

**Ein universelles System zur Anwendung von Biosignalen
im Biofeedback und als Human Computer Interface**

Diplomarbeit

von

Christoph Veigl

durchgeführt am Institut "integriert studieren",
2006 / 2007

bewertet von Prof. Dr. Wolfgang Zagler

Hintergrund und Motivation

Die Messung und Analyse biologischer Signale des Organismus liefert Informationen über Vitalfunktionen, die wertvoll für die medizinische Diagnostik und zur Kontrolle von Therapieerfolgen sind. Erfolgt die Auswertung der Messergebnisse in Echtzeit (d.h. mit einer sehr geringen Verzögerung), finden sich weitere Einsatzgebiete in der Biofeedbacktherapie, im Design von Schnittstellen für die Mensch-Maschine-Kommunikation (*HCI, Human Computer Interfaces*) oder in der Prothetik und Schrittmachertechnik.

Durch den Trend zur Leistungssteigerung in der Digitaltechnik eignen sich handelsübliche Computer heute für eine qualitative Auswertung von bioelektrischen Signalen und zur raschen Analyse von Live-Kamerabildern. Dies eröffnet Möglichkeiten für die Entwicklung preisgünstiger Hard- und Softwaresysteme und deren Einsatz als Therapie- und Kommunikationshilfe abseits einer Klinik.

Die Betrachtung des kommerziellen Angebotes an Biosignaltechnik zeigt eine Vielzahl von Produkten, deren Einsatzmöglichkeit sich von Ergometrie und Bewegungstherapie über Entspannungs- und Biofeedbacktraining bis zur Überwachung physiologischer Funktionen und zur Verwendung als alternatives Kommunikationsmedium erstreckt. Viele der Lösungen sind dabei stark auf einen Anwendungsbereich eingeschränkt und bieten softwaretechnisch kaum Möglichkeiten für dynamische Konfigurationen oder Erweiterungen. Auch im Bereich der Kommunikationshilfsmittel für Menschen mit motorischer Behinderung existieren viele Speziallösungen, die meist auf einer bestimmten Eingabemodalität aufbauen; hier wäre jedoch eine weitgehende Adaptierbarkeit nützlich, um das System optimal auf die individuellen Bedürfnisse der Benutzer abstimmen zu können.

Projektzielsetzung

Ziel dieser Arbeit ist es, ein kostengünstiges System zur Erhebung, Auswertung und Aufzeichnung von Biosignalen sowie von Sensor- und Videodaten zu realisieren, welches audiovisuelle Darstellungsmöglichkeiten für Biofeedback bietet und motorisch behinderten Menschen alternative Eingabemodalitäten für die Verwendung des Computers zur Verfügung stellt. Die Design-Prämissen für die Entwicklung sind intuitive Konfigurierbarkeit mittels grafischer Benutzeroberfläche, schnelle Signalverarbeitung, ansprechende Aufbereitung der Vitalparameter und Kombinierbarkeit mit bestehenden Lösungen für Umgebungssteuerung und Computerinteraktion.

Die vorliegende Ausarbeitung gibt einen Überblick über die verwendete Technik und stellt die Hardware- und Softwarekomponenten des Systems dar. Diese Komponenten bauen auf freier Open Hardware und Open Software auf und werden als solche veröffentlicht. Dadurch bleiben Aufbau und Funktionalität einsehbar, das System kann von anderen Entwicklern und Entwicklerinnen erweitert und ohne Lizenzgebühren zu geringen Kosten reproduziert oder verändert werden.

Danksagung

Die vorliegende Arbeit wäre ohne die Unterstützung vieler Menschen nicht möglich gewesen. Allen voran möchte ich meinen Eltern, Gerda und Johann Veigl, für ihr Vertrauen, ihre Ausdauer und die finanzielle Hilfe danken. Sie haben mir eine intensive Beschäftigung mit wissenschaftlichen und künstlerischen Themen ermöglicht, die nur in seltenen Fällen monetär einträglich war. Ich konnte mir dadurch einen guten Überblick über das universitäre Angebot verschaffen und einen Bereich finden, der mich fasziniert und in dem ich meine Fähigkeiten sinnvoll einsetzen kann.

Ebenso danke ich meiner Freundin, Astrid Schartner, die mit viel Geduld meine oft nächtelangen Programmier- und Experimentiertätigkeiten toleriert hat, obwohl der praktische Nutzen für sie nicht immer ersichtlich war. Ihre Zuneigung hat mich nicht selten vor einer Kapitulation bewahrt, als diverse Hard- oder Softwareprobleme ausweglos erschienen.

Ein besonderer Dank gilt Jörg Hansmann, dem Entwicklungsleiter des *ModularEEG*-Projektes und Mitbegründer der *OpenEEG*-Gruppe, der sein großes Wissen im Bereich der Elektronik unentgeltlich zur Verfügung gestellt und somit mir und vielen Anderen einen Weg zum Verständnis und zur Anfertigung eigener Biosignalprojekte geebnet hat. Jörg verstand es, mich zu motivieren, und seine Ferndiagnosen von Platinenfehlern haben sich nahezu immer bewahrheitet.

In gleicher Weise danke ich Reiner Münch, dem Designer des *MonolithEEG*, dessen Einsatz zu einer Verbesserung und Miniaturisierung des *ModularEEG*-Konzeptes führte und der mir durch seine Unterstützung einen reibungslosen Einstieg in die Fertigung von *SMD*-Platinen ermöglichte.

Weiters gebührt mein Dank Jim Peters, dessen Arbeiten im Bereich digitaler Filter eine Grundlage für die Entwicklung meiner Applikation darstellten und der es vermochte, die Konzepte der digitalen Signalverarbeitung auf eine verständliche Weise zu vermitteln.

Ich möchte auch allen Benutzerinnen und Benutzern der entstandenen Software danken, die mit ihrem Feedback zu einer Weiterentwicklung des Programms beitragen, im besonderen Jeremy Wilkerson, der sich selbst an der Programmierung neuer Elemente beteiligte und seine Ideen zur Verbesserung der Applikation einbrachte.

Last but not least gilt mein Dank den Betreuern meiner Diplomarbeit: Prof. Dr. Wolfgang Zagler und DI Dr. Paul Panek vom Institut "integriert studieren" der Technischen Universität Wien. Sie ermöglichten mir ein freies Arbeiten und vertrauten auf meinen Vorschlag für das Thema der Arbeit, unterstützten meine Ideen und standen mir mit Erfahrung und ihrem Rat zur Seite.

Zur Struktur dieser Arbeit

Kapitel 1 zeigt die geschichtliche Entwicklung und die theoretische Basis des Bioelektromagnetismus und geht auf die physiologischen Grundlagen der Entstehung bioelektrischer Signale durch elektrochemische Prozesse im vitalen Nerven- und Muskelgewebe ein.

Kapitel 2 beschreibt Arten bioelektrischer Signale, ihre Charakteristika und Anwendungsmöglichkeiten. Die Methode des *Biofeedback* und verschiedene Formen der Feedbacktherapie werden vorgestellt. Es folgt ein Überblick auf Kommunikationstechnik für Menschen mit motorischen Behinderungen, wobei ein besonderes Augenmerk auf moderne biosignalbasierte Schnittstellen, *Brain-Computer Interfaces* und berührungsfreie kameraunterstützte Verfahren (*vision-based Human Computer Interaction*) gelegt wird.

Kapitel 3 stellt die Grundlagen des Messvorganges und die Hardwarekonzepte von Biosignalverstärkern dar. Nach einem Überblick auf kommerziell verfügbare Biosignal-Hardware erfolgt die Darstellung von Aufbau und Funktion der Schaltungen des *OpenEEG* – Projektes.

Kapitel 4 zeigt mathematische Methoden und Algorithmen zur digitalen Signalverarbeitung. Die grundlegenden Funktionen und ihre Anwendung sowie mögliche Strategien für eine Implementierung von Klassifikatoren und Bildverarbeitungsalgorithmen werden vorgestellt.

In **Kapitel 5** wird aus den erhobenen Rahmenbedingungen das Konzept für eine Softwareapplikation für Biofeedback und *HCI*-Anwendung erstellt. Die Anforderungen, der objektorientierte Aufbau, die Klassenhierarchie und die Datenflüsse werden dargestellt und Strategien für das User Interface gezeigt.

Kapitel 6 behandelt Details zur Implementierung wie die Entwicklungsumgebung, die verwendeten Bibliotheken und die Realisierung von *Kernsystem*, *I/O-Verwaltung*, *Timing - Funktionen* und *Bildverarbeitung*. Weiters wird die Erzeugung eines neuen Konfigurationselementes anhand eines konkreten Beispiels beschrieben.

Kapitel 7 zeigt die Anwendung der lauffähigen Software durch eine Beschreibung der Einsatzmöglichkeiten und der Parameter aller zur Verfügung stehenden Elemente.

Kapitel 8 besteht aus Anwendungsbeispielen für die lauffähige Software und zeigt Beispielkonfigurationen für EEG-Feedback, Muskeltraining und *Kamera-Maus*.

Kapitel 9 evaluiert die Ergebnisse anhand von Kennzahlen, Rückmeldungen der Testpersonen und einer Gegenüberstellung mit den Design-Zielen. Abschließend zeigt ein Überblick geplante Verbesserungen und zukünftige Entwicklungen.

In **Anhang A** wird die *BrainBay*-Homepage vorgestellt.

Inhaltsverzeichnis

Hintergrund und Motivation	i
Projektzielsetzung	i
Danksagung	ii
Zur Struktur dieser Arbeit	iii
Inhaltsverzeichnis	iv
Abbildungsverzeichnis	xii
Tabellen	xv
Gleichungen	xv
1. Bioelektromagnetismus und Elektrophysiologie	1
1.1 Geschichtliche Entwicklung des Bioelektromagnetismus	2
1.2 Physiologische und anatomische Grundlagen -.....	6
1.2.1 Die Vorgänge an der Zellmembran.....	6
1.2.2 Die Entstehung des Ruhemembranpotenzials	8
1.2.3 Berechnung des Ruhemembranpotenzials: Die Nernst Gleichung.....	8
1.2.4 Erhaltung des Gleichgewichtspotenzials: die Ionenpumpen.....	9
1.2.5 Funktionale Einheiten in der Zellmembran: die Ionenkanäle	9
1.2.6 Eigenschaften von Kanalproteinen.....	10
1.2.7 Weiterleitung bioelektrischer Signale: Das Aktionspotenzial	11
1.2.8 Übertragung von Information auf andere Zellen: Die <i>Synapse</i>	12
1.2.9 Vom Nerv zum Muskel: Die neuromotorische Endplatte	12
1.2.10 Das Nervensystem	13
2. Eigenschaften und Anwendung von Biosignalen	15
2.1 Elektrische Biosignale	15
2.1.1 Elektromyogramm (EMG).....	16
2.1.2 Elektrokardiogramm (EKG)	17
2.1.3 Elektrookulogramm (EOG)	20
2.1.4 Elektroenzephalogramm (EEG) und Elektrokortikogramm (ECoG).....	21
2.1.4.1 Evozierte und Ereigniskorrelierte Potenziale (EP, EKP, ERP)	23
2.1.4.2 ERD / ERS.....	24
2.1.4.3 Langsame Potenziale (SCP), Laterales Bereitschaftspotenzial (LRP)	24
2.1.5 Elektrodermale Aktivität (EDA)	25

2.2 Nichtelektrische Biosignale	26
2.2.1 Atmung	26
2.2.2 Körpertemperatur	26
2.2.3 Blutvolumenamplitude	27
2.2.4 Augenbewegungen.....	27
2.2.4.1 Infrarotokulographie	27
2.2.4.2 Videookulographie.....	27
2.2.5 Stimme und Sprechen	28
2.2.6 Gestik und Bewegungen	28
2.3 Biofeedbacktherapie und ihre Anwendungen	29
2.3.1 Biofeedback bei chronischen Rückenschmerzen	30
2.3.2 Biofeedback bei Spannungskopfschmerz und Migräne.....	31
2.3.3 Biofeedback bei Hypertonie.....	32
2.3.4 Biofeedback bei Somatisierungspatienten und bei Angststörungen.....	32
2.3.5 Biofeedback in der Therapie des chronischen Tinnitus	33
2.3.6 Biofeedback zur Behandlung von Inkontinenzproblemen.....	34
2.3.7 Biofeedback bei Neurologischen Erkrankungen.....	36
2.3.7.1 Neurofeedback bei epileptischen Anfällen	36
2.3.7.2 Neurofeedback bei Aufmerksamkeitsstörungen	37
2.4 Unterstützungstechnik.....	37
2.4.1 Hilfsmittel für die direkte Eingabe	38
2.4.2 Scanning und alternative Schaltmechanismen	38
2.4.3 Sprach- und Umgebungssteuerungen.....	39
2.4.4 Interaktion durch EMG oder EOG.....	39
2.4.4.1 Musersatz durch EMG.....	40
2.4.4.2 Musersatz durch EOG: EagleEyes.....	40
2.4.4.3 Kombination von EOG und Beschleunigungssensoren	41
2.4.4.4 Kombination von EMG und EOG: MindMouse.....	41
2.4.4.5 Resümee EMG- und EOG-Systeme	41
2.4.5 Interaktion durch Infrarotdetektion von Bewegungsmarkern	42
2.4.6 Interaktion durch Kamerabildanalyse	42
2.4.6.1 Detektion von Gesichtsposition und Gesichtsmerkmalen.....	43
2.4.6.2 Detektion der Blickrichtung (Gaze Tracking).....	44
2.4.5.5 Resümee kamerabasierte Systeme	45
2.5 Brain Computer Interfaces	46
2.5.1 SCP - BCIs	46
2.5.2 μ -BCIs	47
2.5.3 VEP-BCIs	48
2.5.4 P300-BCIs.....	49
2.5.5 Direct-Brain-Interfaces.....	49
2.5.6 BCI2000	50

3. Hardware zur Messung bioelektrischer Signale	52
3.1 Prinzipielles zum Messvorgang	52
3.1.1 Monopolare und bipolare Ableitung	52
3.1.2 Der Elektrode-Haut-Kontakt	53
3.1.3 Noise und Artefakte	54
3.1.3.1 Einkopplung von Störungen, Gleichtaktanteil und CMR	54
3.1.3.2 Artefakte durch andere Biosignale	55
3.1.3.3 Artefakte durch Bewegungen von Elektroden oder Kabeln	55
3.1.3.4 Hohe oder unbalancierte Elektrodenimpedanzen	55
3.1.3.5 Elektrodenpolarisation	56
3.1.3.6 Internes Rauschen der elektronischen Bauteile	56
3.1.4 Der Instrumentenverstärker	56
3.1.5 Filterung und Digitalisierung	57
3.1.6 Die Eichung der Messkette	58
3.1.7 Patientenschutzmaßnahmen und Normen	59
3.1.7.1 Die wichtigsten Anforderungen an die Hardware:	59
3.1.7.2 Kennzeichnung der zertifizierten Medizintechnikgeräte:	59
3.2 Kommerzielle Biosignalverstärker - ein Überblick	60
3.2.1 <i>Noraxon Myotrace200</i>	60
3.2.2 <i>Mindpeak WaveRider</i>	61
3.2.3 <i>Brainmaster Atlantis</i>	61
3.2.4 <i>Symtop EEG-Verstärker</i>	62
3.2.5 <i>Leonardo EEG-Verstärker</i>	62
3.2.6 <i>USBamp</i>	63
3.2.7 <i>Synamp EEG-Verstärker</i>	63
3.3 Freie Hardware: Das <i>OpenEEG</i> - Projekt	64
3.3.1 Der <i>ModularEEG</i> - Verstärker	64
3.3.1.1 Die Verstärkerplatine des ModularEEG: das Analog Board	65
3.3.1.2 Die Hauptplatine des ModularEEG: das Digital Board	65
3.3.1.3 Spezifikationen des ModularEEG:	66
3.3.2 Der <i>MonolithEEG</i> - Verstärker	67
3.3.2.1 Spezifikationen des Monolith EEG:	67
3.3.4 Firmware zur Verwendung der <i>OpenEEG</i> -Hardware	68
3.3.5 Elektroden, alternative Sensoren und Hardwareerweiterungen	69
3.3.5.1 Die Anfertigung von Silberelektroden zur EEG-Ableitung	69
3.3.5.2 Die Anfertigung einer EMG-Kappe mit Beschleunigungssensoren	70
3.3.5.3 Aktive EEG-Elektroden	70
3.3.5.3 Aktive EEG-Elektroden	71
3.3.5.4 Weitere Sensoren	71
3.3.5.5 Datenübertragung per Funk mit Bluetooth	72
3.3.5.6 Offline-Datenaufzeichnung	72

4. Softwaremethoden der Signalverarbeitung	73
4.1 Lineare Systeme	73
4.2 Grundlegendes zur Signalstatistik	74
4.2.1 Erwartungswert und Standardabweichung	74
4.2.2 Histogramm, Wahrscheinlichkeitsdichte	75
4.3 Digitale Filter	75
4.3.1 Faltung (Convolution)	77
4.3.2 Weitere Arten digitaler Filter	78
4.3.3 FFT-Convolution.....	79
4.4 Korrelation und Autokorrelation	79
4.5 Diskrete Fourier Transformation.....	79
4.6 Weiterführende Klassifikationsmethoden	82
4.6.1 Neuronale Netze.....	82
4.6.2 PCA und ICA	83
4.6.3 SVMs.....	83
4.6.4 Merkmalsextraktion in der Bildanalyse	84
4.6.4.1 Template Matching.....	84
4.6.4.2 Haar-Classifer	84
4.6.4.3 Lucas-Kanade Algorithmus	84
4.6.4.4 Active Appearance Models.....	85
4.7 Softwarepakete zur Biosignalverarbeitung	85
4.7.1 Matlab/Simulink	85
4.7.1.1 EEGLab.....	86
4.7.2 Scilab und Octave	86
4.7.3 Tempo	87
4.7.4 BioExplorer.....	87
4.7.5 <i>OpenEEG</i> -Software.....	87
4.7.5.1 FidView - Filter Designer	88
4.7.5.2 BWView - Wavelet EEG-Viewer	88
4.7.5.3 BioEra.....	89
4.7.6 Resumee Biosignal-Software.....	89

5. <i>BrainBay</i> : Entwurf der Applikation	90
5.1 Anforderungskatalog	90
5.2 Überlegungen zur funktionalen Struktur	92
5.2.1 Konzept für den Aufbau von Signalverarbeitungskonfigurationen	92
5.2.1.1 Signalquellen	92
5.2.1.2 Verarbeitungselemente	93
5.2.1.3 Signalsenken	93
5.2.2 Strukturen für die Verwendung der Konfigurationselemente	93
5.2.3 Interne Organisation der Konfigurationselemente	93
5.3 Konzepte für den operativen Ablauf der Signalverarbeitung	94
5.3.1 Möglichkeiten für das Zeitmanagement	94
5.3.2 Strategien für zeitkritische System- und Rechenaufgaben	94
5.4 Objektorientiertes Design	95
5.4.1 Beziehung der Elementklassen zur Basisklasse	96
5.5 Gemeinsame Methoden für die Objektklassen	97
5.6 Konzepte für das User Interface	97
6. Implementierung der <i>BrainBay</i> -Software	98
6.1 Systemvoraussetzungen und Entwicklungsumgebung	98
6.2 Verwendete Bibliotheken	99
6.2.1 Statische Bibliotheken	99
6.2.1.1 Standardbibliotheken (Teil der Visual Studio Umgebung) :	99
6.2.1.2 Open-Source Bibliotheken:	99
6.2.1.3 kommerzielle Bibliotheken für die <i>Matlab</i> -Unterstützung	100
6.2.2 Laufzeitbibliotheken (Dynamic Link Libraries, DLLs)	100
6.3 Prinzipien der Windows Programmierung	100
6.3.1 Nachrichtenverarbeitung in einer Windows-Applikation:	101
6.3.2 Behandlung von Nachrichten: Der <i>Window-Handler</i>	101
6.4 Die Basisklasse für <i>BrainBay</i> - Elemente	103
6.5 Quellcode - Module	104
6.5.1 Header-Datei <i>Base.h</i> :	104
6.5.2 Header-Datei <i>Brainbay.h</i> :	104
6.5.3 Modul <i>Brainbay.cpp</i> :	104

6.5.4 Modul <i>Globals.cpp</i> :.....	104
6.5.4.1 Felder und Variablen für die Element-Verwaltung :	104
6.5.4.2 Bezeichner für Instanz, Haupt- und Dialogfenster :	105
6.5.4.3 Globale Datenstrukturen, Struktur für serielle Schnittstelle, Timing :.....	105
6.5.4.4 Bezeichnungen für die Elemente und Verbindungen :.....	105
6.5.4.5 Funktionsdefinitionen im Modul <i>Globals.cpp</i> :.....	105
6.5.5 Modul <i>Files.cpp</i> :	107
6.5.6 Modul <i>TTY.cpp</i> :.....	109
6.5.7 Modul <i>Timer.cpp</i> :.....	110
6.5.8 Modul <i>Dialogs.cpp</i> :.....	110
6.5.9 Modul <i>Draw.cpp</i> :.....	112
6.5.10 Modul <i>Midi.cpp</i> :	112
6.6 Module der Verarbeitungselemente	113
6.7 Signalverarbeitung durch die Konfigurationselemente.....	116
6.7.2 Realisierung des Zeitmanagements	117
6.8 Verarbeitung der Daten vom Biosignalverstärker	118
6.9 Implementierung der Gesichtsdetektion	120
6.9.1 Verwendete <i>OpenCV</i> -Funktionen.....	120
6.9.2 Initialisierung der Gesichtspose	120
6.9.3 Tracking von Gesichtsmarkmalen	121
6.10 Die Erzeugung eines neuen Verarbeitungselementes.....	121
6.10.1 Einträge in die Header-Datei <i>brainbay.h</i>	122
6.10.2 Erzeugung eines Benutzerdialogs	122
6.10.3 Klassenimplementierung für das Element	123
6.10.3.1 Die Header-Datei für das Element	123
6.10.3.2 Die Quellcode-Datei für das Element.....	124
6.10.4 Menüeintrag zur Einfügung des Elements in eine Konfiguration	128
7. Anwendung von <i>BrainBay</i>	130
7.1 Installation und erster Start des Programms	130
7.1.1 Systemvoraussetzungen	130
7.1.2 Download der aktuellen Programmversion.....	130
7.1.3 Verzeichnisstruktur	131
7.1.4 Einrichten der Applikation.....	131
7.2 Verwendung von Hauptmenü und Statuszeile.....	133
7.3 Der Farbpaletteneditor	134
7.4 Der Tonleitern-Editor.....	134

7.5 Die Erstellung von Design-Konfigurationen	135
7.5.1 Einfügen, Benennen und Löschen von Elementen.....	135
7.5.2 Verbindung von Elementen	136
7.5.3 Darstellung der Eigenschaften einer Verbindung	136
7.5.4 Änderung von Wertebereich und Signalbeschreibung.....	137
7.6 Verwendung der Konfigurationselemente.....	137
7.6.1 Signalquellen.....	137
7.6.1.1 Das <i>EEG</i> - Element	137
7.6.1.2 Das <i>Signal-Generator</i> - Element	139
7.6.1.3 Das <i>Constant</i> - Element	140
7.6.1.4 Das <i>EDF-FileReader</i> - Element.....	140
7.6.1.5 Das <i>TCP-Receive</i> - Element	141
7.6.1.6 Das <i>Camera</i> - Element (Webcam)	142
7.6.1.7 Das <i>Skindialog</i> - Element.....	143
7.6.2 Verarbeitungselemente	144
7.6.2.1 Das <i>Averager</i> - Element	144
7.6.2.2 Das <i>Comparator</i> - Element.....	144
7.6.2.3 Das <i>Correlation</i> - Element.....	145
7.6.2.4 Das <i>Counter / Display</i> - Element.....	145
7.6.2.5 Das <i>Debounce</i> - Element	145
7.6.2.6 Das <i>ERP-Detect</i> - Element.....	146
7.6.2.7 Das <i>Expression-Evaluator</i> - Element	147
7.6.2.8 Das <i>Filter</i> - Element.....	147
7.6.2.9 Das <i>FFT</i> - Element	148
7.6.2.10 Das <i>Integrator</i> - Element	149
7.6.2.11 Die <i>And, Or</i> und <i>Not</i> - Elemente.....	150
7.6.2.12 Das <i>Matlab</i> - Element.....	150
7.6.2.13 Das <i>Magnitude</i> - Element.....	151
7.6.2.14 Das <i>Mixer</i> - Element.....	151
7.6.2.15 Das <i>Sample/Hold</i> - Element	152
7.6.2.16 Das <i>Standard-Deviation</i> - Element.....	152
7.6.2.17 Das <i>Threshold</i> - Element.....	152
7.6.3 Signalziele	153
7.6.3.1 Das <i>AVI-Player</i> - Element	153
7.6.3.2 Das <i>Com-Writer</i> - Element	154
7.6.3.3 Das <i>EDF-FileWriter</i> - Element.....	154
7.6.3.4 Das <i>File-Writer</i> - Element.....	155
7.6.3.5 Das <i>Oscilloscope</i> - Element	155
7.6.3.6 Das <i>Particle-Animation</i> - Element	156
7.6.3.7 Das <i>Midi-Output</i> - Element.....	157
7.6.3.8 Das <i>Mouse-Controller</i> - Element.....	158
7.6.3.9 Das <i>Sound Player</i> - Element	159
7.6.3.10 Das <i>TCP-Sender</i> - Element.....	160
7.6.3.11 Das <i>Media-Player</i> - Element	161
7.6.4 Zusätzliche Elemente	161
7.6.4.1 Das <i>Documentation</i> - Element	161
7.6.4.2 Das <i>Ballgame</i> - Element.....	162

8. Anwendungsbeispiele	163
8.1 Eine Konfiguration für Alpha / Beta - Neurofeedback	163
8.2 Herzschlag-Rückmeldung aus einer Biosignalaufnahme.....	165
8.3 Eine Konfiguration für Muskel-Feedback und -rehabilitation.....	166
8.4 Steuerung eines Films durch Aktivität am Home-Trainer.....	168
8.5 Eine Kamera-Maus mit Klick-Funktionen.....	170
9. Evaluierung von Ergebnissen.....	172
9.1 Beurteilung der Erreichung von Designzielen.....	172
9.2 Bestehende Einschränkungen.....	173
9.2.1 Längerfristige Pläne zur Verbesserung der Applikation.....	174
9.3 Evaluation des Settings zur Muskelrehabilitation.....	175
9.3.1 Ergebnisse und Rückmeldungen.....	175
9.4 Evaluation der Kamera-Maus Konfiguration	176
9.4.1 Schreiben mit Switch Access to Windows (SAW).....	176
9.4.2 Test der Genauigkeit: Das Spiel 'Minesweeper' :	178
9.5 Zusammenfassung und Diskussion.....	178
9.6 Zukünftige Entwicklungen	179
A Die <i>BrainBay</i> - Homepage	180
A.1 Aufbau und Verfügbarkeit	180
A.2 Homepagezugriffe und Statistiken	181
Quellen der Abbildungen.....	182
Bibliographie.....	186

Abbildungsverzeichnis

Abb. 1.1: Elektroskop	3
Abb. 1.2: Leydener Flasche.....	3
Abb. 1.3: Erregung des Nerv-Muskel-Präparates	3
Abb. 1.4: Voltaische Säule	4
Abb. 1.5: Galvanometer.....	4
Abb. 1.6: Schematische Darstellung einer Nervenzelle	6
Abb. 1.7: Zellmembran einer Nervenzelle mit eingelagerten Membranproteinen.....	7
Abb. 1.8: Elektrisches und chemisches Potenzial an der Membran	8
Abb. 1.9: Bildung des elektrochemischen Gleichgewichtes.....	8
Abb. 1.10: Die Na/K-ATPase.....	9
Abb. 1.11: Verlauf eines Aktionspotenzials.....	11
Abb. 1.12: Aktivierung und Inaktivierung eines Natriumkanals	11
Abb. 1.13: Aktivierung und Inaktivierung eines Kaliumkanals.....	12
Abb. 1.14: Elemente der neuromuskulären Endplatte	13
Abb. 1.15: Transmitterfreisetzung an der neuromuskulären Endplatte.....	13
Abb. 1.16: Zentrales und peripheres Nervensystem	14
Abb. 2.1: Ableitung des EMG vom rechten Unterarm	16
Abb. 2.2: Ableitungspunkte des fazialen EMG	16
Abb. 2.3: EMG-Signalaufbereitung durch Mittelung von Absolutwerten	17
Abb. 2.4: EKG-Ableitung nach Goldberger bzw nach Einthoven.....	18
Abb. 2.5: Die sechs Ableitpunkte an der Brustwand nach Wilson	18
Abb. 2.6: Der Verlauf der Erregungsausbreitung im Herzen.....	19
Abb. 2.7: Die charakteristische Form des EKG	19
Abb. 2.8 Die Entstehung des EOG	20
Abb. 2.9: Messung des EOG, horizontale Blicke	20
Abb. 2.10: Elektrodenpositionen nach dem 10/20-System	21
Abb. 2.11: Desynchronisation des Alpha-Rhythmus durch das Öffnen der Augen	22
Abb. 2.12: Darstellung der verteilten Aktivität beim QEEG	22
Abb. 2.13: Die P300-Komponente:.....	23
Abb. 2.14: ERD durch Bewegungsvorstellung	24
Abb. 2.15: Langsame Potenziale bei der Vorbereitung einer Bewegung	24
Abb. 2.16: Typische Ableitungspunkte der EDA, Aufbau der Hautschichten.....	25
Abb. 2.17: Abnahme der Atmenaktivität durch dehnungssensitive Messgürtel	26
Abb. 2.18: Funktionsweise der Infrarot-Plethysmographie	27
Abb. 2.19: IROR-System der Firma Microguide.....	27
Abb. 2.20: Sensoren für Beschleunigung und Lageänderung	28
Abb. 2.21: Ableitungspunkte für das EMG der Rückenmuskulatur	30
Abb. 2.22: Ableitungspunkte für das EMG der Gesichts- und Nackenmuskulatur.....	31
Abb. 2.23: Lage wichtiger Muskelgruppen für die Kontinenzreaktion	34
Abb. 2.24: EMG-Sonde für das Beckenbodentraining	35
Abb. 2.25: Verbesserung der Bewegungsrezeption durch Biofeedback.....	36
Abb. 2.26: Dominante Theta-Aktivität im EEG	37
Abb. 2.27: IntegraMouse der Firma Lifetool.....	38
Abb. 2.28: Sicare-Pilot Umgebungssteuerung	39
Abb. 2.29: Steuerung eines Flugsimulators mit EMG-Signalen.....	40
Abb. 2.30: EagleEyes: Verwendung des Computers mittels EOG	40
Abb. 2.31: EOG-Elektroden und Beschleunigungssensoren in einer Brille	41
Abb. 2.32: HeadMouse extreme.....	42
Abb. 2.33: Tracker Pro	42
Abb. 2.34: Bewegungsverfolgung durch das <i>CameraMouse</i> -System	43
Abb. 2.35: Augendetektion in Echtzeit mit einer Webcam	43
Abb. 2.36: Tracking der Nase für die Cursorsteuerung.....	44
Abb. 2.37: Komponenten und Funktionsweise eines Gaze-Tracking-Systems	44
Abb. 2.38: Schreiben mit dem <i>EyeGaze</i> -System.....	45
Abb. 2.39: Desynchronisation des μ -Rhythmus bei der Vorstellung einer Bewegung.....	47
Abb. 2.40: Zweidimensionale Auswertung von ERD/ERS	47
Abb. 2.41: BCI-gesteuerte Orthese	48

Abbildungsverzeichnis (Fortsetzung)

Abb. 2.42: P300-BCI für das Buchstabieren	49
Abb. 2.43: Komponenten eines Direct-Brain-Interface.....	49
Abb. 2.44: Struktur und Komponenten der BCI2000-Applikation.....	50
Abb. 2.45: Beispiele für mögliche BCI-Applikationen in BCI2000	51
Abb. 2.46: Einstellungsmöglichkeiten der <i>ModularEEG</i> -Signalquelle für BCI2000	51
Abb. 3.1: Typische Elemente einer Messkette	52
Abb. 3.2: Unterschiede zwischen monopolarer und bipolarer Ableitung	53
Abb. 3.3: Der Elektrode-Haut-Kontakt].....	53
Abb. 3.4: Kapazitive Einkopplung von 50Hz-Noise.....	54
Abb. 3.5: Aufbau eines Differenzverstärkers.....	56
Abb. 3.7: Inadequate Samplingrate - Aliasing	58
Abb. 3.6: Ausreichende Samplingrate	58
Abb. 3.8: Tiefpassfilter (<i>Sallen-Key</i> -Schaltung).....	58
Abb. 3.9: Symbole für die Kennzeichnung zertifizierter Medizintechnikgeräte	59
Abb. 3.10: Noraxon Myotrace200.....	59
Abb. 3.11: Mindpeak WaveRider.....	59
Abb. 3.12: Brainmaster Atlantis	61
Abb. 3.13: Symptom EEG Verstärker.....	62
Abb. 3.14: Sagura Leonardo	62
Abb. 3.15: g.tec USBamp.....	63
Abb. 3.16: Neuroscan Synamp	63
Abb. 3.17: Digital-Board und Analog-Board des ModularEEG-Verstärkers.....	64
Abb. 3.18: Blockdiagramm der ModularEEG-Verstärkerstufen und DRL-Schaltung	65
Abb. 3.19: Blockdiagramm der Digitalisierung und Datenübertragung zum PC	66
Abb. 3.20: Der MonolithEEG-Verstärker	67
Abb. 3.21: Die SMD-Platine des MonolithEEG	67
Abb. 3.22: Montage der DIY - Silberlektroden für die EEG-Ableitung	69
Abb. 3.23: Komponenten und Einsatz der Messkappe	70
Abb. 3.24: Aktive Elektroden für das OpenEEG.....	71
Abb. 3.25: Bluetooth-Modul mit SPP	72
Abb. 3.26: VDrive-Modul von FTDI	72
Abb. 4.1: Stationäre und nichtstationäre Signale	75
Abb. 4.2: Histogramm und Wahrscheinlichkeitsdichte	75
Abb. 4.3: Frequenzantwort von Chebyshev- Butterworth- und Bessel-Filtern	76
Abb. 4.4: Sprungantwort von Chebyshev- Butterworth- und Bessel-Filtern.....	76
Abb. 4.5: Auswirkungen von Tiefpass- und Hochpassfiltern.....	78
Abb. 4.6: Diskrete Fourier Transformation	80
Abb. 4.7: Frequenzantwort eines Tiefpass-Filters in der Polardarstellung	81
Abb. 4.8: Aufbau eines Neuronalen Netzes mit 3 Schichten bzw. einer einzelnen Unit.....	82
Abb. 4.9: Klassifikation mit Support-Vektoren.....	83
Abb. 4.10: Einzelne Merkmale eines Haar-Classifiers	84
Abb. 4.11: Analyse von EEG-Daten mit EEGLab.....	86
Abb. 4.12: Scilab - eine freie Alternative zu Matlab	86
Abb. 4.13: 3D-Darstellung von EEG-Daten mit TEMPO	87
Abb. 4.14: Frequenzdarstellung mit BioExplorer.....	87
Abb. 4.15: FidView - Filter Designer von Jim Peters.....	88
Abb. 4.16: Wavelet-Analyse von EEG-Daten mit BwView	88
Abb. 4.17: BioEra Software für Biofeedback, Einsatz am PDA.....	89
Abb. 5.1: Funktion und prinzipieller Aufbau des geplanten Datenverarbeitungssystems.....	90
Abb. 5.2: Grafisches Design der Signalverarbeitung in Simulink.....	92
Abb. 5.3: Objektorientiertes Design, Bezug zur Elementklasse (Auszug)	96
Abb. 5.4: Entwicklungsumgebung Microsoft-VisualC++, Dialog-Editor	98
Abb. 5.5: Weiterleitung von Nachrichten in einer Windows-Applikation	101
Abb. 6.1: Designfenster, Anordnung von Elementen (1).....	116
Abb. 6.2: Designfenster, Anordnung von Elementen (2).....	116
Abb. 6.3: Tracking (links) und Neuinitialisierung (rechts) von Gesichtsmerkmalen	121
Abb. 6.4: Ressourcen-Editor	122
Abb. 6.5: Hinzufügen eines neuen Menüeintrages mit dem Ressourcen-Editor	128
Abb. 7.1: Benutzerdialog für Programmoptionen	132

Abbildungsverzeichnis (Fortsetzung)

Abb. 7.2: Hauptmenü und Design-Fenster der BrainBay-Applikation.....	133
Abb. 7.3: Statuszeile mit Informationen über die Session und zur Kontrolle der Wiedergabe	134
Abb. 7.4: Der Farbpaletten-Editor	134
Abb. 7.5: Der Tonleitern-Editor.....	135
Abb. 7.6: Einfügen von Elementen in die Design-Konfiguration	136
Abb. 7.7: Anzeige der Eigenschaften einer Elementverbindung.....	136
Abb. 7.8: Änderung von Signalbeschreibung und Wertebereich	137
Abb. 7.9: Parameterdialog des EEG-Elementes	138
Abb. 7.10: Wechsel der Position innerhalb einer Archivdatei mithilfe der Statuszeile.....	139
Abb. 7.11: Parameterdialog des Signal-Generator-Elementes.....	139
Abb. 7.12: Parameterdialog des Constant-Elementes	140
Abb. 7.13: Parameterdialog des EDF-FileReader-Elementes	140
Abb. 7.14: Laufender Neuroserver mit aufgebauter Verbindung	141
Abb. 7.15: Parameterdialog des TCP-Receive-Elementes.....	141
Abb. 7.16: Parameterdialog des Camera-Elementes, Live-Bild.....	142
Abb. 7.17: Parameterdialog des Skindialog-Elementes.....	144
Abb. 7.18: Parameterdialog des Averager-Elementes.....	144
Abb. 7.19: Parameterdialog des Comparator-Elementes	144
Abb. 7.20: Parameterdialog des Correlation-Elementes.....	145
Abb. 7.21: Parameterdialog des Counter-Elementes.....	145
Abb. 7.22: Parameterdialog des Debounce-Elementes	145
Abb. 7.23: Parameterdialog des ERP-Detect-Elementes	146
Abb. 7.24: Parameterdialog des Expression-Evaluator-Elementes	147
Abb. 7.25: Parameterdialog des Filter-Elementes	148
Abb. 7.26: Parameterdialog und Ausgabefenster des FFT-Elementes	148
Abb. 7.27: 3D-Darstellung durch das FFT-Element	149
Abb. 7.28: Parameterdialog des Integrator-Elementes	149
Abb. 7.29: Symbole der AND- OR- und NOT-Elemente	150
Abb. 7.30: Parameterdialog des Matlab-Elementes.....	150
Abb. 7.31: Parameterdialog des Magnitude-Elementes.....	151
Abb. 7.32: Parameterdialog des Mixer-Elementes.....	151
Abb. 7.33: Parameterdialog des Sample/Hold-Elementes.....	152
Abb. 7.34: Parameterdialog des Standard-Deviation-Elementes	152
Abb. 7.35: Parameterdialog und Ausgabefenster des Threshold-Elementes	152
Abb. 7.36: Parameterdialog und Ausgabefenster des AVI-Player-Elementes.....	153
Abb. 7.37: Parameterdialog des Com-Writer-Elementes	154
Abb. 7.38: Parameterdialog des EDF-FileWriter-Elementes.....	154
Abb. 7.39: Parameterdialog des FileWriter-Elementes	155
Abb. 7.40: Parameterdialog und Ausgabefenster des Oscilloscope-Elementes.....	156
Abb. 7.41: Parameterdialog und Ausgabefenster des Particle-Animation-Elementes.....	156
Abb. 7.42: Parameterdialog des Midi-Output-Elementes.....	157
Abb. 7.43: Parameterdialog des Mouse-Controller-Elementes.....	158
Abb. 7.44: Parameterdialog des Sound-Player-Elementes.....	159
Abb. 7.45: Lokal laufender Neuroserver mit aufgebauter Sendeverbindung	160
Abb. 7.46: Parameterdialog des TCP-Sender -Elementes	160
Abb. 7.47: Parameterdialog und Ausgabefenster des Media-Player-Elementes.....	161
Abb. 7.48: Parameterdialog des Documentation-Elementes	161
Abb. 7.49: Parameterdialog und Ausgabefenster des Ballgame-Elementes	162
Abb. 8.1: Ausgabefenster einer Konfiguration für das Neurofeedback von EEG-Frequenzbändern..	163
Abb. 8.2: Elemente der Konfiguration für das Neurofeedback von EEG-Frequenzbändern	164
Abb. 8.3: Filterantwort des 50Hz-Bandstop-Resonators.....	164
Abb. 8.4: 3D-Darstellung der Frequenzbänder	164
Abb. 8.5: Ausgabefenster der Konfiguration für das optoakustische EKG-Feedback	165
Abb. 8.6: Elemente der Konfiguration für das optoakustische EKG-Feedback	165
Abb. 8.7: Thresholds zur Erkennung der R-Zacke im EKG	166
Abb. 8.8: Einsatz von EMG-Feedback zum Training der Muskelkontraktion.....	166
Abb. 8.9: Elemente der Konfiguration für das Muskeltraining	167
Abb. 8.10: Filtereinstellung für die Muskelamplitude	167

Abbildungsverzeichnis (Fortsetzung)

Abb. 8.11: Darstellung einer Muskelkontraktion als 3D-Fourier-Analyse.....	167
Abb. 8.12: Laufendes Muskeltraining, Zählen und Aufsummieren der Kontraktionen.....	168
Abb. 8.13: Schaltkontakt zur Ermittlung der Drehgeschwindigkeit.....	168
Abb. 8.14: Konfiguration zur Steuerung eines Filmes durch die Aktivität am Schaltkontakt	169
Abb. 8.15: Die Konfiguration im Einsatz - eine Überlandfahrt, ausgeführt im Wohnzimmer	169
Abb. 8.16: Design-Konfiguration für die Kamera-Maus.....	170
Abb. 8.17: Test der Kamera-Maus mit dem Spiel Minesweeper.....	171
Abb. 9.1: Verwendung von Switch Access for Windows mit der Kamera-Maus.....	177
Abb. A.1: Die BrainBay-Homepage, erreichbar unter http://brainbay.io-res.org	180
Abb. A.2: Geographischer Ursprung der Zugriffe auf die Homepage	181
Abb. A.3: Zugriffsstatistik nach Ländern.....	181

Tabellen

Tabelle 1.1: Ionenkonzentrationen im Intra- und Extrazellulärraum	7
Tabelle 1.2: Permeabilität der Zellmembran für verschiedene Arten von Ionen.....	7
Tabelle 3.1: Beispiele für Hardwareanforderungen der IEC-Norm 60601-1-2.....	59
Tabelle 3.2: Technische Daten <i>Noraxon Myotrace200</i>	60
Tabelle 3.3: Technische Daten <i>Mindpeak WaveRider</i>	61
Tabelle 3.4: Technische Daten <i>Brainmaster Atlantis</i>	61
Tabelle 3.5: Technische Daten <i>Symtop-EEGAmplifier</i>	62
Tabelle 3.6: Technische Daten <i>Sagura Leonardo</i>	62
Tabelle 3.7: Technische Daten <i>g.Tec USBamp</i>	63
Tabelle 3.8: Technische Daten <i>Neuroscan Synamp</i>	63
Tabelle 3.9: <i>ModularEEG</i> , technische Spezifikationen	66
Tabelle 3.10: <i>MonolithEEG</i> , technische Spezifikationen.....	67
Tabelle 3.11: Übertragungsprotokolle <i>P2</i> und <i>P3</i> der OpenEEG-Firmware	68
Tabelle 6.1: Source-Module der Brainbay-Applikation.....	115
Tabelle 9.1: Ergebnisse für das Schreiben mit der Kamera-Maus unter Verwendung von SAW.....	177
Tabelle 9.2: Ergebnisse von Minesweeper mit der Kamera-Maus.....	178

Gleichungen

Gl. 1.1: Nernst - Gleichung (allgemein).....	8
Gl. 1.2: Elektrisches Potenzial.....	9
Gl. 1.3: Nernst - Gleichung (in der Biologie).....	9
Gl. 3.1: Gleichtaktunterdrückung am Differenzverstärker	54
Gl. 4.1: Homogenität eines linearen Systems	73
Gl. 4.2: Additivität eines linearen Systems.....	74
Gl. 4.3: Berechnung des Erwartungswertes.....	74
Gl. 4.4: Berechnung der Standardabweichung	74
Gl. 4.5: Berechnung der diskreten Faltung.....	77
Gl. 4.6: Berechnung der diskreten Kreuzkorrelation	79
Gl. 4.7: Orthogonale Basisfunktionen der DTF	80
Gl. 4.8: Diskrete Fourier Transformation	81
Gl. 4.9: Inverse Diskrete Fourier Transformation	81
Gl. 4.10: Umwandlung der Fourier-Koeffizienten in die Polardarstellung	81

1. Bioelektromagnetismus und Elektrophysiologie

Die wissenschaftliche Disziplin des *Bioelektromagnetismus* beschäftigt sich mit den elektrischen und elektromagnetischen Eigenschaften lebender Gewebe. Diese Eigenschaften sind insbesondere für die Medizin von großer Wichtigkeit, da die bioelektrischen Phänomene an der Zellmembran für die Aufrechterhaltung der Vitalfunktionen essentiell sind. Die Weiterleitung von elektrischen Impulsen entlang von Nervenzellen ist eine Voraussetzung für die Wahrnehmung von Sinneseindrücken, das Zusammenspiel der Körperfunktionen und für die Aufrechterhaltung der *Homöostase*. Die Selbstregulation vitaler Prozesse, die Steuerung des hormonellen Systems und die Reaktion auf Umwelteinflüsse sind nur durch den effizienten Austausch chemoelektrischer Information durch das Nervensystem möglich.

Das theoretische Instrumentarium für ein Verständnis und eine wissenschaftliche Analyse der bioelektrischen Phänomene bilden die Physik, die Chemie und die mathematischen Sätze der Elektrotechnik. Viele Erkenntnisse über chemische und elektromagnetische Zusammenhänge wurden dabei durch die Entdeckung und Erforschung bioelektrischer Phänomene gewonnen. *Bioelektromagnetismus* ist daher eine interdisziplinäre Wissenschaft, welche Erklärungsmodelle aus verschiedenen Bereichen der Naturwissenschaft kombiniert und diese durch ihre Erkenntnisse auch stark beeinflusst hat. Bioelektromagnetismus stellt heute ein vielfältiges Gebiet der Forschung dar, das durch die kostengünstige Verfügbarkeit leistungsstarker, portabler Mikroelektronik zahlreiche Anwendungen in unserem Alltag gefunden hat.

Bioelektromagnetismus behandelt prinzipiell elektrische, magnetische und elektromagnetische Phänomene. Die Verbundenheit dieser Erscheinungsformen wird durch die Maxwell-Gleichungen symbolisiert, welche den Zusammenhang von zeitlich veränderlichen elektrischen und magnetischen Feldern zeigen.

Aufgrund der verschiedenen Frage- und Aufgabenstellungen kann das Gebiet des Bioelektromagnetismus auch in folgende Bereiche gegliedert werden [MAL95]:

Messung von elektromagnetischen Feldern des biologischen Gewebes:

Hierzu zählen *EKG*, *EEG*, die Messung von Muskelaktivität (*EMG*) sowie das *Magnetenzephalogramm (MEG)*. Diese Messmethodik wurde vor allem im 20. Jhdt. stark verbessert, sodass Untersuchungen der Eigenschaften biologischen Gewebes auf organischem sowie auf zellulärem und sub-zellulärem Niveau möglich wurden. Die gewonnenen Daten finden Anwendung in Physiologie und Pathologie, in der Molekularbiologie, in der Entwicklung von Pharmazeutika sowie in bildgebenden Verfahren.

Stimulation von vitalem Gewebe durch elektrische oder magnetische Felder:

Die Elektrotherapie ist eine seit den Anfängen der Medizin bekannte Methode, bei der meist Muskel- oder Nervenzellen durch kontrollierte Stromimpulse stimuliert werden. Schrittmacher, Defibrillatoren, Reizstromtherapie und funktionelle Elektrostimulation (*FES*) sind moderne Anwendungsgebiete dieser Technik. Durch die Entwicklung implantierbarer Schrittmacher gelang binnen weniger Jahrzehnte eine Erhöhung der Lebensqualität für sehr viele Menschen.

Impedanzmessung von Gewebeeigenschaften:

Hier werden, wie bei der elektromagnetischen Stimulation, externe Felder in biologisches Gewebe eingebracht, und dessen Leitfähigkeit durch geeignete Messmethoden geprüft. Die Impedanzmessung kann, so wie die Messung des Magnetfeldes, berührungsfrei durchgeführt werden und gibt Aufschluss über funktionelle Eigenschaften des Gewebes. Anwendungen sind unter anderem die *Impedanztomographie*, die *Magnetresonanztomographie (MRT)* oder die Messung des Hautleitwertes (*Electrodermal Response, EDR*)

Laut den Maxwell'schen Materialgleichungen sind diese drei Bereiche durch das Prinzip der Reziprozität verbunden, welches besagt, dass Gebiete mit hoher Sensitivität für elektrische bzw. magnetische Messung in gleichem Maße elektromagnetisch erregbar sind [MAL95].

Die vorliegende Arbeit beschäftigt sich hauptsächlich mit der Messung bioelektrischer Signale mit Hilfe von Elektroden und geeigneten Verstärkern, weil dadurch ein breites Spektrum von Vitalparametern vergleichsweise einfach und nichtinvasiv erschlossen werden kann. Durch eine flexible Software für die Signalverarbeitung entsteht ein kostengünstiges System, dessen Fähigkeiten von der Visualisierung der Messwerte in Echtzeit über die Extraktion und Verarbeitung von Signalcharakteristika bis zur Telemetrie der Daten reicht.

1.1 Geschichtliche Entwicklung des Bioelektromagnetismus

Der Blitz hat die Menschen als gewaltiges Naturphänomen sicherlich schon in der Urzeit beschäftigt. Erste Aufzeichnungen über Bioelektromagnetismus finden sich 4000 v. Chr.: Ägyptische Hieroglyphen beschreiben einen elektrischen Fisch, vermutlich einen Wels [MAL95]. Heute sind mehrere Fischarten bekannt, die starke elektrische Stromstöße abgeben können: der Torpedorochen *Torpedo panthera*, die Welse *Malapterurus* und *Astroscopus* ("Sternenstarrer") und der Zitteraal *Electrophorus electricus*, von dem Stromstöße bis knapp 900 V gemessen wurden. Das elektrische Organ besteht aus Zellen, die in einigen tausend in Serie geschalteten Säulen, den so genannten *Elektroplaques*, organisiert sind. Eine Säule kann ca. 150 mV erzeugen.

Die Entdeckung der elektrischen Eigenschaft mancher Fische stellt auch den Ursprung der Elektromedizin dar: Scribonius Largus empfahl im Jahre 46 n. Chr. die Anwendung von Stromstößen der Fische bei Kopfschmerzen und Arthritis. Weitere Beschreibungen von elektromagnetischen Quellen beschränken sich bis ins 17. Jhdt. auf Experimente mit Bernstein, welcher durch Reibung an einem trockenen Stoff eine elektrostatische Aufladung erfährt, und magnetischem Eisenerz (Magnetit, benannt nach seinem Fundort im Norden Griechenlands, *Magnesia*) [MAL95].

**Das Studium der Elektrizität wurde ab ca. 1600 systematisch betrieben :
[MAL95, DAV06]**

William Gilbert konstruierte um 1600 das erste Messgerät zur Detektion der Anziehungskraft von aufgeladenem Bernstein: eine leichte, drehbare Metallnadel, die sich zur Quelle dieser Kraft ausrichtet. Er nannte die aufladbaren Substanzen *electricks*, nach dem griechischen Wort für Bernstein (*Elektron*), und das Gerät *Elektroskop* (Abb. 1.1).

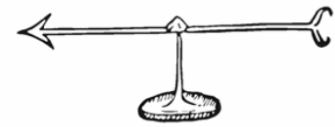


Abb. 1.1: Elektroskop

Otto v. Guericke beschrieb 1665 die Erscheinungen der elektrischen Aufladung und Leitung anhand einer Schwefelkugel, ohne jedoch die Ursachen zu kennen. 1672 erfand v. Guericke die "*Elektriermaschine*", welche durch Drehung und Reibung einer Schwefelkugel elektrische Ladungen trennte und Funken erzeugte.

Ewald v. Kleist und *Pieter van Musschenbroek* erfanden um 1745 das *Leydengefäß* (die "*Leydener Flasche*", Abb. 1.2), den ersten Kondensator. Mit der Leydener Flasche konnte elektrische Ladung gesammelt, verstärkt, aufbewahrt und transportiert werden. Die Flasche besteht aus einem oben offenen zylindrischen Nichtleiter (ein Glasgefäß) und trägt auf seiner Innen- und Außenfläche gegeneinander isoliert zwei leitende Schichten. Die innere Schicht ist zum Aufladen mit einer Metallstange verbunden, die nach außen ragt. Wird die Flasche über die Stange negativ aufgeladen, so entsteht durch Influenz am äußeren Belag eine positive Ladung. Die verdrängten Elektronen entweichen in die Erde, wenn die Flasche nicht isoliert steht.

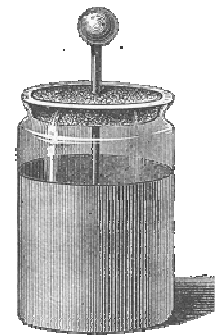


Abb. 1.2: Leydener Flasche

"Damals war es beliebt, einer Menschenkette einen elektrischen Schlag (den so genannten Kleist'schen Stoß) zu versetzen. Man erfreute sich an den zuckenden und springenden Opfern." [ERX1797]. In einem Physiklehrbuch bemerkt Lichtenberg dazu: *"Zu Paris glaubte man vor einigen Jahren gefunden zu haben, dass der Stoß immer 'bey frigidis et impotentibus' aufhöre. Der Graf von Artois, der davon hörte, berief dazu die Castraten der Oper; und man fand die Beobachtung falsch. Auf diese Weise ist die Elektriermaschine um die Ehre gekommen, dereinst als ein nützliches Instrument in den Versammlungssälen der Consistorien und Ehegerichte zu prangen."* [ERX1797]

Leopoldo Caldani (Bologna, Lehrer von *Luigi Galvani*) zeigt 1756, dass ein isoliertes Nerv- Muskel-Präparat (meist der *M. gastrocnemius* und *N. ischias* vom Frosch) durch den Stromstoß eines Leydengefäßes erregt werden kann

Luigi Galvani (Bologna) erweitert und vertieft diesen Befund um 1780 systematisch durch Versuche mit der Elektriermaschine und biologischem Muskelgewebe, es entsteht der Begriff der "*tierischen Elektrizität*". Galvani stellte fest, dass sich Elektrizität in einer Nervenzelle ausbreitet, wenn sie auf einen erregten Muskel gelegt wird. Er entdeckte weiters, dass das Nerv-Muskel-Präparat erregt wird, wenn es gleichzeitig Kontakt mit zwei verschiedenen Metallarten bekommt: die Stromerzeugung durch eine bimetallische Verbindung (siehe Abb. 1.3).

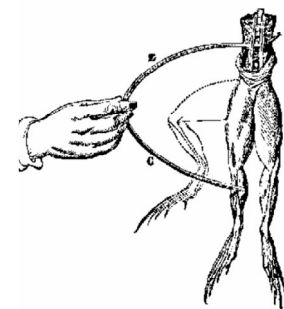


Abb. 1.3: Erregung des Nerv-Muskel-Präparates

Alessandro Volta (Padua) erfindet um 1800 die Batterie (die voltaische Säule, siehe Abb. 1.4). Durch eine Kombination mehrerer Schichten aus Zink und Kupfermetallplättchen, die durch elektrolytische Schichten aus Papier oder Leder getrennt waren, wurde eine kontinuierliche Gleichstromquelle aufgebaut. Volta präsentierte seine Erfindung als "künstliches elektrisches Organ" und löste damit Galvani's Theorie der "tierischen Elektrizität" als exklusive Erscheinungsform weitgehend ab. Die voltaische Säule legte den Grundstein für weitere Entdeckungen in den Bereichen Elektrochemie, Elektromagnetismus und Elektrophysiologie.

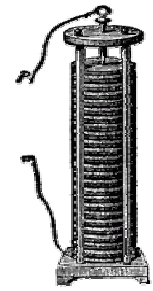


Abb. 1.4: Voltaische Säule

Hans Christian Ørsted (Kopenhagen) beschreibt 1820 das magnetische Feld um einen stromtragenden elektrischen Leiter. Aus dieser Entdeckung und den Arbeiten von *Ampère*, *Schweigger* und *Poggendorff* entsteht das erste elektrische Ladungsmeßgerät, der Galvanometer.

1825 benutzt *Leopoldo Nobili* (Florenz) seinen verbesserten Galvanometer um Ladungsströme im Nerv-Muskel-Präparat zu zeigen: Durch Anordnung zweier magnetischer Nadeln in entgegengesetzter Richtung und Verwendung einer 8-förmigen Spule konnte er Störungen durch das Erdmagnetfeld ausgleichen und eine höhere Messgenauigkeit erreichen (siehe Abb. 1.5). *Ampere* postuliert 1827 seine Theorie der Elektrodynamik.

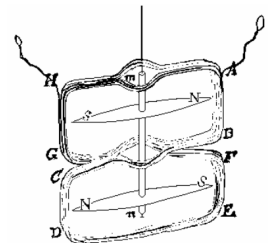


Abb. 1.5: Galvanometer

Emil Du Bois Reymond (Berlin) zeigt 1843, dass ein Muskel in Ruhe auch eine Ladung aufweist (= *Ruhepotenzial*), die bei der Kontraktion weniger wird. Diese Potenzialänderung nennt er "*Aktionspotenzial*". Er erfindet auch den "Stromschneider" (*Rheotom*) mit dem zeitlich begrenzte elektrische Stimuli erzeugt werden können. Der Stimulusartefakt kann nun zum ersten Mal von der physiologischen Antwort getrennt werden (Artefaktunterdrückung), was die ersten Analysen des Zeitverlaufs von Erregungsvorgängen ermöglicht. Diese Entdeckungen legten den Grundstein für die Messung von EEG, EKG und EMG.

Hermann Helmholtz trug durch seine Erkenntnisse zur Zell- und Sinnesphysiologie, den Energieerhaltungssatz und das Reziprozitäts-Theorem (1852) zum Verständnis bioelektrischer Phänomene bei.

James Clark Maxwell (Edinburgh, Cambridge) postuliert 1865 ein Reihe von Gleichungen, die zeitlich veränderliche elektrische Ströme und Elektromagnetismus verbinden.

Richard Caton (Liverpool) registrierte 1875 erstmals elektrische Aktivität in der Hirnrinde von Tieren.

1884 stellt *Svante Arrhenius* (Schweden) die Hypothese auf, dass die Moleküle bestimmter Substanzen durch ihre Lösung in Wasser in positiv oder negativ geladene Ionen zerfallen und so die Leitung elektrischen Stromes ermöglichen.

Augustus Waller (London) benutzt 1887 den von *Gabriel Lippmann* entwickelten Kapillarelektrometer, um durch Herztätigkeit bedingte Spannungsänderungen an der Körperoberfläche aufzuzeichnen: das erste Elektrokardiogramm entstand. *Willem Einthoven* verbessert 1895 die Kapillarelektrometerkurvenanalyse durch mathematische Berechnungen und beobachtet 5 Phasen, die er P, Q, R, S und T nennt.

1889 formuliert *Walter Hermann Nerst* (Deutschland) ein System von Gleichungen, welches das Verhalten von Elektrolyten in Gegenwart elektrischer Ströme beschreibt.

Julius Bernstein (Heidelberg) gilt als der Begründer der Membran-Theorie, er fand 1902 die selektive Durchlässigkeit der Zellmembran von Nervenzellen für bestimmte Ionen.

1924 wird vom deutschen Neurologen *Hans Berger* (Jena) erstmals das EEG eines Menschen aufgezeichnet. Er publizierte seine Ergebnisse 1929 unter dem Titel "*Über das Elektrenkephalogramm des Menschen*", erst 1934 fanden die Entdeckungen konkrete Anwendungen durch die Forschungen des englischen Neurophysiologen *Edgar Douglas Adrian* (Cambridge).

1949 entwickelte *Kenneth Cole* (Chicago) die *Voltage-Clamp* - Methode. Diese entkoppelt die Funktion bestimmter Typen von Ionenkanälen in der Membran von Nervenzellen, indem ein Kompensationsstrom jeder Änderung entgegenwirkt. Durch den Einsatz spezieller Gifte können Typen von Ionenkanälen blockiert und so der Stromfluss durch die nicht blockierten Kanäle gemessen werden. *Thetrodotoxin (TTX)*, das Gift des Pufferfisches, wurde zur Blockierung von Na-Kanälen verwendet.

Alan Hodgkin und *Andrew Huxley* (Cambridge) untersuchten die Vorgänge an der Zellmembran sehr genau und entwarfen 1952 ein mathematisches Modell für die Entstehung der Aktionspotenziale (siehe Kapitel 1.3).

1963 gelang *Gerhard M. Baule* und *Richard McFee* die erste Ableitung eines biomagnetischen Signals vom Herzen (*Magnetokardiogramm, MCG*). Sie verwendeten dabei dasselbe Prinzip zur Minderung magnetischer Artefakte wie *Leopoldo Nobili* 1825. Eine erhebliche Leistungssteigerung der Magnetfeldmessungen gelang *Zimmermann, Thine* und *Harding* 1970 durch die Verwendung supraleitender Materialien in der Spulentechnik (*Superconduction Quantum Interference Device, SQUID*). *David Cohen* gelang damit 1972 erstmals die Messung des Magnetenzephalogramms (*MEG*).

Erwin Neher und *Bert Sakmann* (Göttingen) entwickelten 1976 die *Patch-Clamp Technik*, mit Hilfe derer sich Ströme von einzelnen Ionenkanälen messen lassen. Bei der *Patch-Clamp-Technik* wird der Strom durch eine sehr kleine, wenige μm^2 große Fläche (*Patch*) der Membran gemessen, an die eine vorgegebene Spannung angelegt wird (*Clamp*). In dem *Patch*, dem Membranstückchen, sind nur wenige Kanäle, z.B. ein einziger. Die Isolierung dieses kleinen Stücks wird durch eine Glaspipette erreicht. Betrachtet man den Verlauf des Stromes, so sieht man abrupte Sprünge zwischen zwei verschiedenen Niveaus: Der Kanal öffnet und schließt sich. Der gemessene Strom liegt in der Größenordnung von einigen Picoampere (pA). Zu dessen Messung ist daher ein sehr empfindlicher Messverstärker notwendig.

1.2 Physiologische und anatomische Grundlagen - Aufbau und Funktion der Nervenzelle

Die Nervenzelle (Neuron) ist die strukturelle Einheit des Nervensystems. Neuronen kommunizieren untereinander oder mit *Effektorzellen* durch elektrochemische Prozesse, die durch Konzentrationsunterschiede von Ionen im Intra- und Extrazellulärraum ermöglicht werden.

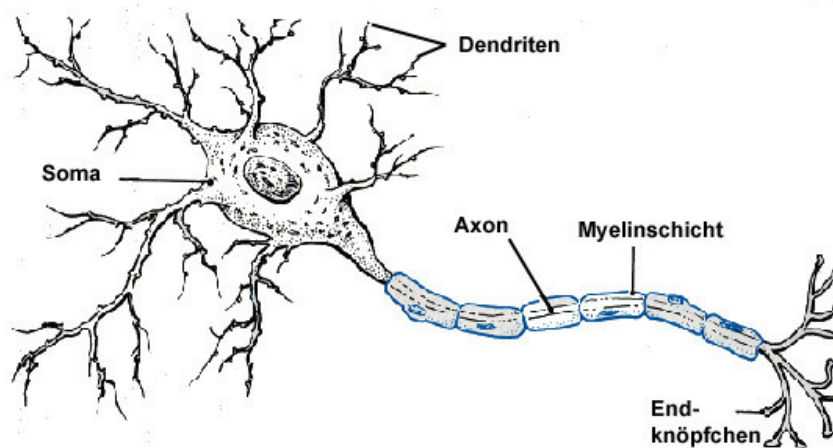


Abb. 1.6: Schematische Darstellung einer Nervenzelle

Eine Nervenzelle besteht aus dem Zellkörper (*Soma* oder *Perikaryon*), welcher den Zellkern (*Nucleus*) und die DNA beinhaltet, den *Dendriten*, die dem Empfang von Impulsen anderer Nervenzellen dienen, und dem *Axon* (*Neurit*), dem Hauptfortsatz des Neurons zur Weiterleitung von Impulsen (siehe Abb.1.6). Das Axon kann eine Länge von einigen Millimetern bis zu einem Meter haben. Das Axon ist durch die *Myelinschicht*, welche von den *Schwann-Zellen* produziert wird, umgeben. Die Myelinisierung dient einer schnelleren (sprunghaften, *saltatorischen*) Fortleitung der elektrischen Potenziale entlang des Axons. Am Ende des Axons befinden sich zahlreiche Verästelungen, an denen die Endknöpfchen sitzen. Diese liegen beinahe an der Oberfläche anderer Nerven- oder Muskelzellen auf und bilden die *Synapsen*. Die synaptische Übertragung erfolgt mittels Transmitterstoffen. Die Transmitter werden dabei von der sendenden Nervenzelle ausgeschüttet und vom empfangenden Neuron aufgenommen, wo sie zu einer Erregung oder Hemmung der Signalfortleitung beitragen [SEE06].

1.2.1 Die Vorgänge an der Zellmembran

Die Zellmembran besteht aus einer *Doppellipidschicht* und stellt eine effiziente Diffusionsbarriere für Ionen dar. Der hydrophobe innere Bereich der Membran verhindert den Durchgang von Ionen und erhält so einen Konzentrationsunterschied (*Ionengradienten*) zwischen Innenraum (*Zytoplasma*) und Außenlösung aufrecht. Kanal- und Transportproteine, die in der Membranwand eingelagert sind, ermöglichen die Diffusion bestimmter Ionen oder den aktiven Transport von Ionen gegen das Konzentrationsgefälle (siehe Abb. 1.7) [FRI98].

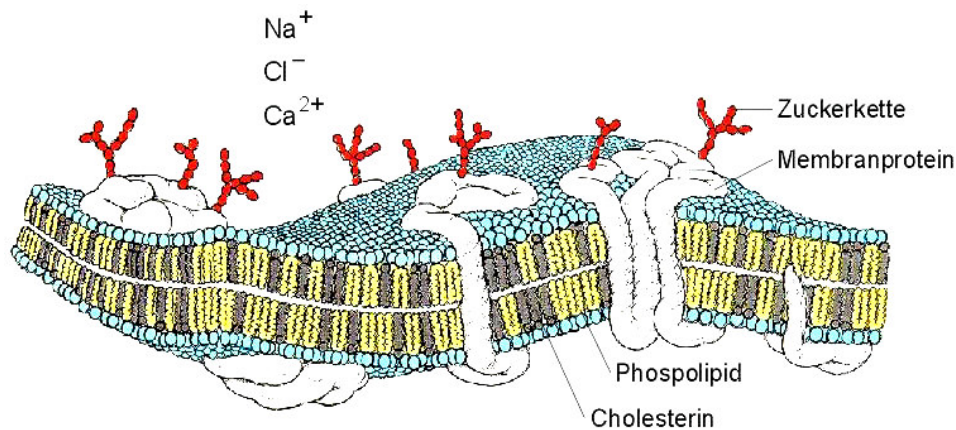


Abb. 1.7: Zellmembran einer Nervenzelle mit eingelagerten Membranproteinen

Die wichtigsten Ionen für den zellulären Signalaustausch sind Natrium (Na^+), Kalium (K^+), Calcium (Ca^{2+}) und Chlorid (Cl^-). Die folgende Tabelle zeigt typische Konzentrationen im Innen- und Außenraum der Zelle:

Konzentration (mMol / Liter)	Na^+	K^+	andere Kationen	Cl^-	organ. Anionen
Intrazellulär	12	155	0	4	155
Extrazellulär	145	4	5	120	0

Tabelle 1.1: Ionenkonzentrationen im Intra- und Extrazellulärraum

Die Ionenverhältnisse lassen sich wie folgt charakterisieren:

Im Außenraum sind viele Natrium- und Chlorid-Ionen gelöst, es handelt sich praktisch um eine verdünnte Kochsalzlösung. Im Zytoplasma hingegen befinden sich viele Kalium-Ionen und organische Anionen (z.B. *Malat* und *Oxalacetat*), die an Stoffwechselvorgängen wie dem *Citratzyklus* beteiligt sind [HEL06].

In diesem System existieren also zwei verschiedene Kräfte:

- Chemische Kräfte, verursacht durch die unterschiedlichen Ionenkonzentrationen, die auf einen Ausgleich der Konzentrationsverhältnisse hinwirken.
- Elektrische Kräfte, die durch ein Ungleichgewicht von positiv geladenen Kationen und negativ geladenen Anionen entstehen

Hinzu kommt, dass die Zellmembran durch eingelagerte Kanalproteine für bestimmte Ionen (vor allem für Kalium) durchlässig ist, während große Ionen schwerer oder gar nicht diffundieren können:

Permeabilität der Membran für Ionen	Na^+	K^+	Cl^-	organ. Anionen
Größe der hydratisierten Ionen	$12 \cdot 10^{-10} \text{ m}$	$7 \cdot 10^{-10} \text{ m}$	$8 \cdot 10^{-10} \text{ m}$	$>15 \cdot 10^{-10} \text{ m}$
relative Permeabilität	4	100	45	120

Tabelle. 1.2: Permeabilität der Zellmembran für verschiedene Arten von Ionen

1.2.2 Die Entstehung des Ruhemembranpotenzials

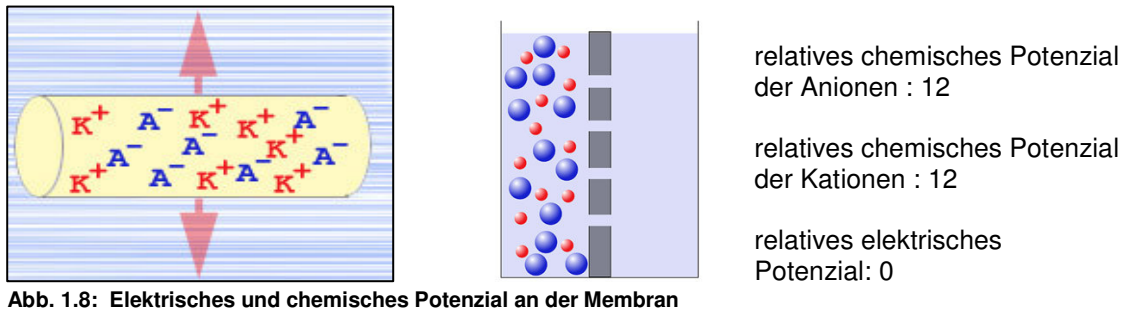


Abb. 1.8: Elektrisches und chemisches Potenzial an der Membran

Abb. 1.8 zeigt idealisiert die Konzentrationsverhältnisse von Kalium-Kationen und organischen Anionen im Intrazellulärraum. Das elektrische Potenzial ist ausgeglichen, es wirken keine elektrischen Kräfte. Das chemische Potenzial strebt jedoch nach einem Austausch mit dem extrazellulären Milieu [HEL06].

Durch die Eigenschaften der semipermeablen Zellmembran können die kleinen Kalium-Kationen durch die Zellwand diffundieren, nicht jedoch die größeren Anionen. Es entsteht ein Ausgleich der Kalium-Konzentration, der durch die Änderung der elektrischen Verhältnisse zu einer Polarisierung der Zellmembran führt:

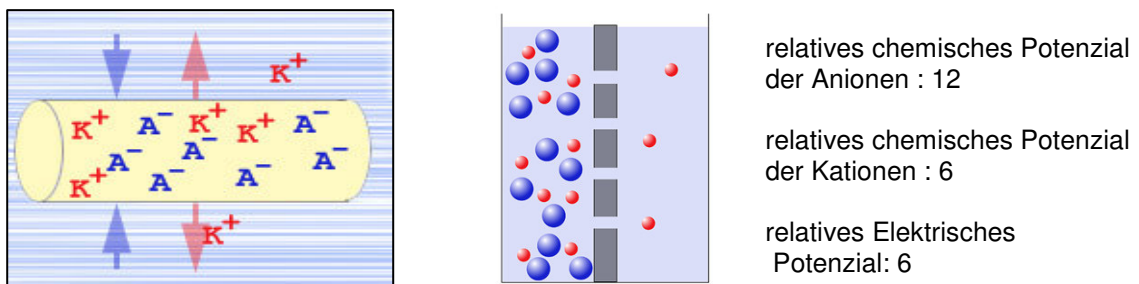


Abb. 1.9: Wanderung kleiner Ionen durch die semipermeable Membran, Bildung des elektrochemischen Gleichgewichtes

Es bildet sich ein Zustand des elektrochemischen Gleichgewichtes, bei dem das intrazelluläre negative elektrische Potenzial und das chemische Potenzial durch den verbleibenden Konzentrationsgradienten gleich groß sind (siehe Abb. 1.9) [HEL06].

1.2.3 Berechnung des Ruhemembranpotenzials: Die Nernst Gleichung

Das chemische Potenzial berechnet sich nach Walter Nernst wie folgt [SEE06]:
 (Gl. 1.1)

$$w_c = \frac{R \cdot T}{z_e \cdot F} \cdot \ln \frac{c_{\text{außen}}}{c_{\text{innen}}}$$

R ... molare Gaskonstante $8,31447 \text{ J mol}^{-1} \text{ K}^{-1}$
 T ... absolute Temperatur in Kelvin
 F ... Faraday-Konstante $96485,34 \text{ C mol}^{-1}$
 z_e ... Valenzzahl der Elektronen
 c ... Konzentration

Das elektrische Potenzial ist gegeben durch:
(Gl. 1.2)

$$w_e = F * E$$

w_e ... elektrisches Potenzial
 F ... Faraday Konstante
 E ... elektrische Spannung

Werden diese beiden Potenziale gleichgesetzt, und R , T und F zu einer Konstanten kombiniert, erhält man eine Adaption der Nernst-Gleichung für die Biologie:
(Gl. 1.3)

$$E = k * \log \frac{c_{\text{außen}}}{c_{\text{innen}}}$$

k ... Konstante (61,54 mV bei 37 °C)
 c ... Konzentration

Bei der angenommenen Konzentration von Kalium-Ionen von 4:155 errechnet sich ein Ruhemembranpotenzial von -97,74 mV, was in etwa den physiologischen Gegebenheiten entspricht.

1.2.4 Erhaltung des Gleichgewichtspotenzials: die Ionenpumpen

Durch die Diffusionsprozesse kommt es zu einem permanenten Austausch von Ionen durch die Zellmembran. So diffundieren einige Natrium-Ionen durch die Membran (*Natrium-Leckströme*). Da sie positiv geladen sind, neutralisieren sie die negative Ladung im Zellinneren und schwächen so das Ruhepotenzial ab. Als Folge davon verlassen Kalium-Ionen das Innere der Nervenzelle, um den Gleichgewichtszustand wiederherzustellen. Auf Dauer würde die Zelle Kalium-Ionen verlieren und dadurch das Ruhepotenzial abbauen.

Die *Natrium-Kalium-Ionenpumpe (Na/K-ATPase)* ist ein Protein, welches dafür sorgt, dass die Natrium-Ionen wieder nach außen und die Kalium-Ionen wieder nach innen transportiert werden (siehe Abb. 1.10). So wird das Ruhepotenzial aufrechterhalten. Dazu wird von der Ionenpumpe Stoffwechselenergie in Form des instabilen Moleküls *Adenosintriphosphat (ATP)* verbraucht. Durch Spaltung eines ATP-Moleküles kann die Na/K-ATPase drei Na-Ionen nach außen und zwei K-Ionen ins Zellinnere befördern. Es handelt sich um einen aktiven Transport von Ionen entgegen den Konzentrationsgradienten. [SEE06]

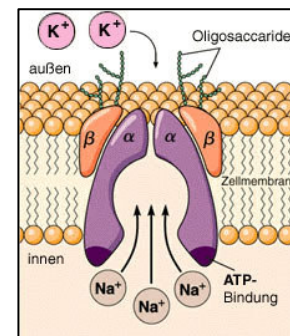


Abb. 1.10: Na/K-ATPase

1.2.5 Funktionale Einheiten in der Zellmembran: die Ionenkanäle

Ionenkanäle sind Proteine, die durch die Anwesenheit bestimmter chemischer Substanzen oder elektrischer Spannung ihre Konformation ändern können und dadurch Ionen in das Zellinnere oder aus dem Zellinneren heraus wandern lassen. Die membrandurchspannenden (*transmembranalen*) Segmente des Proteins enthalten viele hydrophobe und wenige hydrophile Aminosäurereste. Außerhalb der Membran

befinden sich überwiegend geladene (*polare*) Aminosäuren. Die Raumstruktur eines Ionenkanals ist von entscheidender Bedeutung für seine Funktion. Ein Beispiel ist der Mechanismus des Kalium-Kanals: Von Wassermolekülen umgebene (*hydratisierte*) K^+ -Ionen gelangen vom Zytoplasma her in die Pore bis zu einer Höhlung in der Mitte des Kanalproteins. In die engste Stelle der Pore, den so genannten *Selektivitätsfilter*, gelangen die Ionen nur nach Abstreifen der Wassermoleküle. In den beiden Bindestellen des Selektivitätsfilters kommt es zu einer elektrostatischen Wechselwirkung von Ion und polaren Gruppen des Proteins. Diese Bindung ersetzt die Bindung an Wassermoleküle und ermöglicht es dem Ion, die Pore zur Zellaußenseite hin zu verlassen [FRI98].

1.2.6 Eigenschaften von Kanalproteinen

Kanalproteine können zwei Konfigurationen annehmen: den offenen und den geschlossenen Zustand. Im offenen Zustand ist die Ionenbewegung durch die Pore des Proteins möglich. Ein bestimmter Abschnitt des Proteins (*Proteindomäne*) kann die Pore versperren und öffnen. Dieser Abschnitt heißt *Gate*, das Hin- und Herschalten zwischen den beiden Zuständen wird als *gating* bezeichnet. Das *Gating* von Ionenkanälen wird vor allem durch zwei Mechanismen kontrolliert: durch die Bindung spezifischer Substanzen an den Kanal (*ligandengesteuerter Kanal*) oder durch die Membranspannung (*spannungsgesteuerter Kanal*) [FRI98].

Ligandengesteuerte Kanäle

Die Durchlässigkeit der Kanäle wird durch die Anwesenheit bestimmter *Nukleotide* gesteuert. Der Aufbau (*Metabolismus*) und Abbau (*Katabolismus*) dieser Stoffe spielt eine wichtige Rolle bei der Signalübertragung in der *Second Messenger Kaskade*, bei der die kinetischen Eigenschaften der Kanalproteine durch die Nukleotide beeinflusst werden. Ein wichtiger Vertreter dieser zellulären Botenstoffe ist *cGMP* (*zyklisches Guanosin Monophosphat*), das u.a. beim Sehen, bei der Relaxation der glatten Muskulatur und bei der Regulation des Blutzuckersiegels beteiligt ist. Ein cGMP-gesteuerter Kanal ist ohne Anwesenheit von cGMP geschlossen, bei Erhöhung der cGMP-Konzentration wird der geöffnete Zustand immer häufiger, bei sehr hoher cGMP-Konzentration ist der Kanal fast immer offen [FRI98].

Spannungsgesteuerte Kanäle

Um eine Kontrolle der Porenöffnung durch die Membranspannung zu ermöglichen, verfügt ein spannungsgesteuerter Ionenkanal über einen beweglichen Spannungssensor, der Spannungsänderungen in eine Kraft umsetzt. Diese Kraft bewirkt dann beim Protein eine Konformationsänderung, welche die Pore entweder öffnet oder schließt. Der Sensor ist bei vielen Kanälen ein Membransegment, das über positiv geladene Aminosäurereste verfügt. So ist z.B. beim K^+ -Kanal bei Ruhespannung von etwa -70 mV dieses Segment zur Innenseite verschoben und die Pore ist geschlossen. Bei Depolarisation verschiebt sich das Segment zur Membranaußenseite, die Pore wird geöffnet (*Aktivierung*). Nach einer kurzen Verzögerung wird die Pore wieder verschlossen (*Inaktivierung*) [FRI98].

1.2.7 Weiterleitung bioelektrischer Signale: Das Aktionspotenzial

Wird ein Neuron an einer bestimmten Stelle durch einen ausreichend großen positiven Spannungsimpuls erregt (*depolarisiert*), so führt dies durch die Aktivierung spannungsgesteuerter Na-Kanäle zu einer kurzzeitigen reversiblen Änderung des Membranpotenzials, die sich entlang der Zellmembran ausbreitet (siehe Abb. 1.11). Diese Fortleitung des Impulses wird als *Aktionspotenzial* bezeichnet, und ist durch eine verlustfreie Übertragung des Impulses entlang des Axons bis zur den synaptischen Endknoten charakterisiert. Das Aktionspotenzial kann in drei Phasen gegliedert werden:

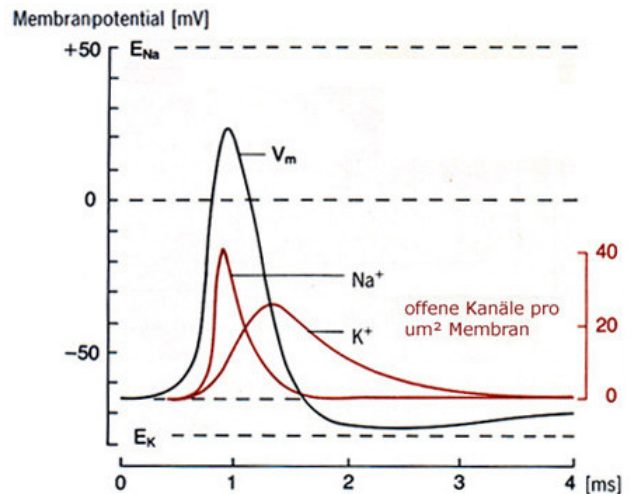


Abb. 1.11: Verlauf eines Aktionspotenzials

Depolarisationsphase

Kennzeichnend für diese Phase ist ein rascher Potenzialanstieg an der Innenseite der Zellmembran, verursacht durch die plötzliche selektive Permeabilitätssteigerung der Membran für Na-Ionen. Na-Ionen strömen vom extrazellulären Milieu ins Zellinnere und führen so zu einer Ladungsumkehr an der Membran. Das Aktionspotenzial ist somit ein Na-Diffusionspotenzial. Das vorher mit ca. -80 mV negative Faserinnere wird ca. 40 mV positiv gegenüber der Außenseite aufgeladen. Dieser Effekt ist selbst-verstärkend, da der Abbau der negativen Aufladung auf der Membraninnenseite die Na-Permeabilität in der unmittelbaren Umgebung weiter erhöht.

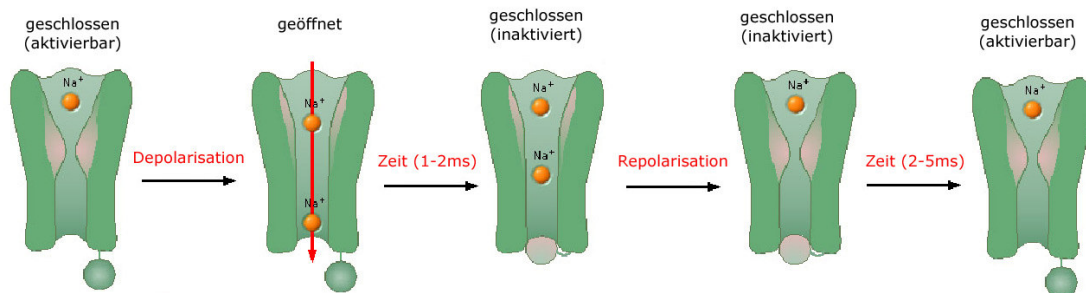


Abb. 1.12: Aktivierung und Inaktivierung eines Natriumkanals

Repolarisationsphase

Die Öffnung der Na-Kanäle ist zeitlich beschränkt: Nach ein bis zwei Millisekunden schließen die Kanäle und bleiben bis zur Repolarisation des Zellinneren inaktiviert. Man beobachtet man einen raschen Rückgang der Potenzialänderung. Die Kalium-Kanäle wurden gleichzeitig mit den Na-Kanälen durch den überschwelligen Reiz aktiviert, öffnen aber langsamer und führen nun zu einem Kalium - Ausstrom und somit zu einem langsamen Rückgang des Potenzials auf den Ruhewert des Membranpotenzials. In dieser Phase kann kein neues Aktionspotenzial ausgelöst werden (Refraktärzeit, siehe Abb. 1.12).

Hyperpolarisation

Durch einen übermäßigen Ausstrom von Kalium-Ionen kann es zu einer deutlichen negativen Aufladung der Membranınnenseite kommen, wodurch der Wert des Ruhepotenzials kurzfristig unterschritten wird. Diese verstärkte Polarisierung in der Refraktärzeit wird *Hyperpolarisation* genannt.

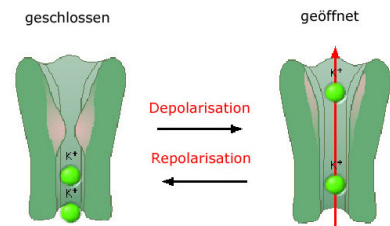


Abb. 1.13: Aktivierung und Inaktivierung eines Kaliumkanals

Das Aktionspotenzial funktioniert nach dem "Alles-oder-Nichts" - Prinzip: Ist der depolarisierende Reiz zu klein, wird kein Aktionspotenzial ausgelöst, und der Reiz wird in seiner unmittelbaren Umgebung abgeschwächt (tonische Leitung). Übersteigt die Depolarisation einen Wert von ca. 40mV, so wird ein Aktionspotenzial ausgelöst und breitet sich nach dem beschriebenen Prinzip aus. Die Nervenleitgeschwindigkeit ist von verschiedenen Faktoren wie dem Fasertyp, -durchmesser und der *Myelinisierung* (siehe Abb. 1.6) abhängig. Sie wird durch die Neurographie ermittelt und liegt im Bereich von 0,5 m/s (Fasertyp C, Schmerzleitung, *viszerale Motoneurone*) bis 120 m/s (Fasertyp A, *Muskelspindel, Sehnenorgan*) [ERL37].

1.2.8 Übertragung von Information auf andere Zellen: Die Synapse

Die Übertragung der elektrischen Signale auf andere Zellen erfolgt durch Synapsen. Es gibt elektrische und chemische Synapsentypen. Bei der elektrischen Synapse fließt der Strom von einer Zelle in die andere direkt über Verbindungsstellen (*Gap junctions*). Bei der chemischen Erregungsübertragung werden durch Depolarisation spannungsgesteuerte Ca^{2+} -Kanäle aktiviert und leiten einen Ca^{2+} -Strom in die Nervenendung. Dort steigt die Ca^{2+} -Konzentration an und führt zur Fusion von transmitterhaltigen *Vesikeln* mit der präsynaptischen Membran. Die Vesikel schütten ihren Inhalt in den synaptischen Spalt aus. Dort interagieren die Transmitterstoffe mit den Rezeptormolekülen der postsynaptischen Membran, aktivieren Kanalproteine und führen zu wiederum zu einer elektrischen Potenzialänderung. Die postsynaptische Potenzialänderung kann erregende Wirkung (*exzitatorisches postsynaptisches Potenzial, EPSP*) oder hemmende (*inhibitorisches postsynaptisches Potenzial, IPSP*) haben. An der postsynaptischen Zelle können zeitliche und räumliche Summationen der eintreffenden synaptischen Potenziale auftreten. [FER06]

Die Transmitterstoffe werden ständig in den Endknötchen des Axons gebildet und in synaptischen Bläschen (*Vesikeln*) gespeichert. Es sind derzeit ca. 100 verschiedene Transmitterstoffe bekannt. Ein Beispiel ist *Acetylcholin*, der Botenstoff für die Aktivierung von Muskelzellen an der neuromotorischen Endplatte.

