.

A VEPESS adathalmaz egy Biosemi Active 2 EEG készülék segítségével, 18 személy fején került rögzítésre. 64 agyi jel és 6 referencia csatornát tartalmaz. A P300 kiváltáshoz vizuális stimulust alkalmaztak, amely során az alanyok bináris döntést kellett hozniuk. Az alany számára szövetséges és ellenséges katonák képei kerültek prezentálásra, amelyek alapján két gomb közül a megfelelőt kellett megnyomni. A felhasználó fő feladata az ellenséges katona képének azonosítása volt a szövetségesek közt, így a céleseményként az ellenséges katona képe jelentette, míg a szövetségeseké adta a háttérstimulusokat. Több esemény is címkézésre került az adatbázisban, azonban számunkra csak a 34-es és 35-ös

címkével ellátottak a relevánsak, ugyanis ezek tartoznak rendre a cél- és háttérstimulusokhoz. Az adathalmaz újra lett mintavételezve 128 Hz-es frekvencián, majd egy 1-40 Hz közötti sávszűrővel került újraszűrésre, hogy a - többnyire lassú-frekvenciás - zavaró melléktermékek kiszűrésre kerüljenek. Két alany adat kizárásra került hiányos jelek miatt. Végző lépésként 1 szekundum hosszúságú minták kerültek kivágásra a halmazból, amelyeknek kezdő időpontját a két stimulus közül bármelyik megjelenése jelentette. A kivágott jelminták átfedhetik egymást időben. Az adathalmaz több háttérstimulushoz köthető jelmintát tartalmaz, így tanítás során ezt a tényt figyelembe kellett venni. [26]

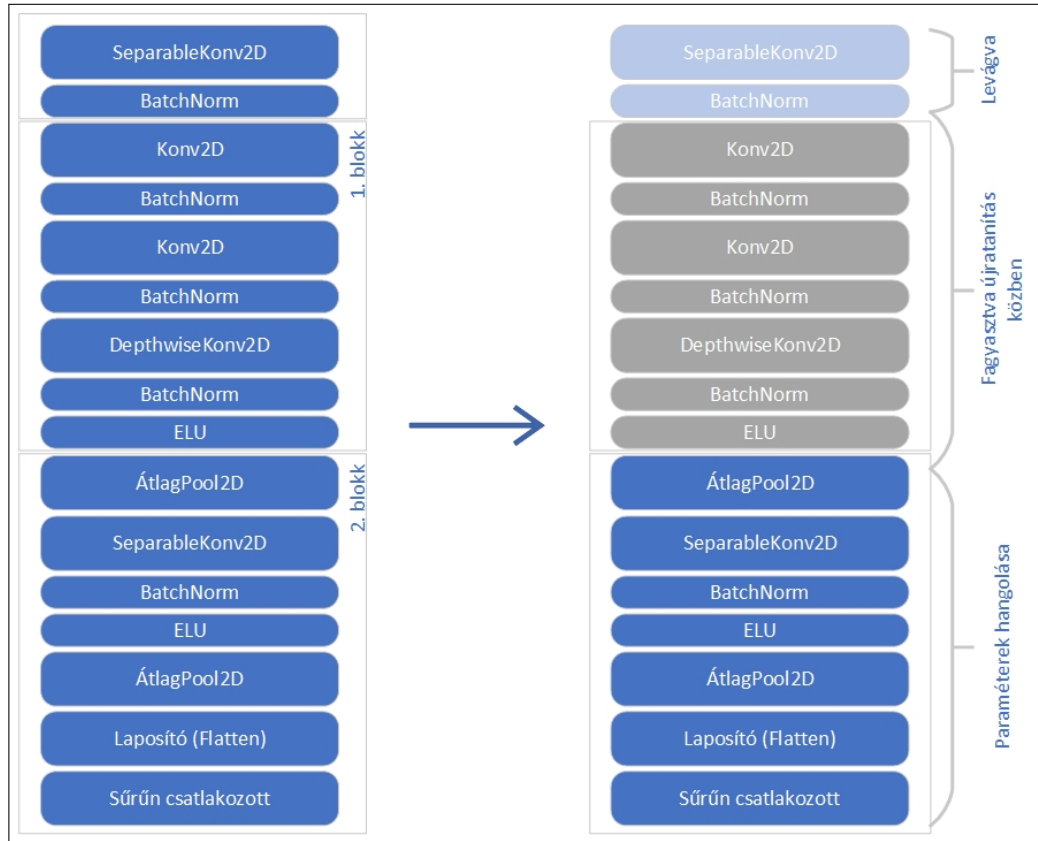
A BCI Competition 3 2-es adathalmaza szintén vizuális stimulus hatására kiváltódó P300 jeleket tartalmaz. 2 alanytól származó 64 agyi jelcsatornából áll. Az alanyoknak az adathalmaz rögzítése során egy - már a korábbi fejezetek során ismertett - P300 betűzőt kellett használniuk. A halmaz melléktermékektől mentesített, 0.1-60 Hz között szűrt és 240 Hz-en digitalizált. A jelek 128 Hz-re kerültek leskálázva, továbbá 1-40 Hz között sávszűrve. A P300 karakterisztikája miatt a külső stimulus megjelenésétől számítva 1 szekundumos időtartamban kerültek az adatok feldarabolásra. A kivágott jelminták átfedhetik egymást időben. Az adathalmaz több háttérstimulushoz köthető jelmintát tartalmaz. [30]

