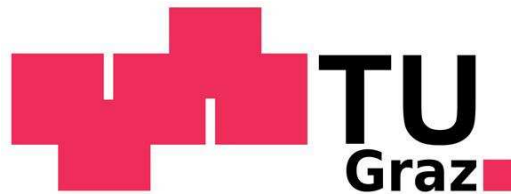


Peter HAGAUER

Prototypentwicklung eines EEG-Headsets
mit On-Board DSP und Implementierung eines Brain-Switchs
mittels Beta-Rebound-Detektion

Diplomarbeit



Institut für Semantische Datenanalyse / Knowledge Discovery
Laboratory of Brain-Computer Interfaces
Technische Universität Graz
Krenngasse 37, A - 8010 Graz
Vorstand: Assoc. Prof. Dipl.-Ing. Dr.techn. Gernot Müller-Putz

Betreuer: Assoc. Prof. Dipl.-Ing. Dr.techn. Gernot Müller-Putz

Begutachter: Assoc. Prof. Dipl.-Ing. Dr.techn. Gernot Müller-Putz

Graz, Oktober 2012

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Ort, Datum

Unterschrift

Kurzfassung

Diese Arbeit dokumentiert die Prototypentwicklung eines EEG-Headsets zur portablen Aufzeichnung des Elektroenzephalogramms (EEG) und anschließender digitaler Signalverarbeitung, um portable Brain-Computer Interfaces realisieren zu können. Das System besteht aus einem miniaturisierten 1-Kanal EEG-Verstärker, einem mit diversen Schnittstellen ausgestatteten Motherboard zur Aufnahme des steckbaren Prozessormoduls CM-BF527 sowie einem Trägersystem zur Kopfmontage sämtlicher benötigter Komponenten. Der digitale Signalprozessor von Analog Devices wird mit dem Betriebssystem uClinux, einer für eingebettete Systeme optimierten Linux-Distribution, betrieben und mit dem Xenomai-Microkernel erweitert, um dem System Echtzeitfähigkeit zu verleihen. Die Softwareentwicklung von Anwenderprogrammen wird mit Open-Source-Programmen durchgeführt und am Beispiel eines Brain-Switches, einem mittels Gedanken betätigten Taster, gezeigt. Um dies bewerkstelligen zu können, wird die als Beta-Rebound bezeichnete spektrale Leistungszunahme durch eine Fußbewegungsvorstellung in einem anwenderspezifischen Frequenzband detektiert.

Schlagwörter: Elektroenzephalogramm (EEG), Brain-Computer Interface (BCI), Digitaler Signalprozessor (DSP), Beta-Rebound, Brain-Switch, uClinux.

Abstract

This work describes the development of a prototype for portable recording of the electroencephalogram (EEG) and digital signal processing to realise Brain-Computer Interfaces, called EEG-Headset. The system consists of a miniaturized one-channel EEG-Amplifier, a well equipped motherboard to connect the Core-Module CM-BF527 and a system for head mounting of all the necessary components. As operating system for the digital signal processor from Analog Devices uClinux is used, which is a Linux distribution for embedded systems. To get realtime capabilities an additional microkernel from Xenomai is attached. Software development for user applications is done with open source programs. As demonstration application, a brain switch, which is a switch controlled by thoughts, is realised. This is done by detection of the power increase in a user-specific frequency band caused by imagination of foot movement, called beta rebound.

Keywords: Electroencephalography (EEG), Brain-Computer Interface (BCI), digital signal processor (DSP), beta rebound, brain switch, uClinux

Inhaltsverzeichnis

Eidesstattliche Erklärung.....	1
Kurzfassung.....	2
Abstract.....	2
Inhaltsverzeichnis.....	3
1 Einleitung.....	6
1.1 Entstehung des EEGs.....	6
1.2 Eigenschaften des EEGs.....	8
1.2.1 Amplitude.....	8
1.2.2 Oszillationen.....	8
1.2.3 Ereignisbezogene (De-) Synchronisation.....	9
1.2.4 Ereigniskorrelierte Potentiale.....	10
1.3 Messung des EEGs.....	14
1.3.1 Elektroden.....	14
1.3.2 Elektrodenposition.....	15
1.3.3 Ableitmethoden.....	15
1.3.4 Vorverstärker.....	17
1.3.5 Verstärker.....	17
1.3.6 Filter.....	17
1.3.7 Analog-Digital-Konvertierung.....	18
1.3.8 Speicherung.....	18
1.3.9 Störungen.....	18
1.4 Anwendungsgebiete.....	18
1.5 Brain-Computer Interface.....	19
1.5.1 Signalgewinnung.....	19
1.5.2 Vorverarbeitung.....	20
1.5.3 Parameterextraktion.....	22
1.5.4 Klassifikation.....	23
1.5.5 Applikations-Interface.....	23
1.5.6 Anwendungen.....	24
1.5.7 Feedback.....	25
1.6 Marktübersicht.....	26
1.6.1 Portable EEG-Systeme.....	26
1.6.2 EEG-Headsets.....	28
1.7 Motivation.....	31
1.8 Zielsetzung.....	31

2 Hardware	32
2.1 Allgemeines.....	32
2.2 EEG-Verstärker.....	33
2.2.1 Vorverstärker.....	33
2.2.2 Erste Verstärkerstufe.....	35
2.2.3 Zweite Verstärkerstufe und Filter.....	37
2.2.4 Versorgungsinterface.....	38
2.2.5 Board-Layout.....	39
2.3 Digitaler Signalprozessor.....	40
2.4 Motherboard.....	42
2.4.1 Spannungsversorgung.....	42
2.4.2 Prozessorschnittstelle.....	43
2.4.3 Joint Test Action Group.....	44
2.4.4 Universal Asynchronous Receiver Transmitter.....	45
2.4.5 Analog-Digital-Konverter.....	45
2.4.6 Driven-Right-Leg-Driver.....	48
2.4.7 EEG-Interface.....	49
2.4.8 Universal Serial Bus.....	49
2.4.9 Parallel Peripheral Interface.....	50
2.4.10 SD-Card.....	50
2.4.11 Ethernet.....	51
2.4.12 Schaltungslayout.....	52
2.5 USB-Adapter.....	53
2.6 Energiespeicher.....	53
2.7 Trägersystem.....	53
2.8 Gesamtsystem.....	54
3 Software	55
3.1 Einführung.....	55
3.2 Entwicklungsrechner.....	56
3.3 Blackfin GNU Toolchain.....	57
3.3.1 Blackfin GCC.....	58
3.3.2 Binutils.....	58
3.3.3 Standard-C-Bibliothek uClibc.....	58
3.3.4 Kernel Header.....	59
3.3.5 FLAT Datenformat.....	59
3.4 uClinux.....	60
3.4.1 Stammverzeichnis blackfin-linux-dist.....	60
3.4.2 Der Linux-Kernel.....	61
3.4.3 Konfiguration.....	64
3.4.4 Kompilation.....	66

3.5	Harte Echtzeit: Xenomai.....	67
3.5.1	Prinzip.....	67
3.5.2	Adeos/ I-Pipe.....	67
3.5.3	Struktur des Xenomai-Linux-Systems.....	69
3.5.4	Systemerweiterung und Konfiguration.....	70
3.5.5	Bereitgestellte Programmierschnittstellen (API).....	70
3.6	Minicom.....	71
3.7	U-Boot-Bootloader.....	72
3.7.1	Aufgaben.....	72
3.7.2	Funktionen.....	72
3.7.3	Umgebungsvariablen.....	73
3.7.4	Laden und booten von uClinux.....	74
3.8	Eclipse.....	75
3.9	Hello World.....	75
4	Portables Brain-Computer Interface.....	76
4.1	Vorgehensweise.....	76
4.2	Signalgewinnung.....	77
4.3	Spektrale Bandleistung.....	79
4.4	Brain-Switch.....	81
5	Ergebnisse.....	83
6	Diskussion.....	91
Anhang A: Bauteile und Bezugsquellen.....	103	
A1:	Bezugsquellen.....	103
A2:	Verwendete Bauteile EEG-Verstärker.....	104
A3:	Verwendete Bauteile Motherboard.....	105
A4:	Kostenaufstellung.....	107
Anhang B: Schaltpläne.....	108	
B1:	EEG-Verstärker.....	108
B2:	Motherboard Teil1.....	109
B3:	Motheboard Teil 2.....	110
Anhang C: uClinux Konfiguration.....	111	
C1:	Kernel-Konfiguration.....	111
C2:	uClinux-Konfiguration.....	120
Anhang D: Programmcodes.....	130	
D1:	C-Quellcode von read_ADC.....	130
D2:	C-Quellcode von BCI_training.....	132
D3:	C-Quellcode von BCI.....	135

1 Einleitung

Nur wenige Jahrzehnte nach der Erfindung des Galvanometers im frühen 19. Jahrhundert experimentierte Richard Caton (1842-1926) an geöffneten Schädel von Katzen und Hasen und konnte dabei äußerst geringe Ströme wechselnder Richtung nachweisen, wenn die beiden Elektroden an unterschiedliche Stellen am Gehirn angebracht waren. Seine Ergebnisse wurden 1875 veröffentlicht und stellen die Geburtsstunde der Untersuchung von Gehirnströmen dar.[11]

Im Jahre 1929 gelang es Hans Berger (1873-1941) basierend auf der Arbeit von Caton und einigen anderen Forschern (Adolf Beck, Napoleon Cybulski u.a.) das erste vom Menschen abgeleitete Elektroenzephalogramm, kurz EEG, aufzuzeichnen. Der von Berger geprägte Begriff Elektroenzephalogramm leitet sich aus dem griechischen „*enkephalon*“ für Gehirn und „*gráphein*“ für schreiben ab.[8]

Seitdem hat sich das EEG zu einem der wichtigsten Verfahren in der klinischen Diagnose und für Forschungszwecke weiterentwickelt.

1.1 Entstehung des EEGs

Das menschliche Gehirn wiegt im Durchschnitt 1,5 kg [121] und setzt sich aus den zwei Zelltypen Gliazellen und Neuronen zusammen. Auch wenn sie beide jeweils ca. 50 % zum Gesamtgewicht beitragen, so sind ca. 90% aller Zellen im Gehirn Gliazellen. Sie bilden ein Stützgerüst für die Nervenzellen (Neuronen), isolieren diese gegeneinander und sind auch am Stoff- und Flüssigkeitstransport im Gehirn beteiligt. Die elektrische Aktivität, die mittels EEG aufgezeichnet werden kann, entsteht jedoch durch die restlichen 10% der Zellen, der etwa 100 Milliarden Neuronen.[6]

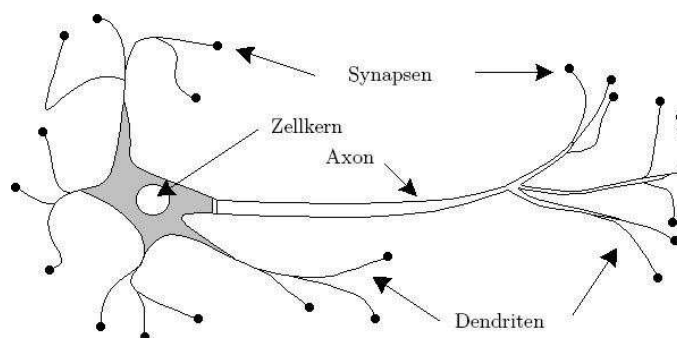


Abbildung 1: Aufbau eines Neurons

Wie in Abbildung 1 dargestellt, besteht ein Neuron im wesentlichen aus dem Zellkörper, den Dendriten und einem Axon. Die Verbindung zwischen mehreren Zellen wird mit Synapsen gewährleistet, wobei ein Axon eines Neurons mit einer Synapse an die Dendriten bzw. direkt am Zellkörper des nächsten Neurons andockt.[6]

Ein Neuron kann bis zu 10.000 Synapsen zu anderen Zellen bilden, wodurch sich ein (noch) nicht zu durchschauendes Netzwerk bildet. Findet eine Kommunikation zwischen zwei Zellen statt, so werden Neurotransmitter (chemische Botenstoffe) in den synaptischen Spalt ausgeschüttet. Durch diese Transmitter wird das Ruhepotential der Zelle, das ca. -70 bis -80 mV beträgt, beeinflusst. Diese als postsynaptische Potentiale (PSP) bezeichneten Potentialänderungen haben eine Amplitude von ca. 5 mV, sind von ca. 10 bis 50 ms Dauer und können in exzitatorische postsynaptische Potentiale („reizende“, spannungssteigernde Wirkung) bzw. in inhibitorische postsynaptische Potentiale („hemmende“, spannungssenkende Wirkung) differenziert werden (Abbildung 2, links). An der Zellmembran des Neurons kommt es zu einer Summation der eingehenden PSP (Superposition). Wird der Schwellwert des Membranpotentials durch mehrere synchron eingehende PSP überschritten, so wird ein Aktionspotential ausgelöst (Abbildung 2, rechts). Dieses Aktionspotential breitet sich entlang des Axons aus und löst eine Freisetzung von Neurotransmittern in den synaptischen Spalt der nächsten Verbindung aus.[11]

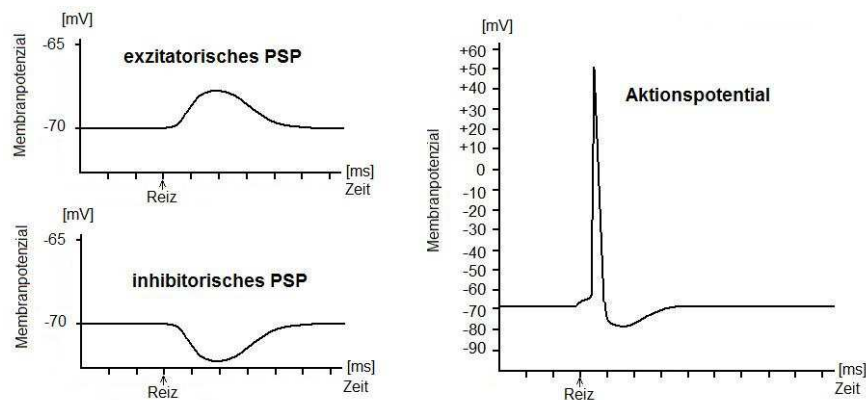


Abbildung 2: postsynaptische Potentiale und Aktionspotential

Bei einer Überschreitung des Schwellwertes kommt es zu einer Depolarisation des Membranpotentials, das sogar positive Werte annimmt. Nach ca. 1 bis 2 ms kommt es zu einer Repolarisation des Membranpotentials, wobei das Ruhepotential deutlich unterschritten (Hyperpolarisation) wird und sich anschließend wieder das Ruhepotential von -70 mV einstellt. Die Zeit zur Wiederherstellung des Ruhepotentials wird als Refraktärzeit bezeichnet und es kann während dieser Dauer kein neues Aktionspotential ausgelöst werden.

Interessant ist, dass das EEG nicht die Folge von Aktionspotentialen darstellt, sondern die Summe der elektrischen Aktivität der unterschwelliger Reize, der PSP, beinhaltet. Aufgrund der großen Neuronendichte in der äußersten Schicht des Kortex, ca. 70.000 pro mm^3 , der anatomisch parallel angeordneten Axonen der Neuronen im Kortex (Pyramidenzellen) und der daraus folgenden apikalen Lage der Dendriten sowie der hohen Synchronizität der PSP (aufgrund der relativ langen Dauer von PSP) kommt es an der Kortexoberfläche zu Potentialverschiebungen, die auch noch nichtinvasiv an der Schädeloberfläche gemessen werden können.

1.2 Eigenschaften des EEGs

Das von der Kopfhaut abgeleitete elektrische Signal kann grundsätzlich differenziert werden in spontanes, ereignisunabhängiges EEG und ereignisbezogene EEG-Phänomene. Das spontane EEG ist die natürliche, immer vorhandene, elektrische Aktivität des Gehirns, ereignisbezogene EEG-Phänomene treten jedoch nur im Zusammenhang mit externen Stimuli oder interner Intention auf und überlagern sich dem spontanen EEG. Da diese Phänomene eine wesentlich kleinere Amplitude als das spontane EEG aufweisen, werden diese maskiert und können erst durch arithmetische Mittelungsverfahren nachgewiesen werden. Um dies bewerkstelligen zu können, wird dem Probanden ein wiederkehrender Stimulus innerhalb eines definierten Zeitfensters (Trials, Epochen) dargeboten und die erhaltenen Signale jeder Epoche arithmetisch gemittelt. Dabei wird das spontane EEG als Hintergrundrauschen mit einem Mittelwert gleich Null betrachtet, wodurch es sich nach einer ausreichenden Anzahl von Epochen auslöscht und die ereignisbezogene Reizantwort sichtbar wird.[11]

1.2.1 Amplitude

Das spontane EEG einer wachen, durchschnittlichen Person hat eine Amplitude von ca. 10 bis 500 μV an der Schädeloberfläche, wobei hohe Amplituden durch eine große Anzahl von synchronen Neuronen unter der Elektrode erklärt werden können, was mit verminderter Aktivierung assoziiert wird. Nimmt die Amplitude des Signals jedoch ab, so sind die Neuronen desynchronisiert, was auf eine Aktivierung der Neuronen schließen lässt, also den aktiven Zustand der Zellen darstellt. Weiters beeinflusst auch der Elektrodenabstand die Amplitude des EEGs, wobei größere Amplituden durch größere Elektrodenabstände erreicht werden können.[93]

1.2.2 Oszillationen

Das spontane EEG beinhaltet Spektralkomponenten von 1 bis 70 Hz. Diese werden in verschiedene charakteristische Frequenzbänder eingeteilt und jeweils verschiedenen mentalen Zuständen zugeordnet und sind Tabelle 1 zu entnehmen.[11]

Frequenzband	Frequenz	Mentaler Zustand
Delta	0,5 – 4 Hz	Tiefschlaf
Theta	4 – 8 Hz	Tiefe Entspannung, Wachtraum
Alpha	8 – 13 Hz	Entspannter Wachheitszustand
Beta	13 – 30 Hz	mentale Aktivität
Gamma	30 – 70 Hz	Intensive Informationsverarbeitung

Tabelle 1: EEG-Frequenzbänder [11]

1.2.3 Ereignisbezogene (De-) Synchronisation

Neben den Oszillationen des spontanen EEGs treten auch ereignisbezogene Oszillationsveränderungen auf, die zeitgebunden durch interne Intention, z.B. eine gedankliche Bewegungsvorstellung, auftreten (siehe Abbildung 3). Nimmt die spektrale Leistung in einem bestimmten Frequenzbereich ab, so spricht man von ereignisbezogener Desynchronisation (ERD), nimmt die Leistung zu, von ereignisbezogener Synchronisation (ERS). [80]

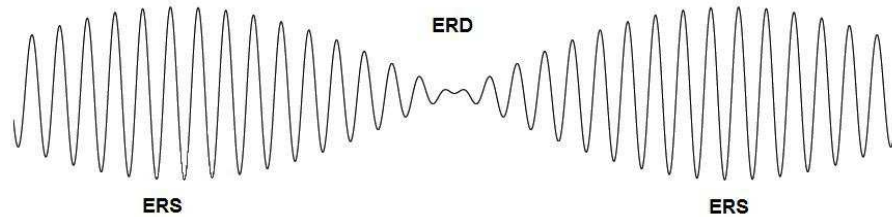


Abbildung 3: ERD/ERS im 12 Hz-Bereich

Da diese spektralen Leistungsänderungen in verschiedenen Frequenzbereichen und auch zeitlich verschoben zu der Intention auftreten, werden zur Visualisierung ERD/ERS-Maps angefertigt. Abbildung 4 zeigt eine solche ERD/ERS-Map: Dabei wird die, über mehrere Epochen gemittelte, spektrale Leistung eines Testintervalls nach der Intention (durchgeführt zur Sekunde 2, dargestellt als vertikaler schwarzer Balken) relativ zu einem Referenzintervall (dargestellt als vertikaler grauer Balken) vor der Intention in einer Zeit-Frequenz-Ebene farblich dargestellt. [53]

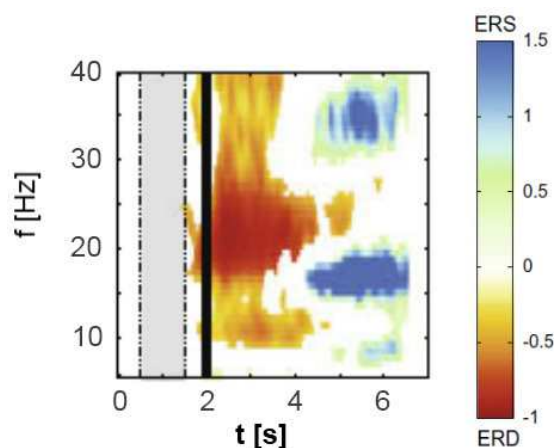


Abbildung 4: ERD/ERS-Map, modifiziert nach [84]

1.2.4 Ereigniskorrelierte Potentiale

Durch Sinnesreize von außen kommt es zu ereignisbezogenen kortikalen Potentialschwankungen, die zeit- und phasengebunden zu einem Reiz auftreten und sich dem spontanen EEG überlagern. Im Folgenden werden kurz einige der bedeutendsten ereigniskorrelierten EEG-Potentialveränderungen vorgestellt, die im Laufe der Jahre der EEG-Forschung identifiziert und untersucht wurden.

1.2.4.1 Slow Cortical Potentials

Gleichspannungsdrifts mit einer Frequenz von unter 1 Hz werden als Slow Cortical Potentials (langsame kortikale Potentiale, SCP) bezeichnet und wurden bereits durch Canton im Jahr 1875 entdeckt, blieben aber bis 1964 durch Walter [112] ohne größerer Beachtung, da die meisten kommerziellen EEG-Verstärker diese niedrigen Frequenzen herausgefiltert hatten. Diese langsamen Potentialveränderungen, dargestellt in Abbildung 5, haben eine Amplitude von einigen μV bis zu über 100 μV und eine Zeitdauer von ca. 300 ms bis zu einigen Sekunden und können das Potential in die positive Richtung (Positivierung) als auch in die negative Richtung (Negativierung) verschieben.[117]

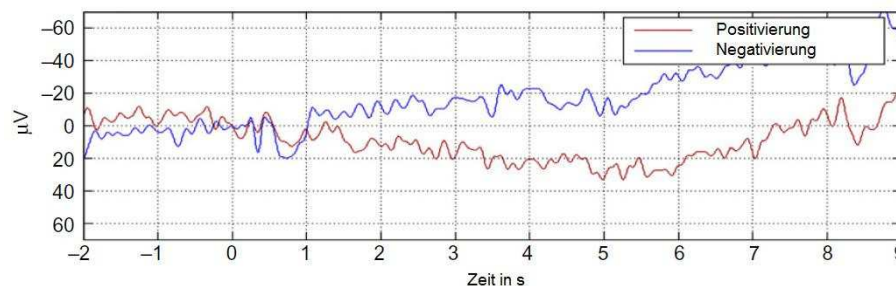


Abbildung 5: Positivierung und Negativierung von SCP, modifiziert nach [117]

Für die Entstehung dieser Potentiale werden größere Neuronenverbände verantwortlich gemacht, die für die Planung und Mobilisierung von zielgerichteten Handlungen zuständig sind. Eine Negativierung zeigt exzitatorische Mobilisierung an und wird mit der Bereitstellung von Ressourcen zur Aufmerksamkeitssteuerung in Zusammenhang gebracht. So konnte gezeigt werden, dass eine Negativierung positive Effekte auf Reaktionszeiten, Wahrnehmung und Kurzzeitgedächtnis hat. Im Gegensatz dazu wird eine Positivierung mit Hemmprozessen, einem Verbrauch von Ressourcen sowie einer reduzierten Erregbarkeit des jeweiligen Kortexareals assoziiert.[117]

SCP können mittels Feedback bewusst beeinflusst werden und kommen in der Therapie von ADHS (Aufmerksamkeitsdefizit-/Hyperaktivitätssyndrom) und anderen Konzentrationsstörungen zum Einsatz [62]. Die physiologische Bedeutung dieser bewusst herbeigeführten Potentialänderungen im EEG ist noch nicht vollständig geklärt.

1.2.4.2 Bereitschaftspotential

Ein sehr bekannter Vertreter der langsamen kortikalen Potentiale ist das Bereitschaftspotential (BP), das 1965 entdeckt wurde und eine etwa 1 Sekunde vor einer willentlichen Bewegung beginnende Potentialverschiebung ist. Es handelt sich dabei um eine langsam ansteigende, negative Spannungsverschiebung von ca. 20 μV , die über dem supplementär-motorischen Kortex messbar ist.[95] Das Bereitschaftspotential ist in Abbildung 6 dargestellt.

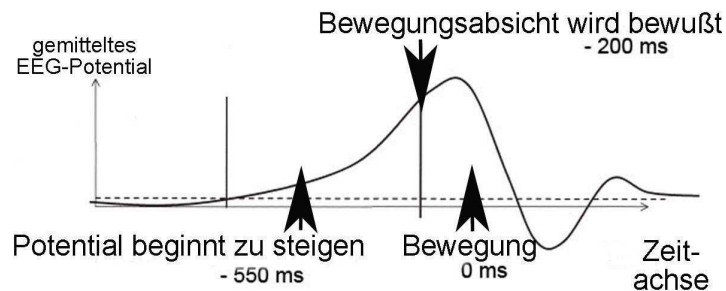


Abbildung 6: Bereitschaftspotential

20 Jahre nach dem ersten Nachweis des BPs, führte der amerikanische Physiologe Benjamin Libet weitere Experimente durch [64]. Die Versuchspersonen sollten sich den Zeitpunkt, wann ihnen eine willentliche Bewegungsabsicht bewusst wurde, anhand einer kreisenden Uhr merken. Da das BP bei allen Versuchspersonen bereits Sekundenbruchteile vor der Bewusstwerdung des Bewegungswunsches nachweisbar war, lösten diese Versuche eine rege Diskussion über den freien Willen aus [60].

1.2.4.3 Evozierte Potentiale

Evozierte Potentiale (EP) sind die elektrische kortikale Antwort auf externe Stimuli und werden je nach aufgetretenen Sinnesreiz unterteilt in:

- akustisch evozierte Potentiale (Ton),
- visuell evozierte Potentiale (Licht),
- somatosensorisch evozierte Potentiale (Berührung, Vibration),
- olfaktorisch evozierte Potentiale (Gerüche).[88]

Die auftretenden Signalmuster (Abbildung 7) sind abhängig von der Art des Stimulus, dem Ableitort und dem aktuellen mentalen Zustand des Probanden. Nach dem Stimulus kommt es zu einer charakteristischen Abfolge von positiven und negativen Potentialänderungen, die je nach Ausrichtung und deren Latenz zum Stimulus bezeichnet werden.

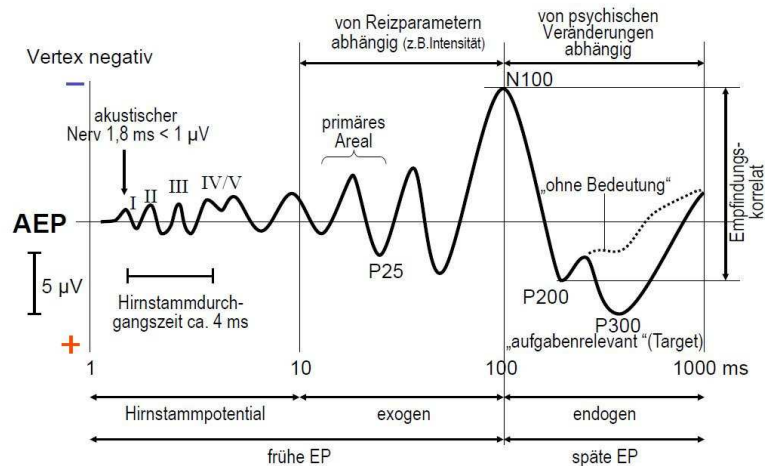


Abbildung 7: Typisches akustisch evoziertes Potenzial [13]

Komponenten mit einer Latenz bis ca. 10 ms sind Hirnstammpotentiale bzw. werden als Antwort von Rezeptoren und des peripheren Nervensystems, spätere Komponenten als vom Gehirn selbst erzeugte Aktivität betrachtet. Eine weit verbreitete weitere Unterscheidung ist die Differenzierung in exogene bzw. endogene Komponenten nach Donchin [42], wobei exogene Komponenten nur eine Abhängigkeit von Reizparametern aufweisen und unabhängig von der psychischen Verfassung (Vigilanz, Aufmerksamkeit, Intelligenz,...) des Probanden sind. Endogene Komponenten haben eine Latenz von über ca. 100 ms und sind von internen kognitiven Prozessen abhängig. So werden die Komponenten von 100 bis 200 ms vor allem durch die Aufmerksamkeit des Probanden beeinflusst, spätere Komponenten wie die P300 zeigen die Erkennung und Unterscheidung von präsentierten Stimuli. Die Latenz von P300 wird beeinflusst durch die Intelligenz des Probanden und nimmt bei bereits bekannten Reizen ab.

1.2.4.4 Steady-State Evozierte Potentiale

Durch einen externen Stimulus wird ein evoziertes Potenzial verursacht. Bereits 1966 wurde durch Regan [90] das Konzept der steady-state evozierten Potenziale (SSEP) beschrieben, die durch repetitive Stimulation ab einer Reizfrequenz von 6 Hz entstehen. Dabei überlagern sich die evozierten Potenziale und münden in einer EEG-Aktivität, die in Dauer und Frequenz den Stimuli entsprechen (Abbildung 8) und mittels Spektralanalyse nachgewiesen werden können. Das Spektrum der SSEP weist die dem Reiz korrespondierende Frequenz (1. Harmonische) auf, es treten jedoch sowohl Oberschwingungen als auch subharmonische Schwingungen auf. [54]

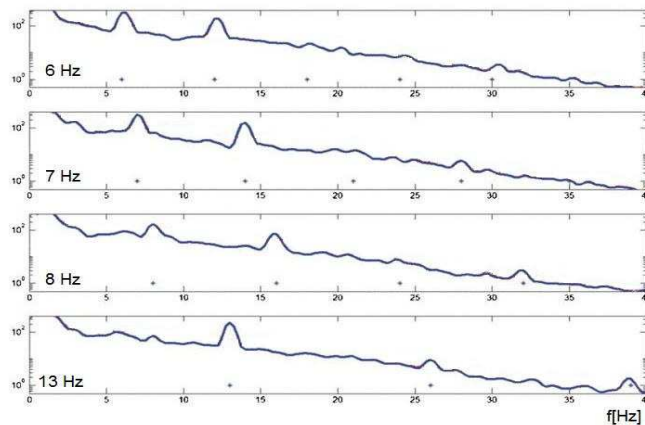


Abbildung 8: SSVEP bei unterschiedlichen Frequenzen, modifiziert nach [74]

Analog zu den EP werden auch SSEP je nach auslösenden Sinnesreiz eingeteilt in:

- Steady-state akustisch evozierte Potentiale (Tonabfolge),
- Steady-state visuell evozierte Potentiale (Lichtblitze) und
- Steady-state somatosensorisch evozierte Potentiale (mech. Reize, Vibration).

1.3 Messung des EEGs

Wie in Abbildung 9 dargestellt, wird mit Hilfe von Elektroden, die auf der Schädeloberfläche angebracht werden, die elektrischen Spannungsschwankungen erfasst und einem Vorverstärker zugeführt. Nach dieser ersten Verstärkung wird das Signal weiter verstärkt und anschließend gefiltert, um Störungen zu unterdrücken. Abschließend wird das nun analog vorliegende EEG auf Papier gedruckt oder digitalisiert, um das Signal mittels Software weiter zu manipulieren bzw. die Daten zu speichern oder zu visualisieren.

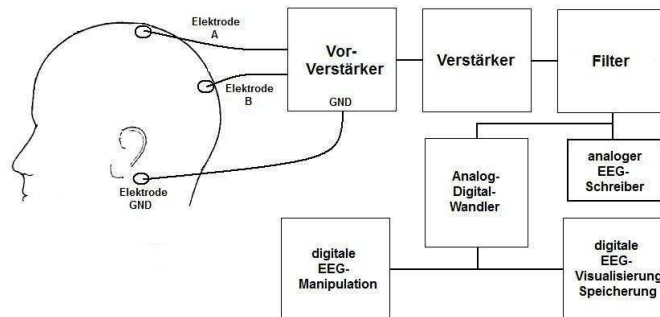


Abbildung 9: Prinzip der EEG-Messung

1.3.1 Elektroden

Zur Erfassung des EEGs werden Elektroden aus Gold, Silber, vergoldete Silber- oder Zinnelektroden, Silberchloridelektroden oder gesinterte Silber/Silberchloridelektroden verwendet [76]. Sie sind in unterschiedlichen Bauformen erhältlich, je nach Anwendung, Hersteller bzw. Befestigungssystem. Die Messstelle wird vor der Messung mit Alkohol gereinigt und alte Hautschichten entfernt (Präparation). Die Elektroden werden mittels EEG-Haube oder einem alternativen Befestigungssystem an der Schädeloberfläche platziert und mit einer abgestimmten Elektrodenpaste gefüllt. Mit Hilfe der Paste soll der Übergangswiderstand auf unter 5 k Ω gesenkt werden. In manchen Fällen werden die Elektroden auch einfach geklebt (z.B. mit Kolodium).[11][99]

Um die Signalqualität zu erhöhen, kann der Vorverstärker direkt an der Elektrode verbaut werden, um so die Induzierung von Störspannungen in den Ableitkabeln zu vermeiden. Diese Bauform wird als aktive Elektrode bezeichnet. [49]

Mittlerweile sind auch Elektroden erhältlich, die ohne Elektrodenpaste und vorheriger Präparation der Kopfhaut auskommen. Sie werden als Trockenelektroden bezeichnet und könnten die Bedienungsfreundlichkeit von EEG-Systemen wesentlich erhöhen. Auch wenn die Signalqualität noch nicht klinischen Standards entspricht, so lassen Vergleichsstudien hoffnungsvoll in die Zukunft blicken [16].

1.3.2 Elektrodenposition

Um EEG-Daten von verschiedenen Personen miteinander vergleichen zu können, ist es nötig, fixe Messpunkte an der Schädeloberfläche zu definieren. Da die Schädelgröße und -form bei jedem Menschen unterschiedlich ist, veröffentlichte Herbert Jasper im Jahre 1958 das sogenannte 10/20-System [56], das bis heute den internationalen Standard darstellt (Abbildung 10).

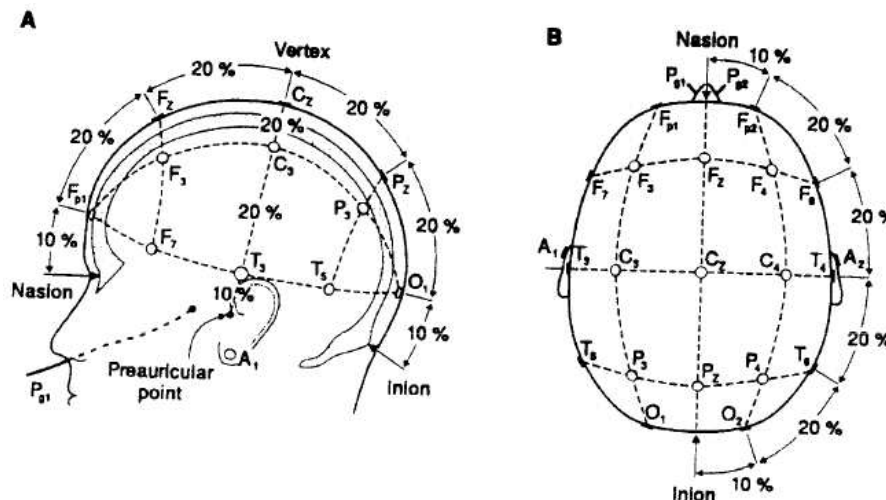


Abbildung 10: Elektrodenpositionen nach dem 10/20-System [56]

Dabei wird die Distanz zwischen den Referenzpunkten Nasion (Nasenwurzel) und Inion (Sporn am Hinterkopf), sowie die Distanz zwischen den präaurikulären Punkten über den Vertex gehend in 10% bzw. 20% Abschnitte (daher auch 10/20-System) aufgeteilt.

Es sind insgesamt 21 Elektrodenpositionen definiert. Werden mehr Ableitpunkte benötigt, kann das 10/10-System (bis zu 81 Positionen) bzw. das 10/5-System (bis zu 320 Positionen) verwendet werden [57].

1.3.3 Ableitmethoden

Um einen EEG-Kanal ableiten zu können, werden 3 Elektroden benötigt, zum einen die Masselektrode und zwei weitere zur Erfassung des EEGs. Je nachdem, wie die Elektroden miteinander verschalten bzw. platziert werden, spricht man von monopolarer bzw. bipolarer Ableitung. [48]

1.3.3.1 Masselektrode

Die erste Elektrode ist die Masselektrode (Ground). Da jeder Verstärker nur einen begrenzten Eingangsspannungsbereich besitzt, würde es ohne einheitliche Masse zwischen der Verstärkerschaltung und Proband zu zu hohen Gleichtaktspannungen und zu einer möglichen Übersteuerung des Verstärkers kommen. Diese Elektrode kann theoretisch an jedem Punkt des menschlichen Körpers angebracht werden, um

jedoch die Induktionsschleife für Störspannungen zu verkleinern, wird sie meist am Kopf montiert. Übliche Elektrodenpositionen für die Masse sind die Stirn, eines der Ohrläppchen oder ein präauriculärer Punkt. [41] [79]

1.3.3.2 Monopolare Ableitung

Bei der monopolaren Ableitung befindet sich nur eine Elektrode über einem Areal mit relevanter Aktivität, die zweite bildet eine Referenz als Bezugspotential, die in unterschiedlichen Methoden gewonnen werden kann: [93]

- Eine Referenzelektrode an einem elektrisch neutralen Ort, üblicherweise wird hierfür die Position A1 verwendet. Diese Variante kann allerdings zu Unsymmetrien führen.
- Verwendung von 2 Referenzelektroden an Position A1 und A2, die mit Hilfe eines Spannungsteilers mit 2 mal 5 kOhm zwischen den beiden Referenzen den Mittelwert bilden. Der Mittelpunkt des Spannungsteilers wird als symmetrischer Referenzpunkt verwendet.

Der Vorteil der monopolaren Ableitung ist die genaue Abbildung hinsichtlich Topographie, Polarität und Amplitude, sie ist jedoch stör anfällig, vor allem bei Störungen die an der Referenzelektrode auftreten. Für mehrkanalige Ableitungen werden alle zusätzlichen Elektroden auf die Referenz bezogen, wodurch für jeden weiteren Kanal nur eine zusätzliche Elektrode benötigt wird. Ein weiterer Vorteil ist, dass sich monopolare Ableitungen rechnerisch in bipolare Ableitungen überführen lassen.

Eine weitere Möglichkeit der Referenzbildung bei mehrkanaligen Ableitungen sind rechnerische Verfahren: [93][120]

- Bei der Common Average Reference (CAR) wird der Mittelwert über alle Elektroden als Referenz ermittelt, wodurch die Referenz weniger stör anfällig wird, sich jedoch lokale Aktivitätsschwankungen und Artefakte auf alle anderen Ableitungen auswirken.
- Für die Local Average Reference oder Laplace-Ableitung wird der Mittelwert der 4 orthogonalen (small Laplacian) bzw. 8 (large Laplacian) benachbarten Elektroden ermittelt und als Referenz verwendet. Dadurch erhöht sich die räumliche Auflösung durch die Eliminierung von entfernten Potentialen, die Amplitude des Signales nimmt jedoch ab.

1.3.3.3 Bipolare Ableitung

Werden beide Messelektroden über Arealen mit relevanter elektrischer Aktivität angebracht, so spricht man von der bipolaren Ableitung. Die Elektroden werden jeweils an zwei benachbarten Punkten montiert und erfassen die Potentialdifferenz zwischen diesen beiden Elektroden. Durch die bipolare Ableitung erreicht man eine hohe räumliche Auflösung, nimmt dafür jedoch kleinere Amplituden in Kauf [109]. Werden mehrere Kanäle abgeleitet, so werden für jeden zusätzlichen Kanal je zwei weitere Elektroden benötigt.

1.3.4 Vorverstärker

Da man bei einer EEG-Messung sehr kleine Potentialdifferenzen zwischen zwei Punkten auf der Kopfoberfläche bestimmen möchte, bieten sich Differenzverstärker an. Zum Einsatz kommen meist Instrumentenverstärker, da sie einen hohen Eingangswiderstand haben, wodurch es zu keiner Belastung der Potentiale kommt. Durch die hohe Gleichtaktunterdrückung von modernen Instrumentenverstärkern wird auch die Störspannung der kapazitiven Einkopplung stark reduziert, da die induzierten Störspannungen auf allen Ableitungen gleich groß sind. [75]

Die Messgenauigkeit eines EEG-Systems wird im wesentlichen durch diese erste Verstärkerstufe bestimmt, da jeglicher Fehler von den folgenden Verstärkern weiter verstärkt werden würde. Daher kommen Präzisionsinstrumentenverstärker mit sehr hohem Eingangswiderstand, möglichst hoher Gleichtaktunterdrückung sowie geringem Verstärkerrauschen zum Einsatz.

1.3.5 Verstärker

Um das EEG weiter verarbeiten zu können, reicht eine einmalige Verstärkung nicht aus. Das Signal wird bis in den Volt-Bereich verstärkt, je nachdem, wie das Signal ausgewertet werden soll. Zum Einsatz kommen Operationsverstärker, die mit wenigen zusätzlichen Bauteilen konventionelle Verstärkerschaltungen (invertierender Verstärker, nicht invertierender Verstärker, o.Ä.) bilden. [106]

In modernen EEG-Systemen wird das Signal nach erfolgter Filterung mittels eines Analog-Digital-Konverters in eine digitale Zahlenfolge zur Speicherung bzw. zur weiteren Signalmanipulation überführt.

1.3.6 Filter

Bei der EEG-Messung wird das erhaltene Spannungssignal mit Hilfe eines Bandpasses gefiltert. Die maximale obere Grenzfrequenz des Filters ergibt sich durch die halbe Abtastrate des Signals, um das Nyquist-Theorem nicht zu verletzen [78]. Sind nur geringe Frequenzen bis in den Beta-Bereich (<30 Hz) von Interesse, so bietet sich eine Tiefpassfilterung mit einer Grenzfrequenz von knapp unter 50 Hz an, da somit auch elektromagnetische Einflüsse des 50 Hz Netzes eliminiert werden können. Sollen auch Frequenzanteile im Gamma-Bereich (>30 Hz) analysiert werden, so ist ein zusätzlicher Notch-Filter zur Entfernung der Störanteile der Netzfrequenz vorzusehen. [106]

Um einen Drift der Nulllinie und langsame Potentialschwankungen zu unterdrücken, wird das Signal mit einem Hochpass gefiltert. Seine Grenzfrequenz soll nahe bei null liegen.

1.3.7 Analog-Digital-Konvertierung

Durch einen Analog-Digital-Konverter (ADC) kann ein analog vorliegendes Signal mit einer vorgegebenen Abtastfrequenz in eine digitale Zahlenfolge überführt werden. Es sind sehr viele verschiedenen Typen von Umsetzern für eine Vielzahl an Anwendungen erhältlich. Sie unterscheiden sich in Umsetzverfahren, Betriebsspannung, Umsetzdauer, digitales Datenformat, Auflösung, Genauigkeit, Nichtlinearitäten, Signal-Rausch-Verhältnis und mehr. [106]

1.3.8 Speicherung

Sobald das EEG digital vorliegt, ist es auch möglich es zu speichern. Der Speicherbedarf einer EEG-Aufzeichnung ist abhängig von der Auflösung des ADC, der Abtastfrequenz, dem Datenformat, eventuell parallel aufgezeichneten Parametern, usw. Bei einer seriellen Abspeicherung der reinen EEG-Messwerte eines Kanals mit einer Auflösung von 16 Bit und einer Abtastrate von 256 Hz würde man 1 MByte in ca. 32 Minuten vollschreiben.

1.3.9 Störungen

Bei der Messung des EEGs wird grundsätzlich zwischen technischen und biologischen Störungen unterschieden. Technischen Störungen, wie Elektrodenrauschen, Verstärkerrauschen, Quantisierungsrauschen, Störspannungen durch elektrische bzw. magnetische Felder, Aliasing oder Sättigungseffekte, sind auf die verwendete Messelektronik zurückzuführen. Biologischen Störungen entstehen durch, im messtechnischen Sinne, unerwünschte Aktivitäten im Organismus, wie z.B. Elektromyogramm (EMG), Elektrookulogramm (EOG), Elektrokardiogramm (EKG) oder Elektrodenartefakte durch Bewegung und Schweiß.[93]

1.4 Anwendungsgebiete

Das EEG ist eine Standarduntersuchung der Neurologie und dient zur Diagnostik von Schlafstörungen, Narkosetiefe, Koma, Epilepsie, Hirntod und mehr.

Durch rechnergestützte Analyse und Signalmanipulation des EEGs ergeben sich weitere Anwendungsmöglichkeiten. So ist es mit Hilfe von Mustererkennung möglich, eine durch Bewegungsvorstellung bewusst herbeigeführte Synchronisierung/Desynchronisierung (ERS/ERD) von Neuronen im motorischen Kortex in Echtzeit zu detektieren, um daraus Steuersignale zu generieren. Durch diese als Brain-Computer Interface (BCI) bezeichnete Technik eröffnen sich neue Anwendungsmethoden zur Kommunikation und Kontrolle.

1.5 Brain-Computer Interface

Mit Hilfe eines Brain-Computer Interfaces (BCI) wird ein direkter Kommunikationskanal für Menschen zur Verfügung gestellt, ohne dabei muskuläre Aktivitäten in Anspruch zu nehmen. Dabei werden EEG-Phänomene ausgenutzt, die durch Intention bzw. gezielte Stimulation ausgelöst und mittels Klassifikationsalgorithmen erkannt werden, um anschließend Steuerungsbefehle zu generieren [82][115]. So ist es auch hochgradig gelähmten Menschen möglich, durch Gedanken ihre Umwelt zu beeinflussen oder zu kommunizieren. Ein BCI-System umfasst sämtliche Hardware und Software, die benötigt wird, um diese Aufgabe bewerkstelligen zu können und wird schematisch in Abbildung 11 dargestellt:[81]

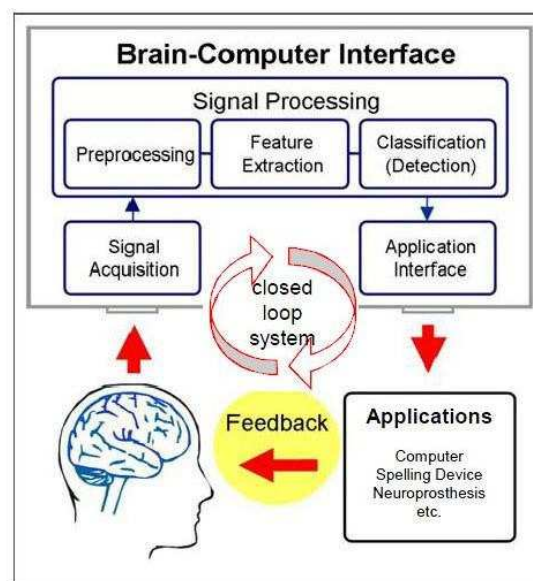


Abbildung 11: Schematische Darstellung eines BCIs [70]

1.5.1 Signalgewinnung

Sämtliche Verfahren, die in der Lage sind, die Aktivität des Gehirns zu erfassen, sind prinzipiell geeignet, um die Basis für ein BCI zu bilden [116]. Das erste und am meisten verwendete Verfahren zur Signalgewinnung für BCI-Systeme ist das EEG, da es im Vergleich mit anderen Verfahren einfach, kostengünstig und nicht invasiv ist und eine hohe Zeitauflösung im ms-Bereich besitzt. Der Nachteil ist die geringe Amplitude, da die Elektroden relativ weit entfernt von den signal-generierenden Neuronen sind und der Schädelknochen stark dämpfend wirkt [111]. Um diese Dämpfung zu umgehen und größere Signalamplituden zu erhalten, kommt auch in seltenen Fällen das Elektrocorticogramm (ECoG) zum Einsatz. Bei dieser invasiven Methode werden mehrere, zu einem Array zusammengeschaltete Elektroden direkt bei geöffneten Schädel subdural bzw. epidural auf das Gehirn appliziert [63] bzw. mit Nadelelektroden direkt in das Gewebe eingeführt [110] um die elektrische Aktivität einzelner Neuronen (single cell)

zu erfassen. Der Vorteil dieser Methode ist der verbesserte Signal-Rausch-Abstand und die höhere räumliche Auflösung. Der Nachteil ist die Tatsache, dass der Schädel geöffnet werden muss und dadurch ein hohes Infektionsrisiko besteht. Diese Methode ist daher für Forschungszwecke weniger geeignet.