1.2.9 Vom Nerv zum Muskel: Die neuromotorische Endplatte

Die elektrische Erregung der Nervenfasern wird an der Kontaktstelle von Motoneuron und Muskelfaser auf die Muskelmembran (*Sarkolemm*) übertragen (siehe Abb. 1.14). Dieser Übergang wird neuromotorische Synapse oder *Endplatte* genannt.

In der postsynaptischen Membran befinden sich acetylcholingesteuerte Kanäle (*ACh-Rezeptoren*). Diese öffnen sich, sobald die ACh-Konzentration steigt, und lassen einen depolarisierenden Kationenstrom in die Muskelfaser fließen. Über die Membran der Muskelfaser wird die elektrische Erregung ausgebreitet und gelangt durch Einstülpungen in der Membran (sogenannte *T-Tubuli*) in die Tiefe der Muskelfaser. [FRI98a]

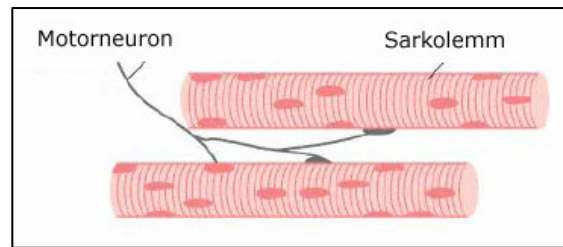


Abb. 1.14: Elemente der neuromuskulären Endplatte

Zeigt das muskelversorgende Motorneuron in der Ruhephase keine oder nur kaum Aktivität, so ist das *Sarkomer*, das die grundlegende funktionelle Einheit des Muskels darstellt, entspannt. Bei Eintreffen von Impulsen in der neuromotorischen Synapse wird Ca^{2+} aus intrazellulären Speichern freigesetzt: Die steigende Calcium-Konzentration bewirkt eine Bindung des aktiven Motorproteins *Myosin* an die *Actin*-Filamente und anschließend eine Bewegung der *Myosin*-Filamente. Dadurch verkürzt sich das Sarkomer, der Muskel kontrahiert (siehe Abb. 1.15). Bei diesem Prozess wird Energie in Form von *ATP* verbraucht.

Sinkt nach Beendigung der elektrischen Stimulation die Calcium-Konzentration wieder, entspannt sich das Sarkomer und der Muskel erschlafft [FRI98a].

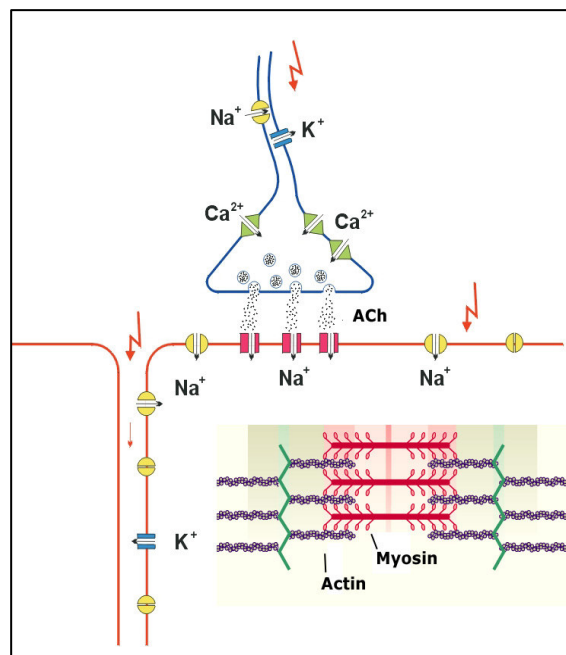


Abb. 1.15: Transmitterfreisetzung an der neuromuskulären Endplatte

Eine erneute Erregungsübertragung ist erst wieder möglich, nachdem transmitterspaltende Enzyme (*Acetylcholinesterase*) den Transmitterstoff deaktivieren, und die Spaltprodukte des Transmitters (*Cholin* und *Essigsäure*) in das Endknöpfchen des Axons zurückdiffundieren, wo sie unter Energieverbrauch zu Acetylcholin resynthetisiert und wieder in den synaptischen Bläschen gespeichert werden [FRI98a].

1.2.10 Das Nervensystem

Die Gesamtheit der Nervenzellen des Organismus bildet das Nervensystem. Das Nervensystem der Wirbeltiere ist in das *somatische* (*Umwelt*-) und in das *vegetative* (*autonome* oder *Innenwelt*-) Nervensystem organisiert.

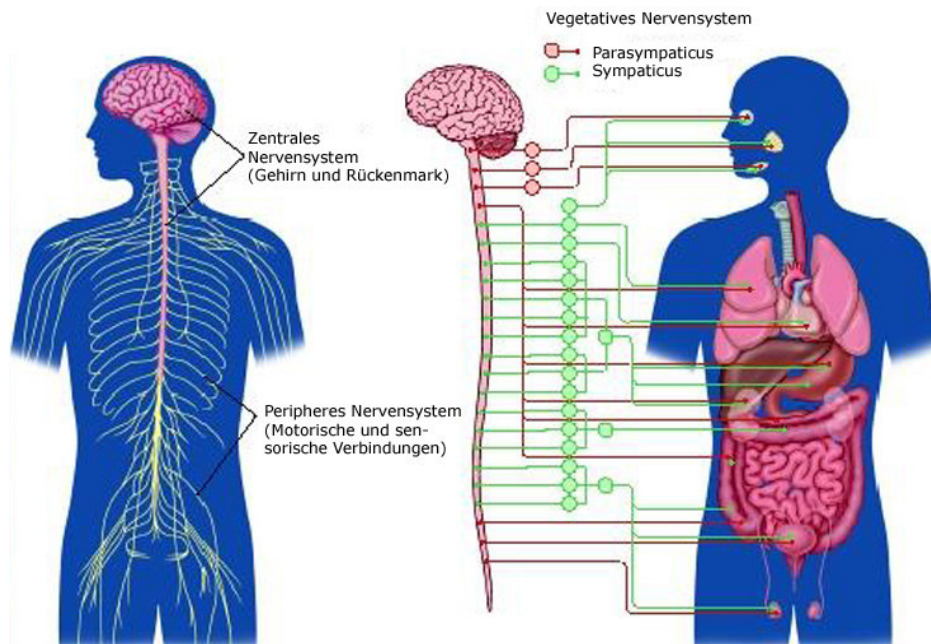


Abb. 1.16: Zentrales- und peripheres Nervensystem, Sympaticus und Parasympaticus

Das vegetative System besteht aus dem *sympathischen* und dem *parasympathischen* Nervensystem (siehe Abb. 1.16). Fast alle Organe werden von beiden Systemen innerviert. *Sympathicus* und *Parasympathicus* wirken antagonistisch, dieses Prinzip ermöglicht eine feine Steuerung der Organe, etwa der Hormondrüsen [FER06].

Der Parasympathicus dämpft den Herzschlag und fördert die Verdauung, er sorgt für Ruhe und Entspannung und Schonung des Organismus. Im Gegensatz dazu bewirkt der Sympathicus eine Leistungssteigerung, etwa werden die Herztätigkeit und der Blutkreislauf angeregt, während die Darmtätigkeit gehemmt wird. Der Sympathicus versetzt den Körper in den Zustand hoher Leistungsbereitschaft, dies bereitet auf Angriff, Flucht, Verteidigung oder außergewöhnliche Anstrengungen vor. Der Transmittersubstanzen des Parasympathicus zur Übertragung an den Synapsen ist Acetylcholin, der Sympathicus verwendet vor allem Adrenalin und Noradrenalin. [VOB06].

Das Leitungssystem von Sympaticus und Parasympaticus liegt zu großen Teilen in den Ganglien außerhalb des Wirbelkanals. Das Sonnengeflecht, eine nur durch die Bauchmuskulatur geschützte Ansammlung miteinander verbundener Ganglien, ist ein wesentlicher Teil des Sympaticus. Der Parasympathicus hat Schwerpunkte im Nachhirn und im Rückenmark der Kreuzregion, und im *Nervus Vagus* (dem 10. Gehirnnerv) [VOB06].

Der *Hypothalamus* im Zwischenhirn ist ein zentrales Steuerungsorgan des vegetativen Nervensystems. Zu den Aufgaben gehören die Regulation der Körpertemperatur, des Kohlenhydratstoffwechsels und des Mineralstoff- und Wasserhaushalts, die Steuerung der Abgabe von Harn und Schweiß und die Steuerung der Sexualität. Die Weiterleitung von Information erfolgt sowohl nervös als auch über Hormone. [VOB06]

2. Eigenschaften und Anwendung von Biosignalen

Mit dem Begriff "*Biosigna*l" können im weitesten Sinne alle messbaren Vitalparameter von Organismen bezeichnet werden. Zu den menschlichen Biosignalen zählen sowohl Stimme als auch Mimik, Gestik, Körperbewegungen oder elektrisch messbare Größen wie Nerven- und Muskelaktivität (*bioelektrische* Signale). Die Art der Messung bestimmt die Modalität der gemessenen Information: Körperbewegungen könnten etwa visuell mit einer Kamera, akustisch über Kontaktmikrofone oder elektrisch durch Muskelelektroden erfasst werden. Welche Form der Messung am besten geeignet ist, entscheidet sich durch die Charakteristik der relevanten Information sowie durch die individuellen Möglichkeiten und Bedürfnisse der beteiligten Personen. Die Wahl des richtigen Instrumentariums bedingt ein Wissen über die quantitativen und qualitativen Signaleigenschaften wie Signalpegel, Bandbreite und Sensorik für die Abnahme. Dadurch können die nötigen Filtereinstellungen, Verstärkungsfaktoren und Abstraten für eine fehlerfreie Digitalisierung der Signale ermittelt werden. Liegen die Messwerte in Form digitaler Zeitreihen vor, eröffnet dies viele Möglichkeiten für eine flexible Anwendung, hier einige Beispiele:

- Archivierung in Datenbanken, Auswertung von Kennzahlen und Statistiken
- Vergleich physiologischer und pathologischer Signalformen, Klassifikation
- Visualisierung in funktionalen Modellen, mehrdimensionale Darstellung
- Überwachung in der Intensivmedizin und durch Telemetrie
- Therapieunterstützende Maßnahmen und Therapieerfolgskontrolle
- Aktive Prothetik und Steuerung von Orthesen
- Schrittmachertechnik, funktionelle Elektrostimulation (FES)
- Biofeedback und Neurofeedback
- Alternative Benutzerschnittstellen (HCI)

Die Auswertung von Biosignalen ist heute ein unerlässliches Hilfsmittel in der medizinischen Forschung und im klinischen Alltag. Die Echtzeitverarbeitung der Signale wird zunehmend in lebenserhaltenden und -unterstützenden Maßnahmen eingesetzt. Im Folgenden werden die Charakteristika einiger Biosignale und die Anwendungsgebiete *Biofeedback* und *Alternative Benutzerschnittstellen* genauer vorgestellt.

2.1 Elektrische Biosignale

Bioelektrische Signale entstehen durch die Depolarisation von Nerven- oder Muskelzellen im vitalen Gewebe (siehe Kapitel 1.2). Sie können an der Körperoberfläche durch Oberflächenelektroden (*nichtinvasiv*) oder im Körperinneren (*invasiv*) durch *subdurale* Elektroden (knapp unter der Haut) bzw. durch implantierbare Elektroden gemessen werden. An der Körperoberfläche lässt sich nur die Summe der Aktivität vieler Entladungen der tieferen Schichten messen, eine genaue Lokalisation der Quellen ist schwierig und nur durch viele Messpunkte annähernd möglich. Implantierbare Elektroden erlauben eine höhere Auflösung und ein besseres *Signal/Rausch-Verhältnis* (*Signal/Noise-Ratio*, *SNR*). Der medizinische Aufwand und die möglichen Komplikationen sind jedoch viel größer. Ein großer Teil der bioelektrischen Signale lässt sich in ausreichender Qualität von der Körperoberfläche ableiten. Durch geeignete Elektroden und ein elektrolytisches Gel wird der Kontakt zur Haut hergestellt. Die elektrische Leitung im Körper basiert auf einer Wanderung von Ionen,

die Leitung in der Elektrode bzw. in den Kabeln basiert auf der Bewegung von Elektronen. An der Kontaktstelle von Haut und Elektrode erfolgt die Umwandlung einer elektrolytischen Stromleitung durch Ionenwanderung in eine Stromleitung in Festkörpern durch reine Elektronenwanderung. Die folgenden bioelektrischen Signale können durch Elektroden abgenommen werden, sie zeigen je nach Signalart und Ableitungspunkt unterschiedliche Ausprägungen in Pegelstärke und Frequenzband:

2.1.1 Elektromyogramm (EMG)

Durch das Elektromyogramm werden die Depolarisationen von Muskelzellen gemessen. Das EMG wird meist durch Oberflächenelektroden abgeleitet (Abb. 2.1a). In seltenen Fällen werden Nadelelektroden zur subkutanen Einzelzelleitung (*single motor unit, SMU*) oder implantierbare Elektroden verwendet. Eine Abnahme an der Körperoberfläche wird meist durch Klebe- oder Napfelektroden durchgeführt und ist verglichen mit den invasiven Techniken unproblematisch. Bei der (bipolaren) Ableitung ist darauf zu achten, dass sich die beiden Elektroden in Faserrichtung des Muskels befinden. Der Abstand sollte etwa 5 cm betragen, also nicht zu groß sein [SEE06].

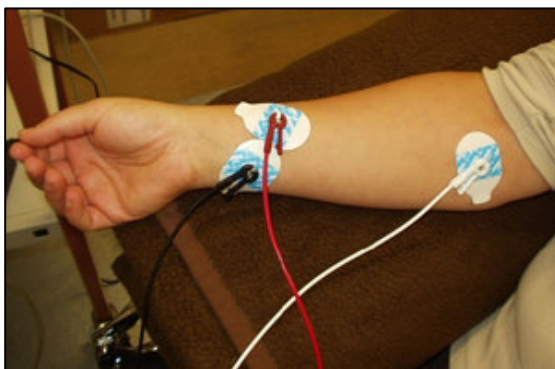
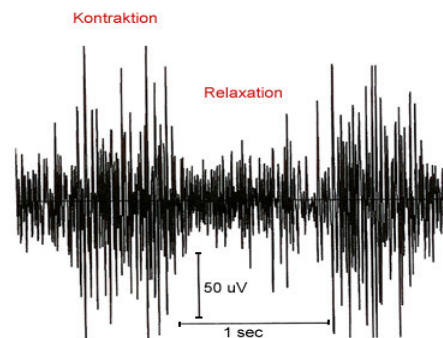


Abb. 2.1: (a) Ableitung des EMG vom rechten Unterarm



(b): Messung einer Muskelkontraktion

Die messbare Signalamplitude ist von der Größe der Elektrode, der Größe des Muskels, der Qualität der Elektrodenverbindung und der Muskelaktivität abhängig. Sie kann beim Oberflächen-EMG bis zu 1-2 mV betragen, ist aber im Normalfall in der Größenordnung von 400 μV (siehe Abb. 2.1b). Das aktive Frequenzband ist beim EMG durch die gleichzeitige, zufällige Aktivität vieler Zellen relativ hoch. Die schnellsten Frequenzanteile liegen im Bereich einiger kHz, wobei ein Großteil der Aktivität unter 500Hz stattfindet [DAH05].

Häufige Ableitungspunkte für das EMG sind der Nacken- und Rückenbereich zur Behandlung von Verspannungen, die Gesichts- und Stirnmuskulatur (*faziales EMG*, Abb. 2.2) zur Therapie chronischer Kopfschmerzen, die Muskeln des Unterarmes zur Verwendung als Kontroll- und Steuermedium, sowie Bizeps und Beinmuskulatur in der Sportphysiologie und Biomechanik des Bewegungsapparates [DAH05].

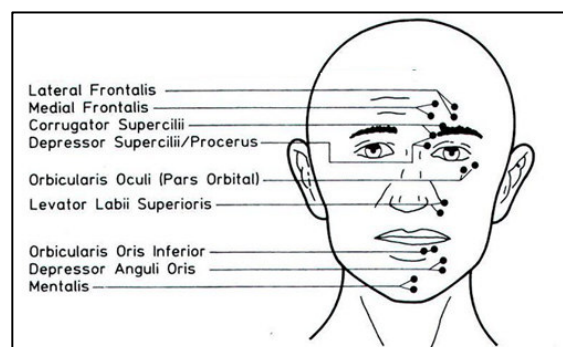
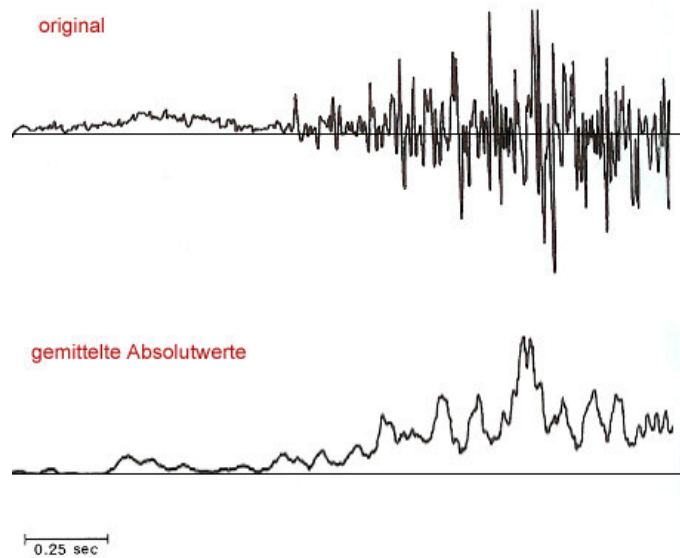


Abb. 2.2: Ableitungspunkte des fazialen EMG



Eine nützliche Technik zur Signalaufbereitung des EMG ist die Mittelung der absoluten oder quadrierten Signalwerte, da dadurch ein direkter Bezug zur Muskelanspannung hergestellt werden kann. Abb. 2.3 (oben) zeigt das originale EMG-Signal, das aus schnellen Schwankungen von positiven und negativen Werten besteht (Hauptfrequenzbereich 50-200 Hz, Amplitude ca. 400 μ V). Abb.2.3 (unten) zeigt das aufbereitete Signal, das als ein Maß für die makroskopische Anspannung des Muskels verwendet werden kann.

Abb. 2.3: EMG-Signalaufbereitung durch Mittelung von Absolutwerten

Weiters eignet sich die *Fourier-Transformation* zur Analyse des Frequenzgehaltes. (Weitere Informationen zur Signalaufbereitung siehe 4. Softwaremethoden.)

Das EMG ist eine wichtige diagnostische Möglichkeit zur Erkennung von Muskelsuffizienz und ein Kontrollinstrument für Rehabilitation und Muskelaufbau. In der *Neurographie* wird die Reizantwort des Muskels auf eine elektrische Stimulation des Nerven zur Feststellung der Nervenleitgeschwindigkeit gemessen.

2.1.2 Elektrokardiogramm (EKG)

Das EKG misst die Depolarisation des Herzmuskels und hat große Bedeutung in der Diagnostik von Herzkrankheiten und in der Vorbeugung von Herz-Kreislauferkrankungen. Das EKG konnte aufgrund seiner starken Amplitude bereits 1887 erstmals gemessen werden. Seither etablierten sich mehrere Varianten der Ableitung, von denen heute vor allem die Ableitungen nach Wilson, Einthoven und Goldberger von Bedeutung sind. Ein vollständiges Standard-EKG wird an 12 Stellen abgeleitet [DAV]:

3 bipolare Ableitungen von den Extremitäten (nach Einthoven, Abb 2.4):

- I: rechter Arm (-) zum linker Arm (+), lateral
- II: rechter Arm (-) zum linken Bein (+), superior-inferior
- III: linker Arm (-) zum linken Bein (+), superior-inferior

Die Elektrode am rechten Arm ist meist rot, die am linken Arm gelb und die am linken Bein grün markiert, man spricht von einer *Ampelableitung*.

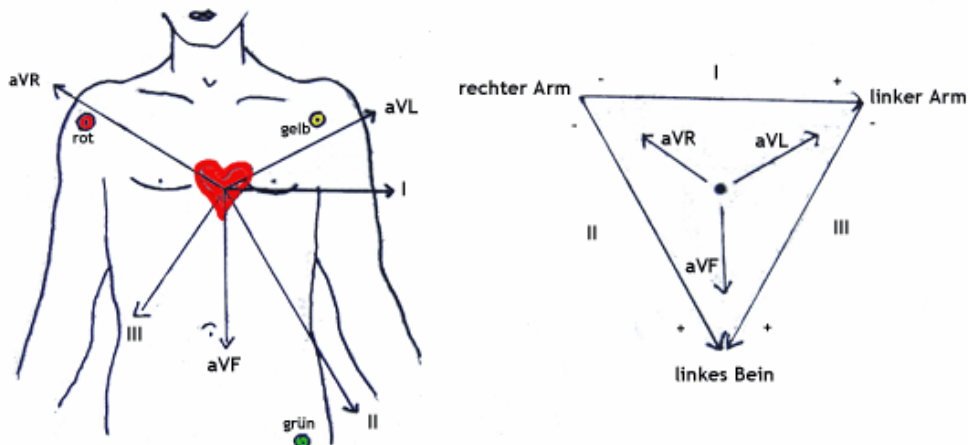


Abb. 2.4: Die drei Hauptachsen bei der EKG-Ableitung nach Goldberger (l.) bzw nach Einthoven (r.)

3 unipolare Ableitungen von den Extremitäten (nach *Goldberger*, Abb. 2.4): Eine Extremität (differente Elektrode) wird gegenüber der gemeinsamen Referenz der beiden anderen Extremitäten (indifferente Elektrode) abgeleitet:

- aVR: rechter Arm (+) zum Linken Arm+ linkem Fuß (-)
- aVL: linker Arm (+) zum rechten Arm + linkem Fuß (-)
- aVF: linker Fuß (+) zum rechten + linken Arm

6 unipolare Brustwandableitungen (nach *Wilson*, Abb. 2.5):

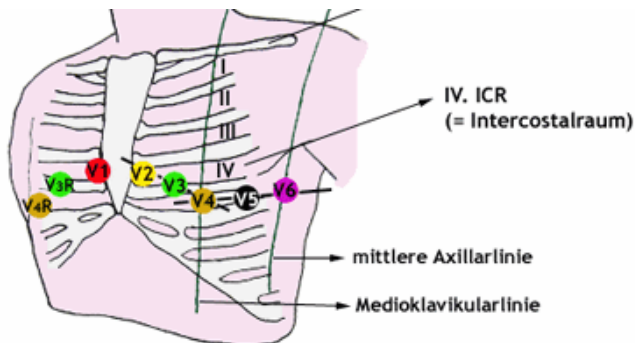


Abb. 2.5: Die sechs Ableitpunkte an der Brustwand nach Wilson

V1, V2, V3, V4, V5, V6 werden entlang des Zwischenraumes der vierten und fünften Rippe (*Interkostalraum*), beginnend rechts des Brustbeines jeweils gegen die indifferente Elektrode (Zusammenschluss der drei Extremitätenableitungen) abgeleitet.

Das EKG wird als Summenvektor der einzelnen Ableitungspunkte berechnet. Dies ist eine starke Vereinfachung der komplexen 3-dimensionalen Erregungsausbreitung im Herzen. Dennoch kann ein/e geübte/r Kardiologe/in aus der EKG-Ableitung Indikationen für eine Reihe von Herzbeschwerden wie *Bradykardie* (zu langsamer Herzrhythmus), *Tachykardie* (zu schneller Herzrhythmus), Erregungsleitungsstörung (*Schenkelblock*, *AV-Block*), Herzwandverdickungen (*Hypertrophien*), Rhythmusstörungen (*Vorhof-* oder *Kammerflimmern*) oder einen *Myokardinfarkt* erkennen. Eine alternative Darstellungsform ist das *Vektor-EKG*, bei dem die Erregungsfortleitung als Vektorschleife räumlich dargestellt wird [CAR06].

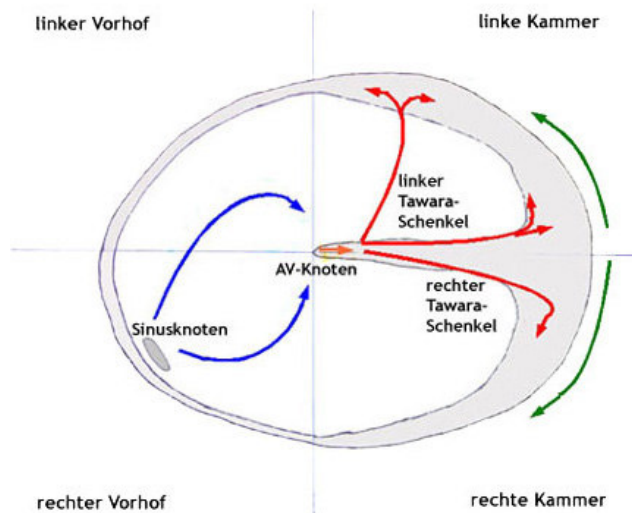


Abb. 2.6: Der Verlauf der Erregungsausbreitung im Herzen

Die Pumpfunktion des Herzens geht im Normalfall vom Sinusknoten aus (siehe Abb. 2.6). Der Sinusknoten ist ein autonomer Impulsgenerator für die Erregung des Herzmuskels. Die Impulsgeschwindigkeit wird durch den Sympaticus oder Parasympaticus (*Nervus Vagus*) gesteigert bzw. verlangsamt. Vom Sinusknoten breitet sich die Erregung über die Vorhöfe (*Atrium*) zum *Atrio-Ventrikular-Knoten (AV-Knoten)* und von dort über die *Tawara-Schenkel* und das *His-Bündel* in die Herzkammern (*Ventrikel*) aus [CAR06].

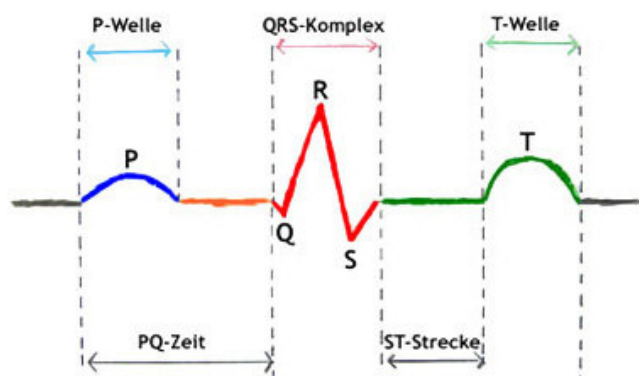


Abb. 2.7: Die charakteristische Form des EKG

Die charakteristische Form des EKG lässt sich in *P-Welle*, *QRS-Komplex* und *T-Welle* gliedern (Abb. 2.7). Die P-Welle entsteht durch die sequentielle Aktivierung (Depolarisation) des rechten und linken Vorhofes. Der QRS-Komplex entsteht durch die Depolarisierung der rechten und linken Ventrikel, und die T-Welle durch die Repolarisierung der Ventrikel. [CAR06]

Die Höhe und Ausprägung der Wellen sowie ihre Intervalle geben Aufschluss über ein physiologisches oder pathologisches Geschehen. Die höchste Amplitude im EKG ist normalerweise die R-Zacke, sie erreicht bis zu 1,5 mV in der Oberflächenmessung. Die verwendeten Elektroden für die Extremitätenableitungen sind meist Elektrodenklammern, für die Brustwandableitungen werden Klebe- oder Saugnapfelektroden verwendet.

Herzratenvariabilität (HRV)

Die Herzratenvariabilität ist ein Parameter, der die kurzfristige Änderung der Herzfrequenz durch Belastung oder auch durch die Atmung angibt. Die Fluktuation in der Frequenz ergibt sich durch Wechselwirkungen der Schrittmacherzellen des Herzens mit dem sympathischen und parasympathischen Nervensystem. Der Abstand zweier aufeinander folgender Herzschläge wird gemessen (*NN-Intervall, normal-to-normal*) und die Standardabweichung der Werte (*SDNN*) sowie weitere Parameter werden berechnet. Auswertung bzw. Feedback der HRV dient zur Risikoanalyse für Herzerkrankungen und zum Training der Adaptionfähigkeit des Herzens [TE100].

2.1.3 Elektrokulogramm (EOG)

Das Elektrokulogramm ist eine Messung Augenbewegungen mittels Elektroden. Die Muskulatur des Auges erlaubt prinzipiell eine Bewegung in drei Freiheitsgraden. Durch das EOG können vertikale und horizontale Augenbewegungen erfasst werden (nicht die Rotationsbewegung des Auges um seine Längsachse).

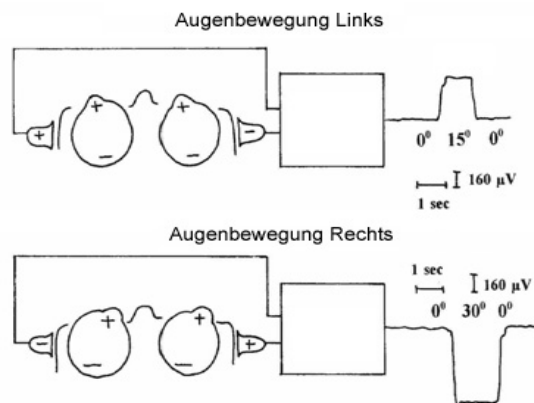
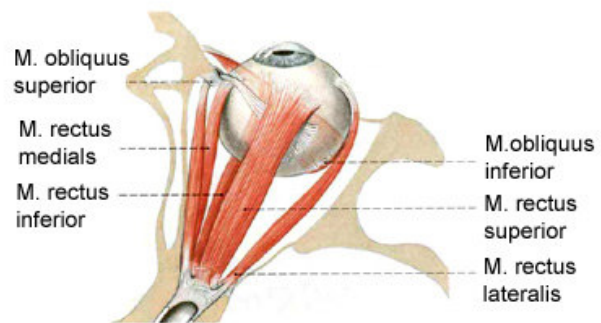


Abb. 2.8 (a): Die Entstehung des EOG durch den Dipolcharakter des Auges



(b): Die Muskulatur für die Bewegung des Auges

Die physiologische Grundlage des EOG ist eine stetige Potenzialdifferenz zwischen der Netzhaut (*Retina*) und der Hornhaut (*Cornea*), wodurch das Auge die Eigenschaften eines Dipols aufweist (*Cornea-Retinales Potenzial*, siehe Abb. 2.8a). Die Montage der Elektroden für die horizontale Messung erfolgt links und rechts neben den Augen, zur vertikalen Messung werden Elektroden knapp oberhalb und unterhalb des Auges befestigt. Durch das EOG können verschiedene Arten von Augenbewegungen wie *Sakkaden*, glatte Fortbewegungen, *Nystagmus* oder Lidschlag erkannt werden. Das EOG findet Anwendung in psychophysiologischen Untersuchungen (Änderung der Augenbewegungen bei kognitiven Aufgaben oder unter Stressbedingungen), in der Diagnostik (Pathologie der Augenmuskulatur, Indikation neurologischer Erkrankungen, Untersuchungen des Gleichgewichtsorganes), in der Schlaforschung und im Bereich alternativer Computerschnittstellen. Der Messbereich für die Blickrichtung ist verglichen mit anderen Verfahren sehr hoch und beträgt etwa ± 70 Grad, bei einer Genauigkeit von 1,5 - 2 Grad. Nachteile sind die Beeinflussung durch Fehlerquellen wie Muskelpotenziale oder unerwünschten Lidschlag.

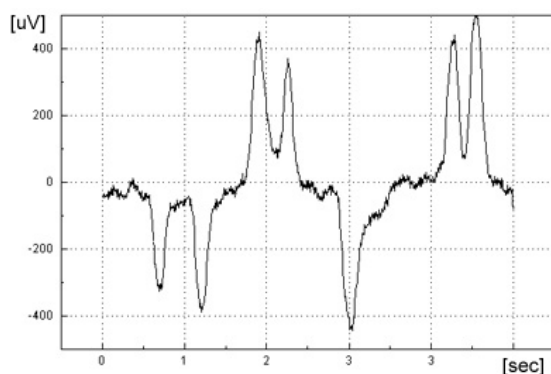


Abb. 2.9: Messung des EOG, horizontale Blicke

Die Amplitude des EOG beträgt bis zu 800 μV , der relevante Frequenzbereich liegt zwischen 0 Hz (Gleichspannung bei Fixierung eines Punktes) bis zu etwa 20 Hz (Sakkaden, Nystagmus).

Abb. 2.9 zeigt eine EOG-Ableitung der Horizontalbewegungen, es wurden mehrfach Blicke nach links und rechts ausgeführt.

2.1.4 Elektroenzephalogramm (EEG) und Elektrokortikogramm (ECoG)

Beim Elektroenzephalogramm wird die Depolarisation von Nervenzellen des Gehirns gemessen. An der Kopfoberfläche ist nur die Summe von Aktionspotenzialen größerer Zellverbände messbar, das Signal wird durch den Schädelknochen und die Kopfhaut weiters gestreut. Es können durch das Oberflächen-EEG vor allem die Aktivitäten der Großhirnrinde (*Kortex*) erfasst werden. Ein Großteil der Signale wird durch die postsynaptischen Potenziale der Pyramidenzellen produziert. Ein synchronisiertes Depolarisieren von 10000 Zellen (ca. 1mm³ Volumen der Hirnrinde) erzeugt an der Kopfhaut Potenzialänderungen von etwa 1 μ V.

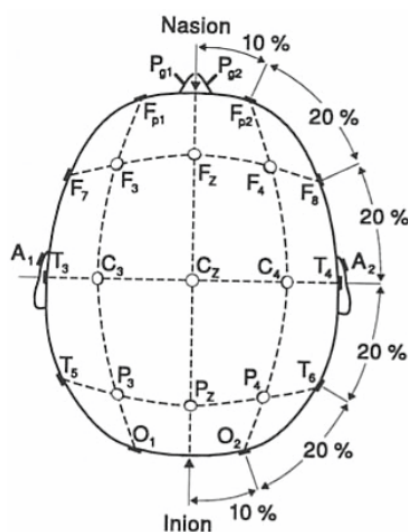


Abb. 2.10: Elektrodenpositionen nach dem 10/20-System

Bei der Ableitung unterscheidet man unipolare und bipolare Konfigurationen. Bei der unipolaren Messung werden die Elektroden über den Messpunkten gegen eine indifferente Elektrode an einem elektrisch wenig aktiven Bereich (z.B. am Ohr läppchen) abgeleitet. Die bipolare Ableitung misst die Differenzspannung zweier aktivierter Bereiche. Die Ableitung des Oberflächen-EEG erfolgt meist mit chlorierten Silberelektroden (*Ag/AgCl*) und einem elektrolytischen Gel (Kontaktgel). Aktive Elektroden mit Vorverstärker bieten besseren Schutz gegenüber Störsignalen und Einstrahlungen. Meist werden zur Platzierung und Fixierung Elektrodenhauben (*Caps*) verwendet, die Befestigungsmöglichkeiten an den international genormten Positionen des 10/20-Systems bieten. (siehe Abb. 2.10).

Der relevante Frequenzbereich des EEG liegt im Bereich von 0Hz (Gleichspannungskomponente, negative Aktivierung) bis etwa 100Hz, wobei sich ein Großteil des Frequenzgehaltes unter 40Hz befindet. Zur Klassifikation des EEG hat sich die Benennung folgender Frequenzbereiche etabliert:

- Delta* (1 - 4 Hz) : tritt überwiegend im Schlaf auf, stark im Tiefschlaf
- Theta* (4 - 8 Hz) : vermehrt bei Träumen und Meditation, bei Kleinkindern
- Alpha* (8 - 13 Hz) : im entspannten Wachzustand, bei geschlossenen Augen
- Beta* (13 - 32 Hz) : im aktiven Wachzustand

Die Zusammensetzung des EEG-Signals aus diesen Frequenzkomponenten zeigt sich im freilaufenden EEG (*free running EEG, FREEEG*) ohne die Anwendung mathematischer Verfahren wie z.B. Mittelung über mehrere Versuche. Das physiologische EEG eines Erwachsenen im Wachzustand zeigt wenig Delta- und Theta-Anteile, deutlich ausgeprägte Alpha-Anteile, die beim Schließen der Augen zunehmen und Beta-Anteile, die in den höheren Frequenzbereichen abnehmen (siehe Abb. 2.11).

Im Schlaf sind einzelne Schlaf-Stadien (Einschlafphase, leichter, mitteltiefer und Tiefschlaf, Traumphase / REM-Schlaf) und spezielle Wellen (sog. Schlaf-Spindeln) erkennbar. Die REM-Phase nimmt ca. 20 Prozent der Schlafes ein, der Tiefschlaf etwa 30 Prozent.

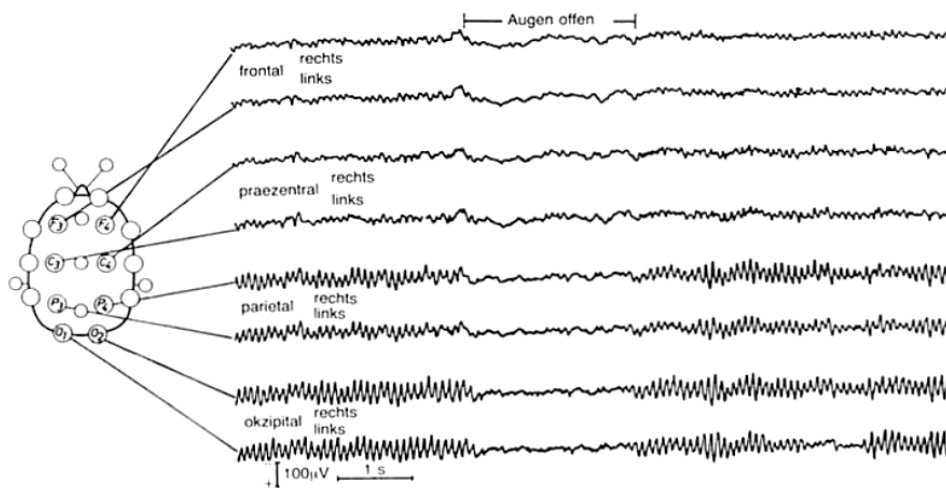


Abb. 2.11: Ableitung des EEG, Desynchronisation des Alpha-Rhythmus durch das Öffnen der Augen

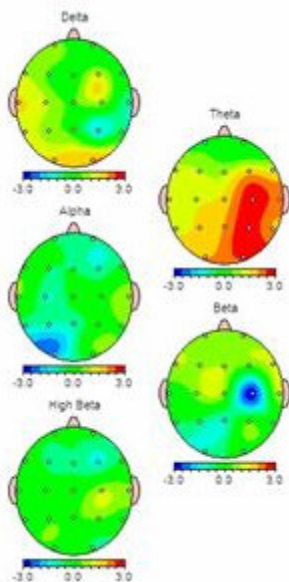


Abb. 2.12: Darstellung der verteilten Aktivität beim QEEG

Ein pathologisches EEG äußert sich u.a. durch das Auftreten inadäquater Frequenzen (etwa hohe Delta-Werte im Wachzustand). Eine genaue Bestimmung der Quellen einer Aktivität ist durch das EEG nur durch die gleichzeitige Ableitung vieler Kanäle und die Anwendung mathematischer Rücktransformationen wie der *Independent Component Analysis (ICA)* möglich. Im quantitativen EEG (QEEG) wird durch eine hohe Anzahl von Kanälen (128 oder mehr) eine 2-dimensionale oder räumliche Darstellung der Hirnaktivität ermöglicht (siehe Abb. 2.12).

Die Amplitude des EEG erreicht normalerweise kaum mehr als 200 μV . Das EEG ist damit ein sehr schwaches bioelektrisches Signal, was die Anforderungen an die Verstärkertechnik wesentlich erhöht. Die EEG-Messung wird mit Differenzverstärkern durchgeführt, wobei meist ein Hochpassfilter zur Unterdrückung des Gleichspannungsanteils verwendet wird, um hohe Signalverschiebungen durch Polarisierung der Elektroden zu vermeiden. Wird auch der Gleichspannungsanteil gemessen, sollte der Verstärker einen sehr hohen Messbereich besitzen.

Das EEG findet im Bereich der Medizin viele Anwendungen: In der Diagnostik können verschiedene Formen der Epilepsie durch die Charakteristik der Anfalls- und Ruhe EEGs unterschieden werden (*ikterale und interikterale Spikes*), Alzheimer, Hirntumore oder neuronale Disfunktionen können oft im EEG erkannt werden. Kognitionswissenschaft und Physiologie verwenden das EEG zur funktionalen Zuordnung mentaler Fähigkeiten und zur Erforschung des Ablaufes neuronaler Aktivierungen. In der Pharmazie dient das EEG zur Verifikation von Wirkstoffen, in der Schlafanalyse zur Schlafstadien-Klassifikation und in der Anästhesie zur Kontrolle des Narkoseverlaufs. In der Rehabilitations- und Kommunikationstechnik findet das EEG in *Brain-Computer Interfaces* Anwendung (BCI, siehe Kapitel 2.5).

Ein entscheidender Vorteil der EEG-Ableitung ist die hohe zeitliche Auflösung: Potenzialänderungen im Bereich von Millisekunden können durch das EEG problemlos gemessen werden. Diese zeitliche Genauigkeit kann (derzeit) von Systemen mit besserer räumlicher Auflösung wie *fMRI* oder *PET* nicht erreicht werden.

Bei *intracranialen* EEG-Ableitungen werden die Potenziale nicht von der Kopfhaut sondern durch einen operativen Eingriff von tieferen Schichten abgenommen. Man unterscheidet hier verschiedene Arten: Bei der **subduralen Ableitung** werden die Elektroden unter der Kopfhaut (aber außerhalb des Schädelknochens) angebracht. Beim **Elektrokortikogramm (ECoG)** werden die Potenziale direkt von der Oberfläche des Kortex (durch Öffnung der Schädeldecke) abgenommen. Die Ableitung erfolgt meist mit Streifen- oder Matrixelektroden (*grid-electrodes*). Bei der **subkortikalen Ableitung** werden Elektroden in tiefere Hirnregionen implantiert. Diese Ableitpositionen ermöglichen weitaus stärkere Signale als das Oberflächen-EEG. Durch subdurale Messungen können etwa epileptische Herde vor einer Operation genau lokalisiert werden, durch kortikale oder Tiefenableitungen ist die Messung der Aktivität einzelner Nervenzellen möglich [SEE06]. Die Implantierung von Elektroden zur künstlichen Wiederherstellung motorischer Funktionen (*Direct Brain Interface, DBI*; siehe 2.5.5) wird seit längerem wissenschaftlich erforscht [LEV00].

2.1.4.1 Evozierte und Ereigniskorrelierte Potenziale (EP, EKP, ERP)

Da im Oberflächen-EEG die kleinen Signalanteile aus tieferen Hirnschichten nicht feststellbar sind, weil sie im Vergleich mit dem Restsignal eine viel zu geringe Amplitude aufweisen (der Signal/Rausch-Abstand ist zu klein) verwendet man die Methode der Mittelung mehrerer Versuche (*trial-averaging*). Hierfür wird zu einem bekannten Zeitpunkt ein visueller, auditiver, somatosensorischer oder kognitiver Reiz präsentiert und die Reaktion auf diesen Reiz im EEG aufgezeichnet. Durch die vielfache Wiederholung des Versuches (zB. 80 oder 150 Einzel-Trials) und die Addition der Signalwerte rund um den Präsentationszeitpunkt (*phase-locking*) verschwinden die Signalanteile, die mit der Reaktion auf den Reiz nichts zu tun haben durch Mittelung der zufälligen Anteile: Der Signal/Rausch-Abstand steigt. Man spricht bei dem ermittelten Signal von einem *evozierten Potenzial (evoked potenzial, EP)*, es stellt die unmittelbare physiologische Reaktion auf den Reiz dar. Je nach Art des präsentierten Stimulus lassen sich *visuell evozierte Potenziale (VEP)*, *auditorisch evozierte Potenziale (AEP)* oder *somatosensorisch evozierte Potenziale (SSEP)* messen, deren Form charakteristisch ist. Abweichungen in den EPs können auf pathologische Veränderungen bzw. auf den Ort der Veränderung hindeuten. Die Amplitude der EPs reicht von Bruchteilen eines μV bis zu einigen μV .

Unter dem *ereigniskorrelierten Potenzial (EKP, event-related potenzial ERP)* versteht man die evozierten Potenziale höherer Hirnschichten. EKPs werden durch kognitive Prozesse wie Erinnerung, Aufmerksamkeit oder Erwartung eines Reizes beeinflusst. EKPs werden ebenfalls durch Mittelung aus dem Testsignal herausgehoben. An der Schwelle zur Bewusstwerdung kann eine charakteristische Signalkomponente im EKP identifiziert werden, die *P300-Komponente (P3, siehe Abb. 2.13)*.

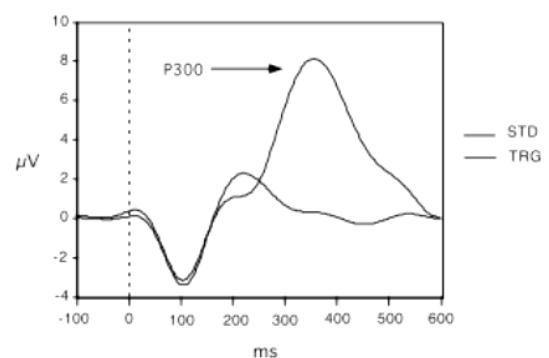


Abb. 2.13: Die P300-Komponente

Sie bezeichnet eine signifikante Positivierung im Signal, ca. 300ms nach einem Reiz, der relevant für eine aktuelle Aufgabe ist. Man findet die P300-Komponente vor allem in parietalen Bereichen des Kortex bei einer bewussten Reaktion auf den Stimulus. Sie eignet sich zur Verwendung in Brain Computer Interfaces, die auf dem so genannten *Oddball-Paradigma* beruhen, wo ein bestimmter Reiz von anderen, nicht relevanten Stimuli unterschieden wird (siehe 2.5 Brain Computer Interfaces).

2.1.4.2 ERD / ERS

Die ereignisbezogene Zunahme bzw. Abnahme von Frequenzaktivität im kontinuierlichen EEG wird als *Event-Related Synchronisation (ERS)* bzw. *Event-Related Desynchronisation (ERD)* bezeichnet. Diese Änderung in der Magnitude bestimmter Frequenzbänder ist auch ohne phasengebundene Mittelung im EEG festzustellen, und hat ihre Ursache in einer großflächigen, periodisch-synchronen Aktivierung der beteiligten Nervenzellen [DUR04]. Eine Synchronisation tritt zumeist dann auf, wenn die betroffenen neuronalen Areale in Ruhe, also nicht unmittelbar an einer Informationsverarbeitung beteiligt sind. Die am besten erforschte Region im Bezug auf das Synchronisierungsverhalten sind die motorischen bzw. supplementär-motorischen Gehirnregionen (*Gyrus präcentralis*). Es konnte festgestellt werden, dass kurz vor und während der Ausführung einer Bewegung Desynchronisationen im Alpha- und Beta-Frequenzband stattfinden (*ERD*, Abb. 2.14). Diese Phänomene treten auch bei einer reinen Vorstellung von Bewegungen auf. Der so genannte *μ -Rhythmus* (bei etwa 10 Hz) tritt also verstärkt auf, wenn die motorischen Areale des Kortex nicht aktiv sind. Da diese Hirnaktivität willkürlich kontrollierbar ist, eignet sie sich für die Verwendung in Brain-Computer Interfaces (*μ -BCI*, siehe Kapitel 2.5.2)

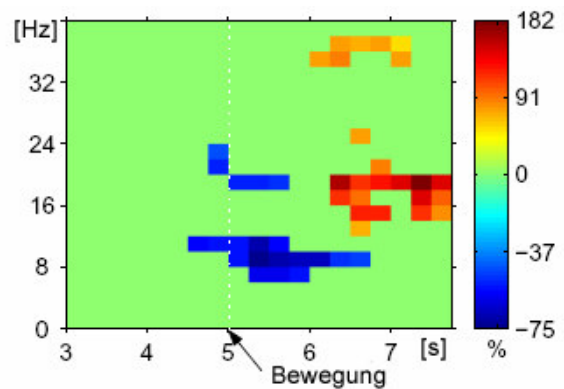


Abb. 2.14: ERD durch Bewegungsvorstellung

2.1.4.3 Langsame Potenziale (SCP), Laterales Bereitschaftspotenzial (LRP)

Langsame Hirnpotenziale (*slow cortical potentials, SCP*) sind ereignisbezogene Veränderungen in den untersten Frequenzkomponenten des EEG bzw. in dessen Gleichspannungsanteil. Sie entstehen durch großflächige Negativierungen oder Positivierungen des Kortex und drücken eine erhöhte oder verminderte Aktivierbarkeit der neuronalen Informationsverarbeitung aus [BIR00]. Das *Laterale Bereitschaftspotenzial (lateral readiness potential, LRP)* ist eine langsame Negativierung im EEG, die zur Vorbereitung einer Aktion dient, wenn diese bereits erwartet werden kann. Die Potenzialverschiebung kann mehrere Sekunden bis zum Eintreten des Ereignisses aufrechterhalten werden. Die Beeinflussung der langsamen Potenziale zur Verwendung in Brain-Computer Interfaces kann erlernt werden, eine Detektion im freilaufenden EEG ist möglich [BIR00]. Abb. 2.15 zeigt ge-

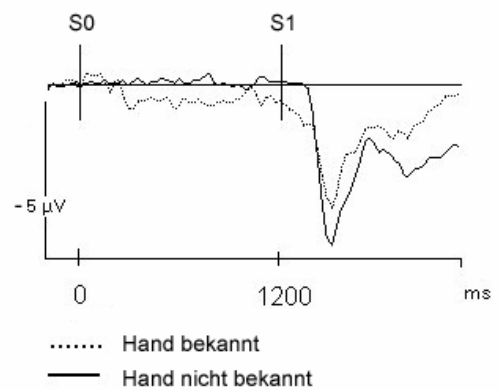


Abb. 2.15: Langsame Potenziale bei der Vorbereitung einer Bewegung

mittelte EEG-Signale bei der Bewegung der linken oder rechten Hand, ausgelöst durch ein akustisches Signal zum Zeitpunkt S1. Wird zuvor beim Zeitpunkt S0 die Seite bekanntgegeben, tritt das LRP auf.

2.1.5 Elektrodermale Aktivität (EDA)

Die Elektrodermale Aktivität bezeichnet Phänomene der Leitfähigkeitsänderung der oberen Hautschicht (*Stratum Corneum* der *Epidermis*, siehe Abb. 2.16b). Man unterscheidet zwischen dem Leitfähigkeitsniveau (*SCL*, *skin conductance level*) und der Reizantwort (*SCR*, *skin conductance response*). Die Einheit des Hautleitwertes ist das Siemens [S]. Der Hautwiderstand ist der Kehrwert des Hautleitwertes [Ohm]. Das Langzeitniveau des Hautwiderstands wird mit *SRL* bezeichnet (*skin resistance level*) und die Spontanänderung mit *SRR* (*skin resistance response*).

Streng genommen gehört der Hautleitwert bzw. -widerstand nicht zu den bioelektrischen Signalen, da zur Messung eine elektrische Spannung angelegt wird. Es existiert jedoch auch eine passive (*endosomatische*) Messmethode, bei der elektrische Effekte der Änderung des Hautpotenzials ohne externe Spannungsquellen gemessen werden (*skin potenzial level*, *SPL*).

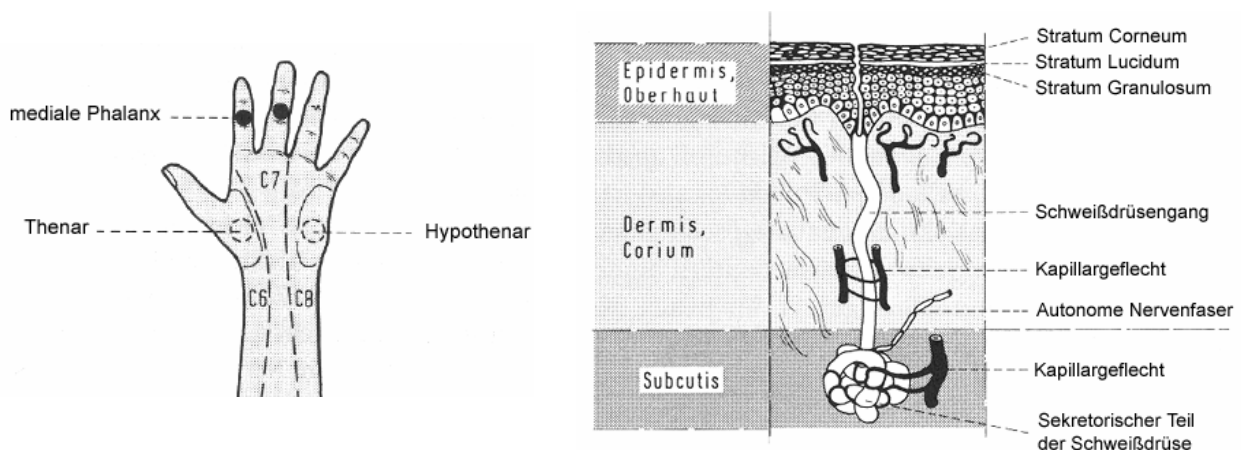


Abb. 2.16: (a) Typische Ableitungspunkte der EDA (b) Aufbau der Hautschichten und Schweißdrüsen

Die Ursache des veränderlichen Hautleitwertes liegt vor allem in der Schweißdrüsenaktivität, welche den elektrischen Widerstand der Epidermis verändert. Der Hautleitwert ist dadurch ein aussagekräftiger psychophysiologischer Parameter. Es finden sich Anwendungen in der Psychotherapie, in der Suchtbehandlung, im Stressmanagement und bei der Unterstützung von Entspannungstechniken (siehe 2.4 Biofeedback).

Die Messung des Hautleitwertes erfolgt meist zwischen Zeige- und Mittelfinger (jeweils an der medialen Phalanx) oder am Daumen- und Handballen (*Thenar* und *Hypothenar*, siehe Abb. 2.16a). Typische Leitwerte liegen zwischen 2 und 100 μS , die Werte können jedoch (auch inter-subjektiv) stark schwanken, sodass für eine ausreichend genaue Aufzeichnung Biosignalverstärker mit hoher Auflösung (16 Bit oder mehr) notwendig sind.

2.2 Nichtelektrische Biosignale

Im Folgenden werden einige Biosignale kurz beschrieben, die nicht direkt elektrisch messbar sind. Man benötigt geeignete Sensoren, die einen Wechsel der Modalität durchführen, damit sie elektrisch gemessen und digitalisiert werden können.

2.2.1 Atmung

Die Atmung dient als grundlegende Vitalfunktion zur Aufnahme von Sauerstoff und zur Abgabe von Kohlendioxid im Zusammenhang mit dem oxydativen Stoffwechsel. Zudem unterstützt die Atmung die Temperatur- und pH-Regulation und ermöglicht die gesprochene Kommunikation und das Riechen. Die Basis für die Inspiration und Expiration sind die Atemmuskeln (Zwerchfell und Interkostalmuskeln) und die Elastizität des Thorax- und Lungengewebes [STR00]. Standardmesssysteme in der *Pneumologie* sind die *Spirometrie* zur Messung des Atemvolumens und die *Pneumotachographie* zur Ermittlung der Atemstärke. Sie dienen zur Ermittlung der Vitalkapazität und zur Diagnostik von Defiziten und Lungenerkrankungen. Für das Biofeedback werden aufgrund der angenehmeren Verwendbarkeit andere Atemsensoren wie elastische Dehnungsmessstreifen, Temperaturfühler für die ausströmende Atemluft oder Mikrofone verwendet. Durch das Atemfeedback kann die Erlernung von Atemtechniken erleichtert oder ein ausgewogenes Verhältnis von Brust- und Bauchatmung trainiert werden.

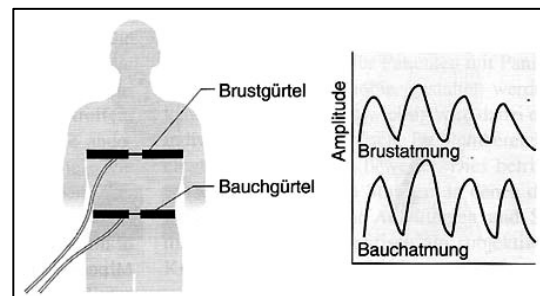


Abb. 2.17: Abnahme der Atmenaktivität durch dehnungssensitive Messgürtel

2.2.2 Körpertemperatur

Die Körpertemperatur beim gesunden Menschen liegt zwischen 36 und 37 Grad Celsius. (Die Extremitäten und oberen Gewebeschichten können geringere Temperaturen aufweisen). Das Wärmeregulationszentrum des Organismus ist der Hypothalamus (ein Teil des Zwischenhirns), wo die Thermorezeptoren des Körpers zusammenlaufen [KRE01]. Eine Regulation der Körpertemperatur erfolgt durch die Veränderung der Durchblutung und durch Schwitzen. Die Körpertemperatur schwankt tagsüber, in der Nacht ist sie geringer. Sinkt die Körpertemperatur unter 26 Grad oder steigt sie über 41 Grad ist dies lebensgefährlich. Unter körperlicher Belastung steigt die Temperatur auf bis zu 39 Grad an, der Körper kann dann höhere Leistungen erbringen. Die Messung der Temperatur erfolgt über einen wärmeempfindlichen Widerstand (*NTC*) oder berührungsfrei durch Messung der Infrarotabstrahlung. Moderne Temperaturfühler können die Körpertemperatur binnen weniger Zehntelsekunden messen. Im Biofeedback ist das Temperaturfeedback eine wichtige Methode, die aktive Kontrolle der Körpertemperatur kann dadurch erlernt werden. Diese Methode wird unter anderem zur Entspannung, zur Durchblutungsförderung und zur Bekämpfung von Phantomschmerzen eingesetzt [SHE05].

2.2.3 Blutvolumenamplitude

Bei der *Infrarot-Photoplethysmographie* werden Erweiterungen oder Verengungen der Blutgefäße (*Vasodilatation* bzw. *Vasokonstriktion*) durch die Reflexion von Infrarotlicht gemessen. Das Infrarotlicht wird von einer IR-Leuchtdiode ausgesendet, dringt ins Gewebe ein und wird zu einem Teil reflektiert. Das reflektierte Licht trifft auf einen Phototransistor, der seine Leitfähigkeit entsprechend ändert. Diese Methode eignet sich zur Abnahme der Herzfrequenz von der Körperperipherie, zur Messung der Blutvolumenamplitude und für das Vasokonstriktionstraining im Biofeedback.

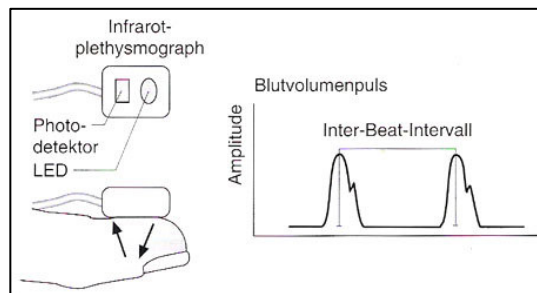


Abb. 2.18: Funktionsweise der Infrarot-Plethysmographie

2.2.4 Augenbewegungen

Die folgenden Methoden zur Messung von Augenbewegungen basieren im Gegensatz zum EOG auf optischen Prinzipien und werden ohne Elektroden gemessen:

2.2.4.1 Infrarotokulographie

Neben dem EOG kann die *Infrarotokularographie (IROR)* zur Registrierung von Augenbewegungen verwendet werden. Diese Messmethode basiert auf der Reflexion von Infrarotlicht durch die Hornhautoberfläche und der Registrierung dieser Reflexion durch eine Diodenzeile oder eine geeignete Kamera (siehe Abb. 2.19). Hierzu wird das Auge mit Infrarotlicht bestrahlt. Ein Vorteil dieser Methode ist der berührungsfreie Einsatz, wodurch Lesebewegungen oder Augenbewegungen zur Steuerung einer Computerschnittstelle komfortabel abgenommen werden können. Je nach Art der Montage der Messvorrichtung werden ortsfeste und kopfgestützte Apparaturen unterschieden. Ortsfeste Geräte haben den Vorteil des höheren Komforts, die möglichen Kopfbewegungen sind jedoch stark eingeschränkt [FIN05].

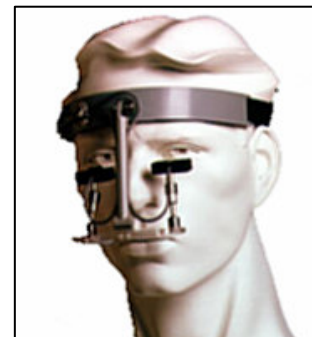


Abb. 2.19: IROR-System der Firma Microguide

2.2.4.2 Videookulographie

Bei der *Videookulographie* wird die Stellung des Auges durch Merkmalsextraktion aus einem Videobild ermittelt. Die zeitliche Auflösung des Verfahrens ist durch die Aufzeichnungsgeschwindigkeit der Kamera und die Geschwindigkeit der Merkmalsextraktion festgelegt. Gute Ergebnisse lassen sich durch hochauflösende Kameras mit ausreichender Bildfrequenz (> 50 Hz) erzielen. Durch die steigende Rechenleistung und die sinkenden Preise der Videoausrüstung gewinnt die Videookulographie zunehmend an Bedeutung. Zur Merkmalsextraktion werden die Helligkeitsunterschiede zwischen Iris und Bindehaut (*Limbus-Tracking*) oder zwischen Iris und Pupille (*Pupillen-Tracking*) verwendet. Oft findet auch hier die Reflexion von Infrarot-

licht an der Cornea (*Cornea-Reflex-Methode*) unterstützende Anwendung [SCN00]. Durch die Auffindung des Zentrums der Pupille mittels Bildanalyseverfahren und einer Kalibrierung mit der Kopfposition kann die Blickrichtung geschätzt werden. Ein Nachteil dieses Verfahrens ist die Fehleranfälligkeit gegenüber Kopfbewegungen, meist beträgt die Toleranz hier nur wenige Zentimeter. Dennoch bietet die Videookulographie vor allem beim Einsatz als Kommunikationshilfsmittel viele Vorteile gegenüber anderen Verfahren. Einige Beispiele für kommerzielle Produkte siehe Kapitel 2.4.6.

2.2.5 Stimme und Sprechen

Die Stimme zählt zu den wichtigsten Kommunikations- und Interaktionsmitteln des Menschen. Will man die Stimme digital aufzeichnen, benötigt man dafür ein Mikrofon und eine Digitalisiereinheit mit geeigneter Auflösung und Abtastrate (für gute Qualität etwa 16 Bit für den gegebenen Signalpegel, bei einer Samplefrequenz von etwa 10 kHz). Das Frequenzspektrum der Stimme liegt zwischen 200 und 5000 Hz.

Sprechen und Stimme haben in der Rehabilitations- und Unterstützungstechnik einen wichtigen Stellenwert, da sie für blinde und motorisch behinderte Menschen als Ersatz für verloren gegangene Fähigkeiten dienen können. Anwendungsbeispiele sind Spracherkennungssoftware, Sprechsteuerungen und Stimmsyntheseprogramme.

2.2.6 Gestik und Bewegungen

Gestik und Bewegungen gewinnen zunehmend Bedeutung in der Computerinteraktion. Die Erkennung von Emotionen anhand des Gesichtsausdrucks, der Körperhaltung und -bewegung ist mittlerweile ein eigener Forschungszweig in der Schnittstellenforschung geworden. Die Verwendung von Computern oder Robotern soll dadurch intuitiver und komfortabler gestaltet werden. Viele derartige Systeme basieren auf der Auswertung eines Kamerabildes. Durch geeignete Bildverarbeitungsalgorithmen wird versucht, den Gesichtsausdruck zu klassifizieren oder Körperposition und Aktivitäten der Gliedmaßen zu erkennen.

Zur Erkennung von Bewegungen können auch Sensoren für Beschleunigung (*Accelerometer*) oder Lageänderung (*Gyro-Sensor*) eingesetzt werden. Diese Instrumente sind als integrierte Schaltungen verfügbar und haben keine externen mechanischen Komponenten. Es sind ein-, zwei- und dreiachsige ICs erhältlich, die Werte können analog oder digital ausgelesen werden.



Abb. 2.20: Sensoren für Beschleunigung und Lageänderung

Möglichkeiten zur Messung von Kräfteinfluss bietet der Einsatz von *Piezo-Wandlern* oder resistiven *Druck-* oder *Flexionssensoren*. Je nach Einsatzgebiet können diese Sensoren eine Alternative zur Messung von Bewegungen mittels EMG darstellen.

2.3 Biofeedbacktherapie und ihre Anwendungen

Viele Biosignale des gesunden menschlichen Organismus sind willkürlich beeinflussbar, wie etwa die Atemfrequenz oder Bewegungen der Skelettmuskulatur. Bei anderen Biosignalen gelingt dies nicht oder nur durch entsprechendes Training. Beim Biofeedback werden die Messwerte von Vitalparametern optisch oder akustisch dargestellt, um eine willkürliche Beeinflussung zu erleichtern. In der Biofeedbacktherapie wird versucht, körperliche Funktionen durch Lernprozesse in therapeutisch wertvoller Form zu beeinflussen. So können Ziele wie Entspannung, die Vermeidung von Fehlhaltungen und Überreaktionen oder die Aktivierung von Vitalfunktionen oft ohne den Einsatz von Medikamenten erreicht werden. Die Biofeedbacktherapie eignet sich zur Unterstützung einer Physio- oder Psychotherapie und bringt in machen Bereichen Erfolge, wo traditionelle Methoden versagt haben. Sie ist in den USA bereits ein weit verbreiteter Ansatz zur Heilung bei bestimmten Indikationsgebieten und findet in Europa zunehmend Verbreitung. Biofeedback zeichnet sich durch hohe Akzeptanz bei Patientinnen und Patienten sowie durch die weitgehende Vermeidung von Nebenwirkungen aus. Die für das Biofeedback erforderlichen Apparate wie Messverstärker oder Computer dienen hier nur zum Erlernen der Selbstkontrolle über die gewünschten Körperfunktionen, sie werden nach einer erfolgreichen Therapie nicht mehr benötigt. Für die Durchführung des Biofeedback ist ein geeigneter Therapieplan zu erstellen, der meist mehrere Sitzungen umfasst. Die Auswahl der Methode und die Art der Rückmeldung sollte durch einen erfahrenen Therapeuten oder eine erfahrene Therapeutin durchgeführt werden, sodass die Anwendung entsprechend den lernpsychologischen Regeln und motivationspsychologischen Aspekten für die Person positiv wirkt, da der Lernprozess durch eine falsche Art der Rückmeldung auch gehemmt werden könnte.

Die kontinuierliche Rückmeldung des physiologischen Prozesses ohne wahrnehmbare Verzögerung und eine Bereitstellung geeigneter Darstellungsformen sind wichtige Eigenschaften der Biofeedbackinstrumente.

Biofeedback dient als ganzheitliche Methode auch der Aktivierung anderer Wirkfaktoren, die zu einer erfolgreichen Behandlung beitragen [RIE00]:

- Die Veränderung der Selbstwirksamkeitserwartung
Der Behandlungserfolg wird umso größer, je stärker die Person die Erfahrung macht, selbst Kontrolle über Funktionen und Bereiche zu erlangen, denen sie bisher hilflos ausgeliefert war
- Die Verbesserung der *Interozeption*
Bei vielen Erkrankungen werden Körpersignale fehlinterpretiert (z.B. bei Panikstörungen). Die Wahrnehmung und richtige Interpretation körperlicher Prozesse kann durch das Biofeedback gefördert werden.

Der positive Effekt von Biofeedback ist durch kontrollierte Therapiestudien belegt, bei manchen Krankheitsbildern (z.B. Migräne oder Inkontinenz) zählt Biofeedback zu den erfolgreichsten Behandlungsmöglichkeiten.

Zu den durch Biofeedback beeinflussbaren Körperfunktionen zählen: Muskelaktivität, Herzrate, Blutdruck, Schweißdrüsenaktivität (als allgemeines Maß der autonomen Erregung), Haut- und Körpertemperatur, elektrophysiologische Prozesse des Gehirns, periphere Durchblutung, Durchmesser von Blutgefäßen und Atemfunktionen.

2.3.1 Biofeedback bei chronischen Rückenschmerzen

Chronische Rückenschmerzen gehören zu den häufigsten Schmerzerkrankungen und rangieren als Ursache für Arbeitsunfähigkeiten an zweiter Stelle aller Krankheitsgruppen [HEU00]. Die *zervikalen* (Halswirbelsäule) und *lumbosakralen* (Lendenwirbelsäule) Regionen der Wirbelsäule sind am häufigsten von Schmerzen betroffen. Während die Beschwerden bei den meisten (ca. 90%) der Betroffenen nur von kurzer Dauer sind, chronifizieren die Beschwerden bei ca. 10% der Betroffenen und stellen eine massive Beeinträchtigung der Lebensqualität dar. Häufig sind degenerative oder funktionelle Veränderungen im Bereich der Bandscheibe, der kleinen Wirbelgelenke oder im Halte- und Stützapparat (also in den Muskeln und Bändern) die Ursache der Schmerzen. Bandscheibenvorfälle sind zwar am bekanntesten, zu- meist sind die Schmerzen jedoch muskulärer Genese und durch eine reflektorische Muskelverspannung, eine Verkürzung der tonischen und eine Schwächung der phasischen Muskulatur gekennzeichnet [HEU00]. Das psychophysiologische Konzept der klassischen Konditionierung nimmt hier einen zentralen Stellenwert ein: Schmerz führt zu einer reflexartigen Muskelverspannung, durch die er weiter verstärkt oder in weiterer Folge auch hervorgerufen wird. Diese physiologischen Prozesse können auf neutrale Reize, die während der schmerzauslösenden Situation präsent waren, konditioniert werden, sodass diese Reize alleine schon erhöhte Aktivität in den betroffenen Muskelgruppen und somit Schmerz auslösen können. Durch das Biofeedback wird versucht, gemeinsam mit dem Patienten oder der Patientin individuelle Zusammenhänge zwischen muskulären Streßreaktionen, Schmerzphysiologie und Schmerzwahrnehmung zu erarbeiten und die Rolle von Kognitionen und Verhaltensweisen bei der Schmerzaufrechterhaltung zu verdeutlichen. Durch die Steigerung von Eigenaktivität und Selbstkontrollmöglichkeiten wird versucht, Fehlhaltungen und Dysbalancen in spezifischen Muskelgruppen zu beeinflussen. Die geeignete Methode ist das EMG-Feedback, wobei zuerst ein *scanning* der relevanten Muskulatur im Sitzen und Stehen durchgeführt wird, um betroffene Muskelgruppen zu identifizieren oder Asymmetrien zwischen linker und rechter Seite festzustellen. Dynamische EMG-Messungen zeigen die Zusammenarbeit beteiligter Muskelstrukturen an Bewegungsabläufen bzw. die Entspannungsphase nach einer Belastung. Bei der Erstellung eines psychophysiologischen Stressprofils wird die Reaktion im EMG bei der Konfrontation mit emotional bedeutsamen Stressoren überprüft. Ein Bezug auf Normwerte sollte nur unter Vorsicht geschehen, da die Eichung der Messinstrumente nicht immer gegeben ist. Liegt eine Muskelverspannung vor, wird das EMG-Training für die Entspannung der betroffenen Muskulatur konfiguriert und eingesetzt. Lässt sich bei einer tastbaren Muskelverhärtung kein erhöhter EMG-Wert feststellen, so deutet das auf eine Muskelverkürzung hin, und das EMG wird zur Unterstützung von Dehnungsübungen eingesetzt.

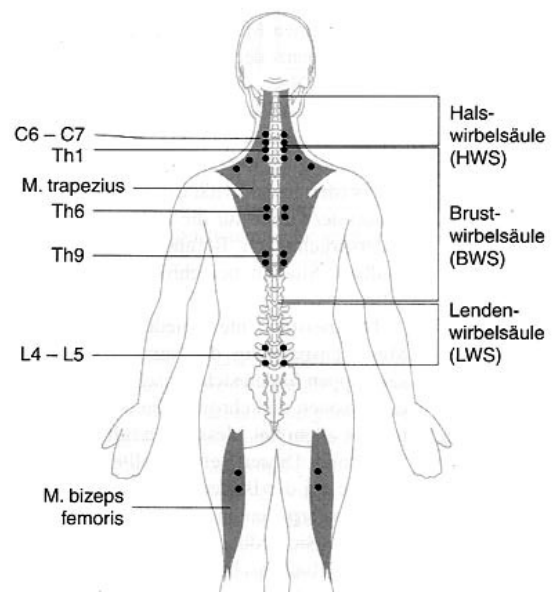


Abb. 2.21: Ableitungspunkte für das EMG der Rückenmuskulatur

2.3.2 Biofeedback bei Spannungskopfschmerz und Migräne

Neunzig Prozent aller Kopfschmerzen gehen auf zwei Kopfschmerzarten zurück: den Spannungskopfschmerz und die Migräne. Beim Spannungskopfschmerz spielen muskuläre Verspannungen eine wichtige Rolle, obwohl dadurch das Schmerzgeschehen nicht vollständig erklärt werden kann. Auch die Entwicklung einer Migräne ist nicht vollständig geklärt, wird aber verbreitet mit einer Instabilität der Blutgefäßregulation im Kopf und eine erhöhte Sensibilität auf aversive Umweltreize (Stress im weitesten Sinne) in Zusammenhang gebracht. Hierbei kann es zu einer exzessiven Verengung der Blutgefäße (Vasokonstriktion), gefolgt von einer starken Erweiterung (Vasodilatation) kommen, die von Entzündungsreaktionen und einer Sensibilisierung der Schmerzrezeptoren in den Gefäßwänden begleitet wird [HEU00a].

Chronische Spannungskopfschmerzen und Migräne unterscheiden sich auch in der Art des Auftretens (konstant bzw. anfallsartig) und in der Intensität (leicht bis mittel bzw. schwer). Bei der Biofeedbackbehandlung wird versucht, möglichst nahe an den schmerz erzeugenden physiologischen Vorgängen anzusetzen.

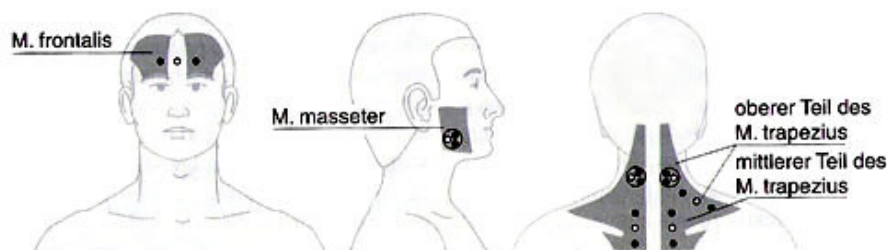


Abb. 2.22: Ableitungspunkte für das EMG der Gesichts- und Nackenmuskulatur

Bei Spannungskopfschmerz wird mit EMG-Feedback versucht, die Anspannung der Kiefer- und Nackenmuskulatur sowie der Stirnmuskulatur zu senken. Geeignete Ableitungspunkte sind der Kiefermuskel (*M. masseter*), der Stirnmuskel (*M. frontalis*) und die Schulter- und Nackenmuskulatur (*M. trapezius*). Zur Erfassung des vegetativen Nervensystems werden die elektrodermale Aktivität und die Fingertemperatur abgeleitet. Nach einer Entspannungsphase und dem Messen der Normalaktivität (*Baseline*) wird ein Stresstest (z.B. durch Rechenaufgaben und emotionale Fragen) und eine anschließende Ruhephase durchgeführt [HEU00a]. Die Auswertung zeigt, mit welchen Körpersystemen die Person besonders stark reagiert hat, und wie die Entspannungsphase gelang. Daraus kann das Vorgehen für die weiteren Biofeedbacksitzungen abgeleitet werden.

Zur Behandlung von Migräne mittels Biofeedback werden neben dem EMG-Feedback zur Minderung von Verspannungen bevorzugt das Temperatur-Biofeedback (Handerwärmungstraining) zur Steigerung der peripheren Durchblutung und das *Vasokonstriktionstraining* eingesetzt.

Beim Vasokonstriktionstraining wird der Blutfluss (genauer die Blutvolumenamplitude, *blood volume pressure, BVP*) in der Schläfenarterie durch *Infrarot-Photoplethysmographie* gemessen. Hierdurch lernt die Person, den Dehnungszustand der Blutgefäße willkürlich zu beeinflussen und eine Verengung oder Erweiterung der Schläfenarterie herbeizuführen, um vor oder während eines Anfalls gegensteuern zu können. Bei etwa 60% der Patienten und Patientinnen lässt sich dadurch eine Reduktion der Migräneanfälle um 50% erreichen. [HEU00a].

2.3.3 Biofeedback bei Hypertonie

Unter Bluthochdruck (Hypertonie) leiden 15% bis 20% aller Erwachsenen in den Industrienationen, an den Folgen der Hypertonie sterben dreimal mehr Menschen als an Krebs [MAR00]. Erbliche Belastung, ethnische Zugehörigkeit, Übergewicht, Persönlichkeitsmerkmale und Stress stellen die wichtigsten Risikofaktoren für die Entstehung einer essentiellen Hypertonie dar. Bei fortgeschrittener Hypertonie treten Organveränderungen wie eine Herzvergrößerung (*Hypertrophie*), Schädigungen des Gehirns, der Nieren oder des Augenhintergrundes ein. Die entwicklungsgeschichtlich sinnvolle Erhöhung des Blutdrucks bei Kampf oder Flucht erweist sich in modernen Zusammenhängen oft als kontraproduktiv, und häufige Erhöhungen des Blutdrucks können zu Gefäßveränderungen führen, die einen hohen Blutdruck weiter begünstigen. Der Blutdruck wird durch das Herzzeitvolumen (das Produkt aus Herzfrequenz und Schlagvolumen) und den totalen peripheren Widerstand (bedingt durch die Elastizität der großen Arterien und den Durchmesser der Arteriolen) bestimmt. Er wird normalerweise durch verschiedene Regulationssysteme (die *Barorezeptoren* in der Aorta, das *Renin-Angiotensin-Aldosteron-System*) bedarfsgerecht angepasst. Eine lang anhaltende Blutdruckerhöhung kann unter bestimmten Umständen jedoch zu einer Adaption dieser Regulationssysteme auf die erhöhten Werte führen [MAR00].

Meist wird der Blutdruck durch das Manschettendruckverfahren nach *Riva-Rocci* mit einer Druckmanschette gemessen. Das Auftreten des Blutflusses bei einem bestimmten Manschettendruck wird dabei durch Geräusche in der Arterie festgestellt. Der Blutdruck ist durch zwei Werte charakterisiert: den systolischen Blutdruck (höchster Druck zum Zeitpunkt des Zusammenziehens des Herzmuskels) und den diastolischen Druck (niedrigster Wert, bei entspanntem Herzmuskel). Von einer Hypertonie spricht man bei wiederholter Messung über 140 / 90 mmHg. Für das Biofeedback des Blutdruckes ist das *Riva-Rocci*-Verfahren nicht geeignet, da eine kontinuierliche Rückmeldung der aktuellen Werte nötig ist. Dazu eignen sich Geräte mit automatischer Regelung des Manschettendrucks und Auswertung des Blutvolumens durch die fotoelektrische Messung (*Plethysmographie*). Diese Geräte sind zwar im Vergleich zum *Riva-Rocci*-Verfahren ungenau, für das intersubjektive Biofeedback aber dennoch geeignet. In mehreren Sitzungen werden der Erwerb und die Verbesserung der Entspannungsfähigkeit, die Identifikation und Beeinflussung von Blutdruckspitzen und ein angemessener Umgang mit Ärger trainiert. Dabei wird das kontinuierliche Biofeedback mit psychologischen Methoden wie Rollenspielen und der Übung von Stressbewältigungsstrategien kombiniert. Biofeedback kann eine medikamentöse Behandlung der Hypertonie und die Änderung des Lebenswandels durch salzarme Ernährung, mehr Bewegung oder eine Reduktion des Genussmittelkonsums nicht ersetzen, aber in der Therapie unterstützend wirken und zur Reduktion der Medikamente beitragen [MAR00].

2.3.4 Biofeedback bei Somatisierungspatienten und bei Angststörungen

Unter somatoformen Störungen versteht man körperliche Beschwerden, die durch organische oder pathophysiologische Faktoren nicht ausreichend erklärt werden können. Die Beschwerden sind aber weder eingebildet noch vorgetäuscht. Diagnosekriterien für eine Somatisierungsstörung sind das Vorliegen von mindestens vier unterschiedlichen Schmerzsymptomen gastrointestinaler, sexueller und pseudoneurologischer Art. Liegen einige aber nicht alle dieser Beschwerden vor, spricht man

von einem Somatisierungssyndrom. Schätzungsweise leiden 4% - 11% aller Personen in westlichen Industrieländern unter einem behandlungsbedürftigen somatoformen Beschwerdebild [NAN00]. Oft entwickeln die Betroffenen die Angst, es könnte eine ernsthafte Erkrankung die Ursache der Beschwerden sein (*Hypochondrie*). Durch die gesteigerte Aufmerksamkeit werden die wahrnehmbaren Symptome katastrophierend fehlinterpretiert, und die intensiven Ängste bringen neue Symptome hervor. Ein kognitiv-verhaltenstherapeutischer Ansatz kombiniert mit Biofeedback ist eine mögliche Behandlungsstrategie. Das Biofeedback dient hierbei zur Modifikation des organmedizinischen Krankheitskonzeptes, zu einer Steigerung der Selbstwirksamkeitserwartung und zur Motivationsförderung für die psychotherapeutische Behandlung. Ziel ist, dass der Patient oder die Patientin wiederholt die Erfahrung macht, dass psychische Prozesse biologische Funktionen verändern. Dadurch kann eine Uminterpretation von einem organmedizinischen in ein psychophysiologisches Krankheitsbild unterstützt werden [NAN00]. Als Methode eignet sich die Mehrkanalableitung mit EDA, EMG, Temperatur- und Atmungsfeedback. In den Sitzungen werden wiederholt Belastungs- und Stressprovokationstests (Fokussierung auf körperliche Missempfindungen, Leistungsstressoren, Vortragen eines Liedes, Hyperventilation usw.) mit Entspannungsphasen kombiniert und die Messergebnisse gemeinsam mit den Betroffenen analysiert.

Ähnlich ist das Vorgehen bei der Behandlung von Angststörungen. Diese sind bei Frauen die häufigste, bei Männern nach Suchterkrankungen die zweithäufigste psychische Erkrankung. Auch hier ist das Biofeedback vor allem als unterstützende Methode zur psychotherapeutischen Behandlung wirksam. Die größten Gruppen der Angststörungen sind die Panikstörung (plötzliche Angstattacken, gepaart mit heftigen körperlichen Reaktionen), die *Agoraphobie* (Platzangst), die soziale und die spezifische Phobie (Angst vor peinlichem Verhalten in Gesellschaft bzw. vor bestimmten Objekten oder Situationen), die posttraumatische Belastungsstörung (Erinnerung an eine traumatisierende Situation) sowie die generalisierte Angststörung (eine Vielzahl von Auslösern) [KRO00]. Biofeedback dient auch hier mithilfe einer Mehrkanalableitung während der Konfrontation mit angstauslösenden Reizen (*Exposition in vivo*) und der anschließenden Entspannungsphase der Identifikation der Reaktion in den verschiedenen Körpersystemen bzw. der Kontrolle und Unterstützung der Entspannungsfähigkeit. Ist die Exposition nicht möglich (durch individuelle Gründe oder die Art des Reizes) kann die systemische Desensibilisierung angewendet werden, bei der zuerst eine tiefe Muskelentspannung trainiert wird und dann Reize aus einer zuvor ermittelten Angsthierarchie visualisiert werden (*Exposition in sensu*). Bei Anzeichen von Erregung wird die Visualisierung abgebrochen und die Person versucht wieder zu entspannen. Geeignete Parameter für das Feedback sind EMG, EDA, Blutdruck, Puls und Temperatur sowie die Atmung. Ziele sind die Umbewertung der Angstanfälle, die Vermittlung eines Erklärungsmodells und die Reduktion der allgemeinen Anspannung, wodurch auch die Angstanfälle seltener werden [KRO00].

2.3.5 Biofeedback in der Therapie des chronischen Tinnitus

Chronische Ohrgeräusche, welche der oder die Betroffene als Rauschen, Zischen, Brummen oder Pfeifen im Kopf wahrnimmt, die aber keiner äußeren Schallquellen zuzuordnen sind, werden als Tinnitus bezeichnet. Es können Schwankungen in der Lautstärke und Geräuschzusammensetzung über Stunden bis Wochen auftreten. Weitere mögliche Symptome sind Hörverlust und Schwindel. Zu den vielfältigen Auslösern des Tinnitus gehören Lärmschädigungen, Hörsturz und neurologische Erkran-

kungen, die mit einem Hörverlust einhergehen [KRO00a]. Ist die Person von den Symptomen nur gering belastet, spricht man von einem kompensierten Tinnitus. Wird die Lebensqualität stark eingeschränkt (Depressionen, Schlafstörungen, Ängste), spricht man vom dekompenzierten Tinnitus. Ein Hauptziel des Einsatzes von Biofeedback ist eine Unterbrechung des Kreislaufes Aufmerksamkeit - dysfunktionale Bewertung des Geräusches - Stressreaktion und Tinnitus-Verschlimmerung. An der Intensivierung der Belastung sind oft Verspannungszustände und Fehlhaltungen der Kiefer- und Nackenmuskulatur oder nächtliches Zähneknirschen (Bruxismus) beteiligt. Die angewandten Biofeedbackmethoden sind EMG-Ableitungen der Kinn-, Nacken- und Gesichtsmuskulatur, sowie die Ermittlung sympathischer Aktivierung durch EDA und Temperatur. Nächtlicher Bruxismus kann durch die Auslösung eines akustischen Warnsignals bei hohen EMG-Werten im Kieferbereich korrigiert werden, der die betroffene Person zwar nicht gänzlich weckt, aber ausreicht, um die Verspannung im Kieferbereich zu bemerken und ihr entgegenzuwirken. Obwohl der chronische Tinnitus in vielen Fällen nicht durch das Biofeedback heilbar ist, gelingt doch häufig ein besserer Umgang mit dem Symptom und eine Linderung der Beschwerden [KRO00a].

2.3.6 Biofeedback zur Behandlung von Inkontinenzproblemen

Störungen der Beckenbodenfunktionen, wie Urin- und Stuhlinkontinenz, sind eine Belastung für viele Menschen, werden aber aufgrund der Tabuisierung des Intimbereiches selten besprochen. Besonders häufig tritt die Urininkontinenz bei Frauen nach einer Geburt und bei älteren Menschen auf. Für die Urinzurückhaltung (*Retention*) und die Entleerung (*Miktion*) sind im Wesentlichen drei Muskelgruppen zuständig: Der Blasenentleerer (*Detrusor vesicae*) ist ein Muskel, der einen Teil der Harnblase bildet und durch seine Kontraktion den Urin aus der Harnblase presst. Sein Gegenspieler ist der *Musculus sphincter internus*, der den Blasenhal umgürtet (die Region, die den Beginn der Harnröhre bildet). Dieser innere Schließmuskel komprimiert den Harnröhreneingang und verhindert so den Urinaustritt. Der äußere Schließmuskel (*M. sphincter externus* der Harnblase, Teil des Beckenbodens, in Abb. 2.23 nicht eingezeichnet) ermöglicht eine willkürliche Öffnung und Schließung der Harnblase [CUN00].