Harmadik adathalmazként a [7] tanulmányból származó P300 jeleket tartalmazó gyűjteményt használtuk fel. Elérése a [1] link követésével és az  $s_1$  kicserélésével  $s_2$ -n keresztül egészen  $s_{10}$ -ig. A tanulmányban 100 alany vett részt, azonban elérhetővé már csak tíz alanyé került. A halmaz és 8 agyi, illetve 1 referencia jelcsatornát tartalmaz. Az alanyok feladata egy P300 betűző készülék használata volt, amelynek segítségével 5 percnyi kalibráció után a LUCAS nevet kellett lebetűzniük, miközben számolták az egyes kívánt karakterek felvillanásainak számát. A jelrögzítés során kétfajta karaktervillantási stratégiát is követtek. Az egyik esetben sorokat és oszlopokat, míg a másikban csak az egy adott karaktert villantottak véletlenszerűen egymást követően. A tíz .mat kiterjesztésű fájlból kettő a karaktervillantásos stratégia követése során mért jeleket tartalmazza. A jel analóg-digitális átalakítására 256 Hz-en került sor, illetve előre meg lett szűrve egy 0.5-30 Hz-es sávszűrővel. A feldolgozás során mi 1 szekundumos szeleteket vágunk ki a halmazból, ugyanúgy egymást átlapolva, mint az előző két esetben is. Az adathalmaz az előző kettőhöz hasonlóan több P300 komponens nélküli jelet tartalmaz.

### 4.2.3. Transfer Learning

Deep Learning modellek megfelelő módon történő tanítására nagy mennyiségű adat szükséges. Amíg képfeldolgozás területén nagy méretű és sok objektumosztályt magukba foglaló adathalmazok állnak rendelkezésre, addig az EEG alapú BCI rendszerek fejlesztéséhez elérhető mennyiségekre ez nem igaz. Ugyan nem szükséges olyan mélységű architektúrák megalkotása, mint képfeldolgozás területén, azonban az agyi jelek karakterisztikáinak nagy mértékű szórásainak megtanulása nehézséget jelent a modellek számára, amely probléma kezeléséhez a modellnek számos jelet kell látnia. A Transfer Learning alkalmazása részben megoldást jelent az adathiány problémájára.

Transfer Learning alkalmazásával lehetőség van modellek tudásának bővítésére, tanításának gyorsítására, illetve teljesítményének növelésére. Számos módon, illetve stratégiát követve lehet Transfer Learning eljárást alkalmazni. Fő célja, hogy egy modell által korábban megszerzett tudás újra felhasználható legyen. Képfeldolgozás során, ha adott egy modell, amely képes pl. éttermeket felismerni és a cél az az, hogy ugyanazon



4.5. ábra. Transfer Learning alkalmazása az EEGNet architektúrára

modell képes legyen kávézókat is felismerni, akkor nem szükséges teljesen előről betanítani egy hálót, hanem felhasználható az étterem klasszifikáló modell. Ezt teszi lehetővé, hogy egy DL modell a bemenetétől, a kimenete fel haladva rendre alsóbb, majd egyre magasabb szintű és komplexitású információt nyer ki az adatból. A kávézók felismeréséhez ugyanúgy az élek, a sarkok, az ívek és egyéb más információkon keresztül vezet az út, ám ezeket az adatokat a másik, már előre tanított modell megtanulta, így elég, hogyha csak a kávézó specifikus jellemzőket tanulja meg az új háló. Mivel a modellnek Transfer Learning során már csak az új adat-specifikus információk kinyerését kell sikeresen teljesítenie, ezért kevesebb paraméter tanítása szükséges. A kevesebb paraméter gyorsabb tanulást és kisebb mennyiségű adat szükségletét vonzza maga után. A DL EEG specifikus területén jelenleg a szabadon elérhető adathalmazok száma jelentősen elmarad a képi adatbázisok számától. További nehézség, hogy az elérhető halmazok általában kevés alanytól származó és relatíve rövid időtartamú mérések eredményeit tartalmazzák, gyakran zaj- és melléktermék-szűretlenül. Az EEG alapú BCI rendszerek fejlesztése során tehát, két különböző cél is vezérelheti a Transfer Learning alkalmazását; az egyik, hogy adott típusú jeleket tartalmazó adathalmazokon egy jól teljesítő modell tanítása váljon lehetségessé, vagy egy létező adott feladatot elvégző modell betanítása egy másik probléma elvégzésére és így a modell alkalmazhatósági területeinek számának növelése.

Számos eljárási mód szerint lehet alkalmazni Transfer Learning-et, azonban a dolgozat során azon stratégiát mutatjuk be, amely általunk is alkalmazásra került. Első lépésként szükség van egy már előre betanított modellre, amely vagy rendelkezésre áll, vagy előtanítás során áll elő. Egy DL modell a belső paramétereinek és súlyainak hango-

lásával tanul a bemenetén érkező adatokból. A betanított modell belső paramétereinek mentésével és betöltésével lehetséges a megszerzett tudás hordozására, amennyiben az új modell rétegei, amelyekbe a súlyok betöltésre kerülnek, megegyező paraméterekkel rendelkeznek. A tudáshordozás után a modell tanítás finomhangolás nélkül is azonnal tesztelhető, azonban az új probléma-specifikus információk kinyeréséhez célszerű a modell újratanítása, azaz finomhangolása. A finomhangolás során az architektúra belső paraméterei ugyanúgy frissülnek, azonban a jobb kiindulási, inicializálási értékek miatt gyorsabban, akár pontosabban tanulhat a modell, mivel a hordozott súlyok közelebb lehetnek az optimumhoz. Amennyiben megfelelő ismeret áll rendelkezésre a háló belső működéséről, lehetőség van a rétegek fagyasztására. Fagyasztott rétegek súlyai nem kerülnek frissítésre a finomhangolás során, így a tanítási folyamat gyorsabbá válik. Célszerű azon rétegek fagyasztása, amelyek alsóbb szintű információkat tartalmaznak az adatról, mivel minél feljebb haladva az egyre komplexebb információk egyre inkább feladat-specifikusak.