Eine weitere nicht-invasive Signalgewinnungsvariante stellt die Aufzeichnung des Magnetoencephalogramms (MEG) dar. Bei diesem Verfahren werden die magnetischen Felder im fT-Bereich, die durch Stromflüsse im Gehirn entstehen mit Hilfe von supraleitenden Sensoren (SQIDS) gemessen. Vor allem durch die aufwendige Kühlung und den damit verbundenen Kosten kommt dieses Verfahren nur für experimentelle Versuche zum Einsatz. [15]

Neben diesen direkten Verfahren kommen zur Messung der Gehirnaktivität auch indirekte Messverfahren zum Einsatz, die metabolische Parameter bestimmen wie etwa die Nahinfrarotspektroskopie (NIRS) oder die funktionelle Magnetresonanztomographie (fMRI). Beide Verfahren erfassen den Sauerstoffgehalt des Blutes in bestimmten Hirnregionen. Da mit einer Aktivierung dieser Region eine Änderung des Sauerstoffgehaltes einhergeht, kann mit beiden Verfahren ein BCI realisiert werden.

Bei der NIRS wird die Tatsache, dass Licht im nahen Infrarot-Bereich Gewebe durchdringen kann (optisches Gewebefenster von 630 nm bis 1300 nm) ausgenutzt und man in der Lage ist, Konzentrationsänderung mit Hilfe des Beer-Lambert Gesetzes von Oxyhemoglobin (HbO_2) und Deoxyhemoglobin (Hb) zu bestimmen, da diese unterschiedliche optische Absorptionskoeffizienten besitzen. Auf diese Weise sind optische BCI-Systeme realisierbar, sie besitzen jedoch im Vergleich zu EEG-BCI-Systemen eine geringere zeitliche Auflösung im Sekunden-Bereich, da sich der Sauerstoffgehalt im Blut nicht sprunghaft ändern kann. [5][17]

Neben den unterschiedlichen optischen Absorptionskoeffizienten besitzt oxygeniertes und deoxygeniertes Blut auch unterschiedliche magnetische Eigenschaften, die mittels fMRI zur Signalgewinnung ausgenutzt werden können. So besitzt Oxyhemoglobin diamagnetische und Deoxyhemoglobin paramagnetische Eigenschaften, die unterschiedliche Kernspins zur Folge haben und dadurch zu unterschiedlichen Magnetresonanz-Signalen führt. Das gewonnen BOLD-Signal (blood oxygenation level dependent) ist daher zur Signalgenerierung für BCIs geeignet und ermöglicht eine hohe räumliche Auflösung, hat jedoch neben der schlechten zeitlichen Auflösung den Nachteil sehr aufwendig und teuer zu sein. [96]

1.5.2 Vorverarbeitung

Um die für die Klassifikation benötigten Parameter ermitteln zu können, muss das aufgenommene EEG bereinigt werden, da die zu ermittelnden ereignisbezogenen EEG-Phänomene im Vergleich mit dem spontanen EEG eine sehr kleine Amplitude besitzen, mit Artefakten kontaminiert sind und lokale Potentiale von mehreren Elektroden gleichzeitig erfasst werden. Die Hauptaufgabe der Vorverarbeitung liegt daher in der Verbesserung des Signal-Rausch-Verhältnisses (SNR), der Entfernung von Artefakten und der Reduzierung der Dimensionalität des Datensatzes bei Ableitung vieler Kanäle. Um dies zu erreichen stehen verschiedene mathematische Methoden zu Verfügung [14].

Eine der wichtigsten Methoden ist die digitale Filterung, die nicht benötigte spektrale Anteile im Signal eliminiert. Zur Verwendung kommen einfache FIR-Filter und IIR-Filter oder auch adaptive Filter. [78][59]

Bei der Ableitung von mehreren EEG-Kanälen werden lokale Potentiale vielfach von verschiedenen Elektroden erfasst. Um diesen Effekt zu minimieren kommen räumliche Filter zum Einsatz, um die tatsächliche Aktivität zu ermitteln. Dazu werden bipolare Filter, Laplace-Filter oder die CAR-Methode angewandt. [89]

Zur Rausch- und Artefaktentfernung aus dem gemessenen EEG sind auch statistische Verfahren geeignet, wie etwa die Principal Component Analysis (PCA) [91]. Die PCA ist eine einfache mathematische Transformation um korrelierte Variablen (Potential-Mehrfacherfassung) durch unkorrelierte Variablen zu ersetzen und diese nach absteigender Varianz zu sortieren (Principal Components).

Neben der PCA kommen auch andere statistische bzw. algebraische Verfahren zum Einsatz, die in der Lage sind, unabhängige miteinander vermischte Signale zu trennen ohne nähere Kenntnisse über den Mischungsprozess zu haben (Blind Source Separation, BSS). Eine häufig verwendete BSS-Technik in der BCI-Forschung ist die Independent Component Analysis (ICA), die mit unterschiedlichen Algorithmen auf das EEG angewandt wird, um Rauschen oder Artefakte zu entfernen [58]. Es werden aber auch temporale Dekorrelation [69], Transformationen [55][94] und diverse andere Verfahren [119] sowie Kombinationen der unterschiedlichen Methoden verwendet.

1.5.3 Parameterextraktion

Bei der Ableitung des EEGs wird eine große Datenmenge erzeugt, die zur Klassifikation ungeeignet ist. Das Ziel der Parameterextraktion ist daher die Reduzierung dieses komplexen Datensatzes auf eine möglichst geringe Anzahl von Parametern, die aber trotzdem eine zuverlässige Klassifikation ermöglichen sollen. Die Wahl der verwendeten Parameter ist stark abhängig von der experimentellen Strategie (SCP, EP, ERS...), dem Ableitort, der Anzahl der abgeleiteten Kanäle, den zu unterscheidenden Klassen und den individuellen mentalen Fähigkeiten des Probanden.

Zur Datenreduzierung auf eine geringe Anzahl von Parametern steht eine große Anzahl von mathematischen Methoden und Algorithmen zur Verfügung, die einzeln bzw. in Kombination auf die EEG-Signale angewandt werden. Dabei wird nach zeitabhängigen, ortsabhängigen und zeit- und ortsabhängigen Analyseverfahren bzw. inversen Modellen unterschieden: [66]

a) Zeitabhängige Analyseverfahren

- Fourier-Transformation
- Wavelet-Transformation
- Autoregressive Modelle
- Bandpass-Filter
- Template Matching
- Kalman-Filter
- Spike-Detektion

b) Ortsabhängige Analyseverfahren

- Laplace-Filter
- Principal Component Analysis
- Independent Component Analysis
- Common spatial patterns
- Analyse der Amplituden
- Verhältnisse und Differenzen

c) Zeit- und ortsabhängige Analyseverfahren

- Multivariate autoregressive Modelle
- zeit- und ortsabhängige Komponentenanalyse
- Kohärenz

d) Inverse Modelle

1.5.4 Klassifikation

Mit Hilfe der extrahierten Parameter nimmt ein Klassifikator die Zuordnung zu den gesuchten Klassen vor. Es kann zwischen synchroner (computer driven) und asynchroner (user driven) Klassifikation unterschieden werden. Bei der synchronen Klassifikation findet die Detektion nach klassenspezifischen Parametern und deren Zuordnung in einem vordefinierten Zeitfenster, bei der asynchronen findet die Klassifikation kontinuierlich statt. Die einfachste Form eines Klassifikators ist die Schwellwertdetektion, wobei ein Parameter (z.B. eine Bandleistung) mit einem vordefinierten Grenzwert verglichen wird. Da vor allem bei mehrklassigen Problemstellungen die Komplexität der Klassifikation stark ansteigt, kommen weitere mathematische Methoden und Algorithmen zur Anwendung: [66]

- Lineare Diskriminanzanalyse
- Quadratische Diskriminanzanalyse
- Support Vector Machines
- Gaußsche Mischverteilung
- Hidden Markov Model
- Entscheidungsbaum
- Lernende Vektorquantisierung
- Neuronale Netze
- k-nearest-neighbors
- Perzeptron
- reguläre und adaptive Regression

1.5.5 Applikations-Interface

Nach erfolgter Klassifikation muss diese Information in Steuerungsbefehle für Anwendungen übersetzt werden. Für jede Art von Anwendung ist ein eigenes Applikations-Interface zu entwerfen. Im Falle einer Softwareanwendung ist dies ein Programm, das die Ergebnisse in zur Anwendung passende Befehle übersetzt. Soll eine Hardware, wie ein Rollstuhl oder eine Prothese gesteuert werden, so muss eine, auf diese Hardware angepasste, Schaltung entworfen und diese mittels Software angesteuert werden. Die Signalübertragung der Steuerungsbefehle kann drahtlos oder kabelgebunden über eine verfügbare Schnittstelle des Signalverarbeitungs-Rechners erfolgen.

1.5.6 Anwendungen

Der Großteil der Anwendungen, die in den letzten Jahren im Rahmen der BCI-Forschung entstanden sind, sind Programme zu Übungs- bzw. Demonstrationszwecken. Der Schwerpunkt in der Programmentwicklung sind die Wiederherstellung der Kommunikationsfähigkeit (ALS) [92], die Erhöhung der Mobilität (Tetraplegie) [67] und die Wiederherstellung von Körperfunktionen (Tetraplegie, Amputation) [73][72] von beeinträchtigten Personen. Um jedoch BCI-Systeme vom Labor in den Alltag entlassen zu können, um einen größeren Anwenderkreis zu erschließen und damit auch die Anzahl der verfügbaren Anwendungen zu erhöhen, sind noch einige Entwicklungen und Verbesserungen der bestehenden Systeme notwendig: [68]

- Klassifikationsgenauigkeit

In Softwareanwendungen sind Fehler in der Klassifikation unproblematisch, außer, dass dadurch die Transferrate negativ beeinflusst wird.

- Transferrate

Um mit BCI-Systemen eventuell auch nicht beeinträchtigte Personen anzusprechen, müsste die Transferrate erhöht werden, die derzeit bei ca. 24 Bit/min liegt. Auch für Kommunikationsprogramme ist diese Rate zu niedrig, um eine natürliche Kommunikation zu ermöglichen (20 Buchstaben/min).

- Vereinfachung

Das Applizieren des Systems am Benutzer muss vereinfacht werden, da derzeitig erhältliche Systeme einen hohen Aufwand verursachen und ein großer Teil der Zielgruppe diese nicht selbständig verwenden können. So könnte die Entwicklung von zuverlässigen Trockenelektroden einen wesentlichen Fortschritt mit sich bringen.

- Umwelteinflüsse

Die meisten BCI-Systeme werden unter Laborbedingungen getestet und evaluiert und die Probanden sind praktisch von äußeren Umwelteinflüssen abgeschirmt. Es wird noch zu untersuchen sein, welche negativen Auswirkungen diese Ablenkungen bei Verwendung des BCI z.B. im Straßenverkehr auf die Performance ausüben.

Diese und weitere Probleme, wie Preis, Design oder Systemgröße, sind in den nächsten Jahren so weit wie möglich zu lösen, um die Anzahl der verfügbaren Anwendungen zu erhöhen.

1.5.7 Feedback

Das Feedback schließt den Kreis und stellt die Verbindung zwischen zwei unabhängigen Systemen dar, dem BCI-System auf der einen Seite und dem Benutzer auf der anderen. Das präsentierte Feedback spielt vor allem in BCI-Systemen, die nicht auf EP basieren, eine Rolle, da bei EP-basierenden Systemen verschiedene Reize dargeboten werden und die Steuerung durch Aufmerksamkeit auf einen dieser Stimuli erfolgt.

Wird eine Hardware, wie ein Roboter, Rollstuhl, Prothese oder dgl. mit dem BCI gesteuert, so stellt die Reaktion dieser Hardware bereits ein visuelles Feedback dar. In Softwareanwendungen liegt es in der Hand des Programmentwicklers, wie das Feedback gestaltet wird.

Das Feedback kann in verschiedenen Formen dargeboten werden:

- diskret oder kontinuierlich [61],
- abstrakt oder realistisch [77],
- visuell, akustisch oder taktil [65],
- positiv oder negativ [52].

Die Frage, inwieweit das Feedback die Klassifikationsrate beeinflusst, kann nicht allgemein beantwortet werden, da dieser Einfluss zum einen sehr stark von den charakterlichen Eigenschaften des Benützers abhängig ist und zum anderen auch von der Art der Präsentation des Feedbacks. So kann z.B. ein Feedback in Form eines Kontrollbalkens, dessen Länge proportional zu einem Klassifikationsparameter ist, für den einen sehr motivierend wirken, den anderen jedoch frustrieren. Neben diesen psychologischen Aspekten sind auch Veränderungen in den Gehirnmustern durch Feedback dokumentiert [83]. Die Wahl eines geeigneten Feedbacks ist daher ein komplexes Themengebiet und sollte an die BCI-Erfahrung des Benützers angepasst sowie motivierend gestaltet sein [61].

1.6 Marktübersicht

Neben EEG-Standgeräten, die in der neurologischen Forschung und Diagnostik verwendet werden, gibt es mittlerweile auch einige portable EEG-Systeme. Diese lassen sich in zwei Gruppen differenzieren:

- portable EEG Verstärker
- EEG-Headsets

Im folgenden werden kurz die Unterschiede der beiden Systemklassen erläutert und Vertreter beider Gruppen vorgestellt.

1.6.1 Portable EEG-Systeme

Diese Klasse von EEG-Systemen besteht aus einem am Körper (Umhängetasche, Hosentasche o.Ä.) getragenen kleinen EEG-Verstärker. Das EEG wird „klassisch“ mit EEG-Haube und Standardelektroden abgeleitet und über lange Kabel mit dem Verstärker verbunden. Das EEG wird im Verstärkergehäuse digitalisiert und anschließend entweder gespeichert (Datenlogger) oder drahtlos (meist Bluetooth) an einen zum System gehörenden Rechner geschickt und dort gespeichert bzw. visualisiert. Wird ein Laptop verwendet, so bleibt das Gesamtsystem portabel.

1.6.1.1 g.MOBllab von g.tec medical engineering GmbH, Österreich



Abbildung 12: g.MOBllab [87]

Dieser handliche EEG-(Biosignal-)Verstärker (Abbildung 12) besitzt 8 Kanäle mit einer Auflösung von 16 Bit und wird mit einer Abtastrate von 256 Hz betrieben. Die aufgezeichneten Daten können auf einer Micro-SD Karte mit einer maximalen Speicherfähigkeit von 2 GB gespeichert werden oder mittels Bluetooth 2.0 an einen Rechner gesendet werden. Das Gerät wiegt 360 g und wird mit 4 Standard AA-Batterien versorgt, wodurch eine Laufzeit von ca. 25 bis 100 Stunden erreicht wird. g.tec bietet auch Softwarelösungen zur Echtzeit-Datenanalyse auf Basis von Simulink und MATLAB an, wodurch auch BCI-Anwendungen realisiert werden können [87].

1.6.1.2 EEG mobile von SIGMA Medizin-Technik GmbH, Deutschland

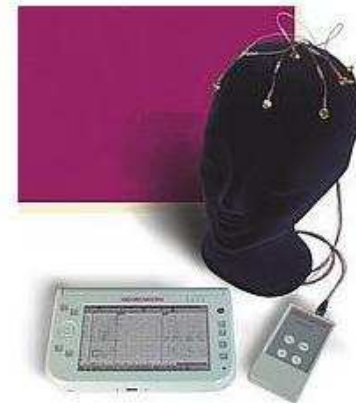


Abbildung 13: EEG mobile [85]

Von SIGMA Medizin-Technik ist ein portables EEG-System (Abbildung 13) mit 1 bis 32 Kanälen, einer Auflösung von 16 Bit und einem Gewicht von nur ca. 200 g erhältlich. Die Abtastrate ist für jeden Kanal frei wählbar, für alle Kanäle gesamt steht eine gesamte Abtastrate von 8 kHz (optional 32 kHz) zur Verfügung. Die Daten können im integrierten SD-Laufwerk aufgezeichnet werden oder die Daten werden per Bluetooth an einen Windows XP Tablet-PC gesendet, auf dem eine Erfassungs- und Auswertungssoftware installiert ist. [85]

1.6.1.3 NeuroScape von Rising Medical Equipment Co., Ltd., China



Abbildung 14: NeuroScape [86]

Unter den Namen NeuroScape stellt die Chinesische Firma Rising Medical Equipment eine Produktpalette von EEG-Systemen mit unterschiedlicher Anzahl von Kanälen (16, 24 und 32 Kanäle), 200 Hz Abtastrate, einer Auflösung von 12 Bit und einer Verstärkung von 10.000 her. Auf der Verstärkerbox (Abbildung 14) ist das 10/20 System abgebildet, um den Elektrodenanschluss für den Bediener des Systems zu vereinfachen. Die Elektrodenimpedanzen werden automatisch ermittelt und mittels LED erfolgt eine Statusanzeige. Die Daten können mittels USB zu einem Rechner transferiert werden und es werden auch Softwaretools zur Datenanalyse, wie Spektralanalyse oder Mapping, angeboten [86].

1.6.2 EEG-Headsets

Wird das gesamte EEG-System in einem auf dem Kopf tragbaren Headset integriert, so spricht man von einem EEG-Headset. Ziel ist die Realisierung eines portablen BCI-Systems um Steuerungsaufgaben zu bewältigen. Dabei kommen miniaturisierte EEG-Verstärker in SMD-Bauweise (Surface mounted device) zum Einsatz oder die Verstärkerschaltung wird als application-specific integrated circuit (ASIC) integriert. Nach der Digitalisierung wird der EEG-Datenstrom per Bluetooth (da lizenzfrei) zu einem PC geschickt um weiter verarbeitet zu werden. EEG-Headsets sind die ersten Vertreter von EEG-Systemen die auch für den Heimanwender bestimmt sind. Um den Tragekomfort zu erhöhen bzw. die Elektrodenmontage zu vereinfachen, können Trockenelektroden zur Ableitung des EEGs verwendet werden, obwohl die Signalqualität noch nicht klinischen Standards entspricht. Hauptanwendungsgebiet ist die freihändige Kontrolle von Spielen, Computer, Fernseher oder Handy. Aktuell gibt es drei kommerzielle Headsets, von zwei weiteren wurden Prototypen präsentiert.

1.6.2.1 Mindwave von NeuroSky, Inc., USA



Abbildung 15: Mindwave [45]

Die Basis für die Produkte von NeuroSky ist ThinkGear, ein ASIC, der einen 1-Kanal EEG-Verstärker, ADC und eine Signalverarbeitungseinheit (On-Board Datenverarbeitung) vereint. Mittels Bluetooth können verschiedene Daten wie die rohen EEG-Daten oder verschiedene Bandleistungsparameter versendet werden. Im mit 512 Hz abgetastete EEG kann mittels mathematischer Algorithmen zwischen zwei mentalen Zuständen, Aufmerksamkeit und Entspannung, sowie Augen blinzeln detektiert werden.

Zum kommerziellen Ersteinsatz kam ThinkGear im MindSet (2007). Darauffolgend kamen 2009 zwei Spiele, der Star Wars Force Trainer von Uncle Milton und Mindflex von Mattel, auf Basis des MindSet auf den Markt.

Im Jahr 2011 erschienen zwei neue BCI-Systeme, XWave zur Steuerung von iPhone/iPad/ iPod bzw. Mindwave (Abbildung 15) für Computeranwendungen die beide für ca. 100 USD käuflich zu erwerben sind. [45]

1.6.2.2 EPOC von Emotiv Systems, Inc., Australien



Abbildung 16: EPOC [44]

Emotiv Systems präsentierte 2009 das EPOC-System (Abbildung 16), ein Steuerungssystem für Windows-PCs. Es verwendet 14 Elektroden sowie einen 2-Achsen Bewegungssensor um bis zu 30 verschiedenen mentale Zustände bzw. Gesichtsausdrücke zu unterscheiden. Es stellt somit kein reines BCI-System dar, da auch EMG Signale und Kopfhaltung/-beschleunigung ausgewertet werden. Zur Verwendung kommen keine Trockenelektroden, sondern vergoldete Elektroden, wobei der Hautkontakt mit einem in Salzlösung getränkten Filzpad erfolgt. Das Signal wird mit 128 Hz abgetastet und hat eine Auflösung von 16 Bit. Mit einem Preis von ca. 300 USD ist EPOC zwar teurer als der MindWave, bietet aber ein breiteres Anwendungsspektrum [44].

1.6.2.3 Enobio von Neuroelectrics, Spanien



Abbildung 17: Enobio [43]

Neuroelectrics, ein Spin-Off von Starlab, vermarktet das von der spanischen Firma Starlab entwickelte und im Jahr 2009 auf den Markt gebrachte EEG-Headset Enobio (Abbildung 17). Das System ist für 8 bzw. 20 Kanäle erhältlich und wiegt nur 65 g. Die Abtastung des EEGs erfolgt mit einer Rate von 500 Hz bei einer Auflösung von 24 bit (50 nV). Zusätzlich ist ein 3-Achsen Beschleunigungssensor verbaut, der mit einer Abtastrate von 100 Hz ausgelesen werden kann. Die erfassten Signale können mit Hilfe von Bluetooth in verschiedenen Datenformaten an einen Rechner gesendet oder mittels SD-Karte gespeichert werden. Die hohe Gleichtaktunterdrückung von 115 dB und das geringe Rauschen von unter $1 \mu\text{V}_{\text{rms}}$ ermöglicht auch die Verwendung von Trockenelektroden, um eine möglichst einfache Handhabung gewährleisten zu können. [43]

1.6.2.4 Prototyp von Imec und Holst Centre, Niederlande



Abbildung 18: Imec Prototyp [46]

Im Februar 2011 präsentierten Imec und das Holst Centre einen Prototypen eines EEG-Headsets (Abbildung 18). Die Elektronik umfasst einen ASIC, Übertragungseinheit (10 m Reichweite, vermutlich Bluetooth Klasse 2) und einen Controller. Das 8-Kanal EEG-System hat laut Hersteller eine Gleichtaktunterdrückung von 120 dB, 1 G Ω Eingangsimpedanz und einen Leistungsbedarf von 9,2 mW bei Übertragung aller 8 Kanäle. Die hohe Eingangsimpedanz soll eine Verwendung von Trockenelektroden möglich machen, derzeit vertraut man allerdings auf Ag/AgCl-Standardelektroden und hofft auf Forschungsfortschritte auf dem Gebiet der Trockenelektroden [46].

1.6.2.5 Prototyp von NeuroFocus Inc., USA



Abbildung 19: Mynd [47]

Der von NeuroFocus im März 2011 vorgestellte Prototyp mit dem Namen Mynd (Abbildung 19) besitzt 21 Elektroden und deckt somit alle im 10/20 System definierten Positionen ab. Das mit Trockenelektroden abgeleitete EEG wird mit 2 kHz abgetastet und kann per Bluetooth zu jedem Bluetooth-fähigen Empfänger gesendet werden. Entwickelt wurde Mynd für die Analyse von menschlichem Kaufverhalten (Neuro-marketing), daher wurde auch ein Eye-Tracking-System implementiert, um die Blickrichtung von Probanden bestimmen zu können. [47]

1.7 Motivation

Im Regelfall wird das EEG in einem klinischen Umfeld zur neuronalen Diagnostik oder unter Laborbedingungen zu Forschungszwecken abgeleitet. Die Patienten/Probanden sind so von negativen äußeren Umwelteinflüssen abgeschirmt und die gewonnenen Datensätze dadurch vergleich- und reproduzierbar. Dieses Messmilieu entspricht jedoch nicht der natürlichen Umgebung der Probanden und die Messung kann durch Anspannung oder Stress negativ beeinflusst werden. Es ist daher von großem Interesse, EEG-Messdaten von Personen in ihrem natürlichen Umfeld und in Alltagssituationen zu gewinnen. Mit Hilfe von tragbaren EEG-Headsets können solche EEG-Datensätze gewonnen und auch Langzeitmessungen ermöglicht werden. Wird das Headset mit einem Prozessor zur Online-Signalanalyse erweitert, so sind auch portable BCI-Systeme umsetzbar, wodurch auch BCI-Heimanwendungen realisiert werden können.

1.8 Zielsetzung

Im Rahmen dieser Arbeit soll ein EEG-Headset entwickelt und ein Prototyp angefertigt werden und abschließend eine einfache BCI-Anwendung implementiert werden. Sämtliche benötigte elektronische Komponenten sollen auf dem Kopf tragbar sein und folgende Spezifikationen erfüllen:

- 3,3 V Spannungsversorgung
- 1-Kanal EEG-Verstärker
- einfache Erweiterbarkeit auf mehrere Kanäle
- ADC mit 250 Hz Abtastrate
- On-Board DSP zur Signalmanipulation
- Online-Signalanalyse
- Möglichkeit zur Signalgenerierung für Steuerungsaufgaben
- Betriebsdauer von mind. 4 Stunden
- Wiederverwendbarkeit für weitere Projekte
- bedienungsfreundlich
- kostengünstig

Das Projekt gliedert sich dabei in 3 Hauptpunkte:

1. Zusammenstellung, Entwurf und Anfertigung der benötigten Hardware,
2. Auswahl und Implementierung von geeigneten Softwaretools zur Programmierung,
3. Realisierung eines Brain-Switch durch Detektion von Fuß-Bewegungsvorstellung.

2 Hardware

In diesem Kapitel werden sämtliche verwendeten Hardwarekomponenten zur Realisierung des EEG-Headsets vorgestellt. Abbildung 20 zeigt eine Übersicht über das gesamte System:

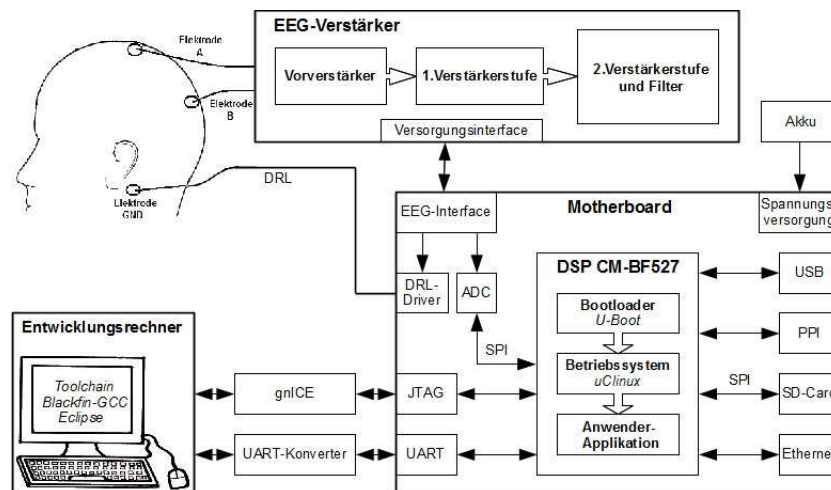


Abbildung 20: Blockschaltbild des Gesamtsystems

Herzstück ist eine zentrale, selbst angefertigte und als Motherboard bezeichnete Platine, auf der sich die Spannungsversorgung, der DSP mit den verwendeten Schnittstellen, der ADC, der Driven-Right-Leg-Driver sowie das Anschluss- und Versorgungsinterface für den ebenfalls selbst angefertigten 1 Kanal EEG-Verstärker befindet.

2.1 Allgemeines

Aufgrund der notwendigen Miniaturisierung sind die Schaltungen in SMD-Bauweise ausgeführt. Da die Bestückung und Verlötlung der gefertigten Platinen händisch erfolgte, wurde die SMD-Rastergröße mit 0805 gewählt.

Alle elektronischen Bauelemente kamen aufgrund ihrer technischen Daten, Betriebsspannungsbereich, Preis und Verfügbarkeit in Verwendung und sind mit Verweis auf das entsprechende Datenblatt, der Bezugsquelle sowie dem Preis im Anhang A aufgelistet. Im Anhang B sind die Schaltungen in der Gesamtübersicht dargestellt. Der Schaltungsentwurf und das Routing erfolgte mit EAGLE 5.1.0 light, die anschließende Fertigung der Platinen erfolgte durch die Firma LeitOn GmbH.

2.2 EEG-Verstärker

Schaltungstechnischer Ausgangspunkt für die Entwicklung des EEG-Verstärkers ist das OpenEEG project [104], eine Veröffentlichung eines EEG-Verstärkers mit Schaltungsdesign und Dokumentation. Die Schaltung ist für eine empfohlene Betriebsspannung von 9 bis 12 V bzw. optional durch kleine Änderungen von 5 V ausgelegt, besitzt einen nominellen Verstärkungsfaktor von 7812,5 und einen Bandpass von 0,4 Hz bis 59 Hz.

Diese grundlegenden Daten zeigen bereits, dass eine Adaption der Schaltung nötig ist, die grundsätzliche Konzeption der Verstärkerschaltung allerdings erhalten bleiben kann, da bereits in vorangegangenen Projekten positive Ergebnisse damit erzielt wurden. Es soll daher lediglich die Versorgungsspannung auf 3,3 V gesenkt, die Verstärkung auf einen Eingangsspannungsbereich von 750 μ V angepasst sowie die Grenzfrequenzen des Filters adaptiert werden. Dies hat jedoch zur Folge, dass die wesentlichen elektronischen Bauteile durch andere ersetzt und deren äußere Beschaltung adaptiert werden müssen.

Die Verstärkerplatine ist für zwei EEG-Kanäle ausgelegt, es wurde jedoch nur Kanal 1 bestückt.

2.2.1 Vorverstärker

Um das Potentialfeld auf der Kopfhaut nicht zu belasten, kommt als Vorverstärker ein Single-Supply, Rail to Rail Präzisionsinstrumentenverstärker, der AD8553 von Analog Device, zum Einsatz. Er zeichnet sich durch seine hohe Gleichtaktunterdrückung von 130 dB, einem geringem Verstärkerrauschen von 0,7 μ V und seinem hohen Verstärkungs-Bandbreitenprodukt von 1 MHz aus. Die äußere Beschaltung (Abbildung 21) ist dem Datenblatt entnommen, mit Ausnahme der Teilung des verstärkungsbestimmenden Widerstandes, der in zwei gleich große Widerstände geteilt wurde, um die gemeinsame Gleichtaktspannung abgreifen zu können.

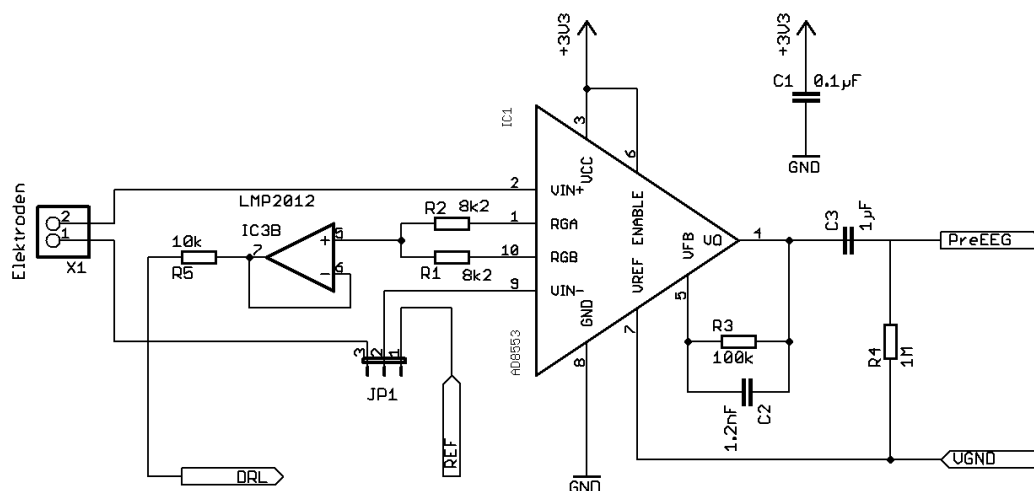


Abbildung 21: Schaltung Vorverstärker

Die EEG-Elektroden werden an X1 angeschlossen und das Signal an die Differenzeingänge weitergeleitet, wobei mit Hilfe des Jumpers JP1 zwischen monopolarer (Jumper auf Pin 1-2) bzw. bipolarer (Jumper auf 2-3) Ableitung gewählt werden kann. Bei der monopolarer Ableitung wird nur eine EEG-Elektrode an Pin 2 von X1 benötigt, die Verbindung zur Referenzelektrode erfolgt über das Versorgungsinterface vom Motherboard.

Die Verstärkung der Schaltung berechnet sich aus:

$$G_0 = 2 * \left(\frac{R_3}{R_1 + R_2} \right)$$

Bei einer gewählten Verstärkung von $G = 12$ und einem im Datenblatt empfohlenen Widerstand $R_3 = 100 \text{ k}\Omega$ errechnet sich R_1 und R_2 zu:

$$12 = 2 * \left(\frac{100000 \Omega}{R_1 + R_2} \right) \rightarrow R_1 + R_2 = 16666,66 \Omega \rightarrow R_1 = R_2 = 8333,33 \Omega$$

Der nächstgelegene Widerstandswert der Normreihe E24 ist $R_1 = R_2 = 8,2 \text{ k}\Omega$, dies führt zu einer tatsächlichen Verstärkung von:

$$G_0 = 2 * \left(\frac{R_3}{R_1 + R_2} \right) = 2 * \left(\frac{100 \text{ k}\Omega}{16,4 \text{ k}\Omega} \right) = 12,2$$

Durch den symmetrischen internen Aufbau des Verstärkers kann zwischen R_1 und R_2 die Gleichtaktspannung (common mode voltage), also jene Signale, die zeitgleich und mit gleicher Phase an beiden Eingängen anliegen, abgegriffen werden. Diese Spannung wird mit dem aus IC3B gebildeten Spannungsfolger entkoppelt, um dieses Potential nicht zu belasten und kann als Bezugsspannung am Patienten (Ground) verwendet werden, um diese Gleichtaktspannung zu kompensieren. In modernen Schaltungsdesigns wird diese Spannung jedoch einer aus dem EKG-Bereich etablierten Schaltung, dem Driven-Right-Leg-Driver (DRL) [114], zugeführt, um diesen Effekt noch weiter zu verstärken. Diese Schaltung befindet sich am Motherboard.

Die Kapazität C_1 ist ein Stützkondensator um Störungen in der Spannungsversorgung zu verringern und wird bei single-supply mit $C_1 = 0,1 \mu\text{F}$ empfohlen.

Der Tiefpass, der von C_2 und R_3 gebildet wird, begrenzt die Bandbreite des Signals, um innerhalb des Leistungsvermögens des Instrumenten-Verstärkers zu bleiben. Die empfohlene Bandbreite ist laut Datenblatt $B = 1400 \text{ Hz}$, wodurch sich C_2 berechnet zu:

$$C_2 = \frac{1}{(B * 2 * \pi * R_3)} = \frac{1}{(1400 * 2 * \pi * 100000 \Omega)} = 1137 \text{ pF} .$$

Gewählt wurde eine Kapazität von $C_2 = 1200 \text{ pF}$, wodurch sich eine Signalbandbreite von $B = 1326,3 \text{ Hz}$ ergibt.

C_3 und R_4 bilden gemeinsam einen Hochpass, der zur Entfernung von Gleichspannungsschwankungen benötigt wird. Die Grenzfrequenz sollte nahe an Null liegen und R_4 sollte möglichst groß sein.

Die Grenzfrequenz f_g berechnet sich aus:

$$f_g = \frac{1}{(2 * \pi * R_4 * C_3)}$$

Durch Wahl der Grenzfrequenz $f_g = 0,2 \text{ Hz}$ ergibt sich

$$R_4 * C_3 = \frac{1}{(2 * \pi * f_g)} = \frac{1}{(2 * \pi * 0,2 \text{ Hz})} = 0,795 \text{ s}$$

Wird $R_4 = 1 \text{ M}\Omega$ gewählt, erhält man

$$C_3 = \frac{0,795 \text{ s}}{(10 * 10^6 \Omega)} = 795 * 10^{-9} \text{ F} = 795 \text{ nF}$$

Der nächstgelegene verfügbare Kapazitätswert ist $1 \mu\text{F}$, welcher auch für C_3 gewählt wurde. Dadurch errechnet sich die tatsächliche Grenzfrequenz f_g mit:

$$R_4 = 1 \text{ M}\Omega$$

$$C_3 = 1 \mu\text{F}$$

$$f_g = \frac{1}{(2 * \pi * R_4 * C_3)} = \frac{1}{(2 * \pi * 1 * 10^6 \Omega * 1 * 10^{-6} \text{ F})} = 0,159 \text{ Hz}$$

2.2.2 Erste Verstärkerstufe

Das vorverstärkte EEG (PreEEG) wird nun einer ersten Verstärkerstufe (Abbildung 22) zugeführt. Es kommt ein nicht invertierender Verstärker zum Einsatz, der mit Hilfe des Präzisionsoperationsverstärker LMP2012 von National Semiconductor gebildet wird. Im 8-Pin MSOP-Gehäuse sind jeweils 2 Operationsverstärker (OPV) verbaut, die mit A bzw. B gekennzeichnet werden.

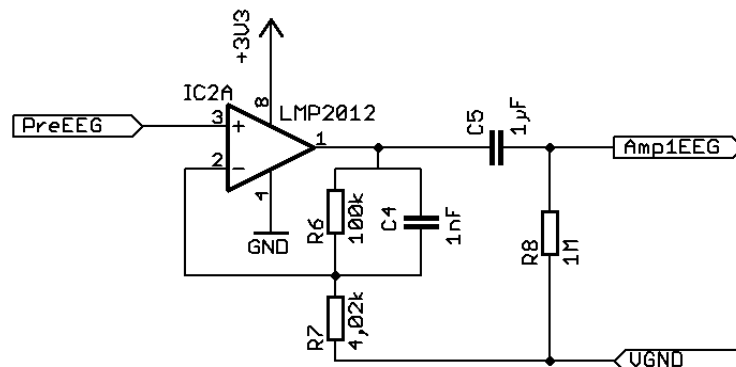


Abbildung 22: Schaltung 1. Verstärkerstufe

Der Verstärkungsfaktor G wird mit dem Spannungsteiler R_6 und R_7 festgelegt und errechnet sich aus

$$G_1 = \frac{(R_6 + R_7)}{R_7}$$

Bei einer gewählten Verstärkung von 26 ergibt sich durch Umformen der Formel ein Widerstandsverhältnis von R_6 : $R_7 = 25 : 1$.

Bei einer Wahl von $R_6 = 100 \text{ k}\Omega$ ergibt sich:

$$R_7 = \frac{R_6}{25} = \frac{(100 \text{ k}\Omega)}{25} = 4 \text{ k}\Omega$$

Für die Wahl des nächstgelegenen erhältlichen Widerstandswertes von $4,02 \text{ k}\Omega$ gilt:

$$R_6 = 100 \text{ k}\Omega$$

$$R_7 = 4,02 \text{ k}\Omega$$

$$G_1 = \frac{(R_6 + R_7)}{R_7} = \frac{(100 \text{ k}\Omega + 4,02 \text{ k}\Omega)}{(4,02 \text{ k}\Omega)} = 25,9$$

Die Kapazität C_4 dient zur Bandbreitenbegrenzung und bildet mit R_6 einen Tiefpass mit einer Grenzfrequenz von

$$f_g = \frac{1}{(2 * \pi * R_6 * C_4)}$$

Der begrenzende Faktor ist das Verstärkungs-Bandbreiten-Produkt (Gain-Bandwidth-Product, GBW) des OPV und wird laut Datenblatt mit 3 MHz angegeben. Die maximal verfügbare Bandbreite beträgt demnach mit einer 26-fachen Verstärkung $115,5 \text{ kHz}$. Aufgrund der Verfügbarkeit wurde $C_4 = 1 \text{ nF}$ gewählt, wodurch sich eine Grenzfrequenz von

$$f_g = \frac{1}{(2 * \pi * R_6 * C_4)} = \frac{1}{(2 * \pi * 100 * 10^3 \Omega * 10^{-9})} = 1591,6 \text{ Hz}$$

ergibt. Die Bandbreite ist daher ausreichend begrenzt und liegt innerhalb des Leistungsvermögens des OPV.

Der darauffolgende von C_5 und R_8 gebildete Hochpass entfernt Gleichspannungsschwankungen und besitzt die gleichen Werte wie jener aus C_3 und R_4 bestehende:

$$C_5 = C_3 = 1 \mu\text{F}$$

$$R_8 = R_4 = 1 \text{ M}\Omega$$

2.2.3 Zweite Verstärkerstufe und Filter

Die zweite und letzte Verstärkerstufe wird mit einer Sallen-Key [100] Verstärkerschaltung realisiert (Abbildung 23). Sie erfüllt zum einen die Aufgabe einer letzten Verstärkung, zum anderen bildet die Schaltung einen Anti-Aliasing-Filter, da das Signal anschließend dem ADC zugeführt wird. Als Ausgangspunkt wurde die äußere Beschaltung des OpenEEG-Projektes herangezogen.

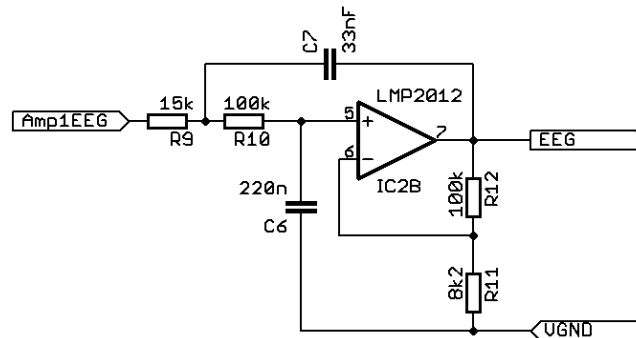


Abbildung 23: Schaltung 2.Verstärkerstufe

Der Verstärkungsfaktor wird durch den aus R_{11} und R_{12} gebildeten Spannungsteiler bestimmt und errechnet sich aus:

$$G_2 = 1 + \frac{R_{12}}{R_{11}}$$

Durch Einsetzen der gewählten Widerstände erhält man:

$$R_{12} = 100 \text{ k}\Omega$$

$$R_{11} = 8,2 \text{ k}\Omega$$

$$G_2 = 1 + \frac{R_{12}}{R_{11}} = 1 + \frac{100\text{k}}{8,2\text{k}} = 13,2$$

Nach dieser letzten Verstärkung ergibt sich ein Gesamtverstärkungsfaktor des EEG-Verstärkers von

$$G = G_0 * G_1 * G_2 = 12,2 * 25,9 * 13,2 = 4171$$

und somit einen erfassbarer Eingangsspannungsbereich von

$$U_{max} = \frac{(3,3V)}{G} = \frac{(3,3V)}{4171} = 791 \mu V \text{ .}$$

Sämtlichen anderen Bauteile bilden ein Tiefpass-Filter 2.Ordnung, um gewährleisten zu können, dass durch den späteren Abtastvorgang kein Aliasing auftritt, das heißt, dass keine Frequenzen über der halben Abtastfrequenz im Signal enthalten sind (Shannon Theorem). Bei einer Abtastfrequenz von 250 Hz müssten demnach sämtliche Frequenzkomponenten über 125 Hz aus dem Signal gefiltert werden. In der Analyse des EEGs spielen meist nur Spektralkomponenten bis 40 Hz eine Rolle, es bietet sich daher an, die Grenzfrequenz des Filters auf knapp unter 50 Hz zu legen, um induzierte Störungen durch die 50 Hz-Netzspannung zu dämpfen.

Die Grenzfrequenz der Schaltung lässt sich mit folgender Vereinfachung berechnen:

$$R_9 = m * R$$

$$R = R_{10}$$

$$C = C_6$$

$$C_7 = n * C$$

$$f_c = \frac{1}{(2 * \pi * R * C * \sqrt{m * n})}$$

Um auf eine Grenzfrequenz von knapp unter 50 Hz zu kommen, werden, bei Verwendung der im OpenEEG Projekt angegebenen Kapazitäten, folgende Werte für die Widerstände herangezogen und die Grenzfrequenz berechnet. Es ergibt sich für:

$$C_7 = 33 \text{ nF}$$

$$C_6 = C = 220 \text{ nF}$$

$$n = \frac{C_7}{C_6} = \frac{(33 \text{ nF})}{(220 \text{ nF})} = 0,15$$

$$R_9 = 15 \text{ k}\Omega$$

$$R_{10} = R = 100 \text{ k}\Omega$$

$$m = \frac{R_9}{R_{10}} = \frac{(15 \text{ k}\Omega)}{(100 \text{ k}\Omega)} = 0,15$$

$$f_c = \frac{1}{(2 * \pi * R * C * \sqrt{m * n})} = \frac{1}{(2 * \pi * 100 * 10^3 \Omega * 220 * 10^{-9} \text{ F} * \sqrt{0,15 * 0,15})} = 48,23 \text{ Hz}$$

2.2.4 Versorgungsinterface

Die Verbindung des Verstärkers mit dem Motherboard erfolgt über die 7-polige Buchsenleiste X2. Die Pinbelegung ist in Abbildung 24 dargestellt:

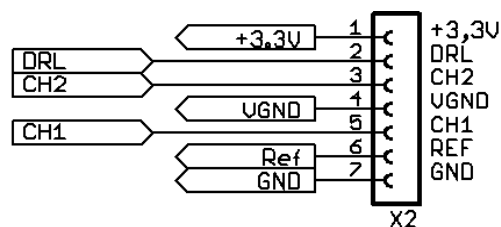


Abbildung 24: EEG-Versorgungsinterface

Die Spannungspfeile, die von X2 nach links orientiert sind, entsprechen vom Motherboard stammenden Signalen. Dies sind die Spannungsversorgung und das virtuelle Ground (VGND), sowie das Signal der Referenzelektrode im Falle einer monopolaren Ableitung. Nach rechts zu X2 orientierte Pfeile symbolisieren die vom EEG-Verstärker generierten, analogen EEG-Kanäle CH1 bzw. CH2 sowie die bereits entkoppelten und summierten Gleichtaktspannungen, die zum Motherboard geschickt werden.

2.2.5 Board-Layout

Abbildung 25 zeigt das Routing der EEG-Verstärker-Platine, links ist die Ansicht von oben, rechts die Ansicht von unten.

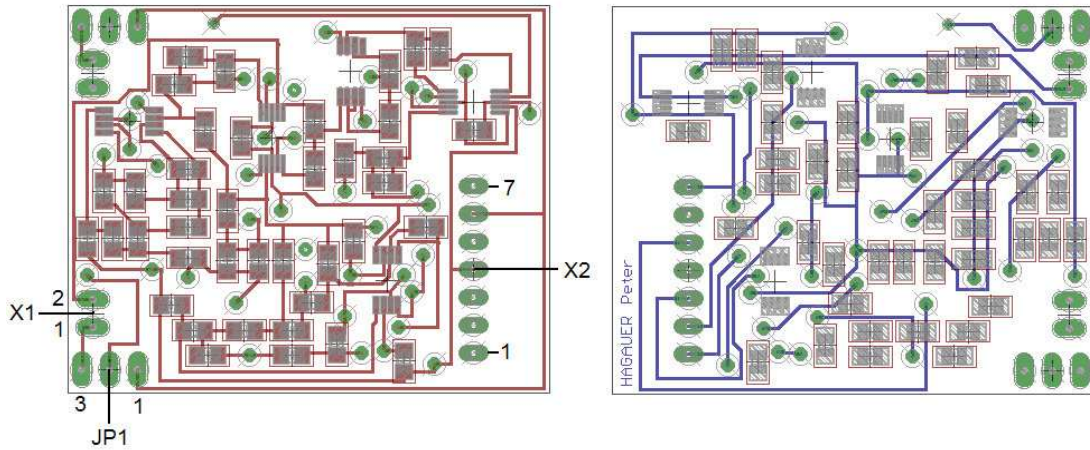


Abbildung 25: Routing der EEG-Verstärkerplatine

In der Darstellung sind die verwendeten Buchsenleisten X1 und X2 sowie der Wahljumper und deren Orientierung auf der Platine eingezeichnet.