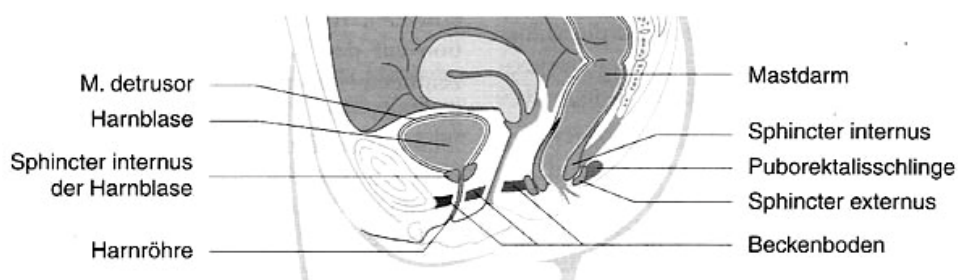


Abb. 2.23: Lage wichtiger Muskelgruppen für die Kontinenzreaktion

Das Zusammenspiel dieser Muskelgruppen ist durch verschiedene, sich hierarchisch beeinflussende Zentren im Rückenmark und im Gehirn gewährleistet. Zunehmende Füllung der Blase aktiviert die Nervenzellen des Blasenentleerungszentrums (Miktionszentrum) im Sakralmark (in den untersten Abschnitten des Rückenmarks), wobei die Aktivität durch sympathische Blaseninnervation so lange unterdrückt wird, bis die willkürliche Hemmung der Blasenentleerung bewusst beendet wird [CUN00].

Als hauptsächliche Formen der Urininkontinenz werden die *Stressinkontinenz* (unwillkürlicher Abgang von Urin z.B. beim Husten, Heben oder Lachen), die *Urge-Inkontinenz* (häufiger, heftiger Harndrang, geringes Harnvolumen) und die *Blasenhypereflexie* (häufiger Harndrang durch neurologische Störungen) unterschieden. Durch Beckenbodentraining lässt sich die Urinkontinenz in vielen Fällen verbessern. Biofeedback kann dieses Training effektiv unterstützen, und je nach Problemstellung gezielt an den jeweiligen Problemzonen ansetzen [CUN00]:

- durch Druckaufnehmer in der Blase lässt sich die Detrusoraktivität trainieren
- durch Rückmeldung der Kontraktionskraft der Beckenbodenmuskulatur kann die Anhebung der Kontraktionsstärke und -dauer geübt werden
- durch simultane Abnahme der Bauchdeckenspannung und der Kontraktionskraft des Beckenbodens lässt sich das richtige Zusammenspiel trainieren

Hauptsächlich wird auf die schwieriger durchzuführende direkte Abnahme der Detrusoraktivität verzichtet und vor allem die Aktivität der Beckenbodenmuskulatur gemessen. Die geeigneten Sensoren sind flüssigkeitsperfundierte Katheter oder Ballonkatheter zur Druckmessung (*Manometrie*) sowie Vaginal- bzw. Analsonden zur Messung des EMGs der Beckenbodenmuskulatur. Die kontinuierliche Rückmeldung der Aktivität erlaubt die sukzessive Anhebung der Stärke und Dauer der Kontraktion, sowie ein Training der Schließkraft bei problematischen Körperhaltungen.

Ebenso kann ein besserer Umgang in Stresssituationen und eine Gewöhnung an größere Füllvolumina der Blase bei Urge-Inkontinenz erreicht werden. Der Einsatz mobiler Geräte ermöglicht ein Training zuhause und verbessert meist den Therapieerfolg [CUN00].

Der Aufbau und die Funktion des Schließmuskelapparates für die Stuhlkontinenz weist deutliche Parallelen zu dem der Blase auf, auch hier spielt der Beckenboden eine bedeutsame Rolle, und die Mechanismen von Entleerung und Schließung werden durch ein System von autonomen und willkürlich gesteuerten Muskelgruppen reguliert. Der innere Sphinkter (*Musculus sphincter ani internus*) ist ein autonom gesteuerter, glatter Ringmuskel, der für den Verschluss des Darmes zuständig ist, unterstützt durch den *M. puborectalis*, der Teil des Beckenbodens ist. Füllt sich das Rektum durch die Peristaltik des Dickdarmes, erschlafft der innere Sphinkter und der äußere Sphinkter (*M. sphincter externus*) sowie der Beckenboden kontrahieren reflektorisch (Kontinenzreaktion). Diese Muskelgruppen können zur Darmentleerung willkürlich gesteuert werden. Das Biofeedback kann hier vor allem bei verminderter Willküranspannung, bei herabgesetzter Sensibilität des Rektums und zur Koordination von Kontraktion und Bauchdeckenanspannung eingesetzt werden. Als Sensorium eignen sich Ballonkatheter und EMG-Sonden. Bei etwa 80% der Patienten und Patientinnen lässt sich durch das Biofeedbacktraining die Frühzeitigkeit der Wahrnehmung, die Schnelligkeit der Reaktion sowie die Kraft und Dauer der Sphinkteranspannung verbessern [CUN00].



Abb. 2.24: EMG-Sonde für das Beckenbodentraining

2.3.7 Biofeedback bei Neurologischen Erkrankungen

Schlaganfall (Hirnblutung) und Gefäßverschluss (ischämischer Insult) sind häufige neurologische Erkrankungen. Nach einem Schlaganfall leidet die Mehrzahl der Betroffenen unter motorischen Störungen, die durch eine intensive Physiotherapie oft weitgehend korrigiert werden können. Die Förderung von Muskelaktivität und die Reduktion von Spasmen zur Erlangung der Selbstkontrolle über Bewegungen sind vorrangige Ziele der Therapie, zur deren Erreichung EMG-Biofeedbackverfahren beitragen können. Biofeedback kann zur Überbrückung fehlender Bewegungsrückmeldung dienen (siehe Abb. 2.25). Unangemessene motorische Reaktionen können dadurch leichter bewusst werden, und der Gebrauch alternativer Strukturen wird gefördert.

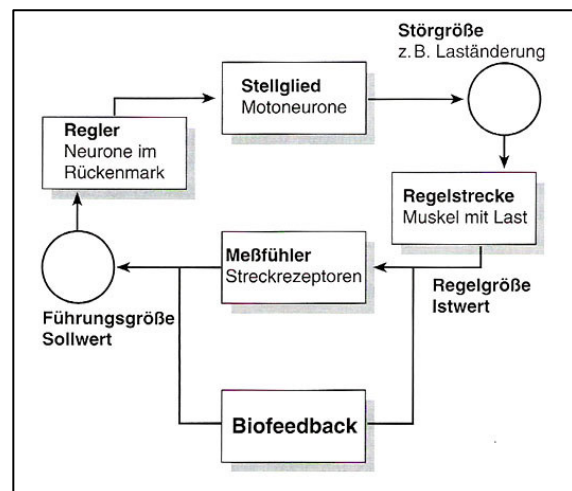


Abb. 2.25: Verbesserung der Bewegungsrezeption durch Biofeedback

Nach einer Hirnschädigung setzen mehrere Mechanismen der Reorganisation ein, etwa die vermehrte Aussprossung von Neuronen am Rand der Gewebeläsion oder die Übernahme gestörter Funktionen durch intakte Strukturen des Gehirns [MÜL00]. Diese Erholungsprozesse sind durch Training beeinflussbar. Hier unterstützt Biofeedback die Wahrnehmung intrinsischer, für den Patienten oder die Patientin kaum beobachtbarer Prozesse. Bei halbseitigen Lähmungen kann durch das gleichzeitige Ableiten der gesunden Seite eine Vergleichsmöglichkeit für das Bewegungsmuster geschaffen werden, was ein Wiedererlernen der Bewegung erleichtert. Durch visuelles Feedback der Standstabilität (gemessen durch eine Messplattform) können Gleichgewichtsstörungen erfolgreich behandelt werden [MÜL00]. Eine EMG-getriggerte Muskelstimulation ermöglicht die Aktivierung schwer gelähmter Muskeln. Hierbei werden kleine Muskelpotenziale, die für eine Bewegung nicht ausreichen, zur Auslösung einer funktionellen Elektrostimulation (FES) des betroffenen Muskels eingesetzt.

2.3.7.1 Neurofeedback bei epileptischen Anfällen

Epilepsien sind hochamplitudige, hochfrequente Serien von Entladungen der Hirnnerven, die auf eine Störung im intra- und extrazellulären Ionenhaushalt bzw. im Transmittergleichgewicht zurückzuführen sind [STR00]. Man unterscheidet fokale (partielle) Anfälle, die einen klaren Ausgangspunkt im EEG haben und von sehr kurzer Dauer sind, von komplex-fokalen Anfällen, bei denen die Person nicht mehr situationsgerecht reagieren kann, und generalisierten Anfällen, die von einer totalen Bewusstlosigkeit und Muskelzuckungen gekennzeichnet sind und nicht im EEG lokalisiert werden können (*Grand Mal*, *Petit-Mal-Absence*, *Myoklonie*). Während ein Teil der Epilepsiepatienten bzw. -patientinnen gut auf eine medikamentöse Therapie zur Steigerung des neuronalen Hemmpotenzials reagiert, erweisen sich andere als medikamentenresistent und es bedarf anderer Therapieformen. Da in mehreren Studien kurz vor und während eines epileptischen Anfalls eine erhöhte kortikale Negativie-

ung festgestellt wurde (welche einer gesteigerten Erregbarkeit entspricht), besteht im Biofeedbacktraining Langsamer Hirnpotenziale (*slow cortical potentials, SCP*) die Möglichkeit, eine geeignete Gegenstrategie zu entwickeln: Erlernt der Proband bzw. die Probandin durch das Feedback des EEG-Signals, bewusst eine Positivierung der langsamen Potenziale herbeizuführen, so kann diese Technik bei Anzeichen eines Anfalles eingesetzt und damit die Anfallswahrscheinlichkeit vermindert werden [STR00].

2.3.7.2 Neurofeedback bei Aufmerksamkeitsstörungen

Die Aufmerksamkeits-Hyperaktivitätsstörung (*Attention Deficit Hyperactivity Disorder, ADHD*) tritt vermehrt bei Kindern und Jugendlichen auf und äußert sich in Lern- und Konzentrationsschwierigkeiten oder Abwesenheit. Die Ursache ist oft eine Unterversorgung mit dem Neurotransmitter *Dopamin* und eine daraus resultierende gestörte Informationsweiterleitung im Gehirn: die Reize können nicht angemessen gefiltert werden und das Kind wirkt impulsiv, träumerisch-abwesend oder unaufmerksam.

Im quantitativen EEG (*QEEG*) zeigt sich ADHD meist durch stark ausgeprägte Frequenzen im Theta-Bereich (4-7 Hz) und wenig Aktivität im Beta-Bereich (13-21 Hz), siehe Abb. 2.26. Es gibt aber auch Formen von ADHD, die sich durch eine Asymmetrie der beiden Hemisphären oder durch starke frontale Alphawellen (7-12 Hz) äußern [DEM05]. Allgemein lässt sich sagen, dass bei ADHD die Frequenzzusammensetzung des QEEG signifikant von den Normwerten abweicht.

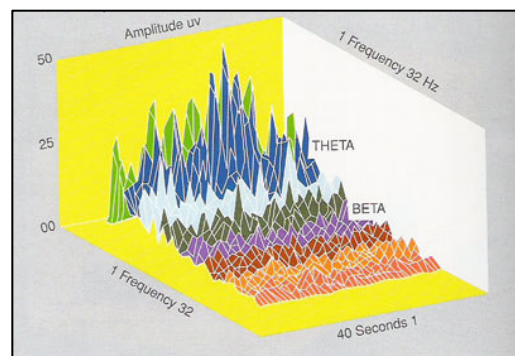


Abb. 2.26: Dominante Theta-Aktivität im EEG

Bei einer Therapie von ADHD durch Neurofeedback wird durch die kontinuierliche Rückmeldung der Frequenzbandaktivität und geeignete Feedback-Protokolle versucht, eine Verschiebung der Frequenzverteilung in Richtung der physiologischen Werte hervorzurufen. Dies wird durch motivierende opto-akustische Rückmeldung oder durch den Einsatz von Feedback-Spielen erleichtert. Neurofeedback stellt bei der Therapie von ADHD eine Alternative zur medikamentösen Behandlung dar oder kann diese effektiv unterstützen [DEM05].

2.4 Unterstützungstechnik für Menschen mit motorischen Behinderungen

Der Einsatz von Biosignalen und bioelektrischen Signalen zur Herstellung alternativer Computerschnittstellen (*Human Computer Interfaces, HCI*) wird seit geraumer Zeit erforscht. In den letzten Jahren entwickelte sich dieser Bereich stark, er ist mittlerweile ein Spezialgebiet der Medizintechnik, und viele verwendbare Produkte sind auf dem Markt. Elektronische Unterstützungsgeräte zur Kommunikation, Interaktion und Umgebungssteuerung sind für viele Menschen zu einer unverzichtbaren Hilfe im Alltag geworden. Die Kombination verschiedener Lösungen wird durch kabellosen Betrieb und ad hoc Netzwerkverbindungen wesentlich erleichtert (*Bluetooth, Zig-Bee, Wireless USB*), der Einsatz des Computers als verbindendes oder zentrales Steuer-

element bietet sich an. Der folgende Überblick zeigt exemplarisch einige herkömmliche Eingabehilfen und Beispiele zur Verwendung von Biosignalen als Eingabehilfsmittel für Menschen mit motorischer Behinderung.

2.4.1 Hilfsmittel für die direkte Eingabe

Viele relativ einfache Eingabehilfsmittel für körperlich behinderte Menschen bewähren sich aufgrund ihrer direkten Funktionsweise und unmittelbaren Wirksamkeit.

Dazu gehören der **Mundstab** und der **Kopfstab** zur Betätigung von Schaltern und zur Verwendung der Computertastatur. Ist die Bewegung von Kopf und Nackenbereich kaum eingeschränkt, ist der Umgang mit dem Mundstab erstaunlich gut erlernbar, sodass geübte Personen eine Schreibgeschwindigkeit von 80 Anschlägen pro Minute und mehr erreichen, was die Kapazität vieler anderer Eingabehilfsmittel übertrifft. Dennoch stellt eine dauernde Verwendung des Mundstabes eine erhebliche Belastung für Kiefer, Kinn- und Nackenmuskulatur dar, sodass Alternativen sicherlich für viele Menschen interessant sind.

Gelingt eine zielgerichtete, koordinierte Bewegung nur schwer, sind Hilfsmittel wie **Tastaturschablonen**, **Großfeldtastaturen** oder auch **Kleintastaturen** für eingeschränkten Bewegungsradius einsetzbar. [ZAG04]

Kinnjoysticks können zur Steuerung eines Rollstuhles oder auch als Eingabegerät für den Computer genutzt werden. Besteht eine eingeschränkte Beweglichkeit der Gliedmaßen, eignen sich spezielle **Trackballs**, **Mini-Joysticks**, oder auch **Fußsteuerungen** für die Computermaus. Maussimulatoren dienen zur Vereinfachung der zweidimensionalen Bewegung des Maus-Cursors mittels Tasten oder anderer Eingabesysteme. Mögliche Varianten reichen von der **5-Tasten-Maus** (direkte Auswahl einer Richtung und der Klickfunktion) bis zum Ersatz der Mausfunktion mit nur einem Taster. Hier wird eine Linie horizontal über den Bildschirm geführt, und der Schalter in dem Moment gedrückt, wo sich die Linie über dem gewünschten Objekt befindet. Danach wird der Vorgang in der vertikalen Richtung wiederholt. Auf diese Weise kann eine beliebige Stelle am Bildschirm geklickt werden [ZAG04].

Ein integriertes Konzept für den Musersatz bietet z.B. die **IntegraMouse** der Firma *LifeTool*. Die Mausfunktionen können durch die Betätigung eines sensiblen Hebels mit den Lippen und durch einen Saug-Blase-Schalter ausgeführt werden. Der Preis der IntegraMouse beträgt etwa 2100 Euro. (<http://www.lifetool.at>)



Abb. 2.27: IntegraMouse der Firma *LifeTool*

2.4.2 Scanning und alternative Schaltmechanismen

Für die Computerverwendung mit einer einzigen Taste existiert eine große Auswahl an Software, die komplexe Aufgaben wie das Schreiben oder die Steuerung der Umgebung durch so genannte **Scanning-Verfahren** erlaubt. Hierbei gibt der Computer ein Set von möglichen Entscheidungen vor, die kontinuierlich durchlaufen werden. Der Benutzer drückt die Taste in dem Moment, wo die gewünschte Option aktiv ist.

Je nach den persönlichen Bedürfnissen können zur Erzeugung der Schaltinformation eine ganze Reihe von Möglichkeiten verwendet werden [ZAG04]:

- Näherungssensoren
- Beschleunigungssensoren
- Neigeschalter oder Fingerbeugeschalter
- Berührungsschalter
- Saug-Blase-Schalter
- Gaumen-Zungen-Schalter
- Lidschlagschalter

Ist zwar die Verwendung einer Steuerung für den Mauszeiger, nicht aber das zusätzliche Auslösen eines Schalters möglich, kann durch geeignete Software die Verweilzeit des Mauszeigers an einer bestimmten Stelle einen Mausklick auslösen (**Dwell-Schalter**, *dwelling*). Die Dwell-Zeit kann eingestellt werden, und meist steht ein Auswahlfenster für alternative Klick-Funktionen wie Rechtsklick, Doppelklick und Ziehen (Drag and Drop) zur Verfügung.

2.4.3 Sprach- und Umgebungssteuerungen

Spracherkennungssysteme für die Eingabe am Computer und zur Steuerung der Haustechnik sind eine entscheidende Erleichterung für viele körperlich behinderte Menschen. Sinnvolle Systeme zeichnen sich durch einen universellen Einsatz mit vielen Geräten des Haushalts aus. Dies wird durch eine Lernfunktion für die Signale der Infrarot-Fernsteuerungen der Geräte und durch die Aussendung von Funksignalen für spezielle Haustechnik ermöglicht. Auf diese Weise können Fernseher, Videorecorder, Sat-Receiver, Musikanlage und durch den Einbau entsprechender Effektoren auch Fenster, Klimaanlage, Beleuchtung, Telefon oder Kopfteil des Bettes selbstständig gesteuert werden. Ein Beispiel ist der **Sicare Pilot** der *Dr. Hein GmbH*, welcher in der Grundausführung etwa 2.500 Euro und mit Funkoption etwa 3500 Euro kostet. (<http://www.sicare.de>). Meist werden die Sprachkommandos trainiert, das System arbeitet also sprecherabhängig. Dies ermöglicht gute Erkennungsraten für die Stimme der bedienenden Person und eine weitgehende Unempfindlichkeit gegenüber den Stimmen anderer Personen [ZAG04].



Abb. 2.28: Sicare-Pilot Umgebungssteuerung

2.4.4 Interaktion durch EMG oder EOG

Eine Reihe von Entwicklungen verwendet Muskelpotenziale (EMG) oder elektrisch gemessene Augenbewegung (EOG) als Eingabemodalität zur Auswahl von Selektionen oder als Alternative zur Computermaus. EMG-Signale erlauben eine sensitive Messung kleinster Muskelbewegungen und ermöglichen dadurch die Betätigung eines (realen oder virtuellen) Tasters zur Verwendung von Scanning-Software oder auch die proportionale Steuerung von Eingabewerten durch die Messung der Muskelkontraktion. Mit EOG und/oder EMG lässt sich die Position des Mauszeigers steuern oder durch die Erkennung des Lidschlages eine Auswahl treffen. Die folgenden Beispiele zeigen mögliche Anwendungen dieser Techniken.

2.4.4.1 Musersatz durch EMG

Barretto und Scargle (2000) verwendeten die EMG-Signale von Gesichtsmuskeln (*M. temporalis* und *M. frontalis*) zur Bewegung des Maus-Cursors und zur Auslösung von Mausklicks. Die charakteristische Frequenzverteilung der EMG-Signale bei bestimmten Bewegungen ermöglicht die Auswahl einer Funktion. Eine weitere Elektrode über der Okzipitalregion des Kopfes misst die Alpha-Aktivität im EEG. Dadurch kann das System durch das Schließen der Augen aus- und eingeschaltet werden. Die Elektroden sind in einem Stirnband befestigt, die Analyse der Signale erfolgt durch eine Signalprozessorkarte [BAR00].

Wheeler und Jorgensen (NASA Ames Research Center) entwickelten 2003 ein System, das EMG-Signale des Unterarms zur Steuerung eines Flugsimulators verwendet. Die Klassifikation der Muskelsignale erfolgt durch *Hidden Markov Modelle (HMM)*, die mit den Daten der unterschiedlichen Bewegungen trainiert werden. Durch ein Armband mit trockenen Elektroden konnte ein virtueller Joystick mit fünf Funktionen realisiert werden. Die Verwendung von Klebeelektroden verbesserte die Zuverlässigkeit der Messung, sodass neun Steuerkommandos unterschieden werden können [WHE03].

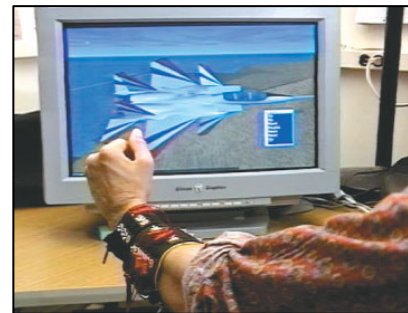


Abb. 2.29: Steuerung eines Flugsimulators mit EMG-Signalen

Jeong, Kim und Choi zeigten 2004 eine Methode für ein HCI-System, bei der das EMG der Gesichtsmuskulatur (*M. frontalis*) zur Steuerung des Mauszeigers abgenommen wird. Durch Zusammenpressen des Kiefers (links oder rechts) und daraus abgeleitete Muster werden sieben Kontrollkommandos unterschieden: Hinauf, Hinunter, Links, Rechts, Linksklick, Rechtsklick und Doppelklick. Auch hier sind die Elektroden zur möglichst einfachen Abnahme in ein Stirnband integriert. Die Signale werden durch einen mobilen 4-Kanal-Verstärker vorverarbeitet und über RS232-Schnittstelle an den PC gesendet. Die Klassifizierung der Aktivitätsmuster des EMGs zur Unterscheidung der beabsichtigten Kommandos erfolgt durch die Differenz der gemittelten Absolutwerte des Signals (*difference absolute mean value, DAMV*) und Neuronale Netze (*Fuzzy Min-Max Algorithmus*) [JEO04].

In einer weiteren Arbeit verwenden *Kim, Jeong und Son* ein EMG-Armband, mit dem sie vier Signale von Muskeln am Unterarm abnehmen. Die durchschnittliche Erkennungsrate für sechs unterschiedliche Kommandos liegt bei 97 Prozent [KIM04].

2.4.4.2 Musersatz durch EOG: EagleEyes

Ein verwendbares System, das den Mauszeiger mittels EOG-Signalen steuert ist *Eagle Eyes*, das am Boston College von *Gips und DiMattia* entwickelt wurde. Mausklicks werden durch eine einstellbare Verweildauer des Maus-cursors ausgelöst (*Dwelling*). Für *Eagle Eyes* wurde eine Reihe von Spiel- und Lernsoftware entwickelt, die jüngere Personen mit Behinderungen unterstützt und das Training mit dem System erleichtert [GIP96].

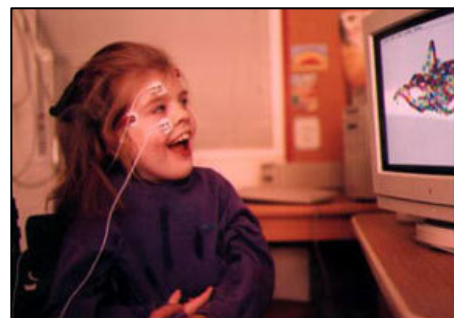


Abb. 2.30: EagleEyes: Verwendung des Computers mittels EOG

2.4.4.3 Kombination von EOG und Beschleunigungssensoren

Wiebe kombiniert in einem System zur Maussteuerung EOG-Elektroden und Beschleunigungssensoren in einer Brille. Das EOG wird am Brillenbügel und an der Kontaktstelle der Brille zur Nase abgenommen, die Elektroden müssen also nicht extra befestigt werden. Durch die Kombination der EOG- und Beschleunigungswerte wird die Blickrichtung errechnet, das System erlaubt die Auswahl der Bewegungsrichtung des Cursors und die Erzeugung von Mausklicks, siehe Abb. 2.31. [WIE03]

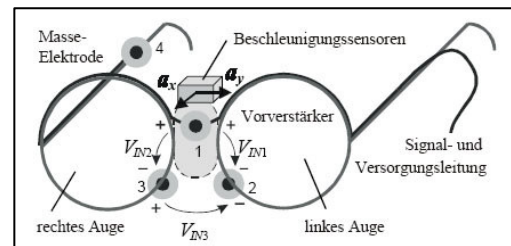


Abb. 2.31: EOG-Elektroden und Beschleunigungssensoren in einer Brille

2.4.4.4 Kombination von EMG und EOG: MindMouse

Ein kommerzielles Produkt zur Verwendung von EMG und EOG als Mausersatz ist die *MindMouse* der Firma *Cyberlink - Brain Actuated Technologies*. (<http://www.brainfingers.com>). Hier werden mittels Stirnband Signale der Gesichtsmuskulatur bzw. langsame Potenzialänderungen durch das EOG abgenommen, durch einen Verstärker vorverarbeitet und von der Software *Brainfingers* analysiert. Die Kontrollinformation wird auch hier aus den Frequenzkomponenten der Signale gewonnen, die Kontrollfähigkeit wird durch das Training mit Feedbackspielen gewonnen. Als zusätzliches Kontrollsignal dient das frontal abnehmbare EEG. Der Preis für das System beträgt etwa 2.000 Euro.

2.4.4.5 Resümee EMG- und EOG-Systeme

Ein allgemeines Problem von EMG- und EOG-Systemen ist die notwendige Fixierung von Elektroden zur Signalabnahme. Obwohl hier schon viele Ideen erfolgreich umgesetzt wurden, etwa aktive Elektroden ohne Kontaktgel oder elastische Bänder mit Elektroden zur einfacheren Befestigung, stellt die Applikation der Elektroden und der oft mangelnde Tragekomfort bei dauerhafter Verwendung einen Nachteil gegenüber berührungslosen Systemen dar. Die zielgerichtete Steuerung eines Cursors ist mit EMG-Systemen nur langsam bzw. in mehreren Schritten möglich, dafür erfolgt die Erkennung der Muskelpotenziale weitgehend zuverlässig sowie unabhängig von der Position des Computerbildschirms und die Verwendung ist meist leicht erlernbar. EMG-Signale eignen sich gut zur Steuerung einer Prothese oder Orthese, wenn die verbleibende willkürliche Muskelaktivität zur Wiederherstellung von Greif- und Bewegungsfunktionen ausreichend ist [BIT06]. Eine Kombination mit funktioneller Elektrostimulation ermöglicht die Unterstützung schwacher oder gelähmter Muskulatur [Y-AM99]. Kommunikationssysteme auf Basis des EOG sind vor allem dann eine sinnvolle Alternative, wenn sich andere Methoden zur Blickrichtungsbestimmung nicht eignen, weil z.B. eine Erfassung mit Kamerasystemen aufgrund von Spasmen nicht möglich ist.

2.4.5 Interaktion durch Infrarotdetektion von Bewegungsmarkern

Eine Reihe von Produkten erlaubt die Steuerung des Mauszeigers durch Kopfbewegungen. Viele Systeme basieren auf der Reflexion von ausgesendetem Infrarotlicht. Dabei sendet eine Basisstation, die am Computerbildschirm montiert wird, weitwinklig Infrarotlicht aus und empfängt die Reflexionen des Lichtes durch eine CCD-Kamera. Das Aufkleben eines Reflektorpunktes auf die Stirn oder auf eine Brille ist eine Voraussetzung für die Funktion. Diese Klebmarker haben etwa 5mm Durchmesser.

Ein Beispiel ist die **HeadMouse** bzw. die neuere **HeadMouse extreme** der Firma *Origin*. Die HeadMouse wurde als zusätzliches Peripheriegerät an den PC angeschlossen und konnte direkt die RS232-Kommandos einer Maus emulieren. Die *Headmouse extreme* ist dabei kleiner und leichter als das Vorgängermodell und kann über USB-Verbindung an den Computer angeschlossen werden.



Abb. 2.32: HeadMouse extreme, Origin

Die Funktionsweise der Geräte **TrackerPro** bzw. **Tracker2000** der Firma *Madentec* beruht auf demselben Prinzip (<http://www.madentec.com>). Die Zuverlässigkeit der Positionserkennung und die Abtastfrequenz sind bei diesen Systemen hoch, die Auslösung von Mausklicks erfolgt über Dwell-Schalter (Verweilzeit) bzw. durch externe Taster. Die maximale Entfernung zum Gerät beträgt ca. 150 cm. Eine uneingeschränkte Bewegungsfreiheit des Kopfes ist zur Verwendung notwendig. Kann nur der Kopf bewegt werden, ist eine Betätigung eines externen Schalters nur schwer möglich, da sich dadurch die Kopfposition ändert und eine genaue Selektion verhindert wird. Ein Saug-Blase-Schalter wäre in diesem Fall eine Alternative. Der Preis für *HeadMouse extreme* und *TrackerPro* liegt bei etwa 1.200 Euro.



Abb. 2.33: Tracker Pro, Madentec

2.4.6 Interaktion durch Kamerabildanalyse

Kameraunterstützte Interaktionssysteme und Kommunikationshilfen erleben durch die steigende Rechnerkapazität enorme Verbesserungen und haben viele andere Systeme in Punkto Komfort und Genauigkeit schon überholt. Die großen Vorteile sind die berührungsfreie Funktion und geringer Hardwareaufwand (je nach System können auch preisgünstige Webcams zum Einsatz kommen). Mit einem Standard-PC können mittlerweile alle Bildverarbeitungsmethoden, die für eine zuverlässige Gesichts- und Augendetektion nötig sind, in Echtzeit durchgeführt werden. Dennoch sind hochqualitative Systeme zur genauen Bestimmung der Blickrichtung auch heute noch relativ teuer, da die Anforderungen an Hardware und Auswertungsalgorithmen hoch sind. Das *Cogain*-Projekt fördert Entwicklung von Standards im Bereich der kamerabasierten Kommunikationssysteme und gibt einen Überblick auf existierende Produkte [COG05]. Prinzipiell lassen sich preisgünstige Systeme, die mithilfe einer Webcam das Gesicht und bestimmte Gesichtsm Merkmale erkennen von hochauflösenden Systemen zur exakten Bestimmung der Blickrichtung unterscheiden.

2.4.6.1 Detektion von Gesichtsposition und Gesichtsmerkmalen

Margrit Betke und Kollegen zeigen in einigen interessanten Forschungsarbeiten die Entwicklung von unterschiedlichen Systemen zur kamerabasierten Computerinteraktion mittels preisgünstiger Webcams. Das **CameraMouse** – System kann beliebige Körperstellen wie die Nase, die Lippe oder den Daumen zur Maus-Kontrolle verwenden. Links- Rechts- oder Doppelclicks werden durch Verweilzeit (*dwelling*) ermöglicht. Das verwendete Verfahren vergleicht die Intensitätswerte des gewählten Merkmals mit dem Live-Kamerabild (*Template Matching*).



Abb. 2.34: Bewegungsverfolgung durch das *CameraMouse*-System

Bei einer Auflösung von 320 x 200 Bildpunkten lässt sich auf einem Pentium IV Rechner eine Framerate von 30 Hz erreichen. Durch ein Zwei-Computer-Setup kann die Rechenleistung eines Systems für die Bildverarbeitung eingesetzt und der zweite Computer für die Benutzeraufgaben verwendet werden. Von zwölf Testpersonen mit Behinderungen, die an der Studie des Jahres 2002 teilnahmen, wollten neun Personen das System weiterverwenden [BET02].

Im **Eye-Keys** – System (2004) setzten *Magee, Scott* und *Betke* eine Standard-Webcam zur Blickrichtungsbestimmung ein. Sie konnten eine robuste Unterscheidung von linker und rechter Blickrichtung durchführen. Diese Information kann ein bestehendes System unterstützen oder zur Verwendung von Scanning-Software dienen [MAG04].

2005 zeigen *Betke* und *Chau* sowie *Grauman, Betke* und *Gips* die Lokalisation der Augen (*Eye-Tracking*) und die Detektion des Lidschlages zur Verwendung in HCI-Systemen (siehe Abb. 2.34). Das Verfahren beruht auf der Extraktion der Augenposition durch die Unterschiede zweier aufeinander folgender Bilder beim Lidschlag. Besteht das so ermittelte Augenpaar eine Plausibilitätsprüfung (Größe und Orientierung im Raum) wird ein Online-Template von der Augenregion erzeugt, das in weiterer Folge zur Lokalisation der Augen und zur Erkennung von Lidschlägen verwendet wird (Korrelationsverfahren). So lassen sich auf einem Pentium IV Computer Frameraten von etwa 30 Bildern pro Sekunde und Erkennungsraten von etwa 93 Prozent erreichen. Das Verfahren ist von der Umgebungsbeleuchtung weitgehend unabhängig und funktioniert auch bei Brillenträgern [BET05].

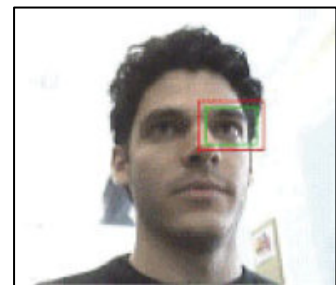


Abb. 2.35: Augendetektion in Echtzeit mit einer Webcam

Das **Nouse**-System von *Dmitry Gorodnichy* ist ein kamerabasiertes Maus-Interface, das eine freie zweidimensionale Bewegung des Bildschirmcursors durch eine Webcam erlaubt. Der Cursor wird dabei durch eine Bewegung des Kopfes bzw. der Nase gelenkt. Die verwendete Kameraauflösung beträgt nur 160 x 120 Bildpunkte, die Verarbeitung erfolgt mit etwa 30 Bildern pro Sekunde. Zur Vorverarbeitung für die Gesichtsdetektion wird eine Segmentation nach der Hautfarbe durchgeführt.

Dazu benötigt das System eine Kalibrierung (manuelle Auswahl der Nase). Danach dienen die konvexe Form der Nase und der Lidschlag als Merkmale für eine Gesichtserkennung. Der Lidschlag wird durch die Veränderungen zweiter Ordnung (die Geschwindigkeit einer Veränderung) in aufeinander folgenden Kamerabildern und eine Plausibilitätsprüfung erkannt. Ein doppelter Lidschlag wird zur Aktivierung / Deaktivierung des Systems eingesetzt, das Klicken erfolgt auch hier durch Verweilzeit (dwelling). Ein Malprogramm und einige Spiele erleichtern die Übung mit dem System [GOR02].



Abb. 2.36: Tracking der Nase für die Cursorsteuerung

2.4.6.2 Detektion der Blickrichtung (Gaze Tracking)

Der Einsatz von hochauflösenden videookulografischen Systemen erlaubt eine genaue Bestimmung der Blickrichtung. Durch entsprechende Software und eine Kalibrierung mit dem Computermonitor kann der betrachtete Punkt bis auf wenige Millimeter errechnet werden. Dadurch können ein Bildschirmkeyboard, ein Sprachsynthesizer oder eine Umgebungssteuerung durch Blicke gesteuert werden. Die Auswahl erfolgt meist durch längere Blicke auf die gewünschte Stelle (dwelling).

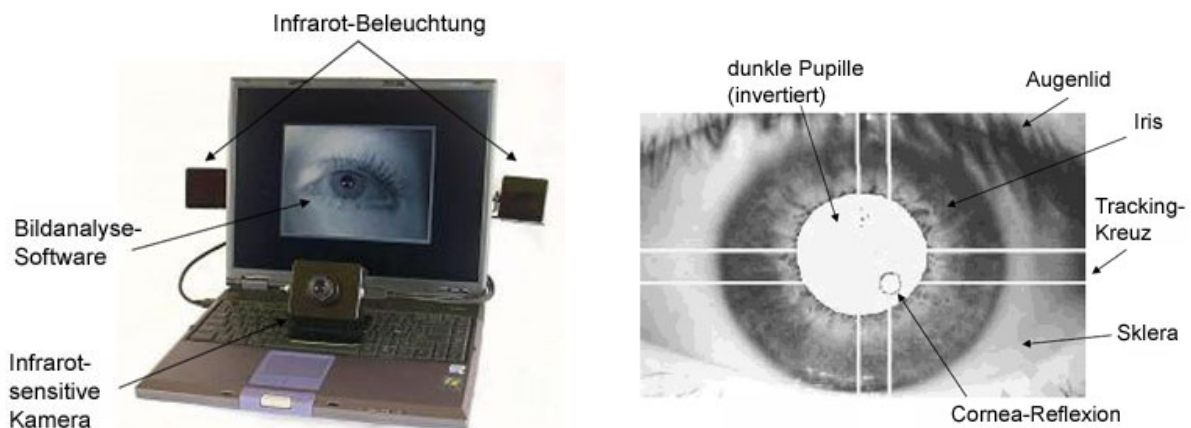


Abb. 2.37: Komponenten und Funktionsweise eines Gaze-Tracking-Systems

Ein großer Teil der auf dem Markt befindlichen *Gaze-Tracking* - Systeme verwendet als unterstützendes Merkmal zur Ermittlung der Augenstellung die Reflexion von Infrarotlicht durch die Cornea. Zusammen mit der Information über den Ort der Pupille können dadurch auch Rückschlüsse auf die Kopfstellung gezogen werden, sodass die Anfälligkeit gegenüber Kopfbewegungen etwas kleiner wird. Allen Systemen gemeinsam sind die Notwendigkeit einer Kalibrierungs-Prozedur und der eingeschränkte Bewegungsspielraum für den Kopf. Die Aktualisierungsfrequenz guter Systeme liegt bei 50 Hz bis 100 Hz.

Das wohl bekannteste Eye-Tracking-System ist **Eyegaze** der Firma *LC Technologies*. *Eyegaze* wird als Komplettsystem (Kamera, Bildschirm mit Schwenkarm, Computer) angeboten und kann als Zusatzgerät zum einem schon vorhandenen Rechner oder als eigenständiges Gerät zur Kommunikationshilfe und Umgebungssteuerung verwendet werden. Am Bildschirmkeyboard lässt sich mit *Eyegaze* eine Schreibgeschwindigkeit von bis zu 100 Buchstaben pro Minute erreichen. Der Preis beträgt 13.000 bis 18.000 Euro.



Abb. 2.38: Schreiben mit dem EyeGaze-System

Die folgende Tabelle gibt Aufschluss über einige wichtige Parameter gegenwärtiger Tracking Systeme [COG05]:

Modell	Hersteller	Genauigkeit (Grad)	zeitliche Auflösung (Hz)	Toleranz für Kopfbewegung (x/y/z, cm)	Entfernung (cm)
R6	<i>Applied Science Laboratories</i>	0,5	50-120	30x30x30	20-200
ERICA	<i>Eye Response Technologies</i>	0,5	60	?	?
Quick Glance	<i>Eye Tech Digital Systems</i>	1	0	10x10	?
EyeGaze	<i>LC Technologies</i>	0,45	60	3.8x3x3.8	66
iViewX	<i>Sensomotoric Instruments</i>	0,5	50	40x40x10	40-100
My Tobii	<i>Tobii Technology</i>	0,5	50	30x15x20	60

Tabelle 2.1: Gegenüberstellung der Merkmale von Gaze-Tracking Systemen

2.4.5.5 Resümee kamerabasierte Systeme

Kamerabasierte Systeme werden laufend verbessert und sind in ihrer einfachen Anwendbarkeit und Komfort kaum zu schlagen. Obwohl sich bereits eine Fülle von Produkten auf dem Markt befindet, ist es bis heute nicht einfach, ein gutes System zu einem vernünftigen Preis zu finden. Viele Ansätze sind noch in Entwicklung, kaum ein preisgünstiges Produkt bietet eine Maussteuerung mit Klick-Funktion ohne externe Schalter. Ein Nachteil der teureren Systeme zur Ermittlung der Blickrichtung ist die meist eingeschränkte Bewegungsfreiheit für den Kopf. Die Betrachtung dieser Umstände führt zur Überzeugung, dass die Entwicklung eines kostengünstigen Systems zur Cursorsteuerung mittels Webcam sinnvoll und machbar ist.

2.5 Brain Computer Interfaces

Brain Computer Interfaces (BCIs) ermöglichen die Verwendung eines Computers durch willkürlich herbeigeführte Veränderungen der Gehirnaktivität. Das EEG wird mittels Verstärkerhardware gemessen und durch Softwarealgorithmen ausgewertet, die ermittelten Parameter dienen zur Zeigersteuerung, zur schriftlichen Kommunikation oder zur Ansteuerung von Effektoren. BCIs haben in den letzten Jahren viel mediale Aufmerksamkeit erregt, da seit Anfang der 1990er Jahre die Ergebnisse kontinuierlich verbessert werden konnten. Die ersten erfolgreichen BCI-Experimente wurden jedoch bereits in den 70er Jahren durchgeführt [VID73].

Neben der *amyotrophen Lateralsklerose (ALS)* verursachen auch andere Krankheiten wie Polyneuropathie, Zerebrallähmung oder subkortikale Blutungen eine komplette Paralyse der Betroffenen (*locked-in*), wobei auch die Kontrollmöglichkeit über Lidschlag oder Augenbewegungen verloren geht, sodass BCI-Systeme zur einzigen Kommunikationsmöglichkeit für diese Menschen werden. Etwa 90-95 Prozent der Betroffenen entscheiden sich im Endstadium der Krankheit gegen eine künstliche Beatmung, wobei die Möglichkeit zur Verwendung eines BCI-Systems nur wenigen Personen bewusst ist. Trotz der enormen Beeinträchtigung beurteilt ein signifikanter Anteil der künstlich beatmeten ALS-Patienten die Lebensqualität mit "zufriedenstellend" oder "gut" [BIR00]. Die Aufrechterhaltung der Kommunikationsfähigkeit durch BCIs ist ein wichtiger Beitrag zur Lebensqualität dieser Menschen.

Derzeit kommen verschiedene Typen von BCI-Systemen zum Einsatz, die auf unterschiedlichen psychophysiologischen Konzepten aufbauen. Die wichtigsten Arten von Brain Computer Interfaces werden im Folgenden vorgestellt:

2.5.1 SCP - BCIs

Die Veränderung der langsamen Potenziale (*slow cortical potentials, SCP*; siehe 2.1.4.3) kann durch Training willkürlich beeinflusst werden. *Birbaumer* entwickelte 1990 ausgehend von Erfahrungen durch die klinische Praxis mit Bio- und Neurofeedback ein Verfahren, mithilfe dessen eine Negativierung bzw. Positivierung größerer Areale des Kortex erlernbar wurde. Durch die Auswertung der Signale konnte ein ALS-Patient Buchstaben wählen und einen Brief schreiben [BIR00]. Das System bekam den Namen *Thought Translation Device* und wird bis heute weiterentwickelt. *SCP-BCIs* erlauben meist eine Selektion aus einem Set von Möglichkeiten von Symbolen oder Buchstaben. Bevorzugte Ableitpunkte sind der Vertex (Position Cz) und der supplementäre Motorkortex (*supplementary motor area, SMA*; Positionen C3/C4). Das Training kann Wochen dauern und besteht aus tausenden Versuchen, eine Veränderung der langsamen Potenziale in die gewünschte Richtung herbeizuführen. Nicht alle Patienten und Patientinnen erlernen dies gleich gut. Die Rückmeldung einer erfolgreichen Aktion durch motivierende Sinneseindrücke, wie ein lachendes Gesicht oder einen harmonischen Klang, ist hilfreich [BIR00]. Die aktuelle Version des *Thought Translation Device* erleichtert das Schreiben durch Wortvorschläge (*Text Prediction*), wodurch eine Schreibgeschwindigkeit von etwa einem Wort pro Minute erreicht wird. Weiters werden durch ein psychophysiologisches Klassifizierungssystem die kognitiven Fähigkeiten von *locked-in* - Patienten/innen erhoben, wobei verschiedene Testprozeduren von der Auslösung auditorischer Hirnstammpotenziale (AEPs) bis zur kognitiven Reaktion auf Sprache (P300, N400-Komponenten) dazu dienen, die Eignung der Personen für das BCI-Training festzustellen [BIR00].

2.5.2 μ -BCIs

Der μ -Rhythmus tritt im Frequenzbereich von 10-12 Hz bei Inaktivität der motorischen Areale des Kortex auf. Er wird bei der Ausführung wie auch bei der Vorstellung von Bewegungen vermindert (*event-related desynchronisation, ERD*; siehe 2.1.4.2). μ -BCIs verwenden diese Synchronisations- bzw. Desynchronisationsphänomene zur Gewinnung von Steuerungssignalen. Die Ableitpunkte liegen wieder bei C3/C4 (*supplementary motor area, SMA*). Oft treten auch Veränderungen im Beta-Band bei 16-24 Hz auf.

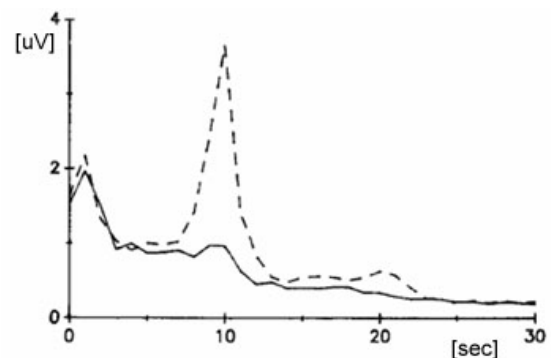


Abb. 2.39: Desynchronisation des μ -Rhythmus bei der Vorstellung einer Bewegung

Da die exakten Werte intersubjektiv verschieden sind, wird vor dem BCI-Training meist eine Offline-Analyse von EEG-Daten der Bewegungsvorstellungen durchgeführt, um die relevanten Frequenzbereiche und den besten Zeitpunkt für eine Unterscheidung herauszufinden. Dies erfolgt durch Mittelung der Frequenzspektren mehrerer Versuche. Auch die geeigneten Elektrodenpositionen können differieren, weshalb eine Verwendung mehrerer Ableitpositionen für die Offline-Analyse sinnvoll ist (*quantitative EEG, QEEG*). Im Online-Betrieb erhält die Person direktes Feedback über die gemessenen Werte. Häufig dient die vertikale oder horizontale Auslenkung eines Cursors zur Auswahl mehrerer Selektionen am rechten bzw. unteren Bildschirmrand (*Basket-Paradigma*).

Wolpaw und Kollegen konnten bereits 1994 eine zweidimensionale Cursorsteuerung zeigen ("*Wadsworth-BCI*"), die bis heute verbessert wird. Dabei werden die vertikale und horizontale Bewegung durch verschiedene Frequenzbereiche gesteuert. Abbildung 2.40 zeigt die Bestimmung der besten Parameter für die vertikale bzw. horizontale Auslenkung. Bei einer neueren Version des *Wadsworth-BCI* (2004) konnten die Personen nach dem Training eines von acht Feldern am Bildschirm in etwa 3 Sekunden auswählen [WOL91].

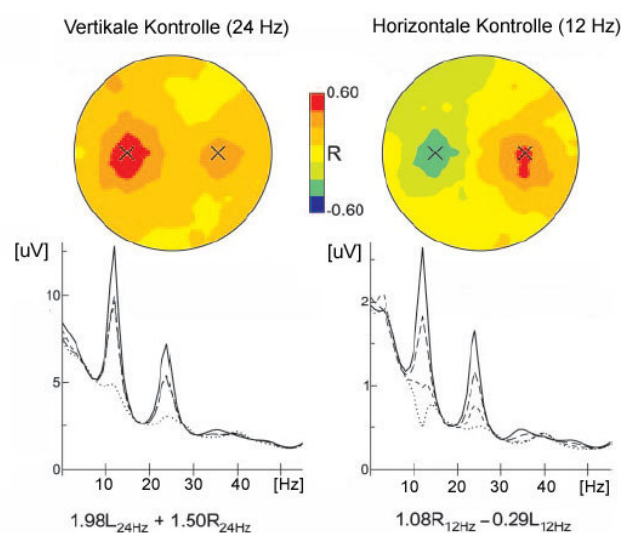


Abb. 2.40: Zweidimensionale Auswertung von ERD/ERS

In Graz entwickelten *Pfurtscheller, Guger, Müller-Putz* und Kollegen μ -BCIs, die zwischen drei Bewegungsvorstellungen unterscheiden konnten. Dazu wurden Klassifikatoren auf Basis neuronaler Netze mit den Frequenzdaten der verschiedenen Bewegungsvorstellungen trainiert. Im Online-Betrieb wurden die Daten mit den ermittelten Werten verglichen und so die wahrscheinlichste Bewegung gewählt.

Da sich dieses System auf die vorhandenen Signaländerungen einstellte, waren schon Erfolge nach kürzeren Trainingszeiten von wenigen Stunden möglich. Das "*Graz-BCI*" wurde unter anderem für die Steuerung einer Orthese zur Ausführung einer Greifbewegung der Hand bei einem 25-jährigen Patienten mit Querschnittslähmung verwendet [MUP05]. Die Abbildung zeigt die Unterstützung der Greiffunktion durch einen *Servo-Mechanismus*, der durch eine Bewegungsvorstellung aktiviert und deaktiviert wird. In weiteren Arbeiten wurde die Greiffunktion mit einer aktiven Prothese realisiert [GUG99] und ein implantiertes Muskelstimulationsgerät (FES) wurde über das BCI angesteuert.



Abb. 2.41: BCI-gesteuerte Orthese

2.5.3 VEP-BCIs

Hier ist das grundlegende Verfahren die Auswertung von evozierten Potenzialen des visuellen Kortex (*visual evoked potentials VEP*, siehe 2.1.4.1). Dabei werden Blink- oder Farbreize am Bildschirm präsentiert, wobei mehrere Auswahlmöglichkeiten bestehen. Durch Konzentration auf die gewünschte Auswahl treten visuell evozierte Potenziale auf (so genannte *steady state VEPs*, *SSVEP*), die für den hinterlegenden Reiz charakteristisch sind. Wird z.B. ein Schachbrettmuster mit einer Frequenz von 5 Hz invertiert, treten *SSVEPs* mit dieser Geschwindigkeit auf. So kann die Bildschirmstelle, auf welche sich die Person konzentriert, erkannt werden. Die Ableitung der VEPs erfolgt am Hinterkopf (parieto-okkipital, Positionen O1, O2 bzw. P3, P4), da dort die Lichtreize von der Retina in den visuellen Kortex münden. Das erste funktionstüchtige *BCI-System* basierte auf *SSVEPs* und wurde bereits 1977 von *Vidal* realisiert. Dabei konnten Versuchspersonen durch Konzentration auf einen von vier Stimuli am Bildschirm die Bewegung durch ein Labyrinth steuern. Das System erreichte eine Erkennungsrate von 90%, was viele moderne BCIs übertrifft [VID73]. Beim *Brain Response Interface* von *Sutter* (1992) wurden gleichzeitig 64 Symbole präsentiert, deren Betrachtung zu geringfügig unterschiedlichen *SSVEPs* führt. Diese Signalmuster werden zuvor in einer einstündigen Trainingssitzung vom Auswerteprogramm erlernt [SUT92]. Ein neueres System von *Gao* (2003) kann 48-90 Symbole unterscheiden und bei einer Erkennungsrate von 87% durchschnittlich alle 4 Sekunden eine Auswahl treffen. Das ergibt eine der höchsten bisher berichteten Transferraten für BCIs von etwa 60-90 Bits pro Minute [GAO03].

VEP-BCIs bieten die Vorteile einer schnellen Erlernbarkeit und Anwendbarkeit, eine gute Erkennungsrate und einen relativ hohen Datendurchsatz. Ein Nachteil ist, dass die Konzentration auf eine Bildschirmstelle willkürlicher Augenbewegungen bedarf, und falls diese noch vorhanden sind könnten etwa auch Gaze-Tracking Systeme eingesetzt werden. Allerdings ist diese Annahme kontrovers diskutiert, da die Möglichkeit der Beeinflussung von *SSVEPs* durch eine reine Fokussierung der Aufmerksamkeit auf die entsprechenden Stimuli (ohne die Blickrichtung zu ändern) schon von *Hillyard* und Kollegen gezeigt wurde [HIL02].

2.5.4 P300-BCIs

Bei dieser Form von BCI wird die P300-Komponente im EEG ausgewertet. Diese tritt bei der Unterscheidung von Reizen an der Schwelle zur Bewusstwerdung auf, etwa 300 Millisekunden nach der Präsentation des Reizes (siehe 2.1.4.1 EKP). Durch die Verwendung eines geeigneten Paradigmas können Buchstaben oder Symbole ausgewählt werden (P300-speller). Hierbei werden in einer Matrix eine Reihe von Buchstaben oder Symbolen präsentiert (zB. 5x5 oder 6x6). Die Zeilen bzw. Spalten der Matrix leuchten zufällig auf, die Person konzentriert sich auf das auszuwählende Symbol und zählt mit, wenn es erleuchtet wird.



Abb. 2.42: P300-BCI für das Buchstabieren

Nach einer Anzahl von Mittelungen des Signals kann der Unterschied in der P300-Komponente festgestellt und das Symbol damit ausgewählt werden. Ein Vorteil der P300-BCI ist die kurze Trainingszeit, da die P300-Komponente nicht erlernt werden muss. Eines der ersten P300-BCIs wurde von *Farewell* und *Donchin* (1988) realisiert. Eine 6x6 Matrix mit den 26 Buchstaben und weiteren Symbolen diente zur Eingabe von Wörtern, die dann über einen Sprachsynthesizer ausgegeben werden konnten. Zur Erkennung der P300 wurde eine schrittweise *Diskriminanz-Analyse* (SWDA) durchgeführt [FAR88]. Spätere Systeme hatten das Ziel, die Anzahl der Mittelungen zur Unterscheidung herabzusetzen und die Erkennungsrate durch andere Algorithmen wie *ICA* oder *Wavelet-Transformation* zu verbessern. *Bayliss* (2001) experimentierte mit einem System, das eine VR-Umgebung zur Stimuluspräsentation nutzt. Hier konnte die P300-Komponente bei der Unterscheidung weniger Symbole teilweise schon durch einen einzigen Versuch erkannt werden [BAY00].

2.5.5 Direct-Brain-Interfaces

Beim Direct Brain Interface (DBI) werden die Signale nicht von der Kopfoberfläche sondern direkt vom Kortex oder von tiefer liegenden Hirnregionen abgenommen. Dadurch lässt sich eine bessere Qualität der Ableitung und ein höheres Signal/Rausch-Verhältnis erreichen.

Die Ableitung erfolgt mittels Elektrodenarrays (*Grid-Elektroden*), oder mittels Glaselektroden, die mit einem neurotrophen Material ausgekleidet sind. Nervenzellen des Gehirns wachsen in das Röhrchen ein und produzieren gut ableitbare Signale an der innenliegenden Goldelektrode.

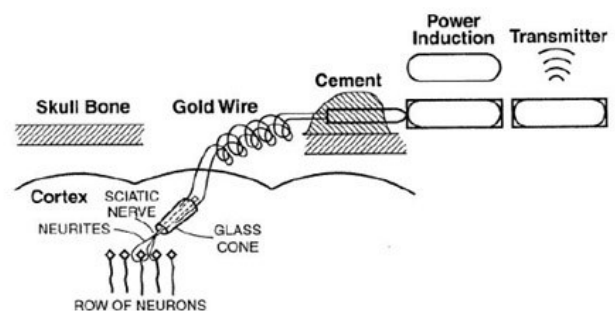


Abb. 2.43: Komponenten eines Direct-Brain-Interface

Die optimale Position der Elektroden wird zuvor über *fMRI*-Untersuchungen (*functional magnetic resonance imaging*) ermittelt. Die Verstärkung und Funkübertragung der Signale kann durch eine unter der Kopfhaut angebrachte Elektronik erfolgen, die durch eine Induktionsspule mit Energie versorgt wird. Die Nachteile implantierter Elektroden sind vor allem der nötige Eingriff, das Infektionsrisiko und bei längerem Einsatz eine mögliche Verschlechterung der Signalqualität. Erfolgreiche DBI-Implantationen wurden von *Huggins* (1997) sowie von *Kennedy* und *Bakay* (1998) durchgeführt [KEN87].

2.5.6 BCI2000

Das *BCI2000*-System wurde von Gerwin Schalk und der BCI-Gruppe des *Wadsworth Center* (Albany, N.Y.) als Forschungs- und Entwicklungsplattform für BCI-Systeme entwickelt. *BCI2000* bietet eine Aufteilung der Gesamtaufgabe in verschiedene Module für Datenerhebung (Sampling und Hardware-Treiber), Signalverarbeitung (Klassifizierung) und User-Applikation (Trainings- und BCI-Paradigma, Visualisierung, Steuerung von Effektoren). Die einzelnen Module kommunizieren über TCP/IP-Verbindungen, könnten also auch räumlich getrennt werden. Das Operator-Modul übernimmt die Konfiguration und Synchronisation der anderen Module [SCH].

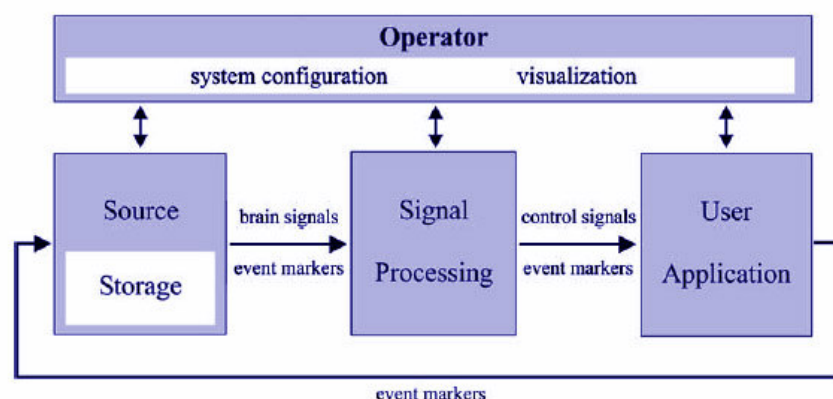


Abb. 2.44: Struktur und Komponenten der BCI2000-Applikation

Auf diese Weise kann eine Vielzahl von BCI-Experimenten unter Einsatz eines Rahmenwerks durchgeführt werden, was Vergleiche erleichtert und viel Programmieraufwand einspart. *BCI2000* kann als freie Software für den wissenschaftlichen Gebrauch vom *Wadsworth Center* bezogen werden. Derzeit wird die Software weltweit von über 90 Personen und Instituten verwendet. Folgende Anwendungen wurden mit *BCI2000* bereits realisiert [SCH]:

- 1D/2D Cursorbewegung mittels ERD/ERS (Abb. A)
- 4-fach-Auswahl zur Buchstabenselektion durch ERD/ERS (Abb. B)
- Cursorkontrolle durch SCP (Abb. C)
- P300 Matrix Buchstabenselektion (Abb. D)
- Auswertung von ECoG-Signalen

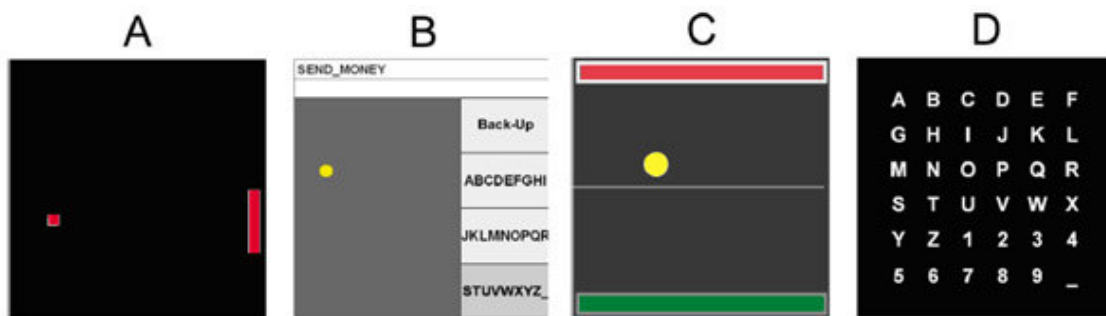


Abb. 2.45: Beispiele für mögliche BCI-Applikationen in BCI2000

Derzeit unterstützt BCI2000 folgende Hardwaresysteme:

EEG-Systeme: *USBamp, MOBIlab (g.tec), Tucker-Davis Pentusa, Biosemi, BioRadio 150 (Cleveland Medical Devices), Refa System (TMSi), Neuroscan, Brain Vision (Brain Products), Micromed EEG, ModularEEG (OpenEEG)*

A/D-Converter Boards der Firmen *Data Translations, National Instruments, Measurement Computing.*

BCI2000 wurde in *Borland C++* unter Windows erstellt, der gesamte Source-Code und die Dokumentation kann vom Wadsworth Center für wissenschaftliche Zwecke kostenfrei bezogen werden. Im Rahmen dieser Arbeit wurde ein Treiber für die *ModularEEG* - Hardware in Form eines *Signal-Source* - Moduls entwickelt und getestet (siehe Abb 2.46). Dieses Modul ist nun Teil der offiziellen Distribution von *BCI2000*.

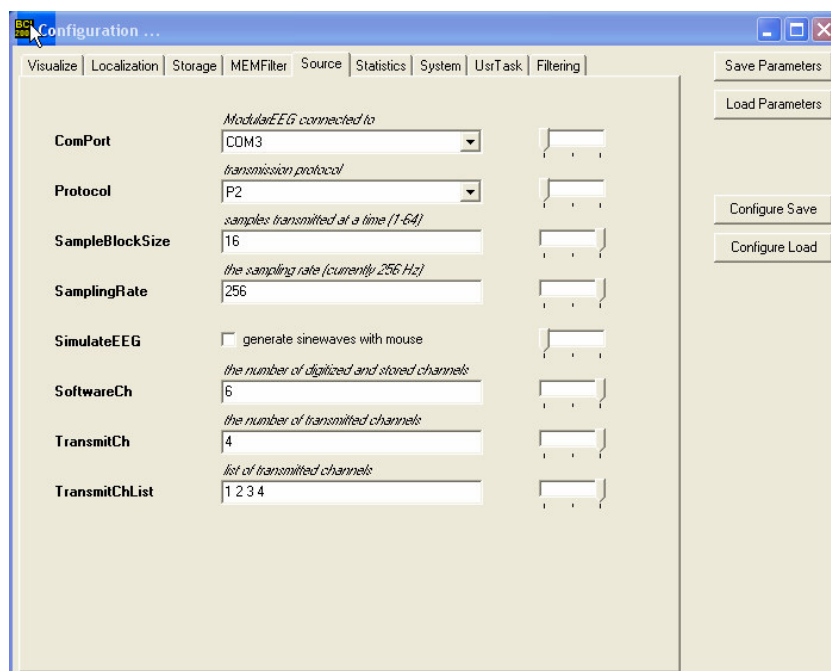


Abb. 2.46: Einstellungsmöglichkeiten der *ModularEEG*-Signalquelle für BCI2000

3. Hardware zur Messung bioelektrischer Signale

Das folgende Kapitel beschäftigt sich mit den Geräten zur Verstärkung und Messung von bioelektrischen Signalen. Nach einer Erläuterung grundlegender technischer Aspekte folgt ein Überblick auf einige kommerzielle Verstärkermodelle. Als kostengünstige Alternative wird die *OpenEEG*-Hardware vorgestellt und genauer beschrieben.

3.1 Prinzipielles zum Messvorgang

Die Messung (Ableitung) eines bioelektrischen Signals vom menschlichen Körper erfolgt durch geeignete Messfühler, die Elektroden. Gemessen wird stets eine Potentialdifferenz zwischen zwei Punkten. Die Elektrode nimmt ein elektrisches Potential auf und leitet es weiter zum Messverstärker. Ein geeigneter Filter entfernt unerwünschte Frequenzbereiche, die Störungen verursachen würden. Dann folgt die Analog/Digital-Wandlung (Digitalisierung, *sampling*). Nach der Digitalisierung steht der Messwert zur Übertragung und Weiterverarbeitung zur Verfügung.

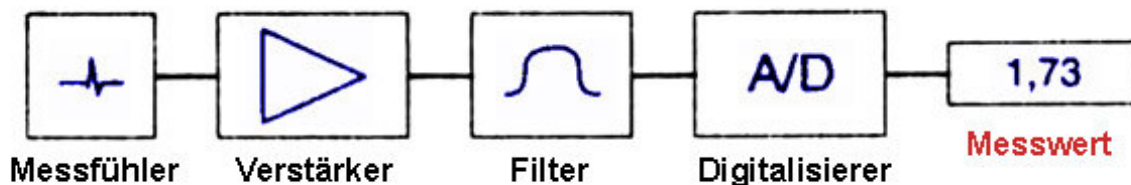


Abb. 3.1: Typische Elemente einer Messkette

Für die richtige Konfiguration dieser Messkette ist das Wissen über die Eigenschaften der zu messenden Signale und die Auswirkungen der verwendeten Komponenten eine wichtige Voraussetzung, da das Ergebnis bei ungeeigneter Wahl der Mittel verfälscht wird. Die bewusste Wahl der Ableitungsart (monopolar oder bipolar, siehe 3.1.1), die Vermeidung von Störungen und Artefakten sowie die richtige Dimensionierung von Filtern und Digitalisiereinheit sind Bedingungen für die Erhebung aussagekräftiger Messwerte.

3.1.1 Monopolare und bipolare Ableitung

Man unterscheidet je nach Positionierung der Elektroden die monopolare von der bipolaren Ableitung. Bei bipolaren Ableitungen werden zwei Elektroden an elektrisch aktiven Punkten der zu messenden Region (z.B. bei einer EEG-Messung am Kopf) angebracht, bei der monopolarer Ableitung befindet sich eine Elektrode an einem elektrisch inaktiven Bereich, etwa am Ohrläppchen (indifferente Elektrode). Dies bewirkt substantielle Unterschiede im Resultat der Messung: Bei einer bipolaren Ableitung löschen sich gleichphasige Signalanteile aus, während sich gegenphasige Anteile verstärken. Zwei Wellen mit derselben Frequenz, Amplitude und Phase würden sich bei einer bipolaren Ableitung gänzlich auslöschen. Die bipolare Ableitung zeigt also vor allem Fluktuationen durch Verschiebungen in den Phasenanteilen der Signale.

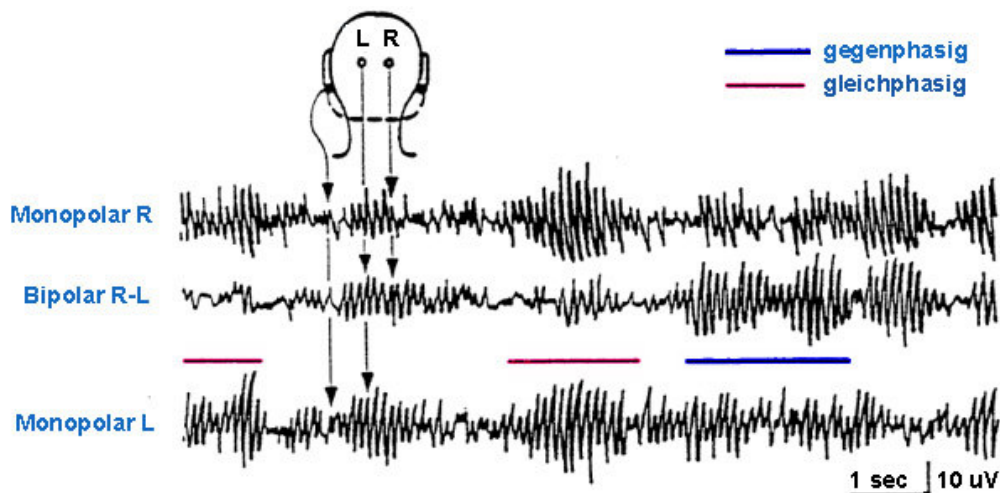


Abb. 3.2: Unterschiede zwischen monopolarer und bipolarer Ableitung [FEH89]

Bei der monopolarer Ableitung wird die Aktivität gegenüber einem kaum veränderlichen Potenzial gemessen, wodurch Auslöschungs- oder Verstärkungseffekte vermieden werden. Allerdings werden bei größerer Entfernung zur indifferenten Elektrode auch Signalquellen in die Messung einbezogen, die zwischen den beiden Elektroden liegen, was oft unerwünschte Artefakte bewirkt. Werden mehrere Positionen monopolar gegen eine gemeinsame Referenz abgeleitet, besteht die Möglichkeit, bipolare Ableitungen rein rechnerisch aus den monopolarer Messergebnissen zu gewinnen [FEH89].

3.1.2 Der Elektrode-Haut-Kontakt

An der Elektrode wird das elektrische Potenzial, das durch Ionenbewegungen im Gewebe entsteht, aufgenommen und durch Elektronenbewegung weitergeleitet. Bei der Oberflächenableitung ist die Haut die Kontaktstelle zwischen Elektrode und Körper. Der Widerstand an dieser Kontaktstelle setzt sich aus dem Widerstand der Haut, der Kapazität der Haut und dem Widerstand des Elektrode-Haut-Kontaktes zusammen. Da der Widerstand kapazitive Anteile hat, spricht man von einer Impedanz. Die Impedanz sollte für eine hochwertige Messung möglichst niedrig gehalten werden. Durch geeignete elektrolytische Gels und Hautreinigung mittels Alkohol kann der Elektrode-Haut-Kontakt bedeutend verbessert werden.

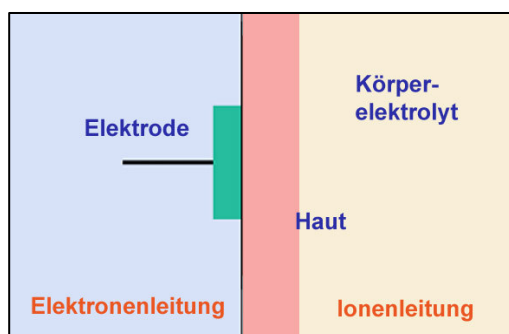
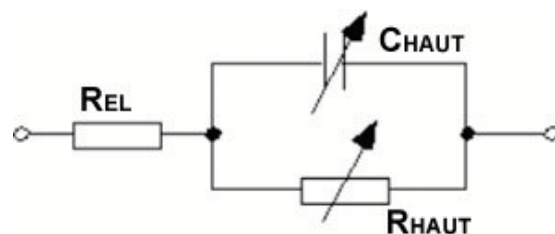


Abb. 3.3 (a): Der Elektrode-Haut-Kontakt



(b): elektrisches Ersatzschaltbild

Zur Ermittlung der Impedanz wird der Widerstand bei einem Wechselstrom geeigneter Frequenz (z.B. 14 Hz) gemessen. Die Impedanz sollte für eine hochwertige Messung etwa 10 kOhm nicht übersteigen.

3.1.3 Noise und Artefakte

Noise (Rauschen) und Artefakte sind unerwünschte Störungen der Messung. Diese können ihren Ursprung in elektrostatischen oder elektromagnetischen Feldern haben, die Spannungen in Kabel oder Messvorrichtung induzieren, aber auch in der Einstreuung anderer Biosignale. Hohe Widerstände durch schlechten Elektrode-Haut-Kontakt führen ebenso zu Artefakten wie ein Verrutschen von Elektroden oder Bewegungen der Elektrodenkabel. Im Folgenden werden die wichtigsten Ursachen von Artefakten und mögliche Strategien zu ihrer Vermeidung kurz besprochen.

3.1.3.1 Einkopplung von Störungen, Gleichtaktanteil und CMR

Elektrostatische Felder oder veränderliche Magnetfelder verursachen durch kapazitive oder induktive Kopplung Störströme in den Kabeln und Leiterschleifen der Ableitvorrichtung, sowie im menschlichen Körper. Besonders starke Einkopplung verursacht die Präsenz von Wechselstromverbindungen der Netzversorgung in der Nähe der Verstärkereinheit bzw. bei einer Verwendung langer, ungeschirmter Elektrodenkabel. (*Netzbrumm, 50Hz-Noise*).

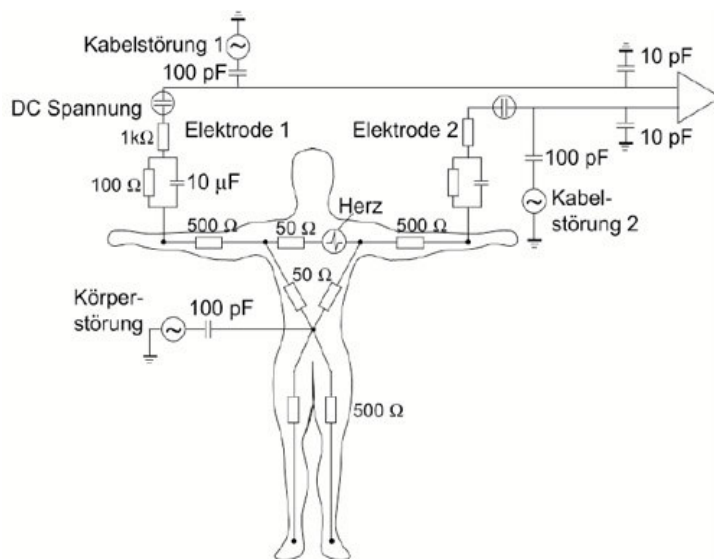


Abb. 3.4 zeigt die kapazitive Einkopplung der Netzspannung in den menschlichen Körper bei einer EKG-Ableitung. Da das Störsignal an allen Teilen des Körpers bzw. an den Verstärkereingängen gleichzeitig (phasengleich) eintrifft, spricht man vom *Gleichtaktanteil (common mode portion)*. Die *common mode rejection-ratio (CMRR)* gibt an, wie stark die gleichphasigen Anteile im Verhältnis zum Nutzsignal unterdrückt werden können [LIS05].

Abb. 3.4: Kapazitive Einkopplung von 50Hz-Noise

$$CMRR = A_D \left(\frac{U_{CM}}{U_A} \right) \quad (\text{Gl. 3.1})$$

A_D ... differentieller Verstärkungsfaktor
 U_{CM} ... am Verstärkereingang anliegende Gleichtaktspannung
 U_A ... Ausgangsspannung

Die *Gleichtaktunterdrückung (common mode rejection, CMR)* ist die logarithmische Schreibweise dieser Kennzahl: $CMR = 20 \cdot \log(CMRR)$. Sie wird in dB gemessen und ist eine wichtige Kenngröße von Differenzverstärkern. Moderne Instrumentenverstärker erreichen Werte von 80 dB bis 120 dB.

Da die Stärke der Einkopplung (vor allem bei Störungen durch Netzbrumm) oft ein Vielfaches des Nutzsignals beträgt, kann das Auftreten exzessiver 50Hz-Wellen den Messbereich des Verstärkers übersteuern und so zum Problem werden.

Folgende Schritte dienen zu einer Verminderung von Einkopplungen:

- eine Erhöhung der Entfernung zu Stromkabeln und elektrischen Geräten
- die Verwendung einer Abschirmungsfläche (Faraday'scher Käfig)
- die Verminderung der Elektrodenimpedanz
- die Vermeidung von Leiterschleifen (*ground loops*)
- ein 50 Hz - Band-Sperr - Filter (*notch filter*) in der Verstärkereinheit
- die Schirmung der Elektrodenkabel bzw. die Verwendung von aktiven Schirm-Treibern (*guarding*).
- Verwendung einer Bezugspotenzialsteuerung (*driven right leg*, siehe 3.3.1.1)

Die Schirmung einer Signalleitung durch einen umgebenden Leiter schützt vor allem gegen Einkopplungen durch elektrische Felder effektiv. Eine Seite der Schirmung wird dabei mit dem Nullpotenzial verbunden. Der Schutz gegenüber Einstreuungen durch magnetische Felder ist schwieriger, da diese die Schirmung durchdringen. Bei der aktiven Schirmung (*guarding*) wird der Gleichtaktanteil über einen zusätzlichen Operationsverstärker, der als Impedanzwandler dient, zur Speisung der Kabelschirmung verwendet. Dadurch wird die Störung durch magnetische Felder vermindert und eine höhere CMR erzielt.

3.1.3.2 Artefakte durch andere Biosignale

Vor allem bei der Messung des EEG ist die Einstreuung stärkerer Biopotenziale anderer Herkunft ein Problem: EOG, EKG und EMG überlagern das kleinere Signal. Haben die Störsignale andere Frequenzkomponenten, können geeignete Filter verwendet werden. Weitere Strategien sind die Wahl anderer Elektrodenpositionen (bipolar, kleinerer Abstand), die Kennzeichnung und nachträgliche Entfernung der Artefakte oder die Verwendung von Softwarealgorithmen zur Artefaktentfernung (z.B. *independent component analysis, ICA*)

3.1.3.3 Artefakte durch Bewegungen von Elektroden oder Kabeln

Diese führen zu einer Schwankung in der Qualität des elektrischen Kontaktes und somit zu Spannungsschwankungen am Verstärkereingang. Weiters kann eine Bewegung der Kabel Störströme induzieren. Eine gute Montage der Elektroden (Kontaktgel) und eine Zugentlastung der Kabel sind sinnvoll.

3.1.3.4 Hohe oder unbalancierte Elektrodenimpedanzen

Hohe bzw. stark unterschiedliche Elektrodenimpedanzen verschlechtern die Gleichtaktunterdrückung des Differenzverstärkers und führen zu starken Artefakten durch Gleichtaktanteile. Ebenfalls problematisch ist eine veränderliche Impedanz (etwa durch Eintrocknung von Gel oder Schweiß). Eine Verminderung der Impedanz bei Oberflächenableitungen kann durch Reinigung mit Alkohol oder durch eine Entfernung alter Hautschichten erreicht werden.

3.1.3.5 Elektrodenpolarisation

Mit der Zeit können sich durch Ladungstrennung an den Elektroden Ionen ansammeln und den Ladungstransport behindern. Dabei entsteht ein neues Potenzial durch die Polarisierung der Elektroden, das viel größer werden kann als die zu messenden Biosignale, wodurch die Messverstärker übersteuert werden könnten. Durch die Verwendung nicht polarisierbarer *Silber/Silberchlorid-Elektroden (Ag/AgCl)* und den Einsatz von Hochpass-Filtern bei der Verstärkung kann Polarisierungseffekten weitgehend entgegengewirkt werden. Die Chlorid-Schicht der Ag/AgCl-Elektroden reagiert mit dem Elektrolyten, durch die chemische Reaktion werden Elektronen ausgelöst, die frei wandern, es kommt zu keiner Ansammlung von Ladungsträgern an den Elektroden.

3.1.3.6 Internes Rauschen der elektronischen Bauteile

Thermisches Rauschen und Ungenauigkeiten in den Bauteilen sind nie gänzlich zu verhindern. Bei Bau der Verstärkerhardware ist auf geringe Toleranzwerte für Kondensatoren und Widerstände zu achten. In hochqualitativen Biosignalverstärkern sollte die Fehlerquelle durch internes Rauschen weniger als ein μV betragen.

3.1.4 Der Instrumentenverstärker

Die umgebenden elektromagnetischen Wechselfelder induzieren Spannungen in den menschlichen Körper, welche die zu messenden bioelektrischen Signale um ein Vielfaches übersteigen (*Gleichtaktanteil*, siehe 3.1.3.1). Um dennoch eine Verstärkung der sehr kleinen Gegentaktsignale bei Unterdrückung des Gleichtaktanteils zu erreichen, werden meist Instrumentenverstärker eingesetzt.

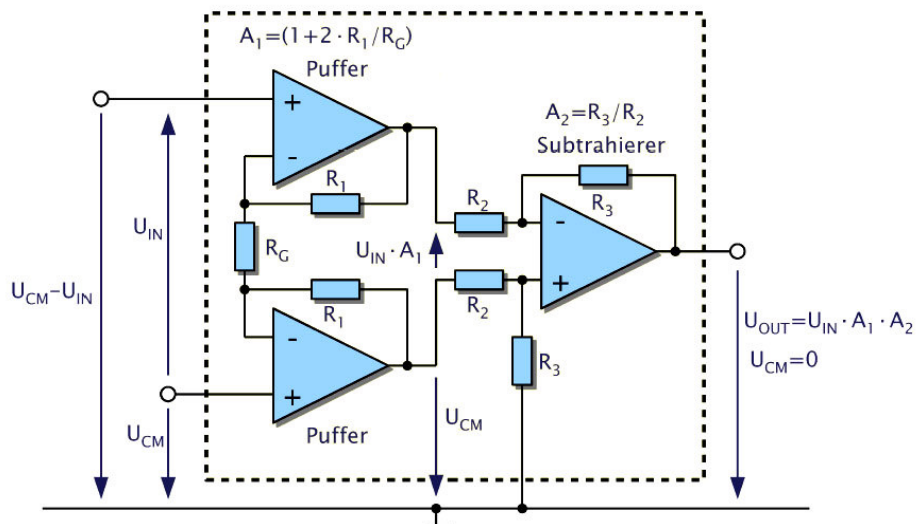


Abb. 3.5 Aufbau eines Differenzverstärkers

Die Abbildung zeigt den prinzipiellen Aufbau eines Instrumentenverstärkers aus drei Operationsverstärkern. Operationsverstärker sind Differenzverstärker, die den Potenzialunterschied an den Eingängen mit sehr hohem Verstärkungsgrad (gängige Werte liegen bei 250000) an den Ausgang weiterleiten. Übersteigt das Potenzial am invertierenden Eingang (-) das Potenzial des nicht-invertierenden Einganges (+) wird der Ausgang negativ. Durch Rückkopplung des Ausganges mit Spannungsteilern an

den invertierenden Eingang können beliebige (kleinere) Verstärkungsfaktoren erreicht werden. In der Abbildung wird deutlich, dass das zu verstärkende Signal direkt mit den nicht-invertierenden Eingängen der ersten beiden Operationsverstärker verbunden ist. Diese dienen als Impedanzwandler (Puffer): die Eingangsimpedanzen sind dadurch gleich groß und erreichen sehr hohe Werte, was wichtig ist, um das Signal nicht durch die Weiterverarbeitung zu belasten und dadurch zu verfälschen. Durch eine besondere Verschachtelung der Rückkopplungswiderstände wird erreicht, dass bereits die Eingangspuffer Gegentaktsignale um einen Faktor größer eins verstärken, aber Gleichtaktsignale unverstärkt bleiben. Der nachfolgende Differenzverstärker eliminiert den Gleichtaktanteil nahezu vollständig, wobei der Gegentaktanteil weiter verstärkt wird [LIS05]. Dadurch lässt sich eine CMR von 80 - 120 dB erreichen (siehe 3.1.3.1). Weitere Kenngrößen von Instrumentenverstärkern sind die maximale Offset-Spannung am Eingang (U_{OE}), die maximale Ruhestromaufnahme (*quiescent current*, I_{QC}), der Eingang Bias-Strom (I_{BIAS}) sowie der Versorgungsspannungsbereich und das eingangsbezogene Spannungsrauschen. Für eine genauere Erklärung dieser Parameter sei auf weiterführende Literatur bzw. auf die Datenblätter von konkreten Modellen verwiesen [KUP05].

3.1.5 Filterung und Digitalisierung

Bei der Digitalisierung wird das analoge Signal abgetastet und diskretisiert. Auflösung und Abtastfrequenz (*Samplerate*) definieren die Qualität der Digitalisierung, wobei im Vergleich zum analogen Ausgangssignal immer Information verloren geht. Die Auflösung (*Quantisierung*) bestimmt die unterscheidbaren Werte vom Minimum bis zum Maximum des Wertebereiches (*Quantisierungsstufen*). Die Auflösung wird in Bit gemessen, beispielsweise können bei einer Auflösung von 10 Bit $2^{10} = 1024$ Werte unterschieden werden, bei 16 Bit sind es 65536. Bei der Digitalisierung wird für einen analogen Wert immer der bestmögliche ganzzahlige Wert ermittelt. Dadurch entsteht ein *Quantisierungsfehler* in der Größe von $\frac{1}{2} \text{ LSB}$ (*least significant bit*, der kleinste unterscheidbare Wert). Anders ausgedrückt wird durch die Abtastung ein zufälliges Störsignal zur Messung hinzugefügt [SMI01].

Die Samplerate gibt an, wie oft eine Diskretisierung vorgenommen wird. Sie liegt bei Biosignableitungen meist im Bereich von 100 - 2.000 Hz. Hier ist entscheidend, dass nur Signale fehlerfrei durch die Abtastung erfasst werden können, die keine Frequenzkomponenten haben, welche größer sind als die Hälfte der Samplerate. (*Nyquist Frequenz*, *Shannon Sampling-Theorem*). Dies ist deshalb der Fall, weil nur Frequenzkomponenten, die langsamer als die halbe Abtastfrequenz sind, durch die digitalisierten Werte eindeutig repräsentiert werden können. Wird z.B. eine Sinuswelle mit einer Frequenz abgetastet, die nur geringfügig größer ist als ihre eigene Frequenz, entsteht durch die Abtastung der Eindruck, es handle sich um eine Welle mit viel niedrigerer Frequenz (*Aliasing*). Alle Frequenzkomponenten, die größer sind als die Hälfte der Samplingfrequenz, erscheinen durch das Aliasing in der Digitalisierung als Störfrequenzen, und verfälschen die tatsächlichen Signalkomponenten in diesen Frequenzbereichen [SMI01].

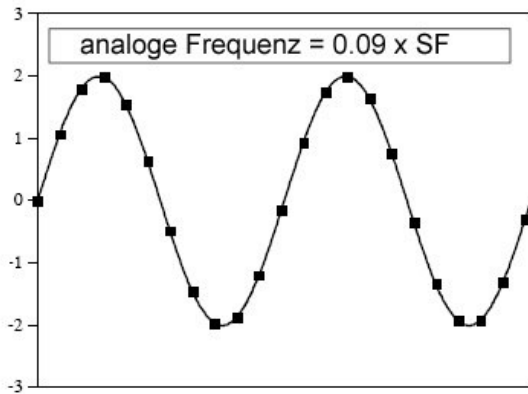


Abb. 3.6: Ausreichende Samplingrate

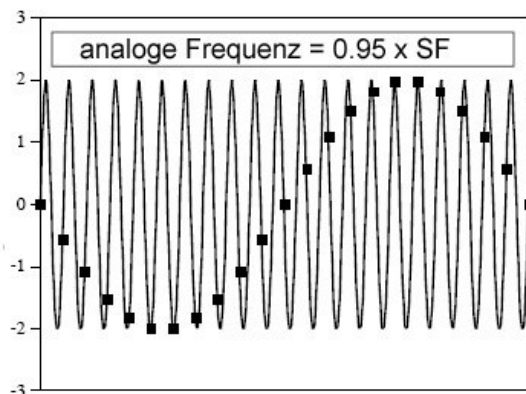


Abb. 3.7: Inadequate Samplingrate - Aliasing

Da durch die Antennenwirkung der Elektrodenkabel und die kapazitive und induktive Einkopplung von Störungen hochfrequente Signalanteile präsent sind, ist es wichtig, diese vor der Digitalisierung aus dem Signal zu entfernen, um Verfälschungen durch Aliasing zu verhindern. Dies geschieht durch geeignete Tiefpass-Filter (*Anti-Aliasing-Filter*) zwischen bzw. in den Verstärkerstufen. Ziel ist, die Frequenzen über der halben Samplefrequenz weitestgehend aus dem Signal zu entfernen. Hierfür eignet sich die modifizierte *Sallen-Key*-Schaltung für aktive Filter [SMI01]:

Die Abbildung zeigt einen Filter 2ter Ordnung (*2-Pol*), bestehend aus Widerständen, Kondensatoren und einem Operationsverstärker. f_c gibt die *Cut-Off* - Frequenz des Filters in Hz an. Die Konstanten k_1 und k_2 bestimmen den Typ des Filters (*Bessel*, *Butterworth* oder *Chebyshev*). Die Filtertypen unterscheiden sich in der Schärfe beim Abschneiden höherer Frequenzen (Roll-Off) und in der Impulsantwort im Zeitbereich.