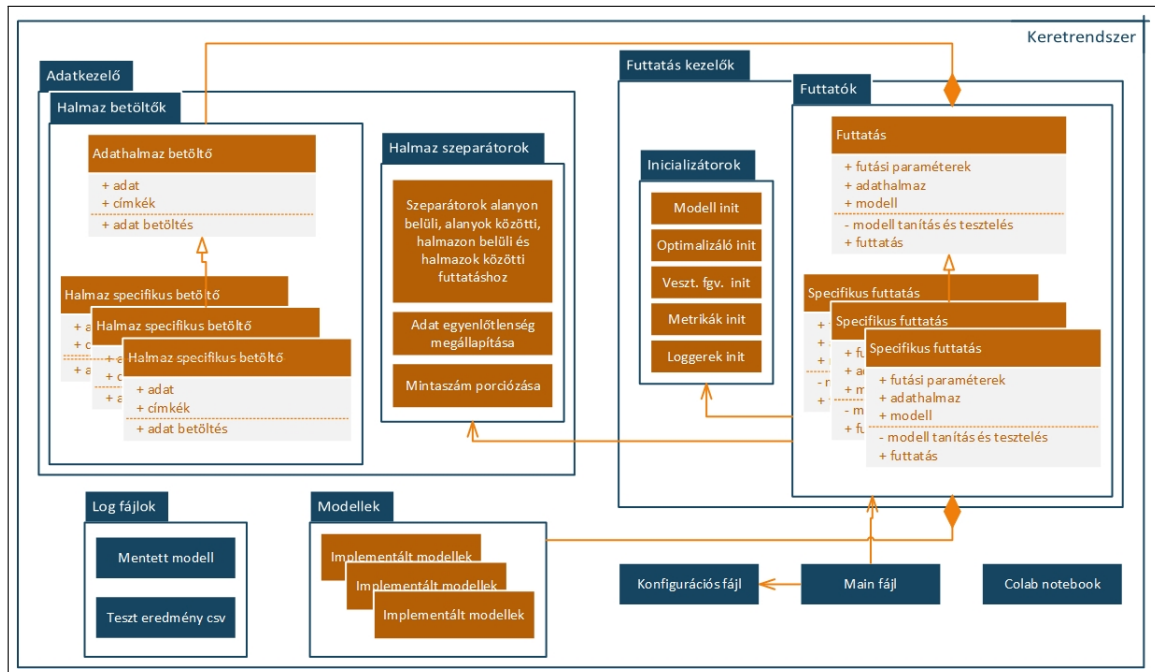
A dolgozat során három adathalmazt használtunk fel, amelyből az egyik - akárcsak az általunk fejlesztett EEG eszköz - 8 agyi jel csatornát tartalmaz. Az EEGNet architektúra felépítése miatt nem oldható meg, hogy a modell eltérő csatornaszámú bemeneti adattal tanítsuk. Miként végső eredményként egy olyan modellt kívántunk használni, amely 8 csatornás jelet képes feldolgozni, az eredeti EEGNet architektúra részben módosításra került. A modell első általános konvolúciós rétege elé két további konvolúciós réteg került hozzáadásra. Az első réteg egy Separable 2 dimenziós konvolúciós réteg, amely a bemeneti adatot egyszerre több csatorna kombinációjaként előálló 8 csatornás adatra skálázza le, miközben 64 darab feature map-et állít elő. A következő réteg egy sima konvolúciós réteg, amely a 64 feature map-et egy  $1 \times 8 \times 128$ -as adatra képi le. Az architektúra ezutáni része megegyezik az eredeti EEGNet-ével. A hozzáadott két réteg biztosítja, hogy a bemeneti adat úgy kerüljön leképzésre 8 csatornára, hogy közben a minden csatorna adata felhasználásra kerüljön és így minél kisebb mértékű legyen az információvesztés. A kiegészített modell tanítása után a hozzáadott két réteg levágásra kerül, így már a 8 csatornás jelet képes az eredeti EEGNet feldolgozni, előtanított paraméterekkel. Az újratanítás során kísérleteztünk rétegek fagyasztásával. Az első konvolúciós réteg, illetve az egész első blokk fagyasztását vizsgáltuk meg. A modell első fele a P300 jel karakterisztikájára jellemző információkat nyeri ki, ezért az a kérdés vezérelte kísérleteinket, hogy egy teljesen más halmaz kalibrálásához elegendő-e a második blokk finomhangolása.

A hozzáadott két konvolúciós réteg hatását a modellre és a tanításra, továbbá a Transfer Learning és rétegfagyasztás eredményességét a 5-ben ismertetjük.

#### 4.2.4. Keretrendszer a modell tanításához és teszteléséhez

A dekóder fejlesztéséhez, a használt adathalmazok feldolgozásához, az EEGNet tanításához és teszteléséhez egy kisebb keretrendszer kiépítésére volt szükség. A szoftver Python nyelven került implementálásra. A rendszer három fő csomagok foglal magába, amelyen túl egy Main, egy konfigurációs és egy notebook fájl található meg.

Az adatok betöltését, kezelését és szétosztását végző kódok egy adatkezelő csomag alá kerültek összegyűjtésre. Az egyes halmazok betöltését és szűrését az `ősosztály`-ból leszármazó, az adott halmazra specializált osztályok végzik. Az adathalmazok `.set`, illetve `.mat` kiterjesztésű fájlokban vannak tárolva. A `scipy.io` csomag `loadmat()` függvénye segítségével könnyedén betölthetők az adatok. Betöltés után az adatokat az `mne.filter` csomag metódusai segítségével szűri meg a betöltő osztály. A modell tanításához szükség



4.6. ábra. Python nyelven épített keretrendszer DL modellek tanítására és tesztelésére

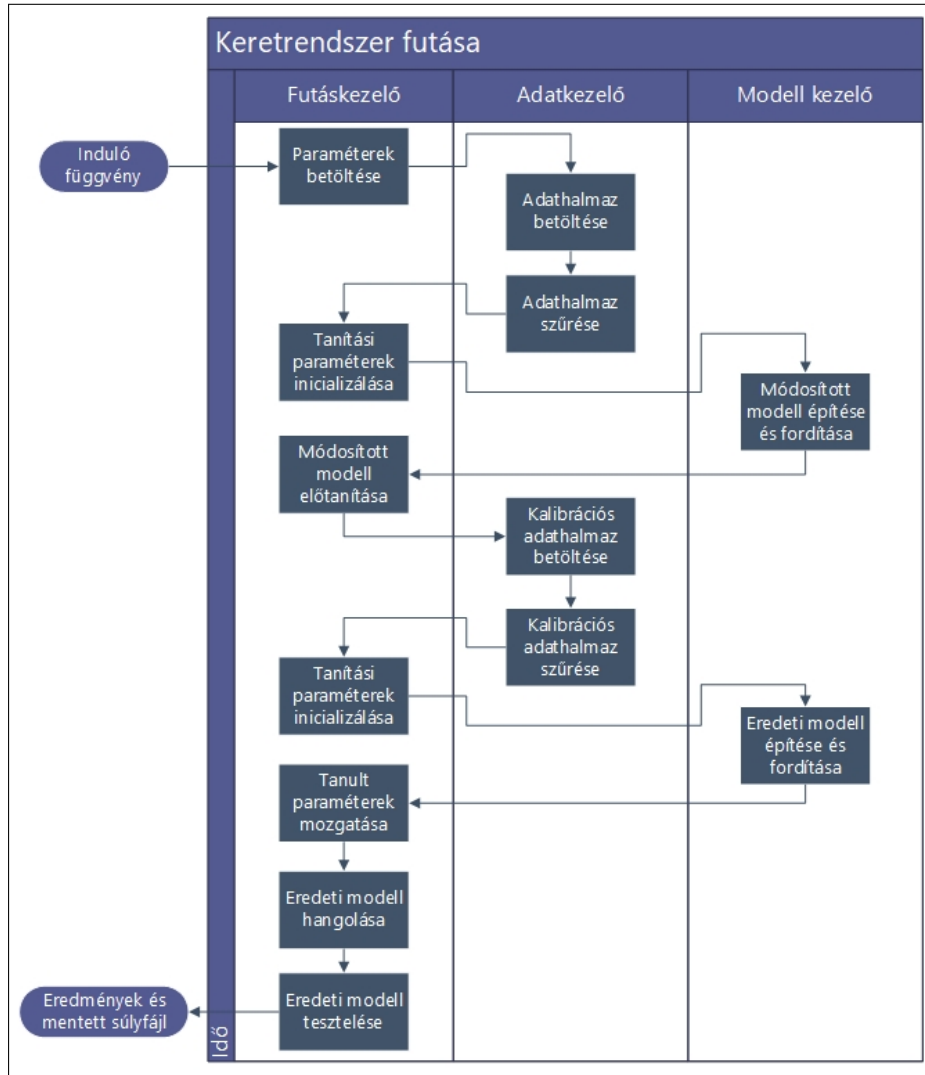
van tanító és teszt halmazokra. Ezen halmazok előállítását az adathalmaz szeparátor metódusok végzik a kívánt validációs stratégiának megfelelően. Ezen kívül az egyes osztályok egymáshoz viszonyított arányainak kiszámítását végző függvény is itt került implementálásra.

A különböző modellek - a módosított és eredeti EEGNet architektúrák - szintén egy csomagba kerültek összegyűjtésre. Mindegyik modell építése, tanítása és tesztelése a Tensorflow és a Keras API párosának segítségével történt. A tanítás és tesztelés GPU-n történt. A futtatások során metrikák - pl. veszteség, pontosság, AUC - segítettek a folyamat figyelését és ellenőrzését, amelyek rögzítéséhez a Tensorboard alkalmazást használtuk. Egy külön *callback* metódus segítségével a tanítás során mindig mentésre kerültek a legjobb validációs eredményt elérő modellparaméterek.

A futtatás kezelő csomagba az inicializátorok és a futtató osztályok kerültek. A tanításhoz szükséges optimalizáló algoritmust, a veszteségi függvényt, magát a modellt és a *callback*-eket inicializáló függvények egy helyen kerültek implementálásra. A sok paraméter miatt, illetve korábbi tapasztalatok alapján célszerű egy konfigurációs fájlba összegyűjteni azokat a adatokat, amelyek alapján a tanítás és tesztelés történik. Ilyen adatok pl. a modell neve, a használt adathalmaz neve és elérése, az optimalizáló függvény paraméterei, illetve a metrikák. Az inicializáló függvények szerepe, hogy ezen adatok alapján hozzák létre a kívánt metódusokat és objektumokat. A futtató osztályok egy közös ősztyárból származnak le, specifikusan egy adott tanítási-tesztelési stratégiának megfelelően. A futtatást kezelő osztály tartalmazza az adathalmazt, a modellt, illetve a felhasznált egyéb adatokat. Az osztályok a tanítást és tesztelést az adott felhasználásnak megfelelően implementálják.

A futtatásokhoz GPU erőforrásra volt szükség, így egy külön, Google felhőben futtatható *Jupyter notebook* került létrehozásra, amely tartalmazza az egész keretrendszer





4.7. ábra. Egy modell tanításának és tesztelésének menete Transfer Learning alkalmazásával

kódját. A *notebook* a Google Colab webes felhőben futtatható nagy teljesítményű GPU-kon, *Linux* operációs rendszer alatt.

## 5. fejezet

# A fejlesztett rendszer tesztelése és eredmények

### 5.1. Rendszerteszt

A rendszert kezdetben az ADS1299-es IC saját négyszögjel generátorával teszteltük, ami  $\pm 1$  mV feszültséget kapcsol az ADC-k bemeneteire. Az IC az adatlapján leírt értékeknek megfelelően mért. Másodpercenként 250 minát vehetünk a csatornákról a leglassabb mintavételezési sebességgel. A minták kiolvashatóak, ha az IC a data ready lábán jelzést ad. A jelzések között eltelt idő oszcilloszkóppal megmérve a mintavételezés frekvenciája megfelelt az elvárt értéknek. Az alkalmazás kimeneti fájljában a felvillanás ideje és a rögzített adatpontok száma alapján megvizsgáltuk, hogy megfelelő számú minta érkezett-e be a rögzítés során. Ebből megállapítottuk, hogy a vizsgált 5 intervallumban nem történt adatvesztés.

### 5.2. Végleges eredmények

Az eredmények jóságának megállapíthatóságának érdekében először az adathalmazokon belüli, közötti és az alanyok adatain belüli kísérletek során elért értékeket prezentáljuk. Ezután a modell módosításának hatását mutatjuk be, végül pedig a Transfer Learning alkalmazása során előállt értékeket adjuk meg.

Ahhoz, hogy a modell több, eltérő csatornaszámú adathalmazt is képes legyen feldolgozni, két további konvolúciós réteget kapott kiegészítésként. Annak vizsgálatára, hogy ez milyen hatással van a teljesítményre több futtatást végeztünk, majd ezeknek az eredményeit átlagoltuk. A vizsgálatot mindhárom adathalmazra elvégeztük, hogy a későbbiekben viszonyítási alapként szolgálhassanak ezen értékek. A futások eredményeit a 5.1 tábla foglalja össze. Mindkét adathalmaz esetében az eredmények alapján a módosított és az eredeti EEGNet modell közel azonos eredményt hozott. A BNCI Horizon szett esetében nem tudunk módosított modellről beszélni, mivel a csatornaszáma megegyezik az eredeti architektúra bemenetén várt csatornaszámmal. A halmazok közötti performanciakülönbséget az eltérő rögzítési mód, körülmények, környezet, illetve utófeldolgozás okozza.

A BNCI Horizon adathalmaz akárcsak az általunk rögzített, 8 csatornán tartalmaz agyjeleket, ezért a Transfer Learning hatását első sorban ezen a halmazon teszteltük. A modellt az előtanítás során a VEPSS vagy a BCIC3D2-es szetre tanítottuk be, majd azt a BNCI halmazon kalibráltuk és teszteltük. Az eredményeket a 5.2 tábla foglalja magába. Továbbá a táblában feltüntetjük az első blokk fagyasztásának hatását a teljesítményre.

	Módosítás nélkül		Módosított	
	Veszteség	Pontosság	Veszteség	Pontosság
Adathalmaz				
VEPESS	0.22	93%	0.24	91%
BCIC3D2	0.5	75%	0.5	76%
BNCI Hori- zon	0.53	75%	-	-

**5.1. táblázat.** Módosítás nélküli és módosított EEGNet architektúrával történő futtatások átlagolt eredményei

	25%-os kalibráció		75%-os kalibráció	
	Veszteség	Pontosság	Veszteség	Pontosság
Előtanító adathalmaz				
VEPESS	0.47	80%	0.48	79%
VEPESS fa- gyasztással	0.53	80%	0.52	80%
BCIC3D2	0.47	80%	0.47	82%
BCIC3D2 fa- gyasztással	0.59	73%	0.56	82%

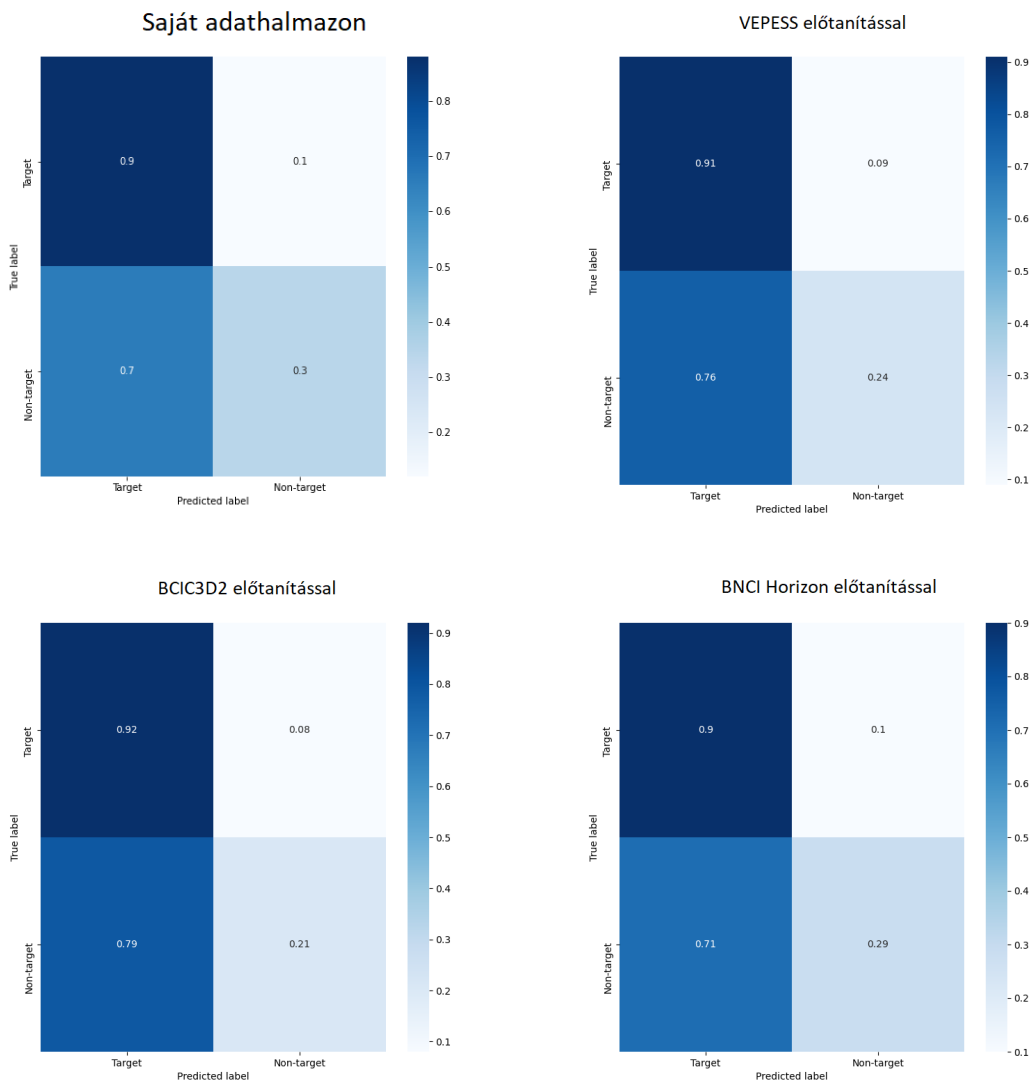
**5.2. táblázat.** Transfer Learning alkalmazása. Tanító adatnak a VEPESS, illetve a BCIC3D2 halmaz került választásra, míg a tesztelést a BNCI Horizon szetten végeztük.

Kalibrációs adatmennyiségként egy adott alany adatának 25%, illetve 75%-át használtuk fel, ezzel azt vizsgálva, hogy a finomhangolásra felhasznált adatmennyiség miként befolyásolja a modell pontosságát. Az eredményekből az látszik, hogy az előtanítás képes volt javítani a modell teljesítményén, azonban a rétegek fagyasztása nagyobb veszteséggel és esetekben predikciós pontosság romlásával járt.

A fejlesztett EEG eszköz által 2x15 percnyi mérés került rögzítésre és felhasználásra tanítás, illetve tesztelés céljával. Az adat szűrése után két fajta futtatást végeztünk vele. Az egyik esetben semmilyen előtanítást nem végeztünk az EEGNet-en, míg a másodikban Transfer Learning-et alkalmaztunk. Első megközelítésben arra lettünk figyelmesek, hogy a tanítás során pontossági értékek jelentősen fluktuálnak, továbbá a tanítás eredményét befolyásolta, hogyha megkevertük az adatokat. Arra következtetésre jutottunk, hogy az adathalmazban sok olyan jelfordul elő, ami olyan szinten zajterhelt, hogy nem használható fel. A zajterhelt jelek közül kizártuk a halmazból azokat, amelyeket meg tudtunk vizsgálni és felhasználhatatlannak bizonyult. Ezen lépések után egyértelműen látszott az értékek stabilizálódása. Az átlagos veszteség és pontosság értékei a 5.3 táblában láthatók. A táblázat mellett megadjuk a tévesztési mátrixot is (5.1), amely a modell által prediktált és a valódi osztályok közötti kapcsolatot mutatja meg. A kalibrált modell **paraméterszáma 1442 db, súlyfájljának mérete pedig 41 kByte**, ami beágyazott rendszerek esetében sem jelent nagy helyigényt.

Előtanító adathalmaz	Veszteség	Pontosság
Saját mérés	0.6	74%
VEPESS	0.62	76%
BCIC3D2	0.6	77%
BNCI Horizon	0.65	78%

**5.3. táblázat.** Az EEGNet teljesítménye a saját adathalmazon. A modell előtanítása eredményjavulást eredményezett.



**5.1. ábra.** Az egyes tesztelesekhez tartozó tévesztési mátrixok. A mátrixok között nagy különbségek nem jelennek meg. Kis mértékben látható, hogy a BCIC3D2 és VEPESS adathalmazokon nagyobb a Target eltalálása, azonban ezzel egyidőben több a hamis Target érték is. Ezen két adathalmaz esetében a modell némileg bátrabban prediktált Target értékeket, míg a BNCI Horizon esetében visszafogottabban. A BNCI Horizon esetében megjelenő némi visszafogottság a valós Non-target értékek jobb prediktálásával járt, így a modell pontosságának értéke is nagyobb, ami a 5.3 megjelenik.

## 6. fejezet

# Konklúziók és tervek a rendszer továbbfejlesztésére

### 6.1. Összefoglalás

A dolgozat során egy P300 betűző BCI rendszer került megvalósításra. A betűző működéséhez egy saját, 8 csatornás EEG eszközt realizáltunk, továbbá egy saját GUI lett fejlesztve a vezérléshez. A rendszert futtató hardver egy NVIDIA Jetson Nano eszköz, amelyen az EEGNet 2 dimenziós konvolúciós mély neurális háló dekódolja a rögzített agyjeleket. Az EEGNet architektúrát több szempont szerint is megvizsgáltuk, illetve Transfer Learning segítségével tettük robusztusabbá. A rendszer működőképességét előben is megvizsgáltuk.

Az EEGNet-tel való kísérletezés eredményeként a megfigyelésünk, hogy a Transfer Learning egy jó alternatívát tud nyújtani, amennyiben kevés adat áll rendelkezésre, azonban az eljárás mikéntjének meghatározásához a modell alapos ismerete és kísérletek elvégzése szükséges. További nehézség, hogy az elérhető adathalmazok túlnyomó többsége előre szűrt és melléktermékektől mentes, így ezen halmazok segítségével nem készíthető fel elegendő mértékben egy DL modell a valós-idejű működésre.

### 6.2. Továbbfejlesztési tervek

Az áramkör alkalmazhatóságát nagyban növelné, ha biztonságosan lehetne mérésre használni az USB-n keresztül is és nem csak a Bluetooth kapcsolat alkalmazásával. A galvanikus leválasztás megnövelné az áramkör helyigényét, de egy értékes új funkcióval bővítené a rendszert az USB megbízhatósága miatt. Ehhez egy digitális izolátor IC szükséges az adatvonalak leválasztásához és egy DC/DC koverter a tápellátás izolálásához.

Fejlesztés közben az ADS1299-es IC-t és a mérés pontos menetét jobban megismerve kiderült, hogy nem szükséges az áramkörre 19 db EEG csatlakozót forrasztani, mivel a belső erősítők invertáló kimenete az IC multiplexerével az SRB1 lábra köthetők. Ezzel 8 csatlakozóval kevesebb beszerzése is elegendő lehet. Az SRB2 lábat az IC multiplexerével a műveleti erősítők nem-invertáló kimeneteire lehet kapcsolni, ami nem feltétlenül szükséges funkció, ezért az ehhez tartozó csatlakozó is kihagyható. Tehát összesen 9 csatlakozóval kevesebb is elegendő, ami pont megegyezik az áramkör hosszabbik oldala mentén található csatlakozók számával, így az áramkör mérete csökkenthető lehet.

A 8 bementi csatorna sok alkalmazás esetén kevésnek bizonyulhat. A rendszer könnyen bővíthető nagyobb csatorna számúra is, mivel az SPI buszra több IC is helyezhető. Ebben az esetben érdemes lehet az ADS1298-as jelű IC választása, mivel az elérhető kisebb méretű tokozásban is. A hátránya, hogy a BGA kiszereléshez legalább 4 rétegű

nyomtatott áramkört kell készíteni, ami költségesebb, de a jeleket a középső rétegeken vezetve a zaj mértéke csökkenthető.

A GUI használata szempontjából nehézséget okoztak azok a karakterek, amik az első oszlopban találhatóak, mivel ilyenkor a karakter váltás és a felvillanás között túl rövid idő telik el az észleléshez. Az új karakter megjelenítése után a későbbiekben legalább egy ciklusnyi várakozási időt be kell iktatni.

Az EEGNet egy modern architektúra, amely nagyon jól teljesít adathalmazokon. Azonban napjainkban egyre jobban terjednek az LSTM - hosszú-rövid idejű memória - típusú architektúrák alkalmazása, így ehhez hasonló modellekkel való kísérletezés rávilágítana más, esetleg alkalmas megközelítésekre. Továbbá szükséges olyan adatok beszerzése, amelyek tartalmaznak zajokat, illetve melléktermékeket. Az ilyen zavarok detektálásával ki lehet szűrni azokat a számunkra nem hasznos jeleket, amelyek csak zavart okoznak a tanítás és használat során. A kis méretű adathalmazok problémájára kutatott különböző mesterséges jelgenerálási módszerekről már számos irodalom szól, így ezek kipróbálása enyhítheti az adathiány okozta nehézségeket.

# Köszönetnyilvánítás

Szeretnénk köszönetet mondani Dr. Szegletes Lucának, a konzulensünknek, aki mindig segített bennünket az ötleteivel és a legkilátástalanabb helyzetekben is biztatott minket. Köszönettel tartozunk Nagy Ákosnak is, aki nélkül nem találtuk volna meg az ADC olvasás kódjában a hibát.

# Irodalomjegyzék

- [1] URL <http://bnci-horizon-2020.eu/database/data-sets/003-2015/s1.mat>.
- [2] Reza Abiri – Soheil Borhani – Eric W Sellers – Yang Jiang – Xiaopeng Zhao: A comprehensive review of EEG-based brain–computer interface paradigms. *Journal of Neural Engineering*, 16. évf. (2019. jan) 1. sz., 011001. p.  
URL <https://doi.org/10.1088%2F1741-2552%2Faaf12e>.
- [3] Djork-Arné Clevert – Thomas Unterthiner – Sepp Hochreiter: Fast and accurate deep network learning by exponential linear units (elus), 2016.
- [4] Shahab Dabana – Eric Sellers – Clemens Holzner – Gunther Krausza and Roberta Carabalonac – Furio Gramaticac – Guenter Edlingera Christoph Gugera: How many people are able to control a p300-based brain–computerinterface (bci)? *Neuroscience Letters*, 462. évf. (2009), 94–98. p.
- [5] Adam Feuer: Hackeeg shield for arduino due.  
URL <https://www.starcat.io/products/hackeeg-shield/>.
- [6] Hossein Gholamalinezhad – Hossein Khosravi: Pooling methods in deep neural networks, a review, 2020.
- [7] Christoph Guger – Shahab Daban – Eric Sellers – Clemens Holzner – Gunther Krausz – Roberta Carabalona – Furio Gramatica – Guenter Edlinger: How many people are able to control a p300-based brain–computer interface (bci)? *Neuroscience Letters*, 462. évf. (2009) 1. sz., 94 – 98. p. ISSN 0304-3940. URL <http://www.sciencedirect.com/science/article/pii/S0304394009008192>.
- [8] Texas Instruments: Texas instruments ads1298.  
URL <https://www.ti.com/lit/ds/symlink/ads1298.pdf>.
- [9] Texas Instruments: Texas instruments ads1299.  
URL <https://www.ti.com/lit/ds/symlink/ads1299.pdf>.
- [10] Intan: Digital electrophysiologyinterface chips.  
URL [http://intantech.com/files/Intan\\_RHD2000\\_series\\_datasheet.pdf](http://intantech.com/files/Intan_RHD2000_series_datasheet.pdf).
- [11] Intel: Intel® dual band wireless-ac 8265.  
URL <https://www.intel.sg/content/www/xa/en/products/wireless/wireless-products/dual-band-wireless-ac-8265.html>.
- [12] Sergey Ioffe – Christian Szegedy: Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.
- [13] Diederik P. Kingma – Jimmy Ba: Adam: A method for stochastic optimization, 2017.



- [14] Mian Lau–King Hann Lim: Review of adaptive activation function in deep neural network. 2018. 12, 686–690. p.
- [15] Vernon J Lawhern–Amelia J Solon–Nicholas R Waytowich–Stephen M Gordon–Chou P Hung–Brent J Lance: Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of Neural Engineering*, 15. évf. (2018. Jul) 5. sz., 056013. p. ISSN 1741-2552. URL <http://dx.doi.org/10.1088/1741-2552/aace8c>.
- [16] Yann LeCun–Patrick Haffner–Léon Bottou–Yoshua Bengio: *Object Recognition with Gradient-Based Learning*. Berlin, Heidelberg, 1999, Springer Berlin Heidelberg, 319–345. p. ISBN 978-3-540-46805-9. URL [https://doi.org/10.1007/3-540-46805-6\\_19](https://doi.org/10.1007/3-540-46805-6_19).
- [17] Yann Lecun–Leon Bottou–Genevieve Orr–Klaus-Robert Müller: Efficient backprop. 2000. 08.
- [18] Jaakko Malmivuo–Robert Plonsey: *Bioelectromagnetism. 13. Electroencephalography*. 1995. 01, 247–264. p. ISBN 978-0195058239.
- [19] Microchip: Microchip rn42/rn42n. URL <http://ww1.microchip.com/downloads/en/DeviceDoc/50002328A.pdf>.
- [20] S. T. Microelectronics: Spbtle-rf0tr. URL <https://www.st.com/resource/en/datasheet/spbtle-rf0.pdf>.
- [21] Nvidia: Jetson nano developer kit. URL <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>.
- [22] OpenBCI: Cyton biosensing board (8-channels). URL <https://shop.openbci.com/collections/frontpage/products/cyton-biosensing-board-8-channel?variant=38958638542>.
- [23] John Polich: Updating p300: An integrative theory of p3a and p3b. *Clinical Neurophysiology*, 118. évf. (2007) 10. sz., 2128 – 2148. p. ISSN 1388-2457. URL <http://www.sciencedirect.com/science/article/pii/S1388245707001897>.
- [24] Rabie A. Ramadan–Athanasios V. Vasilakos: Brain computer interface: control signals review. *Neurocomputing*, 223. évf. (2017), 26 – 44. p. ISSN 0925-2312. URL <http://www.sciencedirect.com/science/article/pii/S0925231216312152>.
- [25] Aya Rezeika–Mihaly Benda–Piotr Stawicki–Felix Gembler–Abdul Saboor–Ivan Volosyak: Brain–computer interface spellers: A review. *Brain Sciences*, 8. évf. (2018) 4. sz. ISSN 2076-3425. URL <https://www.mdpi.com/2076-3425/8/4/57>.
- [26] Kay Robbins–Kyung min Su–W. David Hairston: An 18-subject eeg data collection using a visual-oddball task, designed for benchmarking algorithms and headset performance comparisons. *Data in Brief*, 16. évf. (2018), 227 – 230. p. ISSN 2352-3409. URL <http://www.sciencedirect.com/science/article/pii/S2352340917306285>.
- [27] Nand Sharma: Single-trial p300 classification using pca with lda, qda and neural networks, 2017.
- [28] Nitish Srivastava–Geoffrey Hinton–Alex Krizhevsky–Ilya Sutskever–Ruslan Salakhutdinov: Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15. évf. (2014) 56. sz., 1929–1958. p. URL <http://jmlr.org/papers/v15/srivastava14a.html>.

- [29] Michal Teplan: Fundamental of eeg measurement. *MEASUREMENT SCIENCE REVIEW*, 2. évf. (2002. 01).
- [30] Wadsworth bci dataset (p300 evoked potentials).  
URL [http://www.bbci.de/competition/iii/desc\\_II.pdf](http://www.bbci.de/competition/iii/desc_II.pdf).
- [31] Xiang Zhang–Lina Yao–Xianzhi Wang–Jessica Monaghan–David Mcalpine–Yu Zhang: A survey on deep learning based brain computer interface: Recent advances and new frontiers, 2019.

# Függelék

## F.1. Az elvégzett munka felosztása

Fejezet	Szerző
1	Torma Szabolcs
2	Südi István
3	Torma Szabolcs
??	Südi István
4.1	Südi István
4.2	Torma Szabolcs
5	Südi István és Torma Szabolcs
6	Südi István és Torma Szabolcs