2.3 Digitaler Signalprozessor

Für die Analyse eines digital vorliegenden EEGs wird ein Rechenwerk benötigt. Am besten geeignet für diese Aufgabe sind digitale Signalprozessoren (DSP), da diese für die numerische Bearbeitung abgetasteter Signale optimiert sind. Ein DSP ist gekennzeichnet durch: [7]

- Harvard-Speicherarchitektur

Logische und physikalische Trennung des Speicherbereichs von Programm und Daten, daher gleichzeitiger Zugriff auf Programm und Daten möglich;

- Arithmetisch-logische Einheit (ALU)

In der arithmetisch-logischen Einheit werden Additionen, Subtraktionen sowie Bitmanipulationen in einem Taktzyklus ausgeführt;

- Multiplizierer-Akkumulator

Im Gegensatz zu Microcontrollern ist der Multiplizierer bei DSPs in Hardware implementiert, dadurch ist es möglich, Multiplikationen in einem Prozessortakt durchzuführen. Durch Kombination mit einem Akkumulator kann innerhalb eines Taktes das Ergebnis der Multiplikation zweier Zahlen auch noch zum vorherigen Zwischenergebnis addiert/subtrahiert werden;

- Shifter

Für logische bzw. arithmetische Schiebeoperationen steht ein Barrel-Shifter zur Verfügung, der innerhalb eines Taktes von der Bitzahl unabhängige Schiebeoperationen ausführt.

Für die meisten Prozessoren sind Entwicklungsboards erhältlich, die die Entwicklung von Prototypen erleichtern. Leider sind diese für dieses Projekt nicht geeignet, da sie alle zu groß für eine Kopfmontage sind. Auch im Sinne der Wiederverwertbarkeit für mehrere Projekte kommt daher das steckbare Prozessormodul CM-BF527 (Blockdiagramm in Abbildung 26 dargestellt) von Bluetechnix [33] in Verwendung.

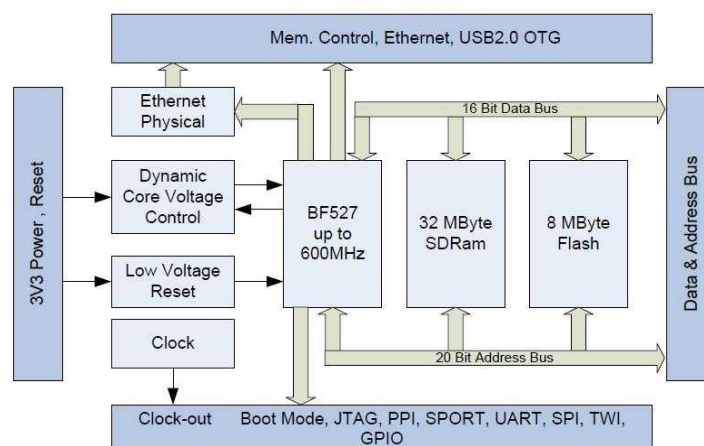


Abbildung 26: Blockdiagramm des Prozessormoduls [33]

Das Modul ist äußerst kompakt (31 x 36 mm) und wird mit zwei 60-poligen Stecker mit dem Motherboard verbunden.

Neben dem DSP BF-527 der Blackfin-Reihe von Analog Device [2][3][24] ist auch ein 10/100 Mbit Ethernet-Treiberbaustein von Micrel sowie USB 2.0 OTG verbaut. Das Modul besitzt folgende technischen Spezifikationen:

- Prozessor ADSP-BF527 von Analog Device mit 600 MHz
- 32 MB SDRAM
- 4 x 2 MB Flash
- Low Voltage Reset
- Dynamic Core Voltage Controller
- Real Time Clock
- 3,3 V Spannungsversorgung
- 350 mA Strombedarf

Über die zwei 60-poligen Stecker kann auf folgende Funktionen und Schnittstellen zugegriffen werden:

- Datenbus
- Adressbus
- Kontrollsignale
- USB 2.0 OTG
- Ethernet
- SPI (Serial Peripheral Interface)
- UART (Universal Asynchronous Receiver Transmitter)
- SPORT (serial port)
- GPIO (General Purpose Input/Output)
- JTAG (Joint Test Action Group)
- TWI (Two Wire Interface)
- PPI (Parallel Peripheral Interface)

2.4 Motherboard

Das gefertigte Motherboard ist die zentrale Platine des EEG-Headsets. Auf der 80 x 50 mm große Platine befinden sich die Steckverbindungen zum Prozessormodul, sämtliche verwendete Schnittstellen, die Spannungsversorgung, der ADC, der Driven-Right-Leg-Driver sowie die Elektrodenanschlüsse für die Ground- und Referenzelektrode.

2.4.1 Spannungsversorgung

Für die elektrische Versorgung der einzelnen Komponenten wird eine stabile, saubere 3,3V Gleichspannung benötigt. Der verwendete Präzisions-Spannungsregler TPS7A4533 (Abbildung 27) von Texas Instruments ist ein Single-Supply-LDO (low dropout) Regler, mit einer Dropoutspannung von nur 300 mV. Er ist daher für den Batteriebetrieb geeignet, da bis zu einer Eingangsspannung von 3,6V die Ausgangsspannung konstant auf 3,3V gehalten wird.

Der Regler besitzt einen Eingangsspannungsbereich bis 20V und kann daher mit einer breiten Palette von Energiespeichern bzw. Netzgeräten gespeist werden.

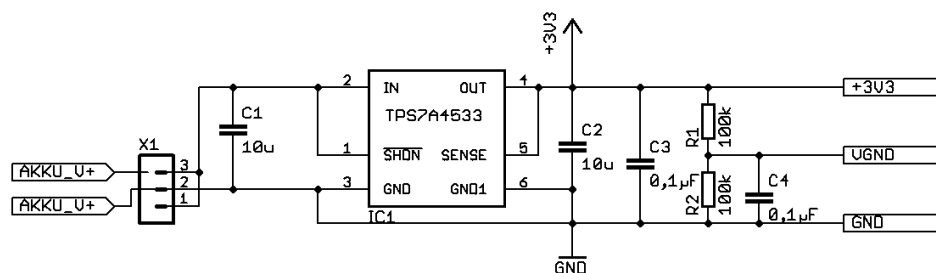


Abbildung 27: Schaltung Spannungsversorgung

Die Eingangsspannung wird an der 3-poligen Buchsenleiste X1 angelegt, wobei in der Mitte (Pin 2) das elektrische Bezugspotential (Ground) anliegt und an den beiden außen anliegenden die Versorgungsspannung (verpolungssicher). Die Kondensatoren C_1 und C_2 stützen die Eingangs- bzw. Ausgangsspannung, deren Kapazitätswerte wurden dem Datenblatt entnommen. Die 3,3V-Ausgangsspannung des Reglers stellt die Spannungsversorgung für alle elektrischen Komponenten dar und wird in den Schaltplänen mit +3V3 bezeichnet, der Bezugspunkt, das elektrische Nullpotential, mit GND. Mit Hilfe des aus R_1 und R_2 gebildeten Spannungsteilers wird eine virtuelle Masse (VGND) erzeugt und diese stellt das logische (nicht elektrische) Nullpotential dar. Dadurch ist es trotz Single-Supply möglich, positive als auch negative Spannungsschwankungen in Bezug auf VGND zu erfassen. Es ist jedoch zu beachten, das VGND nicht belastet werden darf, da es sich mit Belastung in Richtung GND verschieben würde (belasteter Spannungsteiler).

2.4.2 Prozessorschnittstelle

Mit zwei Buchsen von Hirose wird die Steckverbindung des Prozessor-Moduls mit dem Motherboard gewährleistet. In Abbildung 28 wird dies mit X2 (Pins 1 bis 60) auf der linken Seite bzw. X3 (Pins 61 bis 120) auf der rechten Seite des Schaltsymbols dargestellt.

Die Kapazitäten C_5 und C_6 sind empfohlene Größen und wurden dem Datenblatt entnommen.

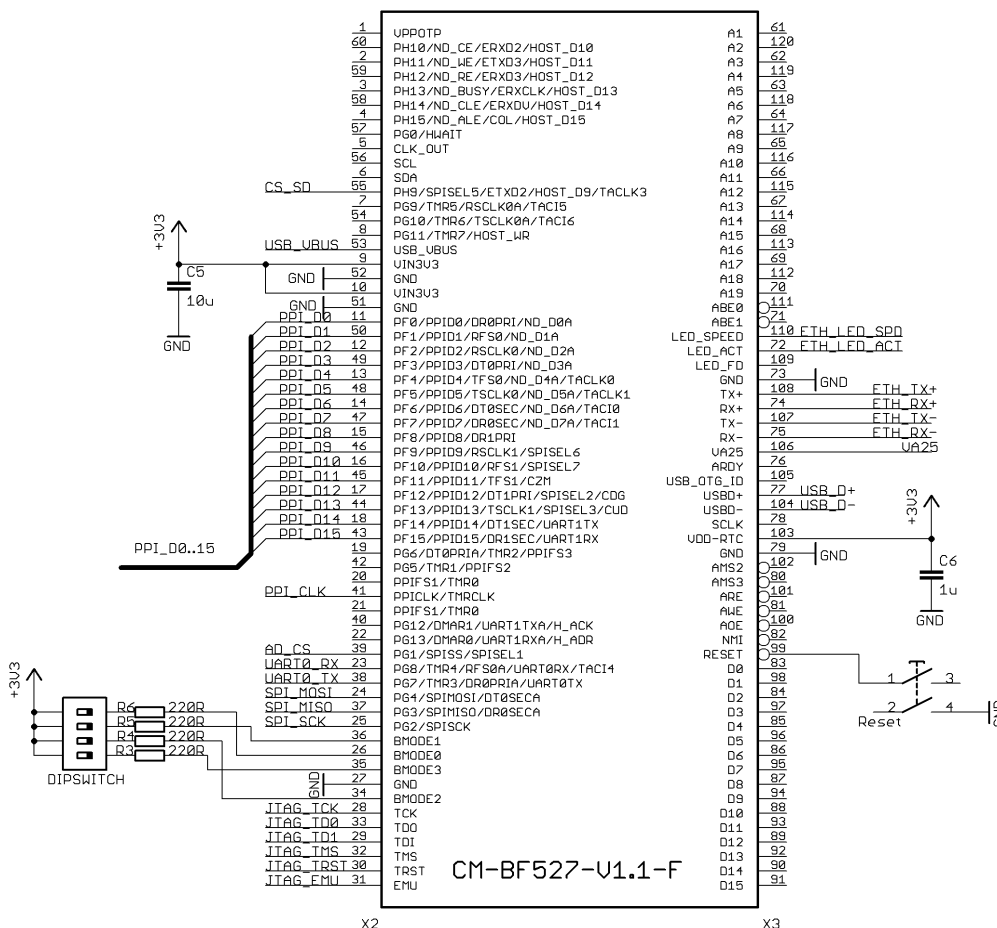


Abbildung 28: Schaltung Prozessor-Schnittstelle

Betriebsnotwendig ist neben der Spannungsversorgung des Prozessormoduls auch noch der RESET-Kreis, realisiert durch einen SMD-Taster DTSM-61 von Diptronics, sowie die BOOT-Mode Auswahl.

Die Auswahl des Boot-Modes erfolgt mit einem SMD-Tip-Switch und wird nach Tabelle 2 gewählt:

Switch Settings BM3,BM2,BM1,BM0	Boot Mode	Description
0000	0 (default)	Idle - No bootReserved
0001	1	Boot from 8- or 16-bit external flash memory
0010	2	Boot from 16-bit asynchronous FIFO.
0011	3	Boot from serial SPI memory (EEPROM or flash)
0100	4	Boot from SPI host device
0101	5	Boot from serial TWI memory (EEPROM/flash)
0110	6	Boot from TWI host
0111	7	Boot from UART0 Host
1000	8	Boot from UART1 Host
1001	9	Reserved
1010	10	Boot from SDRAM
1011	11	Boot from OTP memory
1100	12	Boot from 8-bit NAND flash via NFC using PORTF data pins
1101	13	Boot from 8-bit NAND flash via NFC using PORTH data pins
1110	14	Boot from 16-Bit Host DMA
1111	15	Boot from 8-Bit Host DMA

Tabelle 2: Auswahl des Boot-Modes [33]

Im weiteren Projekt wird stets BOOT-Mode 1 verwendet und daher wurde der Tip-Switch nicht bestückt, sondern durch ein kleines Kabel ersetzt.

Auf die, mit den zusätzlich abgegriffenen Pins verbundene Hardware und Schnittstellen wird einzeln in den folgenden Kapiteln eingegangen.

2.4.3 Joint Test Action Group

Unter Joint Test Action Group (JTAG) versteht man den IEEE-Standard 1149.1, eine Ansammlung von Tools, um integrierte Schaltungen während der Laufzeit testen oder debuggen zu können. Die Verbindung wird mit einem 14-poligen Stecker (2x7 Pinleiste) erreicht, wobei eine Verpolung nicht möglich ist, da bei den korrespondierenden Buchsen Pin 3 verschlossen ist und daher auch Pin 3 in der Pinleiste entfernt werden muss. In Abbildung 29 ist die Beschaltung und Pinbelegung des JTAG-Connectors dargestellt.

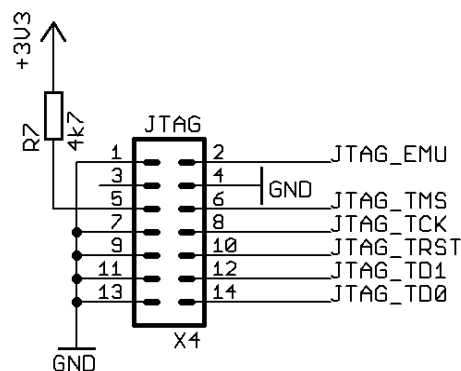


Abbildung 29: Schaltung JTAG-Connector

2.4.4 Universal Asynchronous Receiver Transmitter

Die Universal Asynchronous Receiver Transmitter-Schnittstelle (UART) dient dem seriellen Senden und Empfangen von Daten und stellt den Standard serieller Schnittstellen in PCs bzw. Mikrocontrollern dar. Mittels eines Adapters ist so eine Realisierung verschiedenster serieller Schnittstellen, wie RS232 oder USB, möglich. In Abbildung 30 ist die Beschaltung und Pinbelegung des UART-Connectors dargestellt.

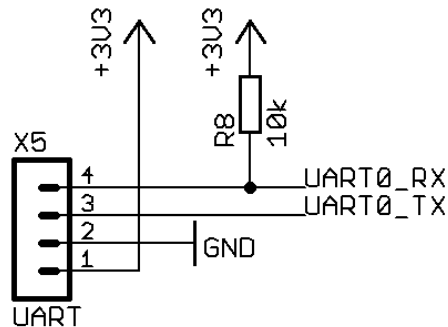


Abbildung 30: Schaltung UART-Connector

2.4.5 Analog-Digital-Konverter

Die analogen (EEG-)Signale werden über das EEG-Interface an den verwendeten Analog-Digital-Konverter ADS8345 (Schaltung siehe Abbildung 31) zugeführt und durch diesen in einen digitalen seriellen Datenstrom übersetzt. Der Konverter ist ein 16-Bit 8-Kanal Analog-Digital-Konverter (ADC) mit single supply und einem SPI-kompatiblen seriellen 4-Wire-Interface.

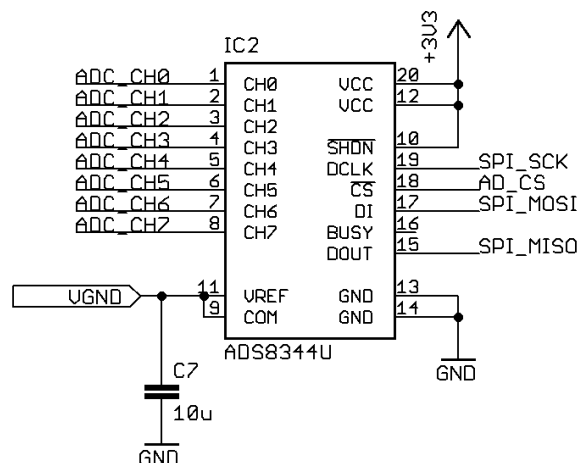


Abbildung 31: Schaltung ADC

Der ADC besitzt eine integrierte Sample-Hold Schaltung und einen On-Board Multiplexer zur Kanalwahl. Die Kommunikation mit dem DSP-Modul erfolgt mittels SPI und eine Konversion benötigt 24 Taktzyklen bzw. 3 Byte. Das erste Byte ist ein Kontroll-Byte und wird vom DSP an den ADC gesendet, adressiert durch die Chip-Select-Leitung SPISEL=1.

Prototypentwicklung eines EEG-Headsets

Der Tabelle 3 ist die Zusammensetzung des Kontroll-Bytes zu entnehmen:

BIT 7 (MSB)	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0 (LSB)
S	A2	A1	A0	—	SGL/DIF	PD1	PD0

Tabelle 3: Struktur des ADC-Kontroll-Bytes [19]

Das erste gesendete Bit ist das MSB (most significant bit) und kennzeichnet den Start (S, Startbit) der Konvertierung und muss immer 1 sein. Die folgenden 3 Bit selektieren den aktiven Kanal (A2 bis A0) und setzen den Multiplexer in Abhängigkeit von Bit 2 (single ended/ differential, SGL/DIF).

Ist das Bit 2 HIGH, dann ist der Single-Ended Mode aktiv und die Kanalwahl wird nach Tabelle 4 festgelegt:

A2	A1	A0	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7	COM
0	0	0	+IN								-IN
1	0	0		+IN							-IN
0	0	1			+IN						-IN
1	0	1				+IN					-IN
0	1	0					+IN				-IN
1	1	0						+IN			-IN
0	1	1							+IN		-IN
1	1	1								+IN	-IN

Tabelle 4: ADC Kanalwahl single ended [19]

Ist das Bit 2 LOW, dann erfolgt die Kanalwahl im Differential Mode nach Tabelle 5:

A2	A1	A0	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
0	0	0	+IN	-IN						
0	0	1			+IN	-IN				
0	1	0					+IN	-IN		
0	1	1							+IN	-IN
1	0	0	-IN	+IN						
1	0	1			-IN	+IN				
1	1	0					-IN	+IN		
1	1	1							-IN	+IN

Tabelle 5: ADC Kanalwahl differential mode [19]

Prototypentwicklung eines EEG-Headsets

Um im Akkubetrieb Energie zu sparen, ist es möglich den ADC in einen Sparmodus zu versetzen, wodurch sich sein Leistungsbedarf von ca. 5 mW auf unter 15 μ W reduzieren lässt. Mit Bit 1 und 0 wird das Verhalten des ADC zwischen den Umsetzungen entsprechend Tabelle 6 gesteuert bzw. auf interne Taktung umgeschaltet:

PD1	PD0	DESCRIPTION
0	0	Power-down between conversions. When each conversion is finished, the converter enters a low-power mode. At the start of the next conversion, the device instantly powers up to full power. There is no need for additional delays to assure full operation and the very first conversion is valid.
1	0	Selects internal clock mode.
0	1	Reserved for future use.
1	1	No power-down between conversions, device always powered. Selects external clock mode.

Tabelle 6: ADC Power Mode [19]

Nach Eingang des ersten Steuerbytes, erfolgt die AD-Konvertierung des gewählten Kanals und der digitalisierte 16 Bit breite Signalwert wird seriell während der nächsten 16 Taktzyklen zum DSP-Modul gesendet. Der umgesetzte Wert wird im binären Zweierkomplement dargestellt und liefert auf die folgenden idealen Eingangsspannungen den entsprechenden, in Tabelle 7 aufgelisteten, Outputcode:

DESCRIPTION	ANALOG VALUE	DIGITAL OUTPUT	
		BINARY TWO'S COMPLEMENT	
		BINARY CODE	HEX CODE
Full-Scale Range	$2 \cdot V_{REF}$		
Least Significant Bit (LSB)	$2 \cdot V_{REF}/65536$		
+Full-Scale	$+V_{REF} - 1\text{LSB}$	0111 1111 1111 1111	7FFF
Midscale	0V	0000 0000 0000 0000	0000
Midscale - 1LSB	$0V - 1\text{LSB}$	1111 1111 1111 1111	FFFF
-Full-Scale	$-V_{REF}$	1000 0000 0000 0000	8000

Tabelle 7: ADC Digitales Datenformat [19]

2.4.6 Driven-Right-Leg-Driver

Um die Gleichtaktspannung zu minimieren, kann der Proband mit dem logischen Nullpotential (VGND) durch eine Elektrode (Masseelektrode) verbunden werden. Diese direkte Verbindung von Proband und Messsystem mit geringer Impedanz reduziert die Gleichtaktspannung, hat aber zwei Nachteile. Zum einen kann es bei nicht-isolierten Systemen zu gefährlichen Stromflüssen über diese Elektrode kommen, zum anderen kann die Gleichtaktspannung durch eine schlechte Elektrodenimpedanz trotzdem einen signifikanten Einfluss auf das Messergebnis nehmen. Daher kommt es zur Verwendung eines im EKG-Bereich etablierten Schaltungskonzeptes, dem Driven-Right-Leg-Driver [114]. Bei EKG-Messungen wird die Masseelektrode am rechten Bein angebracht, woher auch der Name stammt. Die Idee hinter diesem Konzept ist die Erfassung der Gleichtaktspannung und die invertierte Rückführung zum Probanden, wodurch sie sich selbst eliminiert was zu einer erhebliche Verbesserung der Signalqualität führt. Das Erweitern der konventionellen Masseelektrode mit einer DRL-Schaltung reduziert die effektive Elektrodenimpedanz und erhöht die Patientensicherheit, da der zurückgeführte Strom auf ein sicheres Maß begrenzt wird.

Die Schaltung und die Bauteilwerte (Abbildung 32) sind dem OpenEEG-Projekt entnommen und wurden bereits in vorangegangenen Projekten mehrfach getestet.

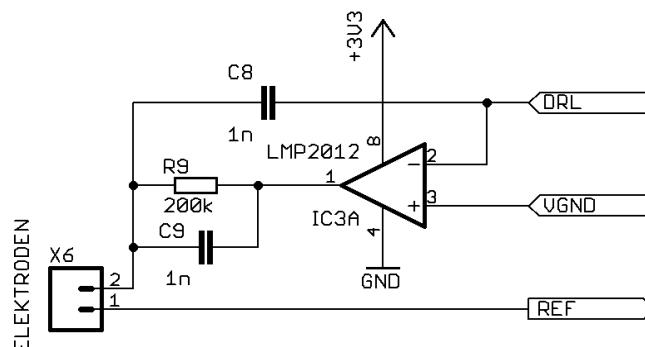


Abbildung 32: Schaltung DRL-Driver

2.4.7 EEG-Interface

Die Schnittstelle zu den 4 möglichen EEG-Verstärkerboards wird mit zwei 14-poligen Buchsenleisten bewerkstelligt. Dabei sind folgende Signale und Spannungspegel abgreifbar:

- +3,3V
- GND
- VGND
- Ref (Referenzsignal der Referenzelektrode)

Des Weiteren sind die Anschlüsse der 8 analogen Eingänge für den ADC sowie der Einspeisepunkt der DRL-Spannungen zum DRL-Driver in den beiden Buchsenleisten untergebracht. In Abbildung 33 ist die Pinbelegung und Verschaltung des EEG-Interfaces dargestellt.

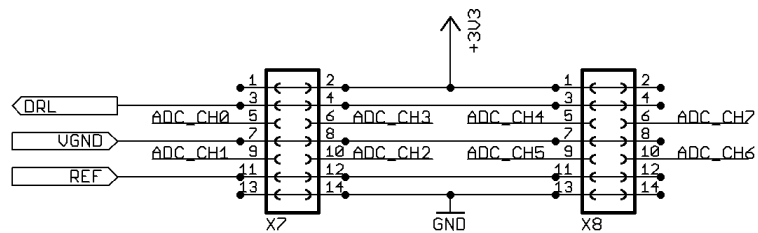


Abbildung 33: Schaltung EEG-Interface

2.4.8 Universal Serial Bus

Universal Serial Bus (USB) ist die derzeit verbreitetste serielle Schnittstelle. Das Prozessormodul besitzt einen On-Board USB 2.0 OTG Treiberbaustein, daher ist zur Implementierung einer USB-Schnittstelle nur mehr eine geringe Beschaltung (Abbildung 34) nötig. Neben der Mini-USB-Buchse komplettiert nur noch eine ESD-Protection-Diodenbaustein die Schnittstelle.

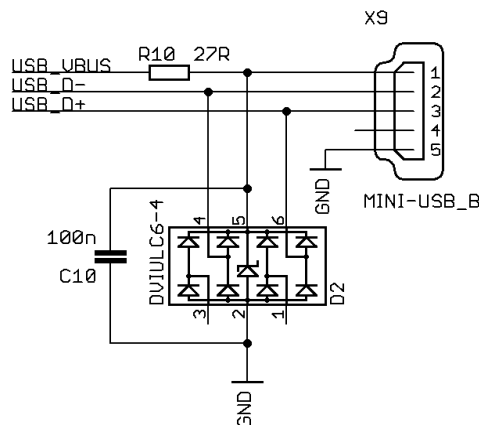


Abbildung 34: Schaltung USB

2.4.9 Parallel Peripheral Interface

Das Parallel Peripheral Interface (PPI) wird als Ausgangsinterface verwendet, um etwaige Aktoren ansteuern zu können, wie z. B. eine IR-Diode zur Realisierung einer Fernbedienung o.dgl. Dabei sind 8 Bit (PPI_D0...D7) und die Versorgungsspannung an der 2x6 poligen Buchsenleiste X10 abgreifbar und 8 Bit (PPI_D8...D15) wurden mit LED bestückt, um Signale bzw. Zustände visualisieren zu können. In Abbildung 35 ist die Beschaltung und Pinbelegung der PPI-Schnittstelle dargestellt.

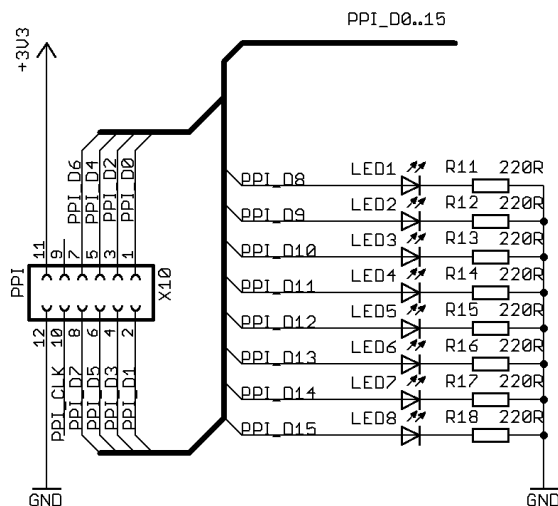


Abbildung 35: Schaltung PPI

2.4.10 SD-Card

Für die zukünftige Möglichkeit von Langzeit-EEG wurde die Anbindung von Micro-SD Speicherkarten (Schaltung siehe Abbildung 36) geschaffen. Es wurde mit dem MicroSD-Card-Connector DM3AT-SF-PEJ geplant. Zum ESD-Schutz ist ein ESD-Protection-Diodenbaustein hinzuzufügen.

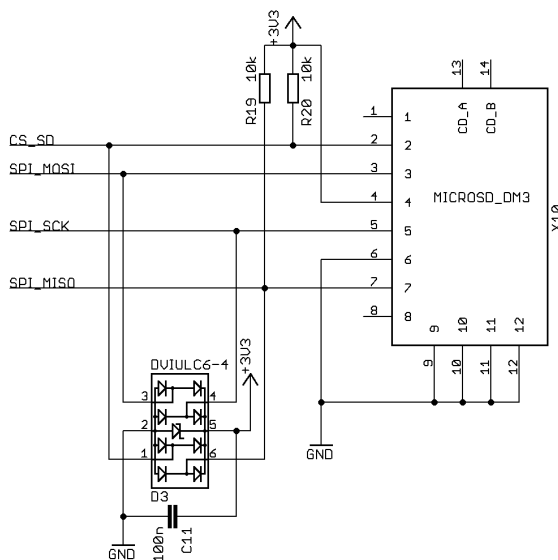


Abbildung 36: Schaltung SD-Card

2.4.11 Ethernet

Um eine zukünftige Netzwerkanbindung möglich zu machen, wurde ein Ethernet-Anschluss geplant. Zum Einsatz kommt die RJ45-Buchse mit integriertem Transformer RJLBC-060TC1 mit zusätzlichem ESD-Schutz. Die Schaltung der Ethernet-Schnittstelle ist in Abbildung 37 dargestellt:

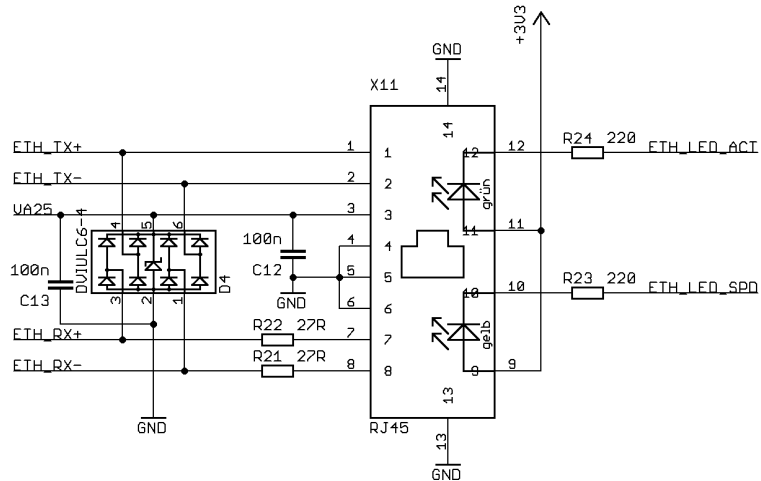


Abbildung 37: Schaltung Ethernet

2.4.12 Schaltungslayout

In Abbildung 38 ist das Routing der Motherboard-Platine dargestellt, die Ansicht von oben gefolgt von der Ansicht von unten. Die verwendeten Schnittstellen und deren Lage und Ausrichtung auf der Platine sind eingezeichnet.

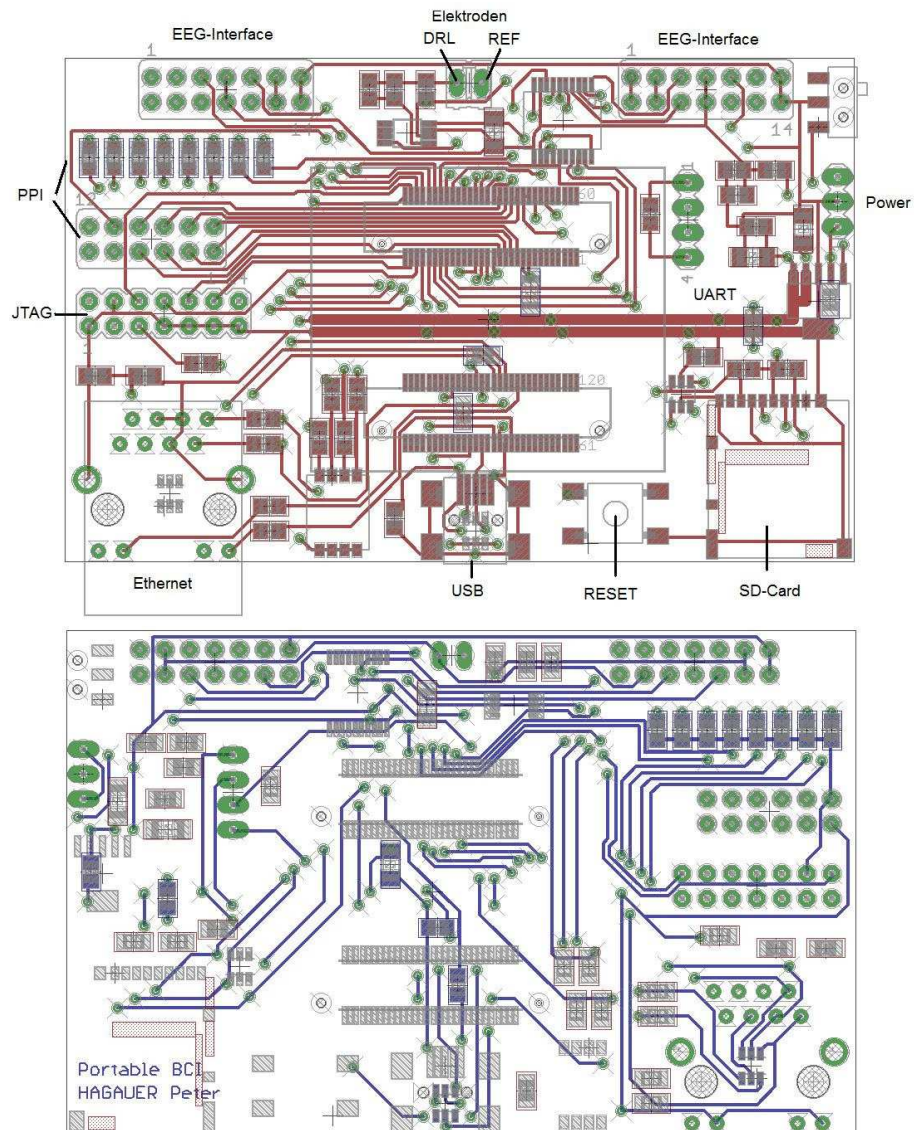


Abbildung 38: Schaltungsdesign Motherboard

2.5 USB-Adapter

Für die Kommunikation zwischen dem DSP-Modul und einem Entwicklungsrechner wird in der Regel eine serielle Schnittstelle verwendet. Da an nahezu jedem handelsüblichen Rechner USB-Schnittstellen verbaut sind, bieten sich USB Adapter an. Bereits an Blackfin-Prozessoren erprobt sind der Conrad Mini-USB zu UART-Konverter [30] und der JTAG-USB Adapter gnICE+ [1][12].

2.6 Energiespeicher

Durch den großen Eingangsspannungsbereich von bis zu 20 V des verwendeten Spannungsreglers, ist die Versorgung mit einer großen Anzahl von Energiespeichern möglich. Um eine möglichst lange Betriebsdauer zu ermöglichen, wird ein Lithium-Polymer-Akku verwendet. Sie zeichnen sich durch eine hohe Energiedichte bei geringer Größe und relativ niedrigem Gewicht aus.

Zur Verwendung kommt der LiPo-Akku „Red Power“ von Pichler, er hat eine Nennspannung von 7,4 V und eine Kapazität von 2200 mAh bei einem Gewicht von 113 g.

2.7 Trägersystem

Um sämtliche Komponenten am Kopf des Probanden befestigen zu können, wird ein größenverstellbarer Befestigungsring eines Fahrradhelmes verwendet. An diesem Ring wird eine selbst hergestellte Grundplatte aus glasfaserverstärktem Kunststoff befestigt und auf ihr die elektronischen Komponenten mit Hilfe von aufgeklebten Klettverschluss-Streifen angebracht. Die Grundplatte ist an der Rückseite fest mit dem Ring verbunden, an der Vorderseite wird die benötigte (mechanische) Spannung mit einem Gummiband gewährleistet. Um eine EEG-Messung an der Position C_z (zur Detektion von Fußbewegungsvorstellung) für verschiedene Kopfgrößen durchführen zu können, ist in der Grundplatte eine Führung für einen Elektroden-Haltungsring ausgeschnitten, um diesen Haltungsring verschieben zu können. Die benötigten Kabel für die DRL- bzw. Referenzelektrode sind für eine Montage an den preaurikulären Punkten vorbereitet. Diese beiden Elektroden können mit doppelseitigen Kleberingen befestigt und mit den vorbereiteten Kabel verbunden werden.

2.8 Gesamtsystem

In Abbildung 39 sind die wesentlichen Komponenten des EEG-Headsets dargestellt:

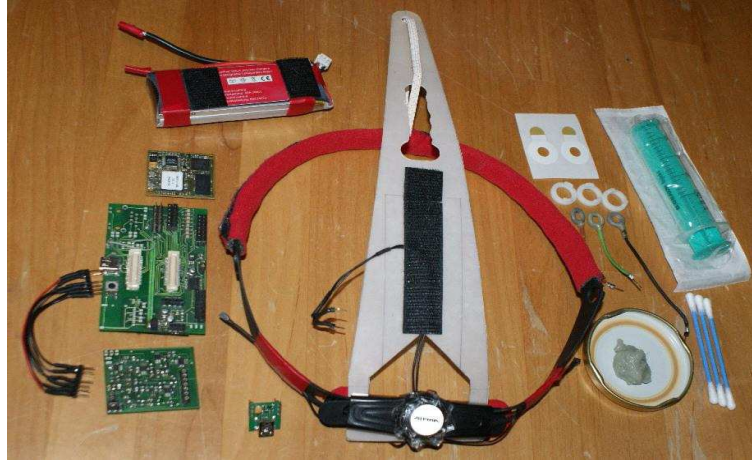


Abbildung 39: Komponenten des EEG-Headsets

In der Bildmitte ist das Trägersystem, links die elektronischen Komponenten und rechts die Elektroden und zur Montage benötigtes Zubehör dargestellt. Die elektronischen Komponenten sind, von oben nach unten: Akku, Prozessor-Modul, Motherboard, EEG-Verstärkerboard mit Anschlusskabel und UART-USB-Adapter. Nicht abgebildete ist das benötigte Impedanz-Messgerät, USB-Kabel und der verwendete Laptop.

3 Software

Zum Programmieren, Debuggen und zur Projektverwaltung kommt bei DSPs der Blackfin-Familie standardmäßig das Programm VisualDSP++ von Analog Device (AD) zum Einsatz. Da aber in allen DSP-Labors der TU-Graz einheitlich Prozessoren von Texas Instruments verwendet werden, liegen keine Lizenzen für AD-DSPs vor. Aus diesem Grund kommt ausschließlich Open-Source Software zum Einsatz.

3.1 Einführung

Jedes Programm, das von einem Prozessor ausgeführt wird, benötigt diverse Kontrollstrukturen, um einen geregelten Ablauf des Programmes gewährleisten zu können. Diese Kontrollstrukturen ermöglichen es einem Programm z.B. auf den Speicher zuzugreifen, Daten aus peripheren Geräten zu lesen bzw. zu schreiben und sie sorgen auch für einen zeitgerechten Ablauf des Programmes. Im einfachsten Fall ist dies eine Endlosschleife, in den meisten Fällen jedoch, aufgrund der immer komplexer werdenden Prozessorstrukturen, übersteigt der Entwicklungsaufwand der Kontrollstrukturen bei weitem den des eigentlichen Programmes. Bei komplexen Systemen kommt daher meist ein Betriebssystem mit integriertem Kernel zum Einsatz, der diese Kontrollstrukturen bereitstellt und somit die Softwareentwicklung wesentlich erleichtert.

Das Betriebssystem uClinux [108], eine für eingebettete Systeme optimierte Linux-Distribution basierend auf dem Linux-Kernel 2.6., unterstützt viele Prozessorarchitekturen, auf denen normalerweise kein Linux lauffähig wäre, da diese über keine Memory Management Unit (MMU) verfügen bzw. die Speicherkapazität stark eingeschränkt ist. Für die Verwendung von uClinux auf Blackfin-Prozessoren wird folgendes benötigt:

- Entwicklungsrechner mit Linux-Betriebssystem
- Blackfin GNU Toolchain
- GNU Compiler Collection (GCC)
- uClinux Distribution mit Linux Kernel
- serielles Terminal, z.B. Minicom
- U-Boot Bootloader
- Eclipse (optional)

3.2 Entwicklungsrechner

Zur Softwareentwicklung für das Blackfin-Board wird ein Entwicklungsrechner (Host) benötigt, mit dessen Hilfe Programme erstellt und gemeinsam mit dem uClinux-Betriebssystem kompiliert werden, um ein lauffähiges uClinux-Image zu erzeugen. Der Rechner benötigt eine freie serielle Schnittstelle, da er als Standard Ein-/Ausgabegerät verwendet wird.

Zur Verwendung kommt ein HP 2140 mini Netbook mit Ubuntu 10.04 LTS Lucid Lynx Betriebssystem. Folgende Softwarepakete müssen installiert sein (in Alphabetischer Reihenfolge):

- autoconf Erstellt automatisch configure-Skripte
- automake Erzeugt GNU-Standardkonforme Makefiles
- bash GNU Bourne Again Shell
- binutils GNU assambler, linker und binäre Werkzeuge
- binutils-dev binutils-Entwicklungsstools
- bison Parsergenerator
- bzip2 Komprimierungsprogramm
- coreutils GNU Bibliothek mit Befehlszeilen-Programmen
- flex Generator für lexikalische Analysen
- gawk GNU awk, Mustererkennung- bzw. Verarbeitungssprache
- gcc GNU C-Compiler
- gettext GNU Internationalisierungs-Programme
- grep sucht Text in Daten
- g++ GNU C++ Compiler
- intltool Hilfsskripte für XML-Internationalisierung
- iputils-ping Überprüfung der Erreichbarkeit von Rechnern im Netz
- libtool Allgemeines Support-Skript für Bibliotheken
- linux-libc-dev Linux-Kernel Header für die Entwicklung
- libgtk2.0-dev Entwicklungsdateien für GTK+ Bibliothek
- libglade2-dev Entwicklungsdateien für libglade
- liblzo1 Datenkompressions-Bibliothek
- liblzo-dev Entwicklungsdateien für liblzo1
- liblzo2-2 Datenkompressions-Bibliothek

- liblzo2-dev Entwicklungsdateien für liblzo2-2
- libncurses5 Bibliotheken für Terminalsteuerung
- libncurses5-dev Entwicklungsdateien für ncurses
- libreadline5 GNU readline- und History-Bibliotheken
- libreadline5-dev Entwicklungsdateien
- m4 GNU Makroprozessorsprache
- make Werkzeug zur Steuerung der Programmierung
- pax-utils Sicherheitsfokussierter ELF-Dateien-Überprüfer
- pkg-config Versionsverwaltung installierter Bibliotheken
- qt3-apps-dev Entwicklungsdateien für Qt3
- rpm Software-Paketverwaltung
- texinfo Dokumentationssystem für Online-Informationen
- zlib1g Kompressionsbibliothek
- zlib1g-dev Entwicklungsdateien

Bei der Installation dieser Pakete werden noch weitere Pakete benötigt, wie z.B. m4 benötigt libc6, diese Abhängigkeiten werden jedoch vom Ubuntu Paketverwalter (Synaptic) automatisch angezeigt und müssen zur Installation nur noch bestätigt werden.

3.3 Blackfin GNU Toolchain

Um ein in C geschriebenes Programm unter uClinux verwenden zu können, wird eine Sammlung von verschiedenen Tools benötigt. Diese sind in der Blackfin GNU Toolchain [10] zusammengefasst und müssen auf dem Entwicklungsrechner installiert werden. Es kommt die bfin-uclinux-Toolchain (Version blackfin-toolchain-2010R1-RC4) in Verwendung, die das unter Linux ausführbare ELF-Format erzeugt. Abschließend wird diese ELF-Datei mittels dem Tool elf2flt in das minimalistische FLAT-Format konvertiert, welches das Standarddatenformat von uClinux darstellt. Die wichtigsten Elemente der Toolchain sind:

1. Blackfin GCC
2. Binutils
3. C Bibliothek
4. Kernel header

3.3.1 Blackfin GCC

Die GNU Compiler Collection [51] ist eine Ansammlung von Compilern für mehrere Hochsprachen (C, C++, Fortran, Java, ...) zur Erstellung von Assembler-Code für verschiedene Prozessorarchitekturen (derzeit ca. 60) und ist der am meisten verbreitetste Compiler in der Open-Source Welt. Da sich die Prozessorarchitekturen von Entwicklungsrechner und DSP unterscheiden, muss man GCC als Cross-Compiler installieren, um Blackfin-Assembler-Code generieren zu können. Nach Installation der Toolchain findet man die benötigten Blackfin-GCC-Tools im Pfad `~/uclinux/bfin-uclinux/bin`. In der Bedienung von GCC und Blackfin-GCC [98] gibt es keinen Unterschied, sie unterscheiden sich lediglich im Aufrufnamen, so wird aus dem Befehl `gcc` um C Code zu kompilieren `bfin-uclinux-gcc` um Blackfin Code zu erzeugen. Der so gewonnene Assembler-Code wird von GCC in eine Objektdatei umgewandelt und an die Binutils zur Weiterverarbeitung übergeben.

3.3.2 Binutils

Die GNU Binary Utilities [50] sind eine Sammlung von Programmen zur Generierung bzw. Manipulation von binären Objektdateien. Die wichtigsten Haupttools sind:

- `as` GNU Assembler: wandelt den in ein Objekt verpackten Assembler-Code in die Maschinensprache um. Zur Weiterverarbeitung werden ELF-Objekte (Executable and Linkable Format) erzeugt.
- `ar` Unix-Archivierungsprogramm, das aus mehreren ELF-Objekten eine statische Archivdatei generiert, die dem Linker übergeben wird. Hier werden vom Programm aufgerufenen Bibliotheken oder andere bereits vorhandenen Programmobjekte mit eingebunden.
- `ld` GNU Linker: die in dem übergebenen Archiv vorhandenen ELF-Objekte werden statisch zu einer ELF-Datei gebunden, die das binäre Standardformat von ausführbaren Programmen unter Linux darstellt.

Mit diesen drei Tools, die von GCC aufgerufen werden, erhält man die gewünschte unter Linux ausführbare Datei. Analog zu GCC werden auch diese Tools mit dem Präfix `bfin-uclinux-` erweitert, um Blackfin spezifische Dateien erzeugen zu können.

3.3.3 Standard-C-Bibliothek uClibc

Da in C keine komplexen Funktionen definiert sind, müssen solche Funktionalitäten, wie z.B. mathematische Funktionen, Datentypen, Ein-/Ausgabefunktionen o.ä., mit Hilfe von Bibliotheken bereitgestellt werden. Die häufigsten Funktionen werden in der genormten Standard C-Bibliothek in Form von Header-Dateien bereitgestellt und dienen als Interface zwischen Programm und Betriebssystemkern. Auf jedem Unix-basiertem System ist die auf Performance und Portabilität ausgelegte GNU C-Bibliothek `glibc` zu finden, die jedoch aufgrund des Speicherplatzbedarfs nicht für den Embedded-Bereich geeignet ist. Daher kommt die konfigurierbare, kleine C-Bibliothek

uClibc [107], eine speziell für uClinux konzipierte Bibliothek, zum Einsatz. So ist es möglich, nur benötigte Funktionen „mit an Board“ zu nehmen, wodurch sich ein einfaches Hello World Programm von ca. 450 KB (mit glibc) Speicherbedarf auf unter 20 KB durch Verwendung von uClibc schrumpfen lässt. Folgende Bibliotheken werden unter anderen von uClibc im Verzeichnis `~uClinux/bfin-uclinux/bfin-uclinux/runtime/usr/lib` bereitgestellt:

- `libc.a` C Bibliothek
- `librt.a` Implementation von Echtzeit-Funktionen
- `libm.a` Mathematische Funktionen und Typen
- `libpthread.a` Thread-Bibliothek
- `libbfdsp.a` Bibliothek für Signalverarbeitung

Die in diesen Bibliotheken integrierten Funktionen (Header-Dateien) sind im Verzeichnis `~uClinux/bfin-uclinux/bfin-uclinux/runtime/usr/include` untergebracht und werden bei Bedarf vom Compiler als Präprozessor Anweisung in den Programmcode kopiert.

3.3.4 Kernel Header

In den Kernel-Headern sind alle vom Betriebssystemkern bereitgestellten Systemaufrufe, Konstantendefinitionen, Datenstrukturen und mehr definiert, die es der C-Bibliothek und kompilierten Programmen ermöglicht, mit dem Systemkernel zu interagieren. Wird die C Bibliothek bzw. eine Applikation kompiliert, die einen Systemaufruf durchführt, so wird von ihr der Kernel-Header benötigt, die in den Verzeichnissen

`~uClinux/bfin-uclinux/bfin-uclinux/runtime/usr/include/linux,`

`~uClinux/bfin-uclinux/bfin-uclinux/runtime/usr/include/asm` und

`~uClinux/bfin-uclinux/bfin-uclinux/runtime/usr/include/asm-generic`

abgelegt sind und zum Kompilationszeitpunkt mit Eingebunden werden müssen.

3.3.5 FLAT Datenformat

Um das unter uClinux verwendete FLAT Datenformat zu erhalten, wird das von der Toolchain bereitgestellte Tool `bfin-uclinux-elf2flt` verwendet und wird durch setzen der `-Wl,-elf2flt`-Flag beim Aufruf von `bfin-uclinux-gcc` involviert. Es konvertiert die vom Linker erzeugte, unter Linux lauffähige ELF-Datei in eine FLAT-Datei, da die Ausführung von ELF-Dateien auf dem Konzept der virtuellen Adressierung basiert und dieses bei Blackfin-Prozessoren ohne MMU nicht umsetzbar ist. FLAT ist ein stark reduziertes, einfaches, unter uClinux ausführbares binäres Datenformat und beinhaltet nur den Programmcode, benötigte Daten und relative Adressen zur Relokation im Speicher. Das Datenformat ist im Header `flat.h` im Verzeichnis `~uClinux/bfin-uclinux/bfin-uclinux/runtime/usr/include/linux` definiert.

3.4 uClinux

uClinux ist eine Linux-Distribution, basierend auf dem Linux-Kernel [103], die von dem uClinux distribution project [108] betreut und mehrmals im Jahr aktualisiert und erweitert wird. uClinux ist optimiert für den embedded Bereich und unterstützt viele Prozessorarchitekturen mit bzw. ohne MMU und ist hoch konfigurierbar. Die als Archiv downloadbare Distribution enthält den aktuellen Linux-Kernel, Konfigurations-Skripte, eine Vielzahl von Bibliotheken; Kernelapplikationen, Flash Tools, unterschiedliche Dateisysteme, Netzwerkdienste, Hardware-Treiber und vieles mehr. In dem bereitgestellten Archiv ist der gesamte C-Source-Code des Projektes enthalten und mittels der Toolchain kann ein lauffähiges uClinux-Image erzeugt werden, daher ist auf kompatible Versionen der Toolchain bzw. uClinux zu achten. Durch das Entpacken des Archives (blackfin-linux-dist-2010R1-RC5.tar.bz) wird das Stammverzeichnis blackfin-linux-dist erzeugt.

3.4.1 Stammverzeichnis blackfin-linux-dist

Das Stammverzeichnis umfasst über 220.000 Objekte, die in verschiedenen Verzeichnissen angeordnet sind und ca. 2,5 GB Speicherplatz benötigen. Folgende wesentliche Unterordner sind in dem Stammverzeichnis ~/blackfin-linux-dist/ zu finden:

- | | |
|-----------------|---|
| 1) linux-2.6.x/ | C-Quellcode des Linux-Kernels |
| 2) config/ | Konfigurationsprogramm von uClinux |
| 3) lib/ | Userspace-Bibliotheken |
| 4) romfs/ | Root-Filesystem |
| 5) tools/ | Zusatzprogramme zur uClinux Kompillierung |
| 6) user/ | Verfügbare Userspace-Programme |
| 7) vendors/ | Herstellerspezifische Konfigurationsskripte |
| 8) images/ | Ablage für lauffähige uClinux-Images |

3.4.2 Der Linux-Kernel

3.4.2.1 Allgemeines

Der Linux-Kernel ist das Herzstück jeder Linux-Distribution und stellt die Schnittstelle zwischen Software und Hardware dar (Hardware Abstraktion). Es werden dabei verschiedene Schichten durchlaufen, wobei die hardwarenahen (unteren) Schichten jeweils die Basis für die darüber liegenden Schichten bilden (Abbildung 40). Die Kommunikation zwischen den Schichten erfolgt durch Schnittstellen, wobei jeweils die obere Schicht eine darunter liegende aufruft.

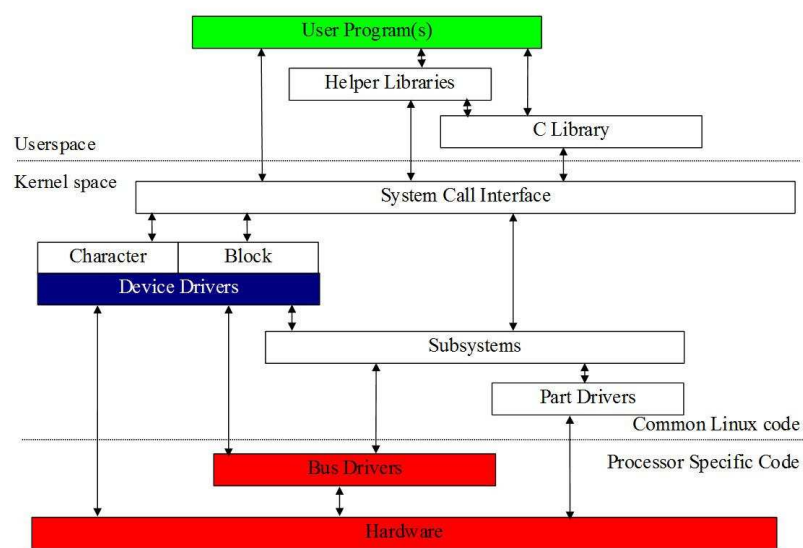


Abbildung 40: Funktion des Kernels

Möchte ein Benutzerprogramm (Userspace) auf die Hardware zugreifen, so werden die vom Kernel bereitgestellten Systemaufrufe verwendet, ohne das nähere Kenntnis der Hardware benötigt wird. Nach einem Systemaufruf wird der Userspace verlassen und der Kernelspace betreten. Mit Hilfe der im Kernel zahlreich integrierten Treiber kann auf die Hardware zugegriffen werden.

Im Wesentlichen besteht der Kernel aus folgenden Komponenten:

- Hardware-Schnittstellen
- Speicherverwaltung
- Prozessverwaltung
- Geräteverwaltung
- Dateisysteme

3.4.2.2 Das Kernel-Verzeichnis

Der C-Source Code des Kernels ist in dem Verzeichnis `~/blackfin-linux-dist/linux-2.6.x/` gespeichert und ist wie folgt organisiert:

- `kernel` Linux-Kernel Code
- `arch` Architektur-spezifischer Kernel Code
- `init` Initialisierung
- `include` Kernel-Header für Kernel-Kompilation
- `mm` Architektur-unabhängiger Memory Management Code
- `ipc` Interprozess-Kommunikation
- `driver` Alle im System verfügbaren Gerätetreiber
- `fs` Verfügbare Filesysteme

3.4.2.3 Vorteile

Auch wenn die Einrichtung eines Betriebssystemkerns umfangreich ist, so ergibt sich eine Unzahl von Vorteilen gegenüber von Bare-Metall Anwendungen, wie z.B.:

- Einfache Applikationsentwicklung

Da die, von jedem Anwenderprogramm benötigten Kontrollstrukturen bereits im Kernel integriert sind, kann man sich beim Programmieren von Anwendungen auf die verwendeten Algorithmen bzw. den Datenfluss konzentrieren, ohne sich mit näheren Hardware-Details beschäftigen zu müssen.

- Wiederverwertbarkeit von bereits geschriebenen C-Programmen

Die System-Aufrufe eines x86-Desktoprechners sind in Linux identisch mit jenen in eingebetteten Prozessoren, daher können diese leistungsfähigeren Rechner mit Standard-Entwicklungstools zur Programmentwicklung herangezogen und bereits erstellte Software mit eingebunden werden.

- Hardware-Abstraktion

Die vom Kernel repräsentierte Hardware-Abstraktion erleichtert die Applikationsentwicklung wesentlich, da das bereitgestellte Programmier-Interface Plattform-unabhängig ist. Ähnliche Geräte, wie z.B. serielle Geräte, verwenden auf allen von Linux unterstützten Prozessoren die gleichen Treiber.

- Partitionierung von Anwenderprogrammen

Sehr umfangreiche und komplexe Algorithmen können in kleine überschaubare Einheiten (Threads) aufgeteilt werden und diese können auch für weitere Projekte wiederverwendet werden, wodurch Mehrfachentwicklungen vermieden werden.

- Scheduling

Der Linux-Kernel ist ein preemptiver Multitasking-Kernel, der es ermöglicht, die Prozessorrechenzeit an unterschiedliche Prozesse gerecht zu verteilen. Der Scheduler verteilt die Rechenzeit in Abhängigkeit der vergebenen Priorität und kann laufende Prozesse auch unterbrechen, um einen mit höherer Priorität Vorrang zu geben.

- Prioritäten

Jedem Task wird eine Priorität zugeteilt, wobei der Linux-Kernel bis zu 40 Prioritätslevels verwalten kann. Die höchste Priorität wird durch den Prioritätslevel 1 repräsentiert und wird vom Scheduler bevorzugt, die niedrigste Priorität ist durch Level 40 gekennzeichnet.

- Preemption

Ein laufender Prozess kann vom Scheduler blockiert werden, wenn ein anderer mit höherer Priorität aktiv wird. Diese Vorgehensweise garantiert, dass vorrangige Aufgaben zuerst abgearbeitet werden. Der blockierte Prozess wird nach Beendigung des wichtigeren Prozesses wieder an der gestoppten Stelle fortgesetzt. Wird ein Prozess unterbrochen, so wird vom Dispatcher der Kontext des aktuellen Prozesses gespeichert, um bei der späteren Wiederbearbeitung sämtliche benötigten Informationen (Hardware-Einstellungen, Register, Stack-Pointer, Program-Counter, ...) zur Wiederherstellung des Systemzustands zum Unterbrechungzeitpunktes zur Verfügung zu haben.

3.4.2.4 Nachteile

Neben der Einarbeitungszeit gibt es leider auch Nachteile bei der Verwendung eines Betriebssystemkerns:

- Boot-Zeit

Der Linux-Kernel benötigt ca. 2-5 Sekunden für den Bootvorgang.

- Interrupt-Latenzzeit

Für einige kritische Kerneloperationen werden Interrupts deaktiviert, da diese nicht unterbrochen werden können. Dies kann zu einer verzögerten Abarbeitung eines Interrupts führen.

- Speicherbedarf

Für ein lauffähiges Linux-System wird mindestens ein Speicher von 4 bis 8 MB SDRAM bzw. 2 MB Flash benötigt.

- Stabilität

Obwohl der Kernel vielfach getestet und in unterschiedlichsten Bereichen eingesetzt wird, kann nicht ausgeschlossen werden, dass unentdeckte Fehler im System enthalten sind. Im Allgemeinen gilt Linux jedoch als sehr stabil.

3.4.3 Konfiguration

In der uClinux-Distribution sind umfangreiche Konfigurationsscripte enthalten, die es ermöglichen, uClinux bzw. den Linux-Kernel an die verwendete Hardware anzupassen. Die Konfiguration lässt sich in drei Schritte einteilen:

- Hersteller- und Produktkonfiguration
- Kernelkonfiguration
- Konfiguration der uClinux-Distribution

Das Konfigurationsmenü wird mit der Konsole vom uClinux-Stammverzeichnis aus mit dem Befehl

`$~/blackfin-linux-dist/ make xconfig` aufgerufen. In Abbildung 41 ist das Eingabefenster der Basiskonfiguration abgebildet.

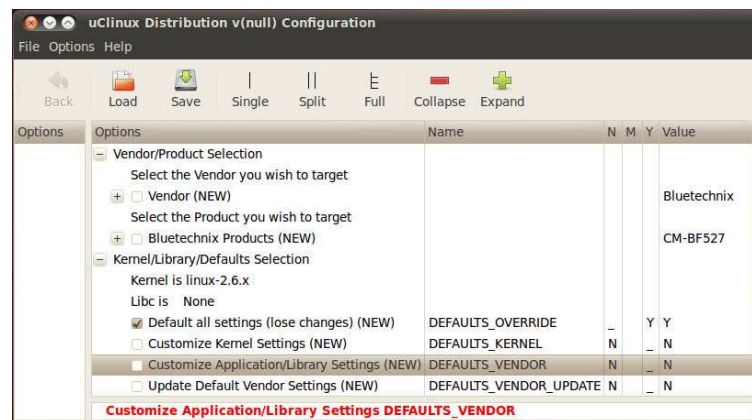


Abbildung 41: Basiskonfiguration

Im Untermenü Kernel/Library/Defaults Selection kann mit Auswahl von Default all settings (nach der Hersteller- und Produktkonfiguration) sehr bequem und schnell ein lauffähiges uClinux-Image erzeugt werden, um die generelle Funktionalität der Toolchain zu testen.

3.4.3.1 Hersteller- und Produktkonfiguration

Im Unterpunkt Vendor/Product Selection des Basismenüs sind sämtlichen unterstützten Hersteller bzw. Prozessoren aufgeführt, in diesem Fall wird Bluetechnix als Hersteller und CM-BF527 als Prozessor ausgewählt.

Die gewählte Konfiguration wird im File `~/blackfin-linux-dist/.config` gespeichert.

3.4.3.2 Kernel-Konfiguration

Durch setzen von Customize Kernel Settings in der Basiskonfiguration erreicht man das Konfigurationsmenü des Linux-Kernels (Abbildung 42).

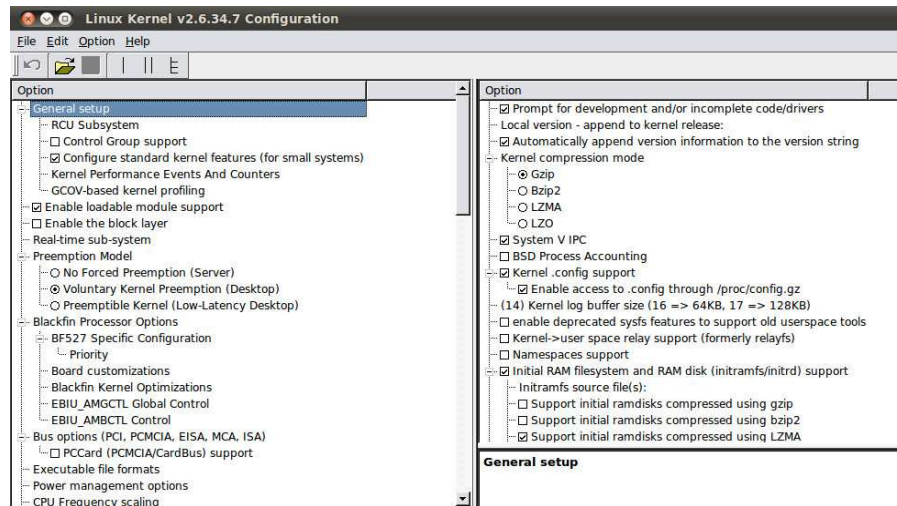


Abbildung 42: Kernel-Konfigurationsmenü

Die gewählte Konfiguration wird im File `~/blackfin-linux-dist/linux-2.6.x/.config` gespeichert und ist dem Anhang C zu entnehmen.

3.4.3.3 uClinux-Konfiguration

Durch Aktivierung des Unterpunktes Customize Application/Library Settings in der Basiskonfiguration wird die Konfiguration von uClinux (Abbildung 43) durchgeführt, um nur die benötigten Applikationen und Bibliotheken mit an Board zu nehmen.

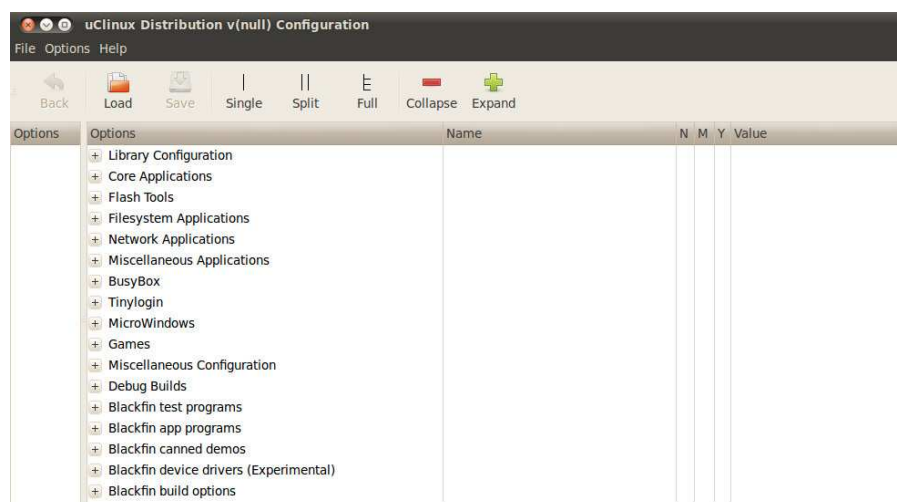


Abbildung 43: uClinux-Konfigurationsmenü

Die gewählte Konfiguration wird im File `~/blackfin-linux-dist/config/.config` gespeichert und ist dem Anhang C zu entnehmen.

3.4.4 Kompilation

Um große und umfangreiche Softwareprojekte in ein ausführbares Programm zu übersetzen, kommt bei Linux, bzw. sämtlichen Unix-Varianten, eines der wichtigsten und nützlichsten Tools zum Einsatz: *make*. Wird *make* aufgerufen, so wird die Übersetzung und Verlinkung zu einem Objekt von einer im Hauptverzeichnis befindlichen Datei, dem *makefile*, gesteuert. Das von uClinux mitgelieferte *makefile* ist ein Script, das angibt, welche Objekte in Abhängigkeit zur gewählten Konfiguration in das zu erstellende Programm einzubinden sind. Um ein lauffähiges uClinux-Image zu erhalten, wird einfach der Konsolenbefehl

```
~/blackfin-linux-dist/make
```

ausgeführt und man erhält nach einiger Kompilierzeit mehrere Objekte im Unterordner

```
~/blackfin-linux-dist/images,
```

 unter ihnen auch das vom Blackfin-Prozessor ausführbare Linux-Image *ulmage*. Wird die Konfiguration verändert, so werden zur neuerlichen Übersetzung mit *make* immer nur veränderte Objekte neu kompiliert, wodurch viel Kompilierzeit gespart wird. Dies ist auch bei hinzufügen von neuen User-Programmen der Fall.

3.5 Harte Echtzeit: Xenomai

Da Linux immer mehr Akzeptanz auch in der Automatisierungstechnik findet, gibt es seit mehreren Jahren Bestrebungen, den Linux-Kernel echtzeitfähig zu machen. Zu größerer Bekanntheit schafften es vor allem RTLinux sowie RTAI. Das neuere Xenomai-Projekt [118] war vorerst ein Entwicklungszweig von Letzterem und spaltete sich schließlich durch Unstimmigkeiten der Entwickler ab und wurde zu einem selbstständigen Projekt.

3.5.1 Prinzip

Um Linux Echtzeitfähigkeit zu verleihen, verfolgt Xenomai einen Dual-Kernel Ansatz. Dazu wird ein stark reduzierter Kernel (Microkernel) in das System integriert, der über einen eigenen Scheduler verfügt und für die Ausführung der Echtzeit-Task verantwortlich ist.

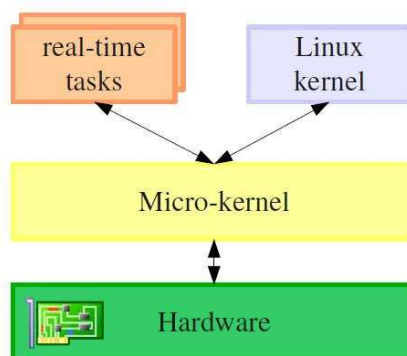


Abbildung 44: Prinzip von Xenomai

Wie in Abbildung 44 ersichtlich, wird der Linux-Kernel nun zu einem untergeordneten System, der vom Xenomai-Scheduler als Echtzeit-Task mit niedrigster Priorität behandelt wird. Der Linux-Kernel kommt immer nur dann zur Ausführung, wenn kein anderer RT-Task ausgeführt wird, kann jedoch jederzeit wieder vom Xenomai-Scheduler unterbrochen werden, um die deterministischen Anforderungen zu erfüllen.

3.5.2 Adeos/ I-Pipe

Durch die Einführung eines weiteren Betriebssystems bzw. Kernels, ergeben sich einige Probleme mit der Ressourcenverteilung und mit dem Zugriff auf diese. Xenomai verwendet hierfür Adeos (Adaptive Domain Environment for Operating Systems) [101], das zwar kein Bestandteil von Xenomai selbst ist, aber die Grundlage für dessen Echtzeitfähigkeit darstellt. Um allen beteiligten Systemen gleichberechtigten Zugriff auf die gemeinsame Hardware zu ermöglichen, übernimmt Adeos die Kontrolle über die wichtigsten HW-Befehle und virtualisiert diese für die teilnehmenden Betriebssysteme.

Die einzelnen Betriebssysteme werden in sogenannte Domänen eingekapselt, in denen sie die alleinige Kontrolle ausüben. Diese Domänen können einen eigenen privaten Adressraum und Software Abstraktionen, wie z.B. Prozesse oder File-Systeme, beinhalten, diese aber auch über Adeos mit anderen Domänen teilen. Dies geschieht mittels der Adeos Event-Pipeline, mit dessen Hilfe die verschiedenen Domänen miteinander verbunden sind und auf kritische Hardware-Ressourcen zugreifen können. In den meisten Fällen handelt es sich hierbei um Software- bzw. Hardware-Interrupts, daher auch der Namenszusatz I-Pipe (Interrupt-Pipe), die über die Pipeline die Domänen in einer durch Prioritäten vordefinierten Reihenfolge durchlaufen.

Xenomai verwendet drei Adeos-Domänen:

- Xenomai-Kernel Primäre Domäne (höchste Priorität),
- Linux-Kernel Sekundäre Domäne (niedrigste Priorität),
- Interrupt-Schild Verwaltungsdomäne, um Interrupts ev. zu sperren.

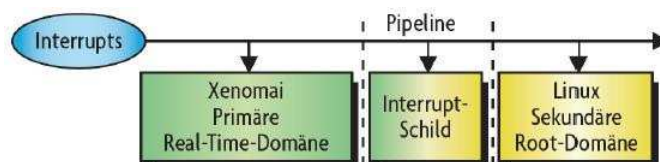


Abbildung 45: Adeos Interrupt-Pipeline

Wie aus Abbildung 45 ersichtlich, besitzt die Xenomai-Domäne die höchste Priorität und registriert daher als erstes das Auftreten eines Interrupts. Ist dieser für den Xenomai-Interrupt-Handler bestimmt, wird er dementsprechend abgearbeitet und im Anschluss entweder in der Pipeline weitergereicht oder terminiert, um auftretende Interrupts vor Linux zu verbergen. Findet der Interrupt jedoch keinen dazugehörigen Handler in der primären Domäne, so wird er in der Pipeline weitergereicht.

Um Prozesse in der sekundären Domäne, die auf keinen Fall unterbrochen werden dürfen, zu schützen, wird eine weitere Domäne zwischen der primären und sekundären Domäne eingefügt, die im Zusammenhang mit Xenomai als Interrupt-Schild bezeichnet wird. Linux kann eine globale Interrupt-Sperre anfordern, was dazu führt, dass sämtliche ankommenden Interrupts an der Sperre auflaufen und diese erst nach Freigabe der Sperre wieder in die Linux-Domäne propagieren können.

3.5.3 Struktur des Xenomai-Linux-Systems

Durch die Eingliederung des Xenomai-Microkernels und der Adeos I-Pipe kann das Gesamtsystem wie in Abbildung 46 dargestellt werden:

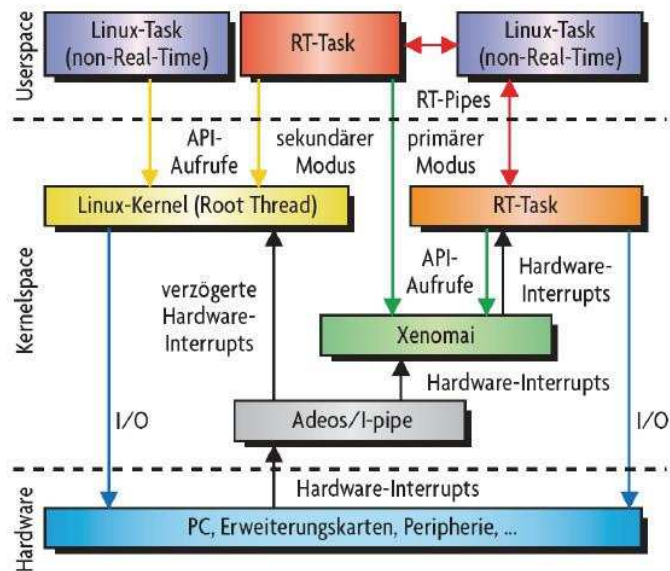


Abbildung 46: Struktur des Xenomai-Linux-Systems

In einem kombinierten Xenomai-Linux-System werden Prozesse in verschiedene Modi unterteilt, je nachdem von welchem Scheduler sie verwaltet bzw. in welcher Domäne sie ausgeführt werden. Daraus ergeben sich drei unterschiedliche Konstellationen:

- primärer Modus
Ist ein Prozess im primären Modus, so kann dieser über Xenomai auf die Hardware zugreifen, ist echtzeitfähig und erste Instanz bei Hardware-Interrupts.
- sekundärer Modus
Im sekundären Modus erfolgt der Hardware-Zugriff über den Linux-Kernel, der Prozess ist nicht echtzeitfähig und letzte Instanz bei Hardware-Interrupts.
- gemischter Modus
Xenomai ermöglicht auch, dass ein Prozess von einem in den anderen Modus übertreten kann. Dies erfolgt jeweils durch einen Systemaufruf der anderen Domäne. Kommt es bei einem Prozess im primären Modus zu einem Linux-Systemaufruf, so wird der Prozess in den sekundären Modus überführt und vice versa.

3.5.4 Systemerweiterung und Konfiguration

Die Adeos I-Pipe und Xenomai werden jeweils mittels Patch in das System eingebunden. Seit der uClinux Version 2008R1-RC8 ist Xenomai bereits inkludiert und muss in der Kernel-Konfiguration im Unterpunkt „Real-Time Subsystem“ aktiviert werden.

Mit der Aktivierung von Xenomai wird das Konfigurations-Skript

```
~/blackfin-linux-dist/user/xenomai/xenomai-2.5.3/scripts/prepare-kernel.sh
```

ausgeführt, um den Linux-Kernel für die Zusammenarbeit mit der Adeos I-Pipe vorzubereiten und den Xenomai-Kern im System zu implementieren. Das Skript `prepare-kernel.sh` wiederum führt den Blackfin-Adeos-Patch

```
~/xenomai-2.5.3/ksrc/arch/blackfin/patches/adeos-ipipe-2.6.33-blackfin-1.13-00.patch
```

aus, um das System mit der Adeos I-Pipe zu erweitern. Nach einer erneuten Kompilierung von uClinux werden die vorgenommenen Änderungen ins System übernommen.

3.5.5 Bereitgestellte Programmierschnittstellen (API)

Neben der eigenen, nativen API stellt Xenomai auch APIs von diversen anderen Echtzeit-Betriebssystemen zur Verfügung, die von Xenomai emuliert werden und die Portierung bestehender Echtzeitanwendungen ermöglicht. Die verbreitetsten sind VxWorks, RTAI, pSOS+ bzw. VRTX und sind im Unterordner `~/xenomai-2.5.3/include` abgelegt. Im weiteren Projekt wird die system-eigene API „native“ verwendet, die folgende Funktionen zur Verfügung stellt:

- Prozessmanagement
- Timing-Services
- Synchronisierungs-Unterstützung
- Interprozess-Kommunikation
- Input-/Output-Handling

Die zur Programmierung benötigten Header-Dateien befinden sich im Ordner

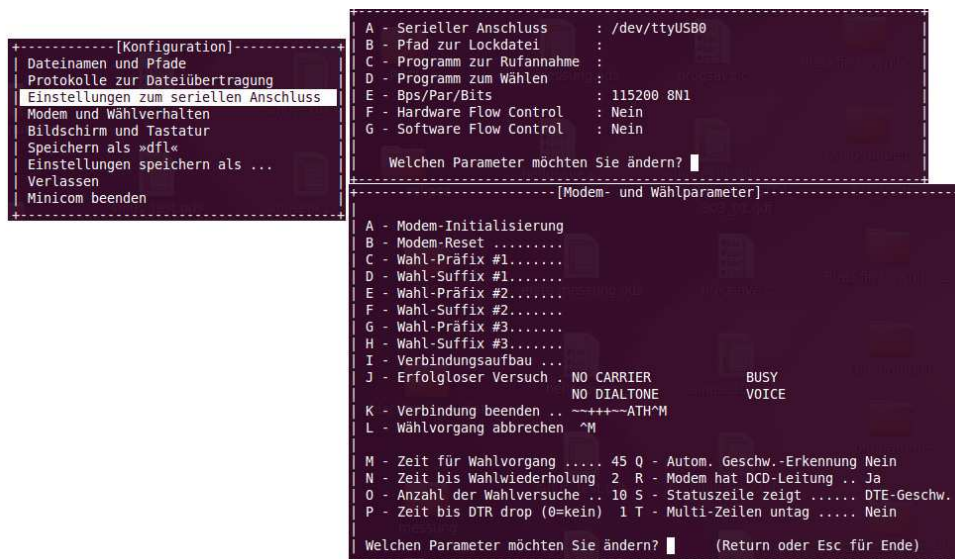
```
~/xenomai-2.5.3/include/native,
```

die ausführliche Dokumentation ist dem Ordner

```
~/xenomai-2.5.3/doc/generated/pdf zu entnehmen.
```

3.6 Minicom

Da der DSP über kein Display oder Eingabeeinheit verfügt, wird der Entwicklungsrechner als Terminal verwendet. Hierfür ist eine serielle Verbindung der UART-Schnittstelle des DSPs mit der USB-Schnittstelle des Entwicklungsrechners nötig und wird mittels UART-USB-Adapter realisiert. Als serielles Terminal zur Kommunikation zwischen dem Entwicklungsrechner und dem DSP kommt das weit verbreitete Programm Minicom zum Einsatz. Minicom ist eine textbasierte Terminalemulation, die auf nahezu jedem Linuxrechner verfügbar ist und vor allem zur seriellen bzw. Modem-Kommunikation Anwendung findet. Um mit dem DSP kommunizieren zu können, muss Minicom einmalig mit dem Konsolenbefehl `~$ minicom -s` konfiguriert werden (Abbildung 47).



```
-----[Konfiguration]-----
| Dateinamen und Pfade
| Protokolle zur Dateiübertragung
| Einstellungen zum seriellen Anschluss
| Modem und Wählverhalten
| Bildschirm und Tastatur
| Speichern als »df1«
| Einstellungen speichern als ...
| Verlassen
| Minicom beenden
+-----+

A - Serieller Anschluss      : /dev/ttyUSB0
B - Pfad zur Lockdatei      :
C - Programm zur Rufannahme :
D - Programm zum Wählen     :
E - Bps/Par/Bits            : 115200 8N1
F - Hardware Flow Control   : Nein
G - Software Flow Control    : Nein

Welchen Parameter möchten Sie ändern? █

-----[Modem- und Wählparameter]-----
A - Modem-Initialisierung
B - Modem-Reset .....
C - Wahl-Präfix #1.....
D - Wahl-Suffix #1.....
E - Wahl-Präfix #2.....
F - Wahl-Suffix #2.....
G - Wahl-Präfix #3.....
H - Wahl-Suffix #3.....
I - Verbindungsaufbau ...
J - Erfolgreicher Versuch . NO CARRIER      BUSY
                          NO DIALTONE      VOICE
K - Verbindung beenden .. ~~~~~ATH^M
L - Wählvorgang abbrechen ^M

M - Zeit für Wählvorgang .... 45 Q - Autom. Geschw.-Erkennung Nein
N - Zeit bis Wahlwiederholung 2 R - Modem hat DCD-Leitung .. Ja
O - Anzahl der Wahlversuche .. 10 S - Statuszeile zeigt ..... DTE-Geschw.
P - Zeit bis DTR drop (0=kein) 1 T - Multi-Zeilen untag ..... Nein

Welchen Parameter möchten Sie ändern? █ (Return oder Esc für Ende)
```

Abbildung 47: Minicom Konfiguration

Nach erfolgter Konfiguration wird durch den Konsolenbefehl `minicom` die serielle Schnittstelle geöffnet und die vom DSP gesendeten Daten werden im Terminal angezeigt. Um Daten vom Entwicklungsrechner zum DSP zu versenden, wird die Tastenkombination `ctrl+a` gefolgt von `s` verwendet und im Anschluss mittels Y-modem protocol die entsprechende Datei ausgewählt.

3.7 U-Boot Bootloader

Um ein Betriebssystem auf einem Prozessor laufen lassen zu können, wird ein Bootloader benötigt. Der bekannteste unter den Bootloadern ist BIOS, der auf nahezu allen x86 Architekturen zum Einsatz kommt. Im Embedded-Bereich bildet U-Boot (universal bootloader) von Denx [105] den heutigen Standard. Sämtliche von Bluetechnix ausgelieferten Core-Module werden bereits mit U-Boot ausgeliefert, daher kommt für dieses Projekt auch U-Boot zum Einsatz.

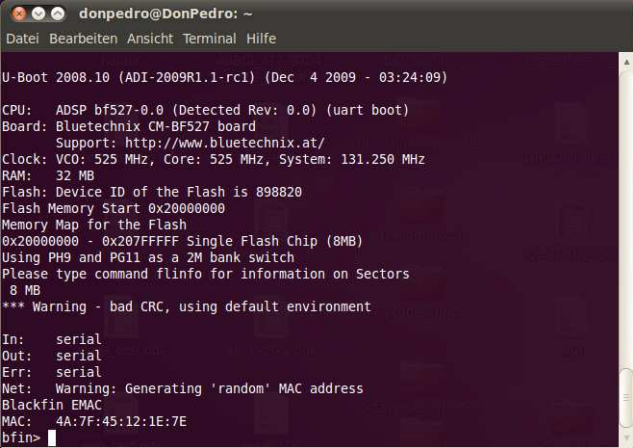
3.7.1 Aufgaben

Der Bootloader ist ein kleines binäres Programm im Flash-Speicher, dessen Hauptaufgabe es ist, das System zu initialisieren, um dem Betriebssystem bei jedem Systemstart bzw. Reset gleichbleibende Systemvoraussetzungen zu schaffen und dieses zu booten. Folgende Aufgaben werden an den Bootloader gestellt:

- Basis-Initialisierung der Hardware (Clocks, Speicher),
- Bereitstellung einer seriellen Verbindung (mittels Terminal),
- Dekomprimierung einer Applikation (z.B. das Betriebssystem),
- Laden einer Applikation und
- Ausführen dieser Applikation.

3.7.2 Funktionen

Neben den Aufgaben, die ein Bootloader zu bewerkstelligen hat, bietet U-Boot ein breites Spektrum an zusätzlichen Funktionen. Durch die Implementation der Busybox Hush Shell (Eingabeaufforderung in Abbildung 48) steht ein mächtiger Kommandozeilen-Interpreter zur Verfügung, mit dessen Hilfe sich sogar komplexe Shell-Skripts realisieren lassen. So ist das Verhalten von U-Boot mittels lokalen Shell-Variablen, globalen Umgebungsvariablen, Konditionsabfragen (if, then, else, ...), Kontrollschleifen (for, while, ...) sowie Kontrolloperatoren (UND, ODER) an individuelle Bedürfnisse anpassbar.



```
donpedro@DonPedro: ~
Datei Bearbeiten Ansicht Terminal Hilfe

U-Boot 2008.10 (ADI-2009R1.1-rc1) (Dec 4 2009 - 03:24:09)

CPU: ADSP bf527-0.0 (Detected Rev: 0.0) (uart boot)
Board: Bluetechnix CM-BF527 board
Support: http://www.bluetechnix.at/
Clock: VCO: 525 MHz, Core: 525 MHz, System: 131.250 MHz
RAM: 32 MB
Flash: Device ID of the Flash is 898820
Flash Memory Start 0x20000000
Memory Map for the Flash
0x20000000 - 0x207FFFFF Single Flash Chip (8MB)
Using PH9 and PG11 as a 2M bank switch
Please type command flinfo for information on Sectors
8 MB
*** Warning - bad CRC, using default environment

In: serial
Out: serial
Err: serial
Net: Warning: Generating 'random' MAC address
Blackfin EMAC
MAC: 4A:7F:45:12:1E:7E
bfin>
```

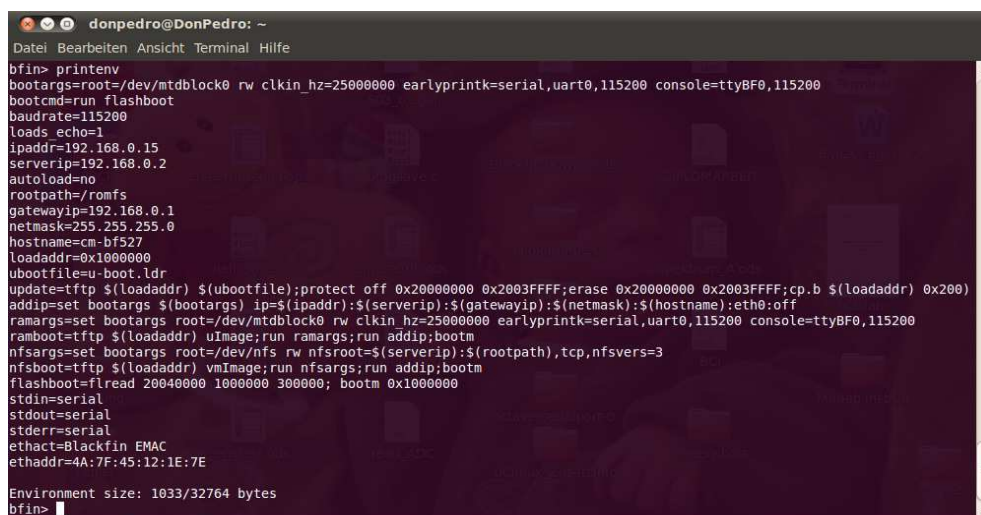
Abbildung 48: U-Boot Eingabeaufforderung

Mit Hilfe der Shell und dem umfangreichen Befehlssatz können Programme von verschiedensten Medien (serielles Port, Ethernet, USB, Flash, ...) geladen und gestartet, der Speicher manipuliert (Daten löschen, kopieren, ...) und die Hardware konfiguriert werden und vieles mehr. Der verfügbare Befehlssatz wird mit dem Konsolenbefehl *help* angezeigt und lässt sich dabei wie folgt grob unterteilen:

- Systeminformationen Anzeigen von Systemparametern
- Speichermanipulation Löschen von Speicherbereichen, kopieren von Daten,...
- Programmausführung Quellangabe und Ausführung bootfähiger Programme
- Netzwerkdienste Übertragungsprotokolle, serielle Verbindung, Ethernet, ...
- Filesystem Anzeige und Manipulation des Filesystems
- Diverses Board-Reset, Hardware diagnose, Datum, USB, SPI, ...
- Umgebungsvariablen Konfiguration von U-Boot

3.7.3 Umgebungsvariablen

Einer besonderen Bedeutung kommt den U-Boot Umgebungsvariablen zu, da mit deren Hilfe grundlegende Einstellungen des Systems vorgenommen werden können bzw. sie Parameter für Befehle in Form von globalen Variablen zur Verfügung stellen. Mit dem Befehl *printenv* (Abbildung 49) werden die verwendeten Umgebungsvariablen angezeigt.



```
donpedro@DonPedro: ~
Datei Bearbeiten Ansicht Terminal Hilfe
bfin> printenv
bootargs=root=/dev/mtdblock0 rw clk_in_hz=25000000 earlyprintk=serial,uart0,115200 console=ttyBF0,115200
bootcmd=run flashboot
baudrate=115200
loads_echo=1
ipaddr=192.168.0.15
serverip=192.168.0.2
autoload=no
rootpath=/romfs
gatewayip=192.168.0.1
netmask=255.255.255.0
hostname=cm-bf527
loadaddr=0x1000000
ubootfile=u-boot.ldr
update=tftp $(loadaddr) $(ubootfile);protect off 0x20000000 0x2003FFFF;erase 0x20000000 0x2003FFFF;cp.b $(loadaddr) 0x200
addip=set bootargs $(bootargs) ip=$(ipaddr):$(serverip):$(gatewayip):$(netmask):$(hostname):eth0:off
ramargs=set bootargs root=/dev/mtdblock0 rw clk_in_hz=25000000 earlyprintk=serial,uart0,115200 console=ttyBF0,115200
ramboot=tftp $(loadaddr) uImage;run ramargs;run addip;bootm
nfsargs=set bootargs root=/dev/nfs rw nfsroot=$(serverip):$(rootpath),tcp,nfsvers=3
nfsboot=tftp $(loadaddr) vmlinux;run nfsargs;run addip;bootm
flashboot=flashread 20040000 1000000 300000; bootm 0x1000000
stdin=serial
stdout=serial
stderr=serial
ethact=Blackfin EMAC
ethaddr=4A:7F:45:12:1E:7E
Environment size: 1033/32764 bytes
bfin>
```

Abbildung 49: U-Boot Umgebungsvariablen

Mit dem Kommandozeilenbefehl *setenv* können die Umgebungsvariablen verändert und anschließend mit *saveenv* gespeichert werden, um die verfügbaren Schnittstellen (Seriell, Ethernet, ...) zu konfigurieren. Folgende Variablen sind für dieses Projekt von essentieller Bedeutung:

- *stdin* Standardmäßige Eingabequelle
- *stdout* Standardmäßige Ausgabe
- *stderr* Standardmäßige Fehlerausgabe
- *baudrate* Übertragungsrate
- *bootargs* Übergabeparameter an den Linuxkernel

3.7.4 Laden und booten von uClinux

Der DSP wird mittels der UART-Schnittstelle über einen UART-USB-Adapter und einem USB-Kabel mit dem Entwicklungsrechner verbunden und das Programm Minicom gestartet. Durch Anlegen der Betriebsspannung bzw. durch Drücken der Reset-Taste erscheint im Terminal die U-Boot-Eingabeaufforderung. Mit dem U-Boot-Befehl *loady* wird die Empfangsbereitschaft im y-mode hergestellt. Mit der Minicom-Tastenkombination *ctrl+a* gefolgt von *s* (Abk. für send) kann nun nach Auswahl von *ymodem* im Minicom-Dialogfenster das Linux-Image `~/blackfin-linux-dist/images/ulmage` zum Versenden ausgewählt werden. Nach der erfolgreichen Übertragung kann das Linux-Image mit dem Befehl *iminfo* (Abbildung 50) überprüft werden:

```
U-Boot 2008.10 (ADI-2009R1.1-rc1) (Dec 4 2009 - 03:24:09)
bfin> loady
## Ready for binary (ymodem) download to 0x01000000 at 115200 bps...
C## Total Size      = 0x001edd89 = 2022793 Bytes
bfin> iminfo

## Checking Image at 01000000 ...
Legacy image found
Image Name:   bf527-2.6.34.7-ADI-2010R1
Created:     2011-12-03  0:26:04 UTC
Image Type:  Blackfin Linux Kernel Image (gzip compressed)
Data Size:   2022729 Bytes =  1.9 MB
Load Address: 00001000
Entry Point: 001295f0
Verifying Checksum ... OK
bfin> |
```

Abbildung 50: U-Boot Informationen über geladene uClinux-Images

Im Anschluss kann Linux mit dem Befehl *bootm* gestartet werden und im Terminal steht nach dem Bootvorgang die gewohnte Linux-Eingabeaufforderung zur Verfügung.

3.8 Eclipse

Um ein Benutzerprogramm für uClinux zu schreiben, ist grundsätzlich kein zusätzliches Programm nötig, man benötigt lediglich einen Texteditor um den Programmcode zu schreiben. Wird diese Textdatei mit der Endung `.c` versehen, kann mittels GCC ein lauffähiges Programm kompiliert werden.

Für umfangreichere Projekte empfiehlt sich jedoch eine grafische Programmieroberfläche, um die Übersichtlichkeit zu erhöhen. Eine weit verbreitete und für viele Programmiersprachen geeignete, da erweiterbare Entwicklungsumgebung, ist Eclipse [102], eine auf Java basierte open-source IDE (integrated development environment) zur Projektverwaltung, Quellcodeverfassung, Kompilierung von Programmen und vieles mehr.

Zum Einsatz kommt Eclipse Helios Service Release 2 mit der Eclipse-Erweiterung CDT 7.0.2. (Eclipse C/C++ Development Tools) zur C-Programmierung. Um Blackfin-spezifischen Code erzeugen zu können, wird des weiteren das Blackfin-Plugin [9] benötigt (Version 1.0.3.).

3.9 Hello World

Um ein Benutzerprogramm unter uClinux laufen zu lassen, sind einige Schritte notwendig, die an dem Programm „Hello World!“ veranschaulicht werden sollen. Als erster Schritt muss der Programmcode mit einem Texteditor bzw. Eclipse verfasst werden und mit der Endung `.c` abgespeichert werden. Folgender Programmcode wird unter dem Namen *hello.c* abgespeichert:

```
#include <stdio.h>

int main()
{
    ("Hello, World!\n");
    return 0;
}
```

Das C-Programm wird nun mit Hilfe der Toolchain und GCC (bzw. Eclipse) kompiliert:

```
bfm-uclinux-gcc -Wl,-elf2flt hello.c -o hello.
```

Das erzeugte Blackfin FLAT-Programm *hello* wird nun in den Ordner

```
~/blackfin-linux-dist/romfs/bin
```

kopiert und ist nach einer erneuten Kompilierung von uClinux mit `make` im System integriert und kann nach der Übertragung am Board Linux-üblich ausgeführt werden.

4 Portables Brain-Computer Interface

Im Rahmen dieses Kapitels soll die Möglichkeit einer Implementierung eines portablen BCI-Systems mit dem EEG-Headset-Prototypen aufgezeigt werden.

Um portable BCI-Systeme realisieren zu können, muss aufgrund der begrenzten Platzressourcen und zur leichteren Handhabung die Anzahl der verwendeten EEG-Kanäle reduziert werden. Des Weiteren ist es sinnvoll, die notwendige Trainingszeit auf ein Minimum zu beschränken, um die Verwendung von BCIs auch von nicht eingeschränkten Personen zu forcieren. Um dies gewährleisten zu können, wird ein stabiles Gehirnmuster zur Klassifikation benötigt, das auch ohne Training des Anwenders beständig auftritt. Eine Möglichkeit dies zu erreichen ist die Verwendung des Beta-Rebounds [84].

Durch Detektion des Beta-Rebounds wird ein gedankengesteuerter Taster (Brain-Switch) realisiert, der mit Fußbewegungsvorstellung betätigt wird [71]. In den folgenden Kapiteln wird die Vorgehensweise und die entscheidenden Schritte der Umsetzung des Brain-Switchs erläutert. Der vollständige Source-Code der benötigten Programme ist dem Anhang D zu entnehmen.

4.1 Vorgehensweise

Bei einer freiwillig durchgeführten Fußbewegung (z.B. einer Dorsiflexion), als auch bei einer passiven, einer vorgestellten oder auch durch Elektrostimulation hervorgerufenen Fußbewegung, kommt es an der mit Fußbewegung korrespondierenden Elektrodenposition C_z kurz vor und während der Bewegung zu einer ereignisbezogenen Desynchronisation (ERD) gefolgt von einer ereignisbezogenen Synchronisation (ERS) im Beta-Band (13-35 Hz) [97]. Diese ERS tritt ca. 1s nach Beendigung der Bewegung auf und kann große Leistungszunahmen im Beta-Band von mehreren 100% mit sich bringen. Diese großen Leistungssprünge im Beta-Bereich werden als Beta-Rebound bezeichnet.

Durch Ableitung des EEGs an der Elektrodenposition C_z kann der Beta-Rebound detektiert und ein Brain-Switch mit nur einem EEG-Kanal realisiert werden. Um das hierfür benötigte Leistungsspektrum zu erhalten, wird das Spektrum des eingelesenen EEGs mit Hilfe der Fouriertransformation ermittelt und anschließend der Betrag quadriert (Parseval'sches Theorem).

Als Klassifikationsparameter kommt die spektrale Leistung einer anwenderspezifischen Frequenz zum Einsatz, die mit einem Trainingsparadigma ermittelt wird.

Abschließend wird die spektrale Leistung der ermittelten Frequenz mit einem Schwellwert verglichen und im Falle einer Überschreitung als Beta-Rebound identifiziert, wodurch ein Steuerungsbefehl ausgelöst wird.

4.2 Signalgewinnung

Um einen digitalisierten Wert des ADC einlesen zu können, muss der SPI-Bus initialisiert und geöffnet werden. Der ADC ist an der SPI-Select-Leitung 1 angeschlossen und wird mit einer Präprozessoranweisung definiert als:

```
#define ADC "/dev/spidev0.1"
```

Anschließend wird der SPI-Bus mit Hilfe des als globalen integer-Variablen definierten File-Deskriptors `fd` mit Schreib- und Leserechten geöffnet:

```
fd = open(ADC, O_RDWR);
```

Der Variable `fd` wird eine negative Zahl zugewiesen, sollte ein Öffnen nicht möglich sein, bei gelungener Öffnung wird `fd` eine Null zugewiesen. Dies ermöglicht eine Fehlerüberprüfung und eine gegebenenfalls Beendigung des Programmes:

```
if (fd < 0)
{
    printf("Cannot open ADC- Port");
    exit (1);
}

printf ("spidev0.1 opened\n");
```

Nach erfolgreicher Öffnung des SPI-Kanals muss an den ADC das Control-Byte gesendet werden, um den verwendeten ADC-Kanal zu wählen. Um eine Umsetzung des ADC-Kanals 0 im Single-Ended Mode ohne zwischenzeitliche Abschaltung zu erreichen, muss die binäre Folge 1000 0111 (0x87 in hexadezimalen Darstellung) an den ADC gesendet werden. Dazu wird eine unsigned 8-bit-Integer (bzw. unsigned char) Variable `tx` definiert und initialisiert:

```
uint8_t tx = 0x87;
```

Die Variable `tx` wird anschließend an den ADC geschickt, um eine Umsetzung zu starten:

```
write(fd,tx,1);
```

Nach kompletter Übertragung sendet der ADC den erfassten digitalen Wert in der Zweierkomplement-Darstellung. Prozessoren der Blackfin-Familie sind für Festkomma-Arithmetik optimiert, daher wird dieser Wert als `fract16`-Datentyp [4] in den Puffer `buf` eingelesen. Der Datentyp wird von der Toolchain bereitgestellt und stellt die digitale Zahl in der Zweierkomplement-Darstellung im Wertebereich von -1 bis 1 dar.

```
#include <fract.h>

fract16 buf[2]={0};

status = read(fd, buf, 2);
```

Mit Hilfe der als lokal definierten Integer-Variable `status` kann analog zu `fd` eine korrekte Datenübertragung überprüft werden.

Um Daten mit einer definierten Abtastrate zu erhalten werden von Xenomai bereitgestellte Funktionen verwendet. Die verwendeten Header-Dateien sind als Präprozessoranweisungen zu deklarieren:

```
#include <native/task.h>
#include <native/timer.h>
```

Das Einlesen der Daten wird in einen Realtime-Task ausgelagert, der global mit

```
RT_TASK demo_task;
```

definiert, von main() mit

```
rt_task_create(&demo_task, "trivial", 0, 99, 0);
```

initialisiert und mittels

```
rt_task_start(&demo_task, &demo, NULL);
```

gestartet wird. Wird die Hauptfunktion main() beendet, so wird auch der Realtime-Task beendet. Daher wird main() mit einer Aufforderung zu einer Tastatureingabe angehalten, während der Realtime-Task läuft. So kann die Programmausführung jederzeit mit einer beliebigen Taste beendet und der RT-Task wieder gelöscht werden:

```
char taste = getchar();
rt_task_delete(&demo_task);
return 0;
```

Innerhalb des RT-Tasks demo() können lokal benötigte Variablen definiert werden. Mit der Funktion

```
rt_task_set_periodic(NULL, TM_NOW, 3906250);
```

wird der RT-Task periodisch gemacht, wobei die Angabe der Periodendauer in ns erfolgt. Innerhalb einer mit while realisierten Endlosschleife wird nun von dieser Funktion periodisch die Funktion

```
rt_task_wait_period(NULL);
```

aufgerufen, die sich immer solange in Warteposition befindet, bis die angegebene Periodendauer abgelaufen ist, um im Anschluss einen Wert vom ADC anzufordern und einzulesen.

Der vollständige C-Source-Code zum periodischen Einlesen des EEGs ist in dem Beispiel-Programm read_ADC enthalten und ist dem Anhang D zu entnehmen. Bei der Kompilation des Programmes müssen die unter

```
~/blackfin-linux-dist/staging/usr/xenomai/lib
```

abgelegten Bibliotheken libxenomai.a und libnative.a verlinkt werden.

Dieses Programm liest das EEG mit einer Abtastrate von 256 Hz ein und zeigt die erhaltenen Werte am Bildschirm an.

4.3 Spektrale Bandleistung

Das C-Programm `read_ADC` enthält neben dem periodischen Einlesen der EEG-Daten auch den Code zur Berechnung der spektralen Bandleistung. Nach jedem eingelesenen Wert wird mit Hilfe der Fouriertransformation das Spektrum der letzten Sekunde ermittelt, um im Anschluss durch Quadratur des Betragsspektrums das Leistungsspektrum zu errechnen. Sämtliche benötigten Schritte, um das Leistungsspektrum zu erhalten, sind mit einem Kommentar „FFT“ im Quell-Code gekennzeichnet.

Zur Berechnung der spektralen Bandleistung wird die von der Toolchain bereitgestellte Signalverarbeitungsbibliothek `libbfdsp.a` verwendet. Diese Bibliothek ist eine Portierung der VisualDSP-Bibliothek `libdsp` von Analog Device und bietet den vollen Funktionsumfang [4].

Als erster Schritt wird das Spektrum der eingelesenen EEG-Daten mit Hilfe der schnellen Fouriertransformation (FFT) ermittelt. In Verwendung kommt der Radix-2 FFT Algorithmus für reelle `fract16`-Eingangsdaten `rfft_fr16`, der in der Header-Datei `filter.h` definiert ist. Da das Spektrum komplex ist, wird auch der komplexe Datentyp `complex_fract` benötigt:

```
#include <fract_complex.h>
#include <filter.h>
```

Die Länge des Eingangsdaten-Arrays, das der FFT-Algorithmus benötigt, muss eine Potenz von 2 und mindestens 8 sein. Um eine Frequenzauflösung des Spektrums von 1 Hz zu erhalten, wird die Länge der Eingangssequenz mit 256 bei einer Abtastrate von 256 Hz gewählt:

```
#define SAMPLES 256
```

Der Algorithmus benötigt neben dem Eingangsdaten-Array `input` auch noch ein Ausgangsdaten-Array `output`, in dem das errechnete Spektrum abgespeichert wird, sowie ein Array halber Länge für die Twiddle-Faktoren, die einmalig zu Beginn in `main()` mit Hilfe der Funktion `twidfftrad2_fr16` berechnet werden:

```
fract16 input[SAMPLES]={0};
complex_fract16 output[SAMPLES]={0};
complex_fract16 twiddle[SAMPLES/2]={0};
twidfftrad2_fr16(twiddle, SAMPLES);
```

Nachdem ein Wert vom ADC in den Puffer `buf` eingelesen wurde, wird jede Stelle des Eingangsdaten-Arrays um eine Stelle nach rechts verschoben und der Wert des Puffers an die nullte Stelle des Arrays kopiert. So wird gewährleistet, dass sich die jeweils letzten 256 Werte im Eingangsdaten-Array befinden.

```
for ( t = SAMPLES; t > 0; input[t] = input[t-1], t--);
input[0] = buf[0];
```


Im Anschluss wird das Spektrum mit Hilfe der FFT ermittelt:

```
rfft_fr16(input, output, twiddle,1,SAMPLES, &blockexp,0);
```

Um aus dem in output gespeicherten Spektrum das Leistungsspektrum psd zu berechnen, werden die einzelnen absoluten Werte quadriert. Für mathematische Manipulationen können bei Verwendung des Datentyps fract nicht die Standard-Mathematik-Operationen (+, -, /, ...) verwendet werden, sondern es muss auf eigens für fract definierte Funktionen zurückgegriffen werden. Diese Funktionen sind in fract_math.h definiert, math_bf.h wird zusätzlich benötigt:

```
#include <fract_math.h>
#include <math_bf.h>
fract32 psd[SAMPLES]={0};
for (t=0; t<SAMPLES; t++)
{
    psd[t]=mult_fr1x32(cabs_fr16(output[t]),cabs_fr16(output[t]));
}
```

Wenn das Programm *read_ADC* mit einem Tastendruck beendet wird, so wird das Leistungsspektrum der letzten Sekunde am Bildschirm ausgegeben.

Die Spektren werden aus den vom ADC eingelesenen Werten ermittelt und werden nicht in Spannungswerte umgerechnet, um Rechenzeit zu sparen. Daher entsprechen die erhaltenen Werte der Spektren nicht den tatsächlichen Leistungswerten, sind aber direkt proportional zu diesen.

Bei der Kompilation sind zusätzlich die von der Toolchain bereitgestellten Bibliotheken *libbfdsp.a* sowie *libm.a* zu verlinken.

4.4 Brain-Switch

Der Brain-Switch wird mit zwei Programmen realisiert: *BCI_training* und *BCI*.

Mit Hilfe des Programmes *BCI_training* wird das anwenderspezifische Frequenzband ermittelt, in der es zu einem Beta-Rebound bei Fußbewegung bzw. der Vorstellung dieser Fußbewegung kommt, um anschließend in diesem Frequenzband die Klassifikation mit dem Programm *BCI* mittels Schwellwertverfahren vorzunehmen.

Mit dem in *BCI_training* definierten Trainingsparadigma (siehe Abbildung 51) wird das Leistungsspektrum eines Referenzzeitraumes ohne Bewegung mit einem Zeitintervall nach der Bewegung verglichen. Das EEG wird dabei in 8 Sekunden andauernde Trials eingeteilt und die über mehrere Trials gemittelten Leistungsspektren im Referenzintervall bzw. im Vergleichsintervall ermittelt. Die Anzahl der Trials wird als Übergabeparameter beim Programm-Aufruf definiert, z.B. *BCI_training 20*.

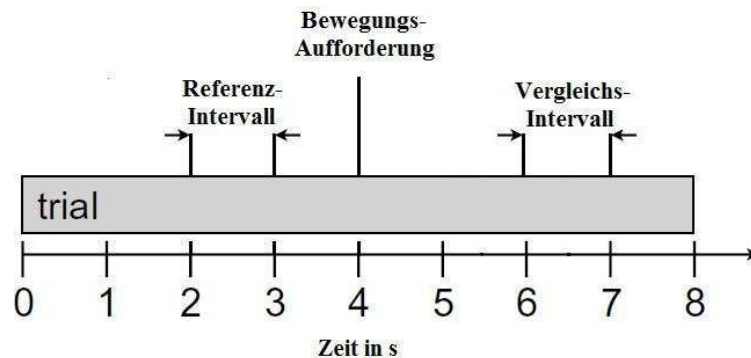


Abbildung 51: Timing des Trainingsparadimas

Zu Beginn jedes Trials wird dem Anwender ein leerer Bildschirm präsentiert. Das analog zu *read_ADC* errechnete Leistungsspektrum *psd* wird nach 3 Sekunden mit dem Referenzspektrum *specref* gemittelt. Zur Sekunde 4 des Trials wird der Anwender durch Einblenden eines Quadrates zu einer kurzen, einmaligen Dorsiflexion bzw. der gedanklichen Durchführung dieser Bewegung, aufgefordert. Diese Bewegung sollte nicht länger als 1s in Anspruch nehmen. Da der Beta-Rebound mit einer Latenz von ca. 1s nach der Bewegung auftritt, wird das Leistungsspektrum *psd*, unter Berücksichtigung von einer halben Sekunde Reaktionszeit auf die Bewegungsaufforderung, zur Sekunde 7 ermittelt und mit dem Vergleichsspektrum *specaktiv* gemittelt. Im Anschluss wird der Bildschirm wieder gelöscht.

Zur Speicherung der gemittelten Leistungsspektren des Referenz- und Vergleichsintervalls werden Arrays des Datentyps *fract32* definiert:

```
fract32 specref[SAMPLES]={0};
```

```
fract32 specaktiv[SAMPLES]={0};
```

Das Timing im Programm *BCI_training* wird durch die Zählvariable *count* sichergestellt, die nach jedem eingelesenen Wert inkrementiert wird und mit Hilfe von *if*-Abfragen den Ablauf des Trainingsparadigmas steuert. Erreicht die Zählvariable *count* den Wert 768 (entspricht der Sekunde 3) wird das gemittelte Referenzspektrum ermittelt. Bei der Mittelwertbildung von *fract*-Datentypen ist darauf zu achten, dass die zu mittelnden Zahlen zuerst durch 2 zu dividieren sind und die Addition erst im Anschluss erfolgt, um einen Datenüberlauf zu vermeiden. Die Division mit 2 erfolgt mit einer logischen Rechtsverschiebung um eine Stelle mit der Funktion *shr_fr1x32*, die Addition wird mit der Funktion *add_fr1x32* durchgeführt.

Im ersten Trial soll jedoch keine Division durchgeführt werden, sondern das Leistungsspektrum *psd* direkt in das Array *specref* kopiert werden. Zu diesem Zwecke wird eine *integer*-Variable *first* mit 0 initialisiert und als Angabe der zu verschiebenden Stellen verwendet. Nach Beendigung des ersten Trials wird diese Variable auf 1 gesetzt, so dass bei allen folgenden Trials die Division mit 2 zur Mittelwertbildung durchgeführt wird.

```
for (t=0; t<SAMPLES; t++)
{
    specref[t]=add_fr1x32(shr_fr1x32(psd[t],first),
                        shr_fr1x32(specref[t],first));
}
```

Bei einem Zählerstand von *count* = 1792, was Sekunde 7 entspricht, wird das Leistungsspektrum des Vergleichsintervalls analog berechnet.

```
for (t=0; t<SAMPLES; t++)
{
    specaktiv[t] = add_fr1x32(shr_fr1x32(psd[t],first),
                            shr_fr1x32(specaktiv[t],first)),t++);
}
```

Nachdem die angegebenen Trials absolviert sind, werden die beiden Spektren *specref* und *specaktiv* am Bildschirm ausgegeben. Wurde ein Beta-Rebound detektiert, so sind spektrale Leistungen von *specaktiv* im Beta-Bereich größer als im Referenzspektrum *specref*. Jene Frequenz, in der die Leistung von *specaktiv* im Vergleich mit *specref* am größten ist, wird zur Klassifikation mit dem Programm BCI herangezogen.

Beim Programmaufruf von BCI wird neben der anwenderspezifische Frequenz auch der Schwellwert mit übergeben, z.B. BCI 24 2000. In diesem Beispiel wird die spektrale Leistung der Frequenz 24 Hz des Leistungsspektrums *psd* nach jedem eingelesenen Wert mittels *if*-Abfrage mit dem Schwellwert 2000 verglichen und bei einer Überschreitung als Beta-Rebound identifiziert. Die nächste Detektion erfolgt erst nach einer Wartezeit von einer Sekunde, um eine mehrfach-Erfassung eines Beta-Rebounds zu vermeiden.

5 Ergebnisse

Im Rahmen dieses Projektes wurde ein portables EEG-Headset realisiert, dessen Hauptkomponenten ein EEG-Verstärkerboard und ein Motherboard zur Aufnahme des DSP-Modules CM-BF527 sind. Durch Verwendung von SMD-Technik konnten die Platinen miniaturisiert werden, um sie auf dem Kopf mittels eigens entwickelten Trägersystem tragbar zu machen. Abbildung 52 zeigt das am Kopf montierte EEG-Headset.

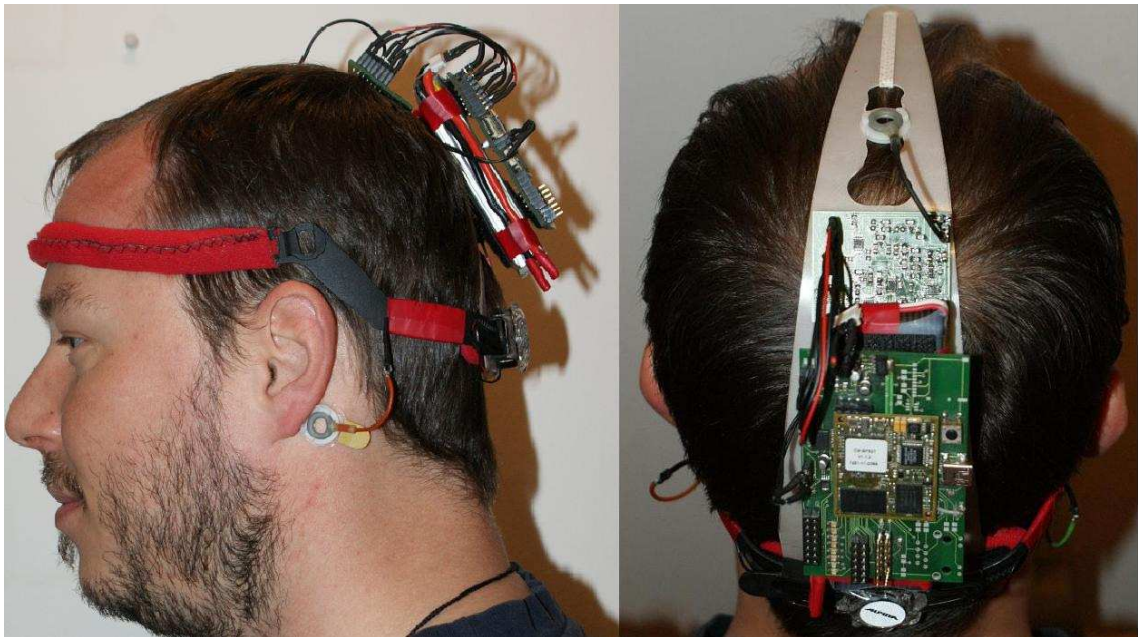


Abbildung 52: EEG-Headset: Ansicht von der Seite und von Oben

Das Verstärkerboard hat die Abmessungen 43 x 35 mm und bietet Platz für zwei Kanäle, wobei einer bestückt und getestet wurde. Das Motherboard misst 80 x 51 mm und bietet neben der Steckverbindung zu dem DSP-Modul noch eine Reihe von verschiedenen Schnittstellen. Das gesamte Headset wiegt nur 200 g, wobei durch den Akku bereits 116 g in Anspruch genommen werden. Das grössenverstellbare Trägersystem zur Kopfmontage der einzelnen Komponenten wurde aus Kunststoff gefertigt und wurde auf die Messung eines EEG-Kanals an der Position C_z ausgelegt.

Prototypentwicklung eines EEG-Headsets

Das EEG-Headset hat folgende, in Tabelle 8 zusammengefassten technische Spezifikationen:

Prototyp EEG-Headset		
externe Spannungsversorgung DC	min. 3,6 max. 20	V
interne Spannungsversorgung DC	3,3 single supply	V
Max. Anzahl der Kanäle	8	
Auflösung	16	Bit
Abtastrate	max. 12,5	kHz/Kanal
Eingangsspannungsbereich	790	μ V
1 LSB	12	nV
Verstärkung	4171	
CMMR	130	dB
Noise	< 2	μ Vpp
Hochpassfilter -3dB	0,159	Hz
Tiefpassfilter -3dB	48,23	Hz
Schnittstellen	USB 2.0 UART PPI Ethernet (opt.) SD-Card (opt.)	
Signalverarbeitungseinheit	CM-BF527	
	600	MHz
Betriebssystem	uClinux	
Kernel	linux-2.6	
Echtzeitkernel	xenomai-2.5.3	

Tabelle 8: Datenblatt EEG-Headset

Die Abbildungen 53 bis 68 zeigen mit dem EEG-Headset aufgezeichnete Spannungsverläufe sowie Leistungsspektren im Leerlauf, Kurzschluss und bei verschiedenen Testspannungen, die mit Hilfe eines Signalgenerators erzeugt wurden. Der Signalgenerator erzeugt sinusförmige Testspannungen mit einer Frequenz von 10 Hz bei einstellbarer Amplitude. Sämtliche dargestellten Spannungsverläufe und Spektren wurden mit dem Programm read_ADC aufgezeichnet und mit OpenOffice.org Calc visualisiert.

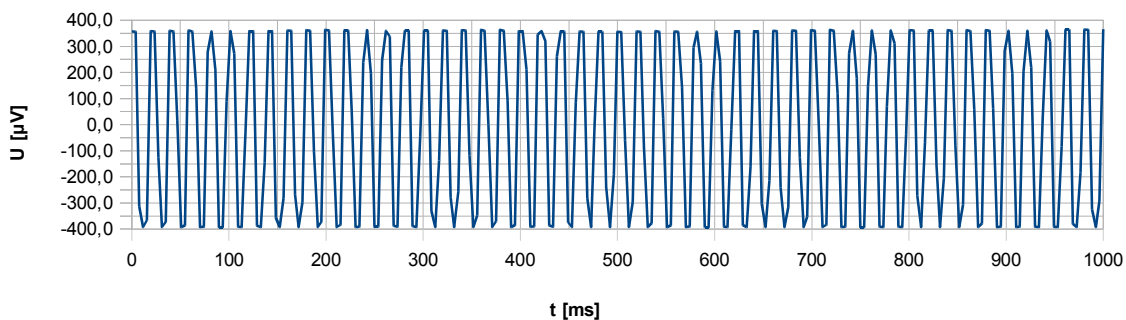


Abbildung 53: Signalverlauf im Leerlauf

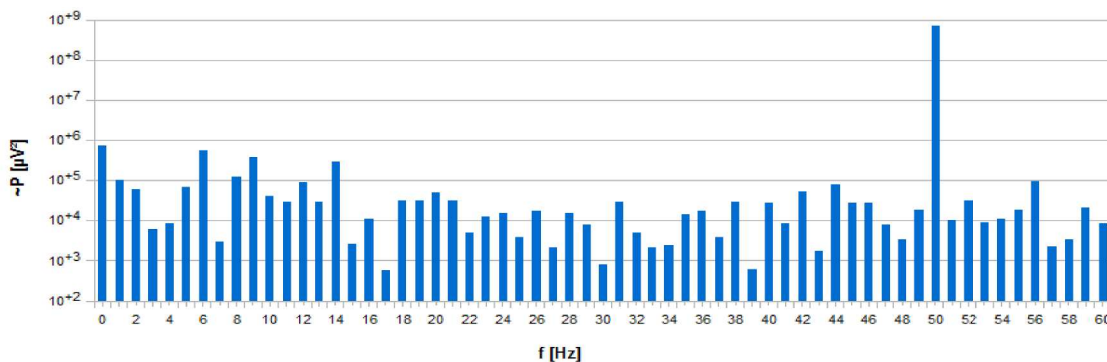


Abbildung 54: Leistungsspektrum im Leerlauf

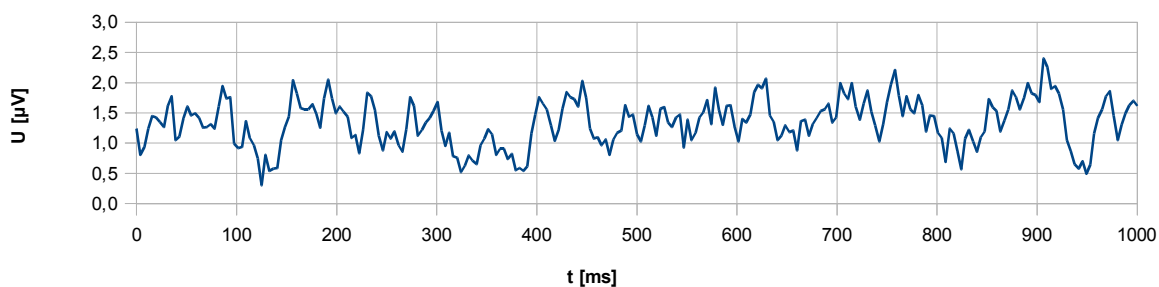


Abbildung 55: Signalverlauf bei Kurzschluss

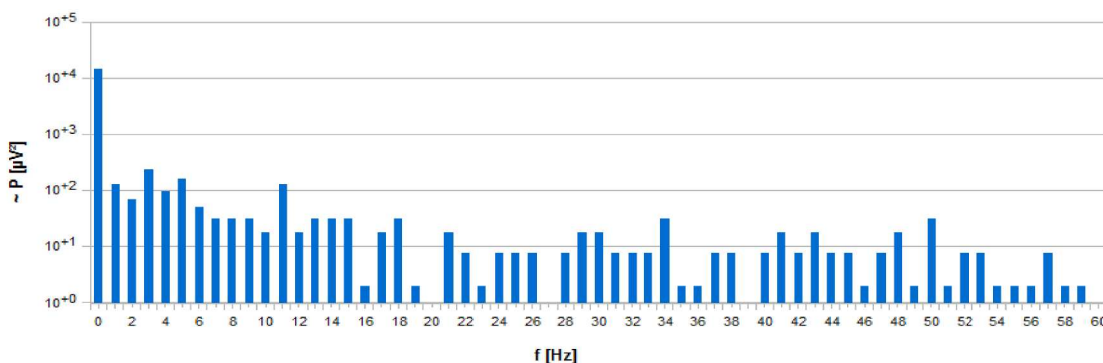


Abbildung 56: Leistungsspektrum bei Kurzschluss

Prototypentwicklung eines EEG-Headsets

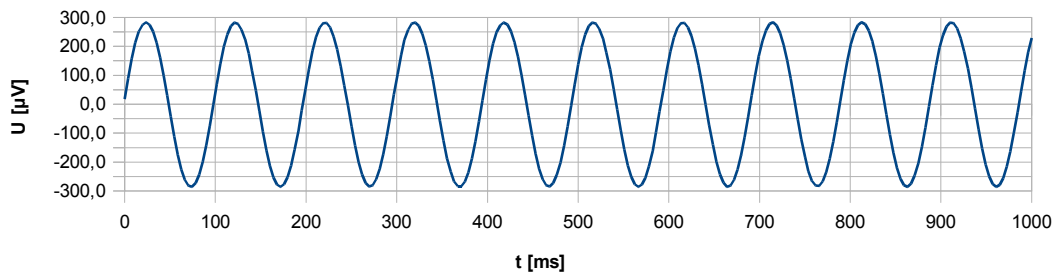


Abbildung 57: Signalverlauf 500 µV

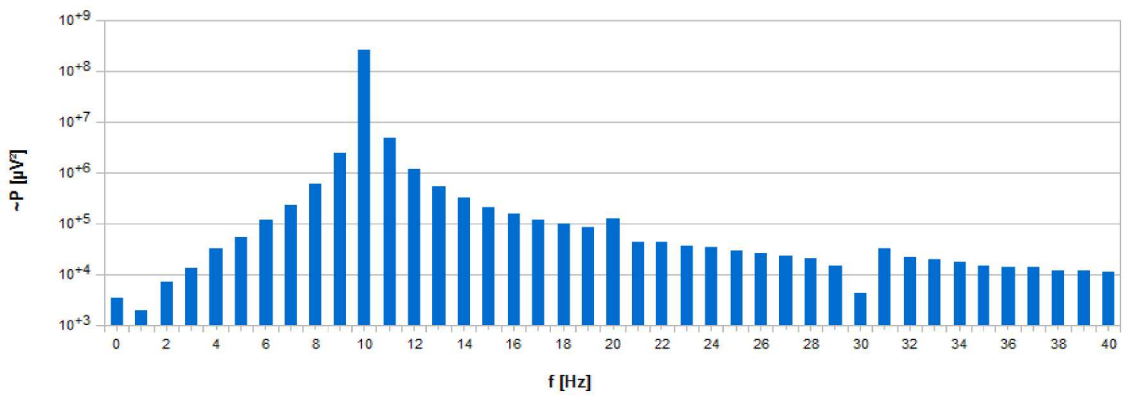


Abbildung 58: Leistungsspektrum 500 µV

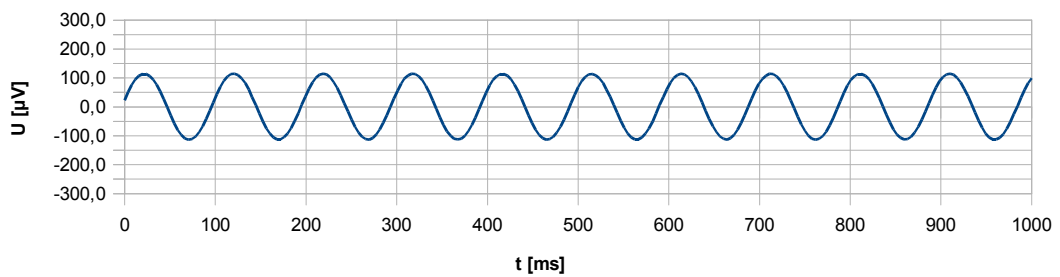


Abbildung 59: Signalverlauf 200 µV

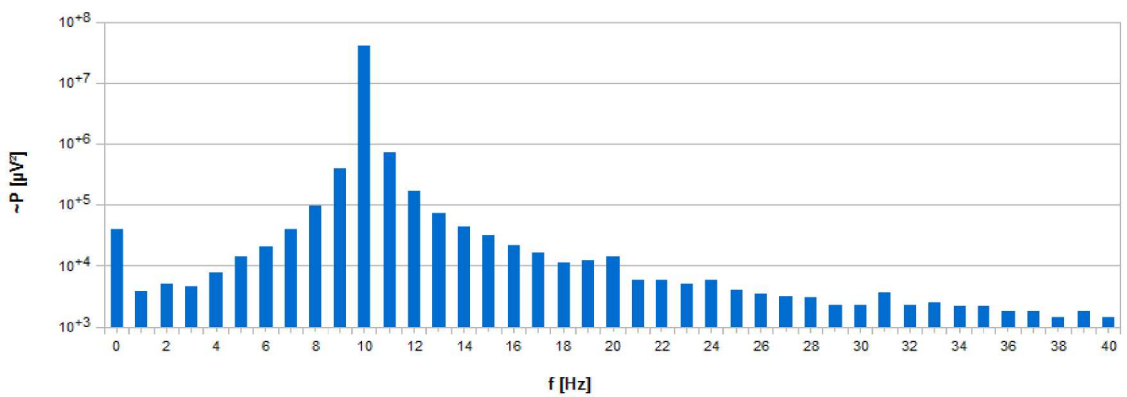


Abbildung 60: Leistungsspektrum 200 µV

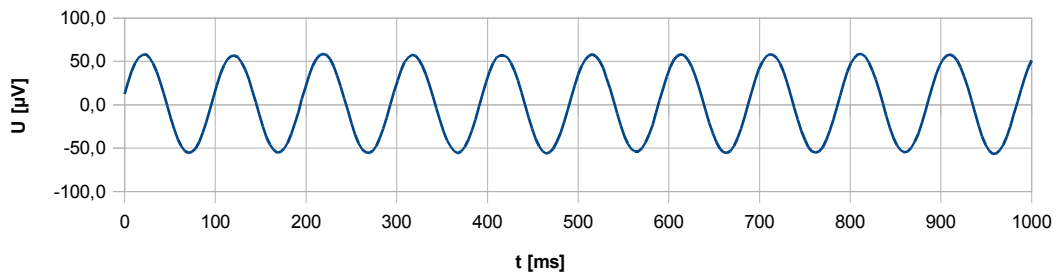


Abbildung 61: Signalverlauf 100 µV

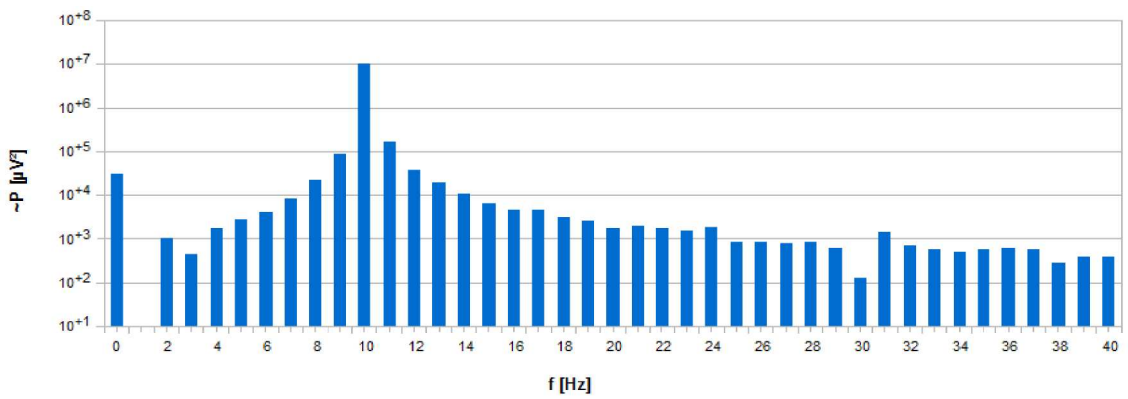


Abbildung 62: Leistungsspektrum 100 µV

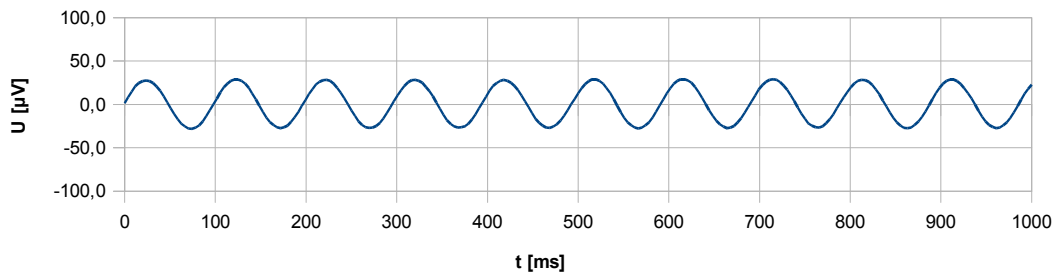


Abbildung 63: Signalverlauf 50 µV

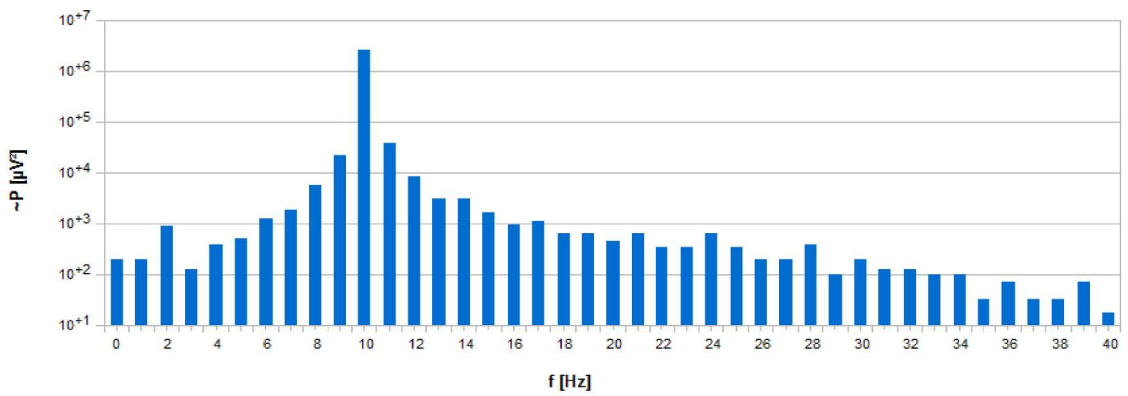


Abbildung 64: Leistungsspektrum 50 µV

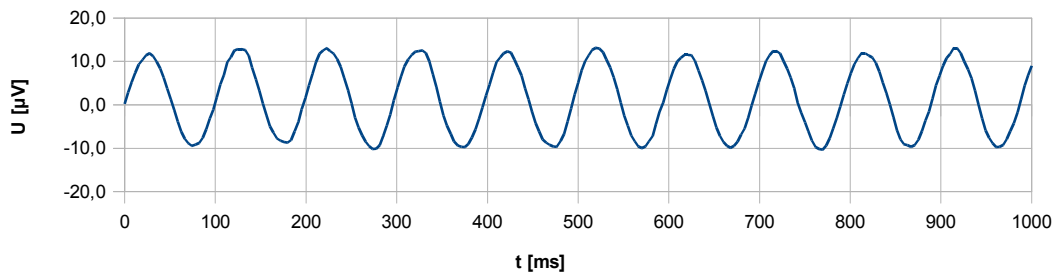


Abbildung 65: Signalverlauf 20 µV

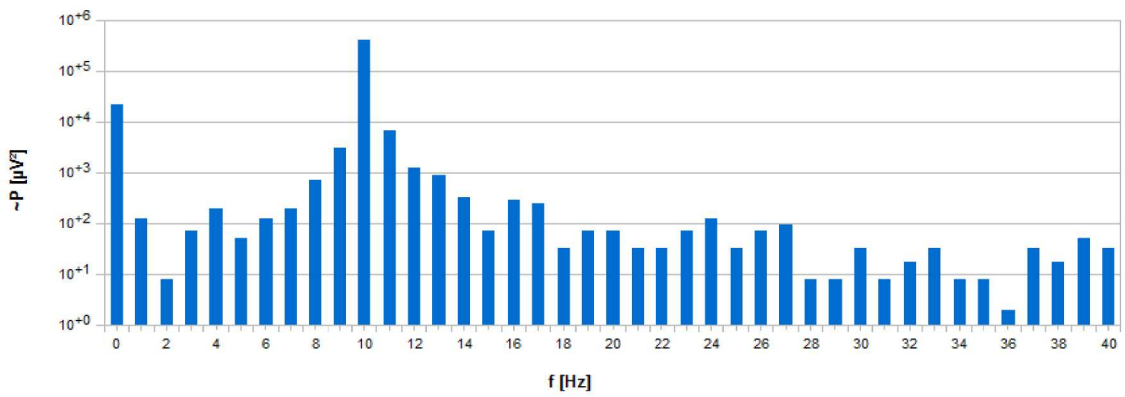


Abbildung 66: Leistungsspektrum 20 µV

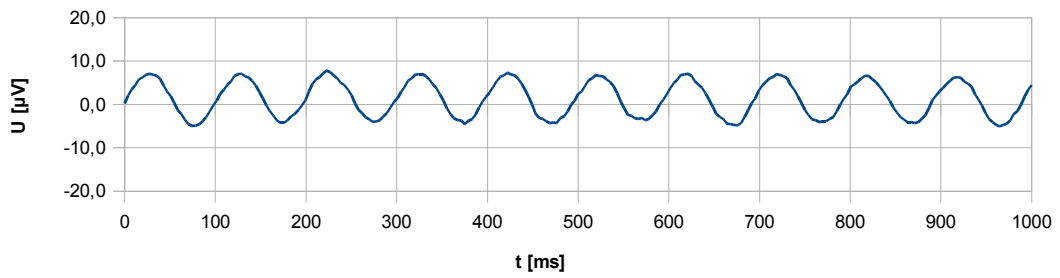


Abbildung 67: Signalverlauf 10 µV

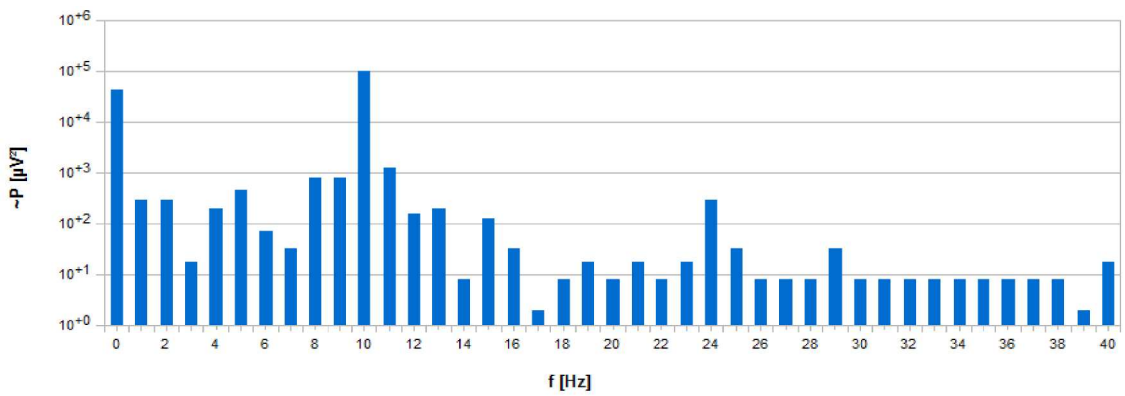


Abbildung 68: Leistungsspektrum 10 µV

Prototypentwicklung eines EEG-Headsets

Im Anschluss an die Testmessungen erfolgte eine Simulation des Brain-Switches mit Hilfe des Signalgenerators. Als erster Schritt erfolgten 5 Trials des Trainingsparadigmas mit dem Programmaufruf `BCI_training 5`. Der Signalgenerator wurde auf $10 \mu\text{V}$ eingestellt und bei jeder Aufforderung zur Bewegungsausführung die Spannung auf $20 \mu\text{V}$ erhöht. Abbildung 69 zeigt die erzielten gemittelten Leistungsspektren:

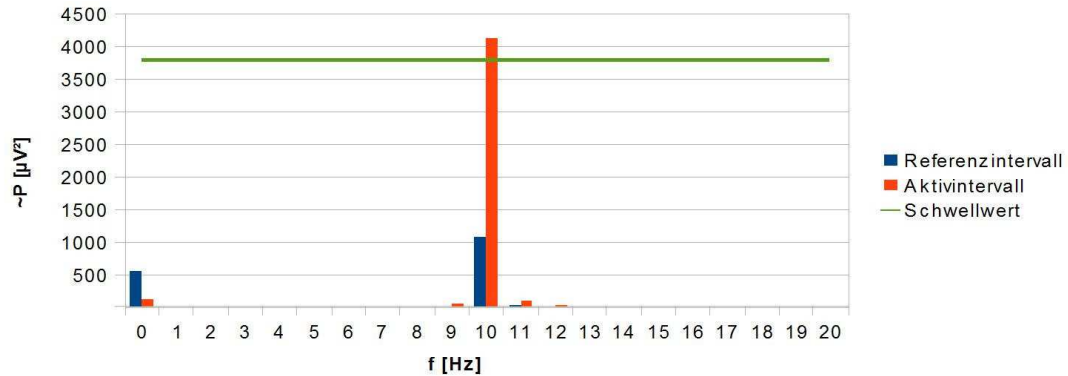


Abbildung 69: Leistungsspektren bei der Brain-Switch Simulation

Anhand der Leistungsspektren für den Referenz- und Aktivintervall kann die Frequenz mit dem größten Leistungsunterschied und ein entsprechender Schwellwert ermittelt werden. In dieser Simulation ist dies der Leistungsanstieg bei 10 Hz, der Schwellwert wurde mit 3800 angenommen.

Mit dem Programmaufruf `BCI 10 3800` wird die BCI-Applikation gestartet und die Schwellwertdetektion mit den übergebenen Parameter gestartet. Jedes mal wenn die Ausgangsspannung des Signalgenerators von 10 auf $20 \mu\text{V}$ erhöht wurde, wurde der Schwellwert überschritten und am Bildschirm „Device ON!“ ausgegeben.

Nachdem das EEG-Headset und der Brain-Switch erfolgreich überprüft werden konnten, wurde das Headset an einem Probanden montiert und ein EEG an der Elektrodenposition C_z abgeleitet.

Zunächst wurde mit dem Programm `read_ADC` ein EEG aufgezeichnet, das bewusst mit biologischen Artefakten kontaminiert wurde. Bei dem in Abbildung 70 abgebildetem EEG ist der Proband aufgefordert worden, mehrmals mit den Augen zu Blinzeln und die Zähne zusammen zu beißen.

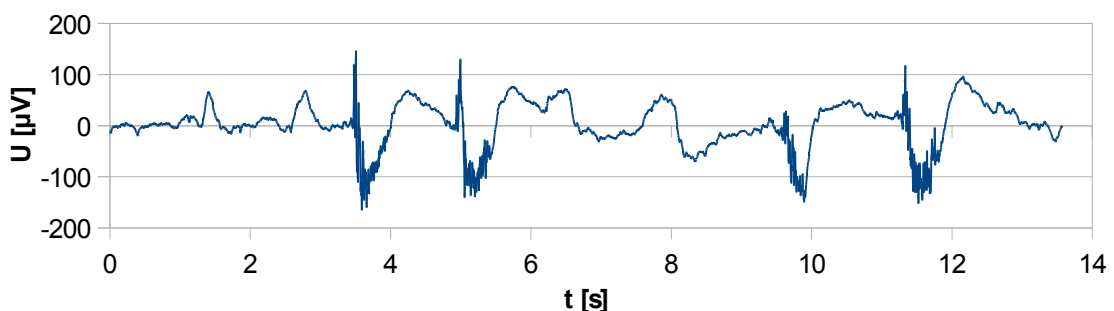


Abbildung 70: EEG mit biologischen Artefakten

Im Anschluss wurde ein EEG aufgezeichnet, um den Alpha-Rhythmus darzustellen. Der Proband hatte dabei zuerst die Augen geöffnet und wurde nach ca. 9 Sekunden aufgefordert, seine Augen zu schließen. Wie aus Abbildung 71 ersichtlich, stellte sich der Alpha-Rhythmus wie erwartet ein.

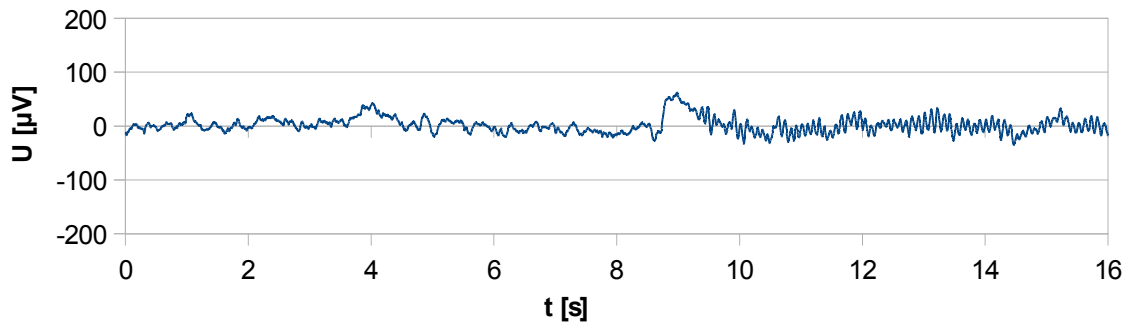


Abbildung 71: EEG mit Alpha-Rhythmus

Um den Beta-Rebound detektieren zu können, absolvierte der Proband nach Aufklärung über den Ablauf der Messung 25 Trials des Trainingsparadigmas. Abbildung 72 zeigt die gemittelten Leistungsspektren für das Referenz- bzw. Vergleichsintervall.

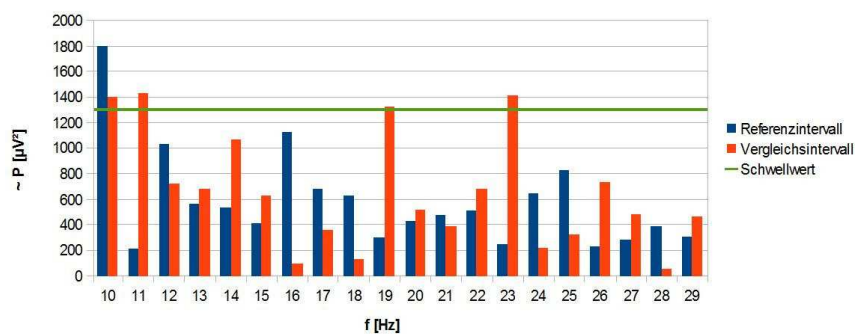


Abbildung 72: Ermittelte Leistungsspektren für den Brain-Switch

Bei dem Probanden konnte ein Beta-Rebound nachgewiesen werden. Die größte Leistungszunahme von über 500 % wurde bei 23 Hz identifiziert und diese Frequenz wurde zur Klassifikation für den Brain-Switch herangezogen. Der Schwellwert wurde mit 1300 festgelegt.

Nach dem Programmstart mit BCI 23 1300 wurde der Proband angehalten, 1 Minute lang keine Bewegungsausführung bzw. Bewegungsvorstellung durchzuführen. Während dieser Zeit wurde der Schwellwert fünf Mal überschritten (falsche Detektionen). Nach Aufforderung einer Bewegungsausführung konnte der Proband in 60% der Fälle einen Beta-Rebound generieren und eine korrekte Detektion auslösen.

Auch nach mehrmaliger Justierung des Schwellwertes konnte keine signifikante Verbesserung der Klassifikationsrate erzielt werden. Bei Erhöhung des Schwellwertes konnten die falschen Detektionen zwar reduziert werden, aber es war dem Probanden kaum noch möglich eine bewusste Überschreitung des Schwellwertes zu erreichen und vice versa.

6 Diskussion

Im Folgenden sollen die einzelnen Komponenten des EEG-Headsets bzw. die damit erzielten Ergebnisse diskutiert und Anregungen für Verbesserungen einer Weiterentwicklung des Prototypen gegeben werden.

Die EEG-Verstärkerschaltung des OpenEEG-Projektes konnte auf eine Versorgungsspannung von 3,3 V adaptiert werden. Der Verstärkungsfaktor der Schaltung wurde auf diese maximale Aussteuerungsgrenze angepasst, um den Spannungsbereich des EEGs von 10 bis 500 μV erfassen zu können. Die Grenzfrequenz der Hochpassfilter wurde gesenkt und die des Tiefpassfilters auf die europäische 50 Hz-Netzfrequenz angeglichen. Die mit dem Verstärker aufgenommen Testspannungen geben den Sinus des Signalgenerators selbst bei einer Amplitude von nur 10 μV sehr gut wieder. Um die Signalqualität zu verbessern, sollte die Ordnung des Filters erhöht werden. Dies könnte durch eine zusätzliche Filterstufe am Ausgang der 2. Verstärkerstufe (Sallen-Key Schaltung) erreicht werden. Eine weitere Möglichkeit bestünde in der Umwandlung der 1. Verstärkerstufe in eine Sallen-Key-Schaltung, analog zu der als 2. Verstärkerstufe in diesem Projekt verwendeten. Die dadurch erzielte Serienschaltung von zwei Filter 2.Ordnung würde ein Filter 4.Ordnung ergeben und sollte zu einer Verbesserung der Signalqualität führen.

Um das EEG-Headset mit vier Verstärkerplatinen (8 EEG-Kanäle) bestücken zu können, sollte die Bauweise der Platinen abgeändert werden. Die Abmessungen der Verstärker-Platine für 2 Kanäle beträgt 43 x 35 mm, diese kann jedoch noch weiter verringert werden. Empfehlenswert wäre die Verwendung einer 4-lagigen Platine mit der Spannungsversorgung in den mittleren beiden Schichten und je einen baugleichen Kanal auf jeder Außenlage. Dadurch sollte eine Reduktion der Abmessungen auf eine Größe von ca. 20 x 15 mm oder kleiner möglich sein. Um diese 4 Verstärkerplatinen auf dem Kopf platzsparend unterbringen zu können, sollten diese stehend montiert werden, wodurch eine Verringerung der Bauhöhe der Platinen nötig werden könnte. Eine liegende Montage der benötigten Anschluss-Buchsenleisten und Jumper wäre ein Lösungsansatz für diese Platzproblem. Des weiteren sollte die Möglichkeit geschaffen werden, die Elektrodenanschlüsse mit dem virtuellen Ground verbinden bzw. jumpern zu können, um die Funktionsweise der DRL-Schaltung im Falle eines unbenutzten Kanales nicht zu beeinträchtigen.

Das entwickelte Motherboard zur Aufnahme des Prozessormoduls CM-BF527 ist mit vielen Schnittstellen ausgestattet und braucht den Vergleich mit handelsüblichen Entwicklungsboards nicht zu scheuen, ist jedoch mit 80 x 51 mm wesentlich kleiner als diese. Eine weitere Schrumpfung wäre, analog zu der EEG-Platine, durch Verwendung einer mehrlagigen Platine mit innen liegender Spannungsversorgung aber noch durchaus möglich. Das Routing der Spannungsversorgung wurde von einem gemeinsamen Sternpunkt aus zu jeder Baugruppe separat ausgeführt, um gemeinsame Impedanzen

und daraus resultierende Störeinflüsse zu minimieren [113]. Diese Vorgehensweise benötigt jedoch viel Platz, der bei mehrlagiger Platinenausführung wesentlich reduziert werden könnte. Die Ausstattung des Motherboards ermöglicht eine vielseitige Verwendbarkeit für eine große Anzahl verschiedenartiger Projekte bzw. als Entwicklungsboard. Durch die Verwendung des Prozessormoduls, dem teuersten Element des Headsets, kann der leistungsfähige Prozessor in mehreren Projekten eingesetzt werden, da er nur über eine Steckverbindung mit dem Motherboard verbunden ist.

Das EEG-Headset wurde mit dem Gedanken konzipiert, ein komplett autonomes BCI-System darzustellen, ohne einen zusätzlichen Rechner zu benötigen. Dies ist jedoch für einen Prototypen unvorteilhaft, da stets die gemessenen EEG-Daten zur Speicherung bzw. Visualisierung benötigt werden. Zur Kommunikation zwischen dem Prozessor und dem Entwicklungsrechner wurde in diesem Projekt ein UART-USB-Adapter mit USB-Kabel verwendet, wodurch die Bewegungsfreiheit des Anwenders eingeschränkt ist. Kommerzielle EEG-Headsets verwenden zu diesem Zwecke meist Bluetooth zur kabellosen Kommunikation mit einem Rechner, dies könnte mit Hilfe eines UART-Bluetooth-Adapters ebenfalls gewährleistet werden.

Das Prozessormodul besitzt 8 MB Flash-Speicher, wobei diese in 4 Speicherblöcke aufgeteilt sind und mit Hilfe von zwei GPIO-Pins, PH9 und PG11, adressiert werden [33]. Leider wurde Pin PH9 als SPI-Chip-Select für die SD-Karte verwendet. Da der SD-Kartenslot nicht bestückt wurde, stellt dies im Moment kein Problem dar. Sollte das Motherboard jedoch mit dieser Funktion erweitert werden, sollte darauf Rücksicht genommen werden, um Probleme beim Datenzugriff zu vermeiden.

Durch die Verwendung der konfigurierbaren Linux-Distribution uClinux mit inkludierten Linux-Kernel 2.6 steht ein lizenzfreies und stabiles Betriebssystem zur Verfügung. Das mit der Toolchain erzeugte Linuximage mit Benutzerprogrammen benötigt lediglich einen Speicherbedarf von 2 MB bei vollem linuxüblichen Leistungsumfang und ist daher gut geeignet für portable Systeme mit eingeschränkten Speicherkapazitäten. Da unter uClinux sämtliche Hardware-Zugriffe ident mit jenen von Linux-Desktop-Varianten sind, ist eine Portierung von bereits bestehenden C-Softwarelösungen problemlos zu bewerkstelligen. Um ein echtzeitfähiges System zu erhalten, wurde mit Xenomai ein zweiter Kernel mit Hilfe der Adeos I-Pipe ins System mit eingebunden. Diese Konstruktion ist jedoch ein Auslaufmodell, da es bereits Bestrebungen gibt, den Linux-Kernel selbst echtzeitfähig zu gestalten.

Zur Softwareentwicklung ist mit blackfin-gcc ein einfach zu bedienender Compiler verfügbar, mit dessen Verwendung aus einfachen Textdateien lauffähige Programme erzeugt werden können. Bei komplexeren Programmen, die mehrere Bibliotheken benötigen, ist eine Verwendung von Eclipse empfehlenswert, da bei dem Blackfin-Datenformat FLAT nur statisches linken unterstützt wird und daher die Reihenfolge der Verlinkung eine essentielle Rolle einnimmt. In den Projekt-Einstellungen von Eclipse ist dies, im Vergleich mit der textbasierten Konsolen-Eingabeaufforderung, aufgrund der graphischen Darstellung wesentlich übersichtlicher zu gewährleisten und auch die Fehlersuche gestaltet sich wesentlich einfacher.

Die im Rahmen dieses Projektes erstellten Programme zur Umsetzung eines Brain-Switchs sind sehr einfach gehalten, geben jedoch die prinzipielle Vorgehensweise zur Realisation eines BCI-Systems wieder.

Verbesserungsmöglichkeiten bestehen in einer Vorverarbeitung des Signals durch digitale Filter, der rechnerischen Elimination von Artefakten und Verfahren zur Erhöhung der Signalqualität, wie z.B. PCA oder ICA.

Das entwickelte BCI-Trainingsparadigma zur Erfassung der anwenderspezifischen Frequenzbänder, in denen es zu einem Beta-Rebound bei Bewegungsvorstellung kommt, besitzt eine starre zeitlich Struktur zur Gewinnung der benötigten Leistungsspektren. So wird das Leistungsspektrum in dem Zeitfenster von 2-3s nach Aufforderung zur Bewegungsvorstellung ermittelt. Die Reaktionszeit, die Dauer der Bewegung sowie die Latenz bis zum Auftreten des Beta-Rebounds sind jedoch bei jedem Menschen unterschiedlich. Eine verbesserte Variante wäre eine ständige Zwischenspeicherung jenes Spektrums und dessen Latenz, mit der jeweils größten Leistung im Beta-Bereich nach der Bewegungsaufforderung und die weitere Verwertung ausschließlich dieses Maximalspektrums in der weiteren Analyse. Mit dieser Variante wären auch Beta-Rebounds detektierbar, die nicht innerhalb der Sekunde 6-7 des Paradigmas auftreten. Eine Erkennung von biologischen Artefakten während eines Trials, wie z.B. Blinzeln oder Zähne knirschen, und Verwerfung der kontaminierten Trials würde das Ergebnis weiter verbessern. Des weiteren sollte eine von der Dauer zufällige Pause zwischen den Trials eingebunden werden, da bei einem Training mit vielen Trials die Erwartungshaltung des Probanden bei immer gleichem Timing negative Einflüsse auf das Ergebnis ausübt.

Das gewählte Schwellwertverfahren ist das einfachste der Klassifikationsverfahren und die Ergebnisse könnten durch komplexere Verfahren signifikant verbessert werden. Bei Beibehaltung des Schwellwertverfahrens sollten, anstatt der Überprüfung der Leistungszunahme in nur einer Frequenz, mehrere Frequenzen bzw. Frequenzbänder auf eine Überschreitung eines Schwellwertes überprüft werden, da es meist in mehreren Bändern zu einem Beta-Rebound kommt. Dies sollte das Auftreten von falschen, unbeabsichtigten Detektionen verringern.

Da der Großteil am Institut für Semantische Datenanalyse realisierten Anwendungen auf MATLAB basierend ist, sollte in Folgeprojekten eine Anbindung an MATLAB realisiert werden. Dadurch könnte auf diese Projekte zurückgegriffen werden und bereits realisierte, komplexere Anwendungen verwendet, sowie die Visualisierung der gemessenen Daten vorgenommen werden.

Sämtliche an das Projekt gesetzten Ziele wurden erreicht und ein leistungsfähiges EEG-Headset realisiert. Hervorzuheben sind die kompakte Bauweise, das geringe Gewicht, die sehr gute Signalqualität des Verstärkers sowie die vielseitigen Einsatzmöglichkeiten des Motherboards. Durch die Verwendung von ausschließlich lizenzfreier Software wurde eine kostengünstige Basis für die Weiterentwicklung von portablen EEG-Systemen geschaffen, auf denen mit einfachen Mitteln auch komplexe BCI-Systeme implementiert werden können.

Literaturverzeichnis

- 1: Analog Device, JTAG-USB Adapter gnICE+
<http://docs.blackfin.uclinux.org/doku.php?id=hw:jtag:gnice-plus> (Mai 2012)
- 2: Analog Devices, ADSP-BF52x Blackfin Processor Hardware Reference
http://www.analog.com/static/imported-files/processor_manuals/BF52xHRM_Rev.1.0.pdf (Mai 2012)
- 3: Analog Devices, Blackfin Processor Programming Reference.
http://www.analog.com/static/imported-files/processor_manuals/Blackfin_pgr_rev2.0.pdf (Mai 2012)
- 4: Analog Devices, C/C++ Compiler and Library Manual for Blackfin Processors.
http://www.analog.com/static/imported-files/software_manuals/50_bf_cc_rtl_mn_rev_5.4.pdf (Mai 2012)
- 5: Bauernfeind G., Leeb R., et al.: Development, set-up and first results for a one-channel near-infrared spectroscopy system. *Biomedical Engineering* 53 (1): 36-43 (2008)
- 6: Bear M.F., Connors B.W., et al.: Neuronen und Gliazellen. In: Bear M.F., Connors B.W., et al.: *Neurowissenschaften*. Spektrum Akademischer Verlag: Kapitel 2. (2009) ISBN: 978-3-8274-2028-2
- 7: Beierlein T., Hagenbruch O.: *Mikroprozessortechnik*. Carl Hanser Verlag (2004), ISBN: 3-446-22072-0
- 8: Berger H.: Über das Elektrenkephalogramm des Menschen. *Archiv für Psychiatrie und Nervenkrankheiten* 87: 527-570 (1929)
- 9: Blackfin Eclipse Plugins: www.blackfin.uclinux.org/eclipse/ (Mai 2012)
- 10: Blackfin Toolchain project:
<http://blackfin.uclinux.org/gf/project/toolchain> (Mai 2012)
- 11: Blinowska K., Durka P.: Electroencephalography (EEG). In: Webster J.G.: *Encyclopedia of Medical Devices and Instrumentation, 2nd Edition*. John Wiley & Sons: Seite 111. (2006) ISBN: 978-0-471-26358-6
- 12: Bluetechnix, JTAG-USB Adapter gnICE+
<http://www.bluetechnix.com/rainbow2006/site/464/Default.aspx> (Mai 2012)
- 13: Brunner C.: Vorlesungsunterlagen Informationsverarbeitung im Menschen (2010)
- 14: Cashero Z.: Comparison of EEG Preprocessing Methods to Improve the Classification of P300 Trials. Master Thesis, Colorado State University (2011)

- 15: Chen X., Bai O.: Towards Multi-Dimensional Robotic Control via Noninvasive Brain-Computer Interface. *International Conference on Complex Medical Engineering* : 1-5 (2009)
- 16: Chi Y.M., Wang Y.T., et al.: Dry and Noncontact EEG Sensors for Mobile Brain-Computer Interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 20 (2): 228-235 (2012)
- 17: Coyle S., Ward T., et al.: On the suitability of near-infrared (NIR) systems for next-generation brain-computer interfaces. *Physiological Measurement* 25 (4): 815-822 (2004)
- 18: Datenblatt Aluminium-Elektrolyt Kondensator von Panasonic: EEE1CS100SR. <http://industrial.panasonic.com/www-data/pdf/ABA0000/ABA0000CE2.pdf>
(Mai 2012)
- 19: Datenblatt Analog-Digital-Converter von Texas Instruments: ADS8345. <http://www.ti.com/lit/ds/symlink/ads8345.pdf> (Mai 2012)
- 20: Datenblatt Buchsenleiste von BKL Electronic: 2,54 mm, 20-polig. http://www.bkl-electronic.de/pdf_datenblatt/10120838.pdf (Mai 2012)
- 21: Datenblatt Chip LED von Kingbright: KP-2012MGC. <http://www.kingbright.com/manager/upload/pdf/KP-2012MGC%28Ver.16%29.pdf>
(Mai 2012)
- 22: Datenblatt Chip-Widerstand von TE Connectivity : CPF Serie. <http://passives.te.com/documents/webservice/fetch.ashx?fileid=9675&docId=6730> (Mai 2012)
- 23: Datenblatt Chip-Widerstand von Panasonic: ERA6AEB. <http://industrial.panasonic.com/www-data/pdf/AOA0000/AOA0000CE26.pdf>
(Mai 2012)
- 24: Datenblatt Digitaler Signal Prozessor von Analog Devices: ADSP-BF527. http://www.analog.com/static/imported-files/data_sheets/ADSP-BF522_BF523_BF524_BF525_BF526_BF527.pdf (Mai 2012)
- 25: Datenblatt ESD Schutzdiode von ST Microelectronics: USBLC6. http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00050750.pdf (Mai 2012)
- 26: Datenblatt Ethernetbuchse von Taimag: RJLBC-060TC1. <http://www.taimag.com/PDF/RJ/RJLBC-xxXT%20series/RJLBC-060TC1%E5%9E%8B%E9%8C%84%282005.09.08%29.pdf> (Mai 2012)
- 27: Datenblatt Kodierbrücke von BKL Electronic: 2,54 mm. http://www.bkl-electronic.de/pdf_datenblatt/10120908.pdf (Mai 2012)

- 28: Datenblatt Kondensator von AVX Corporation: 0805 X7R.
<http://www.avx.com/docs/Catalogs/cx7r.pdf> (Mai 2012)
- 29: Datenblatt Kondensator Keramik von Kemet: C0805 X7R.
[http://www.kemet.com/kemet/web/homepage/kechome.nsf/weben/C4DA2DA1D1C9D3C3CA2570A50016092D/\\$file/KEM_C1002_X7R_SMD.pdf](http://www.kemet.com/kemet/web/homepage/kechome.nsf/weben/C4DA2DA1D1C9D3C3CA2570A50016092D/$file/KEM_C1002_X7R_SMD.pdf) (Mai 2012)
- 30: Datenblatt Mini-USB zu UART-Konverter von Conrad: Version 11/09.
http://www.produktinfo.conrad.com/datenblaetter/175000-199999/197326-an-01-de-MINI_USB_TO_UART_CONVERTER_de_en_fr_nl.pdf (Mai 2012)
- 31: Datenblatt Operationsverstärker von National Semiconductor: LMP2012.
<http://www.ti.com/lit/ds/symlink/lmp2012.pdf> (Mai 2012)
- 32: Datenblatt Präzessions-Instrumentenverstärker von Analog Devices: AD8553.
http://www.analog.com/static/imported-files/data_sheets/AD8553.pdf (Mai 2012)
- 33: Datenblatt Prozessor-Modul von Bluetech: CM-BF527.
http://datasheets.bluetech.at/goto/CM-BF527/CM-BF527_HUM_V2.pdf
(Mai 2012)
- 34: Datenblatt SD Kartenslot von Hirose: DM3AT.
http://www.hirose.co.jp/catalogue_hp/e60900232.pdf (Mai 2012)
- 35: Datenblatt SMD-Taster von Diptronics Manufacturing: DTSMG-6 .
<http://www.dip.com.tw/PDF/DTSM%28G%29-6%282%29-V.pdf> (Mai 2012)
- 36: Datenblatt Spannungsregulator von Texas Instruments: TPS7A4533.
<http://www.ti.com/lit/ds/symlink/tps7a4533.pdf> (Mai 2012)
- 37: Datenblatt Steckverbinder von Hirose: FX8 Serie.
http://www.hirose.co.jp/catalogue_hp/e57800016.pdf (Mai 2012)
- 38: Datenblatt Stiftheile von BKL Elektronik: 2,54 mm, 20-polig.
http://www.bkl-electronic.de/pdf_datenblatt/10120182.pdf (Mai 2012)
- 39: Datenblatt Tip Switches von TE Connectivity: GDH Serie.
www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Customer+Drawing%7F1571983%7FC3%7Fpdf%7FEnglish%7FENG_CD_1571983_C3.pdf%7F1571983-5 (Mai 2012)
- 40: Datenblatt USB-Buchse von Molex: USB Mini-B.
http://www.molex.com/webdocs/datasheets/pdf/en-us/0675031020_IO_CONNECTORS.pdf (Mai 2012)

- 41: Diez P.F., Mut V., et al.: A Comparison of Monopolar and Bipolar EEG Recordings for SSVEP Detection. *32nd Annual International Conference of the IEEE EMBS* : 5803-5806 (2010)
- 42: Donchin E., Callaway E., et al.: Event-Related Brain Potentials in Man. *Academic Press New York*: 349-411 (1978)
- 43: EEG-Headset: Enobio von Neuroelectrics.
<http://www.neuroelectrics.com/enobio> (Mai 2012)
- 44: EEG-Headset: EPOC von Emotiv Systems, Inc.
<http://www.emotiv.com/store/hardware/epoc-bci/epoc-neuroheadset/> (Mai 2012)
- 45: EEG-Headset: Mindwave von NeuroSky, Inc.
<http://store.neurosky.com/products/mindwave-1> (Mai 2012)
- 46: EEG-Headset: Prototyp von IMAC International und Holst Centre.
http://www2.imec.be/be_en/press/imec-news/archive-2011/imeceegmdm-west.html (Mai 2012)
- 47: EEG-Headset: Prototyp Mynd von NeuroFocus, Inc.
http://www.neurofocus.com/pdfs/Mynd_NeuroFocus.pdf (Mai 2012)
- 48: Fehmi L.G., Collura T.: Effects of Electrode Placement Upon EEG Biofeedback Training: The Monopolar-Bipolar Controversy. *Journal of Neurotherapy* 11 (2): 45-63 (2007)
- 49: g.tec medical engineering GmbH, Comparison of Active versus Passive Electrodes. ftp://sierra.iem.pw.edu.pl/BCI/gTEC/gGAMMAsys_ActivePassiveComparison%5B1%5D.pdf (2010)
- 50: GNU Binutils: <http://www.gnu.org/software/binutils/> (Mai 2012)
- 51: GNU Compiler Collection: <http://gcc.gnu.org/> (Mai 2012)
- 52: Gonzalez-Franco M., Yuan P., et al.: Motor imagery based brain-computer interface: a study of the effect of positive and negative feedback. *33rd Annual International Conference of the IEEE EMBS* : 6323-6326 (2011)
- 53: Graimann B., Huggins J.E., et al.: Visualization of significant ERD/ERS patterns in multichannel EEG and ECoG data. *Clinical Neurophysiology* 119 (1): 43-47 (2002)
- 54: Heinrich S.P.: Some thoughts on the interpretation of steady-state evoked potentials. *Documenta Ophthalmologica* 120 (3): 205-214 (2010)
- 55: Jahankhani P., Revett K., Kodogiannis V.: Data Mining an EEG Dataset with an Emphasis on Dimensionality Reduction. *IEEE Symposium on Computational Intelligence and Data Mining* : 405-412 (2007)

- 56: Jasper H.H.: The ten–twenty electrode system of the International Federation. *Electroencephalography and Clinical Neurophysiology* 10 (2): 371-375 (1958)
- 57: Jurcak V., Tsuzuki D., Dan I.: 10/20, 10/10, and 10/5 systems revisited: Their validity as relativehead-surface-based positioning systems. *NeuroImage* 34 (4): 1600-1611 (2007)
- 58: Kachenoura A., Albera L., et al.: ICA: A Potential Tool for BCI Systems. *IEEE Signal Processing Magazine* 25 (1): 57-68 (2008)
- 59: Kavitha P.T., Lau C.T., Premkumar A.B.: Modified Ocular Artifact Removal Technique from EEG by Adaptive Filtering. *6th International Conference on Information, Communications and Signal Processing* : 1-5 (2007)
- 60: Kornhuber H.H., Deecke L.: Bereitschaftspotential und Willensfreiheit. *Schweitzer Archiv für Neurologie und Psychiatrie* 159 (3): 133 (2008)
- 61: Leeb R., Lee F., et al.: Brain–Computer Communication: Motivation, Aim and Impact of Exploring a Virtual Apartment. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 15 (4): 473-482 (2007)
- 62: Leins U., Goth G., et al.: Neurofeedback for children with ADHD: a comparison of SCP and Theta/Beta protocols.. *Applied Psychophysiology and Biofeedback* 32 (2): 73-88 (2007)
- 63: Leuthardt E.C., Schalk G., et al.: A brain–computer interface using electrocorticographic signals in humans. *Journal of Neural Engineering* 1: 63-71 (2004)
- 64: Libet B., Gleason C.A., et al.: Time of conscious intention to act in relation to onset of cerebral activity (readiness-potential). The unconscious initiation of a freely voluntary act. *Brain* 106 (3): 623-642 (1983)
- 65: Lopez-Larraz E., Creatura M., et al.: EEG Single-Trial Classification of Visual, Auditive and Vibratory Feedback Potentials in Brain-Computer Interfaces. *33rd Annual International Conference of the IEEE EMBS* : 4231-4234 (2011)
- 66: McFarland D.J., Anderson C.W., et al.: BCI Meeting 2005—Workshop on BCI Signal Processing:Feature Extraction and Translation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 14 (2): 135-138 (2006)
- 67: Millan J.R., Galan F., et al.: Asynchronous Non-Invasive Brain-Actuated Control of an Intelligent Wheelchair. *31st Annual International Conference of the IEEE EMBS* : 3361-3364 (2009)
- 68: Moore M.M.: Real-World Applications for Brain–Computer Interface Technology. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 11 (2): 162-165 (2003)
- 69: Müller K., Vigarío R., et al.: Blind Source Separation Techniques for Decomposing Event-Related Brain Signals. *International Journal of Bifurcation and Chaos* 14 (2): 773-791 (2004)

- 70: Müller-Putz G.R.: Vorlesungsunterlagen Rehabilitationstechnik (2008)
- 71: Müller-Putz G.R., Kaiser V., et al.: Fast set-up asynchronous brain-switch based on detection of foot motor imagery in 1-channel EEG. *Medical and Biological Engineering and Computing* 48 (3): 229-233 (2010)
- 72: Müller-Putz G.R., Pfurtscheller G.: Control of an Electrical Prosthesis with an SS-VEP-Based BCI. *IEEE Transactions on Biomedical Engineering* 55 (1): 361-364 (2008)
- 73: Müller-Putz G.R., Scherer R., et al.: EEG-based neuroprosthesis control: A step towards clinical practice. *Neuroscience Letters* 382 (1-2): 169-174 (2005)
- 74: Müller-Putz G.R., Scherer R., et al.: Steady-state visual evoked potential(SSVEP)-based communication: impact of harmonic frequency components. *Journal of Neural Engineering* 2 (4): 123-130 (2005)
- 75: Nagel J.H.: Biopotential Amplifiers. In: Bronzino J.D.: *The Biomedical Engineering Handbook: Second Edition*. CRC Press LLC: Kapitel 70. (1999) ISBN: 978-0849385940
- 76: Neuman M.R.: Biomedical Sensors. In: Bronzino J.D.: *The Biomedical Engineering Handbook: Second Edition*. CRC Press LLC: Kapitel 48. (1999) ISBN: 978-0849385940
- 77: Neuper C., Scherer R., et al.: Motor imagery and action observation: Modulation of sensorimotor brainrhythms during mental control of a brain–computer interface. *Clinical Neurophysiology* 120 (2): 239-247 (2009)
- 78: Oppenheim A.V., Schafer R.W., Buck J.R.: *Zeitdiskrete Signalverarbeitung*. Pearson Studium (2004), ISBN: 3-8273-7077-9
- 79: Paukkunen A., Sepponen R.: The effect of ground electrode on the sensitivity, symmetricity and technical feasibility of scalp EEG recordings. *Medical and Biological Engineering and Computing* 46 (9): 933-938 (2008)
- 80: Pfurtscheller G., Lopes da Silva F.H.: Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clinical Neurophysiology* 110: 1842-1857 (1999)
- 81: Pfurtscheller G., Müller-Putz G.R., et al.: Rehabilitation with Brain-Computer Interface Systems. *Computer* 41 (10): 58-65 (2008)
- 82: Pfurtscheller G., Neuper C., et al.: Current Trends in Graz Brain–Computer Interface (BCI) Research. *IEEE Transactions on Rehabilitation Engineering* 8 (2): 216-219 (2000)
- 83: Pfurtscheller G., Neuper C., et al.: Visually guided motor imagery activates sensorimotor areas in humans. *Neuroscience Letters* 269 (3): 153-156 (1999)
- 84: Pfurtscheller G., Solis-Escalante T.: Could the beta rebound in the EEG be suitable to realize a “brain switch”? *Clinical Neurophysiology* 120 (1): 24-29 (2009)

- 85: Portables EEG-System: EEG mobile von SIGMA Medizin-Technik GmbH.
<http://neurowerk.de/produkte/eeg-geraete/neurowerk-eeg-mobile/> (Mai 2012)
- 86: Portables EEG-System: NeuroScape von Rising Medical Equipment Co., Ltd.
<http://risingmed.en.made-in-china.com/product/fMNmFehGhAYq/China-24-32-Channel-Digital-EEG-Neuroscape-EEG-24-EEG-32.html> (Mai 2012)
- 87: Portables EEG-System: g.MOBllab von g.tec medical engineering GmbH.
<http://www.gtec.at/Products/Hardware-and-Accessories/g.MOBllab-Specs-Features> (Mai 2012)
- 88: Quiroga R.Q.: Evoked Potentials. In: Webster J.G.: *Encyclopedia of Medical Devices and Instrumentation, 2nd Edition*. John Wiley & Sons: Seite 233-246. (2006)
ISBN: 978-0-471-26358-6
- 89: Ramoser H., Müller-Gerking J., Pfurtscheller G.: Optimal Spatial Filtering of Single Trial EEG During Imagined Hand Movement. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 8 (4): 441-446 (2000)
- 90: Regan D.: Some characteristics of average steady-state and transient responses evoked by modulated light. *Electroencephalography and Clinical Neurophysiology* 20 (3): 238-248 (1966)
- 91: Schachinger D., Schindler K., Kluge T.: Automatic Reduction of Artifacts in EEG-Signals. *15th International Conference on Digital Signal Processing* : 143-146 (2007)
- 92: Scherer R., Müller G.R., et al.: An Asynchronously Controlled EEG-Based Virtual-Keyboard: Improvement of the Spelling Rate. *IEEE Transactions on Biomedical Engineering* 51 (6): 979-984 (2004)
- 93: Schlögl A: Skriptum zu den Laborübungen in Grundlagen der Biomedizinischen Technik (2005)
- 94: Senapati K., Kar S., Routray A.: A New Technique for Removal of Ocular Artifacts from EEG Signals Using S-Transform. *International Conference on Systems in Medicine and Biology* : 113-116 (2010)
- 95: Shibasaki H., Hallett M.: What is the Bereitschaftspotential?. *Clinical Neurophysiology* 117: 2341-2356 (2006)
- 96: Sitaram R., Weiskopf N.: fMRI Brain-Computer Interface. *IEEE Signal Processing Magazine* 25 (1): 95-106 (2008)
- 97: Solis-Escalante T., Müller-Putz G.R., et al.: Cue-induced beta rebound during withholding of overt and covert foot movement. *Clinical Neurophysiology* [Epub ahead of print]: (2012)
- 98: Stallman R.M., Using the GNU Compiler Collection.
<http://gcc.gnu.org/onlinedocs/gcc-4.4.7/gcc.pdf> (2008)

- 99: Tallgren P., Vanhatalo S., et al.: Evaluation of commercially available electrodes and gels for recording of slow EEG potentials. *Clinical Neurophysiology* 116: 799-806 (2005)
- 100: Texas Instruments, Analysis of the Sallen-Key Architecture.
<http://www.vyssotski.ch/BasicsOfInstrumentation/AnalysisOfTheSallen-KeyArchitecture.pdf> (2002)
- 101: The Adeos Project : <http://home.gna.org/adeos/> (Mai 2012)
- 102: The Eclipse Foundation: <http://www.eclipse.org/> (Mai 2012)
- 103: The Linux Kernel Archives: <http://www.kernel.org/> (Mai 2012)
- 104: The OpenEEG project: <http://openeeg.sourceforge.net/doc/> (Mai 2012)
- 105: The Universal Boot Loader by DENX:
<http://www.denx.de/wiki/U-Boot/WebHome> (Mai 2012)
- 106: Tietze U., Schenk C.: *Halbleiter Schaltungstechnik*. Springer Berlin Heidelberg (2002), ISBN: 3-540-42849-6
- 107: uClibc Homepage: <http://uclibc.org/> (Mai 2012)
- 108: uClinux for blackfin: <http://blackfin.uclinux.org/gf/> (Mai 2012)
- 109: Väisänen J., Ryyänen O., et al.: Analysing Specificity of a Bipolar EEG Measurement. *Proceedings of the 28th IEEE EMBS Annual International Conference* : 1103-1106 (2006)
- 110: Vetter R.J., Williams J.C., et al.: Chronic Neural Recording Using Silicon-Substrate Microelectrode Arrays Implanted in Cerebral Cortex. *IEEE Transactions on Biomedical Engineering* 51 (6): 896-904 (2004)
- 111: Vidal J.J.: Toward Direct Brain-Computer Communication. *Annual review of biophysics and bioengineering* 2: 157-180 (1973)
- 112: Walter W.G., Cooper R., et al.: Contingent negative variation: An electric sign of sensorimotor association and expectancy in the human brain. *Nature* 203: 380-384 (1964)
- 113: Williams T.: *The Circuit Designer's Companion*. Newnes (2005), ISBN: 978-0-7506-6370-0
- 114: Winter B.B., Webster J.G.: Driven-Right-Leg Circuit Design. *IEEE Transactions on Biomedical Engineering* 30 (1): 62-66 (1983)
- 115: Wolpaw J.R., Birbaumer N., et al.: Brain-computer interfaces for communication and control. *Clinical Neurophysiology* 113 (6): 767-791 (2002)
- 116: Wolpaw J.R., Loeb G.E., et al.: BCI Meeting 2005—Workshop on Signals and Recording Methods. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 14 (2): 138-141 (2006)

- 117: Wyckoff S., Strehl U.: Feedback of Slow Cortical Potentials: Basics, Application and Evidence. In: Coben R., Evans J.R.: *Neurofeedback and Neuromodulation Techniques and Applications*. Academic Press: Kapitel 8. (2010) ISBN: 978-0-12-382235-2
- 118: Xenomai: Real-Time Framework for Linux: <http://www.xenomai.org/> (Mai 2012)
- 119: Yang Y.; Li Z., et al.: Noise Source Separation based on the Blind Source Separation. *Chinese Control and Decision Conference* : 2236-2240 (2011)
- 120: Yao D., Wang L., et al.: A comparative study of different references for EEG spectral mapping: the issue of the neutral reference and the use of the infinity reference. *Physiological Measurement* 26: 173-184 (2005)
- 121: Zaidi Z.F.: Gender Differences in Human Brain: A Review. *The Open Anatomy Journal* 2: 37-55 (2010)

Anhang A: Bauteile und Bezugsquellen

A1: Bezugsquellen

Abk.	Firma	Kontakt	Homepage
DK	Digi-Key Corporation	701 Brooks Avenue South Thief River Falls, MN 56701 USA	www.digikey.at
RS	RS Components Handelsgesm.b.H.	Albrechtser Straße 11 3950 Gmünd Austria	www.rs-components.at
C	Conrad Electronic GmbH & Co KG	Durisolstraße 2 4600 Wels Austria	www.conrad.at
DC	Digitale Analoge Komponenten West Electronic Vertriebs GmbH	Schallbruch 19-21 42781 Haan Germany	www.dacomwest.com
BT	Bluetechnix Mechatronische Systeme GmbH	Waidhausenstraße 3/19 1140 Wien Austria	www.bluetechnix.at
LO	LeitOn GmbH	Gottlieb-Dunkel-Str. 47-48 12099 Berlin Germany	www.leiton.de

Tabelle 9: Bezugsquellen und Kontaktdaten

A2: Verwendete Bauteile EEG-Verstärker

EEG-Verstärker, 2 Kanäle								
Abk..	Bezeichnung	Typ	Wert	Gehäuse/ Bauform	Daten- blatt	Bezugs- quelle	Stück- Preis	Kosten
-	Platine	43x35mm		2-lagig	-	LO	44,55	44,55
X1	Buchsenleiste		2 x 1		[20]	C	0,085	0,17
JP1	Stiftleiste		3 x 1		[38]	C	0,085	0,255
-	Jumper		2 x 1		[27]	RS	0,188	0,376
X2	Buchsenleiste		7 x 1		[20]	C	0,085	0,595
Vorverstärker								
IC1	Instrumenten- verstärker	AD8553	-	10-Pin MSOP	[32]	DK	3,30	6,60
IC3	Operations- verstärker	LMP2012	-	8-Pin MSOP	[31]	RS	3,48	3,48
C1	Kondensator	Keramik	0,1 μ F	SMD 0805	[27]	RS	0,141	0,282
C2			1,2nF		[28]	DK	0,13	0,26
C3			1 μ F		[27]	RS	0,11	0,22
R1	Widerstand	Chip	8,2 k Ω	SMD 0805	[23]	RS	0,864	1,728
R2			8,2 k Ω				0,864	1,728
R3			100 k Ω				0,864	1,728
R4			1 M Ω				0,328	0,656
R5			10 k Ω				0,328	0,656
1.Verstärkerstufe								
IC2	Operations- verstärker	LMP2012	-	8-Pin MSOP	[31]	RS	3,48	6,96
C4	Kondensator	Keramik	1nF	SMD 0805	[27]	RS	0,108	0,216
C5			1 μ F				0,11	0,22
R6	Widerstand	Chip	100 k Ω	SMD 0805	[23]	RS	0,864	1,728
R7			4,02 k Ω				0,318	0,636
R8			1 M Ω				0,328	0,656
2.Verstärkerstufe und Filter								
C6	Kondensator	Keramik	220 nF	SMD 0805	[27]	RS	0,179	0,358
C7			33 nF				0,113	0,226
R9	Widerstand	Chip	15 k Ω	SMD 0805	[23]	RS	0,864	1,728
R10			100 k Ω				0,864	1,728
R11			8,2 k Ω				0,864	1,728
R12			100 k Ω				0,864	1,728

Tabelle 10: Bauteile des EEG-Verstärkers

A3: Verwendete Bauteile Motherboard

Motherboard							
Abk.	Bezeichnung	Typ	Wert	Gehäuse/ Bauform	Daten- blatt	Bezugs- quelle	Kosten
-	Platine	80x51mm		2-lagig	-	LO	45,34
Spannungsversorgung							
X1	Buchsenleiste		3 x 1		[20]	C	0,255
IC1	Spannungs- Regulator	TPS7A4533		DCQ	[36]	DK	3,490
C1	Kondensator	Aluminium- Elektrolyt	10 μ F	SMD	[18]	RS	0,192
C2			10 μ F	Case A			0,192
C3		Keramik	0,1 μ F	SMD 0805	[27]	RS	0,141
C4			0,1 μ F				0,141
R1	Widerstand	Chip	100 k Ω	SMD 0805	[23]	RS	0,864
R2			100 k Ω				0,864
Prozessorschnittstelle							
X2	Steck- verbinder	Hirose		FX8-60S-SV	[37]	RS	3,880
X3		FX8					3,880
Reset	Taster	SMD		DTSMG-6	[35]	C	1,590
Boot	Tip Switch	4-polig		GDH04S	[39]	DK	2,010
C5	Kondensator	AI-ELKO	10 μ F	Case A	[18]	RS	0,141
C6		Keramik	1 μ F	SMD 0805	[27]		0,110
R3 - R6	Widerstände	Chip	220 Ω	SMD 0805	[23]	RS	1,504
JTAG							
X4	Stiflleiste		2 x 7		[38]	C	1,190
R7	Widerstand	Chip	4,7 k Ω	SMD 0805	[23]	RS	0,328
UART							
X5	Buchsenleiste		1 x 4		[20]	C	0,340
R8	Widerstand		10 k Ω	SMD 0805	[23]	RS	0,328
ADC							
IC2	Analog-Digital Wandler	ADS8345		SSOP-20	[19]	RS	19,670
C7	Kondensator	AI-ELKO	10 μ F	SMD Case A	[18]	RS	0,192

Tabelle 11: Bauteile des Motherboards Teil 1

Prototypentwicklung eines EEG-Headsets

Motherboard							
Abk.	Bezeichnung	Typ	Wert	Gehäuse/ Bauform	Daten- blatt	Bezugs- quelle	Kosten
DRL-Driver							
X6	Buchsenleiste		1 x 2		[20]	C	0,170
IC3	Operations- verstärker	LMP2012		8-Pin MSOP	[31]	RS	3,480
C8 C9	Kondensator	Keramik	1nF 1nF	SMD 0805	[27]	RS	0,108 0,108
R9	Widerstand	Chip	200 k Ω	SMD 0805	[22]	RS	0,258
EEG-Interface							
X7 X8	Buchsenleiste		2 x 7		[20]	C	1,190 1,190
USB							
X9	USB-Buchse	Molex		Mini-USB-B	[40]	RS	0,668
D2	ESD-Diode	USBLC6		SOT-23-6L	[25]	RS	0,886
C10	Kondensator		0,1 μ F		[27]	RS	0,141
R10	Widerstand		27 Ω		[22]	RS	0,284
PPI							
X10	Buchsenleiste		2 x 6		[20]	C	0,510
R11 - R18	Widerstand	Chip	220 Ω	SMD 0805	[23]	RS	3,008
LED1- LED8	Leuchtdiode	Grün		SMD 0805	[21]	RS	2,760
SD-Card							
X10	Connector	Micro-SD		DM3AT	[34]	DK	2,460
D3	ESD-Diode	USBLC6		SOT-23-6L	[25]	RS	0,886
C11	Kondensator		0,1 μ F		[27]	RS	0,141
R19 R20	Widerstand		10 k Ω	SMD 0805	[23]	RS	0,328 0,328
Ethernet							
X11	RJ45-Buchse			RJLBC	[26]	DC	3,500
D4	ESD-Diode	USBLC6		SOT-23-6L	[25]	RS	0,886
C12 C13	Kondensator	Keramik	0,1 μ F 0,1 μ F	SMD 0805	[27]	RS	0,141 0,141
R21 R22 R23 R24	Widerstand	Chip	27 Ω 27 Ω 220 Ω 220 Ω	SMD 0805	[22] [23]	RS	0,284 0,284 0,376 0,376

Tabelle 12: Bauteile des Motherboards Teil 2

A4: Kostenaufstellung

Kostenaufstellung EEG-Headset			
Baugruppe	Hersteller	Preis	Bezugsquelle
Verstärker-board	Hagauer	€ 81,20	-
Mother-board	Hagauer	€ 111,53	-
DSP Modul CM-BF527	Analog-Device	€ 221,00	BT
DSP-UART Adapter	C	€ 14,95	C
Akku	Pichler	€ 29,95	C
Trägersystem	Hagauer	€ 10,00	-
Diverses	-	€ 5,00	-
Gesamtkosten von:		€ 473,63	

Tabelle 13: Kostenaufstellung

Anhang B: Schaltpläne

B1: EEG-Verstärker

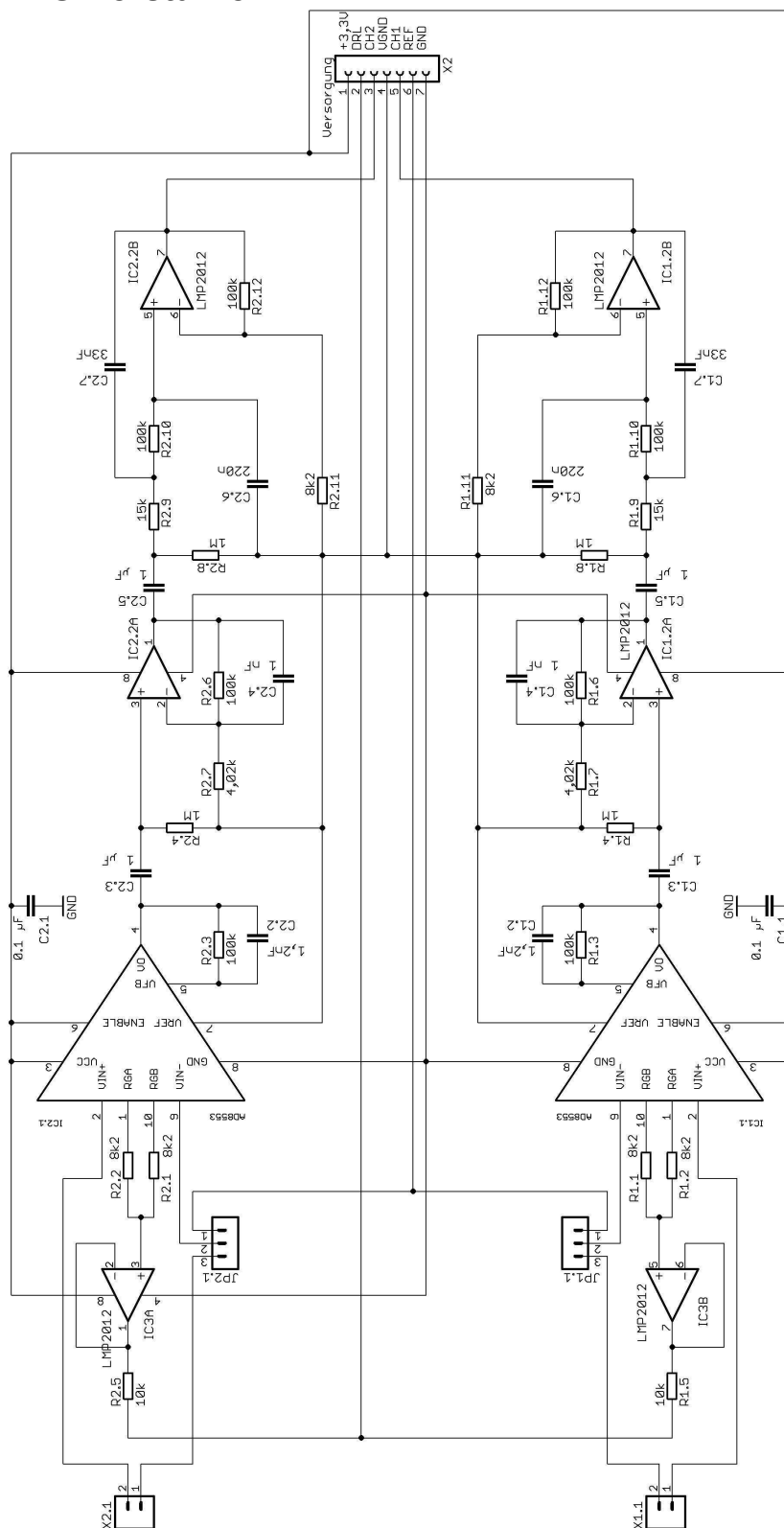


Abbildung 73: Schaltung EEG-Verstärker

B2: Motherboard Teil1

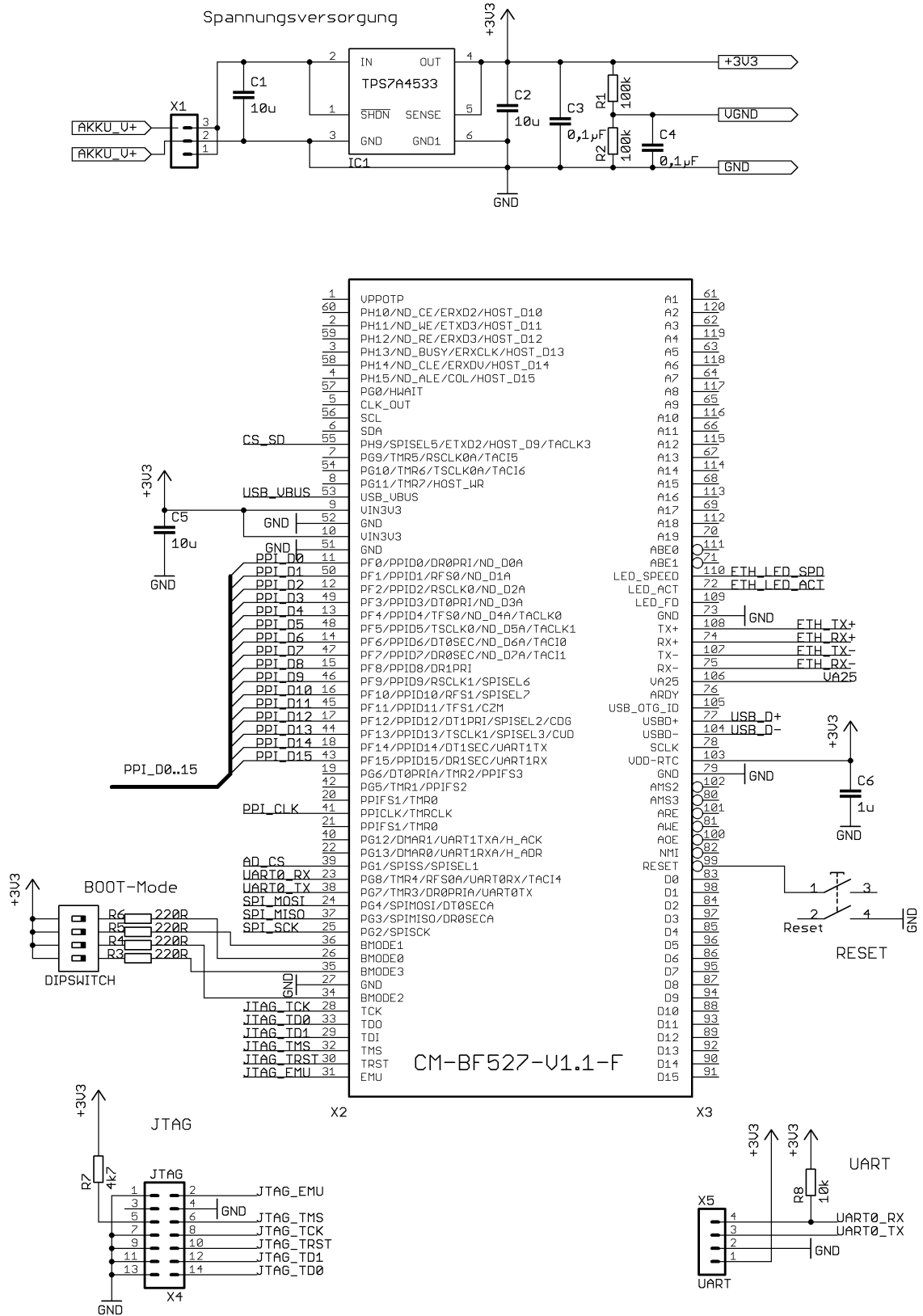


Abbildung 74: Schaltung Motherboard Teil 1

B3: Motheboard Teil 2

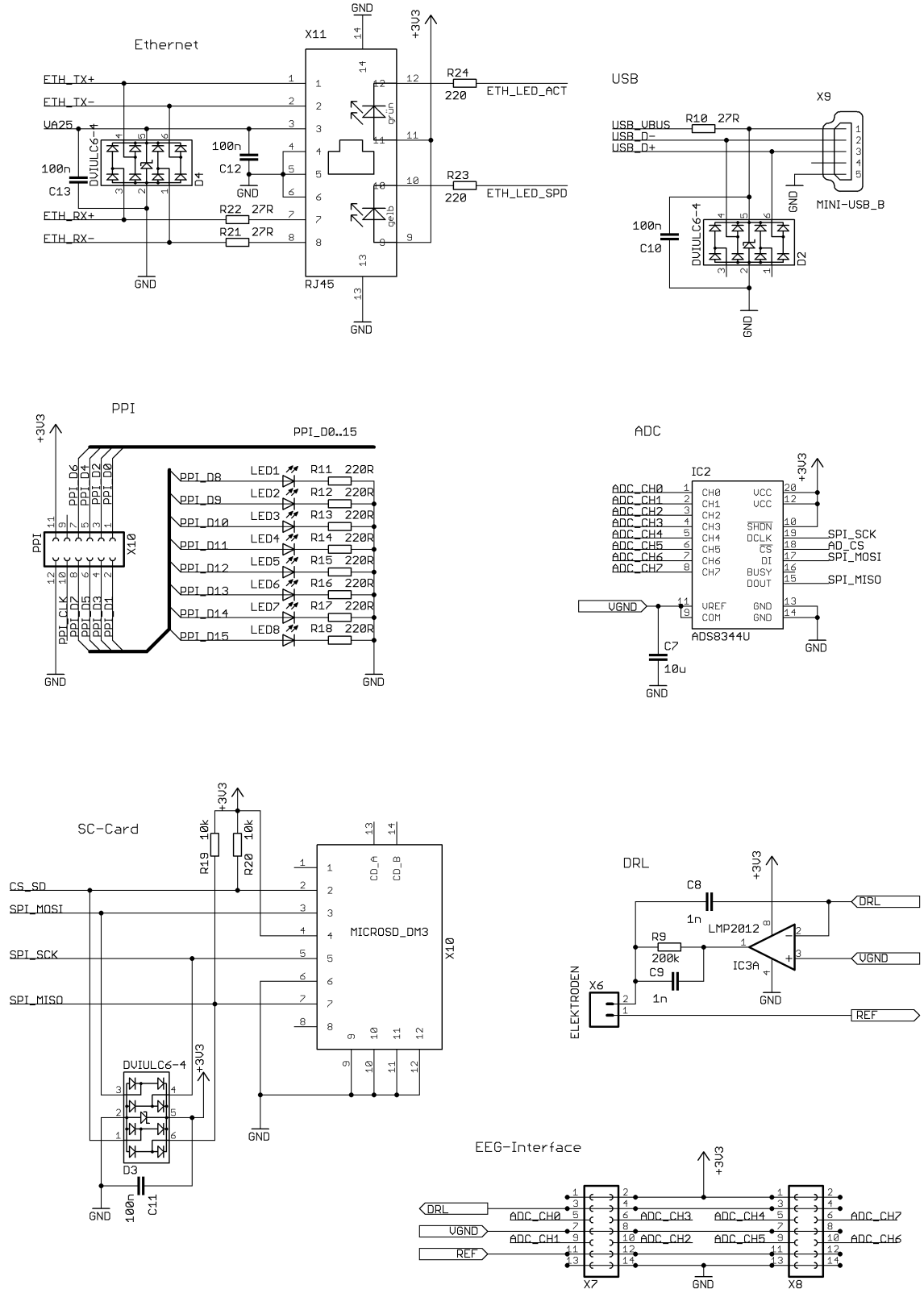


Abbildung 75: Schaltung Motherboard Teil 2

Anhang C: uClinux Konfiguration

Im folgenden die Konfiguration zur eventuellen Systemwiederherstellung. Bei Bedarf kopieren und in einem Textfile mit dem Namen .config ablegen und an die entsprechende Stelle kopieren. Mit make wird die Konfiguration anschließend übernommen.

C1: Kernel-Konfiguration

```
#
# Automatically generated make config: don't edit
# Linux kernel version: 2.6.34.7
# Wed Nov 9 11:55:30 2011
#
CONFIG_SYMBOL_PREFIX=""
# CONFIG_MMU is not set
# CONFIG_FPU is not set
CONFIG_RWSEM_GENERIC_SPINLOCK=y
# CONFIG_RWSEM_XCHGADD_ALGORITHM is not set
CONFIG_BLACKFIN=y
CONFIG_GENERIC_CSUM=y
CONFIG_GENERIC_BUG=y
CONFIG_ZONE_DMA=y
CONFIG_GENERIC_FIND_NEXT_BIT=y
CONFIG_GENERIC_HARDIRQS=y
CONFIG_GENERIC_IRQ_PROBE=y
CONFIG_GENERIC_HARDIRQS_NO_DO_IRQ=y
CONFIG_GENERIC_GPIO=y
CONFIG_FORCE_MAX_ZONEORDER=14
CONFIG_GENERIC_CALIBRATE_DELAY=y
CONFIG_LOCKDEP_SUPPORT=y
CONFIG_STACKTRACE_SUPPORT=y
CONFIG_TRACE_IRQFLAGS_SUPPORT=y
CONFIG_DEFCONFIG_LIST="/lib/modules/$UNAME_RELEASE/.config"
CONFIG_CONSTRUCTORS=y

#
# General setup
#
CONFIG_EXPERIMENTAL=y
CONFIG_BROKEN_ON_SMP=y
CONFIG_INIT_ENV_ARG_LIMIT=32
CONFIG_LOCALVERSION=""
CONFIG_LOCALVERSION_AUTO=y
CONFIG_HAVE_KERNEL_GZIP=y
CONFIG_HAVE_KERNEL_BZIP2=y
CONFIG_HAVE_KERNEL_LZMA=y
CONFIG_HAVE_KERNEL_LZO=y
CONFIG_KERNEL_GZIP=y
# CONFIG_KERNEL_BZIP2 is not set
# CONFIG_KERNEL_LZMA is not set
# CONFIG_KERNEL_LZO is not set
CONFIG_SYSVIPC=y
CONFIG_SYSVIPC_SYSCTL=y
# CONFIG_BSD_PROCESS_ACCT is not set

#
# RCU Subsystem
#
CONFIG_TREE_RCU=y
# CONFIG_TREE_PREEMPT_RCU is not set
# CONFIG_TINY_RCU is not set
# CONFIG_RCU_TRACE is not set
CONFIG_RCU_FANOUT=32
# CONFIG_RCU_FANOUT_EXACT is not set
# CONFIG_TREE_RCU_TRACE is not set
CONFIG_IKCONFIG=y
CONFIG_IKCONFIG_PROC=y
CONFIG_LOG_BUF_SHIFT=14
# CONFIG_CGROUPS is not set
# CONFIG_SYSFS_DEPRECATED_V2 is not set
# CONFIG_RELAY is not set
# CONFIG_NAMESPACES is not set
CONFIG_BLK_DEV_INITRD=y
CONFIG_INITRAMFS_SOURCE=""
# CONFIG_RD_GZIP is not set
# CONFIG_RD_BZIP2 is not set
CONFIG_RD_LZMA=y
# CONFIG_RD_LZO is not set
CONFIG_CC_OPTIMIZE_FOR_SIZE=y
CONFIG_SYSCTL=y
CONFIG_ANON_INODES=y
CONFIG_EMBEDDED=y
CONFIG_UID16=y
# CONFIG_SYSCTL_SYSCALL is not set
CONFIG_KALLSYMS=y
# CONFIG_KALLSYMS_EXTRA_PASS is not set
CONFIG_HOTPLUG=y
CONFIG_PRINTK=y
CONFIG_BUG=y
CONFIG_ELF_CORE=y
CONFIG_BASE_FULL=y
# CONFIG_FUTEX is not set
CONFIG_EPOLL=y
CONFIG_SIGNALFD=y
CONFIG_TIMERFD=y
CONFIG_EVENTFD=y
CONFIG_AIO=y

#
# Kernel Performance Events And Counters
#
# CONFIG_VM_EVENT_COUNTERS is not set
```


Prototypentwicklung eines EEG-Headsets

```
# CONFIG_COMPAT_BRK is not set
CONFIG_SLAB=y
# CONFIG_SLUB is not set
# CONFIG_SLOB is not set
CONFIG_MMAP_ALLOW_UNINITIALIZED=y
# CONFIG_PROFILING is not set
CONFIG_HAVE_OPROFILE=y
CONFIG_HAVE_ARCH_TRACEHOOK=y

#
# GCOV-based kernel profiling
#
# CONFIG_GCOV_KERNEL is not set
# CONFIG_SLOW_WORK is not set
# CONFIG_HAVE_GENERIC_DMA_COHERENT is not set
CONFIG_SLABINFO=y
CONFIG_BASE_SMALL=0
CONFIG_MODULES=y
# CONFIG_MODULE_FORCE_LOAD is not set
CONFIG_MODULE_UNLOAD=y
# CONFIG_MODULE_FORCE_UNLOAD is not set
# CONFIG_MODVERSIONS is not set
# CONFIG_MODULE_SRCVERSION_ALL is not set
# CONFIG_BLOCK is not set
# CONFIG_INLINE_SPIN_TRYLOCK is not set
# CONFIG_INLINE_SPIN_TRYLOCK_BH is not set
# CONFIG_INLINE_SPIN_LOCK is not set
# CONFIG_INLINE_SPIN_LOCK_BH is not set
# CONFIG_INLINE_SPIN_LOCK_IRQ is not set
# CONFIG_INLINE_SPIN_LOCK_IRQSAVE is not set
CONFIG_INLINE_SPIN_UNLOCK=y
# CONFIG_INLINE_SPIN_UNLOCK_BH is not set
CONFIG_INLINE_SPIN_UNLOCK_IRQ=y
# CONFIG_INLINE_SPIN_UNLOCK_IRQRESTORE is not set
# CONFIG_INLINE_READ_TRYLOCK is not set
# CONFIG_INLINE_READ_LOCK is not set
# CONFIG_INLINE_READ_LOCK_BH is not set
# CONFIG_INLINE_READ_LOCK_IRQ is not set
# CONFIG_INLINE_READ_LOCK_IRQSAVE is not set
CONFIG_INLINE_READ_UNLOCK=y
# CONFIG_INLINE_READ_UNLOCK_BH is not set
CONFIG_INLINE_READ_UNLOCK_IRQ=y
# CONFIG_INLINE_READ_UNLOCK_IRQRESTORE is not set
# CONFIG_INLINE_WRITE_TRYLOCK is not set
# CONFIG_INLINE_WRITE_LOCK is not set
# CONFIG_INLINE_WRITE_LOCK_BH is not set
# CONFIG_INLINE_WRITE_LOCK_IRQ is not set
# CONFIG_INLINE_WRITE_LOCK_IRQSAVE is not set
CONFIG_INLINE_WRITE_UNLOCK=y
# CONFIG_INLINE_WRITE_UNLOCK_BH is not set
CONFIG_INLINE_WRITE_UNLOCK_IRQ=y
# CONFIG_INLINE_WRITE_UNLOCK_IRQRESTORE is not set
# CONFIG_MUTEX_SPIN_ON_OWNER is not set

#
# Real-time sub-system
#
CONFIG_XENOMAI=y
CONFIG_XENO_GENERIC_STACKPOOL=y
CONFIG_XENO_OPT_NUCLEUS=y
CONFIG_XENO_OPT_PERVASIVE=y

CONFIG_XENO_OPT_PRIOCP=y
CONFIG_XENO_OPT_PIPELINE_HEAD=y
# CONFIG_XENO_OPT_SCHED_CLASSES is not set
CONFIG_XENO_OPT_PIPE=y
CONFIG_XENO_OPT_PIPE_NRDEV=32
CONFIG_XENO_OPT_REGISTRY_NRSLOTS=512
CONFIG_XENO_OPT_SYS_HEAPSZ=256
CONFIG_XENO_OPT_SYS_STACKPOOLSZ=128
CONFIG_XENO_OPT_SEM_HEAPSZ=12
CONFIG_XENO_OPT_GLOBAL_SEM_HEAPSZ=12
CONFIG_XENO_OPT_STATS=y
# CONFIG_XENO_OPT_DEBUG is not set
# CONFIG_XENO_OPT_SHIRQ is not set

#
# Timing
#
# CONFIG_XENO_OPT_TIMING_PERIODIC is not set
CONFIG_XENO_OPT_TIMING_VIRTICK=1000
CONFIG_XENO_OPT_TIMING_SCHEDLAT=0

#
# Scalability
#
# CONFIG_XENO_OPT_SCALABLE_SCHED is not set
CONFIG_XENO_OPT_TIMER_LIST=y
# CONFIG_XENO_OPT_TIMER_HEAP is not set
# CONFIG_XENO_OPT_TIMER_WHEEL is not set

#
# Machine
#
#
# NMI watchdog
#
# CONFIG_XENO_HW_NMI_DEBUG_LATENCY is not set

#
# Interfaces
#
CONFIG_XENO_SKIN_NATIVE=y
CONFIG_XENO_OPT_NATIVE_PERIOD=0
CONFIG_XENO_OPT_NATIVE_PIPE=y
CONFIG_XENO_OPT_NATIVE_PIPE_BUFSZ=1024
CONFIG_XENO_OPT_NATIVE_SEM=y
CONFIG_XENO_OPT_NATIVE_EVENT=y
CONFIG_XENO_OPT_NATIVE_MUTEX=y
CONFIG_XENO_OPT_NATIVE_COND=y
CONFIG_XENO_OPT_NATIVE_QUEUE=y
CONFIG_XENO_OPT_NATIVE_BUFFER=y
CONFIG_XENO_OPT_NATIVE_HEAP=y
CONFIG_XENO_OPT_NATIVE_ALARM=y
CONFIG_XENO_OPT_NATIVE_MPS=y
# CONFIG_XENO_OPT_NATIVE_INTR is not set
CONFIG_XENO_SKIN_POSIX=y
CONFIG_XENO_OPT_POSIX_PERIOD=0
# CONFIG_XENO_OPT_POSIX_SHM is not set
# CONFIG_XENO_OPT_POSIX_INTR is not set
# CONFIG_XENO_OPT_POSIX_SELECT is not set
# CONFIG_XENO_OPT_DEBUG_POSIX is not set
```

Prototypentwicklung eines EEG-Headsets

```
# CONFIG_XENO_SKIN_PSOS is not set
# CONFIG_XENO_SKIN_UITRON is not set
# CONFIG_XENO_SKIN_VRTX is not set
# CONFIG_XENO_SKIN_VXWORKS is not set
# CONFIG_XENO_SKIN_RTAI is not set
# CONFIG_XENO_OPT_NOWARN_DEPRECATED is not set
CONFIG_XENO_SKIN_RTDM=y
CONFIG_XENO_OPT_RTDM_PERIOD=0
CONFIG_XENO_OPT_RTDM_FILDES=128
# CONFIG_XENO_OPT_RTDM_SELECT is not set

#
# Drivers
#

#
# Serial drivers
#
# CONFIG_XENO_DRIVERS_16550A is not set

#
# Testing drivers
#
# CONFIG_XENO_DRIVERS_TESTING_LEGACY_NAMES is not set
CONFIG_XENO_DRIVERS_TIMERBENCH=y
# CONFIG_XENO_DRIVERS_KLATENCY is not set
# CONFIG_XENO_DRIVERS_IRQBENCH is not set
CONFIG_XENO_DRIVERS_SWITCHTEST=y
# CONFIG_XENO_DRIVERS_SIGTEST is not set
# CONFIG_XENO_DRIVERS_RTDMTEST is not set

#
# CAN drivers
#
# CONFIG_XENO_DRIVERS_CAN is not set

#
# ANALOGY drivers
#
# CONFIG_XENO_DRIVERS_ANALOGY is not set

#
# Real-time IPC drivers
#
# CONFIG_XENO_DRIVERS_RTIPC is not set
# CONFIG_PREEMPT_NONE is not set
CONFIG_PREEMPT_VOLUNTARY=y
# CONFIG_PREEMPT is not set
# CONFIG_FREEZER is not set

#
# Blackfin Processor Options
#
CONFIG_IPIPE=y
CONFIG_IPIPE_DOMAINS=4
CONFIG_IPIPE_DELAYED_ATOMICSW=y
# CONFIG_IPIPE_UNMASKED_CONTEXT_SWITCH is not set

#
# Processor and Board Settings
#

# CONFIG_BF512 is not set
# CONFIG_BF514 is not set
# CONFIG_BF516 is not set
# CONFIG_BF518 is not set
# CONFIG_BF522 is not set
# CONFIG_BF523 is not set
# CONFIG_BF524 is not set
# CONFIG_BF525 is not set
# CONFIG_BF526 is not set
CONFIG_BF527=y
# CONFIG_BF531 is not set
# CONFIG_BF532 is not set
# CONFIG_BF533 is not set
# CONFIG_BF534 is not set
# CONFIG_BF536 is not set
# CONFIG_BF537 is not set
# CONFIG_BF538 is not set
# CONFIG_BF539 is not set
# CONFIG_BF542_std is not set
# CONFIG_BF542M is not set
# CONFIG_BF544_std is not set
# CONFIG_BF544M is not set
# CONFIG_BF547_std is not set
# CONFIG_BF547M is not set
# CONFIG_BF548_std is not set
# CONFIG_BF548M is not set
# CONFIG_BF549_std is not set
# CONFIG_BF549M is not set
# CONFIG_BF561 is not set
CONFIG_BF_REV_MIN=0
CONFIG_BF_REV_MAX=2
# CONFIG_BF_REV_0_0 is not set
CONFIG_BF_REV_0_1=y
# CONFIG_BF_REV_0_2 is not set
# CONFIG_BF_REV_0_3 is not set
# CONFIG_BF_REV_0_4 is not set
# CONFIG_BF_REV_0_5 is not set
# CONFIG_BF_REV_0_6 is not set
# CONFIG_BF_REV_ANY is not set
# CONFIG_BF_REV_NONE is not set
CONFIG_MEM_MT48LC16M16A2TG_75=y
# CONFIG_BFIN_HYSTERESIS_CONTROL is not set
CONFIG_IRQ_PLL_WAKEUP=7
CONFIG_IRQ_DMA0_ERROR=7
CONFIG_IRQ_DMAR0_BLK=7
CONFIG_IRQ_DMAR1_BLK=7
CONFIG_IRQ_DMAR0_OVR=7
CONFIG_IRQ_DMAR1_OVR=7
CONFIG_IRQ_PPI_ERROR=7
CONFIG_IRQ_MAC_ERROR=7
CONFIG_IRQ_SPORT0_ERROR=7
CONFIG_IRQ_SPORT1_ERROR=7
CONFIG_IRQ_UART0_ERROR=7
CONFIG_IRQ_UART1_ERROR=7
CONFIG_IRQ_RTC=8
CONFIG_IRQ_PPI=8
CONFIG_IRQ_SPORT0_RX=9
CONFIG_IRQ_SPORT0_TX=9
CONFIG_IRQ_SPORT1_RX=9
CONFIG_IRQ_SPORT1_TX=9
CONFIG_IRQ_TWI=10
```

Prototypentwicklung eines EEG-Headsets

```
CONFIG_IRQ_UART0_RX=10
CONFIG_IRQ_UART0_TX=10
CONFIG_IRQ_UART1_RX=10
CONFIG_IRQ_UART1_TX=10
CONFIG_IRQ_OPTSEC=11
CONFIG_IRQ_CNT=11
CONFIG_IRQ_MAC_RX=11
CONFIG_IRQ_PORTH_INTA=11
CONFIG_IRQ_MAC_TX=11
CONFIG_IRQ_PORTH_INTB=11
CONFIG_IRQ_TIMER0=12
CONFIG_IRQ_TIMER1=12
CONFIG_IRQ_TIMER2=12
CONFIG_IRQ_TIMER3=12
CONFIG_IRQ_TIMER4=12
CONFIG_IRQ_TIMER5=12
CONFIG_IRQ_TIMER6=12
CONFIG_IRQ_TIMER7=12
CONFIG_IRQ_PORTG_INTA=12
CONFIG_IRQ_PORTG_INTB=12
CONFIG_IRQ_MEM_DMA0=13
CONFIG_IRQ_MEM_DMA1=13
CONFIG_IRQ_WATCH=13
CONFIG_IRQ_PORTF_INTA=13
CONFIG_IRQ_PORTF_INTB=13
CONFIG_BF52x=y
# CONFIG_BFIN527_EZKIT is not set
# CONFIG_BFIN527_EZKIT_V2 is not set
CONFIG_BFIN527_BLUETECHNIX_CM=y
# CONFIG_BFIN526_EZBRD is not set
# CONFIG_BFIN527_AD7160EVAL is not set
# CONFIG_BFIN527_TLL6527M is not set

#
# BF527 Specific Configuration
#

#
# Alternative Multiplexing Scheme
#
# CONFIG_BF527_SPORT0_PORTF is not set
CONFIG_BF527_SPORT0_PORTG=y
CONFIG_BF527_SPORT0_TSCLK_PG10=y
# CONFIG_BF527_SPORT0_TSCLK_PG14 is not set
CONFIG_BF527_UART1_PORTF=y
# CONFIG_BF527_UART1_PORTG is not set
# CONFIG_BF527_NAND_D_PORTF is not set
CONFIG_BF527_NAND_D_PORTH=y

#
# Hysteresis/schmitt Trigger Control
#

#
# Interrupt Priority Assignment
#

#
# Priority
#
CONFIG_IRQ_SPI=10

CONFIG_IRQ_SPI_ERROR=7
CONFIG_IRQ_NFC_ERROR=7
CONFIG_IRQ_HDMA_ERROR=7
CONFIG_IRQ_HDMA=7
CONFIG_IRQ_USB_EINT=10
CONFIG_IRQ_USB_INT0=11
CONFIG_IRQ_USB_INT1=11
CONFIG_IRQ_USB_INT2=11
CONFIG_IRQ_USB_DMA=11

#
# Board customizations
#
# CONFIG_CMDLINE_BOOL is not set
CONFIG_BOOT_LOAD=0x1000

#
# Clock/PLL Setup
#
CONFIG_CLKIN_HZ=25000000
# CONFIG_BFIN_KERNEL_CLOCK is not set
CONFIG_MAX_VCO_HZ=600000000
CONFIG_MIN_VCO_HZ=500000000
CONFIG_MAX_SCLK_HZ=133333333
CONFIG_MIN_SCLK_HZ=27000000

#
# Kernel Timer/Scheduler
#
# CONFIG_HZ_100 is not set
CONFIG_HZ_250=y
# CONFIG_HZ_300 is not set
# CONFIG_HZ_1000 is not set
CONFIG_HZ=250
# CONFIG_SCHED_HRTICK is not set
CONFIG_GENERIC_TIME=y
CONFIG_GENERIC_CLOCKEVENTS=y

#
# Clock event device
#
# CONFIG_TICKSOURCE_GPTMR0 is not set
CONFIG_TICKSOURCE_CORETMR=y

#
# Clock souce
#
# CONFIG_CYCLES_CLOCKSOURCE is not set
# CONFIG_GPTMR0_CLOCKSOURCE is not set
# CONFIG_NO_HZ is not set
# CONFIG_HIGH_RES_TIMERS is not set
CONFIG_GENERIC_CLOCKEVENTS_BUILD=y

#
# Misc
#
CONFIG_BFIN_SCRATCH_REG_RETN=y
# CONFIG_BFIN_SCRATCH_REG_RETE is not set
# CONFIG_BFIN_SCRATCH_REG_CYCLES is not set

#
```

Prototypentwicklung eines EEG-Headsets

```
# Blackfin Kernel Optimizations
#
#
# Memory Optimizations
#
CONFIG_I_ENTRY_L1=y
CONFIG_EXCPT_IRQ_SYSC_L1=y
CONFIG_DO_IRQ_L1=y
CONFIG_CORE_TIMER_IRQ_L1=y
CONFIG_IDLE_L1=y
# CONFIG_SCHEDULE_L1 is not set
CONFIG_ARITHMETIC_OPS_L1=y
CONFIG_ACCESS_OK_L1=y
# CONFIG_MEMSET_L1 is not set
# CONFIG_MEMCPY_L1 is not set
# CONFIG_STRCMP_L1 is not set
# CONFIG_STRNCMP_L1 is not set
# CONFIG_STRCPY_L1 is not set
# CONFIG_STRNCPY_L1 is not set
# CONFIG_SYS_BFIN_SPINLOCK_L1 is not set
# CONFIG_IP_CHECKSUM_L1 is not set
CONFIG_CACHELINE_ALIGNED_L1=y
# CONFIG_SYSCALL_TAB_L1 is not set
# CONFIG_CPLB_SWITCH_TAB_L1 is not set
# CONFIG_CACHE_FLUSH_L1 is not set
# CONFIG_EXCEPTION_L1_SCRATCH is not set

#
# Speed Optimizations
#
CONFIG_BFIN_INS_LOWOVERHEAD=y
CONFIG_RAMKERNEL=y
# CONFIG_ROMKERNEL is not set
CONFIG_SELECT_MEMORY_MODEL=y
CONFIG_FLATMEM_MANUAL=y
# CONFIG_DISCONTIGMEM_MANUAL is not set
# CONFIG_SPARSEMEM_MANUAL is not set
CONFIG_FLATMEM=y
CONFIG_FLAT_NODE_MEM_MAP=y
CONFIG_PAGEFLAGS_EXTENDED=y
CONFIG_SPLIT_PTLOCK_CPUS=4
# CONFIG_PHYS_ADDR_T_64BIT is not set
CONFIG_ZONE_DMA_FLAG=1
CONFIG_VIRT_TO_BUS=y
CONFIG_NOMMU_INITIAL_TRIM_EXCESS=0
CONFIG_BFIN_GPTIMERS=y
# CONFIG_DMA_UNCACHED_4M is not set
# CONFIG_DMA_UNCACHED_2M is not set
CONFIG_DMA_UNCACHED_1M=y
# CONFIG_DMA_UNCACHED_512K is not set
# CONFIG_DMA_UNCACHED_256K is not set
# CONFIG_DMA_UNCACHED_128K is not set
# CONFIG_DMA_UNCACHED_NONE is not set

#
# Cache Support
#
CONFIG_BFIN_ICACHE=y
CONFIG_BFIN_EXTMEM_ICACHEABLE=y
CONFIG_BFIN_DCACHE=y

# CONFIG_BFIN_DCACHE_BANKA is not set
CONFIG_BFIN_EXTMEM_DCACHEABLE=y
CONFIG_BFIN_EXTMEM_WRITEBACK=y
# CONFIG_BFIN_EXTMEM_WRITETHROUGH is not set

#
# Memory Protection Unit
#
# CONFIG_MPU is not set

#
# Asynchronous Memory Configuration
#
#
# EBIU_AMGCTL Global Control
#
CONFIG_C_AMCKEN=y
CONFIG_C_CDPRIO=y
# CONFIG_C_AMBEN is not set
# CONFIG_C_AMBEN_B0 is not set
# CONFIG_C_AMBEN_B0_B1 is not set
# CONFIG_C_AMBEN_B0_B1_B2 is not set
CONFIG_C_AMBEN_ALL=y

#
# EBIU_AMBCTL Control
#
#
CONFIG_BANK_0=0x7BB0
CONFIG_BANK_1=0x7BB0
CONFIG_BANK_2=0x7BB0
CONFIG_BANK_3=0xFFC0

#
# Bus options (PCI, PCMCIA, EISA, MCA, ISA)
#
# CONFIG_ARCH_SUPPORTS_MSI is not set
# CONFIG_PCCARD is not set

#
# Executable file formats
#
# CONFIG_BINFMT_ELF_FDPIC is not set
CONFIG_BINFMT_FLAT=y
# CONFIG_BINFMT_ZFLAT is not set
# CONFIG_BINFMT_SHARED_FLAT is not set
# CONFIG_HAVE_AOUT is not set
# CONFIG_BINFMT_MISC is not set

#
# Power management options
#
#
# CONFIG_PM is not set
CONFIG_ARCH_SUSPEND_POSSIBLE=y

#
# CPU Frequency scaling
#
#
# CONFIG_CPU_FREQ is not set
# CONFIG_NET is not set
```

Prototypentwicklung eines EEG-Headsets

```
# CONFIG_SERIAL_BFIN_PIO is not set
# Device Drivers
# CONFIG_SERIAL_BFIN_UART0=y
# CONFIG_BFIN_UART0_CTSRTS is not set
# CONFIG_SERIAL_BFIN_UART1 is not set
# CONFIG_SERIAL_CORE=y
# Generic Driver Options
# CONFIG_SERIAL_CORE_CONSOLE=y
# CONFIG_SERIAL_BFIN_SPORT is not set
# CONFIG_SERIAL_TIMBERDALE is not set
CONFIG_UEVENT_HELPER_PATH="/sbin/hotplug"
# CONFIG_SERIAL_PTYS=y
CONFIG_STANDALONE=y
# CONFIG_DEVPTS_MULTIPLE_INSTANCES is not set
CONFIG_PREVENT_FIRMWARE_BUILD=y
# CONFIG_FW_LOADER is not set
# CONFIG_LEGACY_PTYS is not set
# CONFIG_SYS_HYPERVISOR is not set
# CONFIG_BFIN_OTP is not set
# CONFIG_MTD is not set
# CONFIG_IPMI_HANDLER is not set
# CONFIG_PARPORT is not set
# CONFIG_HW_RANDOM is not set
# CONFIG_MISC_DEVICES is not set
# CONFIG_R3964 is not set
CONFIG_HAVE_IDE=y
# CONFIG_TCG_TPM is not set
# CONFIG_I2C is not set
# CONFIG_SPI=y
# CONFIG_SPI_MASTER=y
#
# SPI Master Controller Drivers
#
# CONFIG_SPI_BFIN=y
# CONFIG_SPI_BFIN_LOCK is not set
# CONFIG_SPI_BFIN_SPORT is not set
# CONFIG_SPI_BITBANG is not set
# CONFIG_SPI_GPIO is not set
# CONFIG_SPI_XILINX is not set
# CONFIG_SPI_DESIGNWARE is not set
#
# Input device support
#
# CONFIG_INPUT is not set
#
# Hardware I/O ports
#
# SPI Protocol Masters
#
# CONFIG_SERIO is not set
# CONFIG_GAMEPORT is not set
# CONFIG_SPI_SPIDEV=y
# CONFIG_SPI_TLE62X0 is not set
#
# Character devices
#
# CONFIG_BFIN_DMA_INTERFACE=m
# CONFIG_PPS support
# CONFIG_PPS is not set
# CONFIG_BFIN_PPI is not set
# CONFIG_ARCH_WANT_OPTIONAL_GPIOLIB=y
# CONFIG_BFIN_SIMPLE_TIMER is not set
# CONFIG_GPIOLIB=y
# CONFIG_BFIN_SPI_ADC is not set
# CONFIG_GPIO_SYSFS=y
# CONFIG_BFIN_SPORT is not set
# CONFIG_VT is not set
# CONFIG_DEVKMEM is not set
# CONFIG_BFIN_JTAG_COMM is not set
# Memory mapped GPIO expanders:
#
# CONFIG_SERIAL_NONSTANDARD is not set
# CONFIG_GPIO_IT8761E is not set
#
# Serial drivers
#
# CONFIG_SERIAL_8250 is not set
# I2C GPIO expanders:
#
# Non-8250 serial port support
# PCI GPIO expanders:
#
# CONFIG_SERIAL_MAX3100 is not set
#
# CONFIG_SERIAL_BFIN=y
#
# CONFIG_SERIAL_BFIN_CONSOLE=y
# SPI GPIO expanders:
#
# CONFIG_SERIAL_BFIN_DMA=y
```

Prototypentwicklung eines EEG-Headsets

```
# CONFIG_GPIO_MAX7301 is not set
# CONFIG_GPIO_MCP23S08 is not set
# CONFIG_GPIO_MC33880 is not set

#
# AC97 GPIO expanders:
#
# CONFIG_W1 is not set
# CONFIG_POWER_SUPPLY is not set
# CONFIG_HWMON is not set
# CONFIG_THERMAL is not set
CONFIG_WATCHDOG=y
# CONFIG_WATCHDOG_NOWAYOUT is not set

#
# Watchdog Device Drivers
#
# CONFIG_SOFT_WATCHDOG is not set
CONFIG_BFIN_WDT=y
CONFIG_SSB_POSSIBLE=y

#
# Sonics Silicon Backplane
#
# CONFIG_SSB is not set

#
# Multifunction device drivers
#
# CONFIG_MFD_CORE is not set
# CONFIG_MFD_SM501 is not set
# CONFIG_HTC_PASIC3 is not set
# CONFIG_MFD_TMIO is not set
# CONFIG_MFD_MC13783 is not set
# CONFIG_EZX_PCAP is not set
# CONFIG_AB4500_CORE is not set
# CONFIG_REGULATOR is not set
# CONFIG_MEDIA_SUPPORT is not set

#
# Graphics support
#
# CONFIG_VGASTATE is not set
# CONFIG_VIDEO_OUTPUT_CONTROL is not set
# CONFIG_FB is not set
# CONFIG_BACKLIGHT_LCD_SUPPORT is not set

#
# Display device support
#
# CONFIG_DISPLAY_SUPPORT is not set
# CONFIG_SOUND is not set
# CONFIG_USB_SUPPORT is not set
# CONFIG_MMC is not set
# CONFIG_MEMSTICK is not set
# CONFIG_NEW_LEDS is not set
# CONFIG_ACCESSIBILITY is not set
CONFIG_RTC_LIB=y
CONFIG_RTC_CLASS=y
CONFIG_RTC_HCTOSYS=y
CONFIG_RTC_HCTOSYS_DEVICE="rtc0"

# CONFIG_RTC_DEBUG is not set

#
# RTC interfaces
#
CONFIG_RTC_INTF_SYSFS=y
CONFIG_RTC_INTF_PROC=y
CONFIG_RTC_INTF_DEV=y
# CONFIG_RTC_INTF_DEV_UIE_EMUL is not set
# CONFIG_RTC_DRV_TEST is not set

#
# SPI RTC drivers
#
# CONFIG_RTC_DRV_M41T94 is not set
# CONFIG_RTC_DRV_DS1305 is not set
# CONFIG_RTC_DRV_DS1390 is not set
# CONFIG_RTC_DRV_MAX6902 is not set
# CONFIG_RTC_DRV_R9701 is not set
# CONFIG_RTC_DRV_RS5C348 is not set
# CONFIG_RTC_DRV_DS3234 is not set
# CONFIG_RTC_DRV_PCF2123 is not set

#
# Platform RTC drivers
#
# CONFIG_RTC_DRV_DS1286 is not set
# CONFIG_RTC_DRV_DS1511 is not set
# CONFIG_RTC_DRV_DS1553 is not set
# CONFIG_RTC_DRV_DS1742 is not set
# CONFIG_RTC_DRV_STK17TA8 is not set
# CONFIG_RTC_DRV_M48T86 is not set
# CONFIG_RTC_DRV_M48T35 is not set
# CONFIG_RTC_DRV_M48T59 is not set
# CONFIG_RTC_DRV_MSM6242 is not set
# CONFIG_RTC_DRV_BQ4802 is not set
# CONFIG_RTC_DRV_RP5C01 is not set
# CONFIG_RTC_DRV_V3020 is not set

#
# on-CPU RTC drivers
#
CONFIG_RTC_DRV_BFIN=y
# CONFIG_DMADEVICES is not set
# CONFIG_AUXDISPLAY is not set
# CONFIG_UIO is not set

#
# TI VLYNQ
#
# CONFIG_STAGING is not set

#
# Firmware Drivers
#
# CONFIG_FIRMWARE_MEMMAP is not set

#
# File systems
#
CONFIG_FILE_LOCKING=y
```

Prototypentwicklung eines EEG-Headsets

```
CONFIG_FSNOTIFY=y
# CONFIG_DNOTIFY is not set
CONFIG_INOTIFY=y
CONFIG_INOTIFY_USER=y
# CONFIG_QUOTA is not set
# CONFIG_AUTOFS_FS is not set
# CONFIG_AUTOFS4_FS is not set
# CONFIG_FUSE_FS is not set

#
# Caches
#
# CONFIG_FSCACHE is not set

#
# Pseudo filesystems
#
CONFIG_PROC_FS=y
CONFIG_PROC_SYSCTL=y
CONFIG_SYSFS=y
# CONFIG_HUGETLB_PAGE is not set
# CONFIG_CONFIGFS_FS is not set
CONFIG_MISC_FILESYSTEMS=y
CONFIG_NLS=y
CONFIG_NLS_DEFAULT="iso8859-1"
CONFIG_NLS_CODEPAGE_437=y
# CONFIG_NLS_CODEPAGE_737 is not set
# CONFIG_NLS_CODEPAGE_775 is not set
# CONFIG_NLS_CODEPAGE_850 is not set
# CONFIG_NLS_CODEPAGE_852 is not set
# CONFIG_NLS_CODEPAGE_855 is not set
# CONFIG_NLS_CODEPAGE_857 is not set
# CONFIG_NLS_CODEPAGE_860 is not set
# CONFIG_NLS_CODEPAGE_861 is not set
# CONFIG_NLS_CODEPAGE_862 is not set
# CONFIG_NLS_CODEPAGE_863 is not set
# CONFIG_NLS_CODEPAGE_864 is not set
# CONFIG_NLS_CODEPAGE_865 is not set
# CONFIG_NLS_CODEPAGE_866 is not set
# CONFIG_NLS_CODEPAGE_869 is not set
# CONFIG_NLS_CODEPAGE_936 is not set
# CONFIG_NLS_CODEPAGE_950 is not set
# CONFIG_NLS_CODEPAGE_932 is not set
# CONFIG_NLS_CODEPAGE_949 is not set
# CONFIG_NLS_CODEPAGE_874 is not set
# CONFIG_NLS_ISO8859_8 is not set
# CONFIG_NLS_CODEPAGE_1250 is not set
# CONFIG_NLS_CODEPAGE_1251 is not set
# CONFIG_NLS_ASCII is not set
CONFIG_NLS_ISO8859_1=y
# CONFIG_NLS_ISO8859_2 is not set
# CONFIG_NLS_ISO8859_3 is not set
# CONFIG_NLS_ISO8859_4 is not set
# CONFIG_NLS_ISO8859_5 is not set
# CONFIG_NLS_ISO8859_6 is not set
# CONFIG_NLS_ISO8859_7 is not set
# CONFIG_NLS_ISO8859_9 is not set
# CONFIG_NLS_ISO8859_13 is not set
# CONFIG_NLS_ISO8859_14 is not set
# CONFIG_NLS_ISO8859_15 is not set
# CONFIG_NLS_KOI8_R is not set

# CONFIG_NLS_KOI8_U is not set
# CONFIG_NLS_UTF8 is not set

#
# Kernel hacking
#
# CONFIG_PRINTK_TIME is not set
CONFIG_ENABLE_WARN_DEPRECATED=y
CONFIG_ENABLE_MUST_CHECK=y
CONFIG_FRAME_WARN=1024
# CONFIG_MAGIC_SYSRQ is not set
# CONFIG_STRIP_ASM_SYMS is not set
# CONFIG_UNUSED_SYMBOLS is not set
CONFIG_DEBUG_FS=y
# CONFIG_HEADERS_CHECK is not set
CONFIG_DEBUG_SECTION_MISMATCH=y
# CONFIG_IPIPE_DEBUG is not set
# CONFIG_DEBUG_KERNEL is not set
CONFIG_DEBUG_BUGVERBOSE=y
# CONFIG_DEBUG_MEMORY_INIT is not set
# CONFIG_RCU_CPU_STALL_DETECTOR is not set
# CONFIG_TRACEHOOK_SELF_TEST is not set
# CONFIG_SYSCTL_SYSCALL_CHECK is not set
CONFIG_HAVE_FUNCTION_TRACER=y
CONFIG_HAVE_FUNCTION_GRAPH_TRACER=y
CONFIG_HAVE_FUNCTION_TRACE_MCOUNT_TEST=y
CONFIG_HAVE_DYNAMIC_FTRACE=y
CONFIG_HAVE_FTRACE_MCOUNT_RECORD=y
CONFIG_TRACING_SUPPORT=y
# CONFIG_FTRACE is not set
# CONFIG_DYNAMIC_DEBUG is not set
# CONFIG_SAMPLES is not set
CONFIG_HAVE_ARCH_KGDB=y
CONFIG_DEBUG_VERBOSE=y
# CONFIG_DEBUG_MMRS is not set
# CONFIG_DEBUG_DOUBLEFAULT is not set
CONFIG_DEBUG_HUNT_FOR_ZERO=y
CONFIG_DEBUG_BFIN_HWTRACE_ON=y
CONFIG_DEBUG_BFIN_HWTRACE_COMPRESSION_OFF=y
# CONFIG_DEBUG_BFIN_HWTRACE_COMPRESSION_ONE is not set
# CONFIG_DEBUG_BFIN_HWTRACE_COMPRESSION_TWO is not set
CONFIG_DEBUG_BFIN_HWTRACE_COMPRESSION=0
# CONFIG_DEBUG_BFIN_HWTRACE_EXPAND is not set
# CONFIG_DEBUG_BFIN_NO_KERN_HWTRACE is not set
CONFIG_EARLY_PRINTK=y
# CONFIG_CPLB_INFO is not set
CONFIG_ACCESS_CHECK=y
# CONFIG_BFIN_JSRAM_SELF_TEST is not set
# CONFIG_BFIN_PSEUDODBG_INSNS is not set

#
# Security options
#
# CONFIG_KEYS is not set
CONFIG_SECURITY=y
# CONFIG_SECURITYFS is not set
# CONFIG_SECURITY_NETWORK is not set
# CONFIG_SECURITY_PATH is not set
# CONFIG_SECURITY_TOMOYO is not set
# CONFIG_DEFAULT_SECURITY_SELINUX is not set
# CONFIG_DEFAULT_SECURITY_SMACK is not set
```

Prototypentwicklung eines EEG-Headsets

```
# CONFIG_DEFAULT_SECURITY_TOMOYO is not set
CONFIG_DEFAULT_SECURITY_DAC=y
CONFIG_DEFAULT_SECURITY=""
CONFIG_CRYPT=y

#
# Crypto core or helper
#
# CONFIG_CRYPT_MANAGER is not set
# CONFIG_CRYPT_MANAGER2 is not set
# CONFIG_CRYPT_GF128MUL is not set
# CONFIG_CRYPT_NULL is not set
# CONFIG_CRYPT_CRYPTD is not set
# CONFIG_CRYPT_AUTHENC is not set
# CONFIG_CRYPT_TEST is not set

#
# Authenticated Encryption with Associated Data
#
# CONFIG_CRYPT_CCM is not set
# CONFIG_CRYPT_GCM is not set
# CONFIG_CRYPT_SEQIV is not set

#
# Block modes
#
# CONFIG_CRYPT_CBC is not set
# CONFIG_CRYPT_CTR is not set
# CONFIG_CRYPT_CTS is not set
# CONFIG_CRYPT_ECB is not set
# CONFIG_CRYPT_LRW is not set
# CONFIG_CRYPT_PCBC is not set
# CONFIG_CRYPT_XTS is not set

#
# Hash modes
#
# CONFIG_CRYPT_HMAC is not set
# CONFIG_CRYPT_XCBC is not set
# CONFIG_CRYPT_VMAC is not set

#
# Digest
#
# CONFIG_CRYPT_CRC32C is not set
# CONFIG_CRYPT_GHASH is not set
# CONFIG_CRYPT_MD4 is not set
# CONFIG_CRYPT_MD5 is not set
# CONFIG_CRYPT_MICHAEL_MIC is not set
# CONFIG_CRYPT_RMD128 is not set
# CONFIG_CRYPT_RMD160 is not set
# CONFIG_CRYPT_RMD256 is not set
# CONFIG_CRYPT_RMD320 is not set
# CONFIG_CRYPT_SHA1 is not set
# CONFIG_CRYPT_SHA256 is not set

# CONFIG_CRYPT_SHA512 is not set
# CONFIG_CRYPT_TGR192 is not set
# CONFIG_CRYPT_WP512 is not set

#
# Ciphers
#
# CONFIG_CRYPT_AES is not set
# CONFIG_CRYPT_ANUBIS is not set
# CONFIG_CRYPT_ARC4 is not set
# CONFIG_CRYPT_BLOWFISH is not set
# CONFIG_CRYPT_CAMELLIA is not set
# CONFIG_CRYPT_CAST5 is not set
# CONFIG_CRYPT_CAST6 is not set
# CONFIG_CRYPT_DES is not set
# CONFIG_CRYPT_FCRYPT is not set
# CONFIG_CRYPT_KHAZAD is not set
# CONFIG_CRYPT_SALSA20 is not set
# CONFIG_CRYPT_SEED is not set
# CONFIG_CRYPT_SERPENT is not set
# CONFIG_CRYPT_TEA is not set
# CONFIG_CRYPT_TWOFISH is not set

#
# Compression
#
# CONFIG_CRYPT_DEFLATE is not set
# CONFIG_CRYPT_ZLIB is not set
# CONFIG_CRYPT_LZO is not set

#
# Random Number Generation
#
# CONFIG_CRYPT_ANSI_CPRNG is not set
CONFIG_CRYPT_HW=y
# CONFIG_BINARY_PRINTF is not set

#
# Library routines
#
CONFIG_BITREVERSE=y
CONFIG_GENERIC_FIND_LAST_BIT=y
CONFIG_CRC_CCITT=m
# CONFIG_CRC16 is not set
# CONFIG_CRC_T10DIF is not set
CONFIG_CRC_ITU_T=y
CONFIG_CRC32=y
CONFIG_CRC7=y
# CONFIG_LIBCRC32C is not set
CONFIG_DECOMPRESS_LZMA=y
CONFIG_HAS_JOMEM=y
CONFIG_HAS_IOPORT=y
CONFIG_HAS_DMA=y
```


C2: uClinux-Konfiguration

```
#
# Automatically generated make config: don't edit
# Linux kernel version: 2.6.34.7
# Wed Nov 9 11:55:30 2011
#
CONFIG_SYMBOL_PREFIX=""
# CONFIG_MMU is not set
# CONFIG_FPU is not set
CONFIG_RWSEM_GENERIC_SPINLOCK=y
# CONFIG_RWSEM_XCHGADD_ALGORITHM is not set
CONFIG_BLACKFIN=y
CONFIG_GENERIC_CSUM=y
CONFIG_GENERIC_BUG=y
CONFIG_ZONE_DMA=y
CONFIG_GENERIC_FIND_NEXT_BIT=y
CONFIG_GENERIC_HARDIRQS=y
CONFIG_GENERIC_IRQ_PROBE=y
CONFIG_GENERIC_HARDIRQS_NO__DO_IRQ=y
CONFIG_GENERIC_GPIO=y
CONFIG_FORCE_MAX_ZONEORDER=14
CONFIG_GENERIC_CALIBRATE_DELAY=y
CONFIG_LOCKDEP_SUPPORT=y
CONFIG_STACKTRACE_SUPPORT=y
CONFIG_TRACE_IRQFLAGS_SUPPORT=y
CONFIG_DEFCONFIG_LIST="/lib/modules/$UNAME_RELEASE/.config"
CONFIG_CONSTRUCTORS=y

#
# General setup
#
CONFIG_EXPERIMENTAL=y
CONFIG_BROKEN_ON_SMP=y
CONFIG_INIT_ENV_ARG_LIMIT=32
CONFIG_LOCALVERSION=""
CONFIG_LOCALVERSION_AUTO=y
CONFIG_HAVE_KERNEL_GZIP=y
CONFIG_HAVE_KERNEL_BZIP2=y
CONFIG_HAVE_KERNEL_LZMA=y
CONFIG_HAVE_KERNEL_LZO=y
CONFIG_KERNEL_GZIP=y
# CONFIG_KERNEL_BZIP2 is not set
# CONFIG_KERNEL_LZMA is not set
# CONFIG_KERNEL_LZO is not set
CONFIG_SYSVIPC=y
CONFIG_SYSVIPC_SYSCTL=y
# CONFIG_BSD_PROCESS_ACCT is not set

#
# RCU Subsystem
#
CONFIG_TREE_RCU=y
# CONFIG_TREE_PREEMPT_RCU is not set
# CONFIG_TINY_RCU is not set
# CONFIG_RCU_TRACE is not set

CONFIG_RCU_FANOUT=32
# CONFIG_RCU_FANOUT_EXACT is not set
# CONFIG_TREE_RCU_TRACE is not set
CONFIG_IKCONFIG=y
CONFIG_IKCONFIG_PROC=y
CONFIG_LOG_BUF_SHIFT=14
# CONFIG_CGROUPS is not set
# CONFIG_SYSFS_DEPRECATED_V2 is not set
# CONFIG_RELAY is not set
# CONFIG_NAMESPACES is not set
CONFIG_BLK_DEV_INITRD=y
CONFIG_INITRAMFS_SOURCE=""
# CONFIG_RD_GZIP is not set
# CONFIG_RD_BZIP2 is not set
CONFIG_RD_LZMA=y
# CONFIG_RD_LZO is not set
CONFIG_CC_OPTIMIZE_FOR_SIZE=y
CONFIG_SYSCTL=y
CONFIG_ANON_INODES=y
CONFIG_EMBEDDED=y
CONFIG_UID16=y
# CONFIG_SYSCTL_SYSCALL is not set
CONFIG_KALLSYMS=y
# CONFIG_KALLSYMS_EXTRA_PASS is not set
CONFIG_HOTPLUG=y
CONFIG_PRINTK=y
CONFIG_BUG=y
CONFIG_ELF_CORE=y
CONFIG_BASE_FULL=y
# CONFIG_FUTEX is not set
CONFIG_EPOLL=y
CONFIG_SIGNALFD=y
CONFIG_TIMERFD=y
CONFIG_EVENTFD=y
CONFIG_AIO=y

#
# Kernel Performance Events And Counters
#
# CONFIG_VM_EVENT_COUNTERS is not set
# CONFIG_COMPAT_BRK is not set
CONFIG_SLAB=y
# CONFIG_SLUB is not set
# CONFIG_SLOB is not set
CONFIG_MMAP_ALLOW_UNINITIALIZED=y
# CONFIG_PROFILING is not set
CONFIG_HAVE_OPROFILE=y
CONFIG_HAVE_ARCH_TRACEHOOK=y

#
# GCOV-based kernel profiling
#
# CONFIG_GCOV_KERNEL is not set
# CONFIG_SLOW_WORK is not set
```

Prototypentwicklung eines EEG-Headsets

```
# CONFIG_HAVE_GENERIC_DMA_COHERENT is not set
CONFIG_SLABINFO=y
CONFIG_BASE_SMALL=0
CONFIG_MODULES=y
# CONFIG_MODULE_FORCE_LOAD is not set
CONFIG_MODULE_UNLOAD=y
# CONFIG_MODULE_FORCE_UNLOAD is not set
# CONFIG_MODVERSIONS is not set
# CONFIG_MODULE_SRCVERSION_ALL is not set
# CONFIG_BLOCK is not set
# CONFIG_INLINE_SPIN_TRYLOCK is not set
# CONFIG_INLINE_SPIN_TRYLOCK_BH is not set
# CONFIG_INLINE_SPIN_LOCK is not set
# CONFIG_INLINE_SPIN_LOCK_BH is not set
# CONFIG_INLINE_SPIN_LOCK_IRQ is not set
# CONFIG_INLINE_SPIN_LOCK_IRQSAVE is not set
CONFIG_INLINE_SPIN_UNLOCK=y
# CONFIG_INLINE_SPIN_UNLOCK_BH is not set
CONFIG_INLINE_SPIN_UNLOCK_IRQ=y
# CONFIG_INLINE_SPIN_UNLOCK_IRQRESTORE is not set
# CONFIG_INLINE_READ_TRYLOCK is not set
# CONFIG_INLINE_READ_LOCK is not set
# CONFIG_INLINE_READ_LOCK_BH is not set
# CONFIG_INLINE_READ_LOCK_IRQ is not set
# CONFIG_INLINE_READ_LOCK_IRQSAVE is not set
CONFIG_INLINE_READ_UNLOCK=y
# CONFIG_INLINE_READ_UNLOCK_BH is not set
CONFIG_INLINE_READ_UNLOCK_IRQ=y
# CONFIG_INLINE_READ_UNLOCK_IRQRESTORE is not set
# CONFIG_INLINE_WRITE_TRYLOCK is not set
# CONFIG_INLINE_WRITE_LOCK is not set
# CONFIG_INLINE_WRITE_LOCK_BH is not set
# CONFIG_INLINE_WRITE_LOCK_IRQ is not set
# CONFIG_INLINE_WRITE_LOCK_IRQSAVE is not set
CONFIG_INLINE_WRITE_UNLOCK=y
# CONFIG_INLINE_WRITE_UNLOCK_BH is not set
CONFIG_INLINE_WRITE_UNLOCK_IRQ=y
# CONFIG_INLINE_WRITE_UNLOCK_IRQRESTORE is not set
# CONFIG_MUTEX_SPIN_ON_OWNER is not set

#
# Real-time sub-system
#
CONFIG_XENOMAI=y
CONFIG_XENO_GENERIC_STACKPOOL=y
CONFIG_XENO_OPT_NUCLEUS=y
CONFIG_XENO_OPT_PERVASIVE=y
CONFIG_XENO_OPT_PRIOCLP=y
CONFIG_XENO_OPT_PIPELINE_HEAD=y
# CONFIG_XENO_OPT_SCHED_CLASSES is not set
CONFIG_XENO_OPT_PIPE=y
CONFIG_XENO_OPT_PIPE_NRDEV=32
CONFIG_XENO_OPT_REGISTRY_NRSLOTS=512
CONFIG_XENO_OPT_SYS_HEAPSZ=256
CONFIG_XENO_OPT_SYS_STACKPOOLSZ=128
CONFIG_XENO_OPT_SEM_HEAPSZ=12

CONFIG_XENO_OPT_GLOBAL_SEM_HEAPSZ=12
CONFIG_XENO_OPT_STATS=y
# CONFIG_XENO_OPT_DEBUG is not set
# CONFIG_XENO_OPT_SHIRQ is not set

#
# Timing
#
# CONFIG_XENO_OPT_TIMING_PERIODIC is not set
CONFIG_XENO_OPT_TIMING_VIRTICK=1000
CONFIG_XENO_OPT_TIMING_SCHEDLAT=0

#
# Scalability
#
# CONFIG_XENO_OPT_SCALABLE_SCHED is not set
CONFIG_XENO_OPT_TIMER_LIST=y
# CONFIG_XENO_OPT_TIMER_HEAP is not set
# CONFIG_XENO_OPT_TIMER_WHEEL is not set

#
# Machine
#

#
# NMI watchdog
#
# CONFIG_XENO_HW_NMI_DEBUG_LATENCY is not set

#
# Interfaces
#
CONFIG_XENO_SKIN_NATIVE=y
CONFIG_XENO_OPT_NATIVE_PERIOD=0
CONFIG_XENO_OPT_NATIVE_PIPE=y
CONFIG_XENO_OPT_NATIVE_PIPE_BUFSZ=1024
CONFIG_XENO_OPT_NATIVE_SEM=y
CONFIG_XENO_OPT_NATIVE_EVENT=y
CONFIG_XENO_OPT_NATIVE_MUTEX=y
CONFIG_XENO_OPT_NATIVE_COND=y
CONFIG_XENO_OPT_NATIVE_QUEUE=y
CONFIG_XENO_OPT_NATIVE_BUFFER=y
CONFIG_XENO_OPT_NATIVE_HEAP=y
CONFIG_XENO_OPT_NATIVE_ALARM=y
CONFIG_XENO_OPT_NATIVE_MPS=y
# CONFIG_XENO_OPT_NATIVE_INTR is not set
CONFIG_XENO_SKIN_POSIX=y
CONFIG_XENO_OPT_POSIX_PERIOD=0
# CONFIG_XENO_OPT_POSIX_SHM is not set
# CONFIG_XENO_OPT_POSIX_INTR is not set
# CONFIG_XENO_OPT_POSIX_SELECT is not set
# CONFIG_XENO_OPT_DEBUG_POSIX is not set
# CONFIG_XENO_SKIN_PSOS is not set
# CONFIG_XENO_SKIN_UITRON is not set
# CONFIG_XENO_SKIN_VRTX is not set
# CONFIG_XENO_SKIN_VXWORKS is not set
```

Prototypentwicklung eines EEG-Headsets

```
# CONFIG_XENO_SKIN_RTAI is not set
# CONFIG_XENO_OPT_NOWARN_DEPRECATED is not set
CONFIG_XENO_SKIN_RTDM=y
CONFIG_XENO_OPT_RTDM_PERIOD=0
CONFIG_XENO_OPT_RTDM_FILDES=128
# CONFIG_XENO_OPT_RTDM_SELECT is not set

#
# Drivers
#

#
# Serial drivers
#
# CONFIG_XENO_DRIVERS_16550A is not set

#
# Testing drivers
#
# CONFIG_XENO_DRIVERS_TESTING_LEGACY_NAMES is not set
CONFIG_XENO_DRIVERS_TIMERBENCH=y
# CONFIG_XENO_DRIVERS_KLATENCY is not set
# CONFIG_XENO_DRIVERS_IRQBENCH is not set
CONFIG_XENO_DRIVERS_SWITCHTEST=y
# CONFIG_XENO_DRIVERS_SIGTEST is not set
# CONFIG_XENO_DRIVERS_RTDMTEST is not set

#
# CAN drivers
#
# CONFIG_XENO_DRIVERS_CAN is not set

#
# ANALOGY drivers
#
# CONFIG_XENO_DRIVERS_ANALOGY is not set

#
# Real-time IPC drivers
#
# CONFIG_XENO_DRIVERS_RTIPC is not set
# CONFIG_PREEMPT_NONE is not set
CONFIG_PREEMPT_VOLUNTARY=y
# CONFIG_PREEMPT is not set
# CONFIG_FREEZER is not set

#
# Blackfin Processor Options
#
CONFIG_IPIPE=y
CONFIG_IPIPE_DOMAINS=4
CONFIG_IPIPE_DELAYED_ATOMICSW=y
# CONFIG_IPIPE_UNMASKED_CONTEXT_SWITCH is not set

#
# Processor and Board Settings
#
```

```
#
# CONFIG_BF512 is not set
# CONFIG_BF514 is not set
# CONFIG_BF516 is not set
# CONFIG_BF518 is not set
# CONFIG_BF522 is not set
# CONFIG_BF523 is not set
# CONFIG_BF524 is not set
# CONFIG_BF525 is not set
# CONFIG_BF526 is not set
CONFIG_BF527=y
# CONFIG_BF531 is not set
# CONFIG_BF532 is not set
# CONFIG_BF533 is not set
# CONFIG_BF534 is not set
# CONFIG_BF536 is not set
# CONFIG_BF537 is not set
# CONFIG_BF538 is not set
# CONFIG_BF539 is not set
# CONFIG_BF542_std is not set
# CONFIG_BF542M is not set
# CONFIG_BF544_std is not set
# CONFIG_BF544M is not set
# CONFIG_BF547_std is not set
# CONFIG_BF547M is not set
# CONFIG_BF548_std is not set
# CONFIG_BF548M is not set
# CONFIG_BF549_std is not set
# CONFIG_BF549M is not set
# CONFIG_BF561 is not set
CONFIG_BF_REV_MIN=0
CONFIG_BF_REV_MAX=2
# CONFIG_BF_REV_0_0 is not set
CONFIG_BF_REV_0_1=y
# CONFIG_BF_REV_0_2 is not set
# CONFIG_BF_REV_0_3 is not set
# CONFIG_BF_REV_0_4 is not set
# CONFIG_BF_REV_0_5 is not set
# CONFIG_BF_REV_0_6 is not set
# CONFIG_BF_REV_ANY is not set
# CONFIG_BF_REV_NONE is not set
CONFIG_MEM_MT48LC16M16A2TG_75=y
# CONFIG_BFIN_HYSTERESIS_CONTROL is not set
CONFIG_IRQ_PLL_WAKEUP=7
CONFIG_IRQ_DMA0_ERROR=7
CONFIG_IRQ_DMAR0_BLK=7
CONFIG_IRQ_DMAR1_BLK=7
CONFIG_IRQ_DMAR0_OVR=7
CONFIG_IRQ_DMAR1_OVR=7
CONFIG_IRQ_PPI_ERROR=7
CONFIG_IRQ_MAC_ERROR=7
CONFIG_IRQ_SPORT0_ERROR=7
CONFIG_IRQ_SPORT1_ERROR=7
CONFIG_IRQ_UART0_ERROR=7
CONFIG_IRQ_UART1_ERROR=7
CONFIG_IRQ_RTC=8
```

Prototypentwicklung eines EEG-Headsets

```
CONFIG_IRQ_PPI=8 # Hysteresis/schmitt Trigger Control
CONFIG_IRQ_SPORT0_RX=9 #
CONFIG_IRQ_SPORT0_TX=9 #
CONFIG_IRQ_SPORT1_RX=9 #
CONFIG_IRQ_SPORT1_TX=9 # Interrupt Priority Assignment
CONFIG_IRQ_TWI=10 #
CONFIG_IRQ_UART0_RX=10 #
CONFIG_IRQ_UART0_TX=10 #
CONFIG_IRQ_UART1_RX=10 # Priority
CONFIG_IRQ_UART1_TX=10 #
CONFIG_IRQ_OPTSEC=11 CONFIG_IRQ_SPI=10
CONFIG_IRQ_CNT=11 CONFIG_IRQ_SPI_ERROR=7
CONFIG_IRQ_MAC_RX=11 CONFIG_IRQ_NFC_ERROR=7
CONFIG_IRQ_PORTH_INTA=11 CONFIG_IRQ_HDMA_ERROR=7
CONFIG_IRQ_MAC_TX=11 CONFIG_IRQ_HDMA=7
CONFIG_IRQ_PORTH_INTB=11 CONFIG_IRQ_USB_EINT=10
CONFIG_IRQ_TIMER0=12 CONFIG_IRQ_USB_INT0=11
CONFIG_IRQ_TIMER1=12 CONFIG_IRQ_USB_INT1=11
CONFIG_IRQ_TIMER2=12 CONFIG_IRQ_USB_INT2=11
CONFIG_IRQ_TIMER3=12 CONFIG_IRQ_USB_DMA=11
CONFIG_IRQ_TIMER4=12
CONFIG_IRQ_TIMER5=12 #
CONFIG_IRQ_TIMER6=12 # Board customizations
CONFIG_IRQ_TIMER7=12 #
CONFIG_IRQ_PORTG_INTA=12 # CONFIG_CMDLINE_BOOL is not set
CONFIG_IRQ_PORTG_INTB=12 CONFIG_BOOT_LOAD=0x1000
CONFIG_IRQ_MEM_DMA0=13
CONFIG_IRQ_MEM_DMA1=13 #
CONFIG_IRQ_WATCH=13 # Clock/PLL Setup
CONFIG_IRQ_PORTF_INTA=13 #
CONFIG_IRQ_PORTF_INTB=13 CONFIG_CLKIN_HZ=25000000
CONFIG_BF52x=y # CONFIG_BFIN_KERNEL_CLOCK is not set
# CONFIG_BFIN527_EZKIT is not set CONFIG_MAX_VCO_HZ=600000000
# CONFIG_BFIN527_EZKIT_V2 is not set CONFIG_MIN_VCO_HZ=500000000
CONFIG_BFIN527_BLUETECHNIX_CM=y CONFIG_MAX_SCLK_HZ=133333333
# CONFIG_BFIN526_EZBRD is not set CONFIG_MIN_SCLK_HZ=27000000
# CONFIG_BFIN527_AD7160EVAL is not set
# CONFIG_BFIN527_TLL6527M is not set #
# # Kernel Timer/Scheduler
# #
# BF527 Specific Configuration # CONFIG_HZ_100 is not set
# # CONFIG_HZ_250=y
# # CONFIG_HZ_300 is not set
# # CONFIG_HZ_1000 is not set
# # Alternative Multiplexing Scheme CONFIG_HZ=250
# # CONFIG_SCHED_HRTICK is not set
# CONFIG_BF527_SPORT0_PORTF is not set CONFIG_GENERIC_TIME=y
CONFIG_BF527_SPORT0_PORTG=y CONFIG_GENERIC_CLOCKEVENTS=y
CONFIG_BF527_SPORT0_TSCLK_PG10=y
# CONFIG_BF527_SPORT0_TSCLK_PG14 is not set #
CONFIG_BF527_UART1_PORTF=y # Clock event device
# CONFIG_BF527_UART1_PORTG is not set #
# CONFIG_BF527_NAND_D_PORTF is not set # CONFIG_TICKSOURCE_GPTMR0 is not set
CONFIG_BF527_NAND_D_PORTH=y CONFIG_TICKSOURCE_CORETMR=y
# #
```

Prototypentwicklung eines EEG-Headsets

```
# Clock source
#
# CONFIG_CYCLES_CLOCKSOURCE is not set
# CONFIG_GPTMR0_CLOCKSOURCE is not set
# CONFIG_NO_HZ is not set
# CONFIG_HIGH_RES_TIMERS is not set
CONFIG_GENERIC_CLOCKEVENTS_BUILD=y

#
# Misc
#
CONFIG_BFIN_SCRATCH_REG_RETN=y
# CONFIG_BFIN_SCRATCH_REG_RETE is not set
# CONFIG_BFIN_SCRATCH_REG_CYCLES is not set

#
# Blackfin Kernel Optimizations
#
#
# Memory Optimizations
#
CONFIG_I_ENTRY_L1=y
CONFIG_EXCPT_IRQ_SYSC_L1=y
CONFIG_DO_IRQ_L1=y
CONFIG_CORE_TIMER_IRQ_L1=y
CONFIG_IDLE_L1=y
# CONFIG_SCHEDULE_L1 is not set
CONFIG_ARITHMETIC_OPS_L1=y
CONFIG_ACCESS_OK_L1=y
# CONFIG_MEMSET_L1 is not set
# CONFIG_MEMCPY_L1 is not set
# CONFIG_STRCMP_L1 is not set
# CONFIG_STRNCMP_L1 is not set
# CONFIG_STRCPY_L1 is not set
# CONFIG_STRNCPY_L1 is not set
# CONFIG_SYS_BFIN_SPINLOCK_L1 is not set
# CONFIG_IP_CHECKSUM_L1 is not set
CONFIG_CACHELINE_ALIGNED_L1=y
# CONFIG_SYSCALL_TAB_L1 is not set
# CONFIG_CPLB_SWITCH_TAB_L1 is not set
# CONFIG_CACHE_FLUSH_L1 is not set
# CONFIG_EXCEPTION_L1_SCRATCH is not set

#
# Speed Optimizations
#
CONFIG_BFIN_INS_LOWOVERHEAD=y
CONFIG_RAMKERNEL=y
# CONFIG_ROMKERNEL is not set
CONFIG_SELECT_MEMORY_MODEL=y
CONFIG_FLATMEM_MANUAL=y
# CONFIG_DISCONTIGMEM_MANUAL is not set
# CONFIG_SPARSEMEM_MANUAL is not set
CONFIG_FLATMEM=y
CONFIG_FLAT_NODE_MEM_MAP=y

CONFIG_PAGEFLAGS_EXTENDED=y
CONFIG_SPLIT_PTLOCK_CPUS=4
# CONFIG_PHYS_ADDR_T_64BIT is not set
CONFIG_ZONE_DMA_FLAG=1
CONFIG_VIRT_TO_BUS=y
CONFIG_NOMMU_INITIAL_TRIM_EXCESS=0
CONFIG_BFIN_GPTIMERS=y
# CONFIG_DMA_UNCACHED_4M is not set
# CONFIG_DMA_UNCACHED_2M is not set
CONFIG_DMA_UNCACHED_1M=y
# CONFIG_DMA_UNCACHED_512K is not set
# CONFIG_DMA_UNCACHED_256K is not set
# CONFIG_DMA_UNCACHED_128K is not set
# CONFIG_DMA_UNCACHED_NONE is not set

#
# Cache Support
#
CONFIG_BFIN_ICACHE=y
CONFIG_BFIN_EXTMEM_ICACHEABLE=y
CONFIG_BFIN_DCACHE=y
# CONFIG_BFIN_DCACHE_BANKA is not set
CONFIG_BFIN_EXTMEM_DCACHEABLE=y
CONFIG_BFIN_EXTMEM_WRITEBACK=y
# CONFIG_BFIN_EXTMEM_WRITETHROUGH is not set

#
# Memory Protection Unit
#
# CONFIG_MPU is not set

#
# Asynchronous Memory Configuration
#
#
# EBIU_AMGCTL Global Control
#
CONFIG_C_AMCKEN=y
CONFIG_C_CDPRIO=y
# CONFIG_C_AMBEN is not set
# CONFIG_C_AMBEN_B0 is not set
# CONFIG_C_AMBEN_B0_B1 is not set
# CONFIG_C_AMBEN_B0_B1_B2 is not set
CONFIG_C_AMBEN_ALL=y

#
# EBIU_AMBCTL Control
#
CONFIG_BANK_0=0x7BB0
CONFIG_BANK_1=0x7BB0
CONFIG_BANK_2=0x7BB0
CONFIG_BANK_3=0xFFC0

#
# Bus options (PCI, PCMCIA, EISA, MCA, ISA)
```

Prototypentwicklung eines EEG-Headsets

```
#
# CONFIG_ARCH_SUPPORTS_MSI is not set
# CONFIG_PCCARD is not set

#
# Executable file formats
#
# CONFIG_BINFMT_ELF_FDPIC is not set
CONFIG_BINFMT_FLAT=y
# CONFIG_BINFMT_ZFLAT is not set
# CONFIG_BINFMT_SHARED_FLAT is not set
# CONFIG_HAVE_AOUT is not set
# CONFIG_BINFMT_MISC is not set

#
# Power management options
#
# CONFIG_PM is not set
CONFIG_ARCH_SUSPEND_POSSIBLE=y

#
# CPU Frequency scaling
#
# CONFIG_CPU_FREQ is not set
# CONFIG_NET is not set

#
# Device Drivers
#
#
# Generic Driver Options
#
CONFIG_UEVENT_HELPER_PATH="/sbin/hotplug"
CONFIG_STANDALONE=y
CONFIG_PREVENT_FIRMWARE_BUILD=y
# CONFIG_FW_LOADER is not set
# CONFIG_SYS_HYPERVISOR is not set
# CONFIG_MTD is not set
# CONFIG_PARPORT is not set
# CONFIG_MISC_DEVICES is not set
CONFIG_HAVE_IDE=y

#
# SCSI device support
#
CONFIG_SCSI_MOD=y
# CONFIG_SCSI_DMA is not set
# CONFIG_SCSI_NETLINK is not set
# CONFIG_PHONE is not set

#
# Input device support
#
# CONFIG_INPUT is not set

#
# Hardware I/O ports
#
# CONFIG_SERIO is not set
# CONFIG_GAMEPORT is not set

#
# Character devices
#
CONFIG_BFIN_DMA_INTERFACE=m
# CONFIG_BFIN_PPI is not set
# CONFIG_BFIN_SIMPLE_TIMER is not set
# CONFIG_BFIN_SPI_ADC is not set
# CONFIG_BFIN_SPORT is not set
# CONFIG_VT is not set
# CONFIG_DEVKMEM is not set
# CONFIG_BFIN_JTAG_COMM is not set
# CONFIG_SERIAL_NONSTANDARD is not set

#
# Serial drivers
#
#
# CONFIG_SERIAL_8250 is not set

#
# Non-8250 serial port support
#
# CONFIG_SERIAL_MAX3100 is not set
CONFIG_SERIAL_BFIN=y
CONFIG_SERIAL_BFIN_CONSOLE=y
CONFIG_SERIAL_BFIN_DMA=y
# CONFIG_SERIAL_BFIN_PIO is not set
CONFIG_SERIAL_BFIN_UART0=y
# CONFIG_BFIN_UART0_CTSRTS is not set
# CONFIG_SERIAL_BFIN_UART1 is not set
CONFIG_SERIAL_CORE=y
CONFIG_SERIAL_CORE_CONSOLE=y
# CONFIG_SERIAL_BFIN_SPORT is not set
# CONFIG_SERIAL_TIMBERDALE is not set
CONFIG_UNIX98_PTYS=y
# CONFIG_DEVPTS_MULTIPLE_INSTANCES is not set
# CONFIG_LEGACY_PTYS is not set
# CONFIG_BFIN_OTP is not set
# CONFIG_IPMI_HANDLER is not set
# CONFIG_HW_RANDOM is not set
# CONFIG_R3964 is not set
# CONFIG_TCG_TPM is not set
# CONFIG_I2C is not set
CONFIG_SPI=y
CONFIG_SPI_MASTER=y

#
# SPI Master Controller Drivers
#
#
CONFIG_SPI_BFIN=y
# CONFIG_SPI_BFIN_LOCK is not set
```

Prototypentwicklung eines EEG-Headsets

```
# CONFIG_SPI_BFIN_SPORT is not set
# CONFIG_SPI_BITBANG is not set
# CONFIG_SPI_GPIO is not set
# CONFIG_SPI_XILINX is not set
# CONFIG_SPI_DESIGNWARE is not set

#
# SPI Protocol Masters
#
CONFIG_SPI_SPIDEV=y
# CONFIG_SPI_TLE62X0 is not set

#
# PPS support
#
# CONFIG_PPS is not set
CONFIG_ARCH_WANT_OPTIONAL_GPIOLIB=y
CONFIG_GPIOLIB=y
CONFIG_GPIO_SYSFS=y

#
# Memory mapped GPIO expanders:
#
# CONFIG_GPIO_IT8761E is not set

#
# I2C GPIO expanders:
#
#
# PCI GPIO expanders:
#
#
# SPI GPIO expanders:
#
# CONFIG_GPIO_MAX7301 is not set
# CONFIG_GPIO_MCP23S08 is not set
# CONFIG_GPIO_MC33880 is not set

#
# AC97 GPIO expanders:
#
# CONFIG_W1 is not set
# CONFIG_POWER_SUPPLY is not set
# CONFIG_HWMON is not set
# CONFIG_THERMAL is not set
CONFIG_WATCHDOG=y
# CONFIG_WATCHDOG_NOWAYOUT is not set

#
# Watchdog Device Drivers
#
# CONFIG_SOFT_WATCHDOG is not set
CONFIG_BFIN_WDT=y
CONFIG_SSB_POSSIBLE=y

#
#
# CONFIG_SSB is not set

#
# Multifunction device drivers
#
# CONFIG_MFD_CORE is not set
# CONFIG_MFD_SM501 is not set
# CONFIG_HTC_PASIC3 is not set
# CONFIG_MFD_TMIO is not set
# CONFIG_MFD_MC13783 is not set
# CONFIG_EZX_PCAP is not set
# CONFIG_AB4500_CORE is not set
# CONFIG_REGULATOR is not set
# CONFIG_MEDIA_SUPPORT is not set

#
# Graphics support
#
# CONFIG_VGASTATE is not set
# CONFIG_VIDEO_OUTPUT_CONTROL is not set
# CONFIG_FB is not set
# CONFIG_BACKLIGHT_LCD_SUPPORT is not set

#
# Display device support
#
# CONFIG_DISPLAY_SUPPORT is not set
# CONFIG_SOUND is not set
# CONFIG_USB_SUPPORT is not set
# CONFIG_MMC is not set
# CONFIG_MEMSTICK is not set
# CONFIG_NEW_LEDS is not set
# CONFIG_ACCESSIBILITY is not set
CONFIG_RTC_LIB=y
CONFIG_RTC_CLASS=y
CONFIG_RTC_HCTOSYS=y
CONFIG_RTC_HCTOSYS_DEVICE="rtc0"
# CONFIG_RTC_DEBUG is not set

#
# RTC interfaces
#
CONFIG_RTC_INTF_SYSFS=y
CONFIG_RTC_INTF_PROC=y
CONFIG_RTC_INTF_DEV=y
# CONFIG_RTC_INTF_DEV_UIE_EMUL is not set
# CONFIG_RTC_DRV_TEST is not set

#
# SPI RTC drivers
#
#
# CONFIG_RTC_DRV_M41T94 is not set
```

Prototypentwicklung eines EEG-Headsets

```
# CONFIG_RTC_DRV_DS1305 is not set
# CONFIG_RTC_DRV_DS1390 is not set
# CONFIG_RTC_DRV_MAX6902 is not set
# CONFIG_RTC_DRV_R9701 is not set
# CONFIG_RTC_DRV_RS5C348 is not set
# CONFIG_RTC_DRV_DS3234 is not set
# CONFIG_RTC_DRV_PCF2123 is not set

#
# Platform RTC drivers
#
# CONFIG_RTC_DRV_DS1286 is not set
# CONFIG_RTC_DRV_DS1511 is not set
# CONFIG_RTC_DRV_DS1553 is not set
# CONFIG_RTC_DRV_DS1742 is not set
# CONFIG_RTC_DRV_STK17TA8 is not set
# CONFIG_RTC_DRV_M48T86 is not set
# CONFIG_RTC_DRV_M48T35 is not set
# CONFIG_RTC_DRV_M48T59 is not set
# CONFIG_RTC_DRV_MSM6242 is not set
# CONFIG_RTC_DRV_BQ4802 is not set
# CONFIG_RTC_DRV_RP5C01 is not set
# CONFIG_RTC_DRV_V3020 is not set

#
# on-CPU RTC drivers
#
CONFIG_RTC_DRV_BFIN=y
# CONFIG_DMADEVICES is not set
# CONFIG_AUXDISPLAY is not set
# CONFIG_UIO is not set

#
# TI VLYNQ
#
# CONFIG_STAGING is not set

#
# Firmware Drivers
#
# CONFIG_FIRMWARE_MEMMAP is not set

#
# File systems
#
CONFIG_FILE_LOCKING=y
CONFIG_FSNOTIFY=y
# CONFIG_DNOTIFY is not set
CONFIG_INOTIFY=y
CONFIG_INOTIFY_USER=y
# CONFIG_QUOTA is not set
# CONFIG_AUTOFS_FS is not set
# CONFIG_AUTOFS4_FS is not set
# CONFIG_FUSE_FS is not set

#

# Caches
#
# CONFIG_FSCACHE is not set

#
# Pseudo filesystems
#
CONFIG_PROC_FS=y
CONFIG_PROC_SYSCTL=y
CONFIG_SYSFS=y
# CONFIG_HUGETLB_PAGE is not set
# CONFIG_CONFIGFS_FS is not set
CONFIG_MISC_FILESYSTEMS=y
CONFIG_NLS=y
CONFIG_NLS_DEFAULT="iso8859-1"
CONFIG_NLS_CODEPAGE_437=y
# CONFIG_NLS_CODEPAGE_737 is not set
# CONFIG_NLS_CODEPAGE_775 is not set
# CONFIG_NLS_CODEPAGE_850 is not set
# CONFIG_NLS_CODEPAGE_852 is not set
# CONFIG_NLS_CODEPAGE_855 is not set
# CONFIG_NLS_CODEPAGE_857 is not set
# CONFIG_NLS_CODEPAGE_860 is not set
# CONFIG_NLS_CODEPAGE_861 is not set
# CONFIG_NLS_CODEPAGE_862 is not set
# CONFIG_NLS_CODEPAGE_863 is not set
# CONFIG_NLS_CODEPAGE_864 is not set
# CONFIG_NLS_CODEPAGE_865 is not set
# CONFIG_NLS_CODEPAGE_866 is not set
# CONFIG_NLS_CODEPAGE_869 is not set
# CONFIG_NLS_CODEPAGE_936 is not set
# CONFIG_NLS_CODEPAGE_950 is not set
# CONFIG_NLS_CODEPAGE_932 is not set
# CONFIG_NLS_CODEPAGE_949 is not set
# CONFIG_NLS_CODEPAGE_874 is not set
# CONFIG_NLS_ISO8859_8 is not set
# CONFIG_NLS_CODEPAGE_1250 is not set
# CONFIG_NLS_CODEPAGE_1251 is not set
# CONFIG_NLS_ASCII is not set
CONFIG_NLS_ISO8859_1=y
# CONFIG_NLS_ISO8859_2 is not set
# CONFIG_NLS_ISO8859_3 is not set
# CONFIG_NLS_ISO8859_4 is not set
# CONFIG_NLS_ISO8859_5 is not set
# CONFIG_NLS_ISO8859_6 is not set
# CONFIG_NLS_ISO8859_7 is not set
# CONFIG_NLS_ISO8859_9 is not set
# CONFIG_NLS_ISO8859_13 is not set
# CONFIG_NLS_ISO8859_14 is not set
# CONFIG_NLS_ISO8859_15 is not set
# CONFIG_NLS_KOI8_R is not set
# CONFIG_NLS_KOI8_U is not set
# CONFIG_NLS_UTF8 is not set

#
# Kernel hacking
```


Prototypentwicklung eines EEG-Headsets

```
# CONFIG_DEFAULT_SECURITY_TOMOYO is not set
# CONFIG_PRINTK_TIME is not set
CONFIG_ENABLE_WARN_DEPRECATED=y
CONFIG_ENABLE_MUST_CHECK=y
CONFIG_FRAME_WARN=1024
# CONFIG_MAGIC_SYSRQ is not set
# CONFIG_STRIP_ASM_SYMS is not set
# CONFIG_UNUSED_SYMBOLS is not set
CONFIG_DEBUG_FS=y
# CONFIG_HEADERS_CHECK is not set
CONFIG_DEBUG_SECTION_MISMATCH=y
# CONFIG_IPIPE_DEBUG is not set
# CONFIG_DEBUG_KERNEL is not set
CONFIG_DEBUG_BUGVERBOSE=y
# CONFIG_DEBUG_MEMORY_INIT is not set
# CONFIG_RCU_CPU_STALL_DETECTOR is not set
# CONFIG_TRACEHOOK_SELF_TEST is not set
# CONFIG_SYSCTL_SYSCALL_CHECK is not set
CONFIG_HAVE_FUNCTION_TRACER=y
CONFIG_HAVE_FUNCTION_GRAPH_TRACER=y
CONFIG_HAVE_FUNCTION_TRACE_MCOUNT_TEST=y
CONFIG_HAVE_DYNAMIC_FTRACE=y
CONFIG_HAVE_FTRACE_MCOUNT_RECORD=y
CONFIG_TRACING_SUPPORT=y
# CONFIG_FTRACE is not set
# CONFIG_DYNAMIC_DEBUG is not set
# CONFIG_SAMPLES is not set
CONFIG_HAVE_ARCH_KGDB=y
CONFIG_DEBUG_VERBOSE=y
# CONFIG_DEBUG_MMRS is not set
# CONFIG_DEBUG_DOUBLEFAULT is not set
CONFIG_DEBUG_HUNT_FOR_ZERO=y
CONFIG_DEBUG_BFIN_HWTRACE_ON=y
CONFIG_DEBUG_BFIN_HWTRACE_COMPRESSION_OFF=y
# CONFIG_DEBUG_BFIN_HWTRACE_COMPRESSION_ONE is not set
# CONFIG_DEBUG_BFIN_HWTRACE_COMPRESSION_TWO is not set
CONFIG_DEBUG_BFIN_HWTRACE_COMPRESSION=0
# CONFIG_DEBUG_BFIN_HWTRACE_EXPAND is not set
# CONFIG_DEBUG_BFIN_NO_KERN_HWTRACE is not set
CONFIG_EARLY_PRINTK=y
# CONFIG_CPLB_INFO is not set
CONFIG_ACCESS_CHECK=y
# CONFIG_BFIN_ISRAM_SELF_TEST is not set
# CONFIG_BFIN_PSEUDODBG_INSNS is not set

#
# Security options
#
# CONFIG_KEYS is not set
CONFIG_SECURITY=y
# CONFIG_SECURITYFS is not set
# CONFIG_SECURITY_NETWORK is not set
# CONFIG_SECURITY_PATH is not set
# CONFIG_SECURITY_TOMOYO is not set
# CONFIG_DEFAULT_SECURITY_SELINUX is not set
# CONFIG_DEFAULT_SECURITY_SMACK is not set

# CONFIG_DEFAULT_SECURITY_TOMOYO is not set
CONFIG_DEFAULT_SECURITY_DAC=y
CONFIG_DEFAULT_SECURITY=""
CONFIG_CCRYPTO=y

#
# Crypto core or helper
#
# CONFIG_CCRYPTO_MANAGER is not set
# CONFIG_CCRYPTO_MANAGER2 is not set
# CONFIG_CCRYPTO_GF128MUL is not set
# CONFIG_CCRYPTO_NULL is not set
# CONFIG_CCRYPTO_CRYPTD is not set
# CONFIG_CCRYPTO_AUTHENC is not set
# CONFIG_CCRYPTO_TEST is not set

#
# Authenticated Encryption with Associated Data
#
# CONFIG_CCRYPTO_CCM is not set
# CONFIG_CCRYPTO_GCM is not set
# CONFIG_CCRYPTO_SEQIV is not set

#
# Block modes
#
# CONFIG_CCRYPTO_CBC is not set
# CONFIG_CCRYPTO_CTR is not set
# CONFIG_CCRYPTO_CTS is not set
# CONFIG_CCRYPTO_ECB is not set
# CONFIG_CCRYPTO_LRW is not set
# CONFIG_CCRYPTO_PCBC is not set
# CONFIG_CCRYPTO_XTS is not set

#
# Hash modes
#
# CONFIG_CCRYPTO_HMAC is not set
# CONFIG_CCRYPTO_XCBC is not set
# CONFIG_CCRYPTO_VMAC is not set

#
# Digest
#
# CONFIG_CCRYPTO_CRC32C is not set
# CONFIG_CCRYPTO_GHASH is not set
# CONFIG_CCRYPTO_MD4 is not set
# CONFIG_CCRYPTO_MD5 is not set
# CONFIG_CCRYPTO_MICHAEL_MIC is not set
# CONFIG_CCRYPTO_RMD128 is not set
# CONFIG_CCRYPTO_RMD160 is not set
# CONFIG_CCRYPTO_RMD256 is not set
# CONFIG_CCRYPTO_RMD320 is not set
# CONFIG_CCRYPTO_SHA1 is not set
# CONFIG_CCRYPTO_SHA256 is not set
# CONFIG_CCRYPTO_SHA512 is not set
```

Prototypentwicklung eines EEG-Headsets

```
# CONFIG_CRYPTO_TGR192 is not set
# CONFIG_CRYPTO_WP512 is not set

#
# Ciphers
#
# CONFIG_CRYPTO_AES is not set
# CONFIG_CRYPTO_ANUBIS is not set
# CONFIG_CRYPTO_ARC4 is not set
# CONFIG_CRYPTO_BLOWFISH is not set
# CONFIG_CRYPTO_CAMELLIA is not set
# CONFIG_CRYPTO_CAST5 is not set
# CONFIG_CRYPTO_CAST6 is not set
# CONFIG_CRYPTO_DES is not set
# CONFIG_CRYPTO_FCRYPT is not set
# CONFIG_CRYPTO_KHAZAD is not set
# CONFIG_CRYPTO_SALSA20 is not set
# CONFIG_CRYPTO_SEED is not set
# CONFIG_CRYPTO_SERPENT is not set
# CONFIG_CRYPTO_TEA is not set
# CONFIG_CRYPTO_TWOFISH is not set

#
# Compression
#
# CONFIG_CRYPTO_DEFLATE is not set

# CONFIG_CRYPTO_ZLIB is not set
# CONFIG_CRYPTO_LZO is not set

#
# Random Number Generation
#
# CONFIG_CRYPTO_ANSI_CPRNG is not set
CONFIG_CRYPTO_HW=y
# CONFIG_BINARY_PRINTF is not set

#
# Library routines
#
CONFIG_BITREVERSE=y
CONFIG_GENERIC_FIND_LAST_BIT=y
CONFIG_CRC_CCITT=m
# CONFIG_CRC16 is not set
# CONFIG_CRC_T10DIF is not set
CONFIG_CRC_ITU_T=y
CONFIG_CRC32=y
CONFIG_CRC7=y
# CONFIG_LIBCRC32C is not set
CONFIG_DECOMPRESS_LZMA=y
CONFIG_HAS_IOMEM=y
CONFIG_HAS_IOPORT=y
CONFIG_HAS_DMA=y
```

Anhang D: Programmcodes

Die der Kombilierung der Programme werden *libxenomai.a*, *libnative.a*, *libbfdsp.a* und *libm.a* benötigt.

D1: C-Quellcode von *read_ADC*

```
//Projekt: EEG-Headset
//Program: read_ADC
//Author: Hagauer Peter
//Date: 29.11.2011
//
//Demonstration Programm, that shows how to read the EEG in real-time and how to calculate the Power Spectrum.
//Prints every value from the ADC on display.
//Runs till any Button is pushed and then prints the Power Spectrum of the last second.
//Call with: read_ADC
//
//Program is written for ADC-Channel 0!!!

#include <stdio.h>
#include <signal.h>
#include <unistd.h>
#include <stdlib.h>
#include <fcntl.h>
#include <fract.h>
#include <fract_complex.h>
#include <fract_math.h>
#include <math_bf.h>
#include <filter.h>
#include <sys/mman.h>
#include <native/task.h>
#include <native/timer.h>

#define ADC "/dev/spidev0.1"
//#define USB "/dev/ttyGS0"
#define SAMPLES 256

RT_TASK demo_task;
int fd;
complex_fract16 twiddle[SAMPLES/2]={0}; //Twiddle Table of FFT
fract16 input[SAMPLES]={0}; //Input sequenz of FFT
complex_fract16 output[SAMPLES]={0}; //Output of FFT
fract32 psd[SAMPLES]={0}; //PSD of output
```

Prototypentwicklung eines EEG-Headsets

```
void demo(void *arg)
{
    RTIME now, previous;
    int status, t, blockexp;
    int end = 0;
    uint16_t tx[] = { 0x87 }; //WRITE: CH0: 10000111 = 0x87
                                //      CH1: 11000111 = 0xC7
    fract16 buf[4]={0};

    rt_task_set_periodic(NULL, TM_NOW, 3906250); // in nano-seconds, 256 Hz
    previous = rt_timer_read();

    while (end == 0)
    {
        rt_task_wait_period(NULL);
        now = rt_timer_read();
        previous = now; // to display the calculation time, if needed

        write(fd,tx,2);
        status = read(fd, buf, 2);

        for ( t = SAMPLES; t > 0; input[t] = input[t-1], t--); // Circular Buffer
        input[0] = buf[0];
        rfft_fr16(input, output, twiddle, 1, SAMPLES, &blockexp, 0);
        for ( t = 0; t < SAMPLES; psd[t] = mult_fr1x32(cabs_fr16(output[t]), cabs_fr16(output[t]), t++);
        printf("%d \n", input[0]);
    }
}

int main(int argc, char* argv[])
{
    int t;

    fd = open(ADC, O_RDWR);
    if (fd < 0)
    {
        printf("Cannot open ADC- Port");
        exit (1);
    }
    printf ("spidev0.1 opened\n");
    twidfftrad2_fr16(twiddle, SAMPLES);

    mlockall(MCL_CURRENT|MCL_FUTURE); //Avoids memory swapping for this program

    rt_task_create(&demo_task, "trivial", 0, 99, 0);
    rt_task_start(&demo_task, &demo, NULL);
}
```

```
char taste = getchar(); //Quits programm

printf("Spectrum of last second:\n");
for ( t=0;t<70;printf("%d: %d\n",t,psd[t]), t++);
printf("\n");

rt_task_delete(&demo_task);
return 0;
}
```

D2: C-Quellcode von BCI_training

```
//Projekt: EEG-Headset
//Program: BCI_training
//Author: Hagauer Peter
//Date: 29.11.2011
//
//BCI Training Program, runs till any Button is pushed
//
//Call with: BCI_training TRIALS
//
//TRIALS: Number of training- runs, 8 s each
//You can get appropriate FREQ for BCI

#include <stdio.h>
#include <signal.h>
#include <unistd.h>
#include <stdlib.h>
#include <fcntl.h>
#include <fract.h>
#include <fract_complex.h>
#include <fract_math.h>
#include <math_bf.h>
#include <filter.h>
#include <sys/mman.h>
#include <native/task.h>
#include <native/timer.h>

#define ADC "/dev/spidev0.1"
#define SAMPLES 256

int TRIALS = 0;
RT_TASK demo_task;
int fd;
complex_fract16 twiddle[SAMPLES/2]={0}; //Twiddle Table
fract16 input[SAMPLES]={0}; //Input sequenz
```

Prototypentwicklung eines EEG-Headsets

```
complex_fract16 output[SAMPLES]={0}; //Output of FFT
fract32 psd[SAMPLES]={0}; //PSD of output
fract32 specref[SAMPLES]={0}; //Spektra of reference- intervall
fract32 specaktiv[SAMPLES]={0}; //Spektra of aktiv- intervall

void demo(void *arg)
{
    RTIME now, previous;
    int status, t, s, blockexp;
    int end = 0;
    int count = 0;
    int first = 0;
    int on = 0;
    uint16_t tx[] = { 0x87 }; //WRITE: CH0: 10000111 = 0x87
                                // CH1: 11000111 = 0xC7
    fract16 buf[4]={0};

    rt_task_set_periodic(NULL, TM_NOW, 3906250);
    previous = rt_timer_read();

    while (end == 0)
    {
        rt_task_wait_period(NULL);
        now = rt_timer_read();
        previous = now;
        write(fd,tx,2);
        status = read(fd, buf, 2);
        for ( t = SAMPLES; t > 0; input[t] = input[t-1], t--); // Ringbuffer
        input[0] = buf[0];
        rfft_fr16(input, output, twiddle, 1, SAMPLES, &blockexp, 0);
        for ( t = 0; t < SAMPLES; psd[t] = mult_fr1x32(cabs_fr16(output[t]), cabs_fr16(output[t])), t++);
        count++;
    }
    //+++++Trainingsparadigma Timing+++++
    // Trialdauer = 8s
    // Referenzzeitraum: 2-3s
    // Aufforderung zur Bewegungsvorstellung bei Sek. 4
    // Aktivintervall: 6 bis 7

    // Gewinnung der Referenz- PSD
    if (count == 768) //Sekunde 3
    {
        for ( t = 0; t < SAMPLES; specref[t] = add_fr1x32(shr_fr1x32(psd[t], first), shr_fr1x32(specref[t], first)), t++);
    }
}
```



```
char taste = getchar();
printf("END of BCI- training\n");

printf("REF:\n");
for ( t=0;t<50;printf("%d: %d\n",t,specreff[t]), t++);
printf("\n");
printf("AKT:\n");
for ( t=0;t<50;printf("%d: %d\n",t,specaktiv[t]), t++);
printf("\n");
printf("AKT - Ref:\n");
for ( t=0;t<50;printf("%d %d\n",t,specaktiv[t]-specreff[t]), t++);
printf("\n");

rt_task_delete(&demo_task);
return 0;
}
```

D3: C-Quellcode von BCI

```
//Projekt: EEG-Headset
//Program: BCI
//Author: Hagauer Peter
//Date: 29.11.2011
//BCI Program, runs till any Button is pushed
//
//Call with: BCI_FREQ_THRES
//FREQ: Frequency of Interest
//THRES: Threshold of Detection
//You can get FREQ with help of BCI_Training

#include <stdio.h>
#include <signal.h>
#include <unistd.h>
#include <stdlib.h>
#include <fcntl.h>
#include <fract.h>
#include <fract_complex.h>
#include <fract_math.h>
#include <math_bf.h>
#include <filter.h>
#include <sys/mman.h>
#include <native/task.h>
#include <native/timer.h>

#define ADC "/dev/spidev0.1"
#define SAMPLES 256
```


Prototypentwicklung eines EEG-Headsets

```
int FREQ=0;
int THRES=0;
RT_TASK demo_task;
int fd;
complex_fract16 twiddle[SAMPLES/2]={0}; //Twiddle Table
fract16 input[SAMPLES]={0}; //Input sequenz
complex_fract16 output[SAMPLES]={0}; //Output of FFT
fract32 psd[SAMPLES]={0}; //PSD of output
fract32 specref[SAMPLES]={0}; //Spektra of reference- intervall
fract32 specaktiv[SAMPLES]={0}; //Spektra of activ- intervall

void demo(void *arg)
{
    RTIME now, previous;
    int status, t, blockexp;
    int end = 0;
    int count = 0;
    int on = 0;
    uint16_t tx[] = { 0x87 }; //WRITE: CH0: 1000111 = 0x87
                                // CH1: 11000111 = 0xC7

    fract16 buf[4]={0};

    rt_task_set_periodic(NULL, TM_NOW, 3906250); // in ns
    previous = rt_timer_read();

    while (end == 0)
    {
        rt_task_wait_period(NULL);
        now = rt_timer_read();
        write(fd,tx,2);
        status = read(fd, buf, 2);
        for ( t = SAMPLES; t > 0; input[t] = input[t-1], t--); // Circular Buffer
        input[0] = buf[0];
        rfft_fr16(input, output, twiddle,1,SAMPLES, &blockexp,0);
        for ( t = 0; t < SAMPLES; psd[t] = mult_fr1x32(cabs_fr16(output[t]),cabs_fr16(output[t])), t++);
        count++;

        if (psd[FREQ] > THRES)
        {
            {
                if (count > 512) //was last detection over a second ago
                {
                    printf("Device ON!\n");
                    count = 0;
                }
            }
        }
    }
}
```