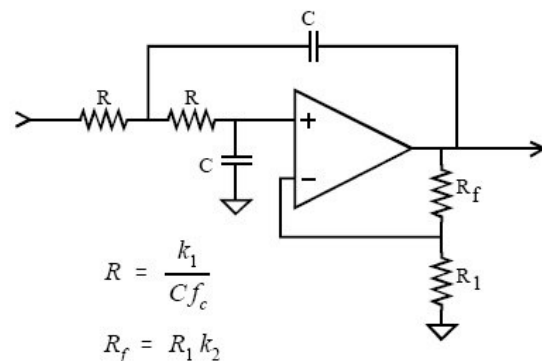


Abb. 3.8: Tiefpassfilter (*Sallen-Key*-Schaltung)

Als *Anti-Aliasing-Filter* werden häufig *Bessel*-Filter verwendet, die zwar im Frequenzbereich nicht so genau abschneiden wie etwa *Butterworth*- oder *Chebyshev*-Filter, dafür aber im Zeitbereich die wenigsten Verfälschungen durch Überschießen oder Nachschwingen verursachen.

3.1.6 Die Eichung der Messkette

Durch die Verwendung mehrerer Verstärkerstufen und Filter ist die theoretische Auswirkung auf die Signalquelle praktisch zu überprüfen. Eine geeignete Methode ist die Eichung des Systems durch ein Kalibrier-Signal bekannter Stärke. Hierbei wird ein Rechtecksignal mit regelbarer Frequenz und Amplitude durch einen Signalgenerator erzeugt und die Auswirkung von Verstärkung und Filterung kann durch die Impulsantwort überprüft werden. Eine genaue Eichung ist Voraussetzung für den Erhalt richtiger und überprüfbarer Messwerte.

3.1.7 Patientenschutzmaßnahmen und Normen

Vor allem bei medizintechnischen Geräten ist eine Sicherheit von Patientinnen und Patienten oberstes Gebot, und so stellen auch die Normen und Richtlinien für die Schutzmaßnahmen in Biosignalverstärkern hohe Anforderungen an Design und Bauteile. Die Zulassung zum medizinisch geprüften Gerät ist teuer und langwierig, die Checklisten umfassen hunderte Seiten. Die relevanten Regelwerke sind:

- die internationale Basishnorm **IEC 60601-1** für Medizinische Elektrische Systeme
- die DIN EN ISO 14971 Risikomanagement für Medizinprodukte
- die Richtlinie **89/336/EWG** "Elektromagnetische Verträglichkeit"
- die deutsche **VDE 0750** bzw. **VDE 0752**

Es werden verschiedene Schutzklassen für die Geräte und Anforderungen für die Konstruktion der Hardware, die Spezifikationen der Bauteile, die Programmierung und Funktionalität der Software und verschiedene Arten des Einsatzes definiert. Dazu gehören auch Schutzmaßnahmen für mögliche Fehlerquellen, Bauteile-Tests mit starken Überspannungen und verschiedene Verwendungsszenarien. Auf diese Art soll die größtmögliche Sicherheit und eine Kompatibilität mit anderen medizintechnischen Geräten gewährleistet werden [SIT98].

3.1.7.1 Die wichtigsten Anforderungen an die Hardware:

- Erfüllung der Bestimmungen für ausreichenden Schutz vor Elektroschocks auch und gerade im Fehlerfall und beim Auftreten von Spannungsspitzen
- Isolation und Patientensicherheit bei Überspannungen
- Erfüllung der Richtlinien und Normwerte der elektromagnetischen Abstrahlung und Sicherheit im Falle starker elektromagnetischer Einstrahlung [CON]

Auszug aus den Hardwareanforderungen der IEC 60601-1-2 :	
Elektrostatistische Entladung:	6KV (Kontakt) 8 kV (Luft)
Elektromagnetische Unempfindlichkeit:	3 V/m bei 80 MHz - 2 GHz
Spannungsspitzen:	2KV (Stromversorgung) 1 kV (Ein-Ausgänge)
Magnetfelder:	3 A/m bei 50/60 Hz

Tabelle 3.1: Beispiele für Hardwareanforderungen der IEC-Norm 60601-1-2

3.1.7.2 Kennzeichnung der zertifizierten Medizintechnikgeräte:



Symbol for
EN60601-1 type BF



Symbol for
EN60601-1 class II

Abb. 3.9: Symbole für die Kennzeichnung zertifizierter Medizintechnikgeräte

3.2 Kommerzielle Biosignalverstärker - ein Überblick

Viele der derzeit erhältlichen Biosignalverstärker haben als Zielgruppe nicht den professionellen Markt für Kliniken und Medizintechnik, sondern es sollen vermehrt private Anwender angesprochen werden. Durch die zunehmende Popularität von Biofeedback und Entspannungstechniken konnten sich mehr und mehr Firmen in diesem Marktsegment etablieren. Dies führte zu einer größeren Auswahl an Geräten und günstigeren Preisen für die Hardware. Auch die Anwendung wurde durch die Miniaturisierung der Geräte und die Möglichkeit kabelloser Datenübertragung komfortabler.

Im unteren Preissegment findet man heute einige Geräte, die für viele Anwendungsbereiche eine ausreichend genaue Messung ermöglichen. Diese low-cost-Produkte unterscheiden sich vor allem in der Auflösung, Abtastrate und Anzahl der Kanäle von teureren Biosignalverstärkern und werden als Heim-Trainingsgeräte vertrieben. Standalone-Biofeedback-Geräte ermöglichen beispielsweise das Beckenbodentraining, EDA-Feedback oder Atemfeedback. Hier sind etwa die Systeme der Firma *Insight Instruments* zu nennen, für die allerdings keine genauen Spezifikationen verfügbar sind: das *Myself* - Beckenbodentrainingsgerät (120 €) und das *Respirate* Atemfeedbacksystem (350 €). Das HRV-Trainingsmodul *Stressball* (720 €) ist ein PC-gebundenes Trainingssystem für EKG/HRV-Feedback [INS06]. Im Folgenden werden die Spezifikationen einiger Biosignalverstärker genauer vorgestellt:

3.2.1 Myotrace 200

Noraxon Myotrace200	
Kanäle	2 EMG
Auflösung (Quantisierungsstufen)	8 bit
Verstärkung	k.A.
1 LSB (höchste Verstärkungsstufe)	1 Mikrovolt
Noise	0,7 μ V rms
CMMR	k. A.
Hochpass Filter	10 Hz
Tiefpass Filter	500 Hz
Samplerate	1 kHz
Schnittstelle	Stand-Alone - Gerät
Spannungsversorgung	aufladbare Batterie
Medizinzulassung	nein
Preis	1.500 €
Hersteller	Noraxon http://www.noraxon.com

Tabelle 3.2: Technische Daten *Noraxon Myotrace200*



Abb. 3.10: Noraxon Myotrace200

Das Noraxon Myotrace200 ist ein EMG-Trainingsgerät, das 2 Kanäle für Muskelfeedback zur Verfügung stellt. Die Rückmeldung erfolgt optoakustisch über Display und Kopfhörer. Das Myotrace200 ist für ein unimodales 2-Kanal-Gerät mit einem Preis von etwa 1.500 € etwas teuer.

3.2.2 WaveRider Pro

WaveRider Pro	
Kanäle	5 (1 GSR)
Auflösung (Quantisierungsstufen)	8 bit
1 LSB (höchste Verstärkungsstufe)	0,17 Mikrovolt
Noise	k.A.
CMMR	100 dB (Notch Filter 50Hz)
Hochpass Filter	0,5 Hz
Tiefpass Filter	40 Hz (-70 dB bei 50Hz)
Samplerate	255 Hz
Schnittstelle	seriell
Spannungsversorgung	9 V - Batterie
Medizinzulassung	nein
Preis	1.500 €
Hersteller	Mindpeak http://mindpeak.com

Tabelle 3.3: Technische Daten *Mindpeak WaveRider*



Abb. 3.11: Mindpeak WaverRider Pro

Beim *Wave-Rider* sind vier Kanäle für EEG, EKG, EMG oder EOG nutzbar, ein Kanal dient der Ableitung der EDA. Die Verbindung zum Computer erfolgt über die serielle Schnittstelle. Eine 2-Kanal-Version ist für unter 1000 € erhältlich und gehört damit zu den preisgünstigsten kommerziellen EEG-fähigen Biosignalverstärkern.

3.2.3 Brainmaster Atlantis

Atlantis 4 Kanal Biosignal-Verstärker	
Kanäle	4 (+ 4 AUX Signale)
Auflösung (Quantisierungsstufen)	24 bit
1 LSB (höchste Verstärkungsstufe)	0,023 Mikrovolt
Noise	< 1 μ Vpp
CMMR	120 dB (Notch Filter 50Hz)
Hochpass Filter	-
Tiefpass Filter	k.A.
Samplerate	4 kHz
Schnittstelle	USB
Spannungsversorgung	USB
Medizinzulassung	ja
Preis	2.500 €
Hersteller	Brainmaster Inc., http://www.brainmaster.com

Tabelle 3.4: Technische Daten *Brainmaster Atlantis*



Abb. 3.12: Brainmaster Atlantis

Die *Atlantis*-Reihe ist die Fortsetzung der *Brainmaster*-Produkte für Bio- und Neurofeedback. Das gezeigte Modell bietet im Unterschied zu den preisgünstigeren 2-Kanal-Modellen der Brainmaster AT-2W-Reihe eine Medizinzulassung. Ein spezielles Feature ist die kontinuierliche Impedanzprüfung auch während der Messung.

3.2.4 Symtop EEG-Verstärker

SYMTOPEEG-Amplifier	
Kanäle	16-41
Auflösung (Quantisierungsstufen)	16 bit
1 LSB (höchste Verstärkungsstufe)	0.5 Mikrovolt
CMMR	98 dB (Notch Filter 50Hz)
Noise	< 2,5 μ Vpp
Hochpass Filter	1 Hz / 3 Hz / 10 Hz
Tiefpass Filter	15 / 30 / 45 / 60 / 120 Hz
Samplerate	1 kHz
Schnittstelle	USB
Spannungsversorgung	Netzteil
Medizinzulassung	ja
Preis	4.000 € (16 Kanäle)
Hersteller	Symtop, http://www.symtop.com

Tabelle 3.5: Technische Daten *Symtop-EEGAmplifier*

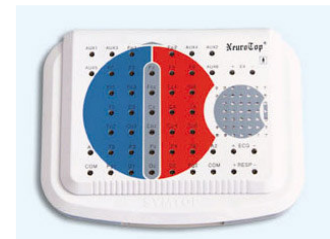


Abb. 3.13: Symtop EEG-Amplifier

Dieses Gerät bietet wählbare Filtereinstellungen und befindet sich in der unteren Preisklasse für Vielkanalverstärker. Noise-Charakteristik und Samplerate entsprechen allerdings nicht den Voraussetzungen zur Messung evozierter Potenziale mit hoher Bandbreite und geringer Amplitude.

3.2.5 Leonardo EEG-Verstärker

Leonardo 24 Kanal EEG-Verstärker	
Kanäle	24 (kombinierbar bis 256)
Auflösung (Quantisierungsstufen)	16 bit
1 LSB (höchste Verstärkungsstufe)	0,1 Mikrovolt
CMMR	130 dB
Noise	< 0,6 μ Vpp
Hochpass Filter	0,016Hz - 1kHz
Tiefpass Filter	0,01Hz - 5kHz
Samplerate	200 - 10 kHz
Schnittstelle	seriell / USB
Spannungsversorgung	Netzteil
Medizinzulassung	ja
Preis	5.000 €
Hersteller	Sagura Inc. http://www.polysomnograph.com

Tabelle 3.6: Technische Daten *Sagura Leonardo*

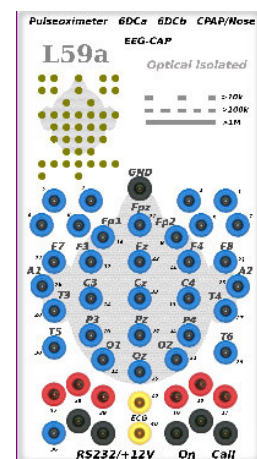


Abb. 3.14: Sagura Leonardo

Der Leonardo EEG-Verstärker zeichnet sich durch gute Noise-Werte, hohe Sampling-Rate und (für einen 24-Kanal-Verstärker) günstigen Preis aus. Als Zusatzgeräte sind Module zur drahtlosen Übertragung (*LanBox*), externe Stimulationsgeräte für optoakustische Triggergenerierung und spezielle Joysticks verfügbar.

3.2.6 USBamp

g.tec USBamp	
Kanäle	16
Auflösung (Quantisierungsstufen)	24 bit
1 LSB (höchste Verstärkungsstufe)	30 nV
CMMR	k.A.
Noise	< 0,3 μ V rms
Hochpass Filter	frei konfigurierbar
Tiefpass Filter	frei konfigurierbar
Samplerate	38,4 kHz
Schnittstelle	USB
Spannungsversorgung	Netzteil
Medizinzulassung	ja
Preis	10.100 €
Hersteller	<i>g.tec Guger Technologies,</i> http://www.gtec.at

Tabelle 3.7: Technische Daten *g.Tec USBamp*



Abb. 3.15: g.Tec USBamp

Eine Besonderheit des *g.tec USBamp* ist der hohe Eingangsspannungsbereich von +/- 250 mV, sodass kaum die Gefahr einer Verstärkersättigung durch Offset-Spannungen besteht. Der Verstärker eignet sich daher gut für DC-Messungen (zB. von *SCPs*). Die Zeitkonstanten der Input-Filter sind per Software konfigurierbar.

3.2.7 Synamp EEG-Verstärker

Das NeuroScan *Synamp2* - System ist ein hochwertiger EEG-Verstärker für den medizinischen und wissenschaftlichen Einsatz. Es zeichnet sich durch hohe Eingangsimpedanzen (10 GOhm), hohe Samplingrate und Auflösung aus, dadurch können sowohl Signale mit einer hohen Bandbreite wie EKPs als auch DC-Potenziale erfasst werden.

Neuroscan Synamp2 EEG Verstärker	
Kanäle	64 monopolar, 4 bipolar
Auflösung (Quantisierungsstufen)	24 bit
1 LSB (höchste Verstärkungsstufe)	3 nV
CMMR	108 dB
Noise	< 4 μ Vpp
Hochpass Filter	DC / 0,5 Hz
Tiefpass Filter	3.500 Hz
Samplerate	20 kHz
Schnittstelle	USB
Spannungsversorgung	Netzteil
Medizinzulassung	ja
Preis	32000 € (mit Software 48.000 €)
Hersteller	<i>Neuroscan,</i> http://www.neuro.com

Tabelle 3.8: Technische Daten *Neuroscan Synamp*



Abb. 3.16: Neuroscan Synamp

3.3 Freie Hardware: Das *OpenEEG* - Projekt

Das *OpenEEG* Projekt versucht seit 2001, kostengünstige Alternativen für die Verstärkung und Digitalisierung von EEG-Signalen zu entwickeln. Die hierfür entwickelte Hardware kann durch Veränderung der Verstärkungskonstanten auch zur Ableitung anderer bioelektrischer Signale wie EMG, EKG oder EOG verwendet werden. Zusätzlich ist die Messung von Sensorinformationen wie Hautleitwert, Temperatur oder Beschleunigung möglich. Bisher wurden Baupläne für drei unterschiedliche Konzepte zur Verstärkertechnik publiziert und von zahlreichen Enthusiasten weltweit umgesetzt. Dadurch stieg auch die Anzahl verwendbarer Open-Source Software für diesen Bereich, sodass heute verschiedene Pakete zum kostenlosen Download bereit stehen, deren Einsatzmöglichkeit sich von der Wavelet-Analyse von Signalen bis zu experimentellen BCI-Applikationen erstreckt. Die *OpenEEG*-Community organisiert sich primär über eine Mailing-Liste, die eine Hilfestellung für neue Mitglieder und ein Forum für Entwicklungen darstellt. Grundsätzliche Informationen zum Projekt gibt es auf der Homepage <http://openeeg.sf.net>, wo auch die Schaltpläne und Bauanleitungen für die Hardware verfügbar sind. Speicherplatz für die Homepage und Mailingliste werden von <http://www.sourceforge.net> kostenlos zur Verfügung gestellt.

Die mittlerweile ausgereiften Hardwarekonzepte heißen *ModularEEG*, *MonolithEEG* und *SoundCardEEG*. Das *SoundCardEEG* verwendet die Soundkarte des PC zur Digitalisierung der Signale (mittels FM- oder AM- Modulation des EEG-Signals und Demodulation durch eine PC-Software). Im Folgenden werden *ModularEEG* und *MonolithEEG* genauer vorgestellt, weitere Details zum *SoundcardEEG* siehe [OPE04].

3.3.1 Der *ModularEEG* - Verstärker

Der *ModularEEG* - Verstärker wurde von Jörg Hansmann entwickelt [HAN03]. Das System besteht aus zwei bis vier Platinen (*Printed Circuit Boards, PCB*) im Format 8 x 10 cm². Eine Platine beherbergt den Mikrocontroller zur Digitalisierung und Aufbereitung der Daten (*Digital Board*), jeweils zwei Verstärkerkanäle sind auf den weiteren Platinen (*Analog Boards*) untergebracht. So ist die Anzahl der Kanäle von zwei bis sechs wählbar.

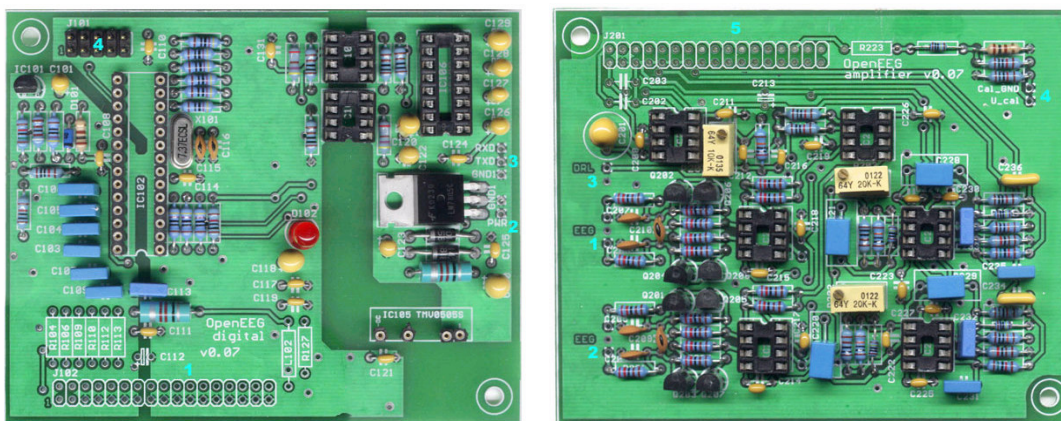


Abb. 3.17: Digital-Board (links) und Analog-Board (rechts) des *ModularEEG*-Verstärkers

3.3.1.1 Die Verstärkerplatine des ModularEEG: das Analog Board

Das zu verstärkende Signal gelangt über die Elektrodenkabel zur Verstärkerplatine. Dort sorgt als erstes eine Schaltung für eine Strombegrenzung zum Schutz von Benutzer/in und Bauteilen (*ESD-Protection*). Danach folgt die erste von drei Verstärkungsstufen, die von einem hochwertigen Instrumentenverstärker (*INA114*, *Texas Instruments*) durchgeführt wird. Der Instrumentenverstärker bietet eine hohe Unterdrückung des Gleichtaktanteils ($CMR > 115$ dB), der eingestellte Verstärkungsfaktor für den Gegentaktanteil ist 12. Um die Gleichtaktunterdrückung weiter zu erhöhen, wird eine Bezugspotenzialsteuerung (*driven right leg circuit*, *DRL*) verwendet [RIJ94]. Hierbei wird das Gleichtaktsignal invertiert und als aktives (output-) Signal über eine Elektrode an den Benutzer zurückgesendet, wodurch eine signifikante Verminderung der Gleichtaktanteile durch Auslöschung erreicht werden kann.

Zwischen der ersten und der zweiten Verstärkerstufe befindet sich ein Hochpassfilter zur Entfernung von Gleichspannungsanteilen, die etwa durch Elektrodenpolarisierung entstehen. Die untere Grenzfrequenz dieses Filters beträgt 0.16 Hz. Der Verstärkungsfaktor der zweiten Stufe kann mittels Potentiometern von 6 bis 100 eingestellt werden. So ist auch eine Verwendung für stärkere Biosignale wie EOG oder EKG möglich. Die dritte Verstärkerstufe beinhaltet einen Tiefpassfilter dritter Ordnung mit 59 Hz Grenzfrequenz (*Anti-Aliasing-Filter*, siehe 3.1.5). Der gesamte Verstärkungsfaktor für das Originalsignal beträgt 1.152 bis 19.200.

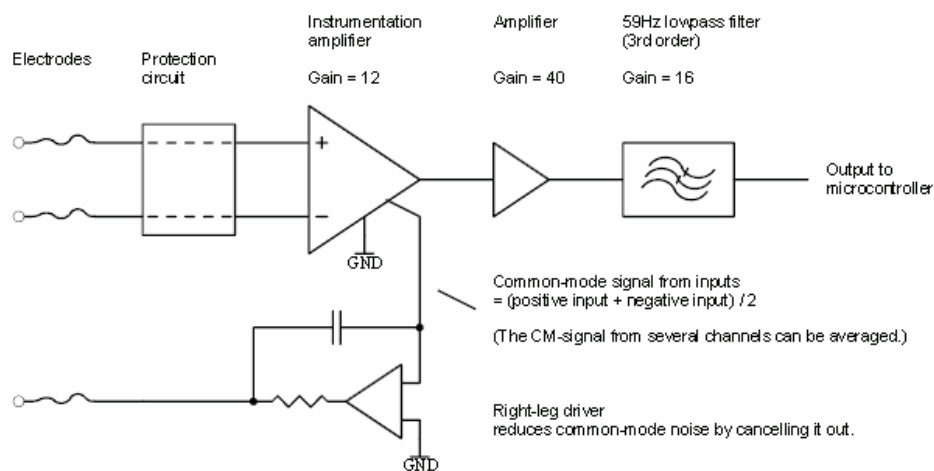


Abb. 3.18: Blockdiagramm der ModularEEG-Verstärkerstufen und DRL-Schaltung

3.3.1.2 Die Hauptplatine des ModularEEG: das Digital Board

Im Wesentlichen beinhaltet die Digitalplatine Komponenten für die Herstellung einer geregelten, galvanisch getrennten Betriebsspannung, weiters Komponenten zur Erzeugung der Referenzspannung für die Verstärkung, einen Mikrocontroller und eine Übertragungseinheit für den Datenaustausch mit dem PC. Die galvanische Trennung dient als Patentenschutzmaßnahme und wird durch einen DC-DC-Konverter (*TMV0505*) bzw. Optokoppler für die Datenleitungen gewährleistet. Die Isolationsspannung des Konverters beträgt 3.000V (Spitze) bzw. 1.000V (kontinuierlich). Der Mikrocontroller vom Typ *Atmel AVR ATmega8* führt die Digitalisierung der verstärkten Signale und die Datenübertragung durch. Der *ATmega8* ist ein 8 Bit RISC-Prozessor mit sechs internen A/D-Wandlern (10 bzw. 8 Bit), einer Ausgabemöglich-

keit für pulswellenmodulierte Signale (*PWM-Output*) und einem seriellen *UART-Interface (Universal Asynchronous Receiver Transmitter)*. Das Programm des Mikrocontrollers, die *Firmware*, steuert die A/D-Wandlung der Messwerte und ihre Aufbereitung in Pakete des Übertragungsprotokolls. Die Datenpakete werden über das *UART-Interface* des Controllers ausgegeben. Eine galvanische Trennung durch Optokoppler dient zum Schutz vor Kurzschlüssen über die Datenleitungen. Die Optokoppler halten die Isolationsbarriere bei Spannungen von 2.500V eine Minute, bei 480V kontinuierlich. Ein Schnittstellenkonverter (*Max232*) hebt die Signalpegel vom *TTL-Level (5V)* auf den Pegel der *RS232-Schnittstelle (+/-10V)* an. Dadurch können die Pakete an die serielle Schnittstelle eines Endgerätes (PC, PDA) übermittelt werden [GRI03].

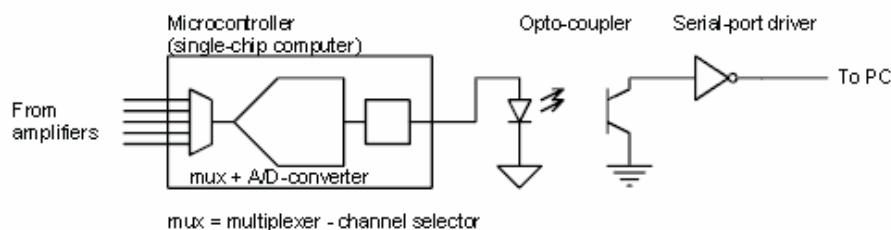


Abb. 3.19: Blockdiagramm der Digitalisierung und Datenübertragung zum PC

Die Firmware für den Mikrocontroller wird am PC in der Programmiersprache C (*GNU AVRgcc*) entwickelt, dann in Maschinensprache für den AVR übersetzt und in das EEPROM des Chips übertragen. Alle dafür nötigen Entwicklungstools orientieren sich am ANSI-C-Standard und können über <http://www.avrfreaks.net> kostenlos bezogen werden. Die Übertragung in den Mikrocontroller kann mit dem *STK200-Interface* über den Parallelport des PC oder mit dem *STK500-Interface* über die serielle Schnittstelle erfolgen [GRI03].

3.3.1.3 Spezifikationen des Modular EEG:

Anzahl der Kanäle	2 - 6
Auflösung	vier 10 bit, zwei 8 bit
kleinste messbare Eingangsspannung	0.5 μ V
größte messbare Eingangsspannung	+/-1000 μ V
Weißes Rauschen	~ 1 μ Vpp
Stromverbrauch	70 mA (2 Kanäle)
Isolationsspannung (Spitze)	2.500V (1 Minute)
Isolationsspannung (kontinuierlich)	480 V
Betriebsspannung	9 - 15 V

Tabelle 3.9: *ModularEEG*, technische Spezifikationen [HAN03]

Das *ModularEEG* kann nicht als medizinisches Gerät zertifiziert werden, da die kontinuierliche Isolationsspannung zu niedrig ist (siehe 3.1.7 Patientenschutzmaßnahmen). Ein Blitzschlag direkt in das Gerät würde vermutlich die Isolationsbarriere durchbrechen. Das Gerät muss daher auf eigene Verantwortung unter Bedachtnahme auf größtmögliche Sicherheit angewendet werden. Durch die Verwendung eines akkubetriebenen Laptop-Computers und Batteriebetrieb des *ModularEEG* kann das Sicherheitsrisiko auf ein Minimum reduziert werden [GRI03].

3.3.2 Der *MonolithEEG* - Verstärker

Das *MonolithEEG* ist ein kleiner, leichter 2-Kanal EEG Verstärker, der von Reiner Münch 2005 entwickelt wurde [MUE05]. Das Design orientiert sich prinzipiell am *ModularEEG*, bietet jedoch einige Änderungen und Verbesserungen. Durch die Verwendung des *SMD*-Fertigungsprozesses konnten Größe und Gewicht signifikant reduziert werden, die analogen und digitalen Komponenten finden somit auf einer Platine Platz, die doppelseitig bestückt ist. Die geringeren Signalwege und die Verwendung geringfügig steilerer Filter bewirkt eine etwas höhere Gleichtaktunterdrückung, wodurch eine Verwendung von Elektrodenkabeln ohne Schirmung möglich ist. Die Analog-Digitalwandlung erfolgt mit einer Genauigkeit von 10 Bit und einer Abtastrate von bis zu 1kHz. Für die Datenübertragung zum PC oder zu anderen Endgeräten steht eine USB-Verbindung zur Verfügung. Hierfür wird der Schnittstellenkonverterchip *FT232* der Firma *FTDI* verwendet, durch den eine Übertragung der *UART*-Pakete via USB einfach möglich ist. Am PC erfolgt der Empfang der Datenpakete mittels eines virtuellen COM-Ports (*VCP*), es sind Übertragungsgeschwindigkeiten von bis zu 1 MBit/Sekunde möglich. Die Stromversorgung erfolgt ebenfalls über den USB-Port, was zusätzliche Batterien oder Netzadapter überflüssig macht [MUE05].



Abb. 3.20: Der *MonolithEEG*-Verstärker

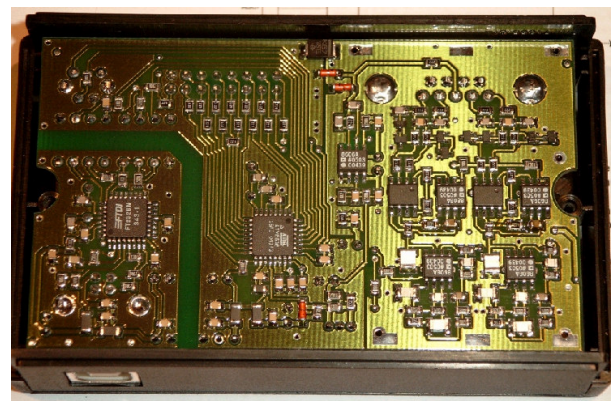
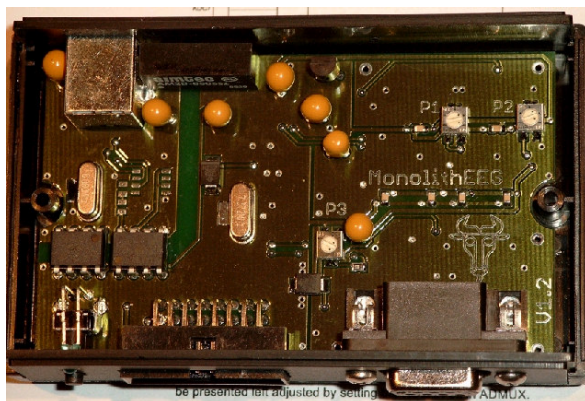


Abb. 3.21: Die *SMD*-Platine des *MonolithEEG* (links: Oberseite, rechts: Unterseite), Größe: 9x5,5cm²

3.3.2.1 Spezifikationen des *Monolith EEG*:

Anzahl der Kanäle	2
Auflösung	10 bit
kleinste messbare Eingangsspannung	0,5 μ V
größte messbare Eingangsspannung	+/-1000 μ V
Weisses Rauschen	~ 1 μ Vpp
max. Offsetspannung bei 100nA	+/- 140 mV
CMRR (DRL)	< 1 μ Vpp/-93db
Stromverbrauch (bei 5 V)	70 mA (2 Kanäle)
Isolationsspannung (Spitze)	2.500V (1 Minute)
Isolationsspannung (kontinuierlich)	480 V
Betriebsspannung	5 V (USB)

Tabelle 3.10: *MonolithEEG*, technische Spezifikationen [MUE05]

3.3.4 Firmware zur Verwendung der *OpenEEG*-Hardware

Die Firmware für den Mikrocontroller steuert die Analog/Digital-Wandlung, verarbeitet die digitalisierten Werte in Datenpakete und sendet die Pakete über das UART-Interface an das Zielgerät. Je nach Ausführung der Firmware ist auch ein bidirektionaler Betrieb möglich, bei dem das Zielgerät auch als Sender fungiert, und z.B. Parameter wie die Sampling-Rate, Start- und Stopp-Kommandos an den Mikrocontroller sendet. Im Rahmen des *OpenEEG*-Projektes sind mehrere Firmware-Protokolle entwickelt worden, darunter auch bidirektionale Varianten. Hier soll nur kurz auf eine einfache, unidirektionale Version eingegangen werden, um die grundlegenden Aufgaben der Firmware zu verdeutlichen. Für ein Senden von Datenpaketen an den PC sind folgende Aufgaben durch die Firmware auszuführen:

- Initialisierung der Ein/Ausgabeports (*Buttons, PWM-Signal*)
- Initialisierung der A/D-Konverter (*Prescaler, Interrupt-Modus*)
- Initialisierung des *Timer-interrupts* (entsprechend der Sampling-Rate)
- Initialisierung des *UART-Interface* (*Baudrate, Daten-Frame*)
- Starten des interruptgesteuerten Programmbetriebes:
 - Vorbereitung eines Datenpaketes im *Timer-Interrupt* (*syncbytes, buttons*)
 - Füllen der Kanalwerte durch den *ADC-Interrupt* (*channel1 - channel6*)
 - Bei vollem Paket: Auslösen der Paketübertragung durch einen *UART-Interrupt* -> Senden der Datenbytes an das Zielgerät
 - Neuer *Timer-Interrupt* nach Verstreichen der *Sampling-Periode*

Dieses Konzept zeigt einen vollständig interruptgesteuerten Programmablauf, im Hauptprogramm werden keine Kommandos ausgeführt. Das erste veröffentlichte Firmware-Protokoll für das *ModularEEG* (*P2-Protokoll*) wurde von *Jörg Hansmann, Jim Peters* und *Andreas Robinson* programmiert und baut die Datenpakete auf diese Weise zusammen. Später folgte das *P3-Protokoll*, welches eine Komprimierung der Kanäle und somit höhere Übertragungs- bzw. Samplingraten ermöglicht, und die Bidirektionalen Versionen *P21* von *Jarek Foltynski* und *P21.2* von *Reiner Münch*. Die Protokolle *P2* und *P3* werden von den meisten Softwareapplikationen unterstützt und sollen nun kurz vorgestellt werden:

Firmware Protokoll P2	Firmware Protokoll P3
Byte 1: Sync Value 0xa5	0ppppppx packet header
Byte 2: Sync Value 0x5a	0xxxxxxx
Byte 3: Version	0aaaaaaa channel 0 LSB
Byte 4: Frame Number	0bbbbbbb channel 1 LSB
Byte 5: Channel 1 Low Byte	0aaa-bbb channel 0 and 1 MSB
Byte 6: Channel 1 High Byte	0ccccccc channel 2 LSB
Byte 7: Channel 2 Low Byte	0ddddddd channel 3 LSB
Byte 8: Channel 2 High Byte	0ccc-ddd channel 2 and 3 MSB
Byte 9: Channel 3 Low Byte	0eeeeeee channel 4 LSB
Byte 10: Channel 3 High Byte	0fffffff channel 5 LSB
Byte 11: Channel 4 Low Byte	1eee-fff channel 4 and 5 MSB
Byte 12: Channel 4 High Byte	
Byte 13: Channel 5 Low Byte	1 and 0 = sync bits.
Byte 14: Channel 5 High Byte	p = 6-bit packet counter
Byte 15: Channel 6 Low Byte	x = auxilary channel byte
Byte 16: Channel 6 High Byte	a-f = 10-bit samples from
Byte 17: Button States (b1-b4)	ADC channels 0 - 5
	- = unused, must be zero

Tabelle 3.11: Übertragungsprotokolle P2 und P3 der *OpenEEG*-Firmware

3.3.5 Elektroden, alternative Sensoren und Hardwareerweiterungen

3.3.5.1 Die Anfertigung von Silberelektroden zur EEG-Ableitung

Prinzipiell können mit der *OpenEEG*-Hardware alle handelsüblichen Elektroden verwendet werden, wobei beim Einsatz des ModularEEG geschirmte Kabel empfohlen werden. Da die Kosten für Elektroden und eine geeignete Befestigungsmöglichkeit (Elektroden-cap bzw. -band) bei einem Kauf im Medizinfachhandel bereits die gesamten Hardwarekosten für den Verstärker übersteigen, ist eine Eigenentwicklung nur konsequent. Als Teil dieser Arbeit wurden aus Reinsilber chlorierbare Silberelektroden angefertigt, die durch ein flexibles Band positioniert werden können. Dieses Setting eignet sich für monopolare oder bipolare Ableitungen von bis zu drei EEG-Kanälen. Die Bezugspotenzialsteuerung wird mit einer handelsüblichen Einweg-EKG-Klebelektrode am Arm befestigt.

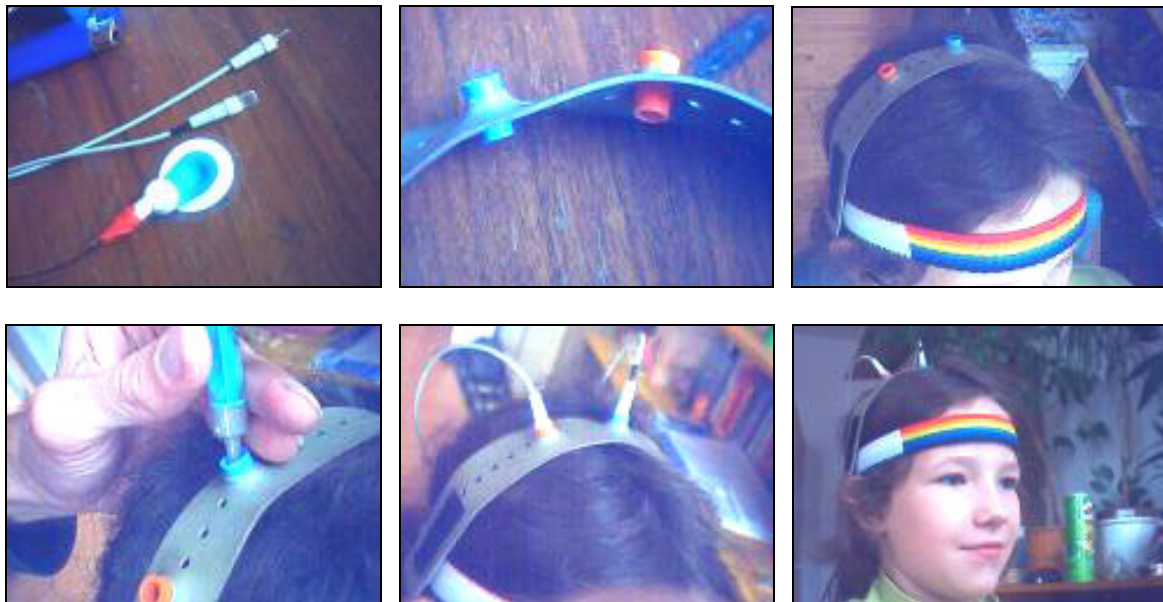


Abb. 3.22: Montage der DIY - Silberelektroden für die EEG-Ableitung

Eine Elektrode besteht aus einem etwa 5mm x 8mm x 1mm großen Stück Reinsilber, das an den Ecken abgerundet wurde. Das Silberstück wird an ein geschirmtes, einpoliges Kabel angelötet. Die Schirmung des Kabels wird konnektorseitig am *ModularEEG* mit dem virtuellen Grundpotenzial (*VGND*) verbunden. Elektrodenseitig bleibt die Schirmung ohne Verbindung und wird mit einem Schrumpfschlauch an der Kabelhülle fixiert. Das Silberstück wird mit Heißkleber in eine Plastikhülse geklebt. Diese Hülse stammt von einem Faserschreiber, sie dient zur Befestigung der Elektrode in einem Plastikröhrchen. Das Röhrchen passt genau in die Löcher eines Gummibandes zur Elektrodenpositionierung. Da das Gummiband durch einen Klettverschluss an beliebiger Stelle am elastischen Kopfband angebracht werden kann, sind fast alle Positionen des 10/20-Systems erreichbar. Für die Vorbereitung der Ableitung wird eine kleine Menge Kontaktgel mit einer Spritze in die Hülse dosiert und mit einem Wattestäbchen leicht einmassiert. Die Elektrode wird mit leichtem Druck eingesetzt, sie kommt in Kontakt mit dem Gel, drückt aber nicht direkt auf die Kopfhaut, sodass auch eine längere Verwendung ohne Schmerzen möglich ist.

3.3.5.2 Die Anfertigung einer EMG-Kappe mit Beschleunigungssensoren

Ausgehend von den Konzepten zur Verwendung von EMG-Signalen und Kopfbewegungen zur Steuerung eines Human-Computer Interface (siehe 2.4.4) wurde eine Stoffkappe mit EMG-Elektroden und einem zweiachsigen Beschleunigungssensor (*ADXL 202E* von Analog Devices) ausgestattet. Der Beschleunigungssensor hat eine Auflösung von 2 Milli-G und misst somit auch Änderungen seiner Lage bezüglich der Erdgravitation. Eine Befestigung des Sensors an der Kappe ermöglicht die Messung von Neigungen des Kopfes (vor und zurück bzw. seitwärts). Die Neigungsinformation wird vom Sensor durch digitale Pulsweitenmodulation und durch analoge Werte ausgegeben, und kann somit direkt vom Mikrokontroller des ModularEEG oder des MonolithEEG verarbeitet werden. Die Digitalisierung der analogen Ausgabewerte durch zwei freie A/D-Wandler ermöglichte eine Verwendung des Sensors ohne Änderung der bestehenden Firmware: Die Beschleunigungswerte werden wie zwei Biosignalkanäle behandelt, digitalisiert und mit dem Übertragungsprotokoll gesendet. Durch eine entsprechende Auswertung in der PC-Software kann die Neigungsinformation zur Steuerung der Cursor-Position verwendet werden.

Die EMG-Elektroden sind modifizierte EKG-Klebelektroden vom Typ *BlueSensor*. Für eine einfachere Applikation in der Kappe wurde der Klebering entfernt, der Kontakt zur Stirn entsteht zur das Aufsetzen der Kappe. Das leitende Element der Elektroden befindet sich in einem gelgetränkten Schwämmchen. Eine Mehrfachverwendung ist durch das Befeuchten des Schwämmchens mit ein paar Tropfen Salzwasser problemlos möglich. Die Elektroden werden über Standard-Konnektoren mit den Kabeln verbunden und sind somit auch leicht zu wechseln. Es werden zwei EMG-Kanäle am *M. frontalis* abgenommen, die Amplitude wird in der Software aufbereitet und dient zum Auslösen von linkem und rechtem Mausklick. In der Evaluation erwies sich das Interface als leicht erlernbar und praktikabel, da schon eine geringe Neigung des Kopfes bzw. ein leichtes Heben der Augenbraue zuverlässig ausgewertet werden konnte. Im praktischen Test erwies sich jedoch eine kamerabasierte Steuerung des Cursors als komfortabler für den Benutzer bzw. die Benutzerin (siehe 7.7.5)

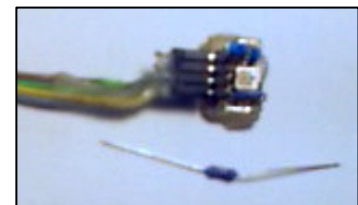
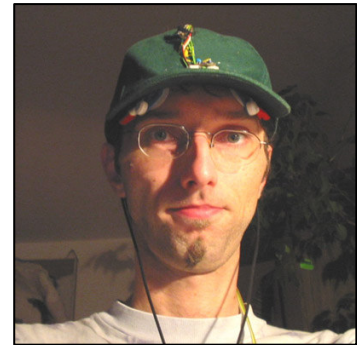


Abb. 3.23: Komponenten und Einsatz der Messkappe

3.3.5.3 Aktive EEG-Elektroden

Im Rahmen des *OpenEEG*-Projektes entstand ein Konzept für aktive EEG-Elektroden, die sich für eine Verwendung ohne Kontaktgel eignen. Durch eine Impedanzwandlung mittels Operationsverstärker (*TLC272*) direkt auf der Elektrode wird das Problem der Signalstörung durch Einkopplung in die Elektrodenkabel entschärft, und eine Verwendung mit höheren, unbalanzierten Impedanzwerten für den Haut-Elektrode-Übergang ist dadurch möglich. Das Design stammt von Jarek Foltinsky und Jörg Hansmann und wurde von einigen Mitgliedern der *OpenEEG*-Gemeinschaft umgesetzt und evaluiert [OPE04]. Obwohl eine erfolgreiche Verwendung ohne Kontaktgel von mehreren Personen berichtet wurde, konnte die zuverlässige und dauerhafte Ableitung des EEG ohne Gel vom Autor dieser Arbeit leider nicht verifiziert werden [ORT05].

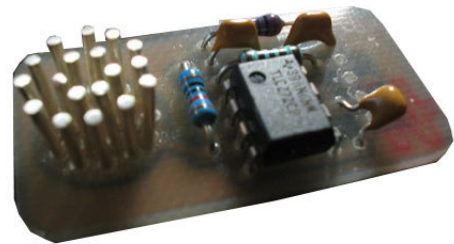


Abb. 3.24: Aktive Elektroden für das OpenEEG

3.3.5.4 Weitere Sensoren

Die Anfertigung von Sensoren für zusätzliche Biofeedbackparameter wie Atmung, Hauttemperatur und Hautleitwert wäre mit einem relativ geringen Aufwand möglich:

- Die **Atmung** könnte durch ein Dehnungselement auf Basis eines Piezowandlers oder durch einen Schiebewiderstand ermittelt werden. Eingearbeitet in einen Gürtel, können diese Bauelemente Änderungen des Brust- oder Bauchumfanges messen. Zur Auswertung würde sich ein freier A/D-Wandler-Port des Mikrokontrollers eignen.
- Zur Ermittlung der **Hauttemperatur** könnte ein wärmeempfindlicher Widerstand (*NTC*) eingesetzt werden. Geeignet wäre z.B. ein SMD-Typ mit 100kOhm Widerstand bei Raumtemperatur. Um einen aktiven Bereich von etwa 25 - 35 Grad Celsius für die Messung der Handtemperatur bei 10 Bit Auflösung zu erreichen, sollte der Wert über Widerstände und einen Operationsverstärker auf den Eingangsbereich der A/D-Ports von 0 - 4 Volt skaliert werden.
- Ein Problem bei der Messung der **EDA** ist die mit 10 Bit zu geringe Auflösung des A/D-Wandlers, da der Wertebereich für den Hautwiderstand sehr groß und eine genaue Messung kleiner Abweichungen dennoch wichtig ist. Eine preisgünstige Möglichkeit wäre die Verwendung eines Timer-ICs (etwa der Reihe 555), der seine Taktgeschwindigkeit durch den Hautleitwert ändert. Die Anzahl der Takte pro Sekunde könnte mit einem digitalen Eingangsport des Mikrokontrollers ermittelt werden. Auf diese Weise könnte durch eine geringere zeitliche Auflösung (welche bei der Ermittlung der EDA nicht kritisch ist) eine ausreichende Genauigkeit der Messung erreicht werden [HAN03].

3.3.5.5 Datenübertragung per Funk mit Bluetooth

Eine Funkübertragung der Daten wäre für manche Anwendungsbereiche wünschenswert. Ein kabelloser Betrieb erhöht den Komfort wie auch die Sicherheit der Verwendung. Eine denkbare Möglichkeit ist der Einsatz eines Bluetooth-Moduls mit *Serial Port Profile (SPP)*. Derartige Module unterstützen die bidirektionale Übertragung von seriellen Daten nach dem *UART*-Standard mit bis zu 230kBaud Übertragungsgeschwindigkeit. Die Daten stehen am Computer durch einen virtuellen Com-Port zur Verfügung und können ohne Softwareänderung übernommen werden. Das Modul könnte beim *ModularEEG* statt dem *MAX232*-Schnittstellenkonverter und beim *MonolithEEG* statt dem *FTDI232*-USB-Chip eingesetzt werden. Die Module bieten eine eingebaute Fehlerkorrektur und Reichweiten von bis zu 10m (Class 2) bzw. bis zu 100m (Class1) [LIS05].



Abb. 3.25: Bluetooth-Modul mit SPP

3.3.5.6 Offline-Datenaufzeichnung

Eine einfache Möglichkeit zur Aufzeichnung einer Messung ohne Computer oder PDA wäre ein USB-Memory Modul wie das *VDRIVE* von *FTDI* (<http://www.vinculum.com>). Dieses Modul bietet einen *UART*-Port, mit dem der Mikrocontroller verbunden werden kann. Gleichzeitig funktioniert das Modul wie ein USB-Memory Stick, und die Daten können nach der Aufzeichnung PC-seitig ausgelesen werden. Einige spezielle Kommandos dienen zum Erstellen einer Datei und zum Lesen und Schreiben von Daten. Die nötige Erweiterung der Firmware für die *OpenEEG*-Verstärker wäre überschaubar (Erstellung der Datei bei Beginn und Schließen der Datei bei Ende der Aufzeichnung).

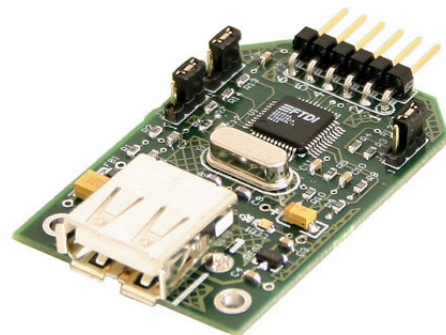


Abb. 3.26: VDrive-Modul von FTDI

4. Softwaremethoden der Signalverarbeitung

Die digitale Signalverarbeitung (*digital signal processing, DSP*) wurde in den letzten Jahrzehnten zu einem Schlüsselement in der elektronischen Datenverarbeitung. Ihre Wurzeln liegen in den 50er und 60er Jahren, wo digitale Computer verfügbar wurden und neue Methoden der Signalverarbeitung vor allem in der militärischen Forschung (Radar- und Sonartechnologie), in der Geologie (Auffindung von Erdölquellen) und in der medizinischen Bildgebung benötigt und angewendet wurden. Durch den Siegeszug der Personal-Computer in den 80er und 90er Jahren eröffnete sich ein weites Feld für die digitale Signalverarbeitung im kommerziellen Markt. Elektronische Geräte wie CD-Player, Mobiltelefone oder Videokonferenz-Systeme wären ohne Signalaufbereitung oder Kompressionsverfahren nicht in dieser Form realisierbar.

Unter einem Signal versteht man grundsätzlich die Beziehung zweier Parameter, z.B. bei einer elektrischen Messung die Beziehung von Zeit und gemessener Spannung. Die Zeit wäre in diesem Fall die unabhängige Variable (*x-Achse, Domäne* bzw. *Abzisse*), die Spannung wäre die abhängige Variable (*y-Achse, Amplitude* bzw. *Ordinate*). Wird die Zeit auf der x-Achse aufgetragen, spricht man von einer Darstellung im Zeitbereich (*time domain*). Weitere gebräuchliche Darstellungsformen für Signale sind der Frequenzbereich (*frequency domain*) und der Ortsbereich (*spatial domain*). Bei der digitalen Signalverarbeitung in Computern liegt das Signal nach der Messung stets als Zeitreihe diskreter Werte vor (siehe 3.1.5 Filterung und Digitalisierung).

In der Auswertung biologischer Signale nimmt die digitale Signalverarbeitung einen wichtigen Stellenwert ein: Eine aussagekräftige Analyse der Werte wird nur durch die Berechnung von Kennzahlen, durch das Herausheben bestimmter Signaleigenschaften oder durch die Transformation in geeignete Darstellungsformen möglich. Dies sind die Domänen der digitalen Signalverarbeitung, die solche Methoden durch Softwarealgorithmen zur Verfügung stellt und eine mathematisch-theoretische Basis zur Signalanalyse in digitalen Computern bietet.

4.1 Lineare Systeme

Ein System ist ein Prozess, der aus einem Eingangssignal ein Ausgangssignal produziert. Man unterscheidet kontinuierliche und diskrete Systeme. Die digitale Verarbeitung von Daten kann durch diskrete Systeme beschrieben werden. Lineare Systeme erfüllen die Kriterien der *Homogenität* und *Additivität*.

Homogenität bedeutet, dass eine Änderung der Amplitude im Eingangssignale an das Ausgangssignal in gleicher Stärke weitergegeben wird:

$$x[n] \xrightarrow{\text{System}} y[n] \Rightarrow kx[n] \xrightarrow{\text{System}} ky[n] \quad (\text{Gl. 4.1})$$

Additivität ist die Eigenschaft, dass ein System aus der Summe von Inputsignalen auch die Summe ihrer Outputsignale erzeugt:

$$x_1[n] \xrightarrow{\text{System}} y_1[n], x_2[n] \xrightarrow{\text{System}} y_2[n] \Rightarrow x_1[n] + x_2[n] \xrightarrow{\text{System}} y_1[n] + y_2[n]$$

(Gl. 4.2)

Lineare Systeme sind verschiebungsinvariant, wenn ihre Eigenschaften zeitlich konstant bleiben. Eine besondere Eigenschaft linearer Systeme ist, dass harmonische (Sinus-) Schwingungen am Eingang auch harmonische Schwingungen am Ausgang erzeugen. Eine Änderung von Frequenz oder Amplitude wird an das Ausgangssignal entsprechend weitergegeben. Die Hintereinanderschaltung von linearen Subsystemen ist kommutativ und erzeugt wieder ein lineares System. Lineare Systeme erlauben die Überlagerung (*Superposition*) von Signalen (*Signalsynthese*) bzw. das Zerlegen eines Signals in Komponenten (*Dekomposition*). Wird ein komplexes Signal $x[n]$ in einfachere Komponenten $x_i[n]$ zerlegt, produziert ein lineares System die Ausgabewerte $y_i[n]$. Die Addition dieser Ausgabewerte führt zu demselben Ergebnis $y[n]$, das vom System aus dem Originalsignal erzeugt wird. Dieses Prinzip ist die Basis der Digitalen Signalverarbeitung: Signale werden durch Dekomposition in einfachere Signalkomponenten zerlegt, modifiziert und wieder zusammengeführt. Besonders wichtig bei der Signalanalyse sind die *Impulsdekomposition*, bei der Systemantworten auf einzelne Impulse untersucht werden (*impulse response*) und die *Fourier-Dekomposition*, bei der ein beliebiges Signal als Summe von Sinus- und Cosinuswellen dargestellt wird.

4.2 Grundlegendes zur Signalstatistik

Die statistische Auswertung eines Signals erlaubt Rückschlüsse auf den zugrundeliegenden Prozess: Handelt es sich um stationäre oder um dynamische Phänomene, in welchem Maße sind Zufalls- und Störgrößen im Signal vorhanden; in wieweit ist das Signal deterministisch oder stochastisch usw. Erwartungswert und Standardabweichung erlauben eine erste Abschätzung dieser Signaleigenschaften.

4.2.1 Erwartungswert und Standardabweichung

Der Erwartungswert μ ist der Mittelwert aller Sample-Werte aus einem Set von N Samples, die Standardabweichung σ ist ein Maß für die durchschnittliche Abweichung der Samples vom Mittelwert. Das Quadrat der Standardabweichung σ^2 ist die *Varianz* des Signals [SMI01]:

$$\mu = \frac{1}{N} \sum_{i=0}^{N-1} x_i \quad (\text{Gl. 4.3})$$

$$\sigma^2 = \frac{1}{N-1} \sum_{i=0}^{N-1} (x_i - \mu)^2 \quad (\text{Gl. 4.4})$$

Abb. 4.1 (links) zeigt ein stationäres Signal mit einem Erwartungswert von 3 und einer Varianz von 0,2. Abb. 4.1 (rechts) zeigt ein nicht-stationäres Signal. Erwartungswert und Varianz können bei diesem Signal nur in kleinen Abschnitten ermittelt werden:

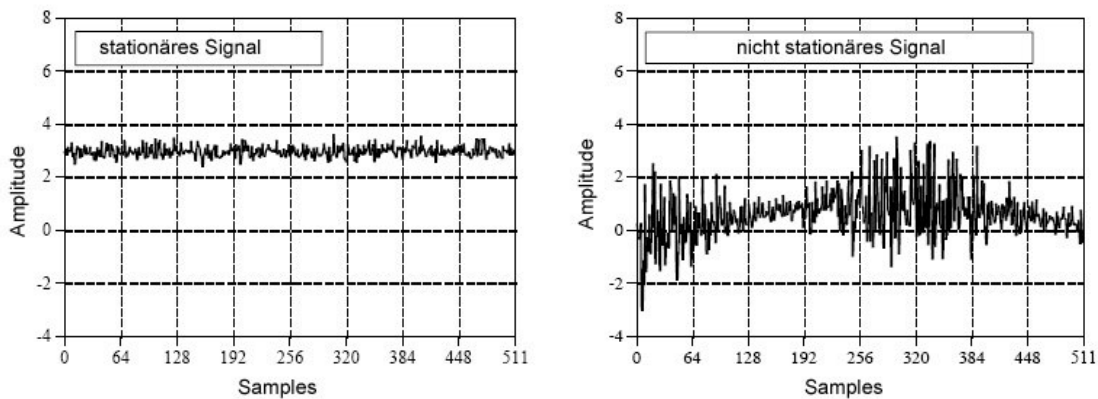


Abb. 4.1: Stationäre und nichtstationäre Signale

4.2.2 Histogramm, Wahrscheinlichkeitsdichte

Beim *Histogramm* werden die verschiedenen Messwerte auf der x-Achse und ihre Häufigkeiten auf der y-Achse aufgetragen, es entsteht eine Übersicht der Häufigkeiten ihres Auftretens. Erwartungswert und Standardabweichung lassen sich aus dem Histogramm ablesen bzw. daraus berechnen. Die *Wahrscheinlichkeitsdichtefunktion* ist eine normierte, kontinuierliche Version des Histogramms, welche die Auftretswahrscheinlichkeiten für einen Wert angibt [SMI01].

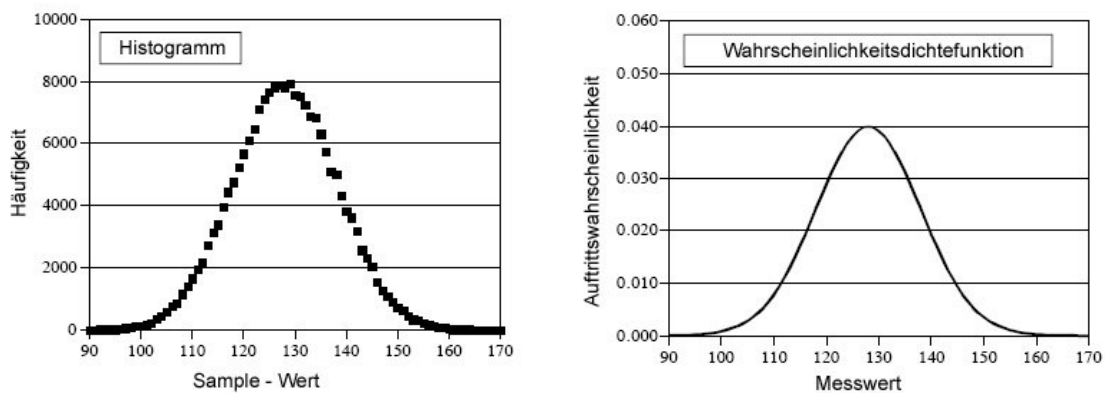


Abb. 4.2: Histogramm und Wahrscheinlichkeitsdichte

Die Addition zufälliger Prozesse erzeugt nach dem *zentralen Grenzwertsatz* eine glockenförmige Wahrscheinlichkeitsdichtefunktion, die *Normalverteilung* bzw. *Gauß-Verteilung*, die in der Natur bzw. auch in der Signalverarbeitung oft anzutreffen ist.

4.3 Digitale Filter

Allgemein dienen Filter zur Separation kombinierter Signale oder zur Restauration verzerrter Signale. Ein Filter kann bestimmte Frequenzbereiche eines Signals ausblenden, während andere Frequenzen unverändert bleiben oder verstärkt werden. Je nachdem, welcher Frequenzbereich den Filter ungehindert passieren kann, spricht man von *Tiefpass*-, *Hochpass* oder *Bandpass-Filtern*. Filter können analog mittels Baugruppen von Widerständen, Kondensatoren und Operationsverstärkern herge-

stellt werden (siehe 3.1.5), sie können aber auch digital durch Softwarealgorithmen realisiert werden. Die *Filter-Ordnung* (Anzahl der Pole) bestimmt die Komplexität und das Verhalten des Filters: eine höhere Ordnung bewirkt ein genaueres Abscheiden von Frequenzen aber auch eine größerer Verzögerung im Zeitbereich. Man unterscheidet verschiedene Filtertypen, die bekanntesten sind *Bessel*-, *Butterworth*- und *Chebyshev*- Filter. Diese unterscheiden sich durch ihre Charakteristik im Zeit- und Frequenzbereich: Das Ausblenden von Signalkomponenten jenseits einer gewünschten Grenzfrequenz (*cut-off frequency*) ist niemals exakt möglich, je nach Filtercharakteristik weist das Dämpfungsverhalten ein Gefälle im unerwünschten *Stopp-Band* auf, sodass Anteile in der Nähe der Grenzfrequenz nicht völlig unterdrückt werden können (Steilheit des Filters, *filter-roll-off*). Eine gute Filtercharakteristik im Frequenzbereich bietet der *Chebyshev*-Filter, gefolgt vom *Butterworth*-Filter und vom *Bessel*- Filter, der im Frequenzbereich eher ungenau abschneidet [SMI01]:

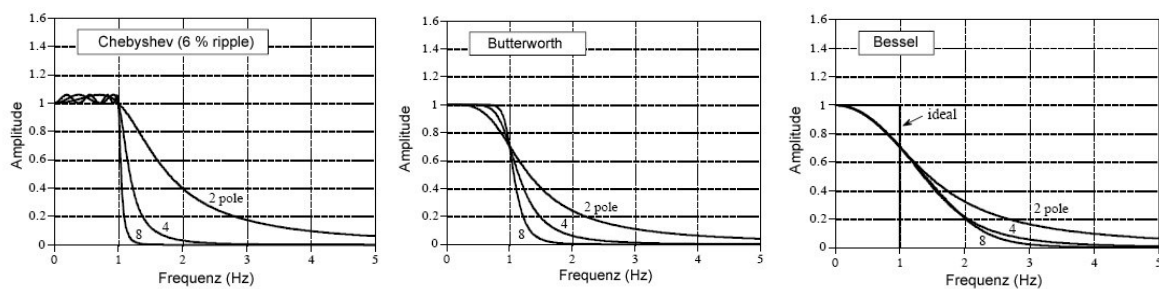


Abb. 4.3: Frequenzantwort von Chebyshev- Butterworth- und Bessel-Filtern

Die Abbildung zeigt das Dämpfungsverhalten von Tiefpassfiltern bei einer Grenzfrequenz von 1 Hz. Der Chebyshev-Filter 8ter Ordnung hat hier das beste Verhalten, wobei Frequenzen ab 1,5 Hz schon fast vollständig unterdrückt werden. Ein Bessel-Filter 8ter Ordnung schafft dies erst ab ca. 4 Hz. Allerdings zeigt der Chebyshev-Filter Unregelmäßigkeiten (*ripple*) im *Passband* (unter 1 Hz), wo das Signal ja unverändert bleiben sollte. Der Butterworth-Filter stellt einen Kompromiss dar und erreicht gute Frequenzcharakteristik bei flachem Passband [SMI01].

Eine genaue Frequenzselektivität eines Filters bewirkt allerdings ein Überschießen und Störschwingungen im Zeitbereich (*overshooting, ringing*): Die Filter verändern die Amplitude sowie die Phase eines Signals, was an der Verarbeitung eines Rechteck-Impulses deutlich wird. Hier zeigt der Chebyshev-Filter die deutlichste Verfälschung, gefolgt vom Butterworth-Filter. Am besten schneidet der Bessel-Filter ab, der kaum Überschießen produziert und im Gegensatz zum Chebyshev-Filter eine lineare Phase zeigt (steigende und fallende Flanke eines Rechteck-Impulses sind nach dem Filter noch symmetrisch) [SMI01].

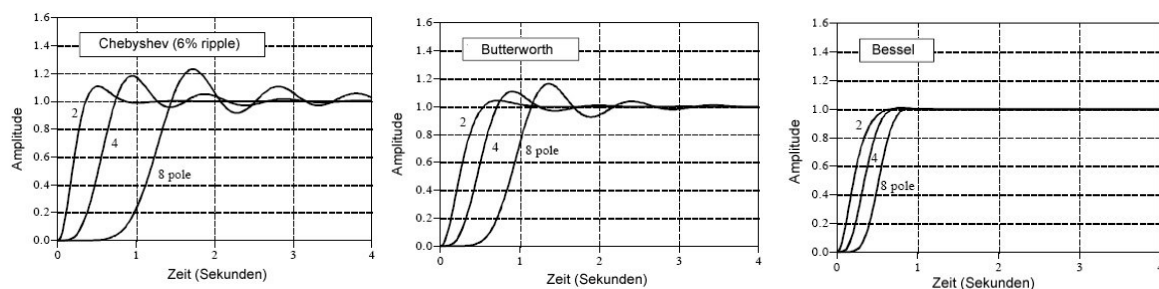
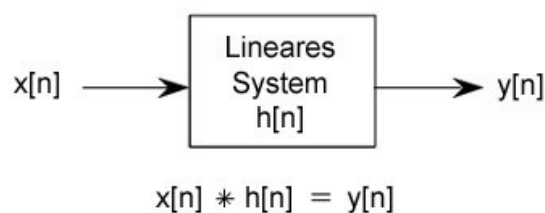


Abb. 4.4: Sprungantwort von Chebyshev- Butterworth- und Bessel-Filtern

Die Verwendung digitaler Filter bietet gegenüber analoger Filtertechnik viele Vorteile, so entfallen komplexe Baugruppen, die Hardwarekosten sind geringer und auch die Filterperformance ist aufgrund der höheren Genauigkeit bei der digitalen Verarbeitung besser.

4.3.1 Faltung (Convolution)

Digitale Filter können u.a. durch die Operation der Faltung (*convolution*) realisiert werden. Die Faltung kann als lineares System beschrieben werden, das aus einem Eingangssignal $x[n]$ und einem Faltungskern $h[n]$ ein Ausgangssignal $y[n]$ erzeugt:



Bei der diskreten Faltung werden die einzelnen Werte des Eingangssignals (*Impuls-Dekomposition*) mit dem Faltungskern multipliziert und zu Werten des Ausgangssignals addiert. Ein Wert des Ausgangssignals wird dabei von mehreren Eingangswerten beeinflusst, da er durch eine Anwendung des (vertikal gespiegelten) Faltungskernes auf die letzten M Samples entsteht. Besteht das Eingangssignal lediglich aus der Delta-Funktion (ein Samplewert = 1, sonst alle Werte *null*), so erhält man als Ausgangssignal den Faltungskern (= die Impulsantwort). Der Faltungskern eines Filters wird auch *Filter-Kern* genannt, er beschreibt das lineare System vollständig. Die mathematische Schreibweise für die (diskrete) Faltung von $x[i]$ mit einem aus M Werten bestehenden Kern $h[j]$ lautet [SMI01]:

$$y[i] = x[i] * h[i] \Leftrightarrow y[i] = \sum_{j=0}^{M-1} h[j]x[i-j] \quad (\text{Gl. 4.5})$$

Die Faltung ist eines der wichtigsten Werkzeuge der digitalen Signalverarbeitung, da durch das Design entsprechender Impulsantworten verschiedenste Filterfunktionen realisiert werden können. Hier zwei Beispiele für Tiefpass und Hochpass-Filter [SMI01]:

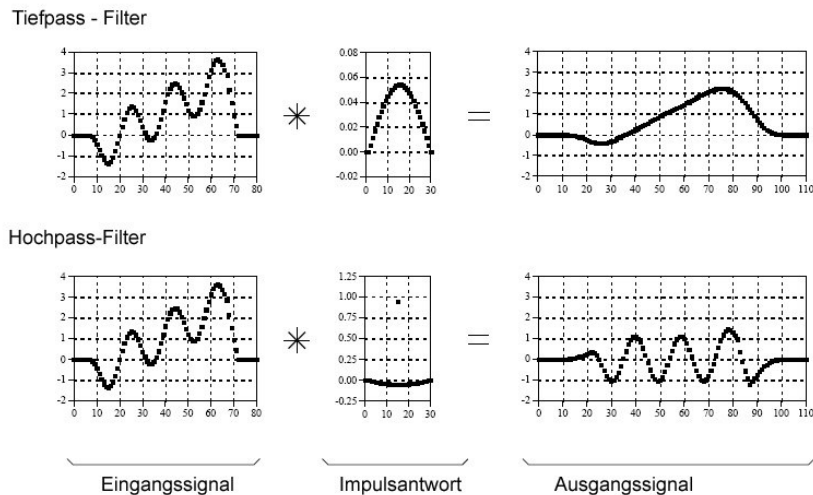


Abb. 4.5: Auswirkungen von Tiefpass- und Hochpassfiltern

Das Ausgangssignal ist nach der Faltung um die Anzahl der Werte der Impulsantwort länger als das Eingangssignal. Ein Filter bewirkt immer eine gewisse Verzögerung. Die Impulsantwort von Tiefpassfiltern besteht oft aus einer Reihe von positiven Werten, was zu einer gewichteten Summe der Eingangswerte führt. Hochpassfilterkerne bestehen oft aus einer positiven Spitze, die von negativen Werten umgeben ist. Soll der Filter das Signal weder verstärken noch dämpfen, muss die Summe der Werte der Impulsantwort *eins* ergeben. Die Grenzfrequenz eines Filters wird durch Streckung oder Stauchung der Impulsantwort beeinflusst. Neben der Unterdrückung oder Hervorhebung von Frequenzen eignen sich Filter zum Hinzufügen oder Entfernen von Echos sowie zur Realisierung mathematischer Operationen wie Differentiation oder Integration [SMI01].

Eine algorithmische Realisierung der Faltung könnte folgendermaßen aussehen:

```
// Eingangssignal: x[80], Impulsantwort: h[30]
// Ausgangssignal: y[110]

int i, j;
for (i = 0; i < 110; i++)
    for (y[i]=0; j=0; j<30; j++)
        if ((i-j >= 0) && (i-j<80))
            y(i) = y(i) + h(j) * x(i-j)
```

4.3.2 Weitere Arten digitaler Filter

Die Form der Impulsantwort bestimmt die Eignung des Filters für bestimmte Zwecke. So hat etwa der *moving average* - Filter, dessen Impulsantwort aus m Werten der Amplitude $1/m$ besteht, ein verzerrungsfreies Verhalten im Zeitbereich, jedoch einen schlechten *roll-off* und somit ungenaues Verhalten im Frequenzbereich. Zur Separation von Frequenzen eignet sich der *windowed-sinc* - Filter, dessen Impulsantwort eine geglättete Form der Funktion $\sin(x)/x$ darstellt. Der *Wiener*-Filter eignet sich für die optimale Erhöhung des Signal-Rausch-Abstandes.

Durch Faltung erzeugte Filter werden als *FIR - Filter* bezeichnet (*finite impulse response*), da die Impulsantwort im Gegensatz zu den rekursiven *IIR-Filtern* begrenzt ist (*infinite impulse response*). Rekursive Filter beziehen nicht nur das Eingangssignal sondern auch die letzten Werte des Ausgangssignals in die Filterwirkung mit ein. Sie werden nicht durch einen Faltungskern sondern durch eine Reihe von *Rekursionskoeffizienten* definiert. IIR- Filter können trotz der (theoretisch) unendlichen Impulsantwort in Computern verwendet werden, indem ab einer bestimmten Genauigkeit gerundet wird. Sie erzielen lange Impulsantworten bei geringem Rechenaufwand und eignen sich (ähnlich den *windowed-sinc* - Filtern) für die Separation von Frequenzen [SMI01].

4.3.3 FFT-Convolution

Ein Problem bei der Berechnung der Faltung sind die langen Rechenzeiten bei größeren Faltungskernen. Da die Faltung im Zeitbereich einer Multiplikation im Frequenzbereich entspricht, und durch den FFT-Algorithmus eine schnelle Variante zur Berechnung der Fourier-Analyse zur Verfügung steht, bedeutet die Multiplikation im Frequenzbereich mit anschließender Rücktransformation einen Geschwindigkeitsgewinn, wenn der Faltungskern länger als 64 Werte ist [SMI01].

4.4 Korrelation und Autokorrelation

Korrelation ist ein Verfahren, um ein Maß für die Ähnlichkeit zweier Signale zu gewinnen, um z.B. festzustellen, ob ein Signal in einem anderen Signal enthalten ist. Das mathematische Verfahren ist bis auf einen Unterschied im dem der Faltung identisch: der Faltungskern wird vorwärts verwendet (also nicht vertikal gespiegelt). Die Korrelation zwischen zwei Signalen heißt *Kreuz-Korrelation (cross-correlation)*, die Korrelation eines Signals mit sich selbst nennt man *Autokorrelation*, sie dient der Aufindung von Periodizitäten. Die mathematische Schreibweise der diskreten Kreuz-Korrelation [SMI01]:

$$R_{xt} = \sum_{j=0}^{M-1} t[M-1-j]x[i-j] \quad (\text{Gl. 4.6})$$

Das Eingangssignal $x[n]$ wird nach dem Zielsignal $t[n]$ durchsucht. Das Ausgangssignal $R_{xt}[n]$ hat an jener Stelle ein Maximum, an der die Form des Zielsignals im Eingangssignal vorkommt. Eine Normierung der Korrelation auf die Werte $[-1,1]$ erfolgt durch die Berechnung des *Korrelationskoeffizienten*. Die Kreuzkorrelation wird auch als *Detektionsfilter (matching filter)* bezeichnet [SMI01].

4.5 Diskrete Fourier Transformation

Die Fourier Transformation ist ein vom französischen Mathematiker Jean Baptiste Fourier entwickeltes Verfahren, ein beliebiges Signal durch eine Summe harmonischer Schwingungen (Sinus- und Cosinuswellen) zu beschreiben. Dies ermöglicht die Darstellung und Modifikation eines Signals durch seine Frequenzkomponenten. Man spricht von einer Transformation des Signals vom Zeit- in den Frequenzbereich, eine Rücktransformation ist verlustfrei möglich.

Je nach Art des Signals (kontinuierlich, diskret, periodisch oder aperiodisch) finden verschiedene Arten der Fourier Transformation Anwendung. Bei der digitalen Signalverarbeitung liegen stets diskrete Werte vor, die mittels der Diskreten Fourier Transformation (DFT) behandelt werden. Ein Signal, das aus N Samples besteht, wird dabei durch Dekomposition in $N/2+1$ Cosinus- und $N/2+1$ Sinus-Wellen unterschiedlicher Frequenz und Amplitude zerlegt. Diese Wellen sind die (orthogonalen) Basisfunktionen der DFT:

$$\begin{aligned}
 c_k &= \cos(2\pi ki / N) & k \dots \text{jeweilige Basisfunktion } [0 \dots N/2] \\
 s_k &= \sin(2\pi ki / N) & i \dots \text{Index des Samples } [0 \dots N-1] \quad (\text{Gl. 4.7})
 \end{aligned}$$

Damit sind konstante Signalanteile (durch c_0) und Schwingungen bis zur halben Sampling-Frequenz ($c_{N/2}$) darstellbar. Die Amplituden der jeweiligen Basisfunktionen werden durch Koeffizienten bestimmt, diese sind die Informationsträger des Frequenzgehaltes. Aus Gründen der Konsistenz mit der komplexen Fourier Transformation heißen die Koeffizienten der Cosinuswellen *Realteil* ($Re[k]$) und die Koeffizienten der Sinuswellen *Imaginärteil* ($Im[k]$) [SMI01].

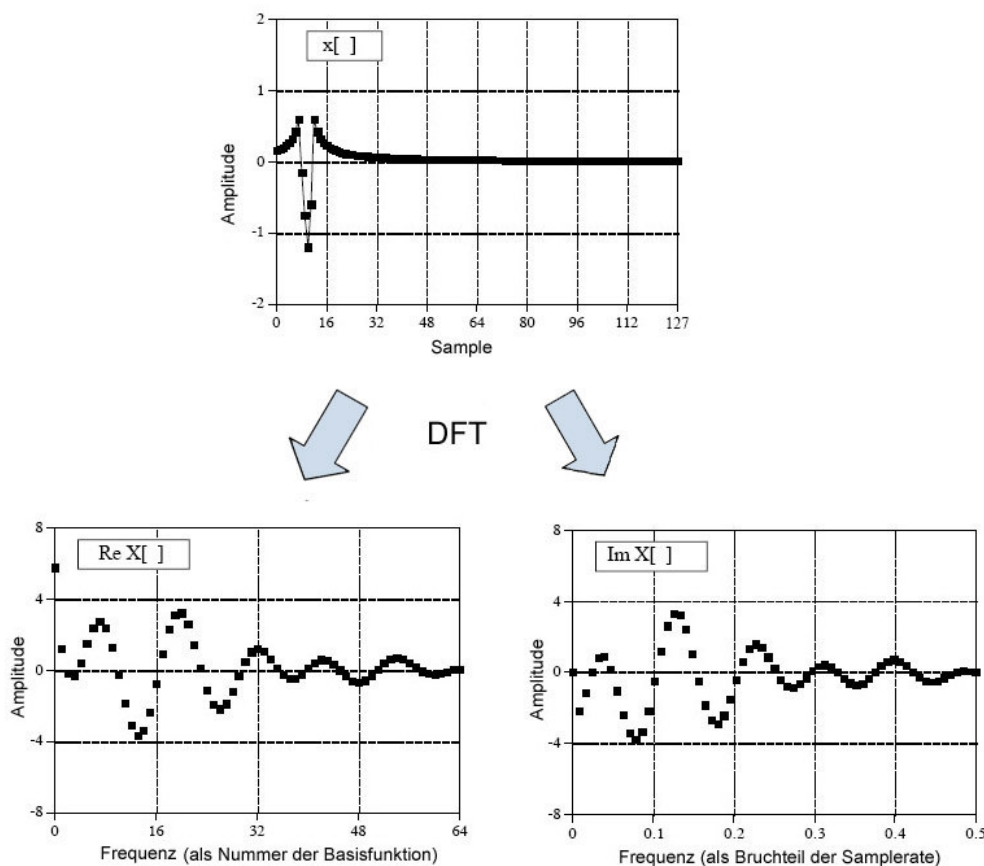


Abb. 4.6: Die Darstellung eines Signales im Frequenzbereich durch die Diskrete Fourier Transformation

Die Abbildung zeigt die Zerlegung eines Signals $x[n]$ mit einer Länge von 128 Samples in 2×65 harmonische Basisfunktionen mit den Cosinus-Koeffizienten $Re X[k]$ und den Sinus-Koeffizienten $Im X[k]$.

Bei der Durchführung der DFT-Analyse werden die Koeffizienten durch Korrelation des Signals mit den Basisfunktionen ermittelt [SMI01]:

$$\operatorname{Re} X[k] = \sum_{i=0}^{N-1} x[i] \cos(2\pi ki / N)$$

$$\operatorname{Im} X[k] = \sum_{i=0}^{N-1} x[i] \sin(2\pi ki / N)$$

k ... jeweilige Basisfunktion [0 ... $N/2$]

i ... Index des Samples [0 ... $N-1$]

$x[i]$.. Signal im Zeitbereich (Gl. 4.8)

$\operatorname{Re} X[k]$... Cosinuskoeffizienten

$\operatorname{Im} X[k]$... Sinuskoeffizienten

Die **inverse DFT** beschreibt die Synthese des Signals aus den Basisfunktionen :

$$x[i] = \sum_{k=0}^{N/2} \operatorname{Re} X[k] \cos(2\pi ki / N) + \sum_{k=0}^{N/2} \operatorname{Im} X[k] \sin(2\pi ki / N) \quad (\text{Gl. 4.9})$$

Für eine bessere Interpretationsmöglichkeit der Signale im Frequenzbereich können die Koeffizienten der Cosinus- und Sinuswellen in die Polardarstellung übergeführt werden: Real- und Imaginärteil werden in *Magnitude* und *Phase* umgewandelt (Gl. 4.10):

$$\operatorname{Mag} X[k] = \sqrt{(\operatorname{Re} X[k])^2 + (\operatorname{Im} X[k])^2}$$

$$\operatorname{Phase} X[k] = \arctan\left(\frac{\operatorname{Im} X[k]}{\operatorname{Re} X[k]}\right)$$

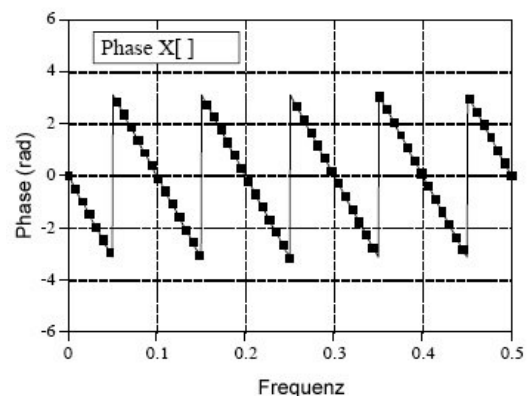
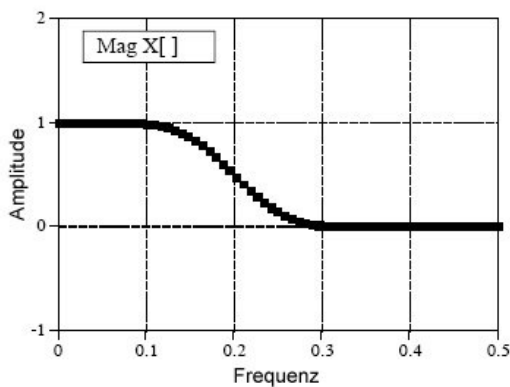


Abb. 4.7: Frequenzantwort eines Tiefpass-Filters in der Polardarstellung

Die Vorteile der Fourier-Transformation sind ihre leichte Anwendbarkeit und ihre vielseitige Verwendbarkeit. Ein Nachteil ist die starke Abhängigkeit von Zeit- und Frequenzbereich: soll ein Signal im Frequenzbereich genau untersucht werden, so benötigt man viele Samples und die zeitlichen Veränderungen "verwaschen". Soll das Signal zeitlich genau analysiert werden, stehen wenige Frequenzkomponenten zur Verfügung. Dieses Prinzip der *Unschärfe* ist zwar unumgänglich, andere Transformationen wie die *Diskrete Wavelet-Transformation* bieten hier durch dynamische Skalierung der Basisfunktionen dennoch Alternativen. Die *Fast Fourier-Transformation (FFT)* ermöglicht eine weitaus schnellere Dekomposition des Signals und bestimmten Voraussetzungen (etwa muss die Anzahl der Samples eine Potenz von 2 sein).

4.6 Weiterführende Klassifikationsmethoden

Die Klassifikation von Daten ist oft die wichtigste Funktionalität von digitalen Signalverarbeitungssystemen. Meist ist die Aufgabenstellung eine Zuordnung des Signals oder Musters zu einer bestimmten Gruppe. So kann die Klassifizierungsaufgabe für ein BCI-System etwa lauten, Bewegungsvorstellungen im EEG zu erkennen, bzw. Artefakte auszuschließen. Die Erkennungsrate ist dabei maßgeblich vom verwendeten Klassifikationsalgorithmus abhängig. Da eine eindeutige Unterscheidung nicht immer möglich ist, wird eine Klassifizierungsstrategie nach der Anzahl der falsch-positiv und falsch-negativ erkannten Muster qualifiziert. Die *ROC-Kurve* (*Receiver Operating Characteristic*) stellt die richtig-positiv erkannten Ziele den falsch-positiven Klassifizierungen gegenüber [SMI01]. Da eine Untersuchung großer Datenmengen aufwändig und zeitintensiv ist, wird als erster Schritt zur Klassifikation meist eine Datenreduktion durch die Extraktion relevanter Merkmale (*feature extraction*) durchgeführt. Bei der Erkennung eines Gesichtes auf einem Bild könnten etwa die Konturen oder Durchmesser und Helligkeit bestimmter Merkmale geeignete Parameter für die Klassifikation sein. Bei einer Analyse von Biosignalen eignen sich oft Frequenzbandaktivitäten oder die Kreuz-Korrelation von Signalen. Entscheidend für die Effizienz der Klassifikation ist die Gewinnung von Merkmalen, welche die relevante Information in möglichst komprimierter Form beinhalten. Der so entstandene Merkmalsraum ist von erheblich geringerer Dimension als die ursprünglichen Daten, somit kann eine Unterscheidung von Eigenschaften leichter durchgeführt werden. Bei der Klassifikation wird der multidimensionale Merkmalsraum in Klassen unterteilt. Meist erfolgt ein *Training* von Klassifikatoren durch die Präsentation von positiven und negativen Beispielen.

4.6.1 Neuronale Netze

Neuronale Netze sind Klassifikationssysteme für Daten, die den biologischen Nervenverbindungen nachempfunden sind. Sie bestehen aus verbundenen Einheiten (*Units, nodes*), welche eingehende Impulse ihrer Verbindungen aufsummieren und begrenzen und an andere Units weitergeben. Eine einfache und gebräuchliche Form neuraler Netzwerke besteht aus drei Schichten (*Input-, Hidden- und Output-Layer*), wobei die Units benachbarter Ebenen voll untereinander verbunden sind.

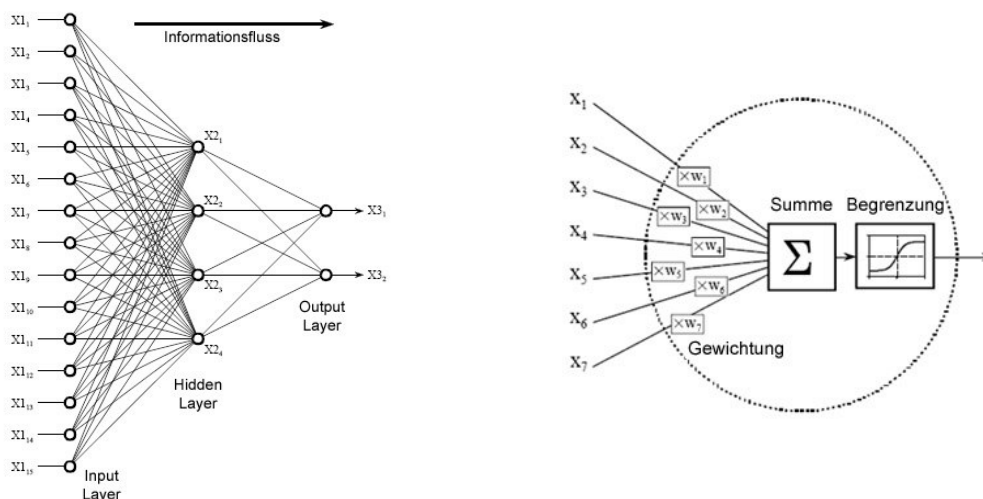


Abb. 4.8: Aufbau eines Neuronales Netzes mit 3 Schichten bzw. einer einzelnen Unit

Andere Arten neuronaler Netze haben etwa Feedbackpfade von Output- zu Input-Schicht oder eine ringförmige Architektur. Die Eingabe für ein mehrschichtiges Netz könnten z.B. Helligkeitswerte von Bildpunkten sein und seine Ausgabe ein Zahlenwert für die Erkennung eines bestimmten Musters. Die Initialisierung eines neuronalen Netzes erfolgt oft mit zufälligen Werten für die Gewichtungen der Verbindungen. Beim Training des Netzes mit Beispieldaten wird der Klassifikationsfehler vom Output in Richtung der Inputschicht zurückgerechnet (*back-propagation*). Dabei werden die Gewichte entsprechend ihres Anteils an einem richtigen oder falschen Ergebnis verstärkt oder abgeschwächt. Das neuronale Netz findet auf diese Weise eine interne Repräsentation für den Informationsgehalt der Daten.

4.6.2 PCA und ICA

Principal Component Analysis (PCA) und *Independent Component Analysis (ICA)* sind Methoden, die Komplexität eines Merkmalsraumes durch die Auffindung von orthogonalen Basisfunktionen (PCA) bzw. von entropiemaximierenden Basisfunktionen (ICA) zu reduzieren. Dadurch eignen sich diese Algorithmen zur Darstellung von Merkmalen mit möglichst wenigen relevanten Parametern. Die Grundannahme ist, dass sich ein Signal aus einer linearen Kombination verschiedener Quellen zusammensetzt. Die PCA liefert durch die *Singulärwertzerlegung* Komponentenfunktionen, das Originalsignal kann durch Multiplikation der Komponenten mit einer Matrix aus Eigenvektoren hergestellt bzw. durch die inverse Matrix in die Komponenten zerlegt werden. Die ICA lockert die Bedingung der Orthogonalität. Sie eignet sich zur Separation unbekannter Signalquellen (*blind source separation*) in statistisch unabhängige Komponenten und wird erfolgreich zur Artefaktkorrektur in Biosignalen eingesetzt, wo etwa der Gehalt des EOGs aus dem EEG bei bestmöglicher Beibehaltung der EEG-Wellen entfernt werden soll. Bei der ICA werden zur Separation von N Quellen mindestens N Signalableitungen benötigt, die Aufteilung in Komponenten erfolgt nach räumlichen Kriterien [MAK00].

4.6.3 SVMs

Support Vector Machines (SVMs) eignen sich für eine binäre Klassifizierung eines Eingabevektors. Das Ergebnis der Klassifizierung ist positiv (enthalten) oder negativ (nicht enthalten). Durch ein Training mit Datensätzen erfolgt eine Aufspaltung des Merkmalsraumes in zwei Hälften, wobei eine Hyperebene durch die Trainingsdaten gelegt wird, die eine maximale Distanz zu positiven und negativen Ergebnissen garantiert (*Rand, Margin*). Bei der Präsentation eines neuen Vektors fällt dieser in eine Hälfte und wird entsprechend klassifiziert. Im Unterschied zu Neuronalen Netzen finden SVMs immer ein globales Minimum [BUR98].

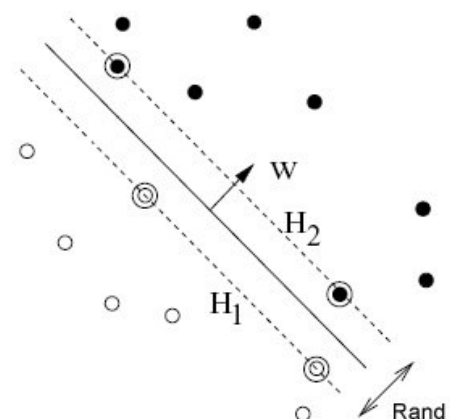


Abb. 4.9: Klassifikation mit Support-Vektoren

4.6.4 Merkmalsextraktion in der Bildanalyse

Bei der Analyse von Kamera- und Videobildern hat man es mit einer sehr großen Datenmenge (Helligkeits- und Farbwerte) und oft mit komplizierten Klassifikationsproblemen zu tun (etwa bei der Gesichts- oder Mimikerkennung). In diesem Bereich hat es in den letzten Jahren intensive Forschungen und Entwicklungen gegeben, hier kann nur ein kleiner Überblick über einige Techniken gegeben werden:

4.6.4.1 Template Matching

Beim Template Matching wird ein Vergleich von Bildintensitätswerten mit einer Vorlage durchgeführt. Meist wird ein kleiner Bildausschnitt mit dem gewünschten Merkmal über das Bild bewegt und die Korrelation mit dem Bildinhalt berechnet. Anhand des Ergebnisses wird entschieden, ob das Merkmal im Bild vorhanden ist.

4.6.4.2 Haar-Classifier

Bei der Klassifikation mittels *Haar-Features* wird ein Bild durch die Übereinstimmungen mit einer Reihe von einfachen Merkmalen (*weak-classifiers*) analysiert, welche bestimmte Eigenschaften hervorheben (siehe Abb. 4.10). Diese Muster werden in verschiedenen Skalierungsstufen und Positionen mit dem Bild korreliert und entsprechend den Vorgaben durch ein Trainingsset von Bildern gewichtet. Dadurch reagieren die einzelnen Elemente auf verschiedene Eigenschaften des Bildes, die für eine Erkennung der relevanten Form wichtig sind. Durch den Vorgang des *boosting* werden die vielen einzelnen Ergebnisse in einer gewichteten Summe zu einem stabilen Klassifikator zusammengeführt. Durch die Kaskadierung mehrere Stufen (*Haar-classifier-cascade*) können sehr unwahrscheinliche Fälle früh ausgeschlossen werden und der Algorithmus wird wesentlich beschleunigt [BRA05].

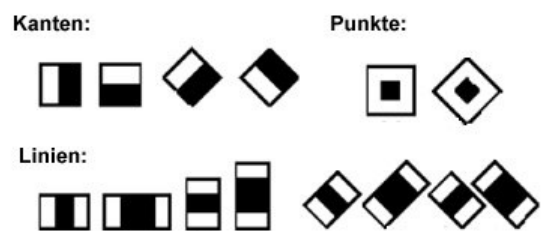


Abb. 4.10: Einzelne Merkmale eines Haar-Classifiers

4.6.4.3 Lucas-Kanade Algorithmus

Der Lucas-Kanade Algorithmus dient zur Detektion und Verfolgung von Bewegungen in aufeinander folgenden Bildern (*optical flow-algorithm*). Er basiert auf der Methode des *Image Alignment*, wo die Entsprechung einer Stelle im darauf folgenden Bild gesucht wird. Durch Translation und Skalierung (affine Transformationen) eines Bildausschnittes (z.B. 7 x 7 Pixel) wird nach einer Lösung für das Minimierungsproblem in der Umgebung der letzten Position gesucht. Meist wird dafür die Methode des steilsten Abstieges verwendet (*steepest descent*). Das Ziel des LK-Algorithmus ist es, die Summe der quadrierten Fehler zwischen dem originalen Bildausschnitt und dem projizierten Ausschnitt im neuen Bild zu minimieren. Der Algorithmus konvergiert gegen ein lokales Optimum und liefert die Koordinaten der neuen Position zurück [BAK02].

4.6.4.4 Active Appearance Models

Active Shape- und *Active Appearance Models* gehören zu den modellbasierten Klassifikationsmethoden für die Bildanalyse. Modelle können auch über den semantischen Gehalt eines Bildes eine Aussage treffen (etwa die Augen- und die Mundpartie eines Gesichtes unterscheiden). Für die Modellbildung werden in Trainingsbildern zusammengehörende Strukturen mit relevanten Informationen versehen (*labelling of landmarks*). Bei der Klassifizierung eines neuen Bildes wird versucht, durch die Veränderung der Modellparameter einen möglichst geringen Unterschied zum Zielbild zu erreichen und die Strukturen dadurch aufzufinden. Die Modelle sind also verformbar und lassen neue Interpretationen zu, durch das Training erhält das Modell Informationen über sinnvolle Variationen der Modellparameter. Um eine möglichst einfache Repräsentation für erlaubte Variationen zu finden, werden aus Daten der verschiedenen Trainings-Bilder durch eine Eigenwertzerlegung die orthogonalen Basisvektoren ermittelt (siehe *PCA* 5.6.2). Bei *Active-Shape* Modellen wird nur die Form bzw. Position von Koordinaten variiert, während *Active-Appearance* Modelle auch die Helligkeits- oder Farbwerte der hinterlegenden Textur einbeziehen. Modellbasierte Methoden beruhen auf dem *Top-Down*-Prinzip, das sich grundlegend von *Bottom-Up* Strategien wie der *Haar*-Klassifikation unterscheidet.

4.7 Softwarepakete zur Biosignalverarbeitung

Das Angebot an Software für die Verarbeitung von Biosignalen erstreckt sich von einfachen Feedback-Spielen für ein bis zwei Signalparameter bis zu komplexen Applikationen für die Analyse von Vielkanal-EEG-Ableitungen. Der folgende Überblick zeigt exemplarisch einige kommerzielle und nichtkommerzielle Lösungen.

4.7.1 Matlab/Simulink

Die *Matlab*-Applikation der Firma *Mathworks* ist durch den großen Umfang an Funktionen und durch die universelle Einsetzbarkeit zu einem Standard in der wissenschaftlichen Forschung geworden. *Matlab* erlaubt die Programmierung eigener Funktionen und Applikationen sowie die Simulation von dynamischen Systemen. Zur Analyse von Biosignalen eignet sich im Besonderen die *Signal Processing Toolbox* mit vielen Funktionen zur Erzeugung von Filtern oder zur Durchführung von Spektralanalyse oder Singulärwertzerlegung. Die *Simulink*-Umgebung ist zur Echtzeitauswertung von Signalen einsetzbar [MAT06].

Durch die internationale Forschungsgemeinschaft werden laufend neue Funktionsbibliotheken erstellt, die spezielle Probleme der Biosignalverarbeitung lösen und teilweise kostenlos bezogen werden können. Ein Beispiel ist die *Matlab-EEG-Toolbox BIOSIG* von Alois Schögl [SCL02].

4.7.1.1 EEGLab

EEGLab ist eine *Matlab*-basierte Analysesoftware für Mehrkanal-Biosignal-Daten, die als OpenSource-Applikation verfügbar und erweiterbar ist. *EEGLab* verfügt über 300 Funktionen zur Datenanalyse und Merkmalsextraktion, die Software eignet sich u.a. für die epochenbasierte *ERP*-Mittelung, zur Artefaktentfernung mittels *ICA*-Verfahren, für das Mapping und die Darstellung von *QEEG*-Daten und zur Lokalisation von Dipolen mit Bezug zu anatomischen *MRI*-Bildern. *EEGLab* besitzt Import-Funktionen für alle gängigen Datenformate, einschließlich *EDF* und *BDF*. Die hauptsächlichen Einsatzgebiete für *EEGLab* sind die Analyse von EEG- und MEG-Daten, prinzipiell eignet sich die Software auch zur Verarbeitung anderer elektrophysiologischer Messwerte [DEL06].

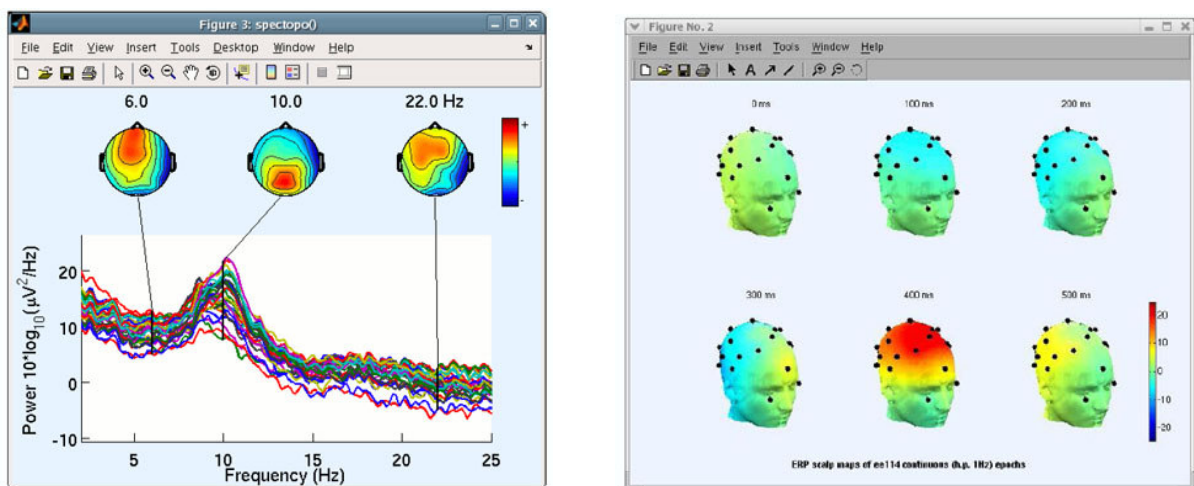


Abb. 4.11: Analyse von EEG-Daten mit EEGLab

4.7.2 Scilab und Octave

Scilab (<http://www.scilab.org>) und *Octave* (<http://www.gnu.org/software/octave>) sind OpenSource Projekte für die mathematische Analyse, die einen großen Teil der *Matlab*-Funktionalität kostenfrei zur Verfügung stellen. *Scilab* bietet auch einen blockorientierten Simulator, der an *Simulink* erinnert. In *Scilab* bzw. *Octave* sind hunderte *Matlab*-Funktionen entweder voll kompatibel oder in geringfügig anderer Schreibweise realisiert. Dennoch ist eine Verwendung komplexer Applikationen wie etwa *EEGLab* mit den OpenSource Paketen meist nicht ohne viele Änderungen möglich.

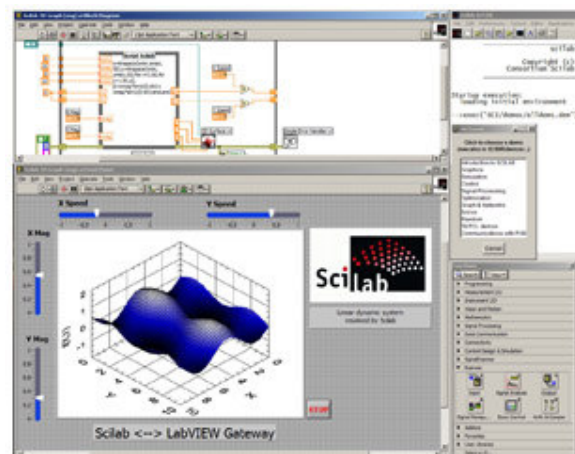


Abb. 4.12: Scilab - eine freie Alternative zu Matlab

4.7.3 Tempo

Tempo (Topographic EEG Mapping Program) ist eine OpenSource Software zur 3D-Visualisierung von EEG-Daten von Aleksandar B. Samardzic. Tempo kann das EDF-Datenformat lesen und eine Animationssequenz von topographischen Aktivitätsniveaus erstellen. Die Verwendung eines 3D-Kopfmodells aus MRI-Daten ist möglich. Die 3D-Ansicht kann beliebig rotiert werden [SAM05]

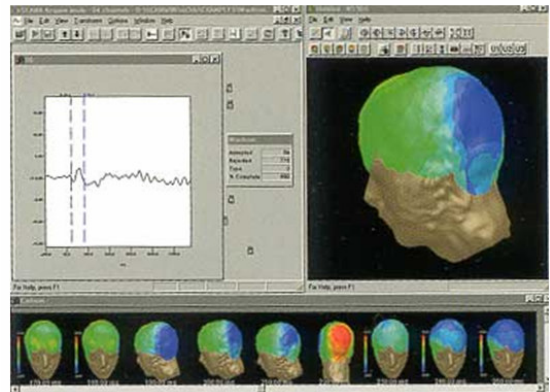


Abb. 4.13: 3D-Darstellung von EEG-Daten mit TEMPO

4.7.4 BioExplorer

BioExplorer ist eine kommerzielle Software für Bio- und Neurofeedback der Firma *Cyberevolution* (<http://www.cyberevolution.com>). Die grafische Benutzeroberfläche erlaubt eine einfache Erstellung von Signalverarbeitungskonfigurationen. Im Unterschied zu den bisher vorgestellten Applikationen liegt eine wichtige Eigenschaft von *BioExplorer* in der sofortigen Rückmeldung von Signalparametern in Echtzeit. Die Software bietet zahlreiche optoakustische Feedbackmöglichkeiten wie Multimedia-Player, Midi-Ausgabe, Oszilloskop und Balkendiagramme oder die 3D-Darstellung der Spektralanalyse. *BioExplorer* unterstützt EEG- und Biosignalverstärker aus dem preisgünstigen Marktsegment. Die Software kostet derzeit etwa 420 Euro.

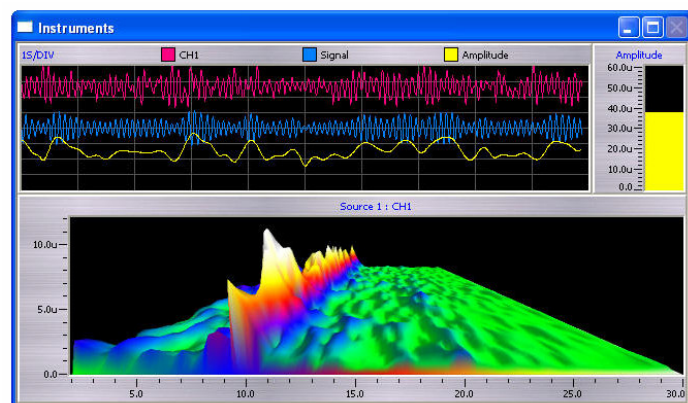


Abb. 4.14: Frequenzdarstellung mit BioExplorer

4.7.5 OpenEEG-Software

Im Rahmen des *OpenEEG*-Projektes entstanden in den letzten Jahren mehrere Applikationen, die verschiedene Anwendungen aus der Biosignalanalyse ermöglichen und großteils als OpenSource frei verfügbar sind. Neben den hier vorgestellten Programmen *BWView*, *FidView* und *BioEra* wurden etwa ein experimentelles BCI-System (*ABI* von Pedro Ortega, <http://www.dcc.uchile.cl/~peortega>) und ein Feedback-Spiel realisiert (*Brainathlon* von Amy Palke, <http://www.webkitchen.com/brainathlon>)

4.7.5.1 FidView - Filter Designer



Abb. 4.15: FidView - Filter Designer von Jim Peters

FidView ist eine von Jim Peters programmierte Software zur einfachen Erstellung digitaler Filter. *FidView* baut auf der OpenSource-Software *mkfilter* von Toni Fisher auf, und ist ebenfalls als OpenSource verfügbar. *FidView* erlaubt die Konfiguration verschiedener Arten von Filtern (*Bessel*, *Butterworth*, *Chebyshev*, *BandStopp*, *IIR*). Frequenz- und Impulsantwort werden graphisch dargestellt. Ein nützliches Feature ist die Bereitstellung eines C-Sourcecodes für die Filterfunktion, der generierte Filter kann somit leicht in eigene Programme übernommen werden. *FidView* verwendet Funktionen der *SDL*-Bibliothek und ist auf Windows und Linux lauffähig [PET03].

4.7.5.2 BWView - Wavelet EEG-Viewer

BwView ist eine OpenSource-Applikation von Jim Peters zur Analyse von Mehrkanal-Biosignaldaten mittels Wavelets. Der Algorithmus basiert auf einer FFT-beschleunigten Faltung mit wählbaren Filterkernen. Jedes Wavelet wird dabei durch eine komplexe Funktion gebildet und durch ein *Blackman*-Fenster begrenzt. Die Skalierung des Wavelets bestimmt die Sensitivität für entsprechende Wellenlängen. Durch Tastenkombinationen kann im Zeit- oder Frequenzbereich navigiert werden, verschiedene Farbcodierungen und Darstellungsvarianten sind verfügbar. *BwView* verwendet Funktionen der *SDL*-Bibliothek und läuft unter Linux und Windows. Der Source-Code ist frei verfügbar [PET02].

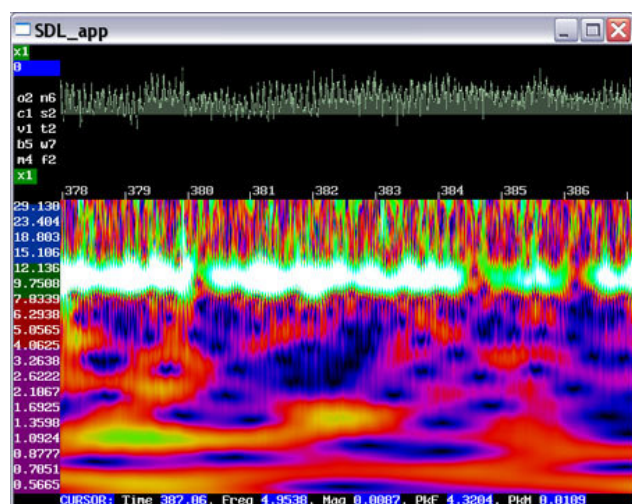


Abb. 4.16: Wavelet-Analyse von EEG-Daten mit BwView

4.7.5.3 BioEra

BioEra (<http://www.bioera.net>) ist eine Biofeedback-Applikation von Jarek Foltynski, die in *JAVA* erstellt wurde und somit weitgehend plattformunabhängig einsetzbar ist. *BioEra* wurde im Rahmen des *OpenEEG*-Projektes entwickelt, und ist als Lite-Version weiterhin kostenfrei verfügbar. Die Vollversion, die laufend erweitert wird und auch auf *PDA*-Computern läuft, ist kostenpflichtig (320 Euro). *BioEra* erlaubt wie *BioExplorer* die grafische Kombination von Funktionsblöcken und bietet viele Elemente für die Signalanalyse und für das optoakustische Feedback in Echtzeit.

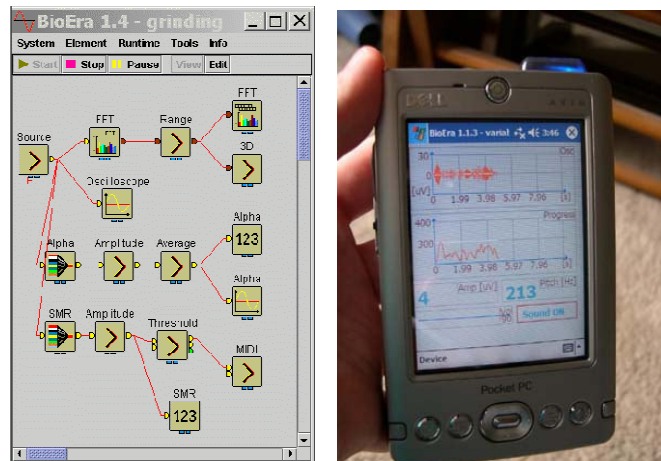


Abb. 4.17: BioEra Software für Biofeedback, Einsatz am PDA

4.7.6 Resumee Biosignal-Software

Die derzeitig verfügbare Software für die Biosignalanalyse ist zum Teil sehr ausge-reift und bietet eine große Vielfalt an Funktionen. Kommerzielle Produkte, die oft ge-meinsam mit Biosignalverstärkern angeboten werden, eignen sich zur professionel-len Auswertung von Vielkanal-Ableitungen und für den Einsatz im klinischen Bereich. Dennoch bieten auch Alternativen aus der OpenSource-Community wie zB. *EEGLab* überraschend viele Funktionen und sind vor allem für die wissenschaftliche For-schung gut geeignet. Die Auswahl an universellen Programmen für die Echtzeitaus-wertung von Signalen und das Biofeedback ist hingegen nicht so groß, im Open-Source-Bereich stehen hier nur wenige Applikationen zur Verfügung.

Die Konzeption und Realisierung einer eigenen Software für die Biosignalanalyse mit speziellen Funktionen für die Unterstützung behinderter Menschen erscheint unter diesen Gesichtspunkten möglich und sinnvoll. Bestehende Applikation wie *BioExplo-er* oder *BioEra* dienen hierbei als gute Beispiele für eine mögliche Implementierung.

5. *BrainBay*: Entwurf der Applikation

Im folgenden Abschnitt werden Anforderungen und gewünschte Funktionalitäten für die zu erstellende Softwareapplikation zusammengefasst, um daraus den strukturellen Aufbau und das Design für die Implementierung abzuleiten. Die grundlegenden Eigenschaften für eine universelle Anwendbarkeit ergeben sich aus den Erhebungen der vorangegangenen Kapitel. Die hier beschriebenen Anforderungen und Abläufe stellen Rahmenbedingungen für die Implementierung dar und definieren die Eigenschaften der zu verwendenden Objektklassen. Der Name der Applikation, *BrainBay*, versinnbildlicht eine virtuelle Verlängerung des zentralen Nervensystems: Das Programm dient als eine frei konfigurierbare Schnittstelle für die Interpretation der gemessenen Biosignale. Grundsätzliches Ziel dieser Software ist die Erhebung, Transformation und Verwertung von Biosignalen unterschiedlichen Ursprungs:

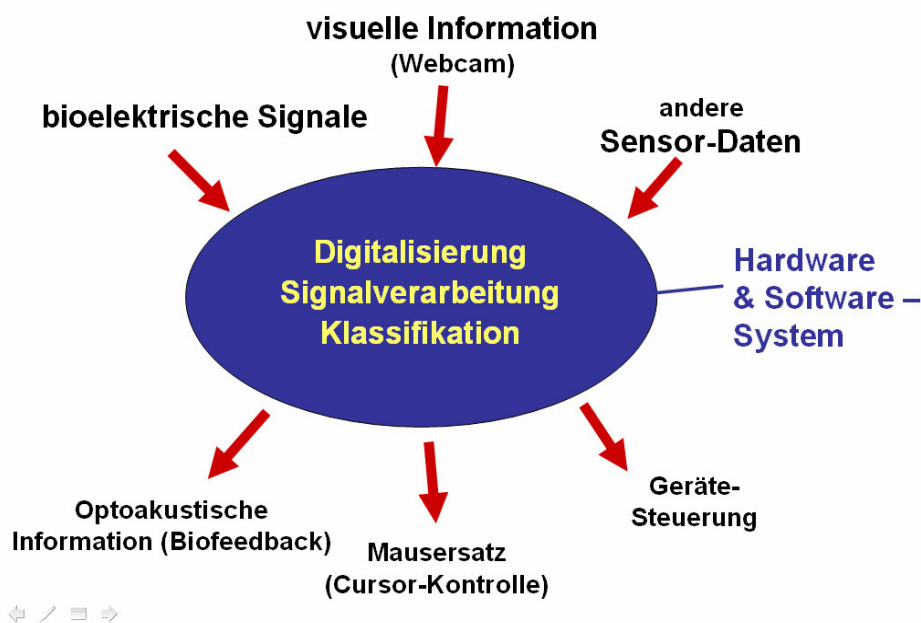


Abb. 5.1: Funktion und prinzipieller Aufbau des geplanten Datenverarbeitungssystems

5.1 Anforderungskatalog

Die erwünschten Eigenschaften und Anforderungen an eine universelle Applikation zur Verarbeitung von Biosignalen mit Anwendungsmöglichkeiten als Human Computer Interface lassen sich wie folgt zusammenfassen:

- kostengünstige Realisierbarkeit und Reproduzierbarkeit des Gesamtsystems
 - Verwendung freier Softwarebibliotheken
 - Anwendbarkeit einer weitläufig verfügbaren Entwicklungsumgebung
 - Kompatibilität mit Verstärkermodellen des *OpenEEG*-Projektes (*ModularEEG*, *MonolithEEG*)
 - Vermeidung lizenzierter Quellcode-Fragmente
 - Veröffentlichung des gesamten Quellcodes als Public Domain (GNU-Lizenz)

- Adaptierbarkeit und intuitive Verwendbarkeit
 - Möglichkeit zur individuellen Konfiguration der Signalverarbeitung
 - Einsatz von modularen Einheiten zur Herstellung von Konfigurationen
 - Basisfunktionen für die Transformation und Kombination von Signalen
 - grafische Benutzeroberfläche für das Konfigurationsdesign
 - klare Zuordnung von Optionen und Parametern zu den funktionalen Einheiten

- Echtzeitverarbeitung und Echtzeitmodifikation von Einstellungen
 - schnelle Verarbeitung durch hardwarenahe Programmierung
 - Ausnützung von Hardwarebeschleunigung (z.B. für Grafikfunktionen)
 - Hinzufügen, Entfernen und Justieren von Verarbeitungselementen während der Laufzeit der Signalverarbeitung

- Offenheit des Laufzeitsystems, Interoperabilität
 - Unterstützung gängiger Dateiformate zur Biosignalarchivierung durch Import- und Exportfunktionen (z.B. *EDF*- oder *CSV*-Dateien)
 - Netzwerkfähigkeit, Telemetrie von Biosignalen, Monitoring,
 - Online- und Offlineanalyse der Daten

- Verwendbarkeit als Biofeedback-Software
 - Möglichkeiten zur visuellen Darstellung von Messwerten (Skalen, Balkendarstellungen, Zähler, ..)
 - Verwendbarkeit für wichtige Biosignalklassen wie EMG, EKG, EEG, EOG, EDA, Temperatur usw.
 - Multimediafähigkeit (Klänge, Musik oder Filme als Feedbackoptionen)
 - Möglichkeiten zur Therapiefortschrittskontrolle

- Multimodalität und Anwendbarkeit als Human Computer Interface
 - Verarbeitung bioelektrischer und visueller Signalquellen (Webcams)
 - Algorithmen zur Merkmalsextraktion aus Videoquellen
 - Erzeugung von Interaktionskommandos zur Computerverwendung (Steuerung der Mausfunktionen, Keyboard-Inputs)
 - Automatischer Start von Konfigurationen und Hintergrundbetrieb
 - Kombinierbarkeit mit bestehenden Lösungen zur Interaktions- und Kommunikationsunterstützung wie Bildschirmtastaturen und Umgebungssteuerungen

- leichte Erweiterbarkeit durch neue Funktionen
 - prinzipielles Rahmenwerk an Klassen und Methoden
 - Dokumentation der nötigen Schritte zur Implementierung neuer Funktionen

Eine fix vorgegebene Hierarchie für die funktionalen Einheiten zur Gewinnung der Rohsignale, Verarbeitung der Daten und Ausgabe der Ergebnisse wäre nicht zielführend. Um all diese Anforderungen innerhalb einer Applikation realisieren zu können, sollte schon die grundlegende Struktur, auf welche die einzelnen Verarbeitungsschritte aufbauen, dynamisch sein.

5.2 Überlegungen zur funktionalen Struktur

Zur Realisierung der Anforderungen bietet sich ein grafisches Design von Signalverarbeitungs-konfigurationen an, das einen problemorientierten Aufbau von Signalquellen, Transformationsfunktionen und Signalenken ermöglicht. Ein gutes Beispiel für einen derartigen Aufbau bietet das *Simulink*-Paket, welches Teil der *Matlab*-Applikation ist:

Hier können Symbole für Funktionen zur Erzeugung, Verarbeitung und Darstellung von Signalen aus einem großen Set gewählt und durch eine grafische Benutzeroberfläche kombiniert werden. Durch das Verbinden dieser Symbole werden Signalwege festgelegt und die aktuelle Konfiguration bleibt übersichtlich und intuitiv erfassbar. Viele der im Anforderungskatalog beschriebenen Aufgaben könnten mithilfe der Simulink-Umgebung realisiert werden, dennoch verbleiben (wie auch bei anderen kommerziell erhältlichen Produkten) einige Nachteile, durch die eine Eigenentwicklung sinnvoll erscheint. Dazu zählen relativ hohe Lizenzgebühren, mangelnde Verwendbarkeit im HCI- oder Biofeedbackbereich, Inflexibilität bezüglich der Integration hardwarenaher Systemerweiterungen und die niedrigere Ausführungsgeschwindigkeit.

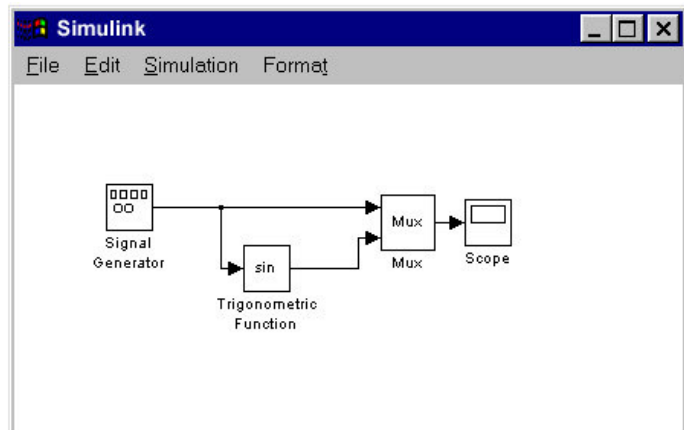


Abb. 5.2: Grafisches Design der Signalverarbeitung in Simulink

5.2.1 Konzept für den Aufbau von Signalverarbeitungs-konfigurationen

Es erweist sich als zielführend, die verschiedenen Funktionen in frei kombinierbare Einheiten zu kapseln, die bei der Erstellung einer konkreten Verarbeitungs-konfiguration durch den Benutzer / die Benutzerin eingefügt, verbunden und abgestimmt werden können. Diese Einheiten werden im Folgenden als **Konfigurationselemente** (kurz: *Elemente*) bezeichnet. Bei einer geeigneten Form der Implementierung kann die Anordnung sowie Parametrisierung der Elemente auch während der Ausführung der Signalverarbeitung durchgeführt werden, sodass die Auswirkungen der Änderungen sofort sichtbar werden und eine Einstellung der gewünschten Systemeigenschaften erheblich erleichtert wird. Auf diese Weise hergestellte Konfigurationen können gespeichert, geladen und auf Wunsch auch automatisch beim Systemstart ausgeführt werden. Aufgrund ihrer Eigenschaften lassen sich mögliche Konfigurationselemente in drei Klassen untergliedern:

5.2.1.1 Signalquellen

Durch diese Art von Elementen können Signale erzeugt und zur weiteren Verarbeitung in der Konfiguration zur Verfügung gestellt werden. Diese Signale könnten im Element selbst synthetisiert werden (ein Signal-Generator) oder aus anderen Quellen stammen, die durch das Element für die Applikation zugänglich gemacht werden (zB. die Messwerte eines Biosignalverstärkers oder ein Datenarchiv).

Eine Signalquelle hat zur Weiterleitung der Daten mindestens eine Verbindungsstelle, an welche weitere Elemente angeschlossen werden können (*Output-Port*).

5.2.1.2 Verarbeitungselemente

Diese Elemente werden zur Ausführung mathematischer Operationen benötigt, welche die Transformation, Kombination oder Interpretation eingehender Signalwerte ermöglichen. Ein Verarbeitungselement braucht mindestens eine eingehende und eine ausgehende Verbindungsstelle (*Input-* und *Output-Ports*). Nützliche Verarbeitungselemente für die Signalbehandlung sind digitale Filter, Mittelungsverfahren, Mustererkennungsalgorithmen, Korrelationsberechnung oder Fourier-Transformation.

5.2.1.3 Signalsenken

Diese Elemente brauchen einen oder mehrere Input-Ports. Sie dienen einer bestimmten Art von Wertschöpfung aus den gewonnenen Signalwerten. So könnte der aktuelle Wert am Bildschirm als Zahl oder Balkengrafik angezeigt, über Netzwerk gesendet oder zur Steuerung eines Kluges, Filmes oder externen Gerätes verwendet werden. Signalsenken könnten effektive Funktionen für eine Maussteuerung oder zur Erzeugung anderer Steuersignale für eine alternative Benutzerschnittstelle bereitstellen.

5.2.2 Strukturen für die Verwendung der Konfigurationselemente

Jedes Element sollte nach dem Einfügen in eine Konfiguration als grafisches Symbol dargestellt werden. Die Input- und Outputverbindungen müssen als Interaktionspunkte gekennzeichnet sein. Durch Anklicken eines Output-Ports wird eine Verbindung zu einem Input-Port eines anderen Elementes gezogen. Von einem Output-Port können mehrere Verbindungen zu anderen Elementen führen. Eine Verbindung wird grafisch als Linie zwischen den Ports dargestellt. Sobald eine Verbindung erstellt wurde, fließt ein Signal vom Output-Port des Quellelements zum Input-Port des Zielelements. Die Signalwerte werden in Form von Gleitkommazahlen mit einer einstellbaren Übertragungsrate weitergeleitet. Erlaubt ein Element die Modifikation von Optionen, sollte für dieses Element ein eigener Benutzerdialog zur Verfügung stehen. Die Art des Benutzerdialoges wird durch die Klasse des Elementes vorgegeben, ein Dialog zeigt immer die aktuellen Werte für ein bestimmtes Element. Verfügt ein Element über grafische Ausgabemöglichkeiten, sollten diese in einem eigenen, frei positionierbaren Fenster dargestellt werden.

5.2.3 Interne Organisation der Konfigurationselemente

Die Elemente müssen intern durch eine Speicherstruktur in Form einer Liste oder eines Feldes organisiert werden. Da die Ausführung der Verarbeitung sequentiell erfolgt und mehrere Signalquellen, Transformationen und Senken in einer Konfiguration existieren können, ist es wichtig, dass die Reihenfolge der Verarbeitung der Anordnung der Elemente in der Konfiguration entspricht. (Ein Quellelement, das Daten an ein Zielelement sendet, sollte vor dem Zielelement aktualisiert werden.) Die Speicherstruktur für die Elemente muss also beim Einfügen neuer Elemente oder beim Ziehen neuer Verbindungen auf Konsistenz überprüft und eventuell neu sortiert werden, damit die richtige Ausführungsreihenfolge gewährleistet bleibt.

5.3 Konzepte für den operativen Ablauf der Signalverarbeitung

Die durch eine bestimmte Konfiguration von Elementen gegebene Form der Signalverarbeitung erfolgt durch eine Abarbeitung der Elemente in der gegebenen Reihenfolge. Die Weiterleitung und Aktualisierung der Werte soll periodisch in einem einstellbaren Intervall (der Samplerate) erfolgen. Werden die Rohdaten direkt von einer Biosignal-Hardware empfangen, wird die Samplerate durch die Geschwindigkeit der Digitalisierung vorgegeben. Für eine originalgetreue Wiedergabe einer Archivdatei sollte das Intervall der Samplerate der Archivdatei entsprechen. Die genaue Einhaltung der Samplefrequenz ist wichtig für die Echtzeitfähigkeit des Systems und für eine verzerrungsfreie Wiedergabe der Daten.

Zur einwandfreien Funktion dieses Ablaufes müssen folgende Ebenen in der Applikation existieren:

- ein Zeitmanagement, das dem Aufruf der Elemente zugrunde liegt
- eine bestimmte Methode pro Element, die zur Durchführung der Aktualisierung und Verarbeitung der aktuellen Werte aufgerufen wird

5.3.1 Möglichkeiten für das Zeitmanagement

Das Zeitmanagement ist, abgesehen von der Menüauswahl, die einzige statische Struktur der gesamten Applikation. Die Art und Reihenfolge der aufgerufenen Elemente wird durch die vom Benutzer / der Benutzerin erstellte Konfiguration vorgegeben. Aufgabe des Zeitmanagements ist die genaue Einhaltung der Intervalle für den Aufruf der Elemente. Zur Realisierung bietet sich die Verwendung des *Timer-Interrupts* an. (Durch den Timer-Interrupt wird eine frei programmierbare Funktion in bestimmten Intervallen vom Betriebssystem aufgerufen.) Die kleinste mögliche Intervalldauer beträgt eine Millisekunde. Dennoch kann nicht gewährleistet werden, dass der Aufruf tatsächlich exakt in diesem Intervall stattfindet, da die CPU unterschiedlich stark belastet wird und vor allem durch starke Beanspruchung durch andere Prozesse Latenzzeiten entstehen. Es ist also die Verwendung einer exakten Systemzeit nötig. In Windows-Systemen ist der genaueste zur Verfügung stehende Zeitgeber der so genannte *PerformanceCounter*. Dieser Wert kann zur Synchronisierung der Elementaufrufe mit der Samplefrequenz verwendet werden. Eine mögliche Vorgehensweise wäre, im Timer-Interrupt den PerformanceCounter abzufragen, und bei Erreichen oder Überschreiten des Intervalls der Samplefrequenz die Elemente in der richtigen Reihenfolge abzuarbeiten. Wird der Timer-Interrupt durch Latenz zu spät ausgelöst, muss dies durch kürzere Intervalle in den folgenden Aufrufen kompensiert werden.

5.3.2 Strategien für zeitkritische System- und Rechenaufgaben

Zeitkritische Aufgaben können nicht im Zuge der periodischen Aufrufe der Elemente ausgeführt werden, da eine kontinuierliche Abarbeitung der Konfiguration mit der eingestellten Geschwindigkeit darunter leiden würde. Deshalb müssen Operationen, die entweder zuviel Rechenzeit benötigen oder den Prozessor durch Warten auf ein Ereignis blockieren könnten, angemessen behandelt oder aus den periodisch aufgerufenen Routinen der Elemente ausgelagert werden. Eine mögliche Strategie wäre,

in dem zur Verfügung stehenden Zeitfenster nur einen Teil einer Berechnung durchzuführen oder nur auf das Eintreffen eines Ereignisses abzufragen (nicht zu warten). Eine andere Strategie ist die Auslagerung derartiger Prozesse in eigene *Threads*. Dadurch wird das Zeitmanagement vom Betriebssystem selbst übernommen. Im Thread kann die zeitintensive Aufgabe durchgeführt werden, und bei Fertigstellung wird das betreffende Element informiert und kann die Daten in Empfang nehmen. Die Verarbeitung von Kamerabildern der Webcam oder die Abfrage von Biosignalwerten über die serielle Schnittstelle sind zeitintensive Aufgaben, die besser in eigenen Threads zu realisieren sind.

5.4 Objektorientiertes Design

Die objektorientierte Design- und Programmieretechnik ist für die Erstellung der Applikation aus folgenden Gründen gut geeignet:

- die Elemente haben gemeinsame Attribute (zB. ihre Position am Bildschirm, die Beschriftung, die Strukturen für Ein- und Ausgabeports)
- die Elemente haben gemeinsame Methoden (zB. das Laden und Speichern von Parameterwerten, die Darstellung des Userdialoges, die Verarbeitung von Input- und Outputwerten)
- Elemente werden in mehrfacher Weise in die Design-Konfiguration eingefügt und benötigen separate Speicherbereiche für ihre lokalen Daten
- die Elemente können untereinander dynamisch verbunden werden, was die Art und Reihenfolge der Ausführung beeinflusst

Diese Eigenschaften der Programmarchitektur legen ein objektorientiertes Design nahe, wobei jedes Element eine Objektklasse implementiert, die von einer gemeinsamen Basisklasse abgeleitet ist. Die Basisklasse beinhaltet die gemeinsamen Attribute und Methoden, die allen Elementklassen vererbt werden. Zusätzlich werden virtuelle Methoden in der Basisklasse deklariert, die nur von denjenigen Elementen implementiert werden, die sich auch benötigen. (So wird etwa die Methode zur Darstellung eines Benutzerdialoges nur von jenen Elementen implementiert, die auch einen Benutzerdialog zur Verfügung stellen.)

Durch das C++ - Kommando *new* (*OB_CLASS_NAME*) wird eine neue Instanz der jeweiligen Objektklasse erstellt. Im Objekt-Konstruktor erfolgt die Initialisierung des Elementes mit Standardwerten. Die Methode *load()* dient zum Laden von Parameterwerten aus der Konfigurationsdatei usw.

5.4.1 Beziehung der Elementklassen zur Basisklasse

Die folgende Übersicht zeigt die Basisklasse und die Vererbung von Attributen und Methoden an die Elementklassen. Es werden dabei alle Attribute der Basisklasse ererbt (diese sind nicht extra für jede Elementklasse dargestellt), aber nur einige der virtuellen Methoden implementiert:

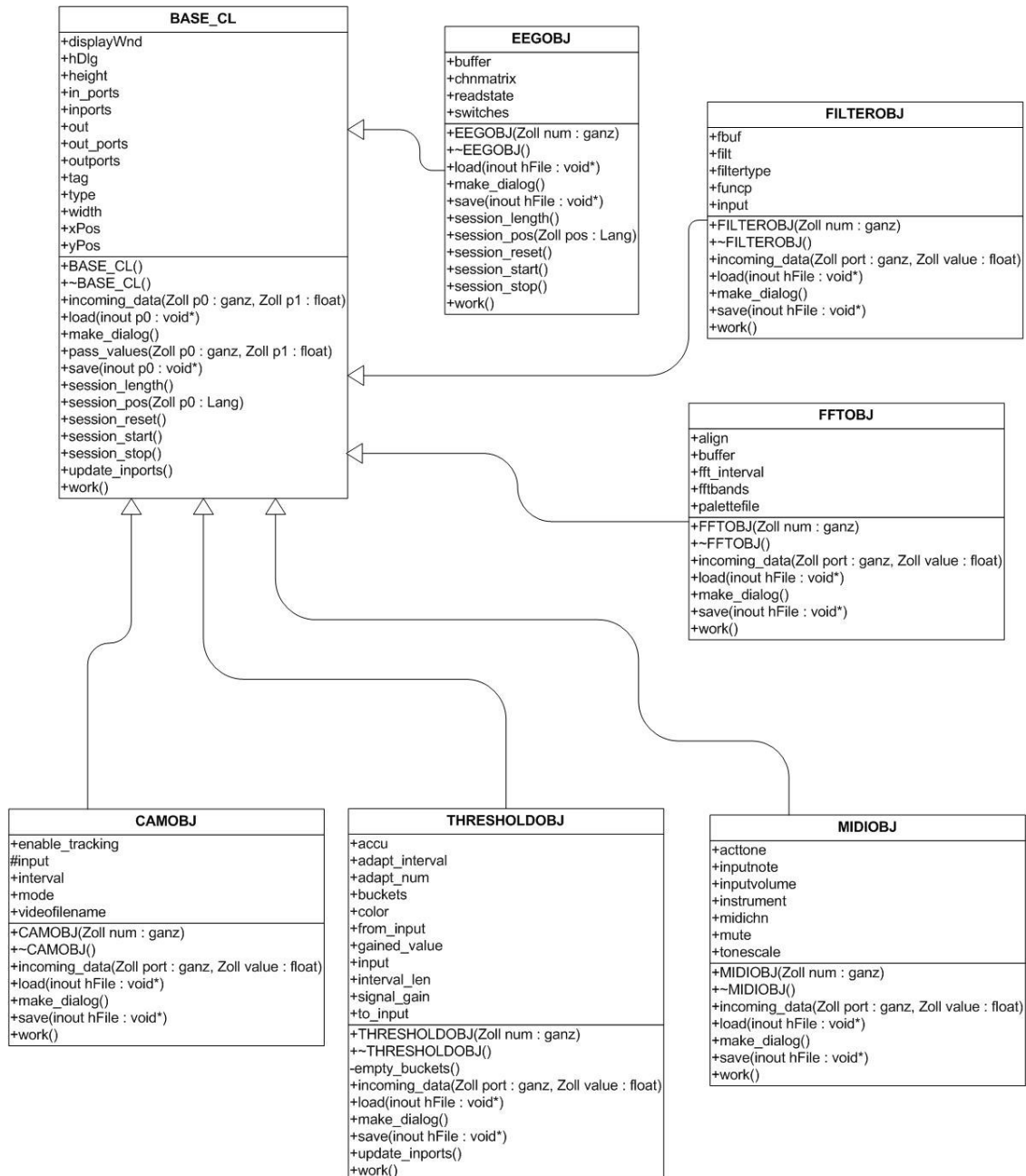


Abb. 5.3: Objektorientiertes Design, Bezug der Elemente zur Basisklasse (Auszug)

Die abgebildete Übersicht stellt nur die prinzipielle Struktur dar, sie zeigt lediglich einige wichtige Elementklassen. Für die Bereitstellung der gewünschten Funktionalität werden weitaus mehr Elementklassen benötigt. Ein Vorteil des objektorientierten Designs liegt auch in der leichteren Erweiterbarkeit durch neue Elementklassen, da ein entsprechendes Rahmenwerk durch die Basisklasse gegeben ist.

5.5 Gemeinsame Methoden für die Objektklassen

Die Konfigurationselemente stellen die funktionalen Bestandteile einer *BrainBay* Konfiguration dar. Als sinnvoll erweisen sich Elemente für Signalquellen, verarbeitende Elemente (Filter, mathematische Transformationen etc.) und Signalsenken. Jedes Konfigurationselement besitzt eine eigene Objektklasse, welche die objektspezifischen Methoden und Variablen beinhaltet. Die virtuellen Methoden der Basisklasse dienen folgenden Aufgaben:

- Empfang und die Weiterleitung von Werten
Methoden: *incoming_data()*; *work()*
- Starten und Stoppen der Verarbeitung
Methoden: *session_start()*; *session_stop()*;
- Positionierung in Archivdateien, Abfrage der Archivlänge
Methoden: *session_pos(int position)*
- Laden und Speichern von Element-Parametern
Methoden: *load(file)*; *save(file)*
- Weiterleitung von Werten an verbundene Elemente
Methode: *pass_values()*;

5.6 Konzepte für das User Interface

Als Strategie für die grafische Benutzeroberfläche eignet sich ein Hauptfenster mit einem großen Zeichenbereich für das Anlegen der Konfiguration. Das Menü des Hauptfensters enthält Funktionen zum Laden und Speichern des Designs, Möglichkeiten für das Einfügen von Elementen und den Aufruf von Dialogen (Optionen, Tools, Hilfe).

Durch das Einfügen eines Elementes in den Zeichenbereich des Hauptfensters wird ein Elementsymbol mit den verfügbaren *Input-* und *Output-Ports* dargestellt. Diese Symbole können grafisch angeordnet und verbunden werden. Eine Verbindung von zwei Elementen definiert einen Datenfluss von Output- zu Input-Port. Die Parameter der einzelnen Elemente können durch spezifische Benutzerdialoge eingestellt werden. Durch das Anklicken eines Elementsymbols wird der Benutzerdialog dargestellt und die aktuellen Einstellungen werden sichtbar und können modifiziert werden.

Elemente, die visuelle Darstellungen zur Verfügung stellen, haben separate Ausgabefenster, die frei am Bildschirm positioniert werden können. Die Positionen und Größen der Fenster werden mit der Designkonfiguration gespeichert, beim nächsten Laden der Konfiguration wird das Bildschirmlayout wiederhergestellt.

6. Implementierung der *BrainBay*-Software

Dieses Kapitel beschreibt die Entwicklungsumgebung für die Applikation und Details zur Realisierung der verschiedenen Softwaremodule. Wichtige Klassen und Funktionen werden vorgestellt und anhand eines Beispiels wird beschrieben, wie neue Elemente zur Applikation hinzugefügt werden können.

6.1 Systemvoraussetzungen und Entwicklungsumgebung

Die Applikation wurde ursprünglich auf einem 1,6 GHz Pentium Laptop Computer mit 256 MB RAM und dem Betriebssystem *Windows XP Service Pack 2* entwickelt. Der Compilervorgang wurde auch auf *Pentium II* Rechnern mit *Windows 2000* und *Windows 98* erfolgreich durchgeführt und eine Verwendung des Programms ist auch auf diesen Plattformen möglich.

Der Quellcode der Software wurde unter Verwendung der *Win32 API* für Systemaufrufe und grafische Benutzeroberfläche in der Programmiersprache C++ erstellt. Das *Microsoft Foundation Classes Framework (MFC)* wurde nicht verwendet. Die integrierte Entwicklungsumgebung *Microsoft Visual C++ 6.0* diente zum Editieren der Quelltexte, zum Design der Benutzerschnittstelle und zur Unterstützung des Compilier- und Link-Vorganges. Ein integrierter Debugger unterstützt beim Testen und bei der Fehlersuche.

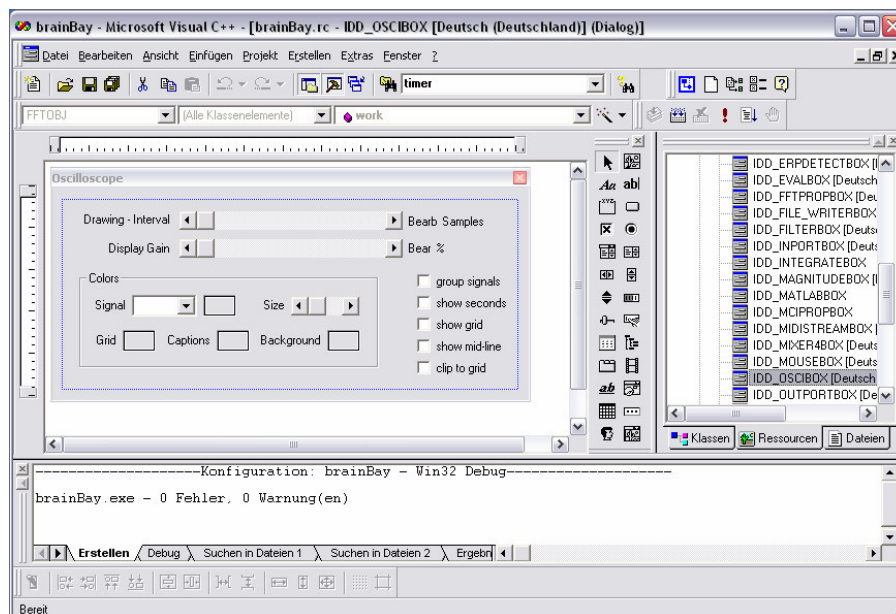


Abb. 5.4: Entwicklungsumgebung Microsoft-VisualC++, Dialog-Editor

Die Projekt-Datei *brainbay.dsw* wird von Visual Studio verwaltet und enthält Informationen über die relevanten Quellcode-Dateien (.cpp), Header-Dateien (.h) und Ressourcen-Komponenten wie Dialog-Boxen, Buttons und Menüs (.rc-file).

Als weitere mögliche Entwicklungsumgebung eignet sich der freie *Minimalist GNU for Windows Compiler (MinGW, <http://www.mingw.org>)*. *MinGW* ist eine Sammlung frei verfügbarer Windows-spezifischer Header-Files und Programmbibliotheken, die in

Verbindung mit dem *GNU-Compiler / Linker* eine Produktion von nativen Windows Applikationen erlaubt, ohne von lizenzpflichtigen Laufzeitbibliotheken abhängig zu sein. *MinGW* bietet allerdings keine Unterstützung für das Design der Benutzerschnittstelle.

Von Borland gibt es ebenfalls eine Entwicklungsumgebung, die mittlerweile kostenfrei verfügbar ist, den Freeware - *C++-Builder BCC5.5* (http://www.borland.com/products/downloads/download_cbuilder.html). *BrainBay* kann mit entsprechenden Änderungen in der Projektdatei mit *BCC5.5* übersetzt werden.

6.2 Verwendete Bibliotheken

6.2.1 Statische Bibliotheken

Da *BrainBay* Funktionen anderer Open-Source - Projekte verwendet, müssen die entsprechenden Bibliotheken für den Link-Vorgang der Applikation zur Verfügung stehen. In *VisualStudio* werden diese Bibliotheken unter dem Menüpunkt "*Project -> Settings -> Linker*" angegeben.

6.2.1.1 Standardbibliotheken (Teil der Visual Studio Umgebung) :

Die folgenden Standardbibliotheken beinhalten Systemfunktionen und Funktionen für die Nachrichtenbehandlung, Dateiverwaltung und Verwendung der Benutzerschnittstelle:

*kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib
shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib comctl32.lib
winmm.lib ,vfw.lib, opengl32.lib glu32.lib glaux.lib*

6.2.1.2 Open-Source Bibliotheken:

Die folgenden Open-Source Bibliotheken stellen Funktionen für hardwarenahe Multimediaprogrammierung, Auswertung mathematischer Terme und Bildverarbeitung zur Verfügung. Die zugehörigen *.lib* - Dateien müssen sich im Bibliotheksverzeichnis der Entwicklungsumgebung befinden (bei Visual Studio '*visual_studio/vc98/lib*') oder über die Pfadangabe für den Linker erreichbar sein:

SDL.lib, SDL_net.lib, SDL_sound.lib, modplug.lib :
Simple Direct Media Layer ([http:// www.libsdl.org](http://www.libsdl.org))

fidlib.lib : Jim Peter's Filter Library (<http://uazu.net/fidlib>)

matheval.lib : Gnu math-evaluator library
<http://www.gnu.org/software/libmatheval/manual/libmatheval.html>

cv.lib cvcam.lib cxcore.lib, highgui: Intel's Open Computer Vision Library *OpenCV*
<http://www.intel.com/research/mrl/research/opencv>

skinstyle.lib : SkinStyle- Win32 Skinning Library von John Roark
<http://www.codeproject.com/dialog/skinstyle.asp>

6.2.1.3 Kommerzielle Bibliotheken für die *Matlab*-Unterstützung

Diese Bibliotheken dienen zur Verwendung der *Matlab-Engine*, die einen Austausch von Daten zwischen *Matlab* und dem *Matlab*-Element von *Brainbay* ermöglicht. Sie sind Teil der *Matlab*-Distribution. In der Standardkonfiguration von *BrainBay* (ohne *Matlab*-Unterstützung werden sie nicht benötigt.

libeng.lib libmx.lib : Matlab engine (<http://www.mathworks.com>)

6.2.2 Laufzeitbibliotheken (Dynamic Link Libraries, DLLs)

Die Laufzeitbibliotheken beinhalten Funktionen der verwendeten Open-Source Komponenten und befinden sich im selben Verzeichnis wie die ausführbare Datei *brainbay.exe*. Sie werden beim Start der Applikation nachgeladen.

BrainBay verwendet die folgenden DLLs:

SDL.dll, SDL_net.dll, SDL_sound.dll: Funktionen des *Simple Direct Media Layer*
cv097.dll, cvcam097.dll, cxcore097.dll, highgui097.dll: Funktionen von *OpenCV*
matheval.dll : Funktionen des GNU math-evaluators

6.3 Prinzipien der Windows Programmierung

Brainbay ist eine ereignisgesteuerte Win32-Applikation (*event-driven*). Die Benutzerinteraktion wird durch Dialogfenster ermöglicht, die Umsetzung der Benutzerkommandos geschieht in den entsprechenden Fensterfunktionen (*window handler*). Die Koordination von Ereignissen wie Tastatur- oder Maus-Eingaben erfolgt in einer Windows-Umgebung durch die Verarbeitung von Nachrichten (*messages*). Eine Nachricht besteht aus einem *message identifier* (16-bit Integer Wert) und zwei Nachrichtenparameter (*message parameters, wParam* und *lParam*, jeweils 32-bit Integer Werte).

Tritt ein Ereignis auf (zB. der Benutzer / die Benutzerin drückt eine Taste, wählt ein Eingabefeld mit der Maus etc.) wird eine Nachricht mit entsprechenden Parametern vom System an die beteiligten Fenster gesendet. Zu bearbeitende Nachrichten werden vom System in einer Warteschlange (*message queue*) gehalten, und so bald wie möglich abgearbeitet. Jede Applikation ist dabei selbst für die Verteilung der Nachrichten an eigene Fensterkomponenten zuständig. Die Nachrichtenschleife im Hauptprogramm erledigt diese Aufgabe durch wiederholtes Abfragen der *message queue* und Aufrufe der Funktionen *TranslateMessage()* und *DispatchMessage()*:

```
MSG msg;
while (GetMessage(&msg, NULL, 0, 0))
{
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}
```

6.3.1 Nachrichtenverarbeitung in einer Windows-Applikation:

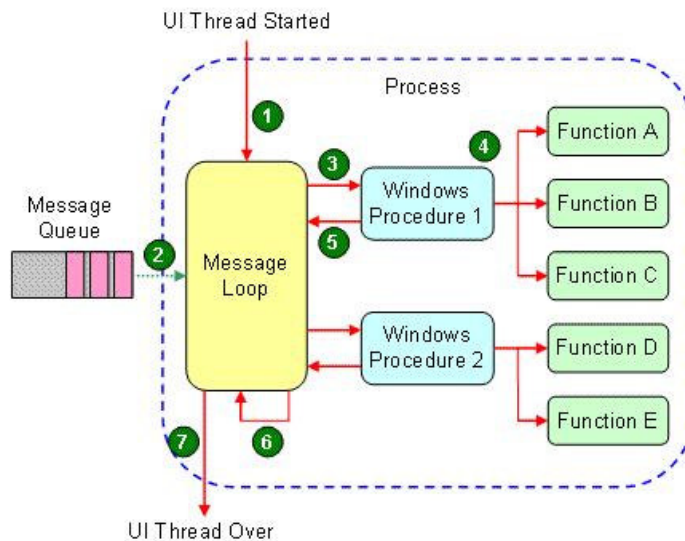


Abb. 5.5: Weiterleitung von Nachrichten in einer Windows-Applikation

Ein Prozess, der eine Nachrichtenschleife beinhaltet, wird *User-Interface Thread* genannt. Ein oder mehrere Fenster werden in diesem Prozess erzeugt und sind ihm zugeordnet. Die Nachrichtenbehandlungsfunktion eines Fensters wird von dem zugehörigen Prozess durch die Funktion *DispatchMessage()* aufgerufen. Es kann zwar auch jeder andere Prozess Nachrichten an das Fenster senden, diese werden aber in der Nachrichtenschleife des Prozesses abgelegt, der das Fenster erzeugt hat. Dies bewirkt, dass alle Nachrichten an ein Zielfenster synchronisiert werden und die Nachrichten in der richtigen Reihenfolge von der Behandlungsroutine abgearbeitet werden können.

Zur Erzeugung neuer Fenster muss eine Fensterklasse (*window class*) beim System durch die *WNDCLASSEX* - Struktur und die *RegisterClassEx(..)* Funktion angemeldet werden. Diese Struktur beinhaltet einen Zeiger zur Nachrichtenbehandlungsfunktion, einen Bezeichner für die Instanz (die ausführbare Datei), die Icons, die Art des Maus-Cursors, die Hintergrundfarbe für das Fenster und den Klassennamen.

Die *CreateWindow(..)* - Funktion erzeugt ein Fenster mit Bezug auf eine angemeldete Fensterklasse und verschiedene Fenstereigenschaften (*window styles*), gegebene Titelzeile und Fensterposition

6.3.2 Behandlung von Nachrichten: Der *Window-Handler*

Durch die *DispatchMessage()* Funktion erfolgt die Verteilung der Nachrichten an die betroffenen Fensterkomponenten. Jede Fensterkomponente benötigt dazu eine Nachrichtenbehandlungsroutine, den sog. *Window-Handler*. Nachfolgend ein Beispiel für die Nachrichtenbehandlung eines sehr einfachen Dialogfensters, das zwei Schaltflächen (*About* und *Exit*) beinhaltet:

```

LRESULT CALLBACK WndProc(
    HWND hWnd,           // window handle
    UINT message,       // message identifier
    WPARAM wParam,      // erster 32-bit Parameter
    LPARAM lParam)      // zweiter 32-bit Parameter
{
    int wmId, wmEvent;

    switch (message)
    {
    case WM_COMMAND:
        wmId    = LOWORD(wParam);
        switch (wmId)
        {
            case IDM_ABOUT:
                DialogBox(hInst, (LPCTSTR)IDD_ABOUTBOX,
                    hWnd, (DLGPROC)About);
                break;

            case IDM_EXIT:
                DestroyWindow(hWnd);
                break;

            default: return DefWindowProc(hWnd, message, wParam,
                lParam);
        }
        break;
        default: return DefWindowProc(hWnd, message, wParam,
            lParam);
    }
    return 0;
}

```

In diesem Beispiel wird nur die Nachricht *WM_COMMAND* behandelt. Eine Nachricht dieses Typs wird gesendet, wenn eine Benutzerinteraktion mit einem Dialogelement wie einem Button, einem Regler oder Listenfeld stattfindet. Die ersten 16-bit des Parameters *wParam* identifizieren das betreffende Dialogelement über eine Konstante. In diesem Fall wird ein neues Dialogfenster dargestellt, falls der Button *IDM_ABOUT* gedrückt wurde. Falls *IDM_EXIT* gedrückt wurde, wird das Fenster geschlossen.

Besitzt ein Fenster andere Kontrollelemente oder werden Zeichenfunktionen verwendet, sind eine Reihe von Nachrichten für diese Aufgaben zu behandeln. Hier nur ein paar Beispiele:

<i>WM_INITDIALOG:</i>	wird gesendet wenn das Dialogfenster erstellt wurde
<i>WM_CLOSE:</i>	beim Schließen des Dialogfensters
<i>WM_HSCROLL, WM_VSCROLL:</i>	bei Veränderung eines Reglers (Scrollbar)
<i>WM_MOUSEMOVE:</i>	wenn sich die Maus über das Fenster bewegt
<i>WM_LBUTTONDOWN:</i>	linker Mausknopf wurde gedrückt
<i>WM_LBUTTONUP:</i>	linker Mausknopf wurde losgelassen
<i>WM_RBUTTONDOWN:</i>	rechter Mausknopf wurde gedrückt
<i>WM_RBUTTONUP:</i>	rechter Mausknopf wurde losgelassen
<i>WM_DBLCLICK:</i>	Doppelklick in das Fenster oder Kontrollelement
<i>WM_PAINT:</i>	das Fenster wird gezeichnet
<i>WM_ACTIVATE:</i>	das Fenster wird aktiviert

Für eine Beschreibung weiterer Nachrichten und Fensterfunktionen sei auf das Microsoft Entwicklernetzwerk (*MSDN*) verwiesen.

6.4 Die Basisklasse für *BrainBay* - Elemente

Die Definition der Basisklasse erfolgt im Modul *base.h* :

```
class BASE_CL          // Name der Basisklasse
{
public:
    int type;           // Typ des Elements
    int inports, outports; // Anzahl von Input und Output Ports
    int xPos, yPos;     // Position des Icons im Designfenster
    int width, height; // Breite und Höhe des Element Icons
    char tag[30];       // Beschriftung des Element Icons
    HWND displayWnd;   // Ausgabefenster für das Element

    OUTPORTStruct out_ports[MAX_PORTS];
    // Name, Beschreibung, Wertebereich, Einheit der Output Ports

    INPORTStruct in_ports[MAX_PORTS];
    // Name, Beschreibung, Wertebereich, Einheit, aktueller Wert
    // der Input Ports

    LINKStruct out[MAX_CONNECTS];
    // Quelle (Elementnummer, Portnummer) und Ziel einer Verbind-
    // dung, Beschreibung, Wertebereich, Einheit der Verbiindung
    HWND hDlg; // Bezeichner für den Benutzerdialog des Elements

    BASE_CL (void) {}; // Konstruktor der Basisklasse,
    // Initialisierung Ports und Verbindungen
    virtual ~BASE_CL (void) {} // Destruktor der Basisklasse
    virtual void work (void) {} // Hauptfunktion des Elements,
    // (aufgerufen mit Samplefrequenz)
    virtual void update_inports (void) {}
    // wird aufgerufen bei einer Änderung der Input-Verbindungen

    virtual void session_start (void){} // Start der Session
    virtual void session_stop (void) {} // Stoppen der Session
    virtual void session_reset (void) {} // Reset der Session
    virtual void session_pos (long pos) {}
    // beim Positionieren auf einen bestimmten Zeitpunkt im Archiv

    virtual long session_length (void) {}
    // zur Ermittlung der Länge des Archivs

    virtual void make_dialog (void) {}
    // Anzeige des Benutzerdialogs

    virtual void load (HANDLE hFile) {}
    // Laden der Elementparameter aus der Konfigurationsdatei

    virtual void save (HANDLE hFile) {}
    // Speichern der akt. Elementparameter in Konfigurationsdatei

    virtual void incoming_data(int port, float value) {}
    // Wert <value> trifft am Eingangsport <port> ein

    void pass_values (int port, float value)
    // Wert <value> wird an die mit <port> verbundenen Elemente
    // weitergeleitet
};
```

6.5 Quellcode - Module

Die folgenden Quellcode- und Header-Dateien stellen nützliche Funktionen, Variablen und Strukturen für die Signal- und Dateiverwaltung, die Kontrolle der Seriellen Schnittstelle und Interaktion mit den Benutzerdialogen zur Verfügung. Durch die Deklaration in der gemeinsamen Header-Datei *brainbay.h* sind diese Funktionen und Speicherbereiche für alle Elementklassen und Programmmodule zugänglich. Sie dienen somit auch als Umgebung für die Neuentwicklung von *BrainBay*-Elementen.

6.5.1 Header-Datei *Base.h* :

Definition der Basisklasse *BASE_CL* (siehe 6.4) und maximale Größen der Datenfelder für Elemente sowie Input- und Output-Ports.

6.5.2 Header-Datei *Brainbay.h* :

Definition globaler Konstanten wie Nummern für die Element-Typen, Element-Namen, Datei-Typen, Datei-Modi, Standardwerte für die serielle Datenübertragung, Typen der Firmware-Übertragungsprotokolle. Hier werden auch die globalen Strukturen deklariert: *GlobalStruct*, *PacketStruct*, *CaptfileStruct*, *TTYStruct*, *TimingStruct*, *FliterStruct*, *EDFheaderStruct*, *EDFChannelStruct*. Eine genauere Beschreibung der Elemente dieser Strukturen findet sich im Quellcode der Headerdatei *brainbay.h*.

6.5.3 Modul *Brainbay.cpp* :

Dieses Modul beinhaltet die Hauptfunktion *WinMain(...)*. Hier werden die verschiedenen Fensterklassen registriert und das Applikationsfenster sowie das Designfenster erzeugt. Die Nachrichtenschleife und die Signalverarbeitung der Konfigurationselemente werden initialisiert. Wenn die Option *GLOBAL.startup* auf *TRUE* gesetzt ist, wird die zuletzt verwendete Design-Konfiguration geladen. Danach wird die Nachrichtenschleife (message queue) abgearbeitet wie in 6.3.1 beschrieben. Die Nachrichtenbehandlungsroutine des Hauptfensters *MainWndHandler(...)* verarbeitet Menüinteraktionen wie das Einfügen neuer Konfigurationselemente oder das Laden einer gespeicherten Design Konfiguration.

6.5.4 Modul *Globals.cpp* :

Dieses Modul beinhaltet die Definitionen globaler Variablen und Funktionen für die Initialisierung, De-Initialisierung und Hilfsfunktionen für die Verwaltung der Elemente:

6.5.4.1 Felder und Variablen für die Element-Verwaltung :

```
BASE_CL * objects[MAX_OBJECTS];  
// Feld mit Zeigern auf die Elemente des aktuellen Designs  
  
BASE_CL * actobject;           // das aktive (ausgewählte) Element  
struct LINKStruct * actconnect; // die aktive (ausgewählte) Verbindung  
  
int PACKETSPERSECOND=DEF_PACKETSPERSECOND;  
// Samplingfrequenz der Signalverarbeitung
```

6.5.4.2 Bezeichner für Instanz, Haupt- und Dialogfenster :

```
HINSTANCE hInst;          // Instanz des Hauptfensters
HACCEL ghAccel;          // Tastatur - Hotkeytabelle (keyboard accele-
                          // rator)
HWND ghWndMain;          // Handle des Hauptfensters
HWND ghWndToolbox;       // Handle des Toolbox-Fensters
                          // (zur Modifikation der Element-Parameter)
HWND ghWndStatusbox;     // Handle des Statusfensters
HWND ghWndSettings;     // Handle des Optionsfensters
HWND ghWndDesign;       // Handle des Design-Fensters
HWND ghWndAnimation;    // Handle des Animationsfensters
HGLRC GLRC_Animation;   // OpenGL-Kontext des Animationsfensters
```

6.5.4.3 Globale Datenstrukturen, Struktur für serielle Schnittstelle, Timing :

```
struct GLOBALStruct GLOBAL; // Einstellungen für Session und
                             // Kommunikation
struct TTYStruct TTY;       // Com-Port Einstellungen
struct PACKETStruct PACKET; // aktuell übertragenes Biosignal-Paket

struct MIDIPORTStruct MIDIPORTS[MAX_MIDIPORTS];
// verfügbare Midi- Eingabe und -Ausgabegeräte

struct TIMINGStruct TIMING;
// aktuelle Zeitmarker und systemspezifische Zeitkonstanten
```

6.5.4.4 Bezeichnungen für die Elemente und Verbindungen :

```
char objnames[50][20] = { OBJNAMES };
// Namen aller Elemente (die Liste OBJNAMES ist in
// brainbay.h definiert)
char dimensions[10][10]= { "uV", "mV", "Hz", "%" };
// Einheiten der Verbindungen
```

6.5.4.5 Funktionsdefinitionen im Modul *Globals.cpp* :

void register_classes (hinstance)

meldet alle verwendeten Fensterklassen der Hauptinstanz beim System an

void GlobalInitialize (void)

Initialisiert die *SDL*-Umgebung für hardwarenahe Multimediafunktionen, setzt den Pfad zur Applikation, die Einstellungen der seriellen Schnittstelle, die Position und Größe des Hauptfensters sowie Statusvariablen und Strukturen für die Zeichenfunktionen. Lädt die Standardeinstellungen für Datenaustausch mit der Biosignalhardware (Firmwareprotokoll und Samplerate), initialisiert die Timing-Variablen und öffnet die Midi-Geräte.

void GlobalCleanup (void)

Gibt den Speicherbereich aller verwendeten Elemente durch Aufruf der Objekt-Destruktoren frei, De-initialisiert die *SDL*-Umgebung, schließt die Midi-Geräte und geöffneten COM-Ports, löscht die Zeichenstrukturen.

void create_object (int type)

Erzeugt ein neues Objekt der durch *<type>* spezifizierten Elementklasse. Ein Zeiger auf das Element wird im Feld *objects[]* und in der Variable *actobject* gespeichert.

void free_object (int objectindex)

Löscht das Element mit dem Index *<objectindex>* aus der Konfiguration und gibt seinen Speicherbereich durch Aufruf des Objekt-Destruktors frei.

void swap_objects (int a, int b)

Tauscht die Position der Elemente mit den Indices *<a>* und ** in im Feld der Elemente, um ein Sortieren der Elemente zu ermöglichen.

int sort_objects (void)

Sortiert alle Elemente der aktuellen Konfiguration anhand ihrer Verbindungen (Quellelemente vor Zielelementen) um eine konsistente Abarbeitung der Elemente zu ermöglichen

void set_dimensions**(struct LINKStruct * act, float max, min, char * dim, * desc)**

Speichert neue Einstellungen für den Wertebereich, die Einheit und die Bezeichnung der Verbindung *<*act>* zwischen zwei Elementen.

void update_dimensions (void)

Propagiert die Einstellungen von Wertebereichen und Einheiten entlang der Signalverbindungen zwischen den Elementen

void update_samplingrate (int newrate)

Setzt einen neuen Wert für die Samplingrate der Signalverarbeitung und re-initialisiert die betroffenen Elemente wie *Filter*, *Magnitude* und *FFT*.

void get_session_length (void)

Ermittelt die Dauer der Archivdatei (Anzahl der Samples) und speichert den Wert in *GLOBAL.session_end*

void set_session_pos (long pos)

Setzt die aktuelle Position der Archiv-Wiedergabe auf Sample Nummer *<pos>*. Ruft die Methode *session_pos()* aller betreffenden Elemente auf.

float size_value (float min, max, x, to_min, to_max, int clip)

Führt eine Transformation des Wertes *x* vom Wertebereich (*<min>*, *<max>*) auf den Wertebereich (*<to_min>*, *<to_max>*) durch.

void reduce_filepath (char* to, char * from)

Kopiert den Filenamen in *<*from>* ohne den führenden Pfad in die string-variable *<*to>*.

int count_inports (BASE_CL * obj)

Gibt die Nummer des höchsten verbundenen Input-Ports des gegebenen Elements zurück.

reset_oscilloscopes (void)

Löscht den Inhalt aller Oszilloskop-Fenster der Konfiguration.

6.5.5 Modul *Files.cpp* :

Dieses Modul stellt Dateibehandlungsroutinen für Archive und Konfigurationsdateien zur Verfügung, wie die Erzeugung neuer Archive, das Lesen und Entpacken von Archivdaten, das Laden und Speichern von Design-Konfigurationen und Einstellungen:

HANDLE create_captfile (LPCTSTR lpFileName)

Erzeugt eine Archivdatei für die Aufzeichnung von empfangenen Biosignaldaten von der Verstärkerhardware (*ModularEEG* oder *MonolithEEG*) im Firmware-Format *P2* oder *P3*. Gibt bei erfolgreicher Erstellung einen Bezeichner für die Datei zurück, andernfalls die Konstante *INVALID_HANDLE_VALUE*.

void open_captfile (LPCTSTR lpFileName)

Öffnet eine Archivdatei zum Lesen. Wenn die Datei geöffnet werden konnte und das Format erkannt wurde, wird in der *CAPTFILE*-Struktur die Art des Firmware-Protokolls (*P2* oder *P3*) und der Bezeichner für die Datei eingetragen. Andernfalls wird eine Fehlermeldung ausgegeben.

void close_captfile (void)

Schließt eine geöffnete Archivdatei, die durch die *CAPTFILE*-Struktur referenziert wird. Die *CAPTFILE*-Struktur wird rückgesetzt und die längste geöffnete Archivdatei wird neu ermittelt.

void read_captfile (int amount)

Liest eine Anzahl von *<amount>* Bytes von einer geöffneten Archivdatei und transferiert die Daten in den Inputpuffer *TTY.readBuf*.

void write_captfile(unsigned char actbyte)

Schreibt ein Byte in eine geöffnete Archivdatei

int open_file_dlg (HWND hDlg, char * szFileName, int type, int flag_save)

Erzeugt einen Benutzerdialog zur Auswahl eines Verzeichnisses und einer Datei. *<hDlg>* ist ein Bezeichner für das aufrufende Fenster, *<szFileName>* enthält den Pfad zum Standardverzeichnis und nach der Auswahl den Pfad zur gewählten Datei, *<type>* wählt die Dateierweiterung zur Filterung der angezeigten Dateien und *<flag_save>* gibt an, ob die Datei zum Schreiben oder zum Lesen geöffnet werden soll. Das Öffnen einer nicht existierenden Datei zum Lesen erzeugt eine Fehlermeldung, das Überschreiben einer existierenden Datei wird durch einen Rückfragedialog verifiziert.

BOOL load_from_file(LPCTSTR pszFileName, void * buffer, int size)

Lädt einen Puffer gegebener Größe von einer Datei namens *<pszFileName>*.

BOOL save_to_file(LPCTSTR pszFileName, void * buffer, int size)

Speichert einen Puffer gegebener Größe in eine Datei namens *<pszFileName>*.

BOOL load_configfile(LPCTSTR pszFileName)

Lädt eine Design-Konfiguration mit dem gegebenen Namen. Falls die Datei existiert und eine Konfiguration beinhaltet, wird eine laufende Session gestoppt. Die bestehende Konfiguration wird geschlossen und alle Elemente werden

freigegeben. Die Elemente und Einstellungen der neuen Design-Konfiguration werden aus der Datei geladen, die neuen Elemente werden erzeugt und die Signalwege verbunden. Falls die Option *GLOBAL.autorun* gesetzt ist, wird die neue Konfiguration gestartet. Nach einem erfolgreichen Laden wird *TRUE* zurückgegeben, ansonsten *FALSE*.

BOOL save_configfile (LPCTSTR pszFileName)

Erzeugt eine Konfigurationsdatei mit dem gegebenen Namen und speichert Art und Einstellung der aktuellen Elemente in diese Datei. Nach einem erfolgreichen Speichern wird *TRUE* zurückgegeben, ansonsten *FALSE*.

BOOL save_settings (void)

Speichert die Applikations-Optionen in eine Datei namens *brainbay.cfg*. Zu den Optionen gehören die Samplingrate, COM-Port Einstellungen, Firmware-Protokoll, Midi-Geräte, automatisches Laden und Starten der letzten Konfiguration und Darstellungsgeschwindigkeiten der Ausgabefenster.

BOOL load_settings (void)

Lädt die Applikations-Optionen aus der Datei *brainbay.cfg*.

int load_next_config_buffer (HANDLE hFile)

Liest eine Zeichenkette bis zum Auftreten der Bezeichnung "*end object*" aus der Datei *<hFile>* und speichert sie in dem Puffer *GLOBAL.configbuffer*. Durch diese Funktion wird ein Parameterblock für die Einstellungen eines Elementes aus der Konfigurationsdatei gelesen.

void load_property (char * desc, int type, void * ad)

Untersucht den Inhalt der Puffers *GLOBAL.configbuffer* nach dem Auftreten der Zeichenkette *<*desc>*, die den Namen eines Elementparameters angibt. Wird der Name gefunden, so wird die nachfolgende Zeichenkette als Wert für den Parameter interpretiert. Der Typ des Parameters ist durch *<type>* bestimmt und kann *Integer (P_INT)*, *Float (P_FLOAT)* oder *String (P_STRING)* sein. Numerische Werte werden von einer Zeichenkette in den entsprechenden Typ umgewandelt. Der erhaltene Wert wird in die Variable *<*ad>* gespeichert, *<*ad>* wird als Zeiger auf den entsprechenden Typ interpretiert. Auf diese Weise können die Eigenschaften von Elementen dynamisch nachgeladen werden, und neue Elementparameter können hinzugefügt werden, ohne die Struktur des Konfigurationsfiles ändern zu müssen

void save_property (HANDLE hFile, char * desc,int type, void * ad)

Speichert einen Parameterwert als String-Zeichenkette in die Datei *<hFile>*. *<*desc>* bezeichnet den Namen des Parameters und wird vor dem Wert gespeichert. *Type* selektiert den Parametertyp als *Integer (P_INT)*, *Float (P_FLOAT)* oder *String (P_STRING)*, *<*ad>* ist ein Zeiger auf die Variable, die den Wert beinhaltet, siehe *load_property()*.

void save_object_basics (HANDLE hFile, BASE_CL * actobj)

Speichert die Position des Elements *<*actobj>* im Design-Fenster, die Anzahl der Input- und Output - Ports, die Bezeichnung und alle Verbindungen des Elements zu nachfolgenden Elementen in einer lesbaren Form als Zeichenkette in die Datei *<hFile>*.

void load_object_basics (BASE_CL * actobj)

Lädt die Position des Elements <actobj> im Design Fester, die Anzahl der Input- und Output-Ports, die Bezeichnung und alle Verbindungen zu nachfolgenden Elementen aus dem Puffer *GLOBAL.configbuffer*.

void parse_edf_header**(EDFHEADERStruct * to, CHANNELStruct * tochn, char * from)**

extrahiert *EDF*-Header und *EDF*-Kanal-Informationen aus der Zeichenkette <from> und speichert sie in die Strukturen <to> bzw. <tochn>.

void generate_edf_header**(char * to, EDFHEADERStruct * header, CHANNELStruct * channels)**

Generiert aus den *EDF*-Header und *EDF*-Kanalinformationen der Strukturen <header> bzw. <channel> eine sequentielle Zeichenkette.

HANDLE open_edf_file**(EDFHEADERStruct * to, CHANNELStruct * tochn, char * filename)**

Zeigt einen Dialog zur Auswahl einer *EDF*-Datei und extrahiert *EDF*-Header und *EDF*-Kanalinformationen aus dieser Datei, falls möglich. Im Erfolgsfall wird ein Bezeichner für die Datei zurückgeliefert, andernfalls die Konstante *INVALID_HANDLE_VALUE*.

HANDLE create_edf_file**(EDFHEADERStruct * from, CHANNELStruct * fromchn, char * filename)**

Zeigt einen Dialog zur Auswahl eines Dateinamens für die Speicherung einer neuen *EDF*-Datei. *EDF*-Header und -Kanalinformationen werden aus den gegebenen Strukturen erzeugt und in die Datei geschrieben. Im Erfolgsfall wird ein Bezeichner für die Datei zurückgeliefert, andernfalls die Konstante *INVALID_HANDLE_VALUE*.

6.5.6 Modul *TTY.cpp* :

Dieses Modul stellt Funktionen für die Verwendung der seriellen Schnittstelle (*COM-Port*) und eigene Prozesse (*Threads*) für die schnelle Ausführung der Lese- und Schreibaufgaben zur Verfügung.

BOOL SetupCommPort (int port)

Initialisiert und öffnet den angegebenen *COM-Port* <port>, etwa *COM3* oder *COM7*. Die gewählten Standardeinstellungen für die Verbindung richten sich nach dem Übertragungsprotokoll der *OpenEEG* - Hardware: 8 Datenbits, kein Paritätsbit, ein Stopbit. Die Baudrate ist durch den Wert von *TTY.BAUDRATE* gegeben und kann (bei Verwendung schneller *USB/Com-Konverter*) bis zu 921600 Bits/Sekunde betragen. Wenn der *Com-Port* erfolgreich geöffnet werden konnte, wird ein Bezeichner für die Verbindung in *TTY.COMDEV* gespeichert und ein *Lese-Thread* wird beim System angemeldet. Dieser *Thread* behandelt ankommende Werte mit der Funktion *ReaderProc()* und ruft nach Erhalt eines vollständigen Datenpaketes eine *Parser-Funktion* zum Decodieren des Paketes und zur Bereitstellung der Biosignalwerte auf. Falls ein Fehler

beim Öffnen des Com-Ports auftritt, wird eine Fehlermeldung erzeugt und die Funktion returniert *FALSE*, andernfalls wird *TRUE* zurückgegeben.

BOOL BreakDownCommPort (void)

Wenn ein Com-Port geöffnet ist, wird dieser geschlossen und der zugehörige Lese-Thread wird beendet.

6.5.7 Modul *Timer.cpp* :

Dieses Modul beinhaltet Funktionen für die Verwaltung der zeitgesteuerten Programmaktivitäten wie die Initialisierung und De-Initialisierung der *Timer-Interrupt* - Routine und die Aktualisierung der *TIMING* - Struktur. Wenn keine Datenübertragung von Biosignaldaten durch einen Com-Port stattfindet, kontrolliert die *Timer-Interrupt* Routine die Verarbeitung von Paketen und den Aufruf der *work()*- Funktionen der aktiven Elemente der Konfiguration.

void init_system_time (void)

Ermittelt die Eigenschaften des *PerformanceCounters* (das ist die genaueste Quelle von Zeitinformation in einem Windows-System) und setzt die Variable *TTY.packettime* entsprechend der gewählten Samlerate *PACKETSPERSECOND*. Der Paketzähler und die aktuelle Session-Zeit werden Null gesetzt.

void process_packets (void)

Diese Funktion führt die Signalverarbeitung durch, indem die Elemente der Design-Konfiguration abgearbeitet werden. Der Aufruf dieser Funktion erfolgt mit der aktuellen Samplerate. Die Datenpakete werden von den Output-Ports der Elemente zu den Input-Ports der nachfolgenden Elemente geleitet und die *work()* - Funktionen aller Elemente der Konfiguration werden aufgerufen. Danach werden der Paketzähler und die Statuszeile aktualisiert.

void CALLBACK TimerProc

(UINT uID, UINT uMsg, DWORD dwUser, DWORD dw1, DWORD dw2)

Die *Timer-Interrupt* Routine, die so oft wie möglich (Periode 1 Millisekunde) aufgerufen wird, kontrolliert die Zeitspanne seit dem letzten Aufruf von *process_packets()*. Wenn die verstrichene Zeit größer oder gleich der Sample-Periode ($1/PACKETSPERSECOND$) ist, wird *process_packets ()* erneut aufgerufen. Wenn Archivdateien durch die Konfiguration gelesen werden, wird ein Paket aus der Archivdatei geladen.

void start_timer(void)

Startet die *Timer-Interrupt* Routine.

void stop_timer(void)

Stoppt die *Timer-Interrupt* Routine.

6.5.8 Modul *Dialogs.cpp* :

Dieses Modul stellt Funktionen für die grafische Benutzeroberfläche und die Verarbeitung von Benutzerinteraktionen zur Verfügung. Es beinhaltet die Nachrichtenbe-

handlungsroutinen für die Standarddialogelemente der Applikation wie das Designfenster, die Statuszeile, das Optionen-Fenster und die Farb- und Tonskaleneditoren.

COLORREF select_color (HWND hwnd)

Erzeugt ein vom System zur Verfügung gestelltes Farbwahlfenster und erlaubt die Auswahl einer Farbe durch die Maus bzw. die Angabe von RGB-Werten. Der Rückgabewert ist die gewählte Farbe.

void update_toolbox_position (HWND hDlg)

Speichert die aktuelle Position des Parameterfensters für die Änderung von Element-Einstellungen in der Struktur *GLOBAL*.

void display_toolbox (HWND hDlg)

Das Dialogfenster *<hDlg>* wird an die gespeicherte Position des Parameterfensters verschoben und in der Reihenfolge der aktuell sichtbaren Fenster an die Spitze gestellt (*Top Layer Window*). Der Eingabefokus wird auf dieses Dialogfenster gesetzt.

void close_toolbox(void)

Falls das Parameterfenster geöffnet ist, wird es geschlossen.

int get_scrollpos (WPARAM wParam, LPARAM lParam)

Fragt die Position eines horizontalen oder vertikalen Schiebereglers (*Scrollbar*) ab und gibt die aktuelle Position zurück.

void color_button (HWND hWnd, COLORREF newcolor)

Setzt die Farbe einer durch *<hWnd>* referenzierten Fensterkomponente auf die neue Farbe *<newcolor>*

LRESULT CALLBACK DesignWndHandler (HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)

Nachrichtenbehandlungsroutine für das Design-Fenster. Hier werden alle Maus- und Tastaturnachrichten zur grafischen Erstellung der Konfiguration behandelt. Dazu gehört die Anordnung und das Verschieben der *Icons* für die Elemente, die Erzeugung von Verbindungen zwischen den Elementen, die Darstellung des elementspezifischen Parameterfensters, das Löschen von Elementen aus der Design-Konfiguration und das *Scrollen* des Design-Fensters. Wird eine neue Verbindung zwischen zwei Elementen gelegt, dient ein Aufruf der *sort_objects()* - Funktion zur Aktualisierung der Elementreihenfolge, damit die sequentielle Reihung für die Signalverarbeitung gewahrt bleibt.

void report (char * Message)

Stellt die Meldung *<*Message>* in einem Dialogfenster dar.

void report_error (char * Message)

Stellt die Fehlermeldung *<*Message>* in einem Dialogfenster dar.

void critical_error (char * Message)

Stellt die Fehlermeldung *<*Message>* in einem Dialogfenster dar und beendet die Applikation.

void add_to_listbox (HWND hDlg, int idc, char * str)

Fügt die Zeichenkette <str> zu einem Listenelement eines Dialogfensters hinzu, welches durch den Dialogbezeichner <hDlg> und die Ressourcennummer <idc> identifiziert wird.

6.5.9 Modul *Draw.cpp* :

Dieses Modul beinhaltet Funktionen zur Erstellung globaler Zeichenobjekte und zur Darstellung des Design-Fensters mit den Element-Icons und -Verbindungen.

void init_draw (void)

Erzeugt einige Zeichenobjekte wie Stifte (*Pens*), Pinsel (*Brushes*) und Zeichensätze (*Fonts*) in der *DRAW*- Struktur.

void draw_object (HDC hdc, WORD t)

Zeichnet das Icon von Element Nummer <t> in den Gerätekontext <hdc>. <hdc> identifiziert die Zeichenfläche des Design-Fensters. Wenn Element <t> gerade selektiert ist, wird es mit einem orangen Rand gezeichnet.

void draw_connections (HDC hdc, WORD t)

Zeichnet die Verbindungen von Element <t> zu den nachfolgenden Elementen in das Design-Fenster.

void draw_captions (HDC hdc, WORD t)

Schreibt die Beschriftungen von Element <t> in das Design-Fenster.

void draw_objects (HWND hWnd)

Zeichnet das Netzwerk aller Elemente mit den Verbindungen und Beschriftungen. <hWnd> ist der Bezeichner des Design-Fensters

6.5.10 Modul *Midi.cpp* :

Dieses Modul stellt Variablen und Funktionen zur Konfiguration von Midi-Ausgabe Geräten und zur Erzeugung von Midi-Nachrichten bereit. Dadurch können die Tonerzeugungsmechanismen einer Soundkarte oder von externen Midi-Geräten wie *Samplern*, *Sequenzern* und *Drum-Computern* genutzt werden.

*** char midi_instnames[256]**

Dieses Feld von Zeichenketten speichert die Namen für 127 Standard Midi-Instrumente und 127 Standard Midi-Controller.

void init_midi (void)

Kopiert die Namen der im System verfügbaren Midi - Ausgabegeräte in die *MIDIPOINTS* - Struktur und speichert die Anzahl der Geräte in *GLOBAL.midiports*.

void midi_Instrument (HMIDIOUT * midiout, int chn, int inst)

Wählt das Instrument <inst> für Kanal <chn> des Midi-Gerätes <*midiout>

void midi_ControlChange (HMIDIOUT * midiout, int chn, int cont, int val)
 Sendet die *Control-Change* Nachricht <val> zum Controller <cont> auf Kanal <chn> des Midi-Gerätes <*midiout>

void midi_NoteOn (HMIDIOUT * midiout, int chn, int note, int vol)
 Spielt den Midi-Ton <note> mit der Lautstärke <vol> auf Kanal <chn> des Midi-Gerätes <*midiout>

void midi_NoteOff (HMIDIOUT * midiout, int chn, int note)
 Beendet einen eventuell noch spielenden Midi-Ton <note> auf Kanal <chn> von Midi-Gerät <midiout>

int midi_open_port (HMIDIOUT * midiout, int portnum)
 Öffnet das Midi-Output Gerät Nummer <portnum> und speichert den Bezeichner in <HMIDIOUT>. Rückgabewert ist *TRUE* bei Erfolg und *FALSE* andernfalls.

void mute_all_midi (void)
 Beendet die Ausgabe auf allen Midi-Kanälen der aktuellen Design-Konfiguration.

6.6 Module der Verarbeitungselemente

Die folgende Tabelle zeigt eine Übersicht auf die Module, die Elemente für Signal-Erzeugung, -Verarbeitung und -Darstellung beinhalten. Die Spalte *Besonderheiten* zeigt spezielle Eigenschaften für das jeweilige Element, die nützlich für die Entwicklung ähnlicher Elemente sein könnten.

Element / Modul	Funktionalität	Besonderheiten
<i>ob_and.cpp</i>	Führt die logische Operation <i>AND</i> mit den zwei Input-Werten aus und gibt das Ergebnis am Output-Port aus	
<i>ob_average.cpp</i>	Gibt den arithmetischen Mittelwert der letzten <i>n</i> Eingabewerte am Output-Port aus	
<i>ob_avi.cpp</i>	Extrahiert ein Bild aus einer <i>AVI</i> Filmdatei und zeigt es in einem Fenster. Die Nummer des Bildes (<i>frame</i>) wird durch den Wert am Input-Port bestimmt.	Verwendet ein <i>OpenGL</i> Fenster zur Darstellung
<i>ob_ballgame.cpp</i>	Implementiert ein Geschicklichkeitspiel (Ping-Pong), kontrolliert durch den Wert des Input-Ports	verwendet ein <i>GDI</i> - Zeichenfenster zur Darstellung des Spieles
<i>ob_cam.cpp</i>	Darstellung und Speicherung eines Live-Kamerabildes, Abspielen aufgezeichneter Kamerabilder, Gesichtserkennung auf Basis der Kamerabilder	Live-Kamerafenster, läuft in eigenem Thread, verwendet die <i>OpenCV</i> -Library
<i>ob_compare.cpp</i>	Vergleicht die Werte zweier Input-Ports, gibt das Ergebnis am Output-Port aus	

Element / Modul	Funktionalität	Besonderheiten
<i>ob_com_writer.cpp</i>	Gibt eine Kommandosequenz auf den Com-Port aus, der zum Einlesen von Biosignalen geöffnet wurde (bidirektional)	
<i>ob_constant.cpp</i>	Gibt einen konstanten Wert am Output-Port aus	
<i>ob_correlation.cpp</i>	Gibt die Kreuzkorrelation zweier angeschlossener Signale am Output-Port aus	
<i>ob_counter.cpp</i>	zählt die Anzahl von <i>TRUE/FALSE</i> Änderungen des Signals am Input Port und stellt das Ergebnis oder den direkten Signalwert in einem Fenster dar.	verwendet ein <i>GDI</i> - Zeichenfenster zur Ausgabe
<i>ob_debounce.cpp</i>	Entfernt <i>TRUE/FALSE</i> Änderungen aus dem Input-Signal	
<i>ob_deviation.cpp</i>	Berechnet die Standard Abweichung des Signals	
<i>ob_doku.cpp</i>	Stellt eine Textbox für Dokumentationszwecke zur Verfügung	
<i>ob_edf_reader.cpp</i>	Liest Werte aus einer <i>EDF</i> -Datei ein und gibt die Signale an den Output-Ports aus	Dynamische Anzahl von Output-Ports
<i>ob_edf_writer.cpp</i>	Schreibt Werte in eine <i>EDF</i> -Datei	Dynamische Anzahl von Input-Ports
<i>ob_eeg.cpp</i>	Baut eine Verbindung zum Biosignal-Verstärker auf, stellt aktuelle Werte oder Archivdaten an den Output-Ports zur Verfügung	Erzeugt einen <i>Lese-Thread</i> zum Empfang der Daten von der seriellen Schnittstelle, taktet beim Empfang von Biosignaldaten die Verarbeitung aller Elemente
<i>ob_erpdetect.cpp</i>	Speichert und mittelt ein Segment des Input-Signals (Aufnahmemodus), gibt die Differenz des aufgezeichneten Segments zum aktuellen Signal aus (Wiedergabemodus)	Stellt das Signalsegment im Benutzerdialog dar.
<i>ob_evaluator.cpp</i>	Evaluiert einen mathematischen Ausdruck, der bis zu sechs Eingabesignale beinhaltet	Ruft die <i>GNU Matheval</i> -Bibliothek auf
<i>ob_fft.cpp</i>	Führt eine <i>fast fourier</i> Transformation (<i>FFT</i>) durch und stellt das Ergebnis in einem Balkendiagramm, einem Spektrogramm oder in einer 3D-Grafik dar	Verwendet ein <i>OpenGL</i> -Fenster zur Darstellung
<i>ob_file_writer.cpp</i>	Schreibt die Eingabewerte in ein Textfile, das zB. von <i>Excel</i> geöffnet werden kann	Dynamische Anzahl von Input-Ports
<i>ob_filter.cpp</i>	Wendet einen einstellbaren Filter auf das Input-Signal an	Zeigt die Filter-Antwort im Benutzerdialog
<i>ob_integrate.cpp</i>	Summiert die Werte des Input Signals	
<i>ob_magnitude.cpp</i>	Gibt die Magnitude eines einstellbaren Frequenzbandes am Output-Port aus	
<i>ob_matlab.cpp</i>	Transferiert ein Feld von Eingabewerten zur <i>Matlab-Engine</i> , gibt den Wert der <i>ANS</i> -Variable am Output-Port aus	Ruft Funktionen der <i>Matlab</i> - Bibliotheken auf

Element / Modul	Funktionalität	Besonderheiten
<i>ob_mci.cpp</i>	Schnittstelle zum <i>Windows Multimedia Player</i> , getriggertes Abspielen von Musik oder Videodaten und die Änderung der Lautstärke oder der Abspielgeschwindigkeit	Ruft die <i>Video for Windows - API</i> auf, zeigt Filme in einem separaten Fenster
<i>ob_midi.cpp</i>	Generiert Midi-Töne entsprechen der Eingabewerte, ermöglicht die Vorgabe von Tonleitern	Ruft die <i>MMsystem-API</i> zur Erzeugung von Midi-Kommandos auf
<i>ob_mixer4.cpp</i>	Kombiniert bis zu vier Eingabesignale mit einstellbarer Gewichtung in ein Ausgabesignal	
<i>ob_mouse.cpp</i>	Steuert den Maus-Cursor und die Klick-Aktionen anhand von Eingabewerten	Verwendet die Funktion <i>mouseevent()</i> zur Manipulation des Maus-Cursors
<i>ob_not.cpp</i>	Führt die logische Operation <i>NOT</i> mit dem Input-Wert aus und gibt das Ergebnis am Output-Port aus	
<i>ob_or.cpp</i>	Führt die logische Operation <i>OR</i> mit dem Input-Wert aus und gibt das Ergebnis am Output-Port aus	
<i>ob_osci.cpp</i>	Stellt die angeschlossenen Signale in einer Oszilloskop-Ansicht dar	Verwendet ein <i>GDI</i> Zeichenfenster
<i>ob_particle.cpp</i>	Berechnet ein grafisches Partikelsystem mit durch Input-Werte einstellbaren Parameters wie Farbe, Gravitation, Geschwindigkeit etc.	Verwendet ein <i>OpenGL</i> Fenster zur Darstellung
<i>ob_sample_hold.cpp</i>	Speichert einen Wert des Input-Ports und gibt diesen kontinuierlich am Output-Port aus	
<i>ob_signal.cpp</i>	Signal-Generator mit einstellbarer Frequenz, Signaltyp, Amplitude, Phase und Offset.	Wählbare Input-Ports für Frequenz und Amplitude
<i>ob_skindialog.cpp</i>	Grafischer Benutzerdialog mit freier Formwahl (<i>Skin</i>) für Buttons und Schieberegler	Ruft die <i>skinstyle</i> - Bibliothek auf, initialisiert den Benutzerdialog
<i>ob_tcp_receive.cpp</i>	Empfängt <i>EDF</i> -Daten von einer LAN- oder Internet Verbindung über <i>TCP</i> .	Baut eine Verbindung mit dem <i>Neuroserver</i> Netzwerk Service auf, dynamische Anzahl von Output-Ports
<i>ob_tcp_sender.cpp</i>	Sendet <i>EDF</i> -Daten zu einer LAN- oder Internet Verbindung über <i>TCP</i> .	Baut eine Verbindung mit dem <i>Neuroserver</i> Netzwerk Service auf
<i>ob_threshold.cpp</i>	Ermöglicht die Einstellung von oberen und unteren Grenzen für ein Signal. Diese Grenze können fix oder variabel (dynamisch aufgrund der vergangenen Signalwerte) gewählt werden	zeigt den aktuellen Signalwert und die Signalgrenzen in einem <i>GDI</i> -Zeichenfenster
<i>ob_translate.cpp</i>	Verstärkt und/oder verschiebt das Input-Signal	
<i>ob_wav.cpp</i>	Durch die Eingabewerte getriggerte Wiedergabe einer Sound-Datei mit einstellbarer Geschwindigkeit	verwendet <i>SDL_SOUND</i> und <i>modplug.dll</i> für die Soundausgabe

Tabelle 6.1 Source-Module der Brainbay-Applikation

6.7 Signalverarbeitung durch die Konfigurationselemente

Die interne Signalverarbeitung erfolgt durch einen Datenaustausch zwischen den Elementen der Design-Konfiguration: Jeder Output-Port eines Elementes kann grafisch mit einem oder mehreren Input-Ports anderer Elemente verbunden werden. Jeder Input-Port kann eine Verbindung aufnehmen. *Floating-Point* Werte werden periodisch entlang dieser Verbindungen transferiert. Die Geschwindigkeit dieser Weiterleitung ist durch die eingestellte Samplefrequenz gegeben. Werden die Biosignal-Rohdaten von der Verstärkerhardware geliefert, so erfolgt die gesamte Weiterleitung durch die Elemente mit dieser Frequenz. Die Reihenfolge der Aufrufe ist durch die Verbindungen der Elemente bestimmt: Elemente, die Daten generieren werden vor Elementen aufgerufen, welche diese Daten empfangen. Die Sortierung der Elemente wird beim Einfügen neuer Elemente bzw. beim Erstellen einer neuen Verbindung durchgeführt. Der Aufruf des Sortieralgorithmus befindet sich in der Nachrichtenbehandlungsroutine für das Design-Fenster (Modul *dialogs.cpp*).

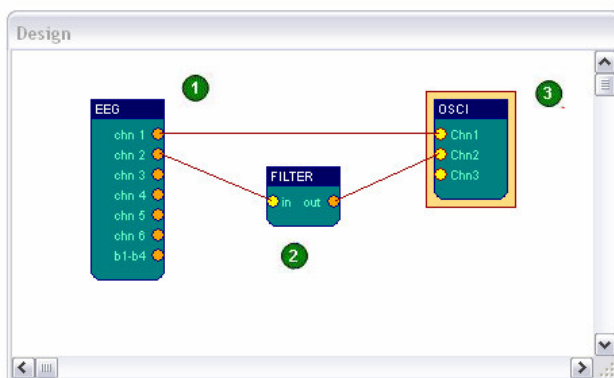


Abb. 6.1: Designfenster, Anordnung von Elementen (1)

Das EEG-Element der abgebildeten Konfiguration empfängt Daten von der seriellen Schnittstelle. Es wird als erstes aufgerufen, danach das Filter-Element und zuletzt das Oszilloskop-Element.

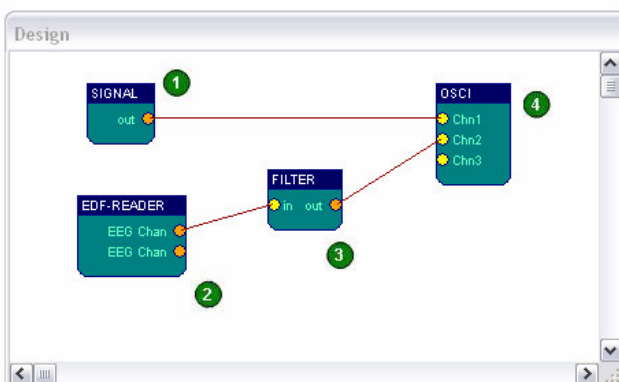


Abb. 6.2: Designfenster, Anordnung von Elementen (2)

Diese Abbildung zeigt eine Konfiguration ohne direkte Verbindung zu einer Biosignal-Hardware und die Ausführungsreihenfolge der Elemente. Die Signalverarbeitung dieser Konfiguration wird periodisch durch die *Timer-Interrupt*-Funktion initiiert.

6.7.1 Weiterleitung der Signalinformation

Die Methode *pass_values (int port, float value)*, welche Teil der Basisklasse *BASE_CL* ist und somit allen Elementen vererbt wird, transferiert Ausgabewerte eines Elements an alle verbundenen Elemente. *Pass_values()* ruft zu diesem Zweck die Methode *incoming_data (port, value)* der betreffenden Elemente auf:


```

void pass_values (int port, float value)
{
    LINKStruct * act_link;
    for (act_link=&(out[0]);act_link->to_port!=-1;act_link++)
        if (act_link->from_port==port)
            objects[act_link->to_object]->incoming_data
                (act_link->to_port,value );
}

```

Der Aufruf von *pass_values()* findet in den *work()* - Methoden der Elemente statt, nachdem die aktuellen Werte für den Output-Port berechnet wurden.

6.7.2 Realisierung des Zeitmanagements

Der Aufruf der Konfigurationselemente erfolgt durch ihre *work()* - Methoden. Durch die Funktion *process_packets()* werden die *work()* - Methoden aller Elemente in der richtigen Reihenfolge ausgeführt. Ziel ist, dass dieser Aufruf möglichst genau im Intervall der aktuellen Samplefrequenz stattfindet. Der Zeitpunkt des Aufrufes wird entweder durch den *Timer-Interrupt* oder durch einen Erhalt eines vollständiges Datenpaketes vom Biosignalverstärker bestimmt. Auf diese Weise werden alle Elemente mit aktuellen Daten versorgt.

```

void process_packets(void)
{
    static double calc;
    int t;
    { ... }
    TIMING.ppscounter++; // Ermittlung der Pakete pro Sekunde
    TIMING.packetcounter++; // Pakete total

    for (t=0;t<GLOBAL.objects;t++)
        objects[t]->work(); // Aufruf der work-Methoden

    if (!TIMING.dialog_update)
    {
        update_statusinfo(); // Aktualisierung der Zeit
    }
}

```

Die *Timer-Interrupt* - Funktion wird periodisch im Millisekundentakt aufgerufen. Durch den aktuellen Wert des *PerformanceCounters* wird entschieden, ob ein Aufruf der Elemente erfolgt, bzw. ob von der Archivdatei gelesen werden soll:

```

void CALLBACK TimerProc(UINT uID,UINT uMsg,DWORD dwUser,
    DWORD dw1,DWORD dw2)
{
    LONGLONG pc;
    QueryPerformanceCounter((_LARGE_INTEGER *)&pc);
    TIMING.acttime=pc;
    check_keys(); // Überprüfung aktueller Hot-Keys
    if (!TIMING.pause_timer)
    {
        if (pc-TIMING.timestamp >= TIMING.pcfreq)
            // Eine Sekunde vergangen -> aktualisiere
            // Packetspersecond
            {
                TIMING.timestamp+=TIMING.pcfreq;
                TIMING.actpps=(int) TIMING.ppscounter;
                TIMING.ppscounter=0;
            }

        if (pc-TIMING.readtimestamp >= TTY.packettime)
            // Lesen von Archivdatei und Sampleintervall erreicht
            // -> neues Paket lesen

```

```

    {
        TIMING.readtimestamp+=TTY.packettime;
        if (CAPTFILE.do_read)
        {
            read_captfile (TTY.amount_to_read);
            ParseLocalInput (TTY.amount_to_read);
            return;
        }
        process_packets(); // Aufruf aller work()-Methoden
    }
}
}

```

6.8 Verarbeitung der Daten vom Biosignalverstärker

Die Kommunikation mit der Verstärkerhardware (*MonolithEEG* oder *ModularEEG*) erfolgt über die serielle Schnittstelle. Dies kann eine Standard- *RS232*-Verbindung oder ein virtueller Com-Port (*VCP*) sein, der durch einen USB-Adapter oder (wie beim *MonolithEEG*) durch die Treibersoftware eines USB-Chips erzeugt wird. Die Lese-Operationen werden in einem eigenen *Lese-Thread* durchgeführt. Die Funktion *SetupCommPort()* initialisiert die Schnittstelle und die erzeugt den *Lese-Thread*:

```

BOOL SetupCommPort(int port)
{
    DWORD dwReadStatId;
    DCB dcb = {0};
    { ... } // Variablen vorbereiten

    TTY.COMDEV = CreateFile( PORTNAME, GENERIC_READ | GENERIC_WRITE,
        0, 0, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL |
        FILE_FLAG_OVERLAPPED, 0);

    if (TTY.COMDEV == INVALID_HANDLE_VALUE) goto failed;

    { ... } // DCB - Settings für Handshake und Timeouts setzen

    if (!SetCommState(TTY.COMDEV, &dcb)) goto failed;
    // Einstellungen für Baudrate und Frame setzten

    if (!SetCommTimeouts(TTY.COMDEV, &(TTY.TIMEOUTSNEW)))
        goto failed;
    // Einstellungen für Timeouts setzen

    SetupComm(TTY.COMDEV, MAX_READ_BUFFER, MAX_WRITE_BUFFER);
    // Puffergrößen setzten

    TTY.READERTHREAD = CreateThread( NULL, 0,
        (LPTHREAD_START_ROUTINE) ReaderProc, (LPVOID) ghWndMain, 0,
        &dwReadStatId);
    // Anmelden und Starten von ReaderProc() als Thread

    if (TTY.READERTHREAD == NULL)
    { report_error("CreateThread failed"); goto failed;}
    return TRUE; // Com - Port erfolgreich initialisiert

failed:
{
    report_error("Com Port not available");
    TTY.CONNECTED=FALSE;
    return FALSE;
}
}

```

Der *Lese-Thread (ReaderProc)* dient zur eventgesteuerten Verarbeitung der eingehenden Datenpakete. Die *Read()* - Funktion wird dafür im *Overlapped* - Modus aufgerufen. Die Funktion *WaitForMultipleObjects()* wartet auf ausstehende Leseergebnisse, ohne das System zu blockieren und führt die Behandlung der aufgetreten Events durch.

```

DWORD WINAPI ReaderProc(LPVOID lpv)
{
    OVERLAPPED osReader = {0}; // Struktur für overlapped-read
    HANDLE      hArray[NUM_EVENTHANDLES];
    // Events zur Steuerung des Thread
    { ... }
    osReader.hEvent = CreateEvent(NULL, FALSE, FALSE, NULL);
    // Erzeugung der Overlapped-Struktur für die Lese-Events
    hArray[0] = osReader.hEvent; // Lese-Event
    hArray[1] = TTY.ThreadExitEvent; // Exit-Event

    while ( (!fThreadDone) && (TTY.CONNECTED) ) {
        if (!fWaitingOnRead)
            // wenn frei: starte Leseoperation auf den seriellen Port
            {
                if (!ReadFile(TTY.COMDEV, TTY.readBuf,
                    TTY.amount_to_read, &dwRead, &osReader))
                {
                    if (GetLastError() != ERROR_IO_PENDING)
                        // Fehler beim Lesen: Lesen abbrechen
                        {
                            TTY.CONNECTED=FALSE;
                            fThreadDone = TRUE;
                        }
                    else fWaitingOnRead = TRUE; // warten auf Daten
                }
                else // Daten sind sofort angekommen
                {
                    if ((!TTY.read_pause) && (GLOBAL.running))
                        ParseLocalInput((int)dwRead); // Paket parsen
                }
            }
        if ( fWaitingOnRead ) { // warten auf die Daten
            dwRes = WaitForMultipleObjects(NUM_EVENTHANDLES, hArray,
                FALSE, READ_TIMEOUT);

            // Warten auf Event
            switch(dwRes) {
                case WAIT_OBJECT_0: // Lese event wurde ausgelöst
                    if (!GetOverlappedResult(TTY.COMDEV, &osReader,
                        &dwRead, FALSE))
                        // Fehler in Lese-Event
                        {
                            if (GetLastError() == ERROR_IO_INCOMPLETE) b
                                break;
                            report_error("GetOverlappedResult failed");
                        }
                    else // Daten sind angekommen
                    {
                        if ((!TTY.read_pause) && (GLOBAL.running))
                            ParseLocalInput((int)dwRead); // Paket parsen
                        fWaitingOnRead = FALSE;
                        break;
                    }
                case WAIT_OBJECT_0 + 1: // Exit-Event ausgelöst
                    fThreadDone = TRUE; // Thread beenden
                    break;
                default: // andere Fehler
                    report_error("Reader Thread does not return");
                    break;
            }
        }
    }
    CloseHandle(osReader.hEvent); // Schließen der Event-Handles
    return 1;
}

```

6.9 Implementierung der Gesichtsdetektion

Das Ziel der Implementierung der Gesichtsdetektion ist es, geeignete Informationen für eine Mauszeigersteuerung per Kopf aus den Kamerabildern zu extrahieren. Die Veränderung der Kopfposition kann dadurch festgestellt und zur Steuerung der x/y - Beschleunigung des Zeigers verwendet werden. Zusätzliche Informationen wie die Erkennung von Bewegungen der Augenbrauen oder des Kinns können zur Verwendung der Klick-Funktionen dienen. Die aktuell ermittelten Koordinaten werden als Ausgabewerte des *Camera*-Elementes in der Konfiguration zur Verfügung gestellt. Detektion und Tracking werden in separaten Schritten durchgeführt.

6.9.1 Verwendete *OpenCV*-Funktionen

Eine relativ einfache Realisierung der Gesichtererkennung und Bewegungsdetektion ist durch Funktionen der *Open Computer Vision Library* von Intel möglich [BAR00],[OPC06]. Diese bietet ein reichhaltiges Set von Funktionen zur Abfrage einer Kamera und zur Bildverarbeitung. Zu den geeigneten Strategien für eine Gesichtserkennung zählt die *Kaskade von Haar-ähnlichen Merkmalsextraktoren (Haar-Classifler*, siehe 4.6.4.2). In der Distribution der *OpenCV*-Library befindet sich auch ein Demoprogramm für eine Gesichtererkennung mittels *Haar*-Klassifikation und eine *XML*-Datei mit trainierten Gesichtern (Frontalansicht) ist verfügbar.

```
capture = cvCaptureFromCAM( 0 ); // Öffnen der Kameraverbindung
frame = cvQueryFrame( capture ); // Abfrage des aktuellen Frame

cascade = (CvHaarClassifierCascade*)cvLoad( fname, 0, 0, 0 );
// Importieren der Haar-Trainingsdatei

faces = cvHaarDetectObjects( frame_copy, cascade, storage,
1.1, 4, CV_HAAR_DO_CANNY_PRUNING , cvSize(70, 70) );
// Ausführung der Klassifikation
```

Nach der Ausführung von *cvHaarDetectObjects()* sind die Positionen erkannter Gesichter als umrandende Rechtecke im Array *faces* gespeichert. Nachdem die Haar-Klassifizierung zu rechenaufwändig für eine schnelle Echtzeitauswertung ist, werden Bewegungen mit Hilfe des *Lucas-Kanade Optical-Flow* Algorithmus detektiert (siehe 4.6.4.3). Der Aufruf der *OpenCV*-Funktion:

```
cvCalcOpticalFlowPyrLK( prev_grey, grey, prev_pyramid, pyramid,
points[0], points[1], count, cvSize(7,7), 5, status, 0,
cvTermCriteria(CV_TERMCRIT_ITER|CV_TERMCRIT_EPS,20,0.03), flags);
```

Durch die Arrays *points[0]* bzw. *points[1]* werden die Positionen der Featurepunkte im vorigen Grauwertbild (*prev_grey*) angegeben bzw. die neu ermittelten Positionen der Punkte im aktuellen Bild (*grey*) zurückgeliefert [OPC06].

6.9.2 Initialisierung der Gesichtspostion

Sobald ein Gesicht durch die Haar-Klassifikation detektiert werden konnte, werden die Positionen der Nase und des Kinns geschätzt. Die Nase eignet sich für eine intuitive Steuerung des Interface und sie ist auch als Tracking-Merkmal gut unterscheidbar [GOR02].

Das Kinn ist eine gute Position um willkürliche Bewegungen abzunehmen, die zur Auslösung von Klicks einsetzbar sind. Die Initialisierungsphase ist abgeschlossen wenn die Koordinaten für Nase und Kinn in zulässigen Bereichen liegen. Die erkannten Punkte werden im Kamerabild grün bzw. gelb markiert.



Abb. 6.3: Tracking (links) und Neuinitialisierung (rechts) von Gesichtsmerkmalen

6.9.3 Tracking von Gesichtsmerkmalen

In den folgenden Einzelbildern wird durch den Lucas-Kanade Algorithmus versucht, die Positionsänderung von Nase und Kinn zu schätzen und die Punkte zu aktualisieren. Die neuen Positionen von Nase und Kinn werden einer Plausibilitätsprüfung unterzogen (Entfernung und Orientierung im Raum). Sind die errechneten Fehlerwerte zu groß, werden die Punkte rot markiert und eine neue Gesichtsdetektion wird durchgeführt (Re-Initialisierung).

Der Grund für die Verwendung des optischen Trackings ist die schnellere Ausführungsgeschwindigkeit für die Echtzeitverarbeitung: Mit dem LK-Algorithmus lassen sich bei der Hintergrundausführung des Programms 15-20 Bilder pro Sekunde verarbeiten, dies wäre beim permanenten Einsatz der Haar-Cascade nicht möglich.

6.10 Die Erzeugung eines neuen Verarbeitungselementes

Um ein neues Element und den zugehörigen Benutzerdialog zur Applikation hinzuzufügen, sind die folgenden Schritte durchzuführen:

- Hinzufügen des Namens und der Identifikationsnummer des Elements zur Header-Datei *brainbay.h*
- Erzeugung eines Benutzerdialogs für das Parameterfenster des Elements
- Ableitung einer Klasse für das Element von *BASE_CL*, Deklaration der verwendeten Variablen und Implementierung der benötigten Methoden. Jedes sinnvolle Element braucht zumindest die Methoden *load()* und *save()* sowie die *Konstruktor*- und *Destruktor*-Methoden. Elemente, die Daten verarbeiten brauchen auch die Methoden *incoming_data()* und *work()*.
- Implementierung der Nachrichtenbehandlungsroutine für den Benutzerdialog
- Hinzufügen eines Menüeintrages zum Einfügen des neuen Elements in die Konfiguration und Aufruf der *create_object()* -Funktion im Hauptmodul *brainbay.cpp* bei Auswahl des Menüpunktes (siehe *MainWndHandler* im Modul *brainbay.cpp*)

Im folgenden Abschnitt werden diese Schritte anhand eines Beispiels gezeigt. Ein neues Element mit dem Namen *demo* wird mit Hilfe der Entwicklungsumgebung *VisualStudio C++ 6.0* zur bestehenden Applikation hinzugefügt.

6.10.1 Einträge in die Header-Datei *brainbay.h*

Am Beginn der Datei *brainbay.h* findet sich die Definition einer eindeutigen Konstanten pro Element. Hier wird eine Zeile für das neue Element *OB_DEMO* hinzugefügt:

```
(....)
#define OB_FILE_WRITER      37
#define OB_DEVIATION       38
#define OB_MCIPLAYER       39
#define OB_DEMO           40
(...)
```

Unterhalb der Konstanten für die Identifikation des Elementes befinden sich die Definitionen für die Standardbezeichnungen der Elemente in einem Feld von Zeichenketten. Hier wird der Name für das neue Element hinzugefügt:

```
#define OBJNAMES "EEG","MIDI", (...),
                "FILEWRITE", "DEVIATION", "MEDIAPLAYER" , "DEMO"
```

6.10.2 Erzeugung eines Benutzerdialogs

Zur Erstellung der grafischen Benutzeroberfläche für das Parameterfenster des neuen Elements wird der *Resource Editor* von *VisualStudio* verwendet. Ein leeres Dialogfenster wird durch einen Klick mit der rechten Maustaste auf den *Dialog* -Eintrag der Ressourcenliste und die Auswahl von *'Dialog einfügen'* erzeugt. Der Standardname (*IDD_DIALOG1*) kann durch Doppelklick auf das Dialogfenster in *IDD_DEMOBOX* umbenannt werden. Dieser Bezeichner wird später zur Erzeugung des Dialogfensters bei Laufzeit benötigt. Die Stil - Einstellungen des Dialogfensters werden wie folgt gewählt: *Überlappend-Dünn, Absolute Position, Modales Fenster, Statische Kante, Im Vordergrund bleiben*. Die Standard Schaltflächen *'OK'* und *'Abbrechen'* werden gelöscht. Der Dialog erhält die Titelbezeichnung *'Demodialog'*. Nun werden die Interaktionskomponenten hinzugefügt, für dieses Beispiel sind dies ein Text-Feld, ein Schieberegler, eine Auswahlbox eine Schaltfläche und statische Texte zur Beschriftung. Alle Komponenten erhalten eindeutige Bezeichner in Form von Konstanten (*IDC_<idname>*). Die Namen dieser Konstanten werden zur leichten Zuordenbarkeit umbenannt auf *IDC_EDIT*, *IDC_SCROLLBAR*, *IDC_COMBO* und *IDC_BUTTON*.

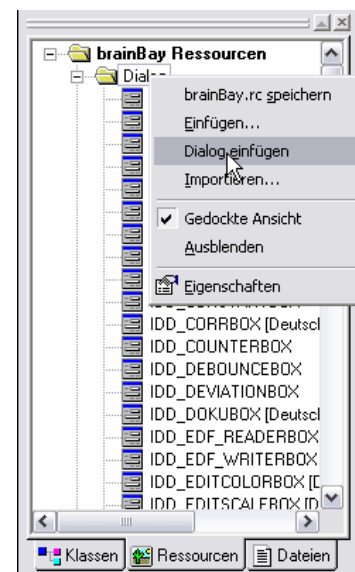


Abb. 6.4: Ressourcen-Editor

6.10.3 Klassenimplementierung für das Element

Zunächst werden eine neue Header-Datei (.h) und eine neue Quellcode-Datei (.cpp) für das Element benötigt, um das Modul zu beherbergen. Die neue Objektklasse wird von der Basisklasse `BASE_CL` abgeleitet. Der einfachste Weg, die neuen Dateien zu erzeugen ist, eine Kopie eines ähnlichen Elements zu erzeugen und die nicht benötigten Stellen zu löschen bzw. brauchbare Stellen durch Umbenennen zu adaptieren. Für dieses Beispiel werden die Dateien `ob_demo.h` und `ob_demo.cpp` benannt. Die neue Objektklasse heißt `DEMOOBJ`. Die Dateien müssen zur VisualStudio Projektumgebung hinzugefügt werden, um übersetzt und eingebunden zu werden. Dies geschieht durch den Menüpunkt *Projekt-> Dem Project hinzufügen -> Dateien*

6.10.3.1 Die Header-Datei für das Element

Die Header-Datei `ob_demo.h` beinhaltet die Deklaration der neuen Klasse:

```
class DEMOOBJ : public BASE_CL
{
    protected:    // Deklaration von geschützten Variablen

    public:
        // public-Variablen können von der
        // Nachrichtenbehandlungsroutine verwendet werden
        int selection;    // Selektierter Eintrag der Auswahlbox
        char string[50]; // Eingabe in das Textfeld
        int pos;         // Position des Schiebereglers
        float in1,in2; // Puffer für die Werte von Port1 und 2

    // STANDARD METHODEN :

    DEMOOBJ(int num);           // Konstruktor
    void make_dialog(void);     // Erzeugt den Benutzerdialog
    void load(HANDLE hFile);    // Lädt die Elementparameter
    void incoming_data(int port, float value);
    // Werte an den Input-Ports eingetroffen
    void save(HANDLE hFile);    // Speichert die Parameter
    void work(void);
    // Berechnungen, Weiterleiten der Werte an die Output Ports
    ~DEMOOBJ();                // Destruktor

    // METHODEN für spezielle Funktionalitäten:

    void update_inports (void);
    // beim Verbinden oder Entfernung von Input-Ports
    void session_start (void);
    // beim Start der Session durch den Benutzer
    void session_stop (void);
    // beim Beenden der Session durch den Benutzer
    void session_reset (void);
    // beim Reset der Session durch den Benutzer
    void session_pos (long pos);
    // beim Positionieren innerhalb der geöffneten Archivdateien

    // Nachrichtenbehandlungsroutine für den Benutzerdialog:
    friend LRESULT CALLBACK DemoDlgHandler
    (HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam);
    private: // Raum für private Variablen
};
```


6.10.3.2 Die Quellcode-Datei für das Element

Die Implementierung der Objektklasse und der Nachrichtenbehandlungsroutine befinden sich in der Quellcodedatei *ob_demo.cpp*. Das Element soll zwei mit den Input-Ports verbundene Signale auf eine Art kombinieren, die durch die Auswahlbox bestimmbar ist. Der Schieberegler dient zur Skalierung des Outputs von 0 - 100%. Die Schaltfläche soll eine Meldung am Bildschirm darstellen. Der folgende Quellcode zeigt die Implementierung der dafür nötigen Standard-Methoden, die Verwendung der zusätzlichen Methoden wird durch Ausgabe von Kontrollmeldungen dargestellt.

Zu Beginn der Quellcodedatei werden die Header-Files inkludiert:

```
#include "brainBay.h" // globale Definitionen und Strukturen
#include "ob_demo.h" // Element-spezifische Deklarationen, Klasse
```

Danach folgt der Konstruktor für das Object. Dieser wird durch das Einfügen des Elements mittels Auswahl des Menüpunktes via *create_object()* aufgerufen:

```
DEMOOBJ::DEMOOBJ(int num) : BASE_CL() // Element Konstruktor
{
    outports = 1; // Ein Output-Port für das Element
    inports = 2; // Zwei Input-Ports für das Element

    strcpy(in_ports[0].in_name, "val1");
    // Input Port 1 bekommt den Namen 'val1'

    strcpy(in_ports[1].in_name, "val2");
    // Input Port 2 bekommt den Namen 'val2'

    strcpy(out_ports[0].out_name, "out");
    // Output Port bekommt den Namen 'out'

    strcpy (string, "hello world");
    // Initialisierung des Output Strings

    // Hier ist Platz für zusätzliche Initialisierungen
}
```

Falls ein Element zusätzliche Funktionalitäten wie ein eigenes Ausgabefenster oder ähnliches bieten soll, werden die nötigen Ressourcen ebenfalls im Konstruktor initialisiert.

Als nächstes wird die Methode *make_dialog()* implementiert, die den Benutzerdialog für das Element (das Parameterfenster) erzeugt. Diese Methode wird aufgerufen, wenn das Symbol für das Element im Design-Fenster mit dem rechten Mausknopf geklickt wird :

```
void DEMOOBJ::make_dialog(void)
{
    display_toolbox(hDlg=CreateDialog(hInst, (LPCTSTR)IDD_DEMOBOX,
                                     NULL, (DLGPROC)DemoDlgHandler));

    // Erzeugung des Dialogfensters, IDD_DEMOBOX ist der Bezeichner
    // der Dialog-Resource.
}
```

Nun folgen die *load()*- und *save()* Methoden. Durch die Verwendung der Funktionen *load_property()* und *save_property()* werden Bezeichner für die jeweiligen Parameter und die zugehörigen Parameterwerte im lesbaren ASCII-Format in der Konfigurati-

onsdatei (*.con*) abgelegt. Die Datei kann somit durch einen Texteditor importiert werden, und eine nachträgliche Änderung oder Erweiterung der Elementparameter ist möglich, ohne dadurch die Konfiguration zu zerstören.

```
void DEMOOBJ::load(HANDLE hFile)
{
    load_object_basics(this);
    // laden von Standardwerten wie Position des Element-Symbols,
    // Anzahl der In- und Output Ports und Beschriftung

    // nun werden die drei elementspezifischen Parameter geladen:
    load_property("pos",P_INT,&pos); // Position des Schiebereglers
    load_property("selection",P_INT,&selection);
    // aktuelle Selektion der Auswahlbox

    load_property("string",P_STRING,string); // Text des Edit-Feldes

    // Hier ist Platz für zusätzliche Initialisierungen
}

void DEMOOBJ::save(HANDLE hFile)
// hFile ist ein Bezeichner für die geöffnete Konfigurationsdatei
{
    save_object_basics(hFile,this); // Speichern der Standardwerte
    save_property(hFile,"pos",P_INT,&pos);
    save_property(hFile,"selection",P_INT,&selection);
    save_property(hFile,"string",P_STRING,string);

    // Die Reihenfolge des Speicherns der Parameter ist unkritisch,
    // da beim Laden der ganze Block von Elementparametern
    // untersucht wird. Hier werden nur String- und Integer-Werte
    // benötigt, für Gleitkommawerte wird der Typ P_FLOAT verwendet.
}
```

Für Elemente, die größere Datenmengen erzeugen, wie zB. Farbtabelle, Tonleitern oder Signalwerte, empfiehlt sich die Ablegung dieser Informationen in externen Dateien, da die Konfigurationsdatei sonst zu stark anwachsen würde. In der Konfigurationsdatei werden in diesem Fall nur die Dateinamen der zusätzlichen Ressourcen abgelegt, die Daten selbst werden in den *load()*- und *save()* Methoden separat geladen bzw. gespeichert. Verwendet das Element zusätzliche Speicherbereiche wie zB. Wertepuffer, ist die Allokation bzw. Freigabe dieser Speicher in der *load()*- Methode durchzuführen, um Mehrfachallokationen beim Laden der Konfiguration zu vermeiden. Beispiele hierfür sind die *Filter*- und *Magnitude*- Elemente.

Nun folgt die Implementierung der *incoming_data()* - Methode, die sich um die Behandlung eintreffender Werte über die Input-Ports kümmert:

```
void DEMOOBJ::incoming_data(int port, float value)
{
    if (value != INVALID_VALUE) // Verwerfen von INVALID_VALUES
    {
        if (port==0) in1 = value;
        // Speichern eintreffender Werte von Input-Port 1
        if (port==1) in2 = value;
        // Speichern eintreffender Werte von Input-Port 2
    }
}
```

Die *work()* -Methode berechnet den Output des Elements entsprechend der Selektion der Auswahlbox und der Einstellung des Schiebereglers. Das Ergebnis wird durch die Methode *pass_values()* an verbundene Elemente weitergeleitet:

```

void DEMOOBJ::work(void)
{
    switch (selection)
        // berechne Wert abhängig von der selektierten Auswahl
    {
        case 0: pass_values(0, in1 * pos / 100.0f); break;
        // erste Option gewählt: skaliertes Wert von Input1
        case 1: pass_values(0, in2 * pos / 100.0f ); break;
        // zweite Option gewählt: skaliertes Wert von Input 2
        case 2: pass_values(0, (in1+in2) * pos ); break;
        // dritte Option: skaliertes Wert von Input1 + Input2
    }
}

```

Der Objekt-Destruktor dient zur Freigabe des Speicherbereiches der Klasse. Hier sollten auch zusätzliche Speicherbereiche, die das Element reserviert hat, freigegeben werden.

```

DEMOOBJ::~~DEMOOBJ() {} // es wurde nichts extra allokiert

```

Nun sind alle Standardmethoden für das Element implementiert. Die zusätzlichen Methoden werden nun zu Demonstrationszwecken gezeigt:

```

void DEMOOBJ::update_inports(void)
// wird aufgerufen, wenn Input-Ports verbunden oder Verbindungen
// gelöscht werden
{
    inports=count_inports(this);
    // Ermitteln des höchsten aktuell verbundenen Input Ports
    // und Aktualisieren der Input-Ports des Elements
    // Dies führt zu einer dynamischen Anzahl von Input Ports

    height=CON_START+inports*CON_HEIGHT+5;
    // neue Höhe des Elementsymbols berechnen

    InvalidateRect(ghWndDesign,NULL,TRUE);
    // Designfenster neu zeichnen
}

void DEMOOBJ::session_start(void)
// wird beim Starten der Session durch den Play-Button aufgerufen
{ report ("Session has been started"); }

void DEMOOBJ::session_stop(void)
// wird beim Beenden der Session durch den Stop-Button aufgerufen
{ report ("Session has been stopped"); }

void DEMOOBJ::session_reset(void)
// wird beim Rücksetzen der Session durch den Reset-Button aufgerufen
{ report ("Session has been reset"); }

void DEMOOBJ::session_pos(long pos)
// wird beim Ändern der Position in einer Archivdatei aufgerufen
{
    char sztemp[50];
    sprintf(sztemp,"Session has been positioned to %d.",pos);
    report(sztemp); // Ausgabe der aktuellen Position
}

```

Nun fehlt noch die Nachrichtenbehandlungsroutine für die Interaktion mit dem Benutzerdialog. Der Dialog wird durch die *make_dialog()* - Methode dargestellt. Die Nachrichten für Initialisierung, Interaktion mit den Dialogkomponenten und Beendigung des Dialogs werden behandelt:

```

LRESULT CALLBACK DemoDlgHandler(HWND hDlg, UINT message, WPARAM
wParam, LPARAM lParam)
{
    DEMOOBJ * st;
    st = (DEMOOBJ *) actobject;
    // actobject zeigt auf das gewählte Element, durch den Type-Cast
    // wird ein Zugriff auf die public-Variablen des Objektes möglich.

    switch( message )          // Behandlung der erhaltenen Nachricht
    {
        case WM_INITDIALOG:    // Der Benutzerdialog wurde erzeugt

            SCROLLINFO lpsi;
            lpsi.cbSize=sizeof(SCROLLINFO);
            lpsi.fMask=SIF_RANGE|SIF_POS;
            lpsi.nMin=1; lpsi.nMax=100;
            SetScrollInfo(GetDlgItem(hDlg, IDC_SCROLLBAR), SB_CTL, &lpsi, 1);
            // setzen des Wertebereichs für den Schieberegler von 1 - 100

            SetScrollPos(GetDlgItem(hDlg, IDC_SCROLLBAR), SB_CTL,
                st->pos, 1);
            // Die aktuelle Position des Reglers wird auf den
            // gespeicherten Wert (pos) gesetzt:

            SetDlgItemText(hDlg, IDC_EDIT, st->string);
            // string im Edit - Feld darstellen

            // füllen der Auswahlbox mit den Selektionsmöglichkeiten:
            SendDlgItemMessage(hDlg, IDC_COMBO, CB_ADDSTRING, 0, (LPARAM) "IN1");
            SendDlgItemMessage(hDlg, IDC_COMBO, CB_ADDSTRING, 0, (LPARAM) "IN2");
            SendDlgItemMessage(hDlg, IDC_COMBO, CB_ADDSTRING, 0, (LPARAM) "IN1+2");

            SendDlgItemMessage(hDlg, IDC_COMBO, CB_SETCURSEL,
                st->selection, 0L ); // Auswahl der aktuellen Selektion
            break;

        case WM_CLOSE:        // Der Benutzerdialog wird geschlossen
            EndDialog(hDlg, LOWORD(wParam));
            return TRUE;

        case WM_COMMAND:     // Nachricht an eine Dialogkomponente

            switch (LOWORD(wParam)) // ID der Dialogkomponente
            {
                case IDC_COMBO: // Auswahlbox wurde geändert
                    if (HIWORD(wParam)==CBN_SELCHANGE)
                        st->selection= SendMessage(GetDlgItem(hDlg,
                            IDC_COMBO), CB_GETCURSEL, 0, 0);
                // neue Auswahl wird in selection gespeichert
                    break;

                case IDC_EDIT: // Textfeld wurde geändert
                    GetDlgItemText(hDlg, IDC_EDIT, st->string, 50);
                    // neuer Inhalt wird gespeichert
                    break;

                case IDC_BUTTON: // Schaltfläche wurde betätigt
                    report (st->string) ;
                    // Anzeige des Textfeldinhaltes als Nachricht
                    break;
            }
            return TRUE;

        case WM_HSCROLL:     // ein Schieberegler wurde geändert
            {
                int nNewPos= get_scrollpos(wParam, lParam);
                // Ermitteln der neuen Position des Schiebereglers
                st->pos=nNewPos ; // Speichern der neuen Position in pos
            }
            break;
    }
}

```

```

        case WM_SIZE:           // Änderung der Position oder Größe
        case WM_MOVE:          // oder Position des Benutzerdialogs
            update_toolbox_position(hDlg);
            // speichern der neuen Position
            break;
        return(TRUE);
    }
    return (FALSE);
}

```

6.10.4 Menüeintrag zur Einfügung des Elements in eine Konfiguration

Um einen Menüeintrag zum Einfügen des Elements in die Konfiguration zu erstellen, wird wieder der Ressourcen-Editor von *VisualStudio* verwendet. Ein Untermenü des *Insert Element* - Pulldown-Menüs (hier wurde *Others* verwendet) wird mit dem neuen Eintrag erweitert. Der Name der Auswahl und der Bezeichner für den Menüpunkt sind einzutragen:

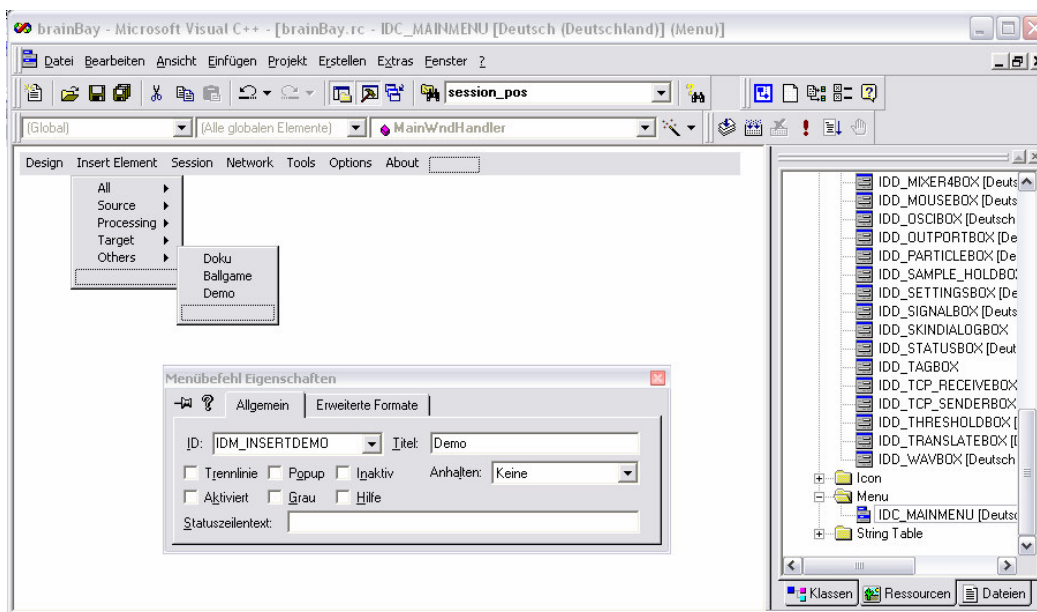


Abb. 6.5: Hinzufügen eines neuen Menüeintrages mit dem Ressourcen-Editor

Die Behandlung des neuen Menüeintrages erfolgt im Modul *brainbay.cpp*. Bei der Auswahl des Menüpunktes soll das Element erstellt und im Design-Fenster angezeigt werden. Dies geschieht durch die Funktion *create_object()*. Eine Zeile mit diesem Aufruf ist der Menüsektion in der Nachrichtenbehandlungsroutine des Hauptfensters hinzuzufügen:

```

( ... )
case IDM_INSERTFILE_WRITER:create_object(OB_FILE_WRITER); break;
case IDM_INSERTDEVIATION:create_object(OB_DEVIATION); break;
case IDM_INSERTMCIPLAYER:create_object(OB_MCIPLAYER); break;
case IDM_INSERTDEMO:create_object(OB_DEMO); break;
( ... )

```

Die Funktion *create_object()* muss für die Erzeugung der Objektklasse noch erweitert werden. Sie befindet sich im Modul *globals.cpp*. Zu Beginn dieses Moduls wird die Header-Datei des neuen Elements eingetragen:

```
( ... )
#include "ob_file_writer.h"
#include "ob_deviation.h"
#include "ob_mci.h"
#include "ob_demo.h"
```

Die *create_object()* - Funktion wird um folgende Zeile erweitert:

```
( ... )
case OB_FILE_WRITER:
    actobject=new FILE_WRITEROBJ(GLOBAL.objects); break;
case OB_DEVIATION:
    actobject=new DEVIATIONOBJ(GLOBAL.objects); break;
case OB_MCIPLAYER:
    actobject=new MCIOBJ(GLOBAL.objects); break;
case OB_DEMO:
    actobject=new DEMOOBJ(GLOBAL.objects); break;
```

Nun kann das neue Element compiliert und getestet werden !

7. Anwendung von *BrainBay*

Das folgende Kapitel beschreibt die Installation und die Verwendung der lauffähigen *BrainBay*-Applikation. Die einzelnen Konfigurationselemente werden vorgestellt, und ihre Verbindung zur lauffähigen Konfiguration wird anhand von konkreten Beispielen diskutiert.

7.1 Installation und erster Start des Programms

7.1.1 Systemvoraussetzungen

BrainBay verwendet das *Windows* - Betriebssystem und läuft unter den Versionen *Windows XP*, *Windows 2000*, *NT* und *Windows 98*. Die Hardwarevoraussetzung für einen Start der Applikation sind ein Pentium-II Laptop- oder Desktop-Computer mit 128MB Ram. Die Echtzeitfähigkeit des Laufzeitsystems ist jedoch prinzipiell von der Komplexität der Konfiguration von Verarbeitungs- und Darstellungselementen abhängig. Wenn die Konfiguration rechenintensive Elemente wie die 3D-Darstellung von FFT-Werten oder Bildverarbeitung von Webcam-Bildern beinhaltet, wird ein entsprechend schnellerer Rechner benötigt. Ein 1,6 Ghz Pentium IV Rechner bietet ausreichende Rechenleistung für Standardkonfigurationen den Bereichen Biofeedback und Echtzeitverarbeitung von Kamerabildern. Eine Grafikkarte mit Hardwarebeschleunigung für *OpenGL* - Kommandos verbessert die Darstellungsgeschwindigkeit mancher Grafikausgaben wie des *FFT*-Elements und des *Particle-Generator* Elements signifikant.

7.1.2 Download der aktuellen Programmversion

Die ausführbare Applikation kann in Form eines komprimierten Paketes von der Internet-Seite <http://brainbay.lo-res.org> heruntergeladen werden. Um dieses Paket zu entpacken, kann ein beliebiges Dekomprimierungsprogramm für *.rar* Dateien verwendet werden, wie zB. das Hilfsprogramm *7-zip* (kostenlos verfügbar unter <http://www.7-zip.org>).

Derzeit beinhaltet das Download-Paket keine eigene Anwendung für die Installation der Applikation. Durch das Dekomprimieren wird ein Unterverzeichnis mit dem Namen *BrainBay* erstellt, welches die ausführbare Datei *brainbay.exe*, die benötigten *.dll* - Dateien (siehe 6.2), Unterverzeichnisse für Programm- und Konfigurationskomponenten sowie Archivdateien und Demo-Konfigurationen enthält.

7.1.3 Verzeichnisstruktur

Im Verzeichnis der Applikation werden die folgenden Unterverzeichnisse erstellt:

- *Archives*: beinhaltet Signalarchivdateien in den Formaten *.EDF* (*European Data Format*) oder *.arc* (*BrainBay*-spezifisches Archivformat für die Firmwareprotokolle *P2* und *P3*). Diese Archivdateien werden von Designkonfigurationen verwendet und können in neue Konfigurationen zum Test von Funktionen oder zur Offline-Analyse eingebunden werden. Dieses Unterverzeichnis wird zum Laden oder Speichern von Archivdateien vorgeschlagen.
- *Configurations*: dieses Verzeichnis ist der Speicherort für Design-Konfigurationen (*.con* - Dateien), welche Zusammenstellungen von Elementen und Elementparameter beinhalten. Ein Set von Demonstrations-Konfigurationen wird bei der Installation in dieses Verzeichnis abgelegt.
- *Graphics*: beinhaltet Grafikdateien als Ressourcen für den Partikel-Generator und andere Elemente, die Visualisierungen ermöglichen.
- *Movies*: beinhaltet Videodateien für die Elemente *AVI-Player* und *MCI-Player* (*.avi* -Dateien)
- *Patterns*: beinhaltet aufgenommene Signalsegmente für das Mustererkennungselement *ERP-Detector* (*.erp* - Dateien)
- *Palettes*: beinhaltet Farbtabelle für Elemente, die Visualisierungen ermöglichen (*.pal* - Dateien)
- *Tonescales*: beinhaltet Tonleitern für das Midi-Element (*.sc* - Dateien)
- *Network*: beinhaltet ausführbare Dateien der *Neuroserver*-Softwareumgebung (<http://openeeg.sourceforge.net/doc/software/NeuroServer>). Die *Neuroserver* Umgebung ermöglicht das Senden und Empfangen von *EDF*-Dateien über eine TCP - Netzwerkverbindung.
- *Sounds*: beinhaltet Audio-Dateien für die *Wav-Player* und *MCI-Player* Elemente (*.wav* Dateien)
- *Skindialogs*: enthält die Definitionsdateien und grafischen Ressourcen für Freiform-Benutzerdialoge des *Skindialog*-Elements (*.ini* und *.bmp* - Dateien)

7.1.4 Einrichten der Applikation

Wenn *BrainBay* zum ersten Mal gestartet wird, öffnet sich das Optionen-Fenster automatisch (Normalerweise wird es durch den Hauptmenüeintrag *Options* geöffnet)

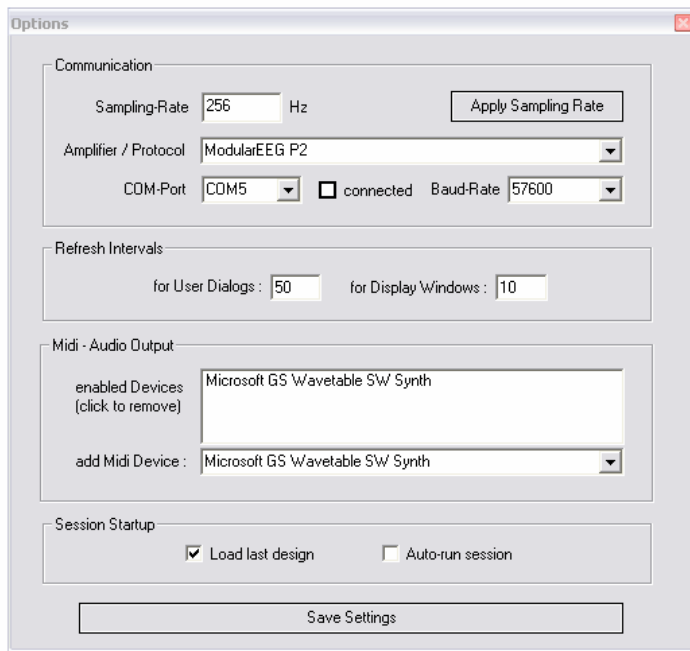


Abb. 7.1: Benutzerdialog für Programmoptionen

Im Bereich *Communication* können die Standardwerte für die Samplerate und die Werte für die serielle Verbindung mit der Biosignal-Hardware eingestellt werden. Die Samplerate gibt die Frequenz für die Signalverarbeitung im Programm vor. Zur Verwendung des *ModularEEG* Verstärkers wird eine Samplerate von 256Hz eingestellt. Der *MonolithEEG* - Verstärker verwendet je nach Einstellung Sampleraten bis zu 1200 Hz. Werden EDF-Signalarchive verwendet, muss die Samplerate entsprechend gewählt werden. Der Bereich *Amplifier / Protocol* dient zur Einstellung der Standardwerte für das Übertragungsprotokoll und für die verwendete Biosignal-Hardware. Je nach Gerätefirmware ist hier *P2*, *P3* oder *P21* auszuwählen. Weiters wird hier der standardmäßig verwendete Schnittstellenport und die Übertragungsgeschwindigkeit eingestellt. 57600 Baud sind für eine Übertragung von 6 Kanälen des *ModularEEG* mit 256 Hz ausreichend. Für die Nutzung der vollen Bandbreite des *MonolithEEG* wird eine Übertragungsgeschwindigkeit von 460.800 Baud benötigt.

Im Bereich *Refresh Intervals* wird die Aktualisierungsgeschwindigkeit der grafischen Displays eingestellt. *User Dialogs* gibt den Wert für die Parameterfenster der Elemente vor, *Display Windows* setzt Zeichengeschwindigkeit für die Ausgabefenster der Elemente wie Oszilloskop-Fenster, FFT-Darstellung oder Balkendarstellungen. Diese Werte werden in Samples angegeben: Ein Wert von 5 bedeutet eine Aktualisierung wenn 5 Samples verarbeitet wurden, das wäre bei einer Samplerate von 256 etwa alle 19,5 Millisekunden.

Der Bereich *Midi-Audio Output* dient zur Auswahl der Midi-Ausgabe-Geräte, die in der Applikation verfügbar sein sollen. Ein Midi-Ausgabegerät könnte eine Soundkarte oder ein externes Midi-Gerät wie ein Keyboard, ein Sampler oder ein Drum-Computer sein. Durch *Cable patch*-Treiber wie die kostenlose Software *MidiYoke* werden virtuelle Midi-Geräte im System angemeldet, was einen Transfer von Midi-Kommandos in andere laufende Applikationen ermöglicht (<http://www.midiox.com/myoke.htm>). Um ein Midi-Gerät in *BrainBay* zu verwenden, wird es in der Auswahlbox ausgewählt und zur Liste der aktiven Geräte hinzugefügt. Um ein Gerät zu löschen, wird es in der Liste doppelt geklickt.

Der Bereich *Session Startup* erlaubt die Einstellungen von Optionen für den Start des Programms: Wenn *load last design* ausgewählt ist, wird die zuletzt verwendete Design-Konfiguration automatisch geladen, durch Auswahl von *auto-run* wird diese auch automatisch gestartet.

7.2 Verwendung von Hauptmenü und Statuszeile

Die wichtigsten Funktionen von *BrainBay* können durch das Hauptmenü und durch die Statuszeile am unteren Fensterrand erreicht werden. Das Menü bietet die folgenden Einträge:

Design: Erstellen einer neuen Designkonfiguration, Laden oder Speichern von Konfigurationen, Beenden der Applikation

Insert Element: Neue Elemente zum Design hinzufügen

Session: Starten oder Stoppen der aktuellen Session

Network: Starten des *Neuroservers* zur Netzwerkübertragung von Biosignaldaten.
(*TCP-Receive* 7.6.1.5 und *TCP-Write Element* 7.6.3.10)

Tools: Editoren für Farbpaletten und Midi-Tonleitern

Options: Einstellungen für Kommunikation und Midi-Output (siehe 7.1.4)

About: Programmversion und Urheberinformationen

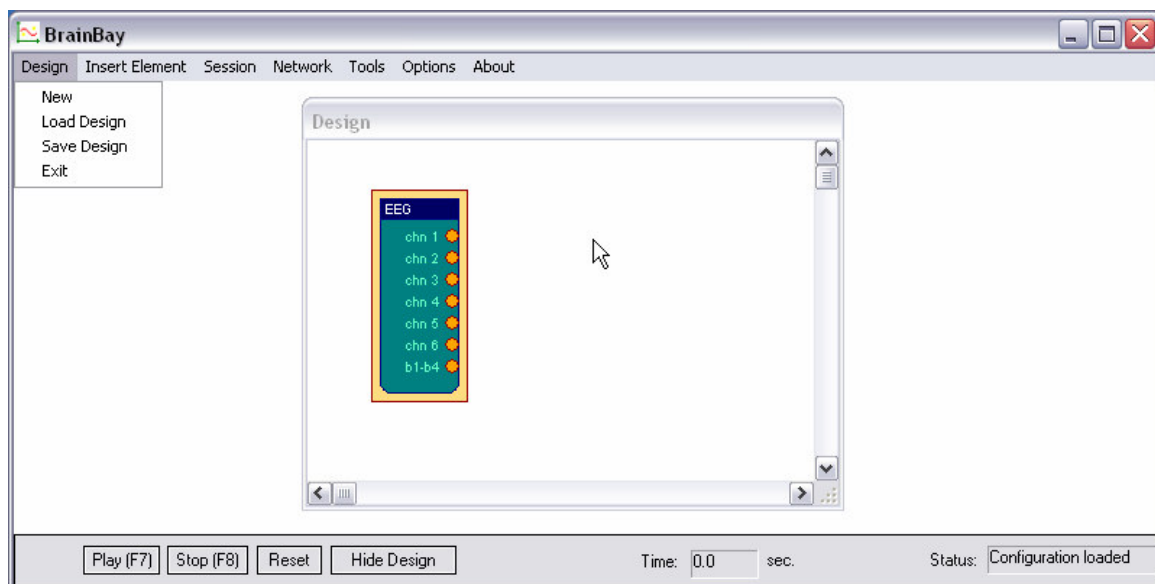


Abb. 7.2: Hauptmenü und Design-Fenster der BrainBay-Applikation

Die Statuszeile am unteren Rand des Applikationsfensters zeigt Informationen wie die aktuelle Laufzeit der Session, den aktuellen Programmstatus und Schaltflächen für die Session-Kontrolle. *Play* und *Stop* (und die entsprechenden Hotkeys *F7* und *F8*) dienen zum Starten und Stoppen der Session. *Reset* stoppt eine laufende Session, setzt die Position von geöffneten Archivdateien auf deren Beginn und setzt die aktuelle Session-Zeit auf null. Die Schaltfläche *Hide/Show Design* (bzw. der Hotkey *F5*) kontrolliert die Sichtbarkeit des Design-Fensters. Die Statuszeile selbst kann durch den Hotkey *F6* ein- und ausgeblendet werden.

Falls die aktuelle Konfiguration Archivdateien beinhaltet (entweder durch die Anwesenheit von EDF- oder File- Leseelementen oder durch eine ausgewählte Archivdatei im EEG-Element) wird die Statuszeile mit der Anzeige der Archivposition und entsprechenden Kontrollelementen für die Archivpositionierung erweitert. Durch die Schaltflächen In und Out können Einstiegs- und Ausstiegspunkte in der Archivdatei definiert werden. Die Schaltfläche Go To ermöglicht die Positionierung der aktuellen

Session-Zeit auf einen bestimmten Zeitpunkt. (weitere Informationen siehe *EEG-Element 7.6.1.1* und *EDF-Reader - Element 7.6.1.4*)

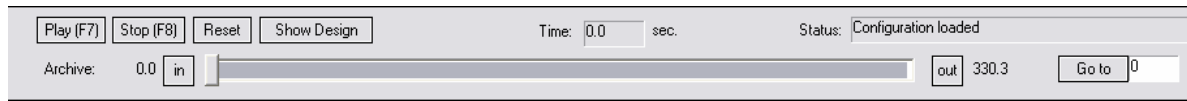


Abb. 7.3: Statuszeile mit Informationen über die Session und zur Kontrolle der Wiedergabe

7.3 Der Farbpaletteneditor

Der Farbpaletteneditor wird durch die Menüauswahl *Tools -> Edit Color Palettes* geöffnet und kann zur einfachen Erstellung von Farbverläufen und Farbfolgen verwendet werden. Diese Farbpaletten werden von einigen Elementen zur visuellen Darstellung verwendet (*FFT-Element*, *Particle-Generator - Element*)

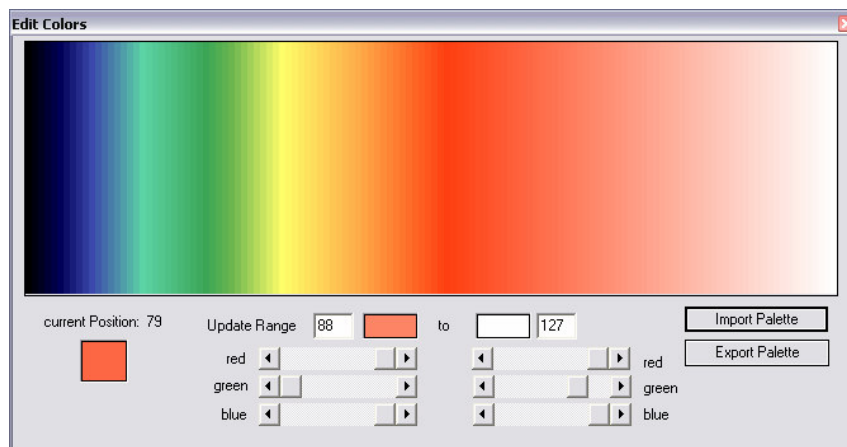


Abb. 7.4: Der Farbpaletten-Editor

Um den Farbverlauf zu ändern, wird ein Bereich durch linken und rechten Mausklick in das Farbfenster gewählt. Der Linksklick gibt den Beginn und der Rechtsklick das Ende der Region an. Durch die Farbwahlfelder oder die Schieberegler für RGB-Werte werden die Farben für die Start- und Endpunkte definiert und ein Farbverlauf über die gewählte Region wird erzeugt. Auf diese Weise können Farbverläufe über Regionen beliebiger Größe mit wenigen Klicks erzeugt werden. Eine Palette besteht aus 127 einzelnen Farbwerten. Bestehende Paletten können durch *Import Palette* geladen und verändert werden. Neue Paletten werden durch *Export Palette* gespeichert.

7.4 Der Tonleitern-Editor

Der Tonleitern-Editor wird durch die Menüauswahl *Tools -> Edit tone Scales* geöffnet und erlaubt die Zusammenstellung und Speicherung von Tonskalen für das Midi-Element. Bei der Generierung von Midi-Noten dienen diese Skalen als Filter, wodurch konventionelle Tonleitern wie Moll, Dur oder Blues sowie eigene Tonabfolgen oder einzelne Töne zur akustischen Meldung von Ereignissen erzeugt werden können.

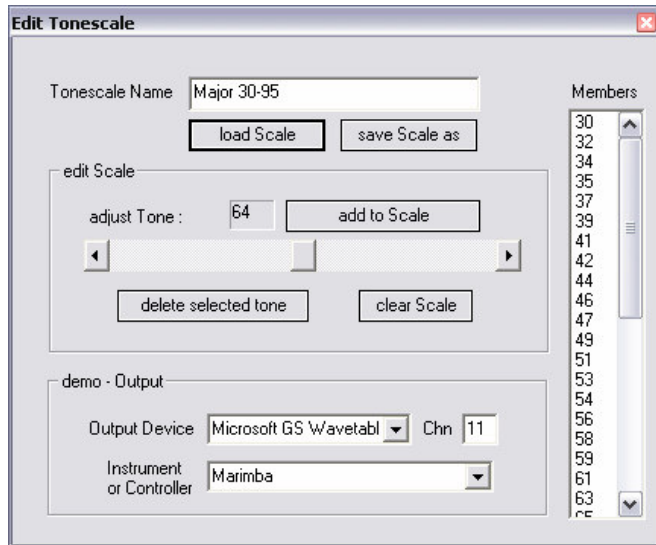


Abb. 7.5: Der Tonleitern-Editor

Der aktuelle Ton wird durch den Schieberegler bestimmt, eine Testausgabe erfolgt über das gewählte Instrument und Midi-Ausgabegerät. Durch das Betätigen von *Add to Scale* wird der aktuelle Ton zur Tonskala hinzugefügt, *Delete* löscht einen in der Liste markierten Ton. Durch eine Bewegung mit den Cursorstasten können Teile oder die gesamte Tonleiter angehört werden. Durch *Load Scale* und *Save Scale* können bestehende Skalen geändert und neue erzeugt werden

7.5 Die Erstellung von Design-Konfigurationen

Die Design-Konfiguration bestimmt die Art der Signalverarbeitung in *BrainBay*. Ihre Erstellung erfolgt grafisch und besteht aus den folgenden Schritten:

- Konfigurationselemente in das Design einfügen
- Signalverbindungen zwischen Elementen herstellen
- Einstellungen der Elemente durch deren Benutzerdialog anpassen
- Das Design mit Archiv-, Simulations- oder Live-Daten testen

7.5.1 Einfügen, Benennen und Löschen von Elementen

Der *Insert Element* - Eintrag des Menüs bietet vier Untergruppen von Elementen:

- Signalquellen (*Signal Sources*)
- Signalziele (*Signal Targets*)
- Verarbeitungselemente (*Processing - Elements*) und
- Andere (*Others*)

Die fünfte Gruppe (*All*) beinhaltet alle Elemente in alphabetischer Reihenfolge. Prinzipiell braucht eine vernünftige Design-Konfiguration mindestens eine Signalquelle (wie einen Biosignalverstärker, einen Signalgenerator oder ein Element zum Lesen von Archivdaten) und ein Signalziel (wie eine Spektralanzeige oder ein Oszilloskop). Ein Element wird durch einen *Links-Klick auf das Zentrum* eines Elementsymbols ausgewählt. Das aktuell gewählte Element wird mit einer gelben Umrandung dargestellt. Es kann durch ein Drücken der *Del*-Taste gelöscht werden. Durch *Doppelklicken auf das Zentrum* eines Elements kann die Beschriftung geändert werden.

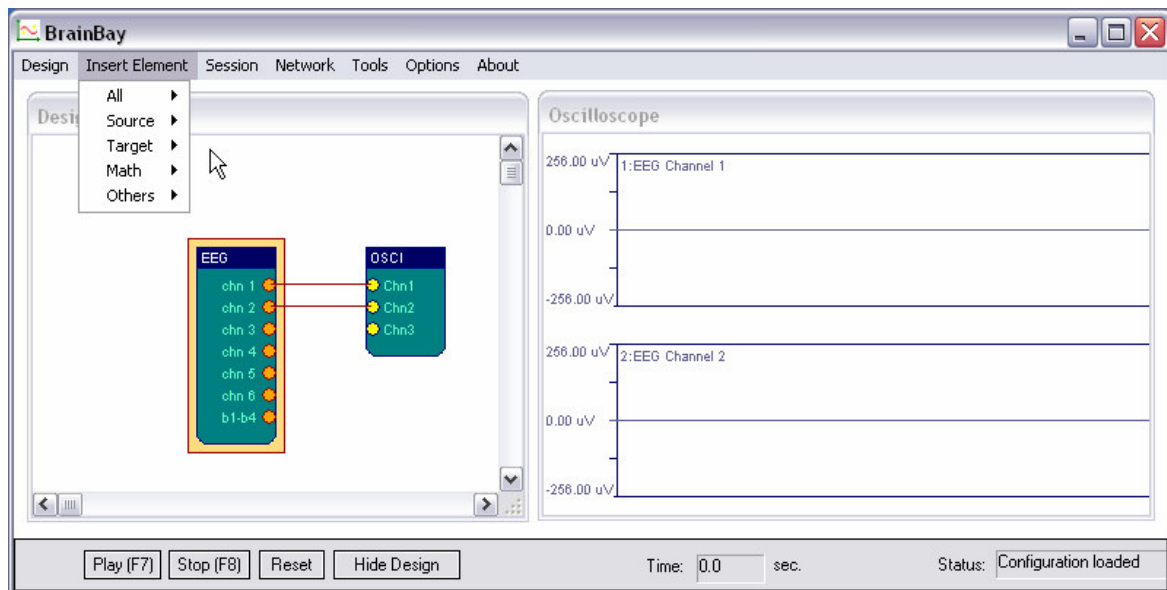


Abb. 7.6: Einfügen von Elementen in die Design-Konfiguration

7.5.2 Verbindung von Elementen

Eine Verbindung zweier Elemente wird erstellt, indem der *Output Port des Quellelements geklickt* und die Verbindung auf einen *Input-Port des Ziel-Elements* gezogen und dort abgelegt wird (*Drag and Drop*). Signalquellen haben mindestens einen Output-Port (orange). Die Abbildung zeigt ein EEG-Element mit sieben Output-Ports (sechs für Signale der Messkanäle, einer für die Stellung der Kontrolltaster). Signalziele haben mindestens einen Input-Port (gelb). Das dargestellte Oszilloskop-Element verändert die Anzahl der Input-Ports dynamisch, es können bis zu 32 Signale angeschlossen werden. Verarbeitungselemente haben meist Input- und Output-Ports.

7.5.3 Darstellung der Eigenschaften einer Verbindung

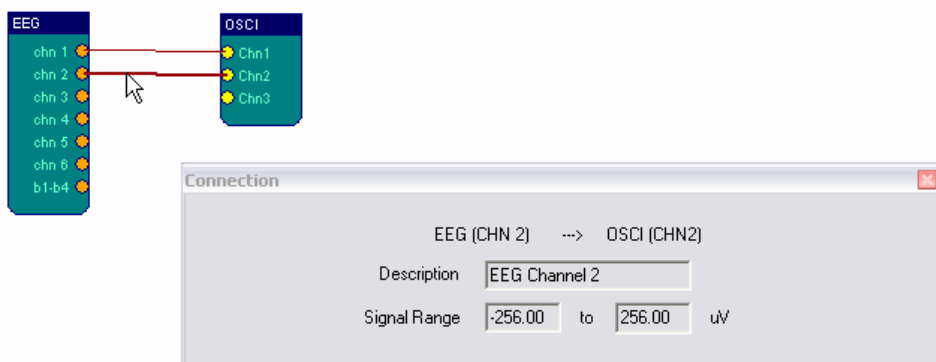


Abb. 7.7: Anzeige der Eigenschaften einer Elementverbindung

Durch einen *Linksklick einer Elementverbindung* werden Informationen wie die Namen der verbundenen Ports, die Bezeichnung der Verbindung, der Wertebereich und die Dimension des Signals in einem Dialogfenster dargestellt.

7.5.4 Änderung von Wertebereich und Signalbeschreibung

Die Verbindungseigenschaften werden durch das *Rechtsklicken eines Ports* geändert. Je nach Art des Ports wird ein *Out-Port*- oder *In-Port* - Dialog geöffnet. Dort können Signalbeschreibung und Wertebereich geändert werden. Falls ein Element Input-Ports und Output-Ports besitzt, können die Signaleigenschaften der Output-Ports von den Input-Ports geerbt werden. Auf diese Weise kann der Wertebereich eines Signals zu den folgenden Elementen propagiert werden.

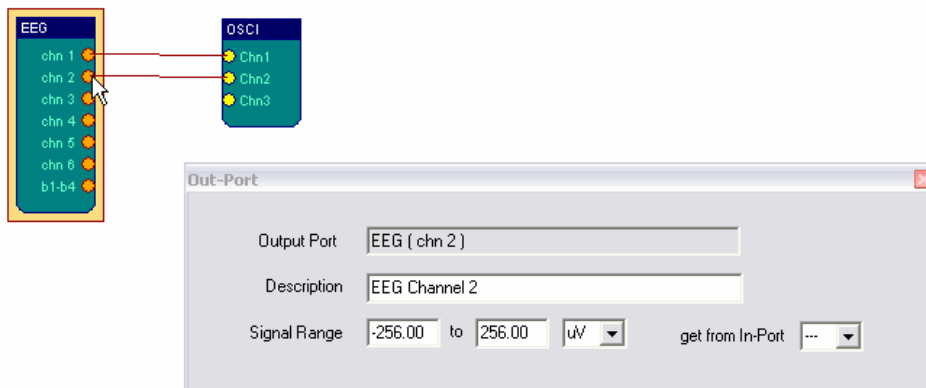


Abb. 7.8: Änderung von Signalbeschreibung und Wertebereich

Es kann nützlich sein, den Wertebereich für ein Element direkt an einem Input-Port zu korrigieren, weil manche Darstellungselemente wie das Oszilloskop, die Midi-Ausgabe oder der Multimedia-Player einen Wert auf die eingestellten Signalgrenzen beziehen. Dadurch kann ein bestimmter Bereich vergrößert oder hervorgehoben werden.

7.6 Verwendung der Konfigurationselemente

Die Konfigurationselemente (Signalquellen, Signalziele und Verarbeitungselemente) sind die funktionalen Einheiten einer Signalverarbeitung in *BrainBay*. Die Einstellungen der Elemente können durch ihren Benutzerdialog (das Parameterfenster) geändert werden. Der Benutzerdialog eines Elements wird durch einen *Rechtsklick in das Zentrum des Elementsymbols* dargestellt.

7.6.1 Signalquellen

7.6.1.1 Das EEG - Element

Dieses Element ist die Schnittstelle zur *OpenEEG* - Verstärkerhardware. Durch diese Biosignalverstärker werden bioelektrische Signale oder Sensorwerte verstärkt und digitalisiert. Derzeit werden die beiden Verstärkervarianten *ModularEEG* und *Mono-lithEEG* unterstützt. Beide Verstärker können mit unterschiedlicher Firmware und Übertragungsprotokollen betrieben werden: *P2*, *P3* und *P21*. *P21* ist ein bidirektionales Übertragungsprotokoll, damit auch Einstellungen an die Verstärkerhardware gesendet werden, wodurch etwa die Änderung von Übertragungsgeschwindigkeit oder Samplerate bzw. das Starten und Stoppen der Übertragung von der Software aus möglich werden. Die Messwerte der einzelnen Biosignalkanäle sind bei laufender Übertragung an den Output-Ports des EEG-Elements verfügbar.

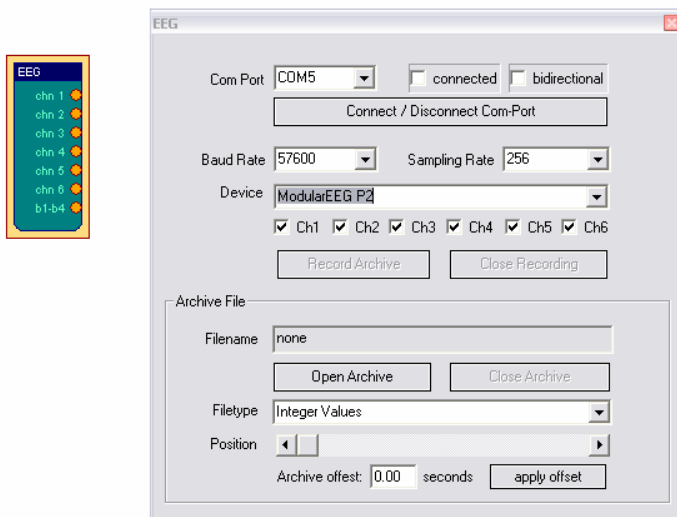


Abb. 7.9: Parameterdialog des EEG-Elementes

Die Abbildung zeigt den Benutzerdialog des EEG-Elements.

Neben der Verwendung als Live-Datenquelle für die Werte des Biosignalverstärkers kann das EEG-Element auch Archivdateien aufzeichnen und wiedergeben. Wird eine Archivdatei zur Wiedergabe geöffnet, stehen keine Live-Daten zur Verfügung und umgekehrt.

Empfang von Live-Daten des Biosignalverstärkers

Zur Vorbereitung der Datenübertragung wird der Com-Port gewählt, mit dem die Verstärkerhardware verbunden ist, und die Baudrate und Samplerate werden eingestellt (normalerweise 57600 Baud, 256Hz). Weiters ist die Firmware-Version des Verstärkers einzustellen (normalerweise ModularEEG-P2 oder ModularEEG-P3). Der Com-Port wird durch Drücken von *Connect/Disconnect Com Port* geöffnet. War das Öffnen erfolgreich, erscheint eine Markierung im Auswahlfeld *connected*, andernfalls wird eine Fehlermeldung ausgegeben, und die Com-Port Einstellungen sollten überprüft werden.

Um den Empfang von Live-Daten zu starten, wird die *Play* - Schaltfläche der Statuszeile gedrückt und der Biosignalverstärker eingeschaltet. Eine Darstellung der empfangenen Signale erfolgt durch das Einfügen eines Oszilloskop-Elements in die Design-Konfiguration und die Verbindung der Signalpfade von EEG- zu Oszilloskop-Element.

Aufzeichnung von Live-Daten in eine Archivdatei

Die Verbindung zum Biosignalverstärker wird wie oben beschrieben hergestellt. Durch die Schaltfläche *Record Archive* erfolgt die Auswahl eines Verzeichnisses und Dateinamens für die Archivdatei (.arc). Das Format der Archivdatei entspricht den Werten des Übertragungsprotokolls, die von der Firmware des Gerätes erhalten werden. Eine Archivdatei kann nur vom EEG-Element wieder gelesen werden. (Im Unterschied dazu können Archive, die mit dem *EDF-FileWriter* - Element erzeugt werden, auch von anderen Programmen importiert werden). Die Aufzeichnung startet nach Bestätigung einer zulässigen Datei bzw. durch das Starten der Session. Durch Auswahl von *Close Recording* wird die Aufnahme beendet und die Archivdatei geschlossen.

Wiedergabe aufgezeichneter Archivdateien:

Vor der Wiedergabe von Archiven durch das EEG-Element muss ein eventuell geöffneter Com-Port geschlossen werden. Dann kann durch Auswahl von *Open Archive* eine bestehende Archivdatei geöffnet werden. Die Einstellungen für Übertragungsprotokoll und Samplerate werden entsprechend der Archivdatei geändert. Durch das erfolgreiche Öffnen einer Archiv-Datei wird die Statuszeile erweitert: Ein zusätzlicher Schieberegler zeigt den Fortschritt der Archivwiedergabe und dient auch zum direkten Positionieren innerhalb des Archives.

Die aktuelle Position im Archiv wird im Feld *Time* angegeben. Die Schaltflächen *Play*, *Stop* und *Reset* kontrollieren die Wiedergabe.

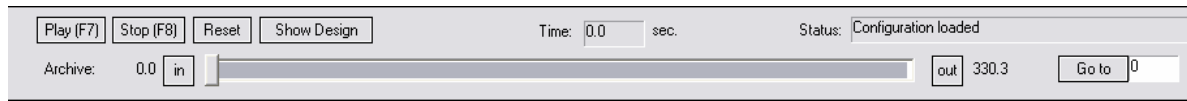


Abb. 7.10: Wechsel der Position innerhalb einer Archivdatei mithilfe der Statuszeile

Mittels der *In-* und *Out-* Schaltflächen kann ein Segment des Archives zur wiederholten Wiedergabe ausgewählt werden. Dies ist sinnvoll, um bestimmte Bereiche des Archives genauer untersuchen zu können. Das Feld *Go to* wird verwendet um an eine bestimmte Position (in Sekunden) der Archivdatei zu springen.

In einer Design-Konfiguration kann derzeit nur ein *EEG-Element* aktiv sein. Durch das *EDF-FileReader* Element ist es dennoch möglich, mehrere Archive (im *European Data Format*) zu öffnen, zu kombinieren bzw. gleichzeitig abzuspielen, da diese Einschränkung für das *EDF-FileReader* - Element nicht gilt. Um den Beginn eines Archivs in Bezug zu anderen geöffneten Archiven festzulegen, wird die Einstellung *Archive Offset* im *EEG-* bzw. *EDF-FileReader* - Element verwendet. Die Anzeige des Session-Fortschritts in der Statuszeile bezieht sich dabei immer auf die Gesamtlänge aller Archive, die momentan geöffnet sind.

Falls ein *EEG-Element* zum Empfang von Live-Daten verwendet wird, richtet sich die Geschwindigkeit der gesamten Verarbeitung nach der Ankunft dieser Daten.

Wird eine Archivdatei wiedergegeben, richtet sich die Wiedergabegeschwindigkeit nach dem gespeicherten Wert der Archivdatei, kann aber durch eine Änderung der Samplerate im Dialogfenster *Options* geändert werden.

7.6.1.2 Das *Signal-Generator* - Element

Der *Signal-Generator* produziert Sinus-, Rechteck- oder Sägezahn-Signale mit einstellbarer Frequenz, Amplitude, Phasenverschiebung und Offset. Die Art des Signals wird über die Auswahlbox festgelegt, die anderen Werte über Schieberegler. Die maximale Amplitude (normalerweise 500) kann über die Einstellungen des Output-Ports geändert werden (durch Rechtsklick auf den Port).

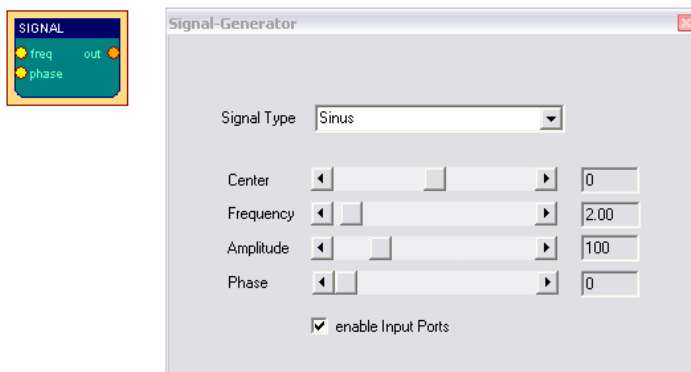


Abb. 7.11: Parameterdialog des *Signal-Generator*-Elementes

Das Auswahlfeld *enable Input Ports* kontrolliert die Verfügbarkeit der Einstellungen von Frequenz und Phase durch Input-Ports. Durch die Verwendung der Input-Ports können diese Werte durch andere Elemente beeinflusst werden.

7.6.1.3 Das *Constant* - Element

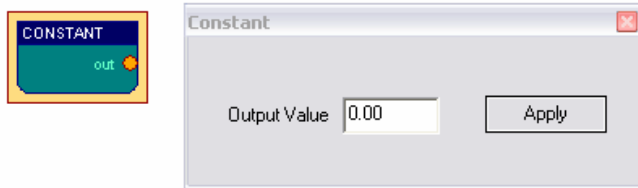


Abb. 7.12: Parameterdialog des Constant-Elementes

Dieses Element gibt einen fixen Wert am Output-Port aus, der durch Betätigung von *Apply* aktualisiert wird.

7.6.1.4 Das *EDF-FileReader* - Element

Das *EDF-FileReader* Element dient zum Öffnen und zur Wiedergabe gespeicherter Signalarchive im *European Data Format*. *EDF* ist ein gebräuchliches Format zum Austausch von Biosignalinformationen [EDF92]. Im *Header* einer *EDF*-Datei werden eine Reihe von Informationen zur Aufnahme wie Datum, Personendaten, Aufnahmegerät, Samplerate, Datensegmente, Wertebereiche und Beschreibungen der Signalkanäle gespeichert. Diese Informationen werden nach dem Öffnen einer *EDF*-Datei im Benutzerdialog des *EDF-FileReader*-Elements dargestellt.

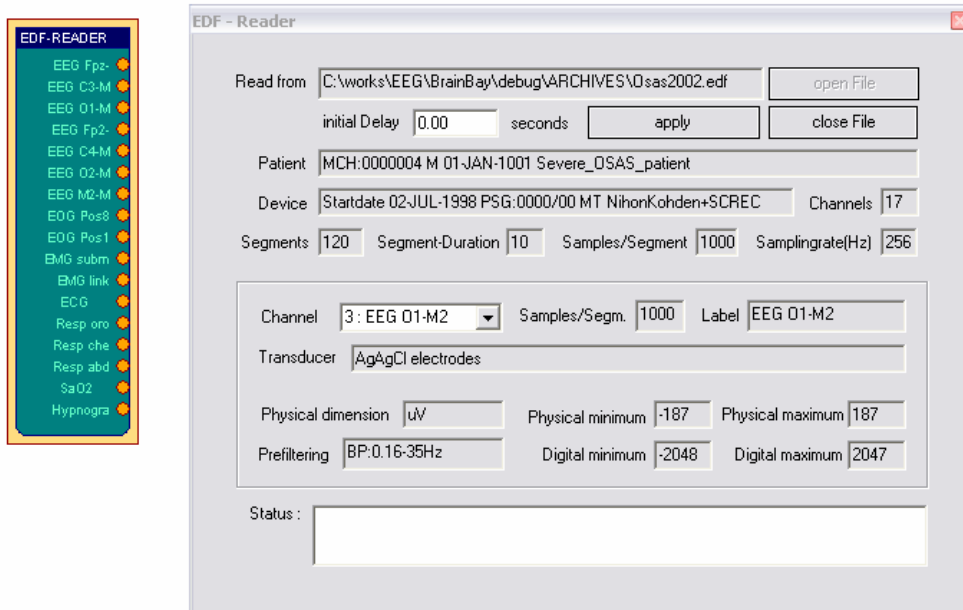
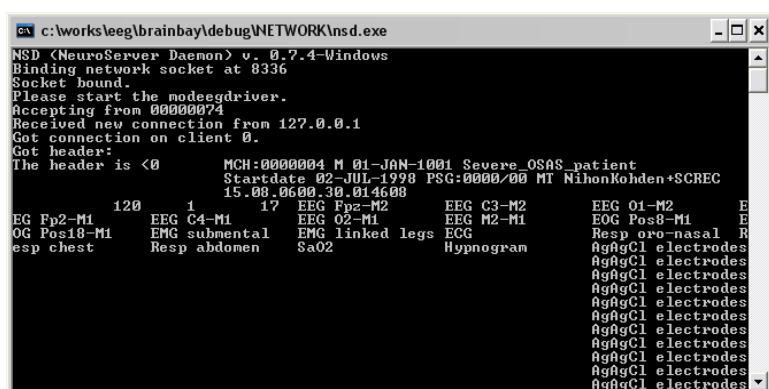


Abb. 7.13: Parameterdialog des *EDF-FileReader*-Elementes

Abb. 7.13 zeigt das *EDF-FileReader* Element nach dem Öffnen eines Archivs mit 17 Kanälen. Um die Informationen zu einem bestimmten Kanal darzustellen, wird dieser in der Auswahlbox *Channel* gewählt. *EDF*-Dateien können sehr viele Kanäle beinhalten, das Kanallimit in *BrainBay* ist derzeit auf 32 eingestellt. Der Wertebereich für die Output-Ports des *EDF-FileReader* Elements wird entsprechend der Werte *Physical Minimum* und *Physical Maximum* pro Kanal eingestellt. Ähnlich wie beim Öffnen einer Archivdatei durch das *EEG*-Element wird die Statuszeile für die Archivkontrolle erweitert (siehe *EEG*-Element 7.6.1.1). Die Position bezüglich anderer geöffneter Archivdateien wird durch das Feld *Initial delay* und Betätigung der Schaltfläche *Apply* eingestellt.

7.6.1.5 Das *TCP-Receive* - Element

Das *TCP-Receive* Element empfängt Signalinformationen über eine Netzwerk (*TCP*) Verbindung. Dazu versucht das Element, eine Verbindung zum *Neuroserver* Netzwerk-Service aufzunehmen. *Neuroserver* ist eine Softwareentwicklung von Rudi Cilibrasi, die ebenfalls Teil des *OpenEEG* -Projektes ist [CIL03]. Zur Verwendung des *TCP*-Elements muss ein *Neuroserver* - Service auf einem bekannten Computer im Netzwerk (Internet oder *Local Area Network, LAN*) oder am lokalen Computer (*localhost*) laufen. *Client*- Programme können Daten an den Neuroserver senden oder Daten von ihm empfangen. Die Übertragung basiert auf dem EDF-Standard für die Annotation und Speicherung der Signale. So können Live-Signale oder Aufzeichnungen weitläufig über Internet ausgetauscht werden.



Der einfachste Weg, das *TCP-Receive* Element zu testen, besteht im Starten eines Neuroservers am lokalen Computer. Dies erfolgt durch Auswahl des Menüpunktes *Network->Start Neuroserver* im Hauptmenü.

Abb. 7.14: Laufender Neuroserver mit aufgebauter Verbindung

Je nach Verbindungsgeschwindigkeit treten bei der Übertragung mittels Netzwerk Latenzzeiten auf. Diese betragen im LAN wenige Millisekunden, können aber bei Internetverbindungen erheblich größer werden. Durch die Schaltfläche *Empty Receive-Buffer* im Benutzerdialog des Elements werden aktuell gespeicherte Werte im Eingangs-Puffer gelöscht und die nächsten empfangenen Werte gehen sofort an die Output-Ports.

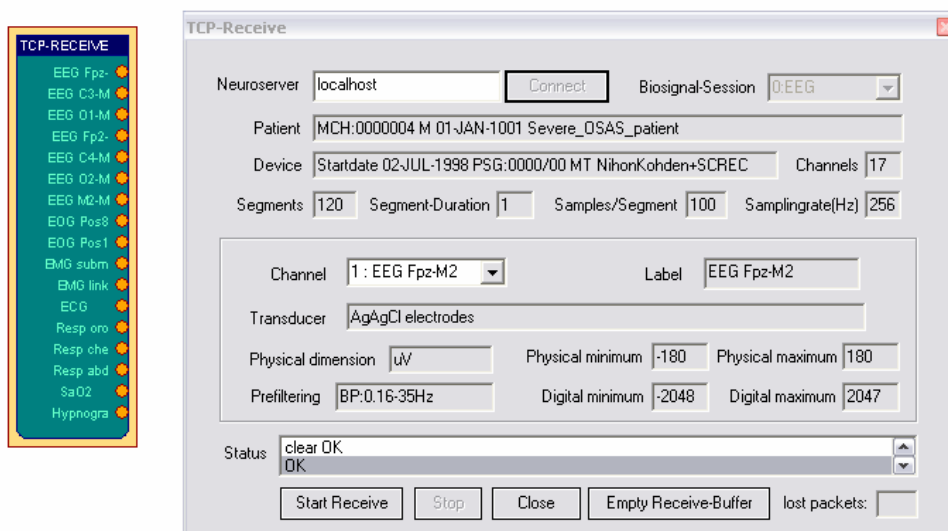


Abb. 7.15: Parameterdialog des *TCP-Receive*-Elementes

Verbindung des TCP-Receive Elements zum Neuroserver

Der Aufbau der Verbindung zum Neuroserver erfolgt durch die Eingabe der IP-Adresse des Host-Computers auf dem der Service läuft. Die Adresse wird im Feld *Neuroserver* eingetragen. Falls der Neuroserver auf demselben Computer läuft, ist hier *localhost* einzutragen. Durch Auswahl der *Connect* - Schaltfläche wird ein Verbindungsversuch ausgeführt. Im Falle eines erfolgreichen Verbindungsaufbaus erscheint im *Statusfeld* des Elements die Zeile 'OK' und die Auswahlbox *Biosignal-Session* wird mit den verfügbaren Datenströmen (*Streams*) des Neuroservers gefüllt. Dies können sowohl Archive oder Live-Übertragungen von Signalen sein. Die Auswahl einer Eintragung öffnet den zugehörigen *Stream*, stellt die EDF-Headerinformation im Benutzerdialog dar und aktualisiert die Output-Ports des Elements. Die Wertebereiche der Output-Ports richten sich nach den Kanaleinstellungen der EDF-Daten.

Während des Datenempfanges wird die Anzahl empfangener Datenpakete im *Statusfeld* aktualisiert. Durch Betätigung von *Stop* wird die Übertragung unterbrochen, *Close* schließt die Verbindung zum Neuroserver.

7.6.1.6 Das *Camera* - Element (Webcam)

Dieses Element kann zur Verarbeitung des Bilddatenstromes einer Webcam oder Digitalkamera verwendet werden. Die Kamera muss mit dem Computer verbunden und im System über Gerätetreiber angemeldet sein. Das *Camera*-Element erlaubt die Wiedergabe und Aufzeichnung der Bildinformation sowie die Anwendung bestimmter Algorithmen zur Merkmallokalisation und Bewegungsverfolgung. Zur Bildverarbeitung wird die *Intel Open Computer Vision library* verwendet (*OpenCV*) [OPC06].

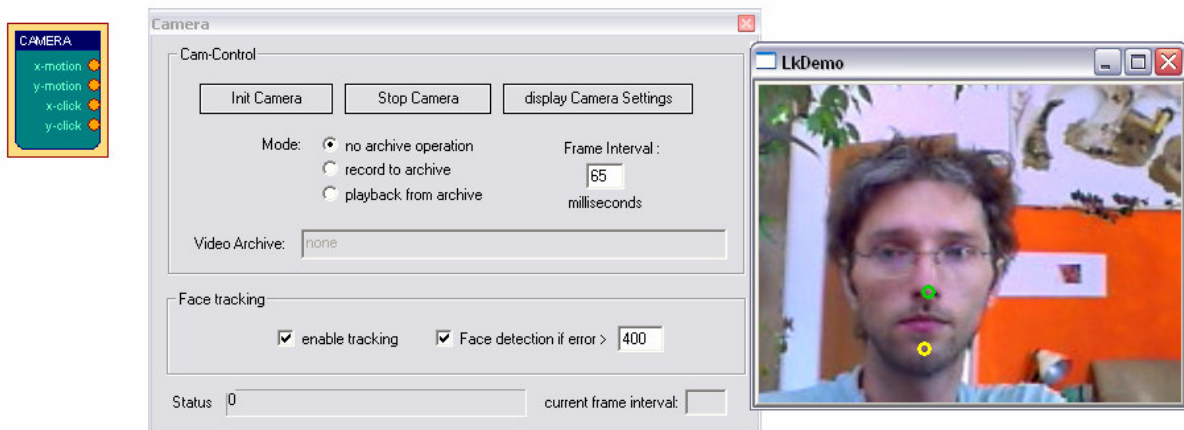


Abb. 7.16: Parameterdialog des Camera-Elementes, Live-Bild

Nach dem Einfügen eines *Camera*-Elements in die Design-Konfiguration wird die Verbindung zur Kamera hergestellt und das Live-Bild wird in einem separaten Fenster dargestellt. Das Zeitintervall für den Empfang von Einzelbildern (*Frames*) wird durch den Wert von *Frame Interval* eingestellt. Das Intervall wird in Millisekunden angegeben, Um eine sinnvolle Weiterverarbeitung von Gesichtspositionen zu ermöglichen, sollte ein Intervall von nicht mehr als 65 Millisekunden (etwa 15 Bilder pro Sekunde) gewählt werden. Mit einer geeigneten Webcam und einem Computer mit ca. 2 GHz CPU-Takt und USB2.0 Verbindung können Frameraten von ca. 25 Bildern pro Sekunde erreicht werden.

Die Betätigung der Schaltfläche *Display Camera Settings* zeigt das Einstellungsfenster für Kameraoptionen, wo Parameter wie Helligkeit und Kontrast, Spiegelung des Bildes usw. justiert werden können. Drei Auswahlknöpfe regeln den Modus des Camera-Elements: '*No archive operation*' wird verwendet, wenn das Live-Bild nur dargestellt bzw. für Bewegungserkennung verwendet werden soll. Wird '*Record to archive*' gewählt, erfolgt eine Videoaufzeichnung in eine AVI-Datei. Diese Datei und die zu verwendende Komprimierungsvariante (*Codec*) sind durch Dialogfenster zu wählen. Die Wahl des Codec entscheidet über Qualität der Aufzeichnung und über das Verhältnis Rechenzeit/benötigte Plattenkapazität. Für die Echtzeitanwendung ist ein Verfahren zu bevorzugen, welches wenig Rechenzeit braucht. Die unkomprimierte Speicherung ist am schnellsten, produziert aber sehr große Dateien, '*Cinepak*' oder '*DiVX5.2*' sind mögliche getestete Alternativen. Nach der Wahl des Codecs werden bei laufender Ausführung der Design-Konfiguration Videobilder in der Datei abgelegt. Durch die Betätigung der *Stop Camera* - Schaltfläche werden die Aufnahme beendet und die Videodatei geschlossen.

Die Auswahl von '*Playback from archive*' ermöglicht die Wiedergabe einer AVI Datei anstatt der Verwendung der Live-Kamera. Die Videodatei wird durch ein Dialogfenster gewählt, sie kann von einer früheren Aufzeichnung durch das *Camera*-Element stammen oder eine beliebige AVI-Datei sein, die mit einem verfügbaren Codec dekodiert werden kann. Nach dem Start der Session wird das Video von der Datei mit der eingestellten Framerate geladen. Falls die Merkmalslokalisation aktiviert ist, wird diese auf Basis der Videodaten aus der Datei durchgeführt.

Merkmallokalisation und Bewegungsverfolgung

Wenn der Auswahlbutton '*enable tracking*' aktiviert ist, führt das Camera-Element eine Gesichtsdetektion durch. Wird ein Gesicht erkannt, werden die Positionen von Nase und Kinn geschätzt. Die aktuellen Bewegungen von Nase und Kinn können an den Output-Ports des Elements als relative Werte zum vorigen Kamerabild ausgelesen werden. Durch geeignete Filter- und Skalierungsvorgänge im weiteren Verlauf der Signalverarbeitung dienen sie als Eingabewerte für das *Mouse-Controller* Element und können so zur Steuerung des Maus-Cursors und für Klick-Operationen verwendet werden. Das *Camera*-Element erlaubt somit die Herstellung universeller Konfigurationen für einen Betrieb der Computermaus durch Kopfbewegungen. (Details dazu siehe Anwendungsbeispiel *Kamera-Maus* 8.5)

7.6.1.7 Das *Skindialog* - Element

Das *Skindialog* Element ermöglicht die Einbindung eines grafischen Freiform Benutzerdialogs (*skinned Dialog Window*). Der Dialog wird aufgrund einer Bitmap-Datei erstellt, die nicht unbedingt viereckig sein muss, da durch eine Maske die durchsichtigen Bereiche des Dialogs angegeben werden. Als funktionale Komponenten können Schaltflächen und Schieberegler sowie Textfelder verwendet werden. Die aktuellen Werte der Schieberegler und Schaltflächen werden an den Output-Ports des Elements ausgegeben. So können spezielle (zB. sehr kleine oder runde) Benutzerdialogfenster für die Steuerung relevanter Einstellungen einer komplexen Konfiguration erzeugt werden.

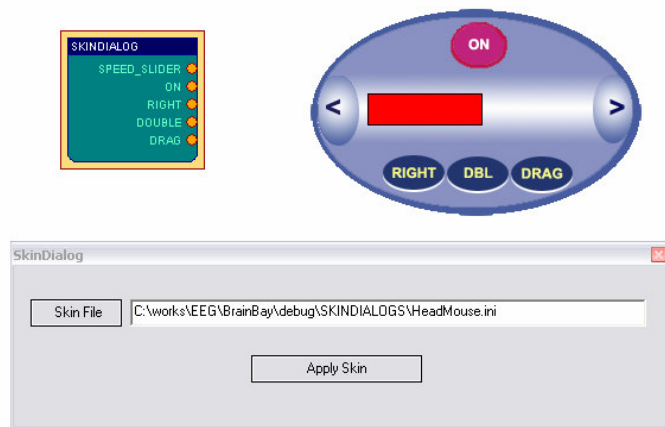


Abb. 7.17: Parameterdialog des Skindialog-Elementes

Die Ressourcendateien für den Dialoghintergrund und die Maske, sowie die aktiven Bereiche für die Dialogelemente sind in einer *.ini*-Datei angegeben. Derzeit muss diese *.ini*-Datei noch manuell (mit einem Texteditor) erzeugt werden. (Details dazu siehe Datei *skin_readme.txt* im Unterverzeichnis *Skindialogs*). Ein Editor zur komfortblen Erstellung dieser Datei ist als zukünftige Entwicklung geplant.

7.6.2 Verarbeitungselemente

Verarbeitungselemente dienen zur Transformation, Kombination oder Klassifizierung von Signalen innerhalb der Konfiguration. Sie haben einen oder mehrere Input- und Output-Ports. *BrainBay* stellt derzeit 19 Verarbeitungselemente zur Verfügung, die im Folgenden beschrieben werden.

7.6.2.1 Das Averager - Element

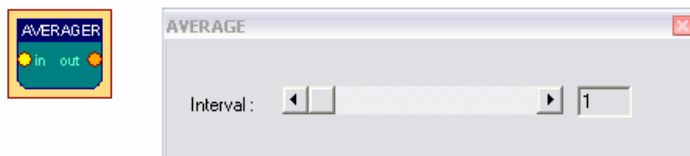


Abb. 7.18: Parameterdialog des Averager-Elementes

Das *Averager* Element gibt das arithmetische Mittel der letzten *n* Samples aus. Die Anzahl der Samples wird durch den Schieberegler *Interval* festgelegt.

7.6.2.2 Das Comparator - Element

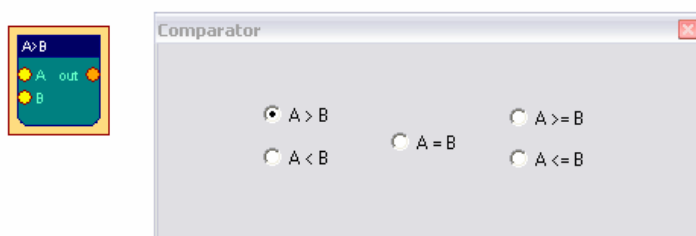


Abb. 7.19: Parameterdialog des Comparator-Elementes

Das *Comparator* - Element vergleicht die aktuellen Werte der beiden Input-Ports auf eine Weise, die durch den Benutzerdialog eingestellt werden kann. Falls die Bedingung erfüllt ist, wird der Wert von Input Port A am Output Port ausgegeben, ansonsten die Konstante *INVALID_VALUE*. Beträgt der Wert eines der Input Ports *INVALID_VALUE*, wird in jedem Fall auch *INVALID_VALUE* ausgegeben.

7.6.2.3 Das *Correlation* - Element

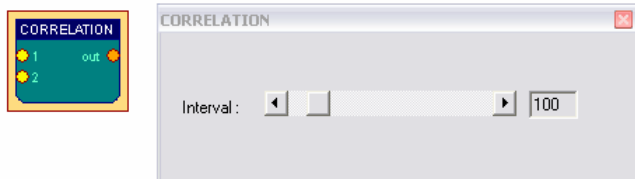


Abb. 7.20: Parameterdialog des Correlation-Elementes

Dieses Element berechnet die Kreuzkorrelation zwischen den zwei Eingangssignalen. Die Anzahl der verwendeten Werte wird durch den Schieberegler *Interval* eingestellt.

Die Korrelation ist ein Maß für den linearen Zusammenhang von zwei Signalen. Der Wertebereich der Ausgabe liegt zwischen 1 (volle positive Korrelation) 0 (unkorreliert) und -1 (volle inverse Korrelation).

7.6.2.4 Das *Counter / Display* - Element

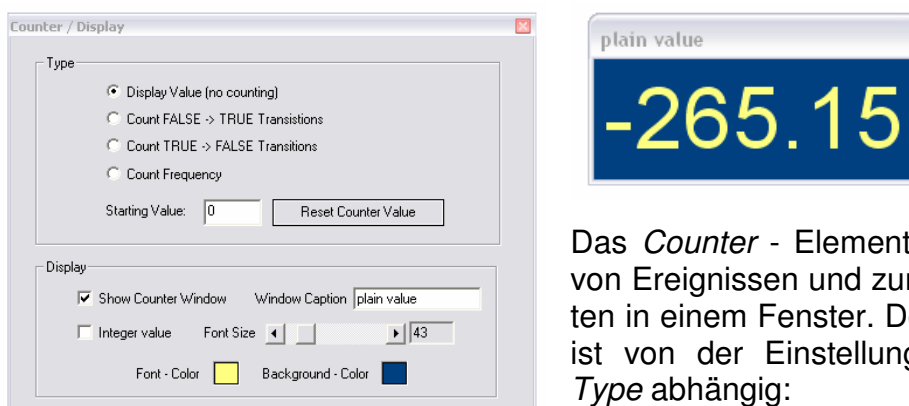


Abb. 7.21: Parameterdialog des Counter-Elementes

Das *Counter* - Element dient zum Zählen von Ereignissen und zur Anzeige von Werten in einem Fenster. Der angezeigte Wert ist von der Einstellung des Parameters *Type* abhängig:

Bei der Auswahl von *Display Value* wird kein Zählvorgang durchgeführt, sondern direkt der Wert des Input-Ports dargestellt. Bei Auswahl von *Count FALSE->TRUE Transitions* zählt das Element die Übergänge von *INVALID_VALUE* zu einem gültigen Wert, *Count TRUE->FALSE Transitions* zählt Übergänge zu *INVALID_VALUE*. Die Auswahl von *Count Frequency* berechnet die Frequenz derartiger Übergänge in Hz. Der aktuelle Wert wird am Output-Port des Elements ausgegeben und durch Auswahl von *Show Counter Window* in einem zusätzlichen Fenster mit wählbarer Schriftgröße und Farbe dargestellt. Durch die Auswahl von *Integer Value* erfolgt ein Runden der Ausgabe auf ganzzahlige Werte. Der Zähler kann durch Drücken von *Reset Counter Value* auf einen einstellbaren Wert zurückgesetzt werden, dieses Zurücksetzen erfolgt auch durch Betätigung von *Reset* in der Statuszeile.

7.6.2.5 Das *Debounce* - Element

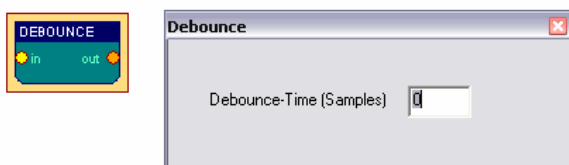


Abb. 7.22: Parameterdialog des Debounce-Elementes

Das *Debounce* - Element unterdrückt ein vermehrtes Aufkommen des Wertes *INVALID_VALUE* in einem Signal. Das Intervall zur Überprüfung wird durch den Benutzerdialog festgelegt.

Durch dieses Element kann ein "Hin- und Herspringen" verhindert werden, wenn sich der Signalwert nahe einem eingestellten Grenzwert bewegt und deshalb unerwünscht oft Aktionen (wie z.B. Mausklicks oder Töne) ausgelöst werden.

7.6.2.6 Das *ERP-Detect* - Element

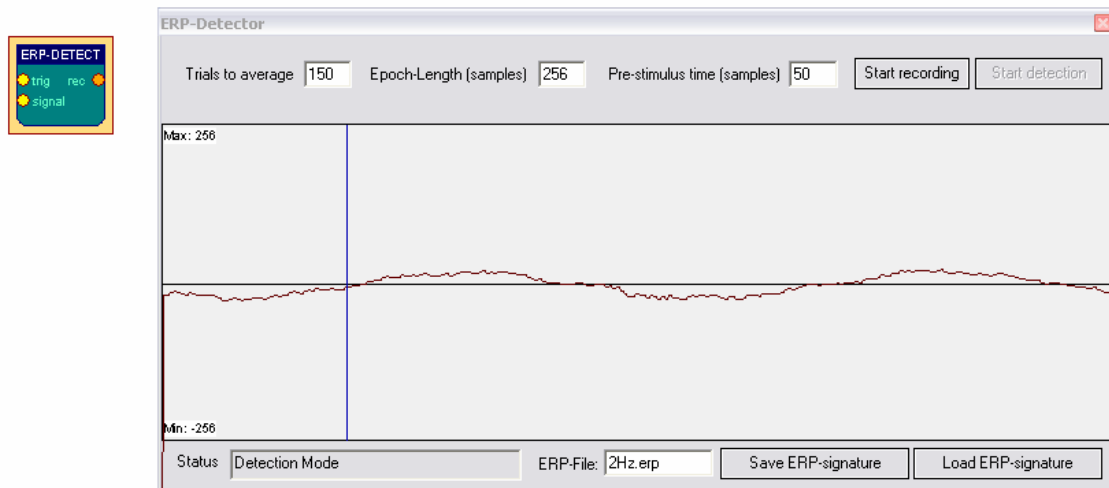


Abb. 7.23: Parameterdialog des ERP-Detect-Elementes

Das *ERP-Detect* - Element kann Segmente eines Signals aufzeichnen, mitteln und im Zeitbereich mit dem kontinuierlichen Datenstrom vergleichen. Die Technik der Mittelung mehrerer Versuche (*Trial Averaging*) ist eine populäre Methode in der Extraktion von *ereigniskorrelierten Potenzialen* (*EKPs*, *event related Potenzials*, *ERPs*). Durch die Mittelung um den definierten Zeitpunkt des Ereignisses fallen die zufälligen Artefakte und der Rausch-Anteil des Signals mit der steigenden Anzahl von Versuchen zunehmend heraus (siehe *ERPs* 2.1.4.1).

Die Länge des Intervalls für Aufzeichnung und Mittelung wird durch *epoch-length* angegeben (in Samples), die Anzahl der Versuche für die Mittelung durch *trials to average* und die Position des Nullpunktes durch *pre-stimulus time*. Der Wertebereich für die Anzeige des aufgenommenen Signals wird vom Input-Port *Signal* übernommen, die Einstellung kann durch *Rechtsklick* dieses Ports verändert werden.

Durch die Schaltfläche *start recording* wird die Aufzeichnung und Signalmittelung gestartet. In diesem Modus wartet das Element auf einen gültigen Wert (ungleich *INVALID_VALUE*) am Input-Port *trigger*. Dieses Trigger-Signal könnte z.B. von einem Element stammen, das einen Schwellwert überprüft (*Threshold-Element*) oder von einem externen Taster zur Markierung des Ereigniszeitpunktes. Sobald das Trigger-Signal empfangen wurde, werden die Messwerte aufgezeichnet und zum internen Puffer des Elements addiert. Dann wird auf den nächsten Triggerwert gewartet. Sobald alle Versuche aufgenommen wurden, wechselt das Element in den Modus zur Mustererkennung, und die aktuellen Werte des Signals werden mit dem aufgezeichneten Muster verglichen. Hierzu wird die Differenz mit dem aktuellen Signal gebildet und der Wert auf 0-100% normiert. Dieser Wert wird am Output Port des Elements ausgegeben. Das aufgezeichnete Segment kann in eine *.erp* -Datei gespeichert werden, was nötig ist falls es wieder verwendet werden soll.

7.6.2.7 Das *Expression-Evaluator* - Element

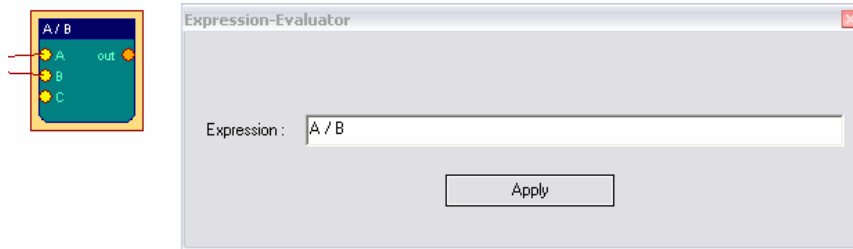


Abb. 7.24: Parameterdialog des Expression-Evaluator-Elementes

Das Element *Expression Evaluator* kann bis zu sechs Input-Ports haben, es dient zur Auswertung von Ausdrücken, welche die angeschlossenen Signale beinhalten. Die Eingabesignale werden mit *A, B, C, D, E, F* referenziert und können mit numerischen Konstanten, Elementarfunktionen, unären und binären Operatoren zu einem mathematischen Ausdruck kombiniert werden, der während der Laufzeit der Konfiguration ausgewertet wird. Das Ergebnis der Auswertung liegt am Output-Port des Elementes an. Die Implementierung des *Evaluator*-Elements basiert auf den Funktionen der freien Bibliothek *GNU-Libmatheval* [SAM04].

Unterstützte Elementarfunktionen:

Exponentialfunktion (exp), Logarithmus (log), Quadratwurzel (sqrt),
Sinus (sin), Cosinus (cos), Tangens (tan), Cotangens (cot), Secans (sec), Cosecans (csc),
Arcussinus (asin), Arcuscosinus (acos), Arcustangens (atan),
Arcuscotangens (acot), Arcussecans (asec), Arcuscotcans (acsc),
Sinus hyperbolicus (sinh), Cosinus hyperbolicus (cosh), Tangens hyperbolicus (tanh),
Cotangens hyperbolicus (coth), Secans hyperboicus (sech), Cosencans hyperbolicus (csch),
Arcussinus hyperbolicus (asinh), Arcuscosinus hyperbolicus (acosh),
Arcustangens hyperbolicus (atanh), Arcuscotangens hyperbolicus (acoth),
Arcussecans hyperbolicus (asech), Arcuscosecans hyperbolicus (acsch),
Absolutwert (abs), Heaviside-Funktion mit Wert 1 (step)

Unterstützte unäre Operation: Vorzeichenwechsel ('-').

Unterstützte binäre Operationen:

Addition ('+'), Subtraktion ('-'), Multiplikation ('*'),
Division ('/') und Exponentiation ('^').

Es gelten die gängigen mathematischen Regeln für die Ausführungsreihenfolge. Klammerung von Ausdrücken ('(' und ')') kann zur Änderung der Reihenfolge verwendet werden. Leerzeichen und Tab-Zeichen in der Zeichenkette des Ausdrucks sind erlaubt, Zeilenwechsel (*newline*) dürfen nicht vorkommen.

7.6.2.8 Das *Filter* - Element

Das *Filter* - Element erlaubt die Erstellung digitaler *Low-Pass, High-Pass, Band-Pass* und *Band-Stop* - Filter zur Signalverarbeitung. Ein *Low-Pass* - Filter filtert hohe Frequenzanteile eines Signals heraus und lässt die tieferen Frequenzen durch. Entsprechend lässt ein *High-Pass* Filter höhere Frequenzen durch und ein *Band-Pass*-Filter lässt Frequenzen innerhalb eines bestimmten Bereichs (*Frequenzband*) passieren. Ein *Band-Stop* - Filter filtert den Inhalt eines Frequenzbandes heraus und lässt umliegende Bereiche passieren. Der Filter kann *Bessel*- oder *Butterworth* - Charakte-

ristik haben. Die Filtertypen unterscheiden sich in der Präzision im Zeit- bzw. Frequenzbereich: Der Besselfilter hat einen flacheren Abfall (*roll-off*) im Frequenzbereich, was bedeutet, dass er in der Dämpfung im Bereich seiner Frequenzgrenze weniger scharf ist). Dafür zeigt der Bessel-Filter im Zeitbereich weniger Überschießen (*Ringing*) und weniger Latenzzeit. Die Einstellung des Typs (*Filter Type*), der Ordnung (*Filter Order*) und der Frequenzgrenzen erfolgt durch Eingaben im Benutzerdialog. Eine höhere Filter-Ordnung führt zu einer schärferen Filtercharakteristik im Frequenzbereich aber auch zu einer längeren Verzögerung im Zeitbereich. Brauchbare Werte für die Filterordnung sind 2 - 8. Durch die Betätigung der Schaltfläche *Apply* wird die Filterantwort (*response*) im Frequenzbereich angezeigt. Der Frequenzbereich für die Darstellung kann durch Eingaben in die entsprechenden Textfelder geändert werden.

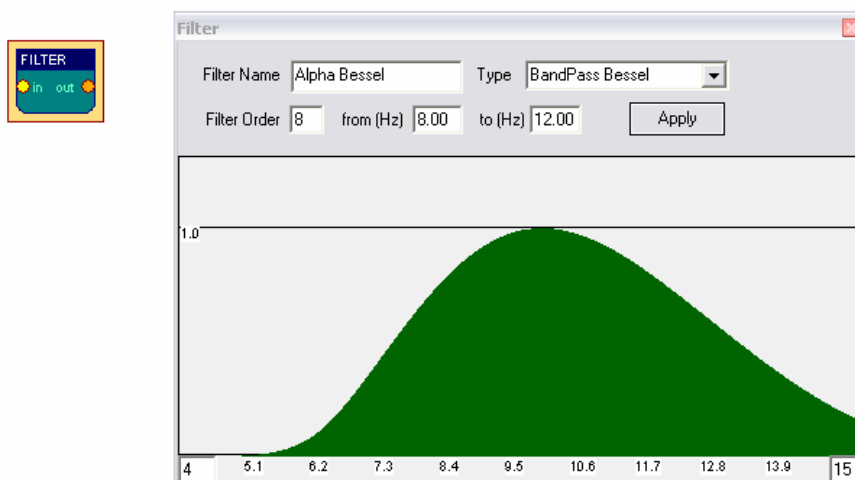


Abb. 7.25: Parameterdialog des Filter-Elementes

Der dargestellte Bandpass-Filter lässt Frequenzen um 10 Hz ohne Dämpfung passieren. Frequenzen unter 6 Hz oder über 14 Hz werden sehr stark gedämpft. Für eine weitere Verwendung von Filtern siehe *Kapitel 8, Anwendungsbeispiele*. Das Filter-Element basiert auf Funktionen der freien *FidLib*-Bibliothek von Jim Peters and Tony Fisher [PET03].

7.6.2.9 Das FFT - Element

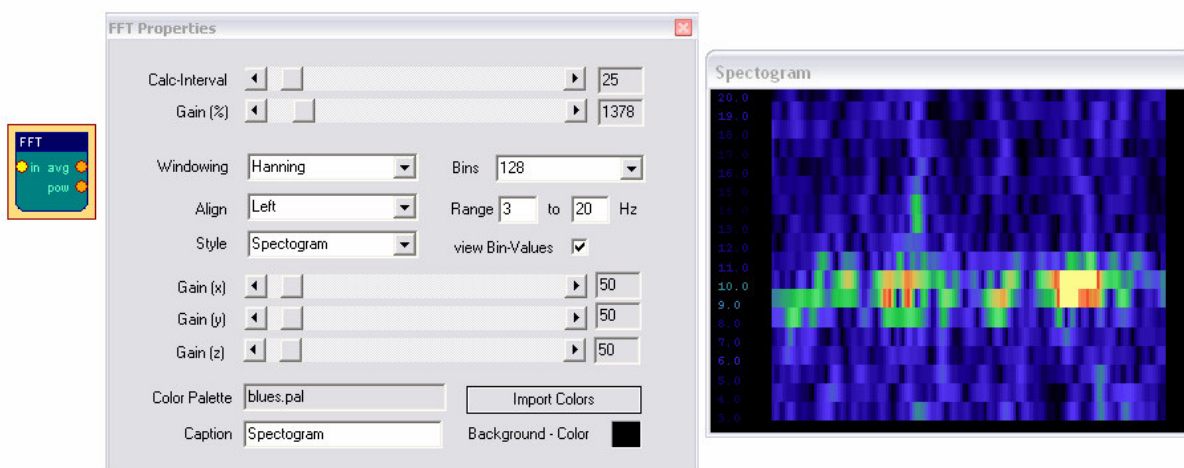


Abb. 7.26: Parameterdialog und Ausgabefenster des FFT-Elementes

Das *FFT* - Element führt eine *Fast Fourier-Transformation* durch und zerlegt das angeschlossene Signal dadurch in seine Frequenzkomponenten. (Vergleichbar mit dem Ohr, das uns verschiedene Tonhöhen zum selben Zeitpunkt hören lässt). Durch die Verwendung des *FFT* - Elements können Veränderungen der Frequenzkomponenten als Balkendiagramm, Spektrogramm oder 3D-Landschaft betrachtet werden. Die Art der Visualisierung ist durch die *Style* - Auswahlbox festzulegen.

Der Parameter *Calc-Interval* legt die Periode der Berechnung (in Samples) fest. Ein Intervall von 10 ergibt bei einer Samplerate von 200 Hz 20 Transformationen pro Sekunde. Die Anzahl der Frequenzkomponenten (*Bins*) legt die Auflösung im Zeit- und im Frequenzbereich fest. Viele Bins bedeuten eine gute Frequenzauflösung aber eine schlechte Auflösung im Zeitbereich. Zur Darstellung von Biosignalwerten mit Sampleraten von 256Hz sind 64 bis 256 Bins geeignet, was einen Kompromiss aus Genauigkeit in Zeit und Frequenz bedeutet. Der Parameter *Range* wählt die Frequenzbänder zur Darstellung aus, indem obere und untere Grenze in Hz angegeben werden. *Align* (*left, right, bottom*) rotiert das Display. Bei einer Darstellungsform als Spektrogramm oder als 3D-Ansicht ist eine Farbpalette zur Darstellung der Intensitätswerte zu wählen. Bei der Verwendung der Balkengrafik dient die Farbpalette zur Unterscheidung verschiedener Frequenzbereiche.

Die *Gain (%)* - Einstellung dient zur Verstärkung oder Dämpfung des Eingangssignals. Die Werte für *Gain(x)*, *-y* und *-z* beeinflussen die Darstellung der berechneten Frequenzkomponenten in der 2D- und 3D- Ansicht. Wenn die 3D -Ansicht verwendet wird, können der Betrachtungswinkel und die Betrachtungsposition durch Klicken und Ziehen mit dem linken und rechten Mausknopf geändert werden.

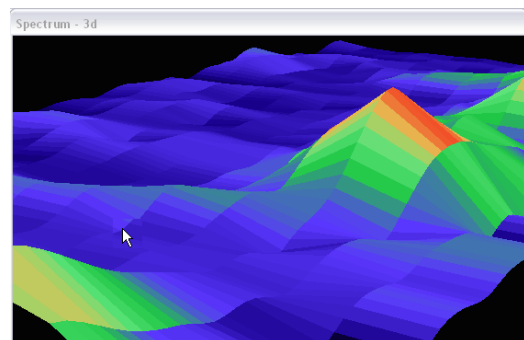


Abb. 7.27: 3D-Darstellung durch das *FFT*-Element

Das *FFT*-Element besitzt zwei Output-Ports. Am ersten Port (*Avg*) lässt sich die Durchschnittsfrequenz der durch *Range* selektierten Frequenzbänder ablesen. Der zweite Port (*Power*) gibt die gemittelte Aktivität der gewählten Frequenzbänder aus.

7.6.2.10 Das *Integrator* - Element

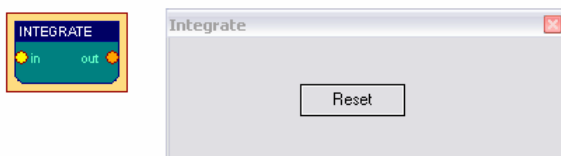


Abb. 7.28: Parameterdialog des *Integrator*-Elementes

Das *Integrator* -Element addiert kontinuierlich die ankommenden Werte. Durch Betätigung von *Reset* wird der Integrationswert auf null gesetzt. Dies geschieht auch durch das Starten oder Rücksetzen der Session (durch die Betätigung von *Start* oder *Reset* in der Statuszeile).

7.6.2.11 Die Elemente *And*, *Or* und *Not*

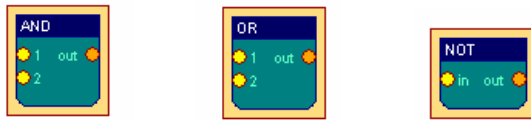


Abb. 7.29: Symbole der AND- OR- und NOT-Elemente

Die Elemente *And*, *Or* und *Not* haben keine Benutzerdialoge. Diese Elemente führen logische Operationen mit den Inputsignalen durch und geben das Ergebnis am Output-Port aus.

And gibt TRUE_VALUE aus, wenn keiner der Inputwerte INVALID_VALUE ist, *Or* gibt TRUE_VALUE aus, wenn höchstens ein Inputwert INVALID_VALUE ist, *Not* gibt TRUE_VALUE aus, der Inputwert INVALID_VALUE ist.

In allen anderen Fällen wird INVALID_VALUE ausgegeben.

7.6.2.12 Das *Matlab* - Element

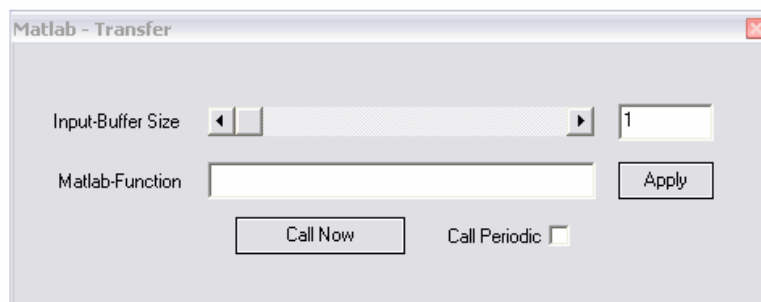


Abb. 7.30: Parameterdialog des Matlab-Elementes

Das *Matlab* - Element kann zum Transfer von Signalinformationen zur *Matlab*-Applikation und zum Empfang einer Rückmeldung verwendet werden. Die Größe des zu übertragenden Wertepuffers wird durch den Schieberegler des Benutzerdialoges eingestellt. Der Datenaustausch während der Laufzeit wird durch die *Matlab-Engine* ermöglicht. Die hierfür nötigen *.dll*-Dateien sind Teil der *Matlab*-Distribution. Das *Matlab* - Element kann nur verwendet werden, wenn sich eine lizenzierte Version von *Matlab* mit den entsprechenden *.dll* und *.lib* - Dateien auf dem System befindet.

Das Element kann bis zu sechs Input-Ports verarbeiten, die mit *A*, *B*, *C*, *D*, *E* und *F* angesprochen werden. Während der Ausführung werden die Input-Werte in Puffern der angegebenen Länge gespeichert. Nach dem Aufruf von *Matlab* durch Betätigung von *call now* können in der *Matlab*-Umgebung Datenfelder mit den letzten *n* Werten der Input-Ports als Variablen *A*, *B*, *C*,... angesprochen werden, und die angegebene *Matlab*-Funktion wird ausgeführt. Das Ergebnis einer Berechnung wird in Form der *ANS*-Variable an *BrainBay* zurücktransferiert und kann am Output-Port des *Matlab*-Elements ausgelesen werden. Durch eine Auswahl von *Call Periodic* wird der Transfer periodisch mit der Samplefrequenz durchgeführt. Dies ist nur bei *Matlab*-Funktionen sinnvoll, die sehr wenig Rechenzeit benötigen. Details zur *Matlab-Engine* siehe [MAT06].

7.6.2.13 Das *Magnitude* - Element

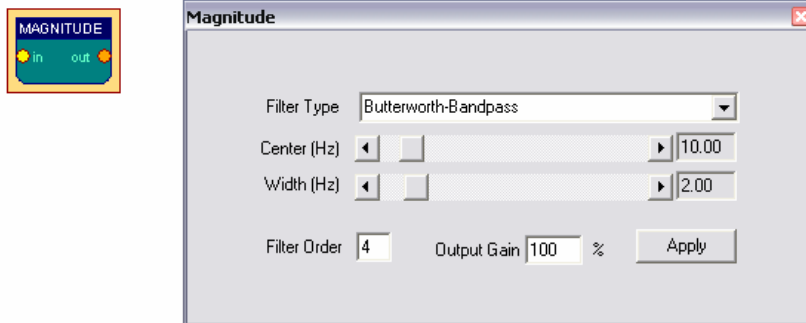


Abb. 7.31: Parameterdialog des Magnitude-Elementes

Das *Magnitude* - Element wird zur Ermittlung der Aktivität in einem bestimmten Frequenzband des Signals verwendet. Diese Funktion wird häufig mit 'Power in Passband' bezeichnet. Durch die Auswahlbox *Filter Type* können *Bessel-Bandpass* oder *Butterworth-Bandpass* gewählt werden (für Details zur Filtereinstellung siehe *Filter* - Element 7.6.2.8 bzw. Jim Peter's *FiView* - Applikation [PET03]).

Das gewünschte Frequenzband wird durch Mitte (*center*) und Breite (*width*) in Hz angegeben. Soll z.B. die *Alpha-Aktivität* eines Gehirnwellensignals im Bereich von 8 bis 12 Hz gemessen werden, wird eine Mitte von 10 Hz und eine Breite von 4 Hz gewählt. Die Filter-Ordnung (*order*) bestimmt die Schärfe des Filters im Frequenzbereich. Eine höhere Ordnung führt allerdings zu einer langsameren Reaktion im Zeitbereich. Das *Magnitude*-Element gibt die Aktivität im eingestellten Frequenzband aus, ein Verstärkungsfaktor (*Output-Gain*) dient zur Skalierung der Ausgabe (in Prozent).

7.6.2.14 Das *Mixer* - Element

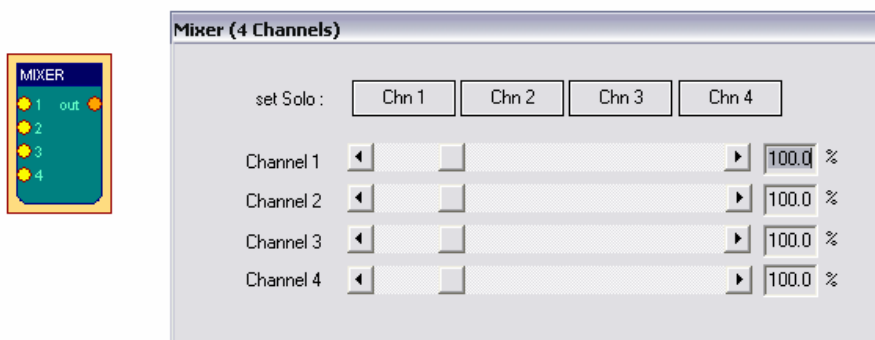


Abb. 7.32: Parameterdialog des Mixer-Elementes

Das *Mixer* - Element erlaubt die Zusammenführung von bis zur vier Signalen mit regelbaren Gewichtungsfaktoren in ein Signal, das am Output-Port ausgegeben wird. Die Verstärkungsraten für die einzelnen Eingabesignale werden durch die Schieberegler *Chanel 1* - *Chanel 4* in Prozent angegeben. Durch Betätigung der Schaltflächen *Chn1* - *Chn4* wird der entsprechende Kanal exklusive ausgegeben.

7.6.2.15 Das *Sample/Hold* - Element

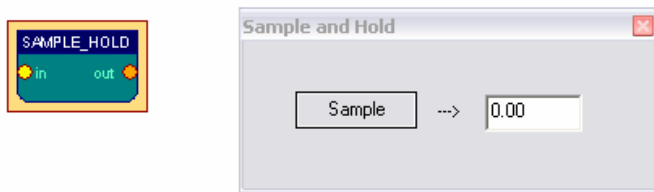


Abb. 7.33: Parameterdialog des Sample/Hold-Elementes

Das *Sample-Hold* - Element speichert den aktuellen Wert am Input-Port, wenn die Schaltfläche *Sample* betätigt wird. Dieser Wert wird daraufhin kontinuierlich am Output-Port ausgegeben.

7.6.2.16 Das *Standard-Deviation* - Element

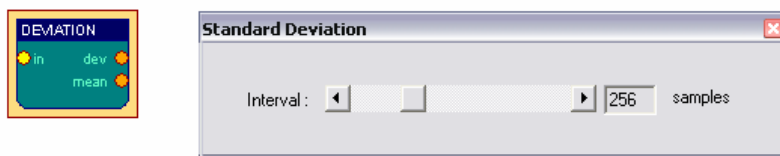


Abb. 7.34: Parameterdialog des Standard-Deviation-Elementes

Dieses Element berechnet die *Standardabweichung* und den *Erwartungswert* einer einstellbaren Anzahl von Signalsamples. Die Ergebnisse werden an den Output-Ports *Dev* und *Mean* ausgegeben.

7.6.2.17 Das *Threshold* - Element

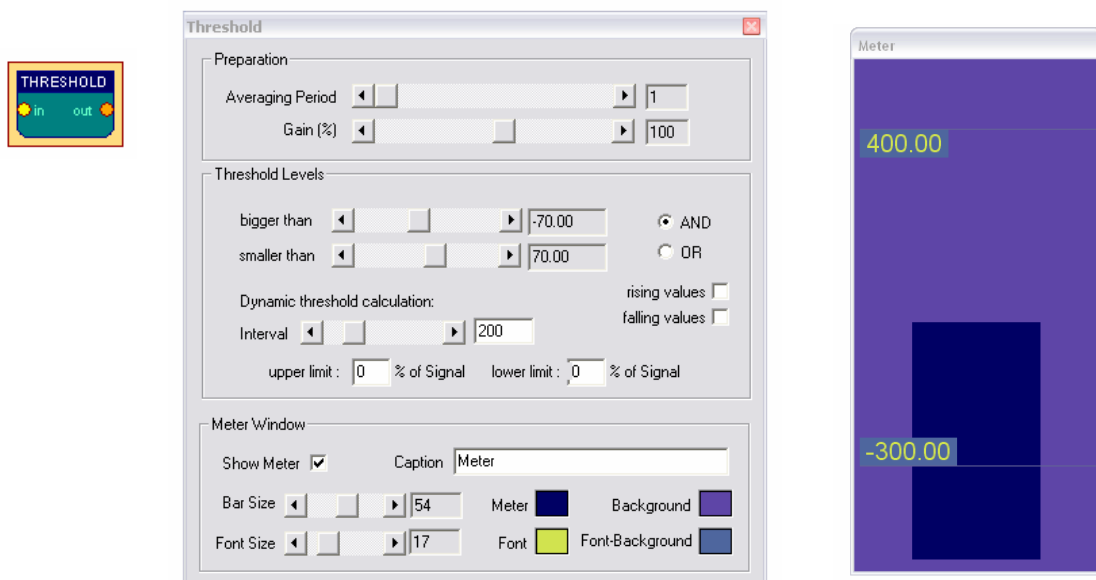


Abb. 7.35: Parameterdialog und Ausgabefenster des Threshold-Elementes

Das *Threshold*-Element ist nützlich für die Rückmeldung der Erreichung bestimmter Aktivitätsniveaus.

Das *Threshold* - Element kombiniert die folgenden Funktionalitäten:

- Mittelung und Verstärkung oder Dämpfung des eingehenden Signals (durch die Verwendung der Schieberegler in der Sektion *Preparation*)
- Einstellung fixer oberer und unterer Grenzen für das Signal (*Thresholds*): Wenn die Eingabewerte im angegebenen Bereich liegen, wird der Wert des Signals ausgegeben, andernfalls die Konstante *INVALID_VALUE*
- Erzeugung von dynamischen oberen unteren Grenzen, durch die Auswertung eines Perzentils der vorangegangenen Eingabewerte
- Einschränkung der Ausgabe auf steigende oder fallende Werte
- Anzeige eines Fensters mit Balkengrafik für den aktuellen Signalwert und die eingestellten Grenzen

Minimum und Maximum für die Einstellung der Grenzen werden vom Input-Port des Elementes übernommen. Die dynamische Berechnung der Grenzen wird durchgeführt, wenn ein Wert ungleich null in die Felder *Upper Limit* oder *Lower Limit* eingetragen wird: die neuen Grenzwerte werden so berechnet, dass die letzten *n* Messwerte (einstellbar durch den Regler *Interval*) innerhalb bzw. außerhalb der Grenze liegen. Wird *Rising-* oder *Falling Values* selektiert, können nur steigende oder fallende Werte das *Threshold*-Element passieren. In der Sektion *Meter Window* können Schriftart und -größe der Beschriftung, Farben und Titel des Anzeigefensters bestimmt werden.

7.6.3 Signalziele

7.6.3.1 Das *AVI-Player* - Element

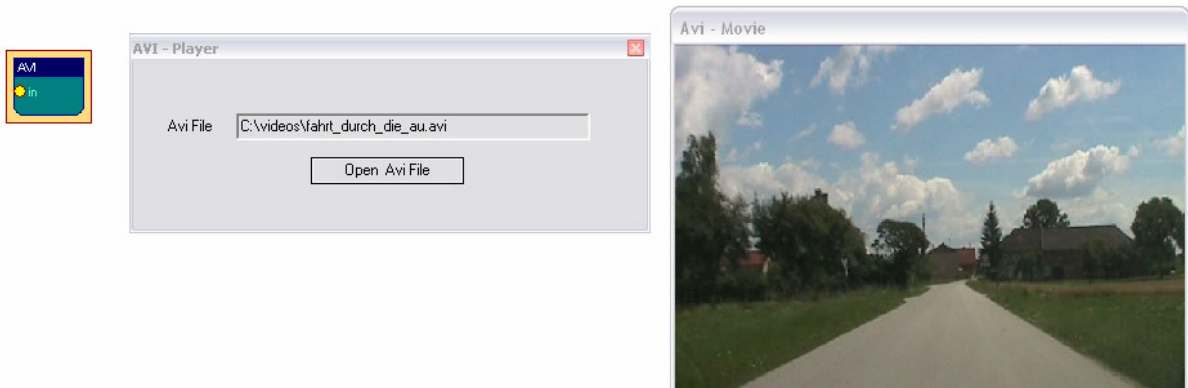


Abb. 7.36: Parameterdialog und Ausgabefenster des AVI-Player-Elementes

Das *AVI-Player* Element dient zur Darstellungen einzelner Bilder aus Filmen im *AVI*-Format. Die *AVI*-Datei kann durch Betätigung von *Open Avi File* gewählt und geöffnet werden. Der Wert am Input-Port des Elementes wird als Nummer des Bildes (*Frame-Number*) interpretiert. Wird durch andere Elemente eine sequentielle Reihenfolge von Werten erzeugt, so lässt sich der Film in entsprechender Geschwindigkeit abspielen, auch ein Rückwärts-Abspielen des Films ist möglich. Die Einzelbilder werden in einem eigenen Fenster dargestellt. Eine eventuell vorhandene Tonspur der *AVI*-Datei wird nicht behandelt. (Diese Funktionalität stellt das *Media Player* - Element zur Verfügung). Ein passender *Video-Codec* muss zur Dekomprimierung der *AVI*-Datei am System installiert sein.

7.6.3.2 Das *Com-Writer* - Element



Abb. 7.37: Parameterdialog des Com-Writer-Elementes

Das *COM-Writer* Element kann verwendet werden, um Kommando- oder Datenwerte an einen angeschlossenen *Monolith-EEG* Biosignalverstärker zu senden, falls dieser im bidirektionalen Protokoll P21 betrieben wird. Dadurch können Einstellungen wie die Baudrate oder die Samplerate sowie digitale Ausgabeports des *MonolithEEG* während der Laufzeit geändert werden. Ein Kommando- oder Datenpaket besteht aus drei Bytes, die im Benutzerdialog des Elements festgelegt werden können. Das Paket wird durch ein Betätigen der Schaltfläche *Send Command* übermittelt. Ist die Option *enable triggered sending* aktiviert, wird das Paket automatisch gesendet, sobald am Input-Port ein Wert ungleich *INVALID_VALUE* eintrifft.

7.6.3.3 Das *EDF-FileWriter* - Element

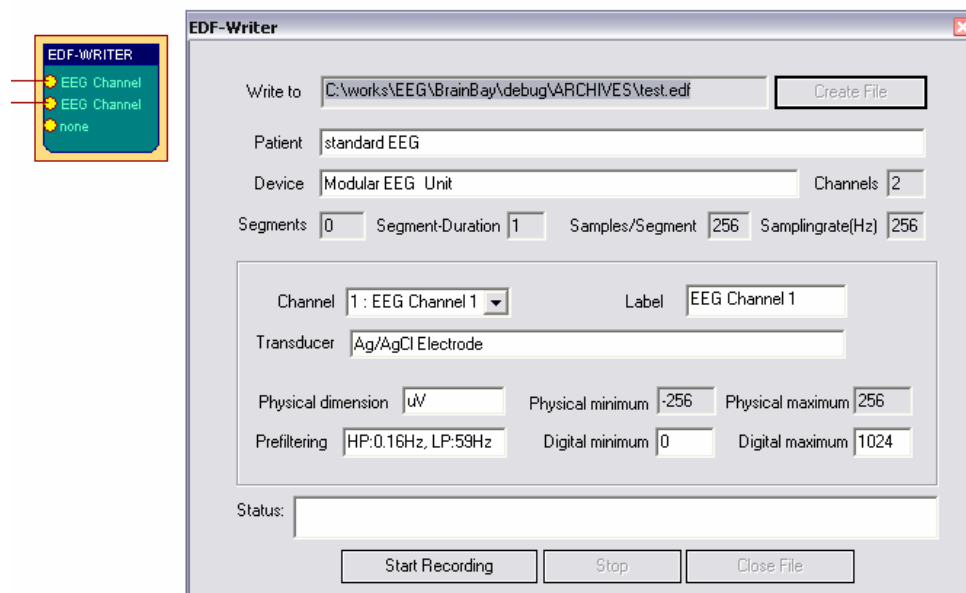


Abb. 7.38: Parameterdialog des EDF-FileWriter-Elementes

Das *EDF-FileWriter* - Element dient zur Erzeugung einer Archivdatei im Format *European Data Format (EDF)*. Eine EDF-Datei speichert an ihrem Beginn eine Reihe von Informationen über die Biosignalaufzeichnung [EDF92]. Das Datum, die Samplingrate, die Signalkanäle sowie die Wertebereiche der Signale werden automatisch vom aktuellen Setting übernommen.

Personeninformation, verwendete Hardware und Kanalbeschreibungen können durch den Benutzerdialog des *EDF-FileWriter*-Elements festgelegt werden. Bei Auswahl eines bestimmten Kanals in der Auswahlbox *Channel* werden die entsprechenden Datenfelder dargestellt.

Die Werte *Physical Minimum* und *Physical Maximum* (der Wertebereich eines Signals) werden vom jeweiligen Input-Port übernommen. Sie können durch einen Rechts-Klick auf den Input-Port geändert werden. Durch Betätigung von *Create File* kann der Speicherort und Dateiname für die EDF-Datei angegeben werden und die Datei wird erzeugt. Dies sollte erst durchgeführt werden, wenn alle zu speichernden Kanäle an die Input-Ports des Elements angeschlossen wurden. Die Speicherung von Werten in die Datei wird durch die Schaltflächen *Start Recording* und *Stop Recording* kontrolliert. Zum Abschluss einer Aufzeichnung wird die Datei durch Betätigung von *Close File* geschlossen.

7.6.3.4 Das *File-Writer* - Element

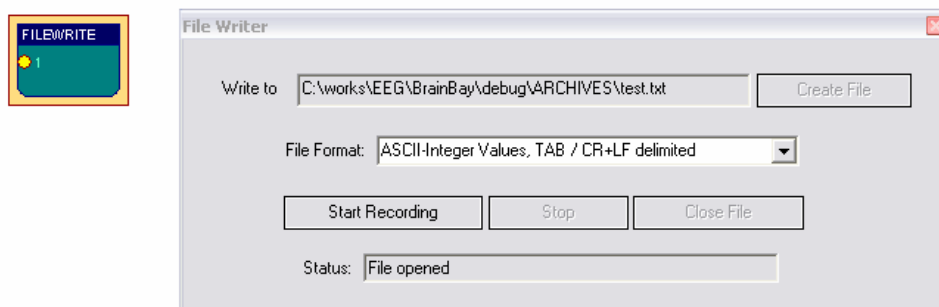


Abb. 7.39: Parameterdialog des *FileWriter*-Elementes

Das *File-Writer* - Element kann zur Erzeugung einer (headerlosen) Datei mit Werten eines oder mehrerer Eingabesignale verwendet werden. Diese Datei dient zum Exportieren von Werten in andere Applikationen wie *Microsoft Excel*, *Labview* oder *Matlab*. Durch die Auswahlbox *File Format* wird die Art der Speicherung beeinflusst. Eine Auswahl von *ASCII-Integer Values* erzeugt eine lesbare Textdatei mit ganzzahligen Werten, *ASCII-Float Values* erstellt eine Textdatei mit Werten, die zwei Nachkommastellen haben. Falls mehrere Signale angeschlossen sind, werden die Signalwerte parallel in Spalten geschrieben, die durch das eingestellte Trennzeichen (*Komma* oder *TAB*) separiert werden. Als Trennzeichen für Zeilen wird die Zeichenkombination *CR/LF* verwendet. Wird das Format *Raw/Wav* ausgewählt, erfolgt die Erstellung einer Binär-Datei mit einem Kanal. Diese kann durch einen Soundeditor für *WAV* - Dateien importiert und bearbeitet werden.

7.6.3.5 Das *Oscilloscope* - Element

Das *Oscilloscope* - Element ist wichtig für die Anzeige aktueller Signalwerte. Die angeschlossenen Signale werden in einem separaten, skalierbaren Fenster dargestellt. Ein Datengitter, das den Wertebereich der Signale und die aktuelle Sekunde der Session-Zeit zeigt, kann durch Auswahl der entsprechenden Optionen eingeblendet werden. Durch die Farbwahlfelder können Hintergrundfarbe, Signalfarbe und Schriftfarbe gewählt werden.

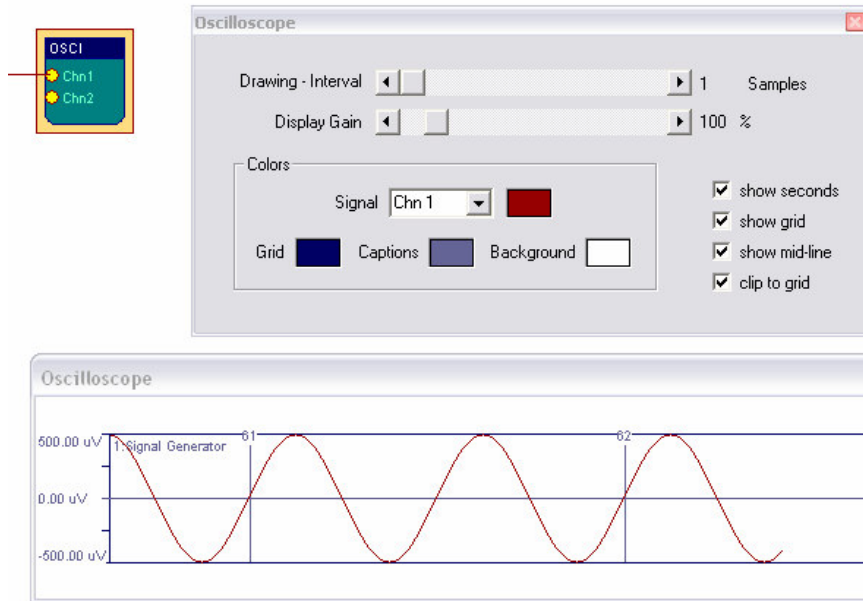


Abb. 7.40: Parameterdialog und Ausgabefenster des Oscilloscope-Elementes

Der Schieberegler *Display-Gain* dient zur Verstärkung oder Dämpfung aller angeschlossenen Signale. Individuelle Vergrößerungen können durch Änderung des Wertebereiches am Input-Port des entsprechenden Signals durchgeführt werden. Durch den Schieberegler *Drawing-Interval* wird die Zeichenperiode des Oszilloskops festgelegt: wird 1 gewählt, erfolgt die Darstellung jedes eintreffenden Wertes, 4 zeigt nur jeden vierten Wert usw. Durch die Auswahl der Option *group* werden alle angeschlossenen Signale überlagert und in einem Zeichenfeld angezeigt, was einen direkten Vergleich der Werte erleichtert.

7.6.3.6 Das *Particle-Animation* - Element

Das *Particle Animation* - Element implementiert ein grafisches Partikelsystem, das sich zur Visualisierung von Signalparametern eignet. Die Einstellungen der Partikel-erzeugung können durch bis zu sechs angeschlossene Signale beeinflusst werden. Dadurch können Effekte wie Fontänen, Flammen oder pulsierende Kreise erzielt werden, die manche Eigenschaften wie ihre Farbe oder Intensität durch die Input-Werte ändern.



Abb. 7.41: Parameterdialog und Ausgabefenster des Particle-Animation-Elementes

Folgende Eigenschaften des Partikel-Systems können durch die Auswahlbox *Parameter* gewählt werden:

- Anzahl von Partikeln im System (*Number of Particles*)
- Erzeugungsgeschwindigkeit (*Generation Interval*)
- Darstellungsverzögerung (*Slowdown*)
- Farbe (*Color*)
- X-, Y-, Z- Position
- X-, Y-, Z- Geschwindigkeit (*Speed*)
- X-, Y-, Z- Gravitation (*Gravity*)
- Lebensdauer eines Partikel (*Life Span*)
- Zufallsanteil an der Geschwindigkeit (*Randomizer*)

Durch *Import Colors* wird eine Farbpalette geladen, der Parameter für *Farbe* gibt einen Index in dieser Palette an. Die obigen Eigenschaften des Partikelsystems werden einzeln ausgewählt und durch einen Schieberegler eingestellt, um ein gewünschtes Gesamtbild zu erzeugen. Durch die Auswahlbox *Remote Control By* lässt sich ein Parameter automatisch vom Wert eines Input-Ports steuern.

Durch Betätigung der Schaltflächen *Get Minimum* und *Get Maximum* wird der aktuelle Wert des Parameters als neuer Minimalwert oder Maximalwert definiert. Wird der Parameter durch einen Input-Port gesteuert, so erfolgt eine Konvertierung vom Wertebereich des angeschlossenen Signals auf den eingestellten Bereich zwischen Minimum und Maximum. Durch *Reset Min and Max* können die Grenzen wieder auf die ursprünglichen Werte zurückgesetzt werden. Da die grafische Realisierung des Partikelsystems (vor allem bei vielen Partikeln) aufwändig ist, wird die Verwendung einer Grafikkarte mit *OpenGL*-Beschleunigung empfohlen.

7.6.3.7 Das *Midi-Output* - Element

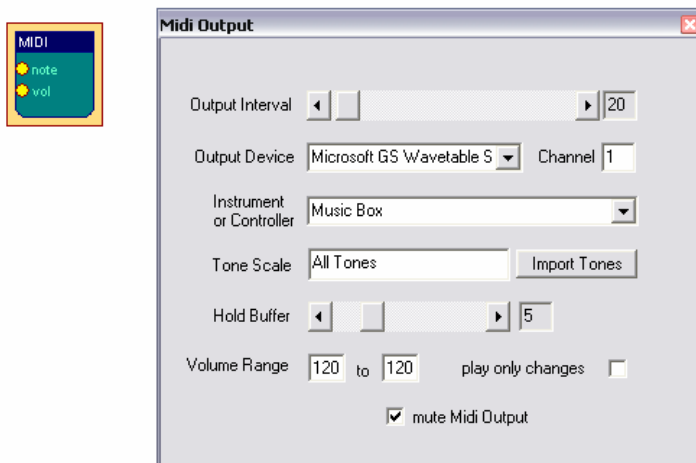


Abb. 7.42: Parameterdialog des *Midi-Output*-Elementes

Das *Midi-Output* - Element stellt Midi-Audio Rückmeldungen für verbundene Signale zur Verfügung. Tonhöhe und Lautstärke werden durch eingehende Signalwerte auf den Input-Ports 1 und 2 beeinflusst. Der Wertebereich der angeschlossenen Signale wird auf die eingestellten Grenzen für die Lautstärke (*Volume Range*) bzw. auf die gewählte Tonleiter (*Tone Scale*) transformiert.

Durch die Auswahlbox *Output Device* wird eines der aktivierten Midi-Ausgabe Geräte geöffnet. (Die Aktivierung von Midi-Geräten erfolgt im Benutzerdialog *Options*). Die Angabe eines Midi-Kanals (*Channel*) von 1 bis 16 erlaubt den parallelen Betrieb mehrerer Midi-Elemente mit verschiedenen Instrumenten auf einem Midi-Ausgabegerät. Die Auswahlbox *Instrument or Controller* selektiert das zu verwendende Instrument oder einen *Midi-Controller*. Midi-Controller sind vor allem bei einem Betrieb externer Midi-Geräte wie Synthesizer oder Drum-Computer sinnvoll. Der Schieberegler *Output Interval* stellt die Periode der Tonerzeugung ein: bei Auswahl eines Intervalls von 1 werden die Töne mit der Samplefrequenz des Systems generiert (das ist normalerweise zu schnell). Eine Einstellung von 128 ergibt bei einer Samplerate von 256 Hz zwei Töne pro Sekunde.

Durch die Betätigung von *Import Tones* kann eine gespeicherte Tonleiter geladen und verwendet werden. Der Schieberegler *Hold Buffer* wählt die Anzahl von Tönen, die gleichzeitig gespielt werden können (ist hier 1 eingestellt, wird jede Midi-Note vor dem Spielen der nächsten Note ausgeschaltet). Ist *play only changes* aktiviert, werden zwei aufeinander folgende Notenwerte nur dann gespielt, wenn sie unterschiedlich sind.

7.6.3.8 Das *Mouse-Controller* - Element

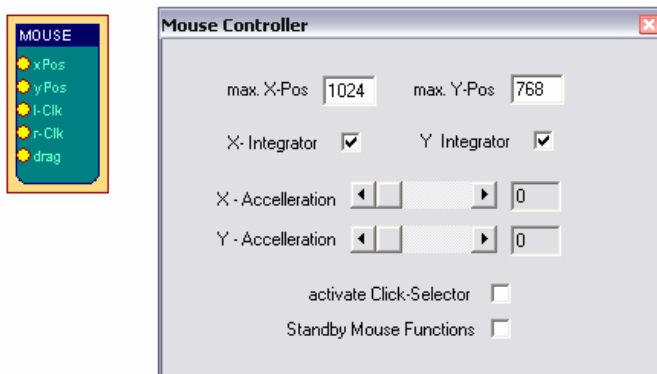


Abb. 7.43: Parameterdialog des *Mouse-Controller*-Elementes

Das *Mouse-Controller* - Element bietet Möglichkeiten zur Steuerung des Maus-Cursors am Bildschirm und dient zum Auslösen von Maus-Klicks. Die aktuelle Auflösung des Bildschirms wird in den Feldern *max. X-Pos* und *max. Y-Pos* angegeben. Während der Laufzeit werden die Werte an den Input-Ports *xPos* und *yPos* zur Bestimmung der Cursor-Position verwendet. Falls die Optionen *x-Integrator* und *y-Integrator* gewählt sind, werden die Input-Werte als relative Änderungen der Position interpretiert (also ein Wert von 2 führt pro Bearbeitungsschritt zum Erhöhen der Cursorposition um 2 Pixel). Andernfalls werden die Werte als absolute Positionen interpretiert.

Ein *Linksklick* wird vom Element ausgelöst, falls ein Wert ungleich *INVALID_VALUE* am Input-Port *l-Clk* ankommt. *Rechtsklicks* und das *Aufnehmen und Ablegen* (*Drag-and-Drop*) werden auf ähnliche Weise durch die Ports *r-Clk* und *drag* ausgelöst. Die Option *activate Click-Selector* dient zur Darstellung eines eigenen Fensters für die Operationen *Rechtsklick*, *Doppelklick* und *Drag-and-Drop*. Mit Hilfe des *Click-Selector*-Fensters können diese Funktionen durch einen normalen *Linksklick* ausgelöst werden (der nächste Klick erzeugt dann die ausgewählte Funktion). Eine Einstel-

lung für *Dwell-Time* ungleich Null führt zu einer automatischen Erzeugung von *Linksklicks*, falls der Maus-Cursor für die eingestellte Verweildauer in einem nahen Umkreis bleibt. So können auch Klicks ausgelöst werden, ohne ein separates Kontrollsignal für den *I-Clk* - Port zu benötigen. Eine sinnvolle Quelle für eine Maussteuerung ist das *Camera*-Element, das durch die Möglichkeiten zur Gesichtslokalisation die Ausgangsdaten für eine Kopfsteuerung der Maus zur Verfügung stellt (siehe Beispielkonfiguration *Kamera-Maus* 8.5).

7.6.3.9 Das *Sound Player* - Element

Der *Sound-Player* kann zum gesteuerten Abspielen von Audio-Dateien verwendet werden. Unterstützte Formate sind *wav*, *mp3*, *voc* und *aiff*. Das Element eignet sich zur Wiedergabe von kurzen Soundeffekten oder Tönen (für längere Musikdateien ist das Element *Media Player* zu verwenden). Die Werte der Input-Ports kontrollieren das Starten der Wiedergabe (*on*), die Lautstärke (*vol*) und die Wiedergabegeschwindigkeit / Tonhöhe (*speed*). Die Wiedergabe erfolgt sobald ein Wert ungleich *INVALID_VALUE* am Port *on* empfangen wird.

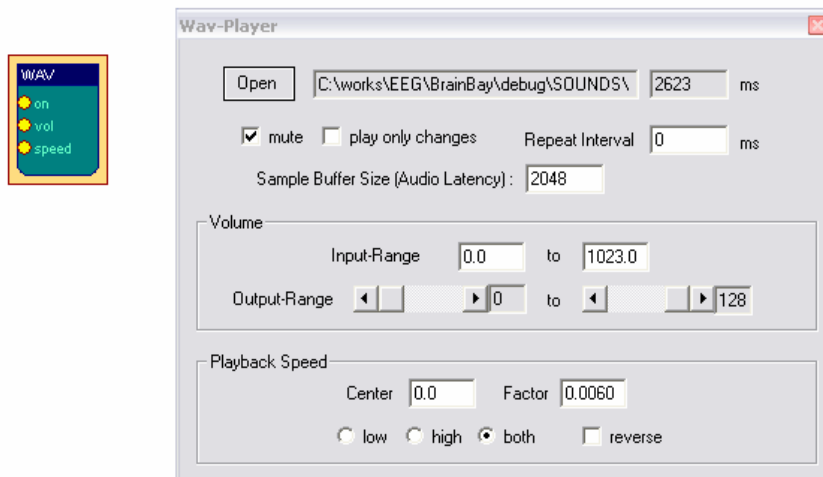


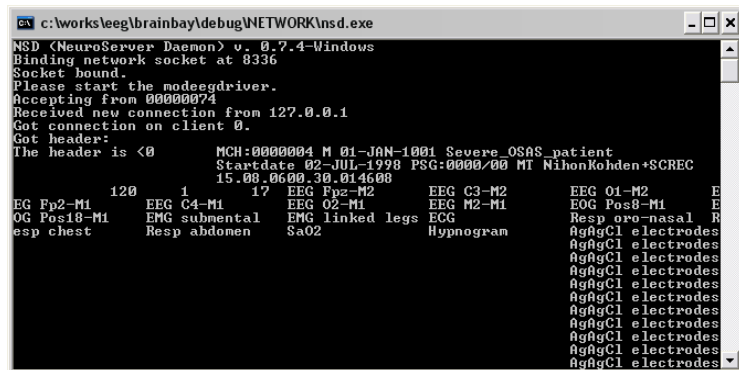
Abb. 7.44: Parameterdialog des *Sound-Player*-Elementes

Falls die Option *play only changes* gewählt ist, erfolgt die Wiedergabe nur einmal pro Wechsel von *INVALID_VALUE* auf einen anderen Wert. Die Einstellung *Repeat Interval* wählt die minimale Dauer der Pause zwischen zwei Wiedergaben. Der Wert *Sample Buffer Size* gibt die Größe des internen Wiedergabepuffers vor. Diese Einstellung beeinflusst die Latenzzeit bei der Wiedergabe und die benötigte Rechenzeit. (Werte um 2000 ergeben eine akzeptable Latenz auf einem 1,6-GHz-Rechner). Die Wertebereiche für die Lautstärke und Wiedergabegeschwindigkeit werden von den angeschlossenen Signalen übernommen und können durch den Benutzerdialog manuell verändert werden. Die Option *reverse* invertiert den Zusammenhang des Input-Ports *speed* und der Abspielgeschwindigkeit. Der Wert für *Center* gibt den Signalwert für die originale Wiedergabegeschwindigkeit an und der Wert *Factor* bestimmt das Verhältnis von Geschwindigkeitsänderung zu Signaländerungen.

7.6.3.10 Das TCP-Sender - Element

Das *TCP-Sender* Element dient zur Übertragung von Signalkanälen über Netzwerk (LAN oder Internet). Die Übertragung erfolgt über eine TCP-Socketverbindung und das *Neuroserver*-Software Framework von Rudi Cilibrasi [CIL03].

Um eine Verbindung aufzubauen, wird die IP-Adresse eines Host-Computers benötigt, auf dem ein *Neuroserver*-Prozess läuft. Dies kann auch der lokale Rechner (*localhost*) sein. Zum (lokalen) Starten des Service wird im Menü *Network* die Option *Start Neuroserver* gewählt.



```
NSD (NeuroServer Daemon) v. 0.7.4-Windows
Binding network socket at 8336
Socket bound.
Please start the modeegdriver.
Accepting from 00000074
Received new connection from 127.0.0.1
Got connection on client 0.
Got header:
The header is <0          MCH:0000004 M 01-JAN-1001 Severe_OSAS_patient
                          Startdate 02-JUL-1990 PSG:0000/00 MI Nihonkohden+SCREC
                          15.00.0600.30.014600
EG Fp2-M1      120      1      1?      EEG Fpz-M2      EEG C3-M2      EEG O1-M2      E
OG Pos18-M1    EEG C4-M1      EEG O2-M1      EEG M2-M1      EOG Pos8-M1      E
esp chest      Resp abdominal  ENG linked legs  ECG              Resp oro-nasal R
                          SaO2              Hypnogram
AgAgCl electrodes
AgAgCl electrodes
AgAgCl electrodes
AgAgCl electrodes
AgAgCl electrodes
AgAgCl electrodes
AgAgCl electrodes
AgAgCl electrodes
AgAgCl electrodes
```

Abb. 7.45: Lokal laufender Neuroserver mit aufgebauter Sendeverbindung

Verbindung des TCP-Sender - Elements mit dem Neuroserver - Service

Beim Aufbau der Verbindung werden Informationen zur Aufnahme durch einen EDF-Header übertragen. Personendaten, verwendete Hardware sowie die Signalkanäle können durch den Benutzerdialog beschrieben werden. Dazu sollten alle zu übertragenden Signale bereits an das Element angeschlossen sein. Um eine Verbindung zum Neuroserver herzustellen wird die IP-Adresse oder 'localhost' in das Textfeld *Neuroserver* eingegeben und *Connect* betätigt. Im Falle einer erfolgreichen Verbindung erscheint in der Statusbox 'Entering EEG mode' und 'OK'.

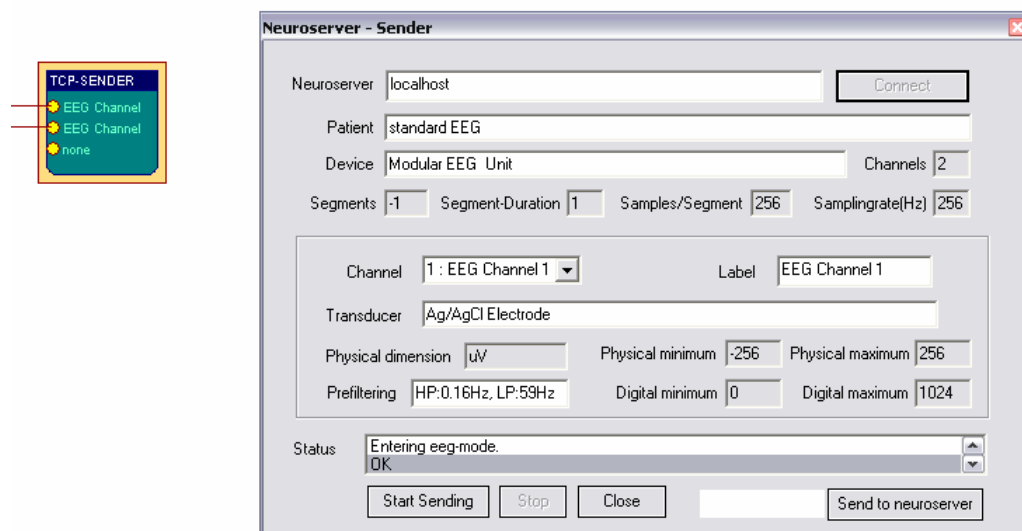


Abb. 7.46: Parameterdialog des TCP-Sender -Elementes

Start Sending und *Stop* werden zur Kontrolle des Datenflusses zum Neuroserver verwendet. Die Schaltfläche *Send to neuroserver* erlaubt das direkte Absetzen eines Neuroserver-Kommandos [CIL03]. Durch Betätigung von *Close* wird die Verbindung zum Neuroserver geschlossen.

5.3.11 Das *Media-Player* - Element

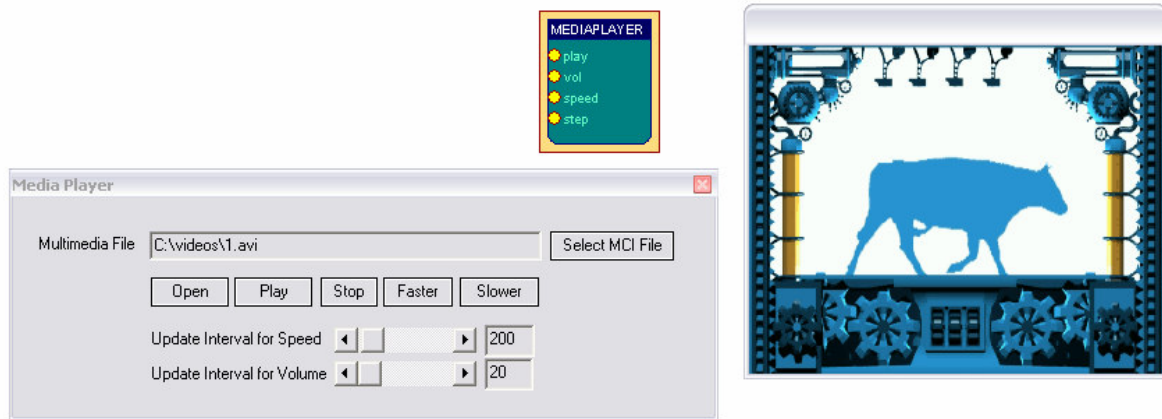


Abb. 7.47: Parameterdialog und Ausgabefenster des Media-Player-Elementes

Dieses Element bietet einen Zugriff auf die *Windows Medienschnittstelle (Media Control Interface, MCI)*. Dadurch können verschiedene Medienformate wie *AVI, WMV, WAV, MP3* wiedergegeben werden. Nach der Auswahl einer Mediendatei durch *Select MCI File* wird die Wiedergabe durch den Erhalt eines gültigen Wertes am Input-Port *Play* gestartet. Lautstärke und Abspielgeschwindigkeit können durch die Ports *Vol* und *Speed* beeinflusst werden (wobei ein Wert von 1000 volle Lautstärke bzw. normale Geschwindigkeit bedeutet). Die Intervalle zur Aktualisierung dieser Parameter können durch Schieberegler eingestellt werden. Durch den Erhalt gültiger Werte am Input-Port *Step* erfolgt ein Einzelschritt in der Mediendatei.

7.6.4 Zusätzliche Elemente

7.6.4.1 Das *Documentation* - Element

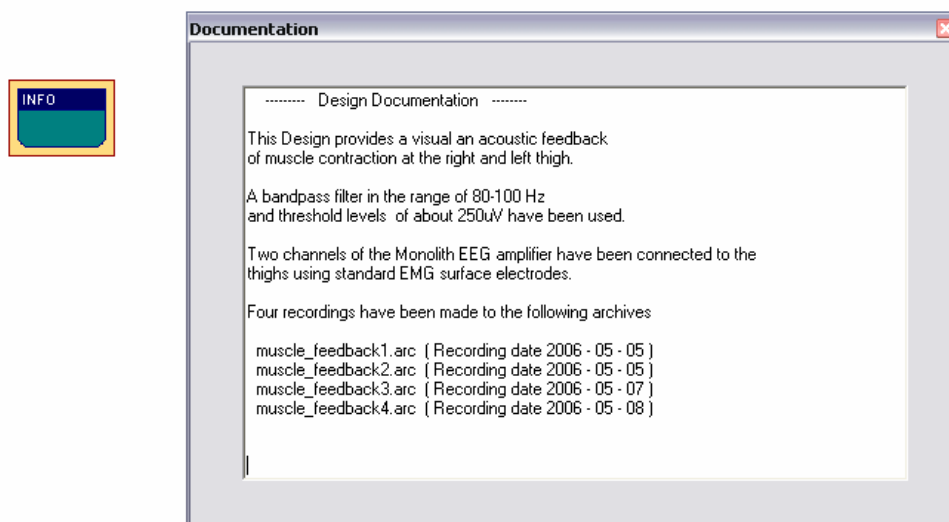


Abb. 7.48: Parameterdialog des Documentation-Elementes

Dieses Element stellt eine einfache Textbox für Dokumentationszwecke zur Verfügung. Es dient zur Beschreibung der Session, erzeugter Archive, der Einstellung von Elementen usw.

7.6.4.2 Das *Ballgame* - Element

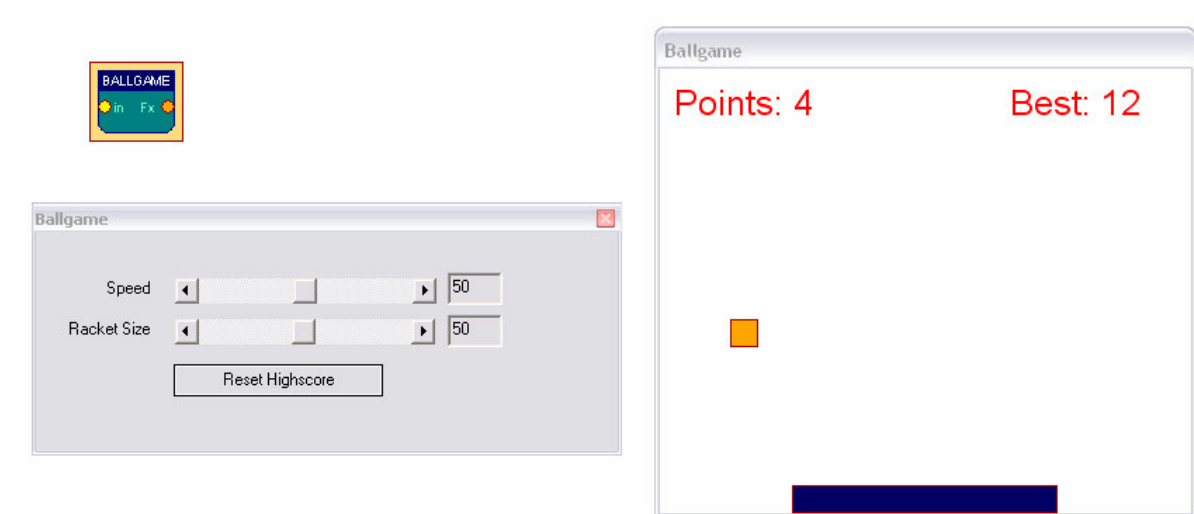


Abb. 7.49: Parameterdialog und Ausgabefenster des Ballgame-Elementes

Das *Ballgame* - Element ist ein Beispiel für ein einfaches Feedback-Spiel, das dem Klassiker '*Pong*' nachempfunden ist: durch einen Balken (*Racket*) soll ein Ball im Spiel gehalten werden, dafür erhält man Punkte. Die Position des Balkens wird dabei durch die Werte des Signals am Input-Port gesteuert. Die Parameter *Racket Size* und *Speed* regeln die Größe des Balkens und die Geschwindigkeit des Balles. Die höchste Anzahl von Treffern wird angezeigt und vom Element gespeichert.

8. Anwendungsbeispiele

Die folgenden Beispiele zeigen praktische Anwendungen von *BrainBay* durch verschiedene Design-Konfigurationen.

8.1 Eine Konfiguration für Alpha / Beta - Neurofeedback

Die folgende Design-Konfiguration dient zur Echtzeitverarbeitung einer monopolaren 2-Kanal-EEG-Ableitung. Drei Silber/Silberchloridelektroden wurden dafür an den Positionen C3, C4 und Cz nach dem 10/20 - System fixiert. Der Referenzpunkt war Cz. Ein abrasives Kontaktgel wurde zur Verbesserung des Elektrode/Haut-Überganges verwendet. Die Impedanz der Verbindung lag bei ca. 10kOhm. Verstärkung und Digitalisierung der Werte erfolgten mit einem MonolithEEG EEG-Verstärker bei 256 Hz Samplerate, unter Verwendung des P2- Firmware - Protokolls zur Datenübertragung an einen PC über die USB-Schnittstelle.

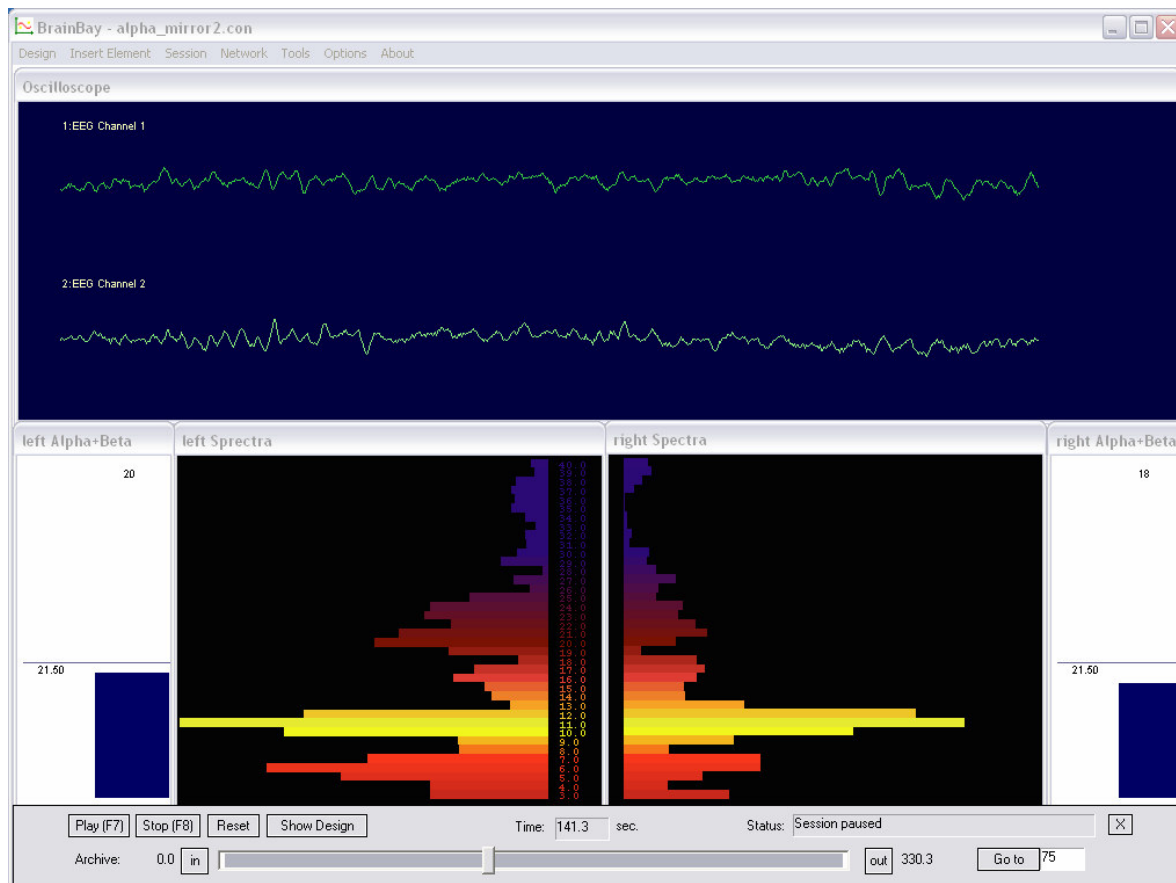


Abb. 8.1: Ausgabefenster einer Konfiguration für das Neurofeedback von EEG-Frequenzbändern

Das Oszilloskop-Fenster zeigt die gemessenen Werte nach der Anwendung eines 50Hz-Filters. Pro Kanal wird eine FFT - Spektralanalyse durchgeführt und der Frequenzgehalt wird durch Balkengrafiken dargestellt (*Brain-Mirror Display*). Die aktuellen Werte die *Alpha*- und *Beta*-Magnituden werden durch Filter berechnet und in separaten Fenstern als vertikale Balken gezeigt. In diesen Fenstern sind Grenzwerte sichtbar, deren Erreichen zur Auslösung eines Audio-Signals führt.

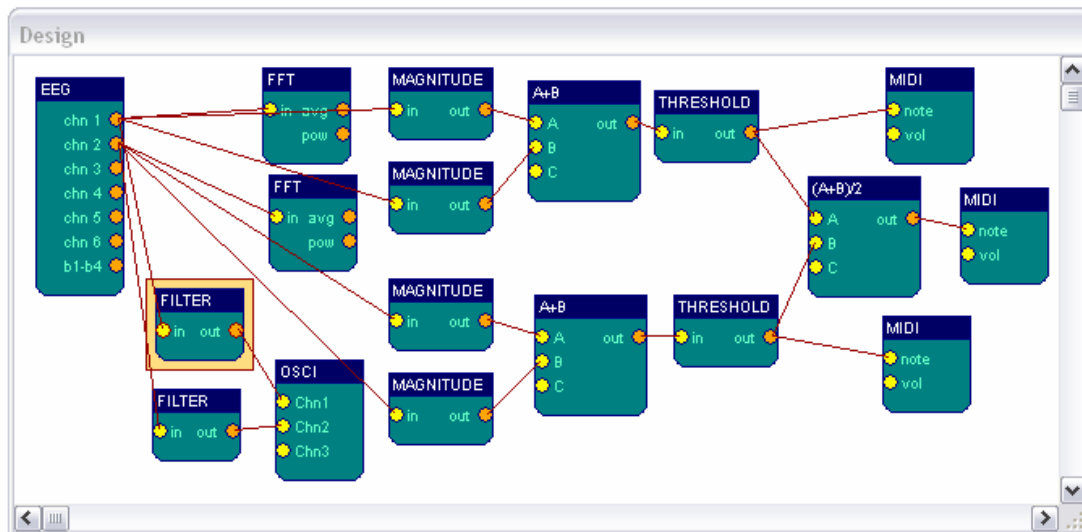


Abb. 8.2: Elemente der Konfiguration für das Neurofeedback von EEG-Frequenzbändern

Die Konfiguration besteht pro Kanal aus den folgenden Elementen:

- ein **Filter** vom Typ 'Bandstop Resonator', Bandstop-frequency 50 Hz, Order 8
- eine Verbindung zum **Oscilloscope**, um den gefilterten Kanal darzustellen
- ein **FFT** - Element, Berechnungsperiode 10 Samples, Verstärkung 250, 128 Bins, Frequenzanzeige 3 - 40 Hz, Typ: Balkendiagramm, importierte Farbpalette
- zwei **Magnitude** - Elemente, Butterworth - Filtercharakteristik, Frequenzbereiche 8-12 Hz und 16-20 Hz, Filter Ordnung 4, Verstärkung 100 %
- ein Expression **Evaluator** - Element, zur Addition der berechneten Magnituden
- **Threshold** - Element, Mittelung von 200 Samples, untere Grenze auf 21,5 μ V
- **Midi** Tongenerator, Verwendung einer benutzerdefinierten Tonleiter und eines Ausgabeintervalls von 70 Samples.

Filterantwort und 3D-Darstellung der Frequenzbänder:

Die Frequenzselektivität des 50Hz-Filter ist im Benutzerdialog zu sehen. Eine Verwendung der 3D-Darstellung für die Fourier-Transformation erfolgt durch Auswahl der Option *3D-Linegrid* im Benutzerdialog des *FFT*-Elements.

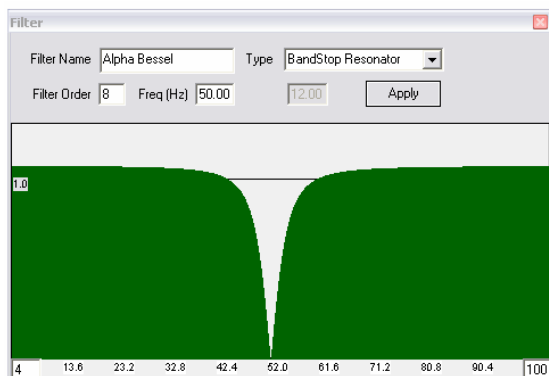


Abb. 8.3: Filterantwort des 50Hz-Bandstop-Resonators

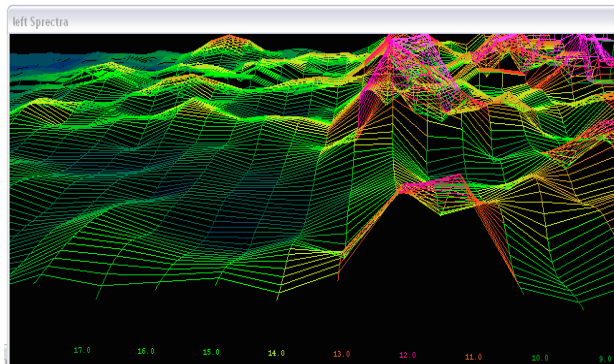


Abb. 8.4: 3D-Darstellung der Frequenzbänder

8.2 Herzschlag-Rückmeldung aus einer Biosignalaufnahme

Die folgende Designkonfiguration extrahiert den Herzschlag aus einer 17-Kanal *EDF*-Aufzeichnung verschiedener Biosignale. Elektrokardiogramm und Frequenzkomponenten der Herzmuskelkontraktion werden dargestellt. Zur Detektion des Herzschlages wird ein dynamischer Schwellwert verwendet, der aufgrund der aktuell erhobenen Signalwerte berechnet wird. Akustisches Feedback für jeden Herzschlag wird durch drei Midi-Generatoren erzeugt, die Herzfrequenz wird numerisch in einem Fenster dargestellt.

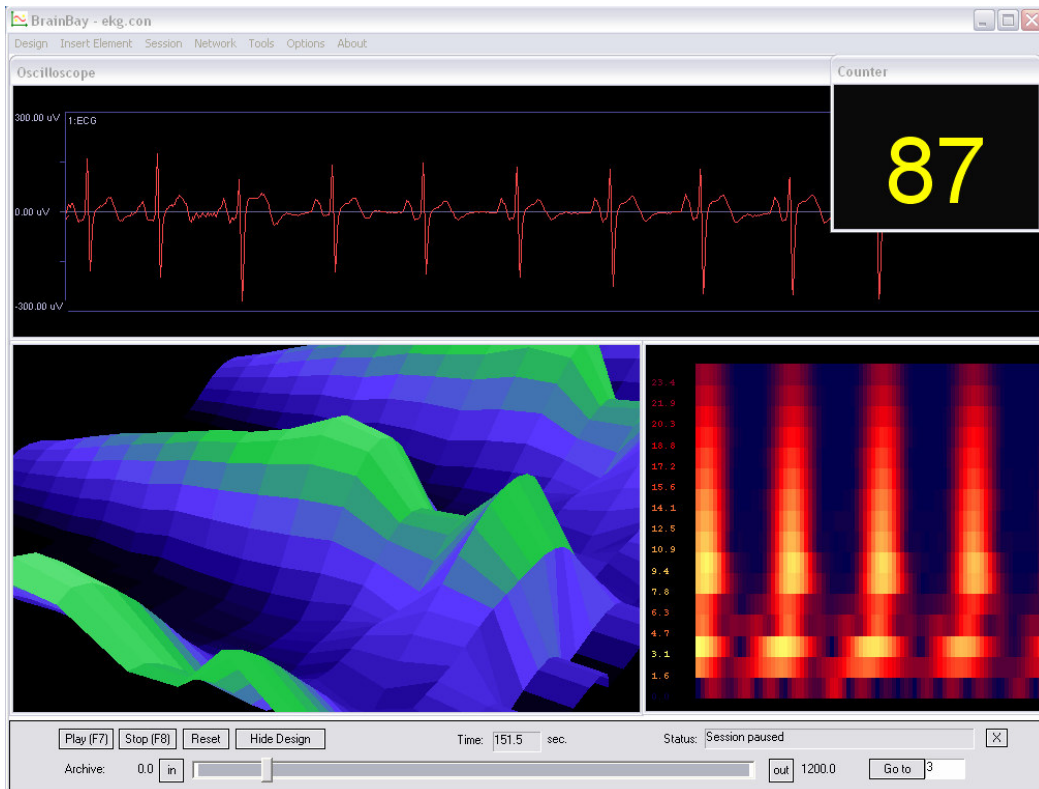


Abb. 8.5: Ausgabefenster der Konfiguration für das optoakustische EKG-Feedback

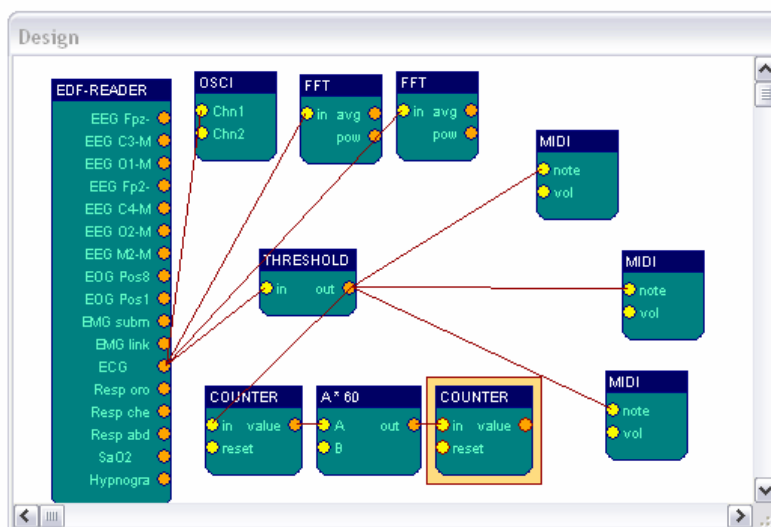


Abb. 8.6: Elemente der Konfiguration für das optoakustische EKG-Feedback

Der Output-Port *ECG* des *EDF-FileReaders* wird mit Anzeige- und Verarbeitungselementen verbunden. Das erste *FFT*-Element läuft im 3D-Modus, das zweite im Spektrogramm-Modus. Beide *FFT*s verwenden 32 Bins und ein Berechnungsintervall von 10 Samples. Die *R*-Zacke des EKGs wird durch das *Threshold*-Element erkannt.

Übertritte des Schwellwertes für die *R-Zacke* werden von einem *Counter* - Element registriert. Dieses läuft im Modus *Count Frequency* und bestimmt so die Frequenz von Schwellwertübertretungen. Die Ausgabe des *Counter*-Elements ist in Hz gegeben, eine Ermittlung der aktuellen Pulsrate (in *BPM*) erfolgt durch Multiplikation mit 60 durch einen *Expression-Evaluator*. Der zweite *Counter* wird zur Darstellung des Wertes verwendet.

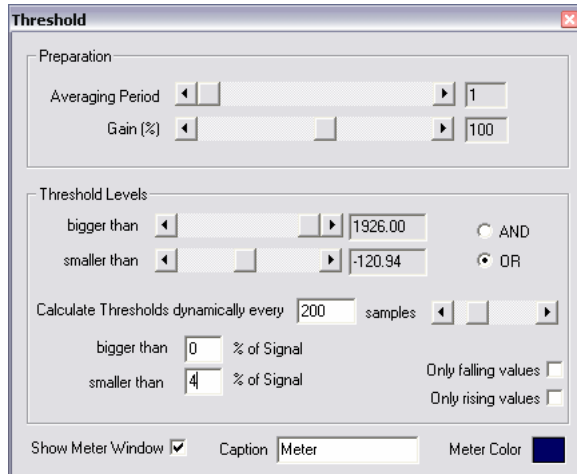


Abb. 8.7: Thresholds zur Erkennung der R-Zacke im EKG

Erkennung der R-Zacke:

Die R-Zacke ist in dieser Ableitung durch einen stark negativen Ausschlag gekennzeichnet. Die Ursache ist wahrscheinlich eine nicht standardisierte Ableiteposition. Zur Erkennung dient das *Threshold*-Element. Da die Amplitude über die Dauer der Aufzeichnung stark schwankt, kommt ein dynamischer Schwellwert zum Einsatz: Die Grenze wird auf < 4% der letzten 200 Werte eingestellt.

Für das Audio-Feedback werden drei *Midi*-Generatoren mit derselben harmonischen Tonleiter verwendet. Da die Dauer einer erkannten Spitze im EKG sehr kurz ist, werden als Tonerzeugungsintervalle kleine Werte von 1, 2 und 3 eingestellt. Die unterschiedlichen Intervalle bewirken Harmonien durch das gleichzeitige Auslösen verschiedener Töne der Tonskalen.

8.3 Eine Konfiguration für Muskel-Feedback und -rehabilitation

Die folgende Designkonfiguration kann zur Durchführung von Muskelfeedback-Training und Rückmeldung von Muskelaktivität für Rehabilitationszwecke verwendet werden. Die Ableitung der Muskelpotenziale wurde durch einen *MonolithEEG*-Verstärker unter Verwendung des bidirektionalen *P21*-Protokolls bei 600 Hz Samplefrequenz durchgeführt. Die Verstärkung des *MonolithEEG* wurde auf einen geringen Wert eingestellt, da die Muskelpotenziale von etwa 500 μV den Verstärker sonst voll angesteuert hätten. Als Oberflächenelektroden wurden Standard *BlueSensor* EMG Klebeelektroden eingesetzt und am rechten und linken Oberschenkel (*m. femoris*) angebracht. Vier Elektroden wurden für eine bipolare 2-Kanal-Ableitung eingesetzt:

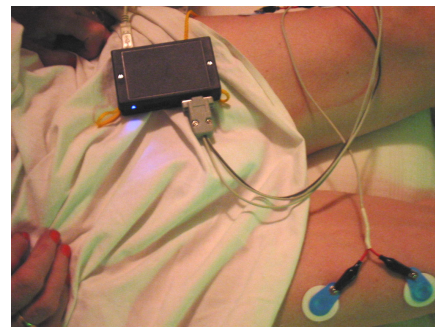


Abb. 8.8: Einsatz von EMG-Feedback zum Training der Muskelkontraktion

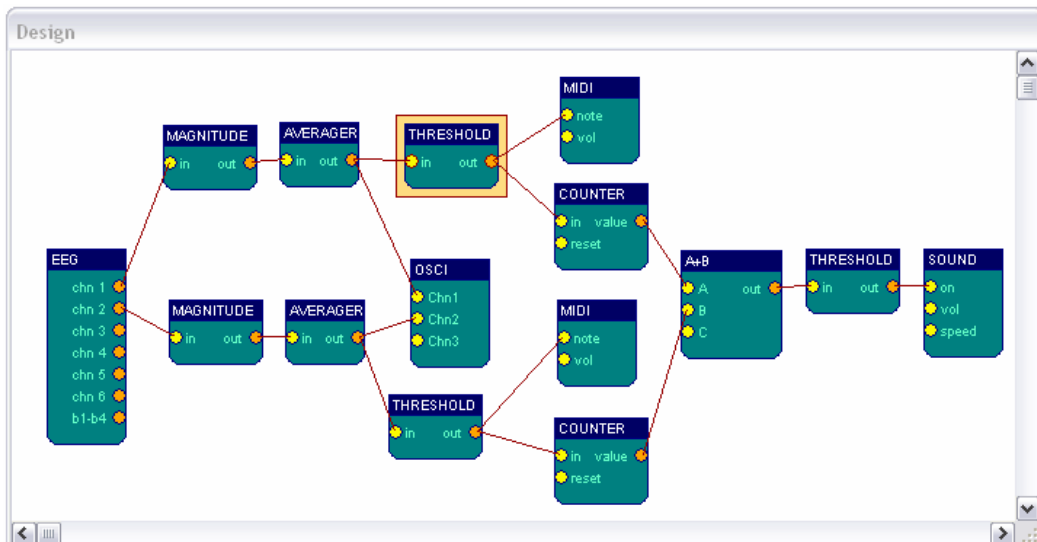


Abb. 8.9: Elemente der Konfiguration für das Muskeltraining

In der Design-Konfiguration wird ein Bandpass-Filter in Form eines *Magnitude* - Elements angewandt, um die Messwerte von Muskelpotenzialen von anderen Signalquellen zu unterscheiden. Die Parameter sind für ein aktives Frequenzband von 85-115 Hz, eine Filterordnung von 4 und eine Verstärkung von 400% eingestellt.

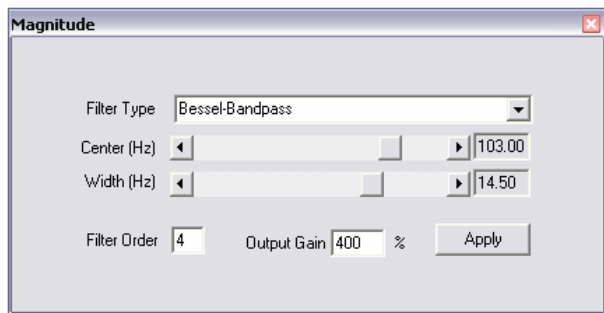


Abb. 8.10: Filtereinstellung für die Muskelamplitude

Die Ausgabe des *Magnitude* - Elements wird vom *Averager* über 300 Samples gemittelt, das entspricht einer halben Sekunde bei 600 Hz Samplefrequenz. So werden schnelle Anteile und starkes Schwingen des Wertes ausgeglichen. Dadurch kann die anschließende Schwellwert-Überprüfung durch das *Threshold* - Element zuverlässig durchgeführt werden und die Anzeige im Oszilloskop-Fenster zeigt klar das aktuelle Aktivitätsniveau. Im *Threshold*- Element wird manuell ein fixer Grenzwert eingestellt, der je nach gewünschter Kontraktionskraft gewählt und bei erfolgreichem Training erhöht wird. Bei Wahl eines niedrigen Grenzwertes führen schon geringste Muskelkontraktionen zur Auslösung des Feedback-Signals durch die angeschlossenen *Midi*-Elemente. Zusätzlich wird durch das Erreichen des Schwellwertes für den linken oder rechten Muskelkanal je ein Zähler erhöht. Erreicht die Summe dieser Zähler den Zielwert, der durch ein zweites *Threshold*-Element vorgegeben wird, erfolgt die Ausgabe eines kurzen Musikstücks durch das *Sound* - Element.

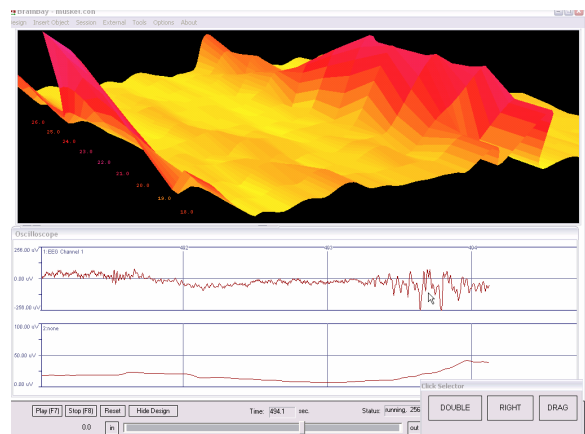


Abb. 8.11: Darstellung einer Muskelkontraktion als 3D-Fourier-Analyse und im Oszilloskop

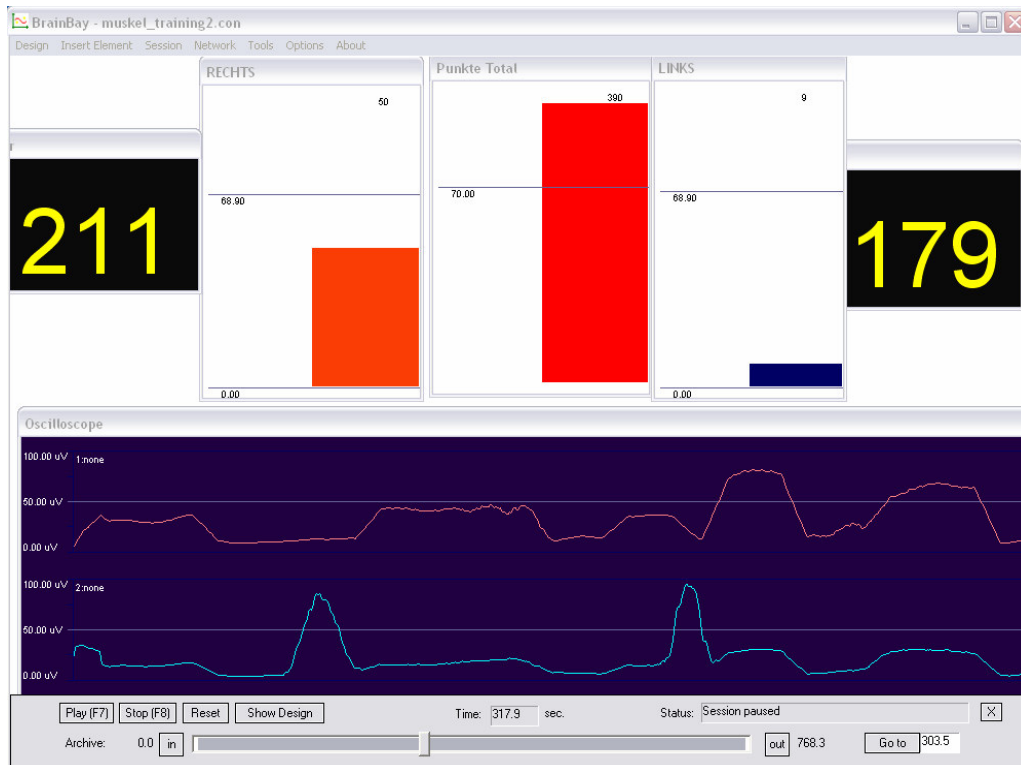


Abb. 8.12: Laufendes Muskeltraining, Zählen und Aufsummieren der Kontraktionen

Die *EMG*-Aktivität an den beiden Ableitungsorten am rechten und linken Oberschenkel wird im Oszilloskop und in den beiden Aktivitätsfenstern *RECHTS* und *LINKS* dargestellt. Dort sichtbar sind auch die eingestellten Grenzwerte. Die Zahlen geben die Anzahl der durchgeführten Kontraktionen an, der mittlere Balken zeigt den Gesamtfortschritt.

8.4 Steuerung eines Films durch Aktivität am Home-Trainer

Die folgende Anleitung illustriert die Verwendung eines externen Schalters zur Kontrolle der Geschwindigkeit einer Filmwiedergabe. Der Schalter dient zum Abfragen der Drehgeschwindigkeit des Rades eines Home-Trainers. Für diese Anwendung eignet sich am besten ein *Reed-Kontakt* (Magnetschalter), da dieser berührungsfrei durch das Passieren eines Magneten geschlossen wird.

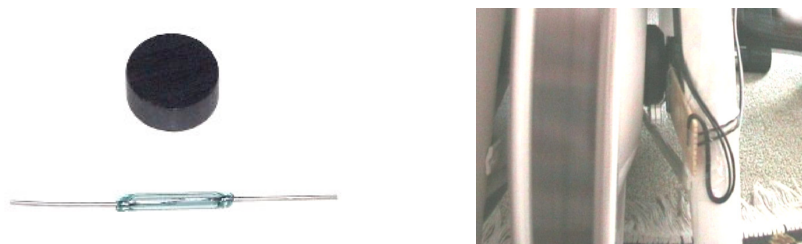


Abb. 8.13: Komponenten und Befestigung des Schaltkontaktes zur Ermittlung der Drehgeschwindigkeit

Abb. 8.13 zeigt einen Magneten und den Reed-Kontakt (links), sowie die Befestigung von Magnet und Reed-Kontakt am Hometrainer (rechts).

Der Schalter wird direkt mit einem digitalen Eingang ($PD1 - PD4$) des *ModularEEG* verbunden. Der momentane Zustand des Schalters ist somit durch den Output-Port $B1-B4$ des *EEG* - Elementes in der Designkonfiguration verfügbar.

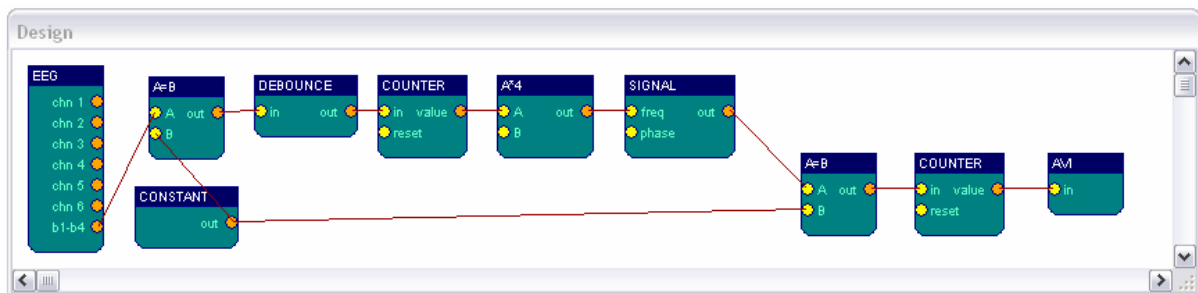


Abb. 8.14: Konfiguration zur Steuerung eines Filmes durch die Aktivität am Schaltkontakt

In der Konfiguration wird der Wert des Output-Ports $B1-B4$ mit einer Konstanten verglichen, die dem geschlossenen Reed-Kontakt entspricht (wenn $B2$ angeschlossen wird führt ein Schließen des Kontakts zum Löschen von Bit 2, also zum Wert 11). Das *Debounce* - Element verhindert Mehrfachimpulse beim Passieren des Magneten. Das erste *Counter* - Element misst die Frequenz des Schaltimpulses, also die Drehgeschwindigkeit des Rades. Dieser Wert kontrolliert die Frequenz eines Signalgenerators, es wird also eine Umwandlung der Frequenz durchgeführt, wodurch die Schaltfrequenz des Reed-Kontaktes von etwa 1-4 Hz in eine Einzelbildfrequenz für die Filmwiedergabe von etwa 5-30 konvertiert wird. Die aktuelle Filmposition, die als Eingabe für das *AVI* - Element dient, wird durch einen zweiten *Counter* berechnet.

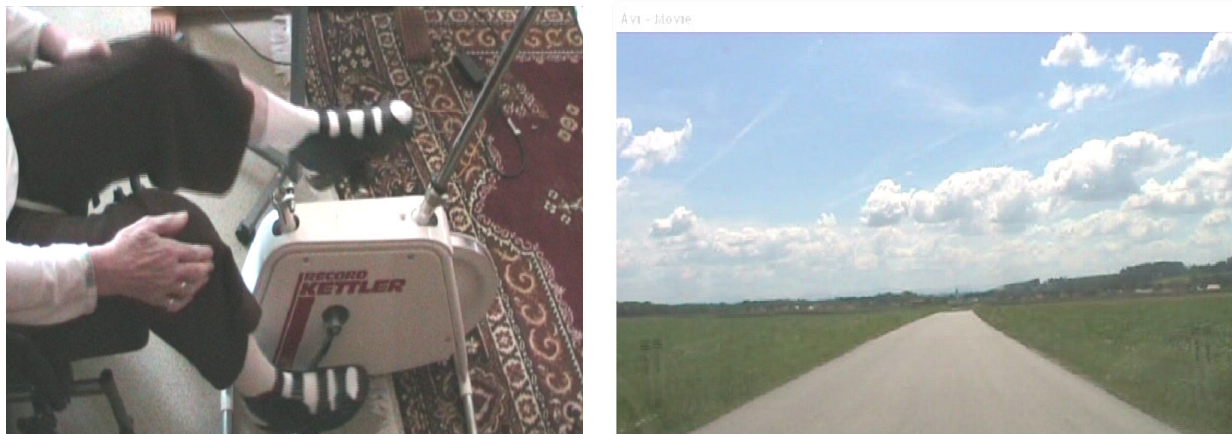


Abb. 8.15: Die Konfiguration im Einsatz - eine Überlandfahrt, ausgeführt im Wohnzimmer

8.5 Eine Kamera-Maus mit Klick-Funktionen

Dieses Konfigurationsbeispiel zeigt die Kombination der beiden Elemente *Camera* und *Mouse-Controller* zu einer einsatzfähigen Kopfsteuerung für den Mauszeiger, die Personen mit Lähmungen der Gliedmaßen als Eingabemedium dienen kann. Die Bewegung des Cursors und verschiedene Klick-Aktionen können dadurch mittels Mimik und Kopfbewegungen ausgeführt werden, ohne die Befestigung von Markern oder Elektroden. Die Grundlage für eine Auswertung der Kopfposition bietet die Option *feature-tracking* des *Camera*-Elements. Ist diese Option gewählt, wird vom *Camera*-Element ein Gesichtserkennungsalgorithmus ausgeführt und die berechneten Positionen von Nase und Kinn werden an den Output-Ports des Elements ausgegeben.

Bei erkannter Gesichtspose werden die Nase mit einem grünen Punkt und das Kinn mit einem gelben Punkt markiert. Ergibt die Plausibilitätsprüfung der Punkte zu große Fehlerwerte, werden die Punkte rot dargestellt, und eine neue Initialisierung der Gesichtserkennung wird durchgeführt.

Die Ausgabewerte des Camera-Elements:

- Ports 1 und 2 (*x/y-Motion*) : relative Änderung der Position der Nase in Pixel
- Ports 3 und 4 (*x/y-Click*) : relative Änderung der Position des Kinns in Pixel

Diese Informationen werden durch Transformationen für das *MouseController* - Element verwendbar gemacht: *x-* und *y-Motion* werden über wenige Messwerte gemittelt (etwa 50 Samples, was ein Zittern vermindert) und anschließend mit einem wählbaren Faktor skaliert. Durch die Skalierung kann die Reaktionsgeschwindigkeit bzw. Genauigkeit der Positionierung bestimmt werden. In *x*-Richtung ist der Faktor negativ, da eine Spiegelung nötig ist, um eine gleichsinnige Bewegung des Maus-Zeigers zu erzielen. Die berechneten Werte dienen als Input für die Ports *xPos* und *yPos* des *Mouse* - Elements. Das Element wird für beide Richtungen im *Integrator* - Modus betrieben, um die relativen Positionsänderungen an den Input-Ports in eine absolute Position am Bildschirm umzuwandeln.

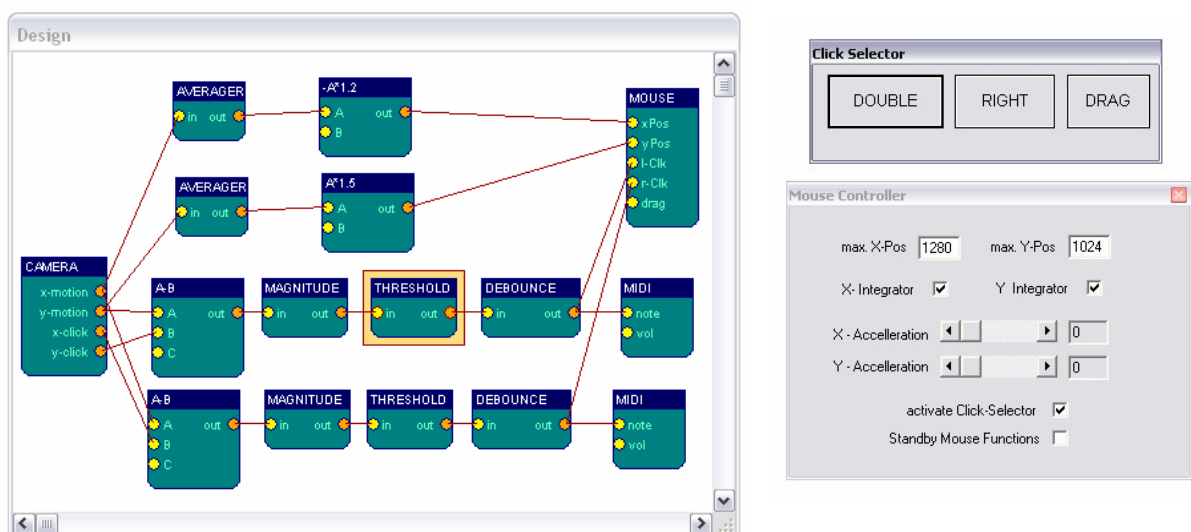


Abb. 8.16: Design-Konfiguration für die Kamera-Maus

Um Links- und Rechtsklicks durchzuführen, werden Kinnbewegungen analysiert. Dazu wird zunächst die relative Änderung von Nase und Kinn bestimmt, indem X- und Y-Werte von Nasen- und Kinnposition durch zwei *Expression-Evaluator* - Elemente subtrahiert werden. Die Bewegung des Kopfes zur Steuerung der Maus (moderate Bewegungen und Drehungen) ändern die Position von Nase und Kinn in nahezu identischer Weise, die Differenz ist nahe Null. Wird aber nur das Kinn horizontal oder vertikal bewegt, erscheinen Schwankungen der Differenz der X- bzw. Y- Werte. Um die Stärke dieser Bewegungen zu messen, werden zwei *Magnitude* - Elemente verwendet, deren Passband auf 3.5 -5.5 Hz eingestellt ist. Dadurch werden langsame Änderungen unterdrückt und schnelle Kinnbewegungen erkannt. Nachfolgende *Threshold* - Elemente legen den Schwellwert fest, ab dem ein Mausclick ausgelöst werden soll. Das *Debounce* - Element verhindert Mehrfachklicks durch ungewolltes Pendeln um den Schwellwert. Die so gewonnene Triggerfunktion für linke und rechte Mausclicks wird mit den Input-Ports *l-clk* und *r-clk* des *Mouse* - Elements verbunden.

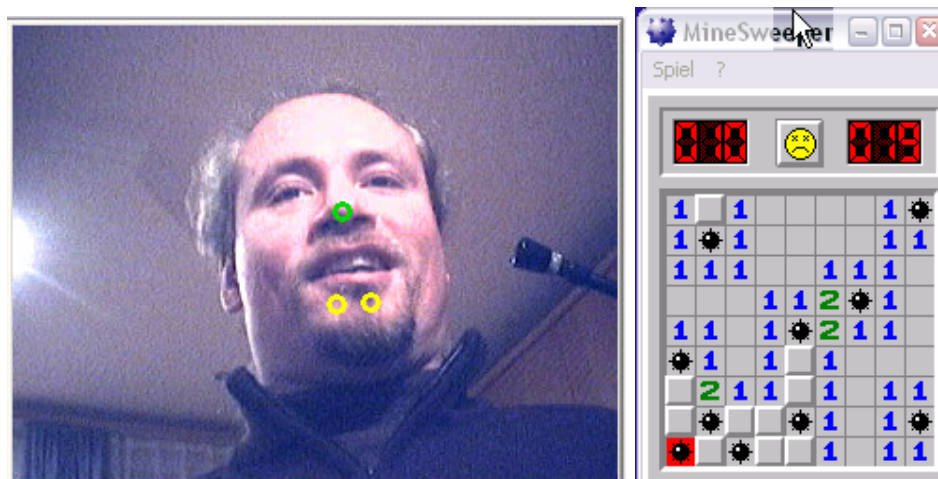


Abb. 8.17: Test der Kamera-Maus mit dem Spiel Minesweeper
(die Anzeige der Kinnposition erfolgte in dieser Version mit zwei Punkten)

9. Evaluierung von Ergebnissen

9.1 Beurteilung der Erreichung von Designzielen

Im Folgenden werden die Charakteristika der implementierten Applikation mit den in Kapitel 5.1 erhobenen Zielen und Anforderung verglichen:

- Kostengünstige Realisier- und Reproduzierbarkeit

Durch die weitgehende Verwendung von Open-Source-Komponenten und umfangreichen Eigenentwicklungen wurde die Software ohne finanzielle Aufwendungen hergestellt. Da das System in offener und dokumentierter Weise nun als Open-Source-Projekt vorliegt, kann sie kostenfrei genutzt sowie reproduziert und erweitert werden. *BrainBay* unterstützt alle derzeit verfügbaren Hardwareplattformen des *OpenEEG*-Projektes, wodurch mit den kleinstmöglichen Hardwarekosten von etwa 150-300 Euro eine Ableitung verschiedener Biosignale mit 2 - 6 Kanälen möglich wird. Handelsübliche Webcams mit einem Preis von ca. 50 Euro können zur Gesichtsdetektion und Bewegungsverfolgung im Sinne einer Kopfsteuerung für den Mauszeiger eingesetzt werden. Somit wurde für weniger als 400 Euro ein System realisiert, das sowohl EEG, EMG, EOG und andere Biopotenziale als auch visuelle Informationen über Körperbewegungen für den Einsatz im Biofeedback und für die Steuerung alternativer Eingabesysteme verfügbar macht.

- Adaptierbarkeit und intuitive Verwendbarkeit

Die grafische Konfiguration der Verarbeitungsmechanismen bewirkt eine Flexibilität in der Herstellung und Adaption von Designkonfigurationen für spezielle Anwendungsbereiche. Die derzeit realisierten Basisfunktionen und -transformationen erlauben den Einsatz digitaler Filter, die Spektralanalyse, eine Laufzeitauswertung von Ausdrücken, das Mixen und Vergleichen von Signalen uvm. Die Konfigurationselemente sind sowohl in ihrem Erscheinungsbild auf der grafischen Benutzeroberfläche als auch in ihrer Implementierung und in der internen Verarbeitung modulare Einheiten, die nach dem Prinzip der objektorientierten Programmierung aufgebaut sind. Jedes Element besitzt einen Benutzerdialog für die Parametereinstellungen, der meist selbsterklärend und leicht verwendbar ist.

- Echtzeitverarbeitung und Echtzeitmodifikation von Einstellungen

Abhängig von der Art und Komplexität der Designkonfiguration können interne Verarbeitungsgeschwindigkeiten von mehreren kHz erreicht werden, Signalgeneratoren und mehrere Filterkanäle wurden bei 40kHz getestet. Normalerweise sind Abtastraten von 100-1000Hz in der Biosignalverarbeitung ausreichend. Bei diesen Samplefrequenzen können auch aufwändige Berechnungen wie 3D-Darstellung der Spektralanalyse, Mustererkennung in Signalen oder Steuerung von Multimedia-Ausgaben in Echtzeit durchgeführt werden.

- Offenheit des Laufzeitsystems, Interoperabilität

Durch die *EDF*- und *File-Writer* Elementen besteht die Möglichkeit, Dateien im *EDF*- oder *CSV*- Format zu importieren bzw. zu exportieren. Durch das *TCP*-Element wird das Senden und Empfangen von Biosignaldaten über Netzwerk / Lan-Verbindung ermöglicht. Somit besteht die Funktionalität für Anwendungen im Bereich Telemetrie und Monitoring.

- Verwendbarkeit als Biofeedback-Software / zur Muskelrehabilitation

Die Extraktion und Skalierung gewünschter physiologischer Parameter aus der Biosignalmessung ist durch die freie Kombinierbarkeit der Verarbeitungselemente möglich. Optoakustische Ausgabemöglichkeiten für das Biofeedback sind durch die Elemente *Midi*, *SoundPlayer*, *MediaPlayer*, *Counter*, *Threshold/Meter*, *ParticleSystem* und *Ballgame* gegeben.

- Anwendbarkeit als Human Computer Interface

Die Kombination des *Mouse-Controller* - Elementes mit anderen Verarbeitungselementen erlaubt eine Steuerung des Mauszeigers und eine Auslösung von Mausklicks aufgrund von Muskelkontraktionen, Sensorwerten oder Kopfbewegungen. Es besteht die Möglichkeit, die Software automatisch beim Starten des Computers aufzurufen bzw. im Hintergrund auszuführen, sodass die Mauskontrolle zur Verwendung anderer Software wie einer Bildschirmtastatur oder einer Umgebungssteuerung eingesetzt werden kann.

Zusammenfassend kann gesagt werden, dass viele der Designziele mit der derzeitig vorliegenden Softwarelösung erreicht worden sind. Dennoch zeigten sich vor allem bei der Verwendung der Software durch Dritte einige Unklarheiten bzw. Änderungs- und Erweiterungswünsche.

9.2 Bestehende Einschränkungen

BrainBay befindet sich in einem Prozess laufender Veränderungen und Entwicklungen. Viele der verbesserungswürdigen Details treten erst bei einer praktischen Verwendung der Software zu Tage. Nachdem *BrainBay* als Downloadpaket im Rahmen des *OpenEEG* - Projektes verfügbar ist (siehe Anhang A *BrainBay Homepage*), konnten einige Probleme durch das Feedback der internationalen Usergemeinschaft erkannt und behoben werden. Ein großer Teil der Verbesserungswünsche betraf bisher Unklarheiten bzw. Fehler in der grafischen Benutzeroberfläche, etwa verdeckte Fenster, fehlende Statuszeile, Probleme bei der Miniaturisierung und Wiederherstellung des Hauptfensters, Beschriftung in grafischen Ausgabedialogen usw. Diese Fehler sind im Allgemeinen leicht zu korrigieren und werden bis zur nächsten Release der Distribution behoben.

Ebenso kamen von manchen Verwendern Anregungen zur Erweiterung der Konfigurationselemente, von einigen Teilnehmern wurden die Änderungen auch aktiv unterstützt. Dazu gehörten die Darstellung von Float-Werten durch das *Counter*-Element, Ausgabe von Werten in eine *CSV*-Textdatei durch den *File-Writer*, dynamische Grenzen für das *Threshold*-Element und die Programmierung des *Correlation*-Elements.

9.2.1 Längerfristige Pläne zur Verbesserung der Applikation

Neben laufenden Korrekturen durch Feedback bestehen einige grundsätzliche Einschränkungen, die seit längerem bekannt sind und deren Lösung intensivere Veränderungen oder Erweiterungen der Programmstruktur zur Voraussetzung haben.

- Die gleichzeitige Verbindung zu mehreren Biosignalverstärkern
Dies ist derzeit nicht möglich, da die interne Struktur des EEG-Elementes auf einen exklusiven Betrieb mit einem Verstärker ausgelegt ist. Die Verwendung mehrerer Geräte könnte jedoch für Mehrkanalableitungen sinnvoll sein.
- Der Austausch beliebiger Datentypen zwischen den Elementen
In der aktuellen Software werden Float-Werte durch die Konfigurationselemente propagiert. Es trat noch nie ein Fall auf, wo die Weiterleitung ganzer Datenarrays oder anderer Datentypen notwendig oder sinnvoll gewesen wäre. Prinzipiell sind diese Fälle aber durchaus denkbar (etwa für die Weiterverarbeitung des gesamten Frequenzspektrums einer FFT-Analyse).
- Die Gruppierung beliebiger Konfigurationselemente
Die Zusammenführung von Funktionsblöcken zu einem neuen Element würde die Übersichtlichkeit größerer Konfigurationen erhöhen. Diese Blöcke könnten auch einzeln abgespeichert und für andere Konfigurationen verwendet werden.
- Die Herausführung wichtiger Konfigurationsparameter durch einen übersichtlichen Benutzerdialog.
Gerade bei komplexen Konfigurationen verlieren Benutzer und Benutzerinnen des Systems leicht den Überblick. Die Verfügbarkeit eines individuell konfigurierbaren Dialogfensters für die Einstellung der jeweils relevanten Parameter ist vor allem für den Einsatz im HCI-Bereich eine notwendige Verbesserung. Das Skindialog-Element stellt diese Funktionalität schon in Ansätzen zur Verfügung, jedoch bedarf es noch eines komfortablen Editors für das Dialogfenster.
- Die Verbesserung des Gesichts- und Bewegungstrackings
Das Camera-Element bietet zum gegenwärtigen Zeitpunkt eine Gesichtserkennung auf Basis einer *Haar-Classifler-Cascade* und ein Bewegungstracking von Merkmalen durch den *Lucas-Kanade Optical Flow* Algorithmus. In der Praxis zeigten sich wie erwartet Unzulänglichkeiten dieser Algorithmen für eine zuverlässige Cursorsteuerung. Dies liegt vor allem am *Drifting* des *Lucas-Kanade* - Trackings, wodurch die anvisierten Punkte (Nase und Kinn) mit der Zeit und vor allem durch Verdeckungen des Gesichts oder schnelle Bewegungen horizontal und vertikal abgelenkt werden. Obwohl durch Heuristiken und eine Plausibilitätsprüfung die Situation verbessert werden konnte und das System prinzipiell einsetzbar ist (siehe 9.3), gibt es immer noch Kombinationen von Bewegungen, die ein Klicken für mehrere Sekunden unmöglich machen oder Mehrfachklicks auslösen. Diese Faktoren sprechen derzeit noch gegen eine dauerhafte Verwendung als Mausersatz. Die Stabilität könnte durch die Verwendung von *Active Appearance Models* zur Gesichts- und Mimikerkennung sicherlich erhöht werden.

9.3 Evaluation des Settings zur Muskelrehabilitation

Für den Test der Design-Konfiguration zum Muskeltraining (siehe 8.3) wurde ein Laptop Computer mit *Pentium IV (M)* -Prozessor und einer Taktfrequenz von 1,6 GHz verwendet (256MB RAM, Betriebssystem *Windows XP professional*). Die Grafikkarte unterstützt eine Hardwarebeschleunigung für *OpenGL*-Kommandos. Obwohl mit schnelleren Systemen eine bessere Performance vor allem bei grafikintensiven Darstellungsformen erzielt werden könnte, wurde ein Computer gewählt, der mittlerweile preisgünstig am Gebrauchtcomputermarkt zu erwerben ist, da viele Menschen, die an einer kostengünstigen Lösung für Biofeedback interessiert sind, nicht die neueste Hardware zur Verfügung haben. Um die praktische Verwendbarkeit des Systems für die Muskelrehabilitation zu überprüfen, wurden mehrere Trainingseinheiten mit einer 70-jährigen Patientin durchgeführt, die seit 20 Jahren an einer progredienten Form der Multiplen Sklerose leidet. Die Patientin ist seit etwa drei Jahren nicht gehfähig und erlitt vor zwei Jahren eine Amputation des rechten Beines unterhalb des Knies, verursacht durch eine diabetesbedingte Insuffizienz der Blutversorgung. Tägliche Übungen und Physiotherapie sind sehr wichtig, um den Allgemeinzustand und die Beweglichkeit der Patientin zu erhalten und zu verbessern. Sie hatte zuvor noch nie mit einem Biofeedbacksystem trainiert. Bei der ersten Ableitung und Rückmeldung von Muskelpotenzialen des *M. femoris* (Oberschenkelmuskulatur) war die Probandin sichtlich überrascht, dass es "funktioniert" und sie starke Ausschläge auf der Skala und akustisches Feedback durch die Muskelkontraktion hervorrufen konnte. Die Muskelaktivität wurde mit Standard EKG-Elektroden (*Blue Sensor*) abgenommen und über ein *MonolithEEG*-System verstärkt. Die Auswertung erfolgte durch ein Designkonfiguration mit Bandpass-Filtern und optoakustischem Feedback (siehe 8.3)

9.3.1 Ergebnisse und Rückmeldungen

Die neue Situation des Muskelfeeds über einen Computer wurde von der Probandin gut akzeptiert und sie trainierte ausgiebig das Erreichen der nötigen Kontraktionslevels, um einen Feedback-Ton auszulösen. Diese Aktivierungsniveaus wurden auf ihre momentanen Fähigkeiten eingestellt, beginnend mit niedrigen Grenzwerten, bei denen die akustische Rückmeldung schon durch sehr leichte Muskelkontraktionen hervorgerufen werden konnte. Im weiteren Verlauf der Übung konnte das Aktivitätsniveau auf den dreifachen Wert erhöht werden. Dies wertete die Patientin als Erfolg, und erhielt so auch Motivation, das Training weiter fortzusetzen. Bei der dritten Trainingseinheit führte die Patientin abwechselnd mit dem linken und rechten Bein jeweils 250 starke Kontraktionen durch.

Es stellte sich heraus, dass die unmittelbare Transparenz des Feedback-Mechanismus zur Erhaltung bzw. zum Aufbau der Trainingsmotivation sehr wichtig ist. Das Feedback sollte dafür so klar und einfach wie möglich gestaltet sein, vor allem wenn wenig feinmotorische Kontrollmöglichkeit besteht. Die Rückmeldung durch Balkendiagramme und Zähler für Anzahl der erreichten Kontraktionen erwies sich in diesem Fall geeigneter als die Verwendung des Ballspieles, wo durch Kontraktion von linkem und rechtem Muskel ein Schläger gesteuert wird, der den Ball im Spielfeld halten soll. Diese Art des Feedbacks wurde von der Patientin als eher demotivierend empfunden, da sie meist nicht in der Lage war, den Schläger in geeigneter Weise zu kontrollieren. Das Biofeedback kann also nur dann sinnvoll zur Therapie beitragen, wenn die Feedbackmechanismen optimal auf den Stand der Fähigkeiten an-

gepasst werden, wozu sich die erstellte Software prinzipiell eignet. Ein Problem ist die seltene Verfügbarkeit von Computern im Haushalt von pflegebedürftigen älteren Menschen. So ist auch im Haushalt der Testperson kein Computer vorhanden, und die Verwendung eines einfachen *stand-alone* Gerätes für das Muskelfeedback wäre eine denkbare Alternative. Im konkreten Fall überlegt die Patientin dennoch die Anschaffung eines Laptop-Rechners, da sich dieser auch für andere Möglichkeiten des Trainings (etwa Feedback am Hometrainer siehe 8.4) eignet.

Gerade bei Personen, die viel Zeit im Bett verbringen müssen, kann das Biofeedback durch ein Monitoring der ausgeführten Bewegungsleistung die Motivation zum Trainieren verstärken. Keinesfalls kann eine konventionelle Physiotherapie mit geführten und aktiven Bewegungen durch ein Biofeedbacksystem ersetzt werden.

9.4 Evaluation der Kamera-Maus Konfiguration

Die Kamera-Maus Konfiguration (siehe 8.5) wurde durch zwei männliche Testpersonen (A und C) im Alter von 38 bzw. 32 Jahren evaluiert. A leidet nach einem Motorradunfall vor 19 Jahren an einer Querschnittlähmung ab dem sechsten Halswirbel, kann also nur seinen Kopf kontrolliert bewegen. Er verwendet häufig einen Computer zur Kommunikation sowie zur Unterhaltung und setzt eine Sprachsteuerung zur Umgebungskontrolle ein. A nutzt seinen Computer meist mittels Mundstab und hat eine erstaunliche Genauigkeit und Geschwindigkeit mit diesem Eingabehilfsmittel erreicht. Für A ist das Kamera-Maus - System neu, allerdings kennt er die IR-gesteuerte *Origin-Headmouse*, die er aber nicht verwendet, da zwar die Cursorsteuerung durch Kopfbewegungen damit möglich ist, er aber keine geeignete und für ihn angenehme Möglichkeit zur Auslösung von Mausklicks in Verbindung mit diesem Gerät gefunden hat. (Die 2000 € teure Headmouse liegt seither unbenutzt im Kasten). C hat schon ausgiebig mit dem Kamera-Maus - System trainiert, während A nur 10 min Trainingszeit bis zur Evaluation der Ergebnisse hatte. Die verwendete Designkonfiguration (8.5) wurde zusammen mit einer *Logitech Quickcam 5000 pro* Webcam eingesetzt. Zwei Applikations-Umgebungen wurden zur Auswertung der Ergebnisse herangezogen: Eine Bildschirmtastatur (*SAW*) und ein Strategiespiel mit kleinen Selektionsflächen (*Minesweeper*).

9.4.1 Schreiben mit Switch Access to Windows (SAW)

Switch Access to Windows (SAW) ist eine kostenlos verfügbare Bildschirmtastatur, welche die Erstellung eines flexiblen, grafischen Selektionssystems für die Eingabe von alphanumerischen Zeichen und das Auslösen von Makros für Systemfunktionen erlaubt. Der Schreibfokus kann auf beliebige Fenster gelegt werden, sodass Standard-Textverarbeitungen, Email-Clients oder Web-Browser benützt werden können. Neben der Auswahl über Mausklicks können die Funktionen von SAW auch durch alternative Eingabemodalitäten wie externe Schalter und automatisches Scanning ausgewählt werden.

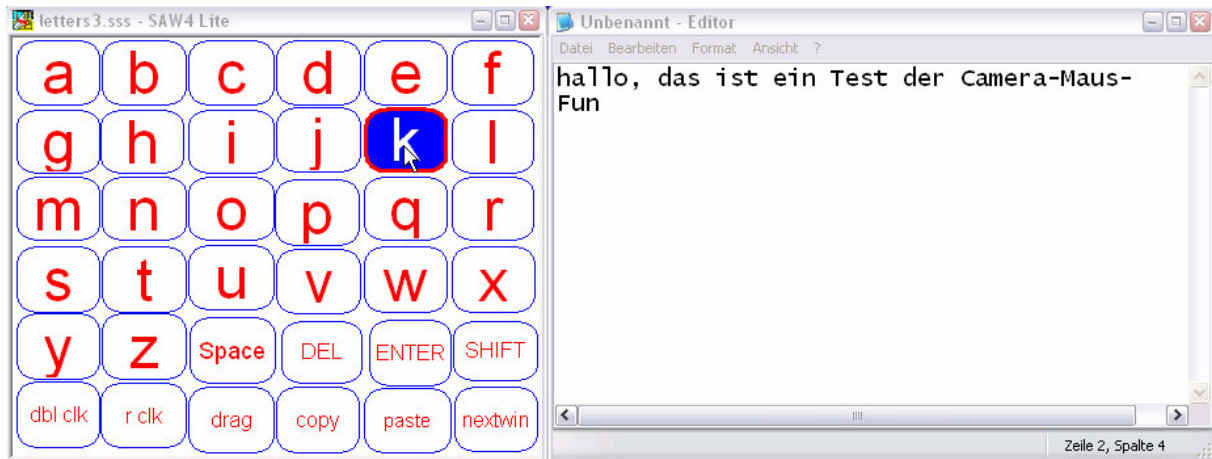


Abb. 9.1: Verwendung von Switch Access for Windows mit der Kamera-Maus - Schreiben mit Notepad

Die Größe und die Anordnung der Buchstaben einer Bildschirmtastatur kann bei SAW beliebig gewählt werden. Hier wurde eine moderat große und alphabetische Anordnung getroffen, damit noch genug Platz am Bildschirm bleibt und die Eingewöhnungsphase erleichtert wird. Die Beschleunigung des Mauszeigers kann durch die Designkonfiguration beliebig eingestellt werden, wobei ein Kompromiss aus Geschwindigkeit und Genauigkeit zu treffen ist, da bei hoher Beschleunigung schon eine geringe Kopfbewegung eine große Bewegung des Mauszeigers bewirkt und der Zeiger leicht zittert. Die gewählte Beschleunigung ist daher auch immer vom gewünschten Einsatz abhängig, sie wurde hier größer gewählt als etwa beim Einsatz für das *Minesweeper*-Spiel (siehe 8.3.2), sodass eine Bewegung des Zeigers vom linken unteren zum rechten oberen Bildschirmrand durch die Lageänderung des Kopfes in etwa zwei Sekunden möglich war. Um eine brauchbare Schreibgeschwindigkeit zu erzielen, erwies sich die zuverlässige und schnelle Erkennung von Mausclicks durch die Kinnbewegung als kritisch: Durch Veränderungen des Bandpass Filters für die Erkennung der Kinnbewegung konnten hier substantielle Verbesserungen in den ersten Trainingsminuten erreicht werden. Die Einstellung für ein schmales Passband im Bereich von 2-3 Hz erwies sich hier als zielführend, um die Beeinträchtigung durch die großflächigeren Kopfbewegungen auszufiltern und die Geschwindigkeit der Kinnbewegungen in vertikale Richtung (für den Linksklick) und in horizontale Richtung (für den Rechtsklick) gut zu erkennen. So konnte mit wenigen Einstellungen die Erkennungsrate verbessert und die physische Anstrengung beim Klicken vermindert werden. Andererseits wurde dadurch auch die Anzahl der falsch-positiv erkannten Klicks erhöht, da die Kinnbewegungen beim Reden oft fälschlich als Klick-Kommandos interpretiert wurden. Eine einfache Zu- und Abschaltbarkeit der Klickfunktion sowie fünf Presets für verschiedene Sensitivitäten wären hier sinnvolle Erweiterungen.

Schreiben mit SAW und der KAMERA-MAUS (10 Minuten)	Testperson C	Testperson A
Zeichen gesamt	337	250
falsch ausgewählt	7	15
Feld verfehlt	2	5
Zeichen pro Minute	32,8	23

Tabelle 9.1: Ergebnisse für das Schreiben mit der Kamera-Maus unter Verwendung von SAW

9.4.2 Test der Genauigkeit: Das Spiel 'Minesweeper' :

Das Spiel "*Minesweeper*" ist ein aussagekräftiger Test für ein Mausersatz-System, da die Selektionsflächen des Spiels sehr klein und unmittelbar nebeneinander angeordnet sind. Obwohl A kaum Trainingsmöglichkeiten mit dem System hatte war die Kontrolle über die Mausfunktion überraschend gut und er konnte das Spiel schon beim ersten Versuch prinzipiell nutzen.

Eine Erhöhung der Genauigkeit bei entsprechender Verminderung der Beschleunigung resultierte in einer höheren räumlichen Auflösung und geringerem Zittern des Zeigers, sodass die Verwendbarkeit des Spieles weiter verbessert wurde und bei der Evaluation gute Trefferquoten trotz der kleinen Schaltflächen erzielt werden konnten.

KAMERA-MOUSE - Spiel Minesweeper	Testperson C	Testperson A
Klicks gesamt	125	100
Zeit	5 min	5 min
verfehlte Felder	10	17
korrekte Treffen	92 %	87%

Tabelle 9.2: Ergebnisse von Minesweeper mit der Kamera-Maus

Wie beim Test der Bildschirmtastatur zeigte sich das System prinzipiell für einen Einsatz geeignet. Abermals zeigte sich die Notwendigkeit benutzerfreundlicher Einstellmöglichkeiten für verschiedene Genauigkeitsstufen. Ein Problem entsteht durch das Abdriften des *Lukas-Kanade*-Algorithmus, weil dadurch manchmal der Kontaktpunkt zu Nase oder Kinn in zu großer Weise verschoben wird. Hier sollte die Plausibilitätsprüfung der Punkte verbessert bzw. ein kontinuierlicher Algorithmus zur Adaption verwendet werden. Als gut zu bewerten ist die Adaption des Systems auf veränderliche Lichtverhältnisse, da sowohl die Kamera bei schlechter Beleuchtung noch gut funktioniert als auch der *LK*-Algorithmus auf Veränderungen der Lichtstärke adaptiert.

9.5 Zusammenfassung und Diskussion

Durch die Kombination von *BrainBay* mit den Biosignalverstärkern des *OpenEEG*-Projektes steht ein sehr flexibles System zur Messung und Auswertung von Biosignalen zu einem Preis von weniger als 200 Euro zur Verfügung. Die Software eignet sich für die Analyse von EMG-, EEG-, EKG- und EOG - Daten und erlaubt eine quantitative und multimediale Darstellung von Messwerten für das Biofeedback in Echtzeit. Durch zusätzliche Sensoren können auch die elektrodermale Aktivität (EDA), Atemaktivität, Körpertemperatur oder Parameter wie Lageänderung und Beschleunigung gemessen und verarbeitet werden. Die Signale werden durch die Anordnung von Verarbeitungselementen mittels grafischer Benutzeroberfläche transformiert, kombiniert oder klassifiziert, eine Archivierung, Wiedergabe und Telemetrie von Signaldaten ist im *EDF*-Format möglich.

Die Software bietet Funktionen zur Auswertung des Live-Kamerabildes einer Webcam mittels Algorithmen für Gesichtserkennung und Bewegungserfassung. Die hierbei erhaltenen Informationen über Bewegungen von Kopf und Kinn dienen zur Steuerung des Mauszeigers und zur Auslösung von Maus-Klicks. Die prinzipielle Verwendbarkeit des Systems als Mausersatz bzw. als Alternative zur anderen Eingabehilfen für Menschen mit Querschnittslähmung bei freier Beweglichkeit des Kopfes wurde evaluiert. Mit einer Bildschirmstatur konnten dabei Schreibgeschwindigkeiten von etwa 25 Zeichen pro Minute erreicht werden. Dies stellt zwar verglichen mit aktuellen Videookulographiesystemen eine langsame Transferrate dar, jedoch sind derartige Systeme viel teurer und schränken eine oft noch vorhandene Beweglichkeit des Kopfes weiter ein. Das Kamera-Maus - System zeigt vor allem bei einer häufigen Verwendung substantielle Vorteile gegenüber infrarotbasierten Eingabehilfen, die einen im Gesicht aufgeklebten Reflektorpunkt und oft einen externen Schalter für die Mausclicks benötigen. Auch gegenüber Eingabesystemen, die auf Elektrodenableitungen beruhen, erweist sich die berührungsfreie Erfassung von Bewegung und Mimik als komfortabel. Dennoch gibt es viele Personen, für die kamerabasierte Systeme aufgrund von unkontrollierbaren Bewegungen nicht gut geeignet sind. Hier kann ein Einsatz von Elektroden zur Ableitung des EMG, EOG oder EEG eine sinnvolle Alternative sein.

Die freie Verfügbarkeit der Software als Open-Source Projekt bietet einige Vorteile auch für den Test und die Weiterentwicklung des Systems. *BrainBay* wird über eine eigene Internetseite (<http://brainbay.lo-res.org>) angeboten (siehe Anhang A). Dort kann die Software als ausführbare Datei sowie als Quellcode bezogen werden und Beschreibungen des Systems sowie ein Benutzerhandbuch sind in englischer Sprache verfügbar. In den letzten Monaten kamen dadurch Kooperationen mit Mitgliedern zweier Universitäten und mehreren Privatpersonen zustande, es wurden drei neue Verarbeitungselemente programmiert, die in die aktuelle Distribution der Software aufgenommen wurden.

Durch das Feedback der internationalen Benutzer und Benutzerinnen von *BrainBay* konnten einige Schwachstellen aufgedeckt und korrigiert werden, andere Probleme bestehen noch und sind Gegenstand zukünftiger Verbesserungen.

9.6 Zukünftige Entwicklungen

Es besteht eine ganze Reihe von Ideen zur Erweiterung der aktuellen Software (siehe 8.2). Um diese auch umsetzen zu können, wird die Vorstellung des Projektes bei nationalen und internationalen Subventionsgebern ein wichtiger Arbeitsschritt sein. Jedenfalls ist geplant, die Software weiterhin als Open-Source- Applikation anzubieten und mit Hilfe des Feedbacks aus der Community Verbesserungen durchzuführen.

Ein weiteres Ziel für die Zukunft ist es, *BrainBay* in Verbindung mit bestehenden und neuen Hardwarelösungen in der Praxis zu testen, und im Bereich der Kommunikationsunterstützung einzusetzen.

A Die *BrainBay* - Homepage

A.1 Aufbau und Verfügbarkeit

Die *BrainBay*-Homepage besteht seit der ersten lauffähigen Projektversion (2004). Im Zuge der Arbeiten für die Diplomarbeit wurde die Homepage neu strukturiert und mit Manuals für Entwicklerinnen und Entwickler sowie einem Benutzerhandbuch ausgestattet. Das Hauptmenü bietet nun Sektionen für eine prinzipielle Erklärung der Einsatzmöglichkeiten, Screenshots, die gesammelten Dokumente und Publikationen, eine Liste der Updates und den Download-Bereich für Programm und Quellcode:

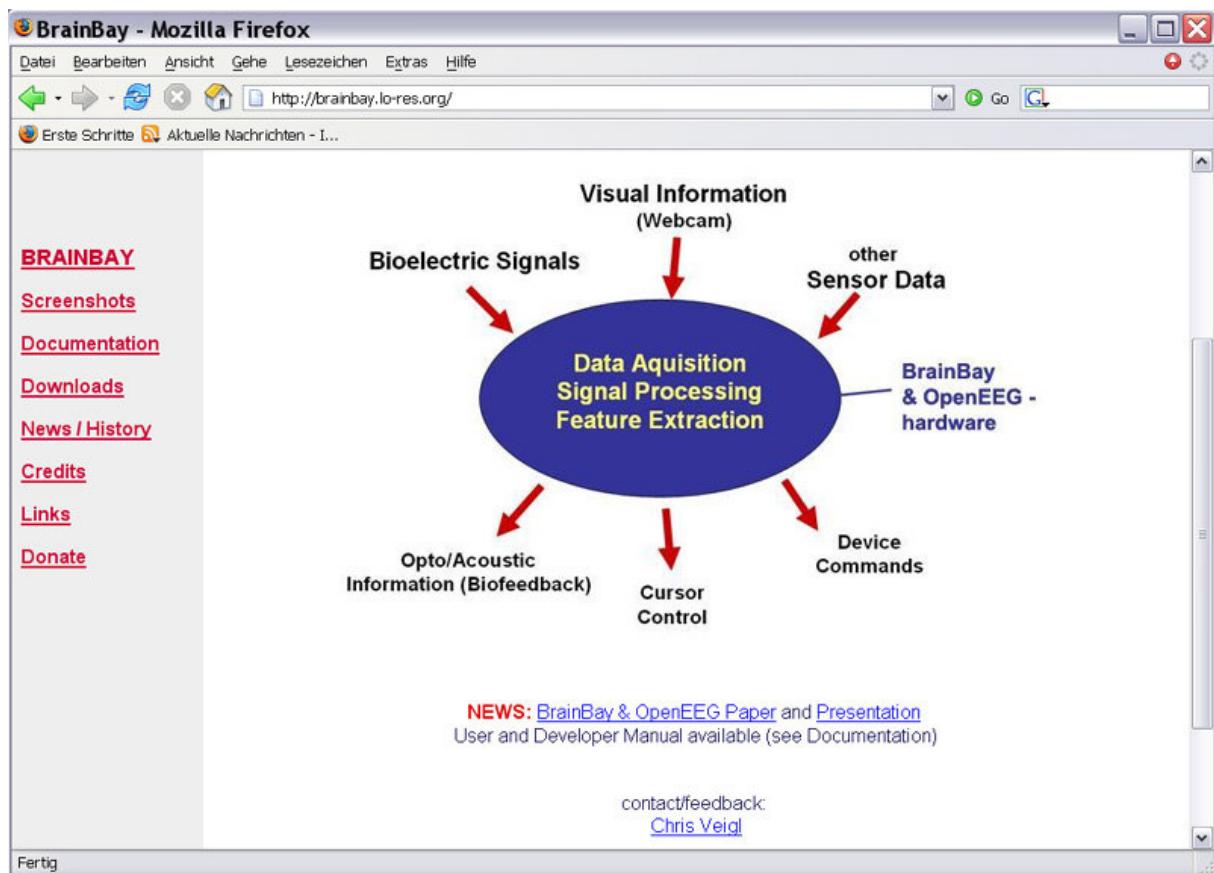


Abb. A.1: Die BrainBay-Homepage, erreichbar unter <http://brainbay.lo-res.org>

Die Homepage wird von einem österreichischen Provider aus der Kunst- und Free-ware-Szene seit mehreren Jahren kostenlos gehostet (<http://lo-res.org>). Diese Unterstützung ist sehr wertvoll für das Projekt und ermöglicht die kostenfreie Verbreitung der Software. Permanente Links bestehen vom *OpenEEG*-Projekt und von privaten Seiten mehrerer Benutzer und Benutzerinnen.

A.2 Homepagezugriffe und Statistiken

Die Statistiken von *Google-Analytics* zeigen die Aufrufe der Seite aus dem Internet mit Angabe der geographischen Lokation der Aufrufer. Die folgende Weltkarte zeigt Seitenbesuche mit Downloads der vergangenen zwei Monate:



Abb. A.2: Geographischer Ursprung der Zugriffe auf die Homepage

Es lässt sich eine starke Konzentration auf Europa und die Vereinigten Staaten erkennen, wobei aber auch Aufrufe aus Russland, China, Indien, Australien, Neuseeland und Brasilien vertreten sind. Die Aufschlüsselung nach Ländern zeigt eine starke Präsenz von Aufrufen aus den USA, gefolgt von Deutschland, Kanada, Italien und den Niederlanden. Österreich ist in dieser Liste an zwölfter Stelle vertreten. Das verhältnismäßig große Interesse aus den USA ist vermutlich auch durch die stärkere Popularität von Bio- und Neurofeedback in den Vereinigten Staaten zu erklären.

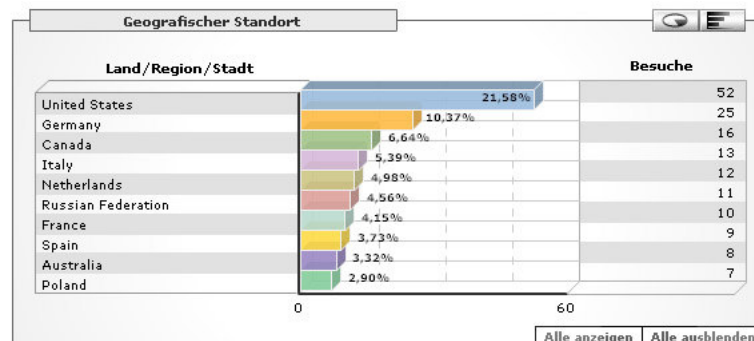


Abb. A.3: Zugriffsstatistik nach Ländern

Der Prozentsatz der wiederkehrenden Besucher liegt derzeit bei 11,9 %. Vor allem nach der Ankündigung eines Software-Updates in der *OpenEEG*-Liste ist ein Anstieg der Zugriffe und der Downloads von der Seite zu bemerken.

Quellen der Abbildungen

Abb.1.1: William Gilbert, Elektroskop, in: [MAL95]

Abb.1.2: Leydener Flasche, Darstellung von ca. 1800, Online Ressource (12.8.2006): <http://de.wikipedia.org/wiki/Bild:Leid-flasch.gif>

Abb 1.3: Erregung eines Nerv-Muskel-Präparates, Luigi Galvani, 1760, in: [MAL95]

Abb 1.4: Voltaische Säule: <http://chem.ch.huji.ac.il/~eugeniik/history/volta.htm>, Online-Ressource (10.7.2006): http://www.ktf-split.hr/glossary/en_o.php?def=voltaic%20pile

Abb.1.5: Nobili-Galvanometer von Leopoldo Nobili, in: [MAL95]

Abb.1.6: Schematische Darstellung einer Nervenzelle, in: Carlson, Niel. A. (1992). Foundations of Physiological Psychology. Needham Heights, Massachusetts: Simon & Schuster. pp. 36

Abb.1.7: Zellmembran und eingelagerte Membranproteine, Online-Ressource (12.8.2006) <http://www.physiology.de/kanal/iono01.htm>

Abb.1.8: Elektrisches und chemisches Potenzial an der Membran, in: [HEL06]

Abb.1.9: Wanderung kleiner Ionen durch die Membran, in: [HEL06]

Abb.1.10: Na/K-ATPase, in: Dr. Gunnar Seemann, in: [SEE06]

Abb.1.11: Verlauf des Aktionspotenzials, Online-Ressource (12.8.2006): <http://content.answers.com/main/content/wp/en/thumb/0/02/300px-Action-potenzial.png>

Abb.1.12: Aktivierung und Inaktivierung des Natriumkanals, Online-Ressource (1.6.2006): <http://www-ibt.etec.uni-karlsruhe.de/postscript/bsfss2003t2.pdf>

Abb.1.13: Aktivierung und Inaktivierung des Kaliumkanals, Online-Ressource (1.6.2006): <http://www-ibt.etec.uni-karlsruhe.de/postscript/bsfss2003t2.pdf>

Abb. 1.14 Neuromuskuläre Endplatte, Online-Ressource (6.8.2006): <http://www.med.uni-marburg.de/stpg/fb20/ltinst/molneurophys/SemMuskel211105LissKurz.pdf>

Abb.1.15: Transmitterfreisetzung an der Neuromuskulären Endplatte, Online-Ressource (6.8.2006): <http://www.med.uni-marburg.de/stpg/fb20/ltinst/molneurophys/SemMuskel211105LissKurz.pdf>

Abb.1.16: Zentrales und peripheres Nervensystem, Online-Ressource (10.9.2006) : http://www.biologyreference.com/images/biol_03_img0336.jpg

Abb 2.1a: Ableitung des EMG vom rechten Untersarm, in: [DAH05]

Abb 2.1b: Messung einer Muskelkontraktion, in: [DAH05]

Abb 2.2: Ableitpunkte des fazialen EMG, in: [DAH05]

Abb 2.3: EMG-Aufbereitung durch Mittelung von Absolutwerten, in: [DAH05]

Abb.2.4: Die drei Hauptachsen der EKG-Ableitung nach Einthoven bzw. nach Goldberger, in [CAR06]

Abb.2.5: Die sechs Ableitpunkte an der Brustwand nach Wilson, in [CAR06]

Abb.2.6: Der Verlauf der Erregungsausbreitung im Herzen, in: [CAR06]

Abb.2.7: Die charakteristische Form des EKG, in: [CAR06]

Abb. 2.8a: Die Entstehung des EOG durch den Dipolcharakter des Auges, Online Ressource (3.8.2006): <http://www.uni-greifswald.de/~fsrpsych/harb/EOG&EMG.pdf>

Abb. 2.8b: Die Muskel für die Augenbewegung, Online Ressource (3.8.2006): <http://www.uni-greifswald.de/~fsrpsych/harb/EOG&EMG.pdf>

Abb.2.9: Messung des EOG, Online-Ressource (3.7.2006): http://www.uni-duisburg.de/FB9/EBS/hauptteil_biosignale.html

Abb.2.10: Die Elektrodenpositionen nach dem 10/20 System, in: [SEE06]

Abb.2.11: Desynchronisation der Alpha-Wellen im EEG beim Öffnen der Augen, in: [SEE06]

Abb.2.12: Quantitative EEG-Messung, Online Ressource (20.10.2006): <http://www.appliedneuroscience.com/JTFA-CD-AbsPower-Z.JPG>

Abb.2.13: Die P300-Komponente im EEG-Signal, in: [SEE06]

Abb.2.14: Die Desynchronisation des Mu-Rhythms durch die Vorstellung einer Bewegung, in: [DUR04]

Abb. 2.15: Langsame Potenziale bei der Vorbereitung einer Bewegung, Online Ressource (13.10.2006): http://www.biosemi.com/cnv_example.htm

Abb. 2.16: Elektrodermale Aktivität, Online Ressource (12.9.2006): http://www.uni-bamberg.de/fileadmin/uni/fakultaeten/ppp_professuren/physiologische_psychologie/Biologische_Psychologie_II/M-edA25.01.2006.pdf

Abb. 2.17: Abnahme der Atmenaktivität durch dehnungssensitive Messgürtel, in: [STR00]

Abb. 2.18: Funktionsweise der Infrarot-Plethysmographie, in: [KRO00]

Abb. 2.19: Microguide Infrared Eye Movement Measuring Systems, Online Ressource (23.9.2006): <http://www.eyemove.com/biro.htm>

Abb. 2.20: Gyro-Sensor und Beschleunigungssensor, Online Ressource (24.9.2006): <http://www.sander-electronic.de/>

Abb. 2.21: Ableitungspunkte für das EMG der Rückenmuskulatur, in: [HEU00]

Abb. 2.22: Ableitungspunkte für das EMG der Gesichts- und Nackenmuskulatur, in: [HEU00a]

Abb. 2.23: Lage wichtiger Muskelgruppen für die Kontinenzreaktion, in: [CUN00]

Abb. 2.24: EMG-Sonde für das Beckenbodentraining, Online Ressource (25.9.2006): <http://www.yatego.com/q,vaginalsonde>

Abb. 2.25: Verbesserung der Bewegungsrezeption durch Biofeedback, in: [MÜL00]

Abb. 2.26: Dominante Theta-Aktivität im EEG bei ADHD, in: [DEM05]

Abb. 2.27: IntegraMouse der Firma Livetool, Online-Ressource (24.10.2006): <http://www.livetool.at>

Abb. 2.28: Sicare-Pilot Umgebungssteuerung, Dr. Hein GmbH, Online-Ressource (28.10.2006): <http://www.sicare.de>

Abb. 2.29: Steuerung eines Flugsimulators mit EMG-Signalen, in: [WHE03]

Abb. 2.30: EagleEyes: Verwendung des Computers mittels EOG, in: [GIP96]

Abb. 2.31: Kombination von EOG und Beschleunigungssensoren in einer Brille, in: [WIE03]

Abb. 2.32: HeadMouse extreme, Origin Inc., Online Ressource (25.8.2006):
<http://www.orin.com/access/headmouse>

Abb. 2.33: Tracker Pro, Madentec Inc., Online Ressource (25.8.2006):
<http://www.sforh.com/pointing/trackerpro-head.html>

Abb. 2.34: Bewegungsverfolgung durch das CameraMouse-System, in: [BET02]

Abb. 2.35: Augendetektion in Echtzeit mit einer Webcam, in: [BET3]

Abb. 2.36: Tracking der Nase für die Cursorsteuerung, in: [GOR02]

Abb. 2.37: Komponenten und Funktionsweise eines Gaze-Tracking-Systems, in: [COG05]

Abb. 2.38: Schreiben mit dem EyeGaze-System, Online-Ressource (14.10.2006):
<http://www.eyegaze.com>

Abb. 2.39: Desynchronisation des μ -Rhythmus bei der Vorstellung einer Bewegung, in: [MUP05]

Abb. 2.40: Zweidimensionale Auswertung von ERD / ERS, in: [WOL04]

Abb. 2.41: BCI-gesteuerte Orthese, in: [MUP05]

Abb. 2.42: P300-BCI für das Buchstabieren, in: [ALL03]

Abb. 2.43: Komponenten eines Direct Brin Interface, in: [KEN87]

Abb. 2.44: Struktur und Komponenten der BCI2000-Applikation, in: [SCH04]

Abb. 2.45: Beispiele für mögliche BCI-Applikationen in BCI2000, in: [SCH04]

Abb. 3.2: Unterschiede zwischen monopolarer und bipolarer Ableitung, in: [FEH89]

Abb. 3.3a: Der Elektrode-Haut-Kontakt, in: [SEE06]

Abb. 3.3b: Elektrisches Ersatzschaltbild für den Elektrode-Haut-Kontakt, in: [SEE06]

Abb. 3.4: Kapazitive Einkopplung von 50Hz-Noise, in: [SEE06]

Abb. 3.5: Aufbau eines Differenzverstärkers, in: [LIS05]

Abb. 3.6: Ausreichende Samplingrate, in: [SMI01]

Abb. 3.7: Inadequate Samplingrate - Aliasing, in: [SMI01]

Abb. 3.8: Tiefpassfilter (Sallen-Key-Schaltung), in: [SMI01]

Abb. 3.9: Symbole für die Kennzeichnung zertifizierter Medizintechnikgeräte, [BRA2] Braintronics Inc. EEG-Amplifiers User Manual, Online Ressource (23.9.2006): <http://www.braintronics.nl>

Abb. 3.10: Noraxon Myotrace200, Online Ressource (2.10.2006): <http://www.noraxon.com>

Abb. 3.11: Mindpeak Waverider, Online Ressource (2.10.2006): <http://www.mindpeak.com>

Abb. 3.12: Brainmaster Atlantis, Online Ressource (2.10.2006): <http://www.brainmaster.com>

Abb. 3.13: Symptop EEG-Amplifier, Online Ressource (2.10.2006): <http://www.symtop.com>

Abb. 3.14: Sagura Leonardo, Online Ressource (2.10.2006): <http://www.polysomnograph.com>

Abb. 3.15: g.Tec USBamp, Online Ressource (2.10.2006): <http://www.gtec.at>

Abb. 3.16: Neuroscan Synamp, Online Ressource (2.10.2006): <http://www.neuro.com>

Abb 3.17: Digital-Board und Analog-Board des ModularEEG-Verstärkers, in: [GRI03]

Abb. 3.18: Blockdiagramm der ModularEEG-Verstärkerstufen und DRL-Schaltung, in: [GRI03]

Abb. 3.19: Blockdiagramm der Digitalisierung und Datenübertragung zum PC, in: [GRI03]

Abb. 3.20: Der MonolithEEG-Verstärker, in: [MUE05]

Abb. 3.21: Die SMD-Platine des MonolithEEG (links: Unterseite, rechts: Oberseite), in: [MUE05]

Abb. 3.24: Aktive Elektroden zur Verwendung mit dem OpenEