

Brain Computer Interface

Studenckie Koło Naukowe Cybernetyki, Politechnika Warszawska

prof. dr hab. Antoni GRZANKA (opiekun naukowy) inż. Tomasz Bolesław CEDRO (koordynator/bci)
inż. Adrian BYSZUK (łączość) inż. Krzysztof CHOJNOWSKI (sieci neuronowe)
inż. Janusz FRĄCZEK (neurofeedback/eeg) lct. Ryszard GOMÓŁKA (neuroanatomia/medycyna)
Grzegorz LESZEK (sieci neuronowe) Piotr TAKIEL (programowanie/bci2000)
mgr inż. Tomasz KAMIŃSKI (sygnały biologiczne) mgr inż. Agnieszka MAŁKIEWICZ (sygnały biologiczne)

Styczeń 2011

Streszczenie

Celem projektu Brain Computer Interface prowadzonego przez Studenckie Koło Naukowe Cybernetyki działające przy Zespole Aparatury Biocybernetycznej Instytutu Systemów Elektronicznych na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej jest zbudowanie od podstaw modularnego urządzenia klasy interfejs człowiek–maszyna czy też mózg–komputer (ang. Brain Computer Interface), gromadząc przy tym wiedzę o istniejących rozwiązaniach komercyjnych i niekomercyjnych, próbując rozpoznać i zestawić parametry istniejących rozwiązań, a przede wszystkim stworzyć bazę porównawczą stanowiącą punkt odniesienia dla dalszych badań i naszych własnych konstrukcji.

Niniejsze opracowanie stanowi esencję prac badawczych w obszarze BCI prowadzonych od kilku lat z inicjatywy Tomasza Cedro w ramach Studenckiego Koła Naukowego Cybernetyki działającego pod nadzorem prof. nzw. dr hab. Antoniego Grzanki oraz członków Zespołu Aparatury Biocybernetycznej. Wyniki przedstawione w niniejszym opracowaniu obejmują opracowania stworzone przez zespół studentów PW w ramach prac dyplomowych, projektów koła, w tym badań wspartych Grantem Rektorskim w roku 2010.

Dofinansowanie w ramach Grantu Rektorskiego pozwoliło nam głównie na zbudowanie znanego, niedrogo i otwartego systemu akwizycji OpenEEG (rozdział ??), a także porównanie jego parametrów i ograniczeń w odniesieniu do własnej konstrukcji wzmacniacza sygnałów biologicznych (Janusz Frączek, rozdział 3.1), oraz mikroprocesorowego systemu wbudowanego do akwizycji, przetwarzania i transmisji danych (Tomasz Cedro, rozdział 2.4, ??, 2.3). Wiele otwartych programów komputerowych, a nawet środowisk programistycznych do pracy z sygnałami biologicznymi, Neurofeedback’iem, czy szerzej BCI pozwoliło nam na praktyczne sprawdzenie możliwości wykorzystania potencjałów bioelektrycznych, w szczególności elektrycznej aktywności mózgu związanej z aktywnością intelektualną, do sterowania urządzeniami czy systemami komputerowymi, a nawet zastosowaniami w medycynie (Agnieszka Małkiewicz, Tomasz Kamiński, Ryszard Gomółka).

Stosunkowo proste urządzenia zwykle wykorzystywane do treningu Neurofeedback mogą już zostać wykorzystane do prostych aplikacji BCI, co sprawdzamy tworząc własny moduł użytkownika w postaci prostej gry komputerowej sterowanej falami EEG w systemie BCI2000 (Piotr Tąkiel, ??). Pokusiliśmy się także o prostą weryfikację wpływu systemu szybkiej nauki języków obcych SITA na aktywność elektryczną mózgu (rozdział ??), oraz próbę stworzenia sztucznej sieci neuronowej będącej w stanie rozpoznawać określone bodźce na podstawie przebiegu sygnału EEG odczytywanego na żywo z pacjenta (Krzysztof Chojnowski, Grzegorz Leszek, rozdział 2.2).

Dostęp do bardziej zaawansowanych urządzeń pomiarowych da nam możliwość dalszego rozwoju, propagowania tematyki wśród studentów PW, a co najważniejsze punkt odniesienia dla własnych konstrukcji. Frekwencja oraz żywe zainteresowanie uczestników Festiwalu Nauki, Pikniku Kół Naukowych PW, oraz wielu konferencji naukowych na których prezentujemy swoje prace wskazują na bardzo duże zainteresowanie tematyką BCI zarówno od strony naukowej jak i użytkowej.

Ze względu na interdyscyplinarny charakter badań oraz obszerność materiału, dokument podzielony został na trzy główne części – teorię, oprogramowanie i sprzęt elektroniczny – z których każda zawiera pewną porcję informacji z danej dziedziny pozwalając na zapoznanie się od podstaw z tematyką BCI i wyrobienie sobie poglądu na tego typu rozwiązanie jako funkcjonalną całość. Opisy są zwarte ale wzbogacone znaczną ilością referencji i odsyłaczy do źródeł informacji. Duża część prac ma swoją dedykowaną dokumentację w postaci stron internetowych lub osobnych dokumentów.

Oczywiście nie sposób jest w tak krótkim dokumencie zmieścić wszelkie niezbędne informacje, zależy nam jednak na tym aby przedstawić wyniki choć części naszych dotychczasowych prac i starać o stworzenie tego typu systemu od podstaw.

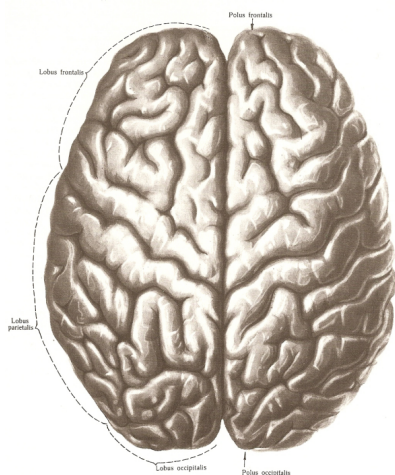
1 Teoria

1.1 Podstawy anatomii i fizjologii ośrodkowego układu nerwowego człowieka (Ryszard Gomółka).

Najważniejszym elementem opracowania jest charakterystyka fal mózgowych, jednak na początek chciałbym zacząć od wstępu w postaci budowy powierzchni mózgu – tego co nas najbardziej interesuje.

1.1.1 Budowa zewnętrzna mózgu

Ciało mózgu (*cerebri*) składa się z dwóch prawie symetrycznych półkul (hemispheria cerebri) oddzielonych wzajemnie szczeliną podłużną (*fissura longitudinalis*). Łącznie obie półkule posiadają eliptyczny kształt, w którym wyróżniamy trzy bieguny: biegun czołowy (*polus frontalis*), biegun potyliczny (*polus occipitalis*), biegun skroniowy (*polus temporalis*).



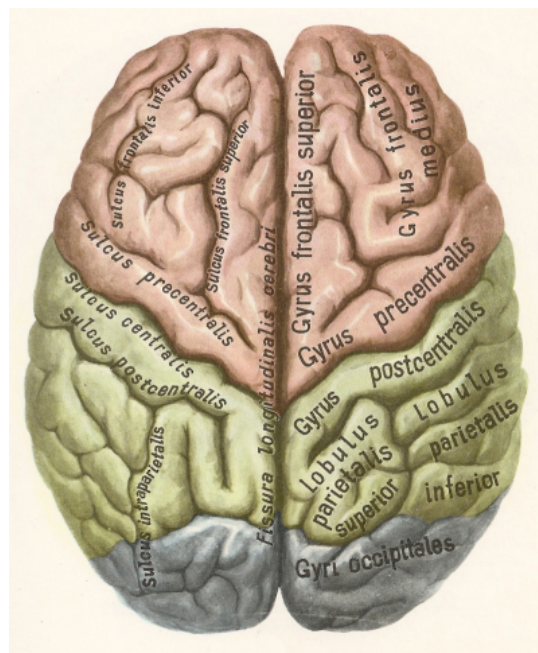
Rysunek 1: Półkule mózgowe. Widok z góry [2].

Półkule pokryte są bruzdami (*sulci cerebri*) ograniczającymi części w postaci zakrętów mózgu (*gyri*). Położenie bruzd i zakrętów wydaje się być dosyć chaotyczne i oparte na wzajemnym dopasowaniu, jednak badania wykazały, że zjawisko to dotyczy jedynie niewielkich bruzd. Niektóre bruzdy mają swoje stałe i niezmiennie położenie – nazwano je bruzdami pierwotnymi lub inaczej głównymi, które rozwijają się już w życiu płodowym. Zmienności osobniczej podlegają jedynie bruzdy drugo i trzeciorzędowe. Najgłębszą bruzdą jest bruzda boczna (*sulcus lateralis*) oddzielająca wyspę (*insula*) półkuli mózgu (bardziej kojarzona ze skroniami) od górnej części mózgowia. Kolejną pod względem wielkości jest bruzda środkowa (*sulcus centralis*), a następną bruzda ciemieniowo-potyliczna (*sulcus parietooccipitalis*) [Ryc. 1,2,3].

Wyżej wymienione bruzdy dzielą obszar kory mózgo-wej na cztery płaty:

- Czołowy (*lobus frontalis*)
- Skroniowy (*lobus temporalis*)
- Ciemieniowy (*lobus parietalis*)
- Potyliczny (*lobus occipitalis*)

Każdy płat okryty jest z zewnątrz łuską kości wcho-dzącej w budowę czaszki o tej samej nazwie co płat okryty.



Rysunek 2: Schemat podziału półkul mózgowych na płaty. Widok z góry [2].

1.1.2 Kora mózgowa

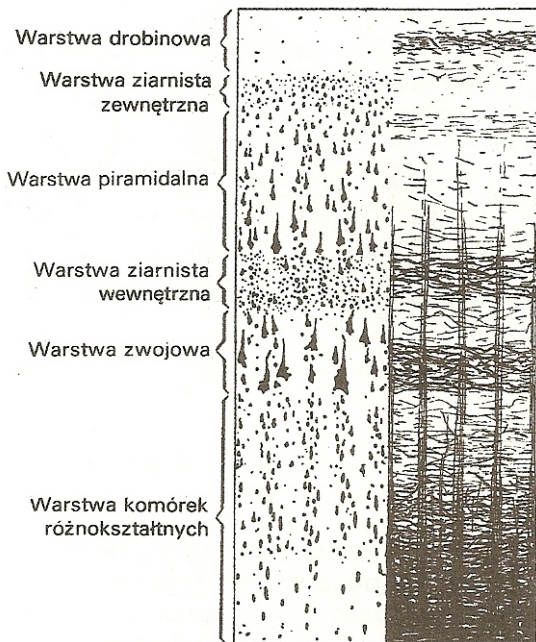
Powierzchnia półkul mózgowych jest pokryta paromilimetrową warstwą komórek istoty szarej o charakterystycznym ułożeniu. Największa część kory nie jest widoczna na zewnątrz, lecz ukryta jest w bruzdach mózgu, które istotnie zwiększają powierzchnię czynną narządu.

Właściwą dla kory mózgo-wej jest budowa warstwowa, w której największa część, bo prawie 95% zawarta jest w 6 warstwach opisanych na rys.4.

1. Warstwa drobinowa, brzeżną. Zawiera względnie niewiele komórek nerwowych. Obecne są tu wyraźnie aksony oraz dendryty neuronów warstw głębszych.



Rysunek 3: Podział mózgu na płaty. Widok od boku [2].



Rysunek 4: Budowa kory mózgowej. Z lewej układ komórek nerwowych, z prawej włókien nerwowych [1].

2. Warstwa ziarnista zewnętrzna. Tu panują komórki piramidalne i ziarniste (podział czysto morfologiczny), których neuryty nie wnikają do istoty białej.
3. W warstwie piramidalnej luźno rozmieszczone są komórki piramidalne, których wielkość zwiększa się w kierunku warstwy 4.
4. warstwa ziarnista wewnętrzna – małe komórki

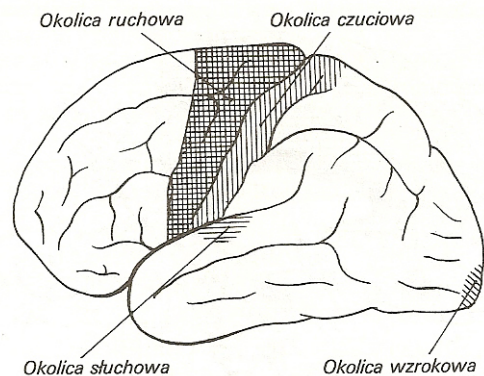
ziarniste

5. Warstwa zwojowa – tu pojawiają się komórki piramidalne olbrzymie (kom. Betza)
6. Warstwa komórek różnokształtnych.

Istotną kwestią jest fakt, iż komórki w korze mózgowej zawsze biegną w pęczkach pionowo, zaś czynność kory mózgowej nie jest nigdy chaotyczna. Potencjał czynnościowy jednego neuronu zostaje szybko rozprzestrzeniony na całą błonę i pęczek, dzięki czemu sygnał jest przenoszony szybko (wraz ze wzrostem grubości neuronu, czy pęczka szybkość sygnału wzrasta). Zjawisko takie jest możliwe dzięki istnieniu i ścisłej współpracy gliu z komórkami nerwowymi. Procesy myślenia i poznawcze rozchodzą się w korze mózgowej powyżej 4 warstwy. Pochodzenie ich jest jednak z warstw poniżej 4 (obwodowy układ nerwowy, układ limbiczny)

1.1.3 Pola kory

Istnieje wiele podziałów kory mózgowej uwzględniających budowę morfologiczną komórek, funkcje, czynności. Najpopularniejszym jest podział Brodmanna (czyś to architektoniczny), który przyjął że kora dzieli się na 52 regiony.



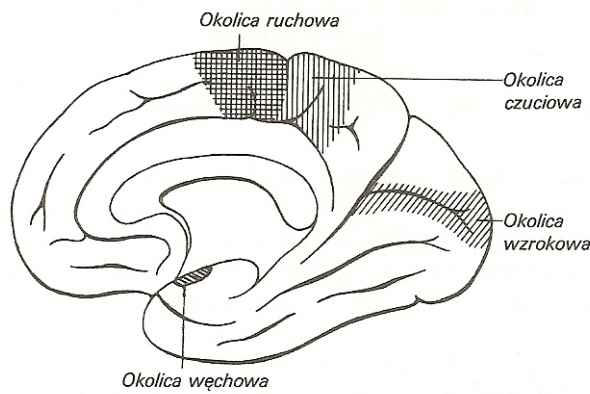
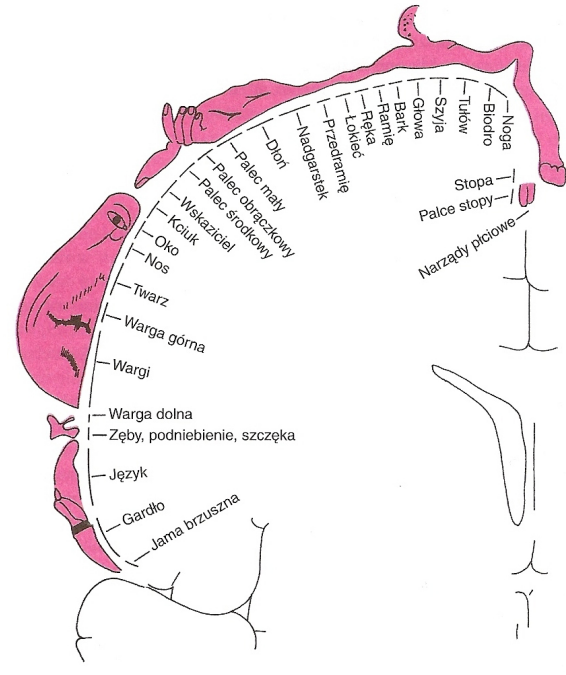
Rysunek 5: Umiejscowienie okolic w korze mózgu. Widok powierzchni górno-bocznej [1].

Ścisła lokalizację wykazują regiony kory, w których zapoczątkowane są włókna ruchowe (kora ruchowa), a także do których dochodzą drogi czuciowe (kora węchowa, słuchowa, czuciowa i wzrokowa). Ponieważ nie ma sensu byśmy wchodzili w szczegółową budowę każdego z wyodrębnionych regionów korowych pozwolę sobie przyjąć bardzo uogólniony podział, który nie jest do końca prawidłowy, acz prosty:

- Płat czołowy – kora ruchowa
- Płat ciemieniowy – kora czuciowa

- Płat skroniowy – kora słuchowa
- Płat potyliczny – kora wzrokowa

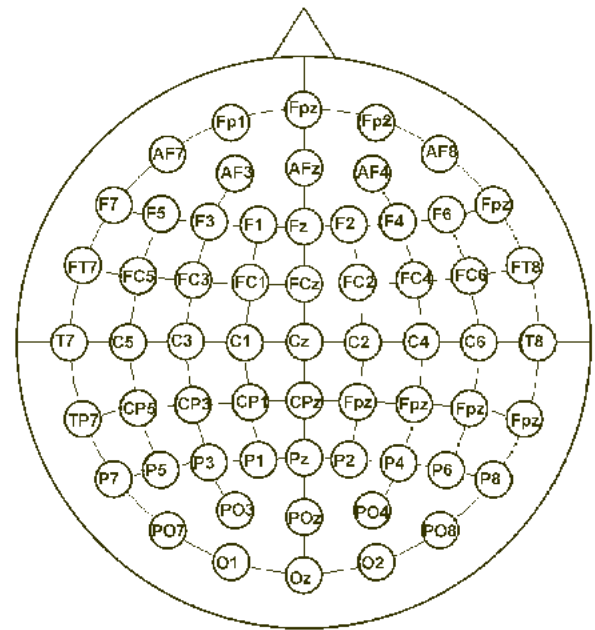
Co dotyczy kory węchowej – jest ona skupiona tak naprawdę pod bryłą płata czołowego w postaci dwóch symetrycznych opuszek węchowych (bulbus olfactorius) oraz pola na wewnętrznej części wysp. Jest to związane z dosyć nowym zjawiskiem filogenetycznym jakim jest powstawanie kory mózgowej u wyższych kręgowców. U niższych kręgowców czynność kory jest związana stricte z odbieraniem bodźców węchowych. Jak widać na przykładzie człowieka u ssaków nastąpił istotny przerost innych rejonów korowych nad korą węchową.



Rysunek 7: Podział pól w korze ruchowej. Przekrój w płaszczyźnie czołowej [3].

Rysunek 6: Umieszczenie okolic w korze mózgu. Widok powierzchni przyśrodkowej [1].

Bardziej konkretny podział, który docelowo powinien nas interesować w badaniach, przedstawiono na rycinach poniżej [Ryc. 5,6,7]. Uwidaczniają one faktyczne rejony ścisłego występowania okolic wymienionych wyżej. Warto nadmienić, że w przypadku kory ruchowej i czuciowej okolice ciała są reprezentowane podobnie (od dołu głowa, szyja, kończyna górna, tułów, kończyna dolna). Oczywiście pojedyncza półkula unerwia przeciwną stronę ciała.



Notka: Treść zawarta w opracowaniu przeznaczona jest wyłącznie dla studentów WEiT/PW uczestniczących w projekcie Neurofeedback. Ma ona za zadanie ujednolicić wiedzę posiadaną przez członków i nie jest przeznaczona do rozpowszechniania, ani w celach komercyjnych.

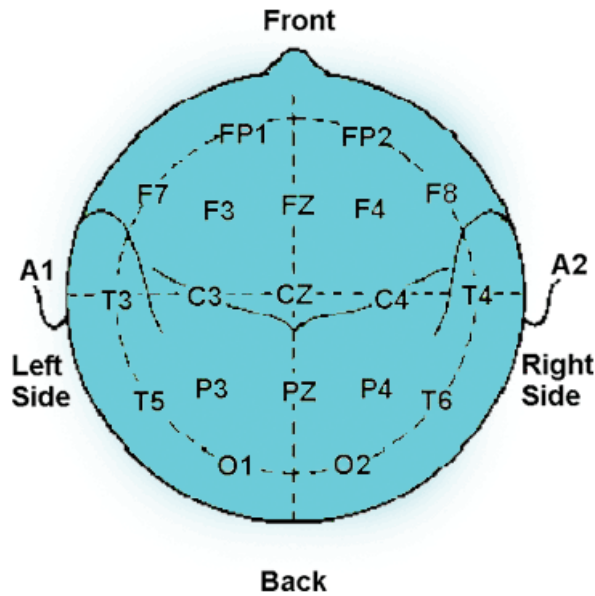
Rysunek 8: Ułożenie elektrod w systemie 10-20. Widok z góry.

1.2 Zasady aplikacji elektrod i cechy sygnału EEG (Ryszard Gomółka)

1.2.1 System 10–20

W każdym układzie aplikacji elektrod EEG najważniejszą czynnością jest wyznaczenie wierzchołka głowy

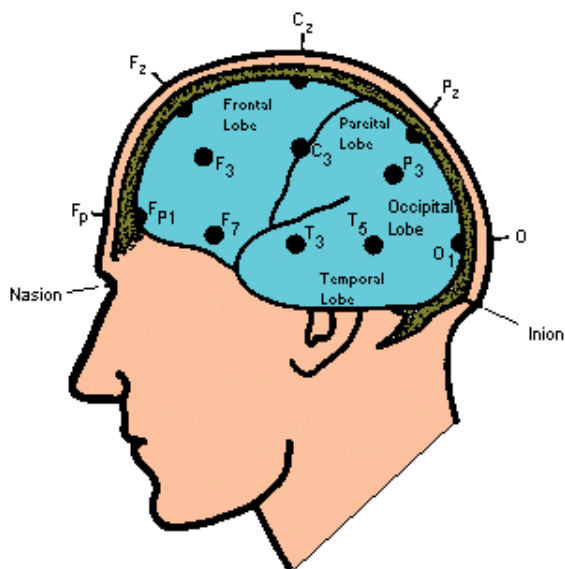
w postaci przecięcia wymiaru strzałkowego (od nasady nosa-„nasion”- po guz potyliczny zewnętrzny-„inion”) i połowy wymiaru wieńcowego (od 1 skrawka ucha do drugiego). Punkt ten to „vertex” (czubek) i w tym miejscu ustawiamy elektrodę „Cz”.



Rysunek 9: Ułożenie elektrod w międzynarodowym systemie 10-20. Widok z góry[4].

Wymiar poziomy określają:

- punkty w 10% odległości wymiaru strzałkowego powyżej nasion i inion
- 10% powyżej skrawków uszy (wymiar wieńcowy)



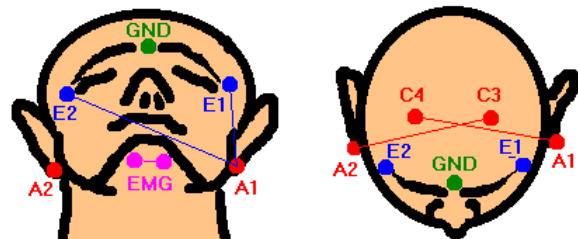
Rysunek 10: Ułożenie elektrod w systemie 10-20. Widok z boku[4].

Elektrody oznaczone liczbami nieparzystymi znajdują się po lewej stronie głowy, parzyste po prawej. Przesuwając się po linii strzałkowej w odległości 20%

od Cz umieszczamy czołowo elektrodę Fz (frontalis) i ciemieniowo Pz (parietalis)[Ryc. 8-11]. Nazwy elektrod pochodzą od rejonów w jakich są podpięte:

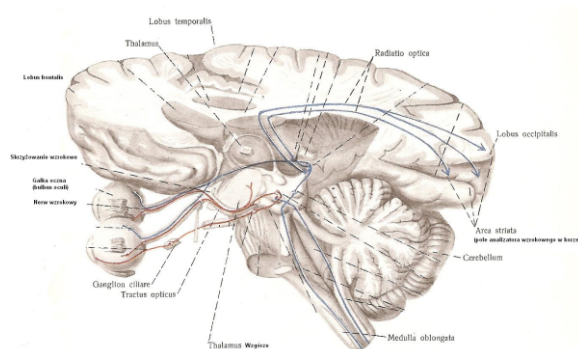
- P – parietalis (ciemień)
- T – temporalis (skronie)
- O – occipitalis (potylicca)
- F – frontalis (czoło)

Główną wadą jest to, że wszystkie odprowadzenia powinno się uziemiać do płątka ucha w celu otrzymaniu najlepszego sygnału. System 10-20 także nie jest jedyny. Poza tym elektrody F7 i F8 są za wysoko żeby odczytywać sygnały przedniej osi- powinno stosować się T1 i T2 umieszczone w 1/3 odległości między otworem słuchowym zewnętrznym i kątem oka.



Rysunek 11: Ułożenie elektrod referencyjnych (A), uziemiającej (GND) i ocznych (E).

Ponieważ początkowo projekt będzie opierać się głównie na potencjałach wywołanych zbieranych w okolicy potylicznej, zamieszczam poniższą rycinę, gdyż w dobry sposób obrazuje ona schemat przechodzenia włókien nerwowych od nerwu wzrokowego aż po sam ośrodek wzrokowy w korze[12].



Rysunek 12: Schemat przebiegu włókien nerwu wzrokowego do okolicy kory wzrokowej. Przekrój strzałkowy.

1.2.2 Podstawowe grafoelementy zapisu EEG u osoby dorosłej

U osoby dorosłej rytmem podstawowym jest rytm alfa. Odkryty został przez Bergera w 1929 roku. Charakteryzuje się on częstotliwością 8-13 Hz. Ujawnia się po zamknięciu oczu, zanika po otwarciu. Jego źródło znajduje się w płacie potylicznym (tam też ma największą amplitudę), zatem uważa się, że jest to rytm tylny podstawowy.

Czynność rytmu alfa zazwyczaj jest symetryczna, jednak największą amplitudę osiąga nad półkulą nie-dominującą. Do przyjęcia jest stosunek 2:1. U śpiącego (prócz odpowiedniej fazy snu) rytm alfa może pojawiać się podczas hiperwentylacji, a w czasie senności może pojawić się przy otwartych oczach. Rytm powinien być obecny w obu półkulach. Może szerzyć się mniej lub bardziej symetrycznie ku przodowi, lecz nie jest to zjawisko patologiczne. Podczas śpiączki alfa przeważa w czole i stanowi złą prognozę dla chorego. 5% populacji nie posiada rytmu alfa podczas spoczynku.

Rytm beta – za podstawowy rytm beta uznaje się czynność powyżej 14 Hz. Posiada on charakter rytmiczny i stanowi tło zapisu u 95% populacji. Amplituda fal waha się znacznie, jednak niezależnie od znanych leków. Maksymalną amplitudę osiąga w rejonach czołowo-centralnych. Nie reaguje na otwarcie oczu. Asymetria półkulowa wskazuje na stan chorobowy.

Theta charakteryzuje 4..7 Hz. U zdrowego człowieka są obecne, lecz nie dominujące i powinny stanowić co najwyżej 5% zapisu. Zwykle występują w stanie czuwania, jednakże mogą być nieobecne. Wyraźne natomiast w linii środkowej i skroniowej. Częstość występowania zwiększa się w czasie senności. U dzieci rytm ten jest rozszany i uogólniony, jednak należy to do charakterystyki fizjologicznej mózgu dziecka.

Delta poniżej 4 Hz. Są to bardzo wolne, wysokonapięciowe potencjały. Nie występują w czasie czuwania, jednak są integralną częścią zapisu w śnie. Ich obecność w czasie czuwania zawsze oznacza dysfunkcję mózgu, lub powstanie zmiany ogniskowej widocznej w diagnostyce obrazowej. Rytmu delta stanowią jednak integralną część zapisu EEG osoby dorosłej. U osób starszych występują często w skroniach, gdzie indziej rzadko.

Rytm mu – zwany rytmem bramkowym z odprowadzeń centralnych, znad kory ruchowej o częstości 7..11 Hz (subharmoniczna beta). Może być jedno lub dwustronny, synchroniczny lub asynchroniczny. Może występować w senności, także z otwartymi oczami.

Lambda – elektrodododatnie przejściowe impulsy z potylicy. Ostro zarysowane, zazwyczaj symetryczne. Zaobserwowane podczas oglądania przez pacjenta interesujących obiektów, obrazów. Są to wzrokowe potencjały wywołane i należą do fizjologii.

1.2.3 Cechy charakterystyczne EEG u dzieci

duża zmienność (większa niż u dorosłych) – dotyczy tak rytmów tła, jak i fal wolnych.

- Wraz z dojrzewaniem obserwuje się stały wzrost częstotliwości dominującego rytmu tylnego.
- Alfa nieobecny po urodzeniu- pojawia się w 2-3 miesiącu życia
- Początkowo dominująca rytmiczna delta, wkrótce rozwija się polirytmiczną deltę + thetę
- Stopniowy wzrost średniej częstotliwości wraz z dorastaniem (w 1 roku przeważa delta, pojawia się theta).
- W wieku 3 lat stwierdza się dobrze zorganizowaną i rytmiczną thetę, pojawia się silniej alfa, zaś delta staje się mniej widoczna
- W wieku 6 lat dominuje zazwyczaj dobrze zorganizowany rytm alfa
- 7,8 lat – zwiększa się średnia częstotliwość alfy osiągając 10 Hz
- podczas dojrzewania cechą prawidłową jest występowanie fal theta.

1.2.4 Artefakty zapisu EEG

Ponieważ większość z Was jest specjalistami w zakresie wykonywania pomiarów biomedycznych, nie ma sensu żebym o czymkolwiek pisał, co jest jasne przy każdym badaniu pacjenta.

Główne artefakty występują ze złego przygotowania i przymocowania elektrod, ruchów pacjenta, odruchów mimowolnych, czynności fizjologicznych organizmu. Najważniejszymi są jednak:

- trzaskanie – nieszczelność między elektrodą, a powierzchnią skóry. Wiąże się ze wzrostem impedancji elektrody.
- artefakty mięśniowe – najczęstsze w okolicy czołowej i skroniowej (żwacze, ruchy języka). Dobrze eliminuje go lekkie otwarcie ust.
- Pocenie się – objawia się jako wolne falujące potencjały kilkusekundowe. Odzwierciedla zwarcie prądów między dwiema elektrodami poprzez powstanie mostka solnego z potu
- Artefakty EKG – możliwe do wyeliminowania poprzez oddzielną rejestrację w 1 kanale i odjęcie sygnału.
- artefakty z tętna – np. elektroda znajduje się blisko tętnicy skroniowej powierzchownej (ogólnie staramy się omijać większe naczynia).

- ruch gałek ocznych – płynny ruch widoczny głównie w odprowadzeniach czołowych i skroniowych.
- nieprawidłowe przygotowanie do badania – zależy od rodzaju badania i zaleceń lekarza.

Z pewnością należy unikać hipoglikemii, która objawia się ogólnym spowolnieniem czynności elektrycznej mózgu. Osoba badana nie powinna być głodna.

1.3 Elektrody pomiarowe EEG (Tomasz Cedro)



Rysunek 13: Elektrody grzybkowe mocowane czepkiem silikonowym.

W badaniach przez nas prowadzonych zastosowane zostały dwa typy elektrod wykonanych ze srebra i chlorku-srebra wyprodukowane przez polską firmę Elmiko z Warszawy:

- miseczkowe – wymagają użycia pasty 1020 zmniejszającej oporność naskórka, co wymusza konieczność zabrudzenia głowy pacjenta, ale pozwala na dosyć długie wykonywanie pomiarów nawet bez użycia czepka mocującego, ponieważ pasta jest dosyć sztywna, lepka i wolnoschnąca. Tego typu elektrody są powszechnie stosowane w treningu Neurofeedback ze względu na prostą aplikację.
- grzybkowe – wymagają uprzedniego namoczenia w soli fizjologicznej lub stosowania specjalnego żelu przewodzącego. Są mocowane do czepka silikonowego, a więc nie mogą być umieszczone samodzielnie na głowie pacjenta, są za to bardziej mobilne od miseczkowych.

Elektrody utrzymuje na określonej pozycji czepki zbudowane z wzajemnie krzyżujących się rurek silikonowych pod którymi umieszcza się elektrody. Występują również „czepki automatyczne” wykonane z solidnego materiału z wbudowanymi elektrodami – mają one



Rysunek 14: Elektroda miseczkowa.

jednak dosyć wysoką cenę i służą do pomiarów wielokanałowych aktywności elektrycznej różnych obszarów mózgu. Elektrody są wielokrotnego użytku, wystarczy je wypłukać ciepłą wodą i wysuszyć. Istnieje również możliwość regeneracji zużytych elektrod u producenta za niewielką opłatą.



Rysunek 15: Silikonowy czepki mocujący elektrody podczas badań.

Obydwa typy elektrod sprawdziły się bardzo dobrze w pomiarach, których dokonywaliśmy. Używaliśmy pięciu elektrod – po dwie na jeden kanał różnicowy i jedna na elektrodę DRL (potencjał odniesienia). Choć elektrody grzybkowe wymagają czepka, to można je stosować nawet po delikatnym zwilżeniu, natomiast elektrody miseczkowe obowiązkowo wymagają zastosowania niedrogiej pasty 1020, są więc w pewnym stopniu od niej zależne. Dodatkowo, elektrody miseczkowe na stałe związane są z przewodem i wtykiem pomiarowym, co może utrudniać zmianę ich położenia na powierzchni głowy. Elektrody grzybkowe podłączane są osobnym przewodem zakończonym zasiskiem krokodylkowym, dzięki czemu można łatwo zmienić podłączenie przewodu sygnałowego a nawet samej elektrody. Niestety w obydwu przypadkach elektrody podłączane były przewodem nieekranowanym, co znacznie utrudniało pomiar wprowadzając znacznej amplitudy zakłócenia z sieci energetycznej po podłączeniu przewodów do urządzenia pomiarowego.

2 Oprogramowanie

2.1 Otwarte programy BCI/Neurofeedback (Tomasz Cedro)

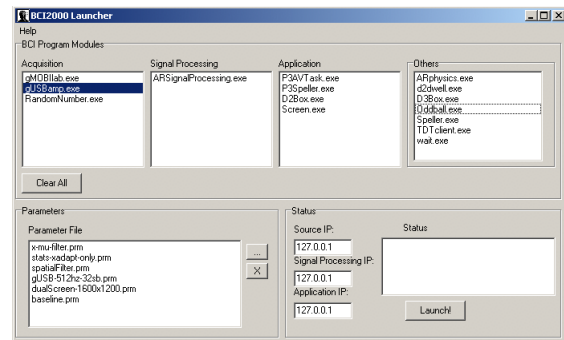
Niniejszy rozdział zawiera opis kilku otwartych rozwiązań składających się na ogólnodostępne systemy Brain Computer Interface. Urządzenia przedstawione poniżej są absolutnie minimalnym kosztem jaki należy ponieść chcąc prowadzić badania w tej dziedzinie i najczęściej sprowadzają się wyłącznie do ceny podzespołów elektronicznych, przewodów i elektrod. Oprogramowanie jest zwykle darmowe, czasami ograniczone licencją na wykorzystanie komercyjne, do niektórych programów dostępny jest kod źródłowy. Nie jest to ani sprzęt ani oprogramowanie wykorzystywane w profesjonalnych laboratoriach, ale pozwala na wstępne zapoznanie się z praktycznymi aspektami pomiarów (oraz problemami do pokonania). Na takich właśnie rozwiązaniach bazują nasze badania, a ich główną zaletą jest możliwość współpracy pomiędzy różnymi programami/urządzeniami lub stworzenie takiej możliwości dzięki otwartej architekturze.

2.1.1 BCI2000



BCI2000 [27] to chyba najbardziej znane zintegrowane środowisko programistyczne do badań nad BCI dla systemu Windows, choć ograniczone licencją do zastosowań niekomercyjnych, którą należy podpisać aby uzyskać dostęp do programu i kodu źródłowego. Dobrze udokumentowane w centralnym repozytorium Wiki [28] i sprawdzone przez różne grupy badawcze na całym świecie stanowi świetną podstawę do stawiania pierwszych kroków, a nawet wykorzystania gotowych przykładów (prosta gra neurofeedback, pisanie tekstu z wykorzystaniem potencjału P300, wybieranie kolorowego pola, itp.).

W rzeczywistości BCI2000 to cały zestaw programów i narzędzi programowych służących do analizy i przetwarzania danych zarówno w czasie rzeczywistym jak i „offline” z pliku. Programy narzędziowe przeważnie pomagają w analizie wyników, natomiast sam system składa się z osobnych programów wymieniających między sobą dane poprzez gniazda sieciowe (socket). Zasadnicze cztery moduły programu, które mogą być uruchomione z pliku wsadowego lub aplikacji okienkowej BCI2000Launcher (rys.16) to:



Rysunek 16: Program uruchamiający moduły systemu BCI2000.

- Source – źródło danych EEG, może być dedykowanym sterownikiem do własnego urządzenia lub istniejącym modulem do standardowej aparatury pomiarowej.
- Signal Processing – moduł w którym następuje przetwarzanie sygnałów.
- User Application – jest programem użytkownika, w którym wybrany element może być sterowany sygnałem z poprzedniego modułu.
- Operator – steruje pracą trzech poprzednich modułów, pozwala na wczytanie/zapisanie konfiguracji środowiska pomiarowego lub samego pomiaru.

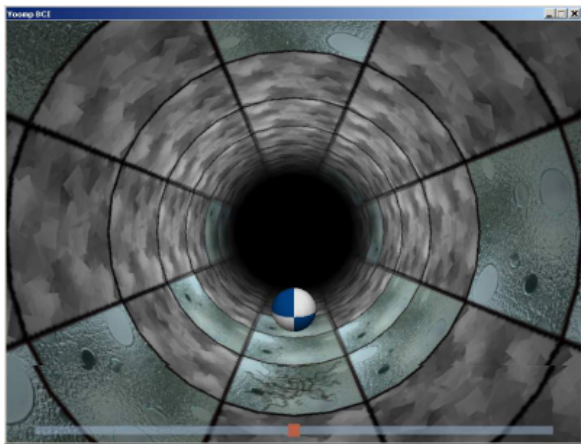
Sesja pomiarowa zwykle składa się z wczytania parametrów doświadczenia z zewnętrznego pliku (aby nie ustawiać ich od nowa za każdym razem) oraz „próbach” w których odbywa się sam pomiar (3.5 minut każda, powtarzanych po ponownej aktywacji przyciskiem Resume na konsoli operatorskiej modułu Operator). Wyniki zapisane na dysku można w czasie późniejszym analizować dodatkowymi programami (np. BCI2000Viewer).

Dla ułatwienia zadania w testowaniu aplikacji istnieje możliwość podłączenia kursora myszy jako wejścia sygnału symulującego aktywność mózgu. Sam pomiar sygnału EEG jest dosyć złożony, podatny na zakłócenia i wymaga wielu (kilkunastu) odprowadzeń, przy czym my dysponowaliśmy dwoma, dlatego ten rodzaj symulacji bardzo pomógł nam w przeprowadzeniu eksperymentów, a nawet stworzeniu własnej aplikacji – gry sterowanej (docelowo) myślami.

Dalsze badania sprowadzają się do skompletowania aparatury pomiarowej EEG i pełnego wykorzystania potencjału systemu odbierając czysty sygnał z kilkunastu odprowadzeń.

2.1.2 Własna aplikacja użytkownika w systemie BCI2000 (Piotr Tąkiel, Tomasz Cedro)

Postanowiliśmy rozpoznać system BCI2000 od strony programowej tworząc prostą grę, która mogłaby być sterowana myślami, a przykładem takiej gry jest piłeczka skacząca w trójwymiarowym tunelu przemieszczającym się nieustannie do przodu (pierwowzorem jest gra Yoomp na 8-bitowy komputer Atari).



Rysunek 17: Gra stworzona przez Piotra Tąkiela do współpracy w czasie rzeczywistym z systemem BCI2000.

Gra stworzona została w oparciu o Podręcznik Programisty BCI2000 [29] oraz bibliotekę graficzną SDL [30], co pozwala na jej łatwe przeportowanie dla systemów Unix (w nowej odsłonie BCI2000 będzie wieloplatformowa). Tuż po uruchomieniu gra daje się sterować klawiszami kursorów, ale również oczekuje na przychodzące pakiety z modułu Operatora BCI2000, a kiedy taki pakiet zostanie wykryty sterowanie odbywa się również z programu BCI2000. Najłatwiej jest przetestować program podłączając myszkę jako wejście sygnału EEG – odpowiedni plik z konfiguracją został załączony do programu.

2.1.3 OpenEEG NeuroServer

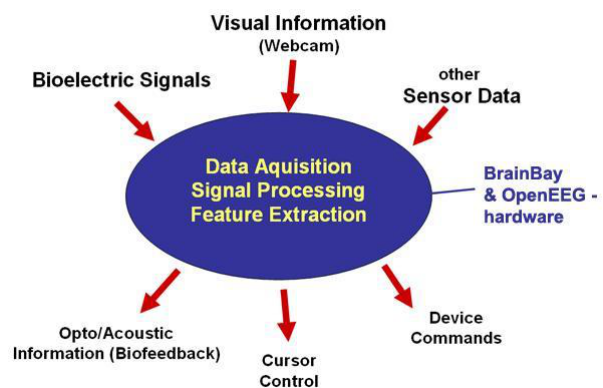
Neuroserver [32] to część projektu OpenEEG, która ma zapewnić dostęp do danych pomiarowych z urządzenia niezależnie od platformy czy systemu docelowego, a taką możliwość obecnie zapewniają gniazda sieciowe (sockets BSD). Ponieważ sterownik oraz format danych może być różny, fizyczne podłączenie urządzenia do komputera również, to NeuroServer stanowi pewien pomost pomiędzy urządzeniem a programem. Neuroserver posiada dodatkowo zewnętrzny, wolnostojący sterownik specyficzny dla wybranego urządzenia, który odbiera dane z fizycznego portu fizycznego urządzenia, dane ze sterownika są następnie przesyłane do

serwera, a stamtąd mogą być pobrane przez dowolny program łączący się pod wskazany adres sieciowy standardowym protokołem – czy to komputera lokalnego czy też dowolnej maszyny sieci internet.

Wiele programów wykorzystuje ten rodzaj połączenia z urządzeniem pomiarowym, ponieważ program nie musi być uruchamiany na komputerze do którego podłączone jest urządzenie pomiarowe, a programista nie musi pisać własnego sterownika, jedynie odbiera gotowe dane pomiarowe. Wykorzystanie mechanizmu gniazd sieciowych to bardzo przyjazny i skuteczny sposób na wymianę danych, ponieważ może być zastosowany w dowolnym medium komunikacji jako warstwa wyższa, zwalniając konstruktora z konieczności oprogramowania niskopoziomowych sterowników zastosowanej magistrali.

Warto również wspomnieć, że tego typu rozwiązanie sprzyja zdecydowanie prototypowaniu, gdyż zamiast sterownika prawdziwego urządzenia można podłączyć sterownik programowy generujący zadane przebiegi wzorcowe niezbędne do testowania aplikacji (np. tak jak w przypadku BCI2000 gdzie sygnał generował ruch myszki).

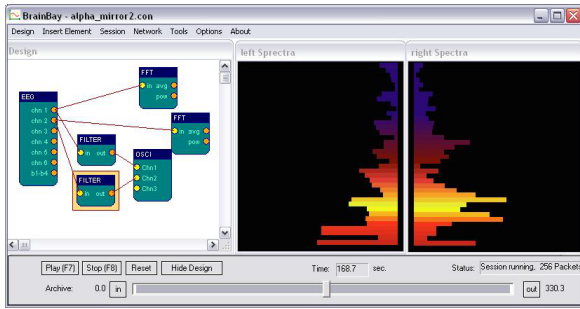
2.1.4 BrainBay



Rysunek 18: Koncepcja systemu Brain Bay.

BrainBay [31] to otwarta modułarna aplikacja stworzona do współpracy z systemami OpenEEG, ModularEEG oraz MonolithEEG, oferująca funkcje Neurofeedbacku poprzez graficzne składanie bloków (podobnie jak w wizualnych środowiskach pomiarowych czy syntezatorach muzycznych). Dane z urządzeniami wymieniane są poprzez NeuroServer [32] a więc gniazda sieciowe (sockets), co zapewnia kompatybilność z innymi rozwiązaniami zgodnymi z systemem OpenEEG.

Generacja obrazu, dźwięku, a nawet gier sterowanych sygnałem EEG jest stosunkowo łatwa i dydaktyczna, ponieważ polega na łączeniu wirtualnymi prze-



Rysunek 19: BrainBay w akcji.

wodami wirtualnych urządzeń na ekranie komputera, co pozwala na swobodne eksperymentowanie z różnymi blokami funkcjonalnymi, pokazuje też wyraźnie strukturę badanego systemu. Jest to zdecydowanie najlepszy program do praktycznej nauki przetwarzania sygnałów, a w dodatku otwarty, darmowy i z dostępnym kodem źródłowym.

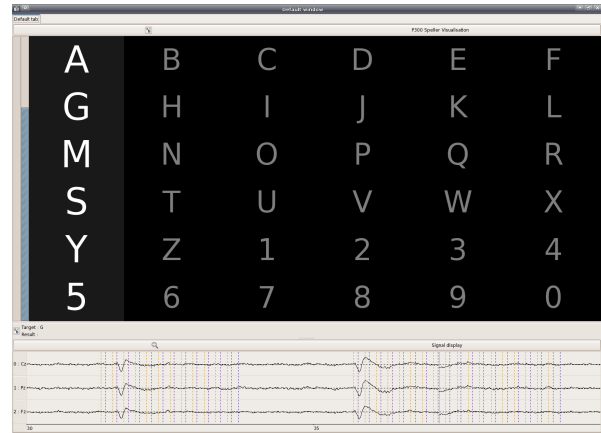
2.1.5 OpenViBE



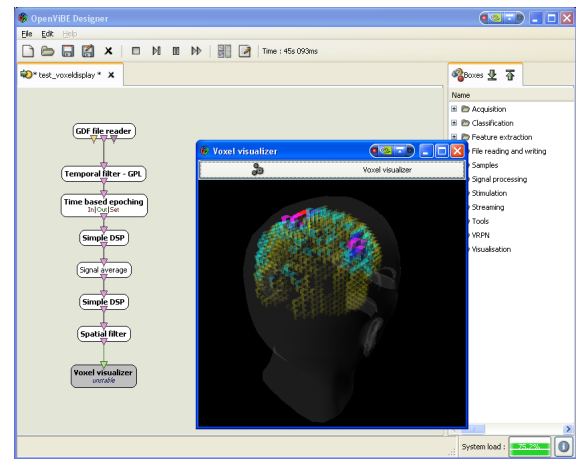
OpenViBE [34] z kolei jest europejskim (francuskim) projektem rozpoczętym w 2006 roku, nakierowanym na interakcję aktywności mógu w rzeczywistości wirtualnej – jak sama nazwa wskazuje „Open Virtual Brain Environment”. Licencja LGPL [37] pozwala na częściowe zamykanie rozwiązań stworzonych z użyciem tych narzędzi zapewniając jednocześnie kod źródłowy do systemu (część komponentów objęta jest licencją GPL [36] i nie może być zamknięta). Oprogramowanie działa w systemach Windows i Linux, może być skompilowane za pomocą GCC a więc jest duża szansa na współpracę z innymi systemami operacyjnymi w razie potrzeby.

Ponieważ program może działać w systemach innych niż Windows, jest również pozbawiony wad i problemów jakie napotkaliśmy podczas uruchamiania naszych prostych urządzeń pomiarowych opartych o port szeregowy, przez co system Windows traktował nasze urządzenie jako mysz systemową często uniemożliwiając wykonanie pomiarów i doświadczeń. Dlatego właśnie podczas prototypowania ważne jest aby mieć kontrolę nad sprzętem i unikać systemów, które rzekomo są mądrzejsze od użytkownika.

Projekt rozwija się prężnie i zapewnia wszystkie podstawowe funkcjonalności akwizycji, przetwarzania, ekstrakcji, klasyfikacji, prezentacji wyników (aplikacje, wi-



Rysunek 20: P300 Speller w OpenViBE (aplikacja wypisująca tekst myślami).



Rysunek 21: Wizualizacja aktywności źródeł sygnału w mózgu.

zualizacje) a nawet połączenia z zewnętrznymi aplikacjami przez Virtual Reality Peripheral Network.

Lista obsługiwanych urządzeń EEG jest na prawdę imponująca i zawiera ponad 11 rodzin urządzeń, przy czym kolejne są w trakcie opracowania. Można śmiało powiedzieć, że jest to wschodząca gwiazda w europejskich badaniach nad otwartą architekturą BCI, a dzięki swoim możliwościom i całkowicie otwartej architekturze przewyższa nawet BCI2000 (objęte mimo wszystko restrykcyjną licencją).

W naszym Kole Naukowym Cybernetyki są już osoby zainteresowane dokładnym zgłębieniem możliwości systemu OpenViBE, a więc w najbliższym czasie można spodziewać się kolejnych rezultatów i wyników badań – potrzebujemy jednak więcej sprzętu pomiarowego (głównie wielokanałowe EEG z odpowiednimi elektrodami) aby móc wykonać pomiary.

2.2 Sztuczne sieci neuronowe w analizie sygnału EEG (Krzysztof Chojnowski, Janusz Frączek, Grzegorz Leszek)

2.2.1 Cel pracy

Celem tej części projektu była próba samodzielnego stworzenia i wykorzystanie sztucznych sieci neuronowych do analizy zebranego wcześniej sygnału EEG. Analiza ta polegała na klasyfikacji sygnału, w zależności od stanu w jakim znajdowała się osoba, poddana badaniu EEG.



Rysunek 22: Pomiar EEG stanowi wejście dla sztucznej sieci neuronowej.

Do tego zadania zastosowano podstawową strukturę sieci neuronowych typu MLP, czyli wielowarstwowego perceptronu. Sieć trenowano metodą wstecznej propagacji błędów z wykorzystaniem algorytmu największego spadku.

2.2.2 Wprowadzenie

Sztuczna sieć neuronowa jest strukturą wzorowaną na budowie ludzkiego układu nerwowego. Składa się ona z połączonych ze sobą sztucznych komórek nerwowych – neuronów. Wykorzystując model matematyczny, przetwarza ona sygnały wejściowe, dając w rezultacie odpowiedź na swoim wyjściu.

Obecnie używa się wiele modeli sztucznych neuronów, wzorowanych na rzeczywistych komórkach nerwowych. Każdy z modeli odwzorowuje pewne cechy strukturalne i funkcjonalne ludzkich neuronów.

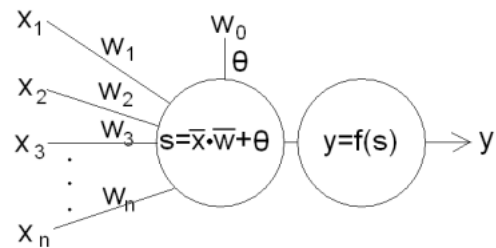
Sztuczne sieci neuronowe, są zdolne do odwzorowywania złożonych funkcji, dzięki nieliniowemu charakterowi neuronów. Ich największą zaletą jest możliwość uczenia się. Dzieje się to przez gromadzenie danych, pokazujących, jak wygląda poszukiwana zależność. Na tej podstawie sieć automatycznie konstruuje nieznane modele.

W zależności od typu rozwiązywanego problemu wykorzystuje się różne algorytmy uczenia sieci neuronowej.

Mogą to być na przykład algorytmy gradientowe do zadań predykcji i klasyfikacji, lub algorytm WTA do celów grupowania danych.

2.2.3 Model neuronu

Sztuczny neuron jest podstawową jednostką każdej sztucznej sieci neuronowej. Podobnie jak neuron rzeczywisty, jego sztuczny odpowiednik może posiadać wiele wejść i jedno wyjście. Jego zadanie, w ogólnym przypadku, polega na. pobraniu danych wejściowych, obliczeniu na ich podstawie liniowego pobudzenia neuronu, obliczeniu wartości funkcji aktywacji i podaniu odpowiedzi na swoim wyjściu. W pracy wykorzystano model neuronu przedstawiony schematycznie na rysunku 23.



Rysunek 23: Model pojedynczego neuronu. s_i – pobudzenie neuronu, $f(s)$ – funkcja aktywacji, x_n – wejścia neuronu, w_n – wagi, θ – polaryzacja.

Na wejścia neuronu podawany jest wektor x . Następnie wyznaczane jest wyjście części liniowej neuronu poprzez obliczenie iloczynu skalarnego wektora wejściowego x , z wektorem wag w oraz zsumowanie tego iloczynu z wartością progu θ , zwanego także polaryzacją. Suma ta podawana jest na wejście funkcji aktywacji. Wartości wag i polaryzacji zmieniają się podczas uczenia neuronu.

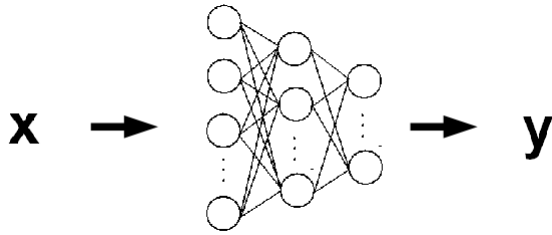
Funkcja aktywacji w perceptronie pełni zwykle rolę przekształcenia nieliniowego. W niniejszej pracy wykorzystano neurony o liniowej (równanie 1) oraz sigmoidalnej (równanie 2) funkcji aktywacji.

$$f(x) = x \quad (1)$$

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

2.2.4 Struktura sztucznej sieci neuronowej

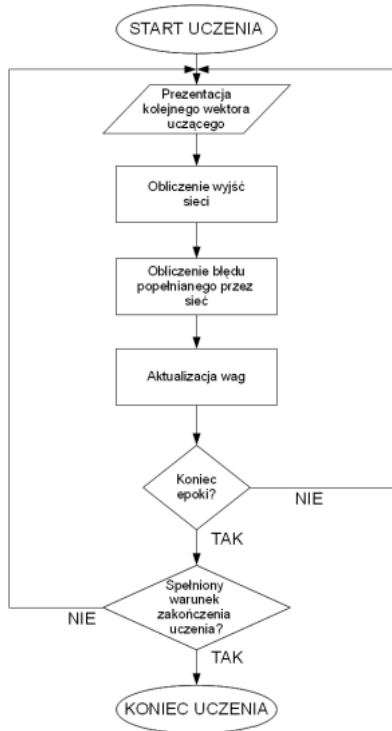
Do realizacji zadania wykorzystano strukturę sieci przedstawioną na rysunku 24. Warstwy ukryte składały się z neuronów sigmoidalnych, natomiast na wyjściu sieci znajdowały się neurony liniowe, których wyjścia stanowiły wyjścia sieci neuronowej.



Rysunek 24: Sieć wielowarstwowa o systemie połączeń zupełnych typu feed-forward.

2.2.5 Uczenie sieci

Sieć neuronowa była uczona z nauczycielem, metodą wstecznej propagacji błędów, której algorytm przedstawia rysunek 25 i 26, natomiast modyfikacja wag zachodziła według algorytmu najszybszego spadku (równanie 3).



Rysunek 25: Schemat blokowy algorytmu wstecznej propagacji błędów – algorytm przyrostowy.

$$w_{ij}^k(n+1) = w_{ij}^k(n) + 2\eta\delta_i^k(n)x_j^k(n) \quad (3)$$

, gdzie:

- w_{ij}^k – waga j-tego wejścia i-tego neuronu w warstwie k-tej,
- η – współczynnik uczenia przyjmujący wartości z zakresu (0; 1),

- x_j^k – oznacza j-te wejście neuronu w warstwie k-tej.

Wartość δ_i^k jest wyznaczana następująco:

$$\delta_i^k(n) = \varepsilon_i^k(n) f'(s_i^k(n)) \quad (4)$$

, gdzie:

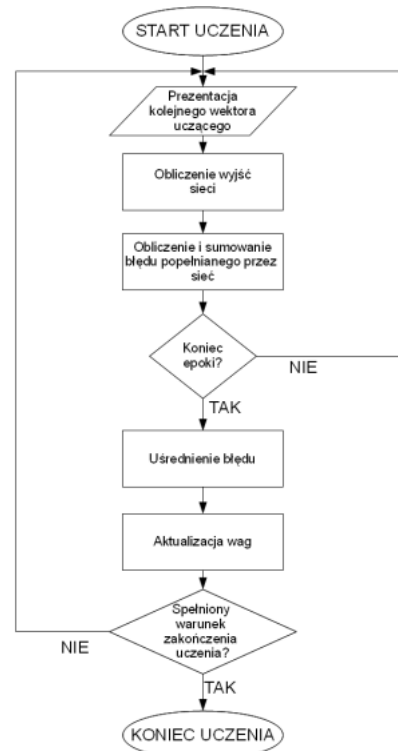
- s_i^k – wyjście liniowe neuronu i-tego, warstwy k-tej,
- $f'()$ – pochodna funkcji aktywacji.

Wartość ε , zależy od numeru analizowanej warstwy i jest wyznaczana inaczej w przypadku warstw ukrytych i warstwy wyjściowej:

$$\varepsilon_i^L(n) = d_i(n) - y_i(n) \quad (5)$$

$$\varepsilon_i^k = \begin{cases} \varepsilon_i^L(n) & , k = L \\ \sum_{m=1}^{N_{k+1}} \delta_m^{k+1}(n) w_{mi}^{k+1}(n) & , k < L \end{cases} \quad (6)$$

W algorytmie wstecznej propagacji błędów kluczową rolę spełnia współczynnik uczenia. Jeżeli przyjmuje on zbyt dużą wartość, minimum globalne funkcji błędów może nie zostać nigdy osiągnięte, natomiast wartość zbyt mała prowadzi do znacznego wydłużenia czasu potrzebnego na uczenie sieci, lub do zatrzymania algorytmu w minimum lokalnym.

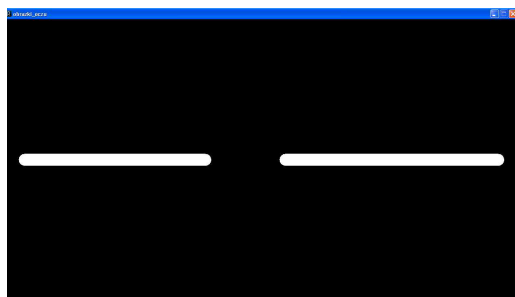


Rysunek 26: Schemat blokowy algorytmu wstecznej propagacji błędów – algorytm grupowy.

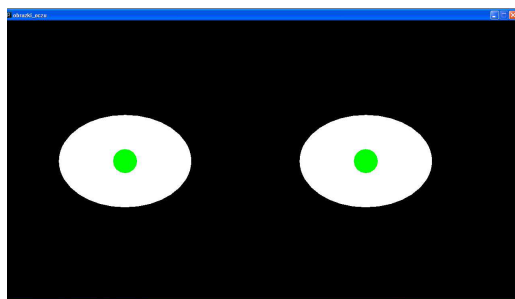
Jako danych uczących używano wyników transformaty Fouriera przebiegu czasowego o określonej długości. Sieć posiadała 35 wejść, co odpowiadało trzydziestu pięciu prążkom widma częstotliwościowego z zakresu od 0 Hz do 50 Hz. Sygnały zbierane były w różnych stanach osoby poddanej badaniu, lub po zaprezentowaniu jej różnych bodźców wzrokowych.

2.2.6 Wyniki i wnioski

W wyniku przeprowadzonych dotychczas prac stworzono sieć neuronową będącą w stanie wykryć, czy oczy badanej osoby są otwarte, czy zamknięte na podstawie sygnału EEG zebranego z potylicy. Sieć neuronowa pełni wówczas rolę klasyfikatora sygnałów.



(a) oczy zamknięte



(b) oczy otwarte

Rysunek 27: Zrzut ekranu z autorskiego programu sztucznej sieci neuronowej wykrywającej otwarcie lub zamknięcie oczu na podstawie sygnału EEG pobieranego w czasie rzeczywistym od pacjenta.

Stwierdzono, że niezbędne jest uczenie sieci neuronowej indywidualnie dla każdej z badanych osób, aby możliwa była poprawna klasyfikacja sygnału.

W dalszej części prac planowane jest rozszerzenie zastosowania sieci neuronowych na klasyfikację sygnałów EEG będących odpowiedzią na prezentowane bodźce wzrokowe oraz słuchowe.

2.3 Tworzenie sterowników własnych urządzeń w Matlab (Tomasz Cedro)

2.3.1 Wstęp

Matlab [21] jest jednym z najpotężniejszych komputerowych narzędzi obliczeniowych stosowanych we współczesnej nauce. Poza prostotą użycia oferuje nieskończone możliwości rozbudowy za pomocą tzw. toolbox'ów. Matlab może być wykorzystany w niemal każdej dziedzinie nauki, nam jednak przyda się do budowania i weryfikacji teoretycznych modeli opisujących rzeczywiste zjawiska nas interesujące, głównie związanych z sygnałami EEG. Aby jednak to było możliwe musimy nauczyć się pozyskiwać dane z fizycznych urządzeń do wirtualnego świata Matlab, czy to za pomocą istniejących już narzędzi, czy raczej własnych sterowników i programów. Celem tego projektu jest rozpoznanie dostępnych już metod pozyskiwania danych, ich rozbudowy lub tworzenia alternatywnych rozwiązań.

Warto również wspomnieć, że Matlab nie jest jedynym programem służącym do modelowania matematycznego – dostępne są również jego darmowe odpowiedniki Open-Source jak Octave [22] czy SciLab [23] – choć nie tak potężne i bogate w możliwości rozbudowy, stanowią dobrą podstawę do legalnej pracy badawczej, a niniejszy opis odnosi się również do tych programów. Nie każdy, zwłaszcza student, może pozwolić sobie na zakup oryginalnej licencji, dodatkowych modułów programowych i sprzętowych do własnych badań. Jak się z resztą okaże, takie drogie i komercyjne rozwiązanie nie zawsze są rozwiązaniem problemu badawczego – a w tym konkretnym przypadku jest to konstrukcja własnego systemu pomiarowego i weryfikacji jego pomiarów, a także dalsza obróbka danych.

2.3.2 Data Acquisition Toolbox

Data Acquisition Toolbox [24] jest zestawem narzędzi do transferu danych z przyrządów pomiarowych. Niestety działa tylko pod windows, jako że bazują na COM i ATL.

Bardzo dobrym dokumentem dla developerów chcących napisać sterownik dla własnego urządzenia jest dokument Data Acquisition Toolbox Adaptor Kit User's Guide [25].

2.3.3 Instrument Control Toolbox

Instrument Control Toolbox [26] zapewnia komunikację z urządzeniami pomiarowymi, głównie po magistrali GPIB (IEEE-488/IEC-625), VISA, TCP/IP. Pozwala sterować urządzeniami pomiarowymi. Pozwala wysyłać oraz odbierać dane binarne oraz tekstowe, synchronicznie oraz blokowo. Zapewniony zestaw m-funkcji użytkownika wykorzystuje do wymiany danych

adaptery (Interface Driver Adaptors), a te z kolei bazują na zewnętrznych sterownikach sprzętu dostarczonych przez producenta danego urządzenia. Umożliwia komunikację/sterowanie na poziomie pojedynczych funkcji/instrukcji oraz bardziej automatycznych obiektów (Interface Objects) posiadających odpowiednie atrybuty, działających zdarzeniowo.

Warstwę komunikacyjną tworzą adaptory (Interface Driver Adaptors) dostarczające dane do m-funkcji użytkownika. Adaptory do komunikacji z urządzeniami wykorzystują sterowniki (drivers). Pełną listę adapterów oraz sterowników obsługiwanych przez system, wraz z dodatkowymi informacjami można uzyskać wywołując komendę `instrhwinfo`. W standardzie dostępne są następujące adaptory:

- GPIB
- Serial (port) (także Linux)
- TCP/IP (także Linux)
- UDP (także Linux)
- VISA

Obsługiwane są dwa typy sterowników:

- VXIplug&play Drivers - (`instrhwinfo visa`)
- IVI Drivers - (`instrhwinfo ivi`)

Matlab Instrument – pełni rolę "tłumacza". Pozwala na pewną translację danych odbieranych oraz wysyłanych do urządzenia i jest włączany pomiędzy zestaw funkcji dostępnych użytkownikowi a funkcje komunikacyjne urządzenia. Można wykorzystać 'Instrument Driver Editor' do tworzenia Driverów o podobnej strukturze działania w przypadku gdy potrzebujemy sterownika podobnego do istniejącego już.

Instrument Control Toolbox nie zapewnia sterowników do urządzeń poza standardem obsługiwanych magistral. (Przypadkowo?) Nie ma frameworku ani instrukcji umożliwiających dodawanie własnych sterowników.

2.3.4 Własne sterowniki

Powstaje pytanie czy można dowolnie rozbudować środowisko o własne sterowniki własnego urządzenia, które mogłyby pracować niezależnie od platformy na której uruchamiany jest Matlab? Odpowiedź jest oczywiście twierdząca :-)

2.3.5 Jak stworzyć własny sterownik urządzenia w Matlabie?

Tworząc nowe urządzenia czy systemy o dużym stopniu zaawansowania, musimy posiadać mechanizm weryfikacji wyników, którym może być MATLAB. Aby

przesyłać dane do matlaba, można wykorzystać gotowe toolbox'y które obsługują jedynie standardowy sprzęt lub napisać własny sterownik – co daje większe możliwości w dostosowaniu metody do potrzeb, nie wymaga też żadnych dodatkowych licencji.

Matlab nie tylko potrafi generować kod wykonywalny na podstawie swoich m-plików (toolbox compiler), ale również wykorzystywać zewnętrzne programy czy biblioteki (o czym można poczytać w dziale "External Interfaces" podręcznika pomocy) do wykonywania potrzebnych operacji. Niemalże nieograniczone możliwości wymiany danych oraz sterowania zewnętrznymi aplikacjami dają:

- obsługa zewnętrznych bibliotek dynamicznych (so/dll)
- wywoływanie zewnętrznych procedur języka C lub Fortran z poziomu plików MEX
- tworzenie od podstaw własnych plików MEX w języku C lub Fortran
- eksportowanie/importowanie danych przy użyciu plików MAT
- wykorzystywanie elementów składowych matlaba w programach C lub Java

Konstruktorów interesuje głównie stworzenie zestawu funkcji umożliwiających konfigurację oraz wymianę danych z urządzeniem, co można rozwiązać na dwa sposoby - plik MEX (dedykowany jedynie dla matlaba) lub bibliotekę dynamiczną. MEX to zestaw dynamicznie linkowanych procedur, stworzonych w języku C lub Fortran, które mogą być wywołane bezpośrednio w wiersza poleceń interpretera matlaba. Stworzenie biblioteki dynamicznej daje możliwość wykorzystania jej również w innych programach - jest to bardziej uniwersalne a więc preferowane opakowanie naszej funkcjonalności. Dokładny opis znajduje się w dziale "MATLAB Interface to Generic DLLs" podręcznika pomocy. Interfejs matlaba do bibliotek dynamicznych

Funkcje umieszczone w bibliotekach dynamicznych mogą być załadowane do pamięci matlaba i stać się bezpośrednio dostępne z wiersza poleceń interpretera. W większości przypadków konwersja typów odbywa się automatycznie (na format matlaba). Można również używać bibliotek dynamicznych stworzonych w innych językach niż C, ale interfejs biblioteki musi być zgodny z językiem C.

2.3.6 Obsługa bibliotek dynamicznych

Otwieranie biblioteki: Aby mieć dostęp do danych zawartych w bibliotece, należy ją wczytać poleceniem `loadlibrary('nazwa_biblioteki', 'nazwa_pliku_naglowkowego')`, gdzie:

`nazwa_biblioteki` to nazwa pliku z biblioteką dynamiczną (so/dll), w której zawarte są interesujące nas funkcje, `nazwa_pliku_naglowkowego` to nazwa wymaganego pliku nagłówkowego zawierającego prototypy funkcji zawartych w bibliotece.

Zamykanie biblioteki: Aby zamknąć wczytaną uprzednio bibliotekę należy wykonać polecenie `unloadlibrary nazwa_biblioteki`.

Zawartość biblioteki i jej funkcje składowe: Aby wyświetlić funkcje zawarte w bibliotece należy wykonać polecenie `libfunctions('nazwa_biblioteki')`. Wynikiem jest tablica tekstowa z nazwami funkcji, `libfunctionsview('nazwa_biblioteki')` to nazwy funkcji wyświetlone są w tabelce w nowym oknie. Do wspomnianych funkcji można dodać przełącznik `-full`, który wyświetli dodatkowe informacje o funkcjach, na przykład ich parametry wraz z typami danych.

Wywoływanie funkcji: Do wykonywania funkcji bibliotecznych służy funkcja `calllib('nazwa_biblioteki', 'nazwa_funkcji', arg1, ..., argN)`, gdzie `nazwa_biblioteki` to nazwa otwartej uprzednio biblioteki, `nazwa_funkcji` to nazwa interesującej nas funkcji, `arg1, ..., argN` to lista parametrów funkcji. Pełną specyfikację funkcji można znaleźć w podręczniku pomocy.

2.3.7 Przykład

Przykład składa się z trzech funkcji. Dwie z nich zwracają łańcuch znaków tekstowych jako wynik wykonania, a trzecia służy do dodawania dwóch liczb. Funkcje te mogą łatwo zostać zastąpione zestawem instrukcji sterujących rozwijanym urządzeniem.

Plik nagłówkowy `test.h`:

```
char* test();
char* test2();
int test_add(int a, int b);
```

Plik źródłowy `test.c`:

```
#include "test.h"

char* test(){
    return "test function 1 result\n";
}

char* test2(){
    return "test function 2 result\n";
}

int test_add(int a, int b){
    return a+b;
}
```

Kompilacja: `gcc -shared -o test.so test.c`

Uruchomienie w matlabie: Zmieniamy aktualny katalog na ten, w którym znajduje się nasza skompilowana biblioteka, a następnie wykonujemy:

```
>> loadlibrary test.so test.h
>> calllib('test', 'test')

ans =
test function 1 result

>> calllib('test', 'test2')

ans =
test function 2 result

>> calllib('test', 'test_add')
??? Error using ==> calllib
No method with matching signature.

>> calllib('test', 'test_add', 1, 2)

ans =
    3

>> unloadlibrary test
```

2.3.8 Podsumowanie

Przedstawiona metodologia stwarza możliwość opracowania, realizacji i weryfikacji wybranego modelu matematycznego w bardziej zaawansowanych pracach nad Brain Computer Interface, ale także szeroko pojętym modelowaniem matematycznym i przetwarzaniem sygnałów (niekoniecznie biologicznych) pochodzących z urządzeń własnej konstrukcji. Rozwiązanie zwalnia z obowiązku zakupu kosztownej aparatury pomiarowej i licencji programowych dramatycznie redukując koszty badań. Co więcej, zastosowanie otwartych narzędzi OpenSource, kosztem nieco większego nakładu pracy i czasu, daje dokładną kontrolę nad tworzoną rozwiązaniem, wymusza znajomość szczegółów. W rezultacie powstaje kompletne i spójne rozwiązanie bardziej świadomego konstruktora i badacza aniżeli jest to w przypadku rozwiązań typu „kliknij-i-zapomnij”.

2.4 Brain Computer Interface Open Protocol (Tomasz Cedro)

Niniejszy rozdział opisuje pierwszy darmowy i w pełni otwarty protokół wymiany informacji z urządzeniami typu Brain Computer Interface. Ponieważ istnieje wiele ośrodków badawczych na świecie pracujących już nad rozwiązaniami BCI stosując zamknięte i komercyjne rozwiązania, a istniejące dotychczas otwarte protokoły projektu OpenEEG nazwane P2 i P3 pozwalają właściwie na podstawową transmisję danych wyłącznie

od urządzenia pomiarowego do komputera, postanowiłem stworzyć *Brain Computer Interface Open Protocol*.

Protokół tego typu pozwoli na jego implementację w najróżniejszych urządzeniach pomiarowych, pozwalając na dwukierunkową transmisję i pełne sterowanie badanym systemem, a przede wszystkim na spójną wymianę danych pomiędzy różnymi systemami pomiarowymi i zespołami badawczymi w celu porównania wyników czy dalszego przetwarzania. Projekt został rozpoczęty w ramach mojej pracy dyplomowej (projekt i prototyp) inżynierskiej [16] i będzie zakończony w ramach pracy dyplomowej magisterskiej [?] w tym roku (pełna implementacja).

BCIOP będzie miał formę biblioteki stworzonej w języku programowania C, ponieważ w chwili obecnej język C wraz z odpowiednim kompilatorem potrafi stworzyć kod wykonywalny na praktycznie każdy procesor, w tym najprostszy mikrokontroler 8-bitowy, w związku z czym będzie możliwość łatwej implementacji w systemach doświadczalnych i eksperymentalnych. Bibliotekę wzbogaci zestaw narzędzi wysokiego poziomu stworzonych w języku Python, ze względu na swoją przenaszalność, a same narzędzia pomogą w takich procesach jak wyszukiwanie błędów transmisji, konwersja danych, czy po prostu analiza strumienia danych.

Niedawno pojawiła się możliwość rozdzielenia funkcji biblioteki na absolutnie niezbędne do przesłania danych (dla najprostszych mikrokontrolerów, których pamięć programu nie przekracza kilku kilobajtów), oraz na pełną implementację (dla rozbudowanych systemów wbudowanych, których funkcje będą jednakowe jak dla komputera sterującego, dzięki czemu sterowanie prostszymi urządzeniami będzie mogło odbywać się zdalnie przez system wbudowany i sieć komputerową). Wprowadzając odpowiednią organizację plików nagłówkowych oraz zawartych w nich symboli, można będzie również wykorzystać BCIOP jako uniwersalny protokół sterujący urządzeniami prototypowymi, niekoniecznie pomiarowymi, poprzez zmianę znaczenia poszczególnych poleceń (Tagów).

Szczegółowe informacje dotyczące konstrukcji protokołu i pakietów znajdują się w manuskrypcie mojej pracy dyplomowej [16], zapraszam również na stronę internetową projektu <http://bciop.sf.net> wkrótce pojawi się tam zaktualizowany opis i kod źródłowy biblioteki.

2.4.1 Konstrukcja Protokołu

Aby zmniejszyć narzut danych kontrolnych względem danych pomiarowych protokół został oparty o konstrukcję TLV, czyli każdy pakiet składa się z trzech części Tag–Length–Value. Tag determinuje funkcję pakietu, Length określa długość danych w oktetach (paczkach 8-bitowych), a Value to przesyłane dane.

BCIOP PACKET		
TAG (T)	LENGTH (L)	VALUE (V)
1 octet length	1 octet length	L octets length

2.4.2 Możliwości protokołu

BCIOP nie dba o adresację lub trasę pakietów – takie zadania powinny załatwić warstwy niższe (np. TCP/IP, USB, itp.). BCIOP może być zastosowany do eksperymentów i prototypowania, ale także w wersji produkcyjnej urządzeń.

Każdy pakiet może być wyprzedzony przez 10101010b czyli 0xAA lub osobny pakiet synchronizujący, który ma pomóc w synchronizacji i wyszukiwaniu błędów.

Format danych może być zmieniony w trakcie pracy, aby ułatwić ich (de)kompozycję i rzutowanie do odpowiedniego typu w programie końcowym lub sterowniku.

Przewidziane są pakiety kontrolne sygnalizujące błąd transmisji lub nieprawidłowy stan urządzenia. W przypadku desynchronizacji transmisji urządzenie zasygnalizuje taki fakt odpowiednim pakietem synchronizacji – to może również pomóc i ułatwić automatyczne wykrywanie prędkości transmisji lub ponowną synchronizację łącza.

2.4.3 Przebieg transmisji

Aby rozpocząć połączenie, host przesyła pakiet INIT-REQUEST z odpowiednim żądaniem do urządzenia, a urządzenie odpowiada podobnym pakietem INIT-RESPONSE zawierającym podstawowe informacje o możliwościach transmisji (np. obsługiwanej wersji protokołu). W następnej kolejności format danych (pakiet DATA_FORMAT) i znacznik czasu (pakiety TIMESTAMP_FORMAT oraz TIMESTAMP_QUANT) powinien być odebrany od urządzenia w celu poprawnej interpretacji otrzymanych danych.

W zależności od sprzętu użytego do wykonania urządzenia pomiarowego występować mogą różne formaty danych preferowanych przez dany sprzęt (CPU, ADC, ..) – próbka może składać się z 1..4 oktetów, ze znakiem lub bez znaku, kodowanych w systemie NKB, U2, Gray, itp. Można więc użyć rzutowania danych albo do typu wymaganego przez sprzęt pomiarowy, albo nastawić sprzęt pomiarowy do wysyłania danych w formie wygodnym dla systemu odbiorczego (o ile sprzęt na to pozwoli).

Znacznik czasu jest niezbędny do poprawnej rekonstrukcji sygnału z próbek – istnieje możliwość ustalenia jego wartości i częstotliwości – każda próbka ma więc swój numer, oddzielony ustaloną chwilą czasu, co niweluje problem nieprawidłowej kolejności ich odbioru lub retransmisji.

Polecenia podzielone zostały na dwie główne kategorie:

- Konfiguracja – najstarszy bit Tag ustawiony na „1” – pozwala na sterowanie urządzeniami
- Transfer Danych – najstarszy bit Tag ustawiony na „0” – związany z transmisją danych pomiarowych

Urządzenie lub Host może przesłać pakiet **ERROR** oznaczający błąd – co może mieć związek z brakiem inicjalizacji urządzenia, błędnymi danymi, przepelnieniem bufora lub innym błędem. Przewidziany został nawet specjalny pakiet błędu zawierający komunikat tekstowy, a więc sygnalizacja sytuacji wyjątkowych nie jest ograniczona. Błędy powinny być informacyjne i nie zakłócać transmisji danych o ile nie są krytyczne. Minimalistyczne urządzenie pomiarowe może wyłącznie sygnalizować błąd krytyczny, na co odpowiednio powinien zareagować program komputera kontrolnego (Hosta). Istnieje również możliwość potwierdzania otrzymania pakietu, w celu weryfikacji poprawnej transmisji. Zaleca się jednak aby pakiety błędnie skonstruowane nie powodowały błędów urządzenia pomiarowego – taki błąd będzie wykryty przez „timeout”, a system uniknie zalewania pakietami.

3 Sprzęt

3.1 Elektroniczny system do prowadzenia treningu Neurofeedback (Janusz Frączek, Agnieszka Małkiewicz)

Celem pracy było stworzenie kompletnego, ekonomicznego systemu przeznaczonego do zbierania i obróbki sygnału EEG w czasie rzeczywistym, które znajdzie zastosowanie w treningu neurofeedback. Urządzenia powstało w Instytucie Systemów Elektronicznych jako praca inżynierska. Niniejsza praca zawiera przedstawienie owych koncepcji, opis uzyskanych wyników oraz nakreślenie dalszych perspektyw rozwojowych.

3.1.1 Istota sygnału EEG w treningu Neurofeedback

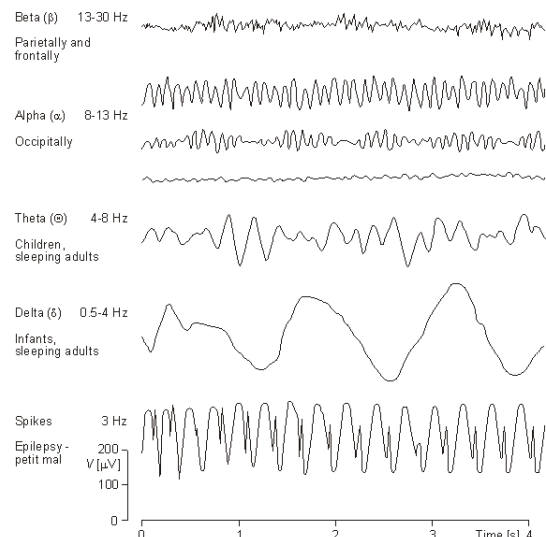
Sygnał EEG obserwuje się w postaci zmian potencjału elektrycznego na powierzchni skóry głowy. Jego źródłem jest aktywność neuronów składających się na skomplikowaną sieć – korę mózgową. Procesy myślowe można zaobserwować, jako rozchodzące się w całej korze mózgowej impulsy elektryczne. Zmiany potencjału elektrycznego, jako wynik wyżej wymienionych aktywności, zachodzą również na powierzchni skóry głowy. Ich amplituda jest bardzo mała – poniżej 100 [μV]. Nazywamy je sygnałem EEG.

Amplituda sygnału EEG spada wraz ze wzrostem częstotliwości, a obserwowana częstotliwość rośnie

wraz z przemieszczaniem się instrumentu badawczego od potylicy, ku przodowi czaszki. Medycyna dokonała wyróżnienia, pod względem częstotliwości, następujących typów fal mózgowych [11]:

- delta – częstotliwość poniżej 4 [Hz]. Ich obecność w stanie czuwania oznacza dysfunkcję mózgu, a w trakcie snu – jego bardzo głęboką fazę. Obserwuje się korelację ich występowania z wydzielaniem hormonu wzrostu.
- theta – częstotliwość 4 – 7 [Hz]. Najsilniejsze amplitudy rejestruje się w odprowadzeniach śródkowych i skroniowych. Charakteryzują senność i wchodzenie w stan nieświadomości. Amplituda rośnie, wraz z wchodzeniem w coraz głębszą, początkową fazę snu.
- alfa – częstotliwość 8 – 12 [Hz]. Ten rytm uwidacznia się najlepiej w stanie rozluźnienia i zamknięcia oczu. Największe amplitudy rejestruje się w okolicach potylicznych.
- beta – częstotliwość powyżej 13 [Hz]. Obejmują swym występowaniem duży zasięg, głównie jednak płaty czołowe. Towarzyszą codziennym aktywnościom mózgu, takim jak komunikacja werbalna, intensywne myślenie logiczne, czy ogólne pobudzenie. W technice biofeedback wyróżnia się trzy podrodzaje tej aktywności: SMR (13 – 15 [Hz]), beta1 (15 – 18 [Hz]) i beta2 (powyżej 18 [Hz]).

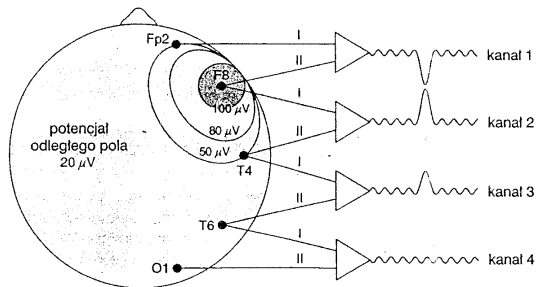
Typowe odmiany rytmów fal mózgowych prezentuje rysunek 28 [8]:



Rysunek 28: Ilustracja reprezentatywnych przebiegów EEG poszczególnych typów [8].

3.1.2 Przetwarzanie sygnału

Jako nieinwazyjną metodę rejestracji stosuje się elektroencefalografię (EEG). Do skóry głowy pacjenta przytwierdza się, zwilżone żelem elektrody. Żel stosuje się w celu obniżenia rezystancji elektroda – skóra. Najczęściej używa się odprowadzeń dwubiegunowych, kiedy to badany jest przebieg zmian różnicy potencjałów między elektrodami położonymi blisko siebie [11].



Rysunek 29: Szkic koncepcji rejestracji EEG, z zastosowaniem dwubiegunowych odprowadzeń. Widok z góry na głowę pacjenta [11].

Zobrazowana na rysunku 29 metoda rejestracji, pozwala na częściową eliminację zakłóceń, które biorą swe źródło z sieci energetycznej lub z organizmu samego pacjenta, takie jak sygnał EKG lub artefakty pochodzące z ruchów mięśni (EMG). Niepożądane sygnały indukują się synfazowo na każdej parze odprowadzeń, a interesujący nas sygnał EEG, zbierany jest różnicowo. Miejsce podłączenia elektrod należy wybrać w zależności od tego, który fragment kory mózgowej stanowi przedmiot badań oraz które rodzaje fal należy poddać obserwacji [11].

3.1.3 Wymagania

Skonstruowane urządzenie, jako sprzęt bezpośrednio integrujący mózg z komputerem i rejestrujące sygnał o amplitudzie poniżej poziomu zakłóceń, spełnia szereg wymagań, w tym najbardziej rygorystyczne normy bezpieczeństwa [8] [10] [13] [14]:

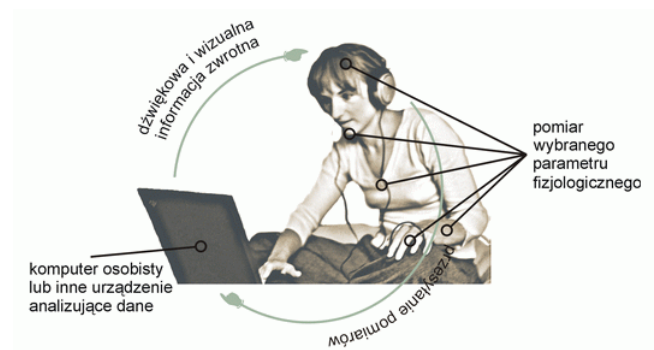
- oporność elektrod musi być mniejsza niż $5 [k\Omega]$
- rezystancja wejściowa wzmacniacza większa niż $100 [M\Omega]$
- czułość na poziomie $5 [\mu V]$
- różnicowy odbiór sygnału i jak najwyższe tłumienie składowej sumacyjnej CMRR – powyżej $80 [dB]$
- najwyższy typ ochrony pacjenta – CF („cardiac floating”). Bariera galwaniczna zabezpiecza przed

przebiegami do $5 [kV]$ i ogranicza prąd upływu pacjenta do $10 [\mu A]$, a w wypadku awarii do $50 [\mu A]$.

- 12 bitowy przetwornik ADC
- częstotliwość próbkowania $80 [Hz]$
- przynajmniej 2 kanały
- analiza sygnału w czasie rzeczywistym
- prezentacja informacji zwrotnej w czasie mniejszym niż $0.5 [s]$ od pojawienia się reakcji bioelektrycznej mózgu

3.1.4 Przykłady zastosowań

W ogólności biofeedback to trening kontroli samoregulującego się systemu za pomocą biofizjologicznego sprzężenia zwrotnego. Ową definicję przedstawia poniższy obrazek 30:



Rysunek 30: Zasada działania biologicznego sprzężenia zwrotnego [12].

System akwizycji mierzy wybrany parametr fizjologiczny i na bieżąco prezentuje jego zmiany. Może to być, dla przykładu, powiększanie się piłki na ekranie lub skręt jadącego samochodu w określonym kierunku. W ciągu kilku kolejnych takich sesji treningowych pacjent uczy się świadomie wpływać na określony parametr fizjologiczny, taki jak impedancja skóry, temperatura ciała lub częstotliwość fal mózgowych. Kiedy parametrem jest EEG, wtedy mamy do czynienia z neurofeedback’iem.

Dany typ fal jest silnie skorelowany z konkretnym stanem umysłu. Człowiek może, do pewnego stopnia, wyćwiczyć świadomą kontrolę nad generacją określonych częstotliwości, a więc wprowadzić mózg w określony tryb pracy, taki jak intensywne logiczne myślenie lub abstrakcyjne fantazjowanie w stanie głębokiego relaksu.

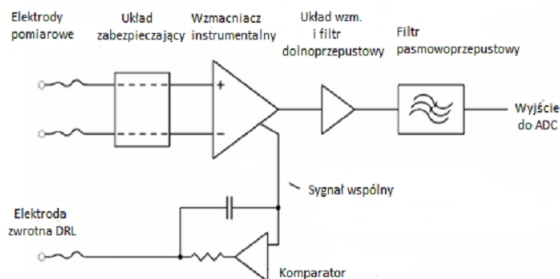
Na prawdziwości powyższych dwóch tez opiera się praktycznie cała celowość stosowania treningu neurofeedback. Pole zastosowań, jakie znajdują tego typu

urządzenia jest tak rozległe, jak mnogość sytuacji życiowych, w których wymagana jest umiejętność kontroli własnych emocji i sprawnego wejścia w pożądany stan umysłu, jak choćby rozluźnienie, albo intensywne, twórcze. Wystarczy wymienić kilka sfer, w których zajmują znaczącą pozycję [12] [8] [13]:

- sport – trening koncentracji i wyciszenia w stanie wyczerpania
- biznes – podnoszenie sprawności intelektualnej w sytuacjach stresowych, podejmowanie ważnych decyzji po uprzednim rozmyśleniu w stanie głębokiego relaksu
- kursy wydajniejszego uczenia się, zwłaszcza przed okresem egzaminów
- terapia uzależnień
- terapia nerwic oraz ADHD
- sterowanie różnymi urządzeniami, jak na przykład proteza ręki

Neurologzy często podkreślają brak skutków ubocznych – przy treningu pod okiem specjalisty oraz fakt poczucia odpowiedzialności pacjenta za wynik własnego leczenia. Chory wie, że droga do wyzdrowienia w dużej mierze zależy od włożonego przezeń wysiłku.

3.1.5 Koncepcja zbudowanego systemu



Rysunek 31: Schemat ideowy wzmacniacza biologicznego [15].

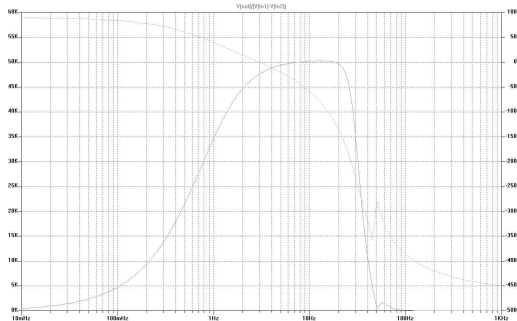
Wzmacniacz biologiczny, który wzmacnia sygnał odebrany z elektrod, zawiera w sobie podstawowe bloki funkcjonalne:

- Elektrody pomiarowe – przez nie urządzenie ma styczność z pacjentem. Działanie ludzkiego systemu nerwowego opiera się na przewodnictwie jonowym. Na elektrodach następuje przejście z przewodnictwa jonowego na elektronowe, w przewodzie.

- Układ zabezpieczający – zabezpiecza delikatny element elektroniczny, jakim jest wzmacniacz instrumentalny, przed przepięciami, jakie mogą nastąpić w momencie zetknięcia ze skórą pacjenta. Taki układ znajduje się wewnątrz scalonego wzmacniacza instrumentalnego.
- Wzmacniacz instrumentalny – zamontowany w postaci scalonej. Zdecydowano się na układ LT1168 firmy Linear Technology. Tworzy on pierwszy i najważniejszy stopień wzmacniającego. Najważniejszy, bowiem od jego parametrów najbardziej zależą szумы i zdolność do separacji zakłóceń całego urządzenia. W swojej konstrukcji składa się z dwóch stopni wzmocnienia i charakteryzuje się bardzo wysokim tłumieniem składowej sumacyjnej, która indukują się na pacjencie w postaci zakłóceń synfazowych. Tłumienie to (CMRR) powinno być wyższe od 80[dB].
- Sygnał zwrotny – wytwarzany przez wzmacniacz odwracający o wzmocnieniu kilkudziesięciu razy. Wzmacnia, wydobyty ze wzmacniacza instrumentalnego, synfazowy sygnał zakłóceń i w odwrotnej fazie kieruje go do pacjenta, poprzez elektrodę DRL. Powoduje to znaczne obniżenie poziomu niepożądanych przebiegów, indukujących się na skórze z sieci energetycznej.
- Zespół filtrów i dalsze stopnie wzmacniające – zastosowano dolnoprzepustową filtrację aktywnym filtrem, przez ustalanie potencjału na wejściu REF wzmacniacza instrumentalnego w zależności od częstotliwości sygnału. Dodano filtr pasmowooprzepustowy, który odcina sieciową składową 50[Hz]. Wszystkie filtry aktywne są jednocześnie kolejnymi stopniami wzmacniającymi. Finalnie urządzenie ma charakterystykę pasmowooprzepustową – rozciągającą się 1 – 30 [Hz], a maksymalne wzmocnienie napięciowe dochodzi do 50 000 [V/V].
- Bariera galwaniczna i zasilanie – ze względów bezpieczeństwa, na wypadek uszkodzenia, koniecznym jest oddzielenie części aplikacyjnej (mającej styczność z pacjentem) od reszty urządzenia, zasilanej z sieci elektrycznej. W zbudowanym układzie, którego część aplikacyjna czerpie zasilanie z baterii, zastosowano transoptory oraz oparte na transformatorach – izolatory cyfrowe. Jest to rozwiązanie bezpieczniejsze od zasilania poprzez bezpieczną przetwornicę. Przez barierę do komputera PC, są przekazywane jedynie sygnały cyfrowe. Pozwoliło to na uniknięcie problemów związanych z nieliniowością i wrażliwością na zmiany temperatury współczynnika wzmocnienia transoptorów – CTR.

- Mikrokontroler – jego zadanie stanowi sterowanie pracą wzmacniaczy biologicznych i transmisja danych do komputera PC. Zastosowano układy firmy Freescale (MCF51AC256).

Charakterystyka częstotliwościowa w skali liniowej wzmacniacza przedstawiona jest na rysunku 33.



Rysunek 32: Charakterystyka częstotliwościowa urządzenia w skali liniowej [15].

3.1.6 Koncepcja interfejsu użytkownika

W urządzeniu postanowiono rozgraniczyć interfejsy, na interfejs obsługi i interfejs pacjenta. Interfejs pacjenta zawiera się w ekranie komputera PC, na którym wyświetlana jest informacja zwrotna w postaci gry i przebiegów czasowych. Interfejs obsługi zawiera się w wyświetlaczu alfanumerycznym LCD i czterech przyciskach, które służą do wprowadzania nastaw oraz informowania o bieżącym stanie urządzenia.

3.1.7 Zdjęcia gotowej konstrukcji

Poniżej zamieszczono zdjęcia zbudowanego urządzenia oraz jego ciekawsze fragmenty.

3.1.8 Uzyskane wyniki pracy

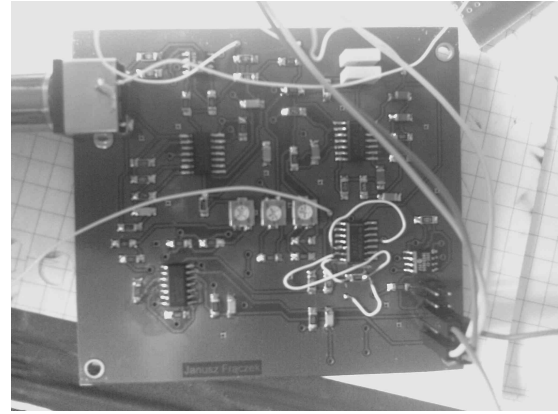
Stworzono, kompletny aparat EEG dla treningu neurofeedback, zgodne z międzynarodowymi normami bezpieczeństwa, który stanowi solidną platformę do badań nad przetwarzaniem sygnału elektroencefalograficznego.

Rozpoczęto analizę sygnału fal mózgowych za pomocą sztucznych sieci neuronowych. Obecnie system rozpoznaje, na podstawie przebiegów potylicznych, czy oczy pacjenta są otwarte, czy zamknięte.

Zintegrowano urządzenie z profesjonalnymi programami do obróbki i prowadzenia badań nad sygnałem EEG, jak Brain Bay i BCI2000.

3.1.9 Perspektywy rozwojowe i wnioski

Skonstruowane urządzenie tworzy bardzo dobry jakościowo oraz ekonomiczny system, mogący konkurować z



(a) płytka wzmacniacza



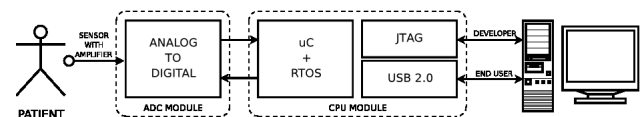
(b) układ testowy

Rysunek 33: Płytki wzmacniacza biologicznego oraz połączenie przewodem ekranowanym wzmacniacza z generatorem testowym, obydwa zaekranowane.

profesjonalnymi urządzeniami przeznaczonymi do treningu neurofeedback. Dalsze plany rozwojowe przewidyują badania nad potencjałami wywołanymi. Odniesiono już pierwsze sukcesy w tej materii – rejestrując wzrokowy potencjał wywołany, po 100 uśrednieniach.

3.2 CeDeROM BCI (Tomasz Cedro)

3.2.1 Opis systemu

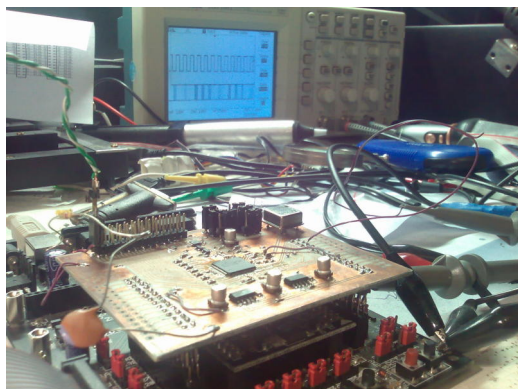


Rysunek 34: Schemat blokowy modułu akwizycji sygnałów cyfrowych opartego o mikrokontroler rodziny ARM [17].

Wykonany w 2009 roku w ramach pracy dyplomowej inżynierskiej cyfrowy moduł akwizycji sygnałów

biologicznych stanowi duży krok w rozpoznaniu otwartych metod i narzędzi, które można wykorzystać do stworzenia od podstaw cyfrowej części modularnego systemu BCI. Urządzenie oparte jest o mikrokontroler z rodziny ARM-7 z wbudowanym sprzętowym kontrolerem USB2.0, a także ośmiokanałowy przetwornik analogowo-cyfrowy wysokiej rozdzielczości 24-bit z wejściami różnicowymi. Pracą systemu steruje mikrosystem czasu rzeczywistego FreeRTOS [40], a całość została oprogramowana wyłącznie używając darmowych i otwartych narzędzi GNU [38], w szczególności ich odmianę przeznaczoną dla mikroprocesorów ARM [39].

Urządzenie może zostać podłączone do dowolnego wzmacniacza sygnałów biologicznych (kompatybilnego z wejściami ADC) w celu odebrania sygnału analogowego, który po zamianie na postać cyfrową w przetworniku zostanie zbuforowany i przesłany do komputera przez magistralę USB. W komputerze dane odbiera dedykowany program wykorzystujący otwartą bibliotekę libusb [41], a może to być również moduł importujący dane do środowiska modelowania numerycznego Matlab lub podobnego (patrz rozdział 2.3).



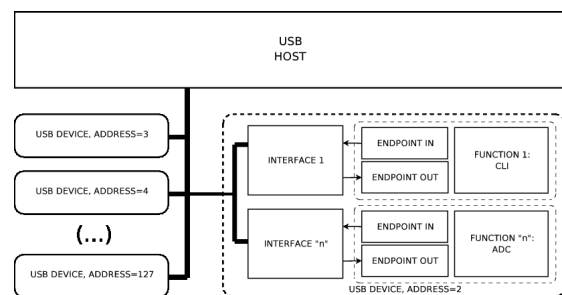
Rysunek 35: Zdjęcie prototypu modułu cyfrowego akwizycji sygnałów biologicznych [17].

3.2.2 Możliwości i ograniczenia

Założeniem organizacyjnym było wykorzystanie wyłącznie otwartych narzędzi, nawet jeśli trzeba byłoby je stworzyć, a założeniem konstrukcyjnym była budowa modularna – każdy blok funkcjonalny powinien mieć formę modułu i być łatwo wymierny zapewniając kompatybilność z pozostałymi elementami systemu. Dzięki takiej organizacji łatwe jest wzajemne porównanie różnych rozwiązań technicznych oraz ich wpływ na pozostałe elementy systemu, co jest szczególnie istotne podczas prototypowania i parametryzacji nowych rozwiązań.

Zdecydowałem się na wykorzystanie pełnego sprzętowego bloku do transmisji danych po magistrali USB 2.0 bez użycia zewnętrznych układów konwersji USB-

UART, co pozwoliło mi na dokładne zapoznanie się z budową, organizacją i funkcjonowaniem tej popularnej magistrali. Zdecydowaną zaletą takiego rozwiązania jest możliwość ukrycia wielu urządzeń logicznych wewnątrz jednego fizycznego systemu mikroprocesorowego (podział na interfejsy i źródła danych) – przedstawione urządzenie posiadało jeden port szeregowy zapewniający konsolę sterującą pracą systemu, oraz dedykowany port danych (których może być więcej). Dużą jednak wadą magistrali USB jest problem z obsługą po stronie systemu operacyjnego na komputerze, ponieważ różne systemy w różny sposób rozwiązują dostęp do zasobów USB, a z biblioteką LibUSB stanowiącą spoiwo transmisji danych przez USB pomiędzy tymi systemami również bywają problemy, ponieważ wciąż jest rozwijana zawiera błędy, a w czasie tworzenia rozwiązania (2008) nie posiadała wsparcia dla wydajnej transmisji danych.



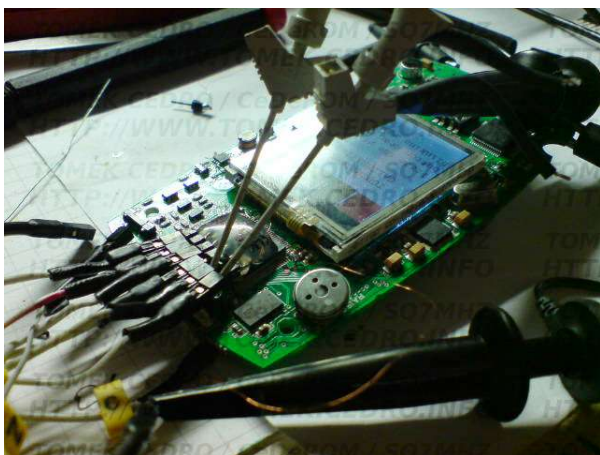
Rysunek 36: Organizacja urządzeń w magistrali USB [17].

Ciekawą cechą jaką oferują współczesne systemy mikroprocesorowe jest wsparcie dla standardu JTAG, który oferuje dostęp do niskopoziomowych zasobów CPU, pamięci i peryferiów systemu mikroprocesorowego. Za pomocą dedykowanego złącza umieszczonego na płycie systemu wbudowanego, specjalnego interfejsu JTAG, oraz oprogramowania (w moim przypadku otwarty program OpenOCD [42]) możliwe jest zaprogramowanie i debugowanie urządzenia w czasie rzeczywistym. Jest to nieoceniona możliwość dla twórcy czy developera systemu wbudowanego, szczególnie na etapie prototypowania (a nawet tworzenia tak podstawowych komponentów systemu jak bootloader).

ARM jest dużą, wciąż rozwijającą się rodziną mikroprocesorów produkowanych przez najróżniejszych wytwórców elementów półprzewodnikowych. Program stworzony w języku C może być łatwo przeniesiony z jednego urządzenia na inny, nowszy i bardziej rozbudowany układ – potrzebna jest jedynie aktualizacja sterowania peryferiów, które mogą się zmieniać pomiędzy poszczególnymi modelami układów i rekompilacja na wskazany układ. To bardzo ważne, że przenosząc program na nowe urządzenie nie trzeba go tworzyć na nowo.

Ograniczenia i problemy płynące z implementacji stosu programowego USB zarówno po stronie urządzenia jak i systemu operacyjnego komputera odbiorczego skłaniają mnie do zastosowania innego rodzaju magistrali w dalszych badaniach. Sieć komputerowa Ethernet i dobrze sprawdzony stos TCP/IP wydaje się być idealnym kandydatem, ponieważ jest stabilny i działa na każdym komputerze/systemie operacyjnym, zapewnia możliwość pracy zdalnej, a komponenty elektroniczne dające dostęp mikroprocesorowi do takiej sieci są ogólnodostępne i tanie. Sockety BSD zapewniają również możliwość tworzenia wielu urządzeń logicznych w ramach jednego systemu wbudowanego, przy czym każda usługa może być przypisana do określonego portu, a nawet może korzystać z istniejących już narzędzi przyjmując na przykład formę strony internetowej.

3.3 Serial Wire Debug (Tomasz Cedro)



Rysunek 37: Trwają prace nad stworzeniem otwartego narzędzia do pracy z układami wyposażonymi w magistralę SWD.

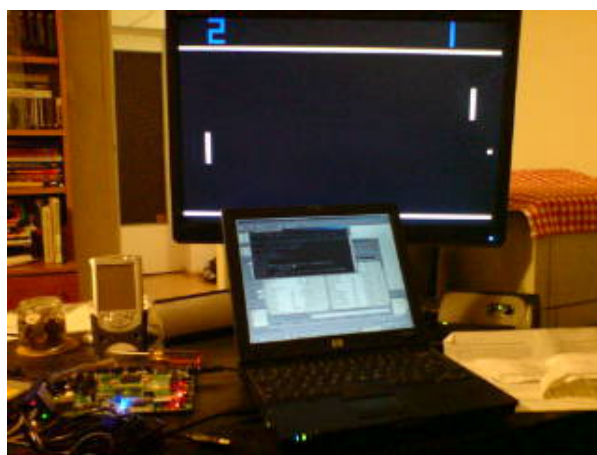
SWD (ang. Serial Wire Debug) [45] to nowa alternatywa dla niskopoziomowego dostępu do zasobów rdzenia mikroprocesora, pamięci i peryferiów, jaki zapewniał do tej pory głównie standard JTAG [44]. SWD pokonuje w pewnych kwestiach ograniczenia JTAG i jest zaimplementowany w nowych mikroprocesorach ARM-Cortex. Wciąż jednak nie ma otwartych narzędzi umożliwiających pracę z tego typu magistralą, dlatego postanowiłem je stworzyć w postaci osobnej biblioteki LibSWD [20], która zintegrowana ze znanymi otwartymi programami UrJTAG [43] i OpenOCD [42] pozwoli na tworzenie, programowanie i debugowanie urządzeń z wykorzystaniem tych nowych wydajnych a przy tym energooszczędnych układów.

Na stronie projektu [19] można śledzić postępy prac – projekt trwa od połowy 2010 i powinien zakończyć

się w połowie 2011. Pomyślnie opracowanie narzędzi pozwoli na stworzenie przenośnej wersji urządzenia NFB/BCI bazując na gotowym zestawie ewaluacyjnym Stm32Primer2 lub dedykowanym urządzeniu.

3.4 BCI PONG (Tomasz Cedro)

Celem projektu jest stworzenie od podstaw systemu BCI działającego niezależnie od komputera w układzie FPGA. Projekt jest w trakcie opracowywania i będzie stanowił bazę dla cyfrowego przetwarzania sygnałów w strukturach FPGA w czasie rzeczywistym. Wizualizacja wyników przyjmie postać pierwszej gry wideo opracowanej przez Atari w 1972 roku, PONG (rys. 38), która już jest zaimplementowana w układzie Xilinx Spartan 3A-DSP.



Rysunek 38: Gra PONG zaimplementowana w układzie FPGA sterowana będzie elektryczną aktywnością mózgu.

3.5 SITA i biofeedback (Tomasz Cedro)

System SITA wspomagający naukę języków obcych bazuje na zjawisku Biofeedbacku, a więc sprzężeniu wzrotnym wspomaganym technicznie występującym w organizmie biologicznym. Urządzenie ma formę okularów z wbudowanymi diodami świecącymi, głośniczką (wyjściem na słuchawki) i czujnikiem oddechu. Podczas sesji Biofeedback oddech synchronizuje pojawianie się i znikanie światła w okularach a także cichy dźwięk buzzera, co z kolei stanowi bodziec zewnętrzny audio-wizualny dla naszego mózgu starającego się ustabilizować oddech. Zależnie od wprawy tego typu sprzężenie prowadzi szybciej lub wolniej do stanu odprężenia sprzyjającemu zapamiętywaniu. System SITA wykorzystuje tę zależność psychologiczną i w momencie relaksu podawane są na słuchawki, wraz z synchronicznym dźwiękiem buzzera, nagrania audio lekcji językowych.



Rysunek 39: Zdjęcie urządzenia SITA.

Dokonując pomiarów EEG u osoby korzystającej z urządzenia zauważyliśmy wyraźny wzrost aktywności fal alfa obliczonej na podstawie widma Fourierskiego wyciągniętego z przebiegu EEG. Co ciekawe po krótkiej sesji relaksacyjnej nawet otwarcie oczu nie powodowało wyraźnego zmniejszenia ich amplitudy. Stan odprężenia i spokoju sprzyja więc nauce i zapamiętywaniu, byćmoże pracy twórczej, a ich ewidentnym wrogiem jest pośpiech i stres. Ważne jednak są trzy etapy procesu nauki – Inicjacja (luźne zapoznanie z materiałem), Zapamiętywanie (w stanie odprężenia), Aktywacja (powtórzenie na głos przychodzących do głowy informacji otrzymanych w stanie relaksu, utrwalenie).

Ponieważ wraz z zestawem otrzymaliśmy instrukcję jak należy tworzyć sesje mające na celu zapamiętywanie informacji, z najbliższym czasie planujemy opracować i przetestować metodę na szybką naukę alfabetu Morse'a, co powinno być łatwym w weryfikacji sposobem na sprawdzenie skuteczności metody.

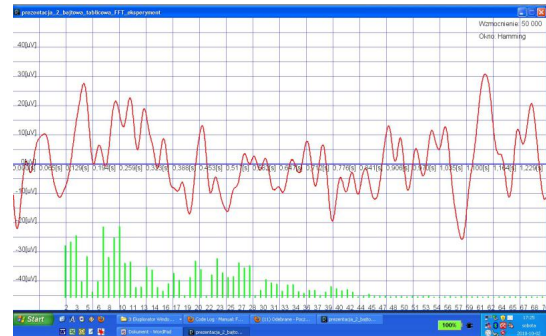
3.6 OpenEEG (Tomasz Cedro)

3.6.1 Wstęp

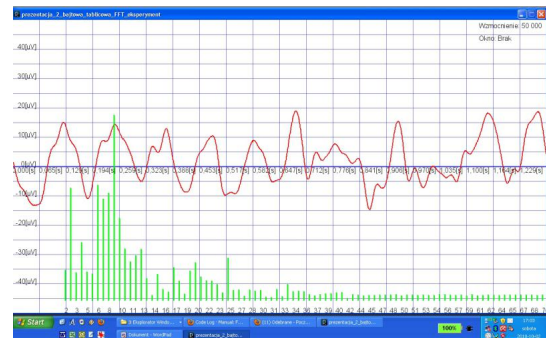
System OpenEEG zakupiony i zbudowany z funduszy przyznanych w ramach Grantu Rektorskiego PW w roku 2010 stanowił nasz punkt odniesienia w pracach badawczych nad istniejącym oprogramowaniem NFB/BCI (patrz rozdział 2.1) oraz własnych konstrukcjach sprzętu pomiarowego EEG (patrz rozdział 3.1 i oprogramowania (patrz rozdział 2.1.2).

3.6.2 Opis systemu

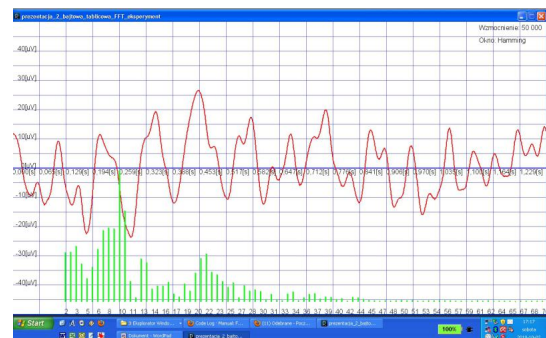
System OpenEEG [33] jest jednym z najbardziej znanych na świecie otwartych systemów EEG stosowanych do różnego rodzaju treningów Neurofeedback a nawet prostych aplikacji BCI. Bardzo prosta konstrukcja po stronie analogowej (scalony wzmacniacz biologiczny i kilka prostych wzmacniaczy operacyjnych), jak i cyfrowej (mikrokontroler 8-bitowy AVR i transmisja RS-232) sprawiają, że urządzenie jest niezawodne i łatwe



(a) bez SITA



(b) SITA, oczy zamknięte



(c) SITA, oczy otwarte

Rysunek 40: Pomiar aktywności elektrycznej mózgu w trakcie sesji Biofeedback SITA.



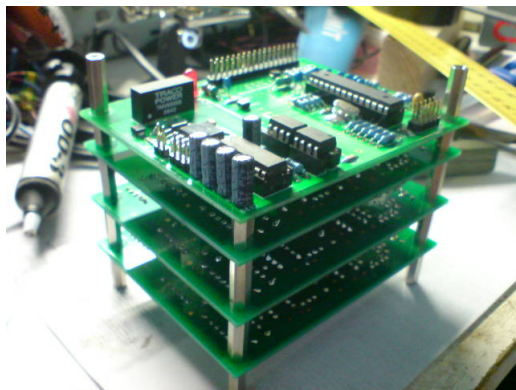
do wykorzystania nawet przez osoby nietechniczne.

Otwartość konstrukcji oznacza pełny dostęp do dokumentacji technicznej (schematów elektrycznych oraz

kodów źródłowych) oraz wsparcia użytkowników z całego świata. Grupa dyskusyjna stanowi pomoc dla osób początkujących, ale także dla tych, którzy system chcą rozbudować o nowe komponenty lub zastosować w swoim projekcie. Lista programów wspierających urządzenie OpenEEG i jego protokół komunikacji (P2 i P3) jest imponująca, dlatego pomimo prostej konstrukcji system ten stanowi punkt odniesienia w wielu projektach badawczych, szczególnie studenckich.

3.6.3 Konstrukcja

System OpenEEG składa się z części cyfrowej oraz analogowej rozdzielonej na dwie osobne płytki drukowane, przy czym płytka analogowa zawiera dwa wzmacniacze sygnałów biologicznych różnicowych od odpowiednimi filtrami pasmowymi (dolno i górnoprzepustowy), oraz odprowadzenie DRL (aktywna elektroda odniesienia), a więc cztery wejścia i jedno wyjście elektrodowe, natomiast na płycie cyfrowej znajduje się mikrokontroler z wbudowanym przetwornikiem analogowo-cyfrowym, separowany galwanicznie układ zasilania, oraz przetwornik napięcia TTL-RS232 (również separowany galwanicznie za pomocą transoptorów). Schemat elektryczny jest dosyć złożony i dostępny na stronie internetowej projektu [33].



Rysunek 41: OpenEEG, wersja 6-kanalowa w trakcie montażu.

Urządzenie może być wyposażone w maksymalnie trzy płytki analogowe, co daje łącznie sześć kanałów pomiarowych różnicowych. Jest to liczba zupełnie wystarczająca do treningu Neurofeedback, ale zbyt mała do bardziej zaawansowanych pomiarów aktywności mózgu w poszczególnych obszarach korowych – elektrody można jednak umieścić bezpośrednio nad badanym obszarem kory aby uzyskać w miarę satysfakcjonujące wyniki (na przykład badana przez nas aktywność ośrodkowa wzroku, patrz rozdział 2.2).

W ramach badań wykonaliśmy dwa urządzenia – bardziej zaawansowane o maksymalnej liczbie kanałów, oraz prostsze z jedną płytką cyfrową i analogową do

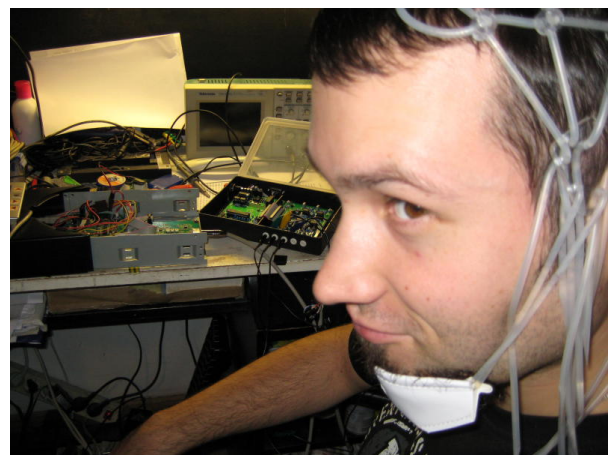


Rysunek 42: OpenEEG, wersja 2-kanalowa z podłączonym urządzeniem do kalibracji EEG.

eksperymentów i ewaluacji elektroniki.

3.6.4 Wyniki badań

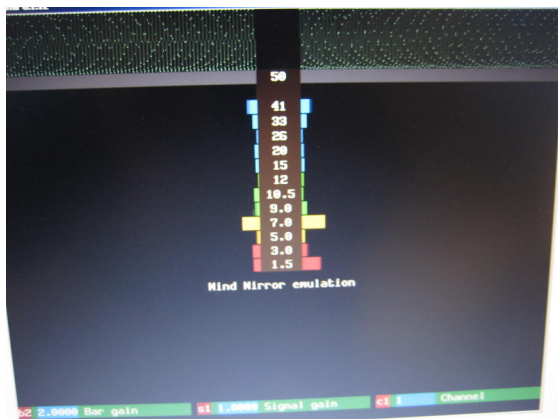
System OpenEEG pozwolił nam na praktyczne zapoznanie się z podstawami pomiarów EEG, ogólnodostępnymi rozwiązaniami programowymi i sprzętowymi systemów NFB i BCI. Podstawowe pomiary wykazały zgodność teorii z praktyką w kwestii zakresu częstotliwości pracy mózgu, oraz umożliwiły pierwsze próby wykorzystania tej aktywności do sterowania obiektami w środowisku wirtualnym (głównie poprzez świadome kontrolowanie fal Alfa, jak w treningu NFB).



Rysunek 43: Badanie EEG wykonywane za pomocą OpenEEG oraz porównawczo naszego autorskiego urządzenia.

OpenEEG ma jednak pewne wady, a dzięki temu poznaliśmy również problemy związane z wykonywaniem pomiarów, umieszczeniem elektrod i zakłóceniami. Z powodu braku filtra pasmowo-zaporowego na 50Hz (a także 60Hz w przypadku tak uniwersalnego rozwiązania) okazał się bardzo podatny na zakłócenia pocho-

dzące od sieci energetycznej, bez ekranowania pomieszczenia, obudowy i przewodów elektrod. Ograniczona liczba kanałów pomiarowych jest również pewną przeszkodą w przeprowadzeniu dokładniejszych badań aktywności poszczególnych obszarów kory mózgowej i ich korelacji z aktywnością fizyczną lub intelektualną – aktywność w korze ruchowej można wykryć zarówno podczas ruszania kończyną jak i samego myślenia o ruszaniu kończyną, jak dowodzą wyniki badań innych grup badawczych – będziemy chcieli zweryfikować te informacje w naszych dalszych badaniach.



Rysunek 44: Zakłócenia sieciowe odbierane przez system OpenEEG wyświetlone w programie EEGMIR – wyraźna przewaga składowej 50Hz w sygnale.

3.6.5 Wnioski

Budowa i uruchomienie systemu OpenEEG stworzyły realną możliwość praktycznego badania EEG oraz realizacji celów projektu BCI/NFB. Urządzenie jest stosunkowo proste i ma pewne wady, od których wolny jest prototyp naszej konstrukcji, jednak modułowa konstrukcja pozwala sądzić, że odpowiednia modyfikacja części analogowej urządzenia (głównie filtrów) poszerzy możliwość zastosowania urządzenia w warunkach domowych bez konieczności stosowania skomplikowanych metod ekranowania.

Możliwość wykonania praktycznego pomiaru EEG, nawet tak prostym urządzeniem, daje nam doświadczenie w wykonywaniu tego typu eksperymentów i badań, otwiera drzwi do rozwoju własnego oprogramowania w tematyce NFB/BCI (patrz rozdział 2.1.2, 2.2) i stanowi punkt odniesienia dla własnych prototypów sprzętu pomiarowego. Nie ukrywamy faktu, że nasze rozwiązanie już osiągnęło parametry lepsze od omawianego systemu (choćby odporność na zakłócenia i możliwość stosowania w warunkach domowych), chąc jednak rozwijać się dalej czujemy potrzebę dokładniejszej weryfikacji rozwiązań w odniesieniu do profesjonalnych komercyjnych konstrukcji.

Spis treści

1	Teoria	2
1.1	Podstawy anatomii i fizjologii ośrodkowego układu nerwowego człowieka (Ryszard Gomółka)	2
1.1.1	Budowa zewnętrzna mózgu	2
1.1.2	Kora mózgowa	2
1.1.3	Pola kory	3
1.2	Zasady aplikacji elektrod i cechy sygnału EEG (Ryszard Gomółka)	4
1.2.1	System 10–20	4
1.2.2	Podstawowe grafoelementy zapisu EEG u osoby dorosłej	6
1.2.3	Cechy charakterystyczne EEG u dzieci	6
1.2.4	Artefakty zapisu EEG	6
1.3	Elektrody pomiarowe EEG (Tomasz Cedro)	7
2	Oprogramowanie	8
2.1	Otwarte programy BCI/Neurofeedback (Tomasz Cedro)	8
2.1.1	BCI2000	8
2.1.2	Własna aplikacja użytkownika w systemie BCI2000 (Piotr Tąkiel, Tomasz Cedro)	9
2.1.3	OpenEEG NeuroServer	9
2.1.4	BrainBay	9
2.1.5	OpenViBE	10
2.2	Sztuczne sieci neuronowe w analizie sygnału EEG (Krzysztof Chojnowski, Janusz Frączek, Grzegorz Leszek)	11
2.2.1	Cel pracy	11
2.2.2	Wprowadzenie	11
2.2.3	Model neuronu	11
2.2.4	Struktura sztucznej sieci neuronowej	11
2.2.5	Uczenie sieci	12
2.2.6	Wyniki i wnioski	13
2.3	Tworzenie sterowników własnych urządzeń w Matlab (Tomasz Cedro)	13
2.3.1	Wstęp	13
2.3.2	Data Acquisition Toolbox	13
2.3.3	Instrument Control Toolbox	13
2.3.4	Własne sterowniki	14
2.3.5	Jak stworzyć własny sterownik urządzenia w Matlabie?	14
2.3.6	Obsługa bibliotek dynamicznych	14
2.3.7	Przykład	15
2.3.8	Podsumowanie	15
2.4	Brain Computer Interface Open Protocol (Tomasz Cedro)	15
2.4.1	Konstrukcja Protokołu	16
2.4.2	Możliwości protokołu	16
2.4.3	Przebieg transmisji	16



Rysunek 45: Nietypowe badanie EEG wykonywane „po godzinach” w celu weryfikacji spowolnienia pracy mózgu przez alkohol potwierdziło nasze przypuszczenia :-)

3 Sprzęt	17	[5] http://www.gtec.at/service/tips.htm
3.1 Elektroniczny system do prowadzenia treningu Neurofeedback (Janusz Frączek, Agnieszka Małkiewicz)	17	[6] http://www.isnr.org/CBCog.cfm
3.1.1 Istota sygnału EEG w treningu Neurofeedback	17	[7] http://webideas.com/biofeedback/research/tdruckman.shtml
3.1.2 Przetwarzanie sygnału	18	[8] <i>Materiały do wykładu „Elektroniczna aparatura medyczna”, dr inż. Ewa Piątkowska, PW, Warszawa 2009</i>
3.1.3 Wymagania	18	[9] <i>Materiały do wykładu „Sztuczne sieci neuronowe w medycynie”, prof. dr hab. Krzysztof Zaremba, PW, Warszawa 2010</i>
3.1.4 Przykłady zastosowań	18	[10] <i>Design and development of medical and electronic instrumentation, David Prutchi, Michael Norris, Copyright © 2005 John Wiley & Sons, Inc.</i>
3.1.5 Koncepcja zbudowanego systemu	19	[11] <i>Primer of EEG with a Mini – Atlas, A. James Rowan, Eugene Tolunsky, Elsevier Science (USA) 2003, w tłumaczeniu Urban & Partner</i>
3.1.6 Koncepcja interfejsu użytkownika	20	[12] http://www.relaksacja.pl
3.1.7 Zdjęcia gotowej konstrukcji . . .	20	[13] http://www.eeg-biofeedback.com.pl
3.1.8 Uzyskane wyniki pracy	20	[14] <i>Wprowadzenie do biologicznej i medycznej techniki pomiarowej, Karsten Meyer – Waarden, Wydawnictwa Komunikacji i Łączności, Warszawa 1980</i>
3.1.9 Perspektywy rozwojowe i wnioski	20	[15] <i>System do akwizycji sygnału EEG, Janusz Frączek, PW, Warszawa 2010</i>
3.2 CeDeROM BCI (Tomasz Cedro)	20	[16] <i>Brain Computer Interface – ARM-based signal acquisition module. Bachelor of Science Diploma Thesis, Tomasz Bolesław Cedro., PW EiTI, Marzec 2009.</i>
3.2.1 Opis systemu	20	[17] <i>Brain Computer Interface, Praca dyplomowa magisterska (w opracowaniu).</i>
3.2.2 Możliwości i ograniczenia	21	[18] http://bciop.sf.net – Brain Computer Interface Open Protocol
3.3 Serial Wire Debug (Tomasz Cedro) . . .	22	
3.4 BCI PONG (Tomasz Cedro)	22	
3.5 SITA i biofeedback (Tomasz Cedro) . .	22	
3.6 OpenEEG (Tomasz Cedro)	23	
3.6.1 Wstęp	23	
3.6.2 Opis systemu	23	
3.6.3 Konstrukcja	24	
3.6.4 Wyniki badań	24	
3.6.5 Wnioski	25	

Literatura

[1] *Anatomia Człowieka. Podręcznik dla studentów medycyny. Pod red. Janiny Sokołowskiej-Pituchowej.*, Wyd. V; Warszawa 1988 PZWL.

[2] *Atlas Anatomii Człowieka, TOM III. R.D. Sinelnikov*, Moskwa 1974 „Medycyna”.

[3] *1. Anatomia i Fizjologia Człowieka. A. Michajlik, W. Ramontowski*, Warszawa 2005 PZWL.

[4] <http://www.brainmaster.com/generalinfo/electrodeuse/eegbands/1020/1020.html>

[5] <http://www.gtec.at/service/tips.htm>

[6] <http://www.isnr.org/CBCog.cfm>

[7] <http://webideas.com/biofeedback/research/tdruckman.shtml>

[8] *Materiały do wykładu „Elektroniczna aparatura medyczna”, dr inż. Ewa Piątkowska, PW, Warszawa 2009*

[9] *Materiały do wykładu „Sztuczne sieci neuronowe w medycynie”, prof. dr hab. Krzysztof Zaremba, PW, Warszawa 2010*

[10] *Design and development of medical and electronic instrumentation, David Prutchi, Michael Norris, Copyright © 2005 John Wiley & Sons, Inc.*

[11] *Primer of EEG with a Mini – Atlas, A. James Rowan, Eugene Tolunsky, Elsevier Science (USA) 2003, w tłumaczeniu Urban & Partner*

[12] <http://www.relaksacja.pl>

[13] <http://www.eeg-biofeedback.com.pl>

[14] *Wprowadzenie do biologicznej i medycznej techniki pomiarowej, Karsten Meyer – Waarden, Wydawnictwa Komunikacji i Łączności, Warszawa 1980*

[15] *System do akwizycji sygnału EEG, Janusz Frączek, PW, Warszawa 2010*

[16] *Brain Computer Interface – ARM-based signal acquisition module. Bachelor of Science Diploma Thesis, Tomasz Bolesław Cedro., PW EiTI, Marzec 2009.*

[17] *Brain Computer Interface, Praca dyplomowa magisterska (w opracowaniu).*

[18] <http://bciop.sf.net> – Brain Computer Interface Open Protocol

- [19] <http://stm32primer2swd.sf.net> – Implementacja Serial Wire Debug w procesorach ARM Cortex.
- [20] <http://libswd.sf.net> – Otwarta Biblioteka Serial Wire Debug.
- [21] <http://www.mathworks.com>
- [22] <http://www.gnu.org/software/octave>
- [23] <http://www.scilab.org>
- [24] <http://www.mathworks.com/products/daq/>
- [25] http://www.mathworks.com/access/helpdesk/help/pdf_doc/daq/daqug.pdf
- [26] <http://www.mathworks.com/access/helpdesk/help/toolbox/instrument/>
- [27] <http://www.bci2000.org/>
- [28] <http://www.bci2000.org/wiki/>
- [29] http://www.bci2000.org/wiki/index.php/Programming_Reference:Contents
- [30] <http://www.libsdl.org/>
- [31] <http://www.shifz.org/brainbay/>
- [32] <http://openeeg.sourceforge.net/doc/sw/NeuroServer/>
- [33] <http://openeeg.sourceforge.net/>
- [34] <http://openvibe.inria.fr/>
- [35] <http://brain.fuw.edu.pl/edu/EEG:Skrypt>
- [36] <http://www.gnu.org/licenses/gpl.html>
- [37] <http://www.gnu.org/licenses/lgpl.html>
- [38] <http://www.gnu.org/>
- [39] <http://www.gnuarm.com/>
- [40] <http://www.freertos.org/>
- [41] <http://www.libusb.org/>
- [42] <http://openocd.sourceforge.net/>
- [43] <http://urjtag.sourceforge.net/>
- [44] <http://en.wikipedia.org/wiki/JTAG>
- [45] <http://www.arm.com/products/system-ip/debug-trace/coresight-soc-components/serial-wire-debug.php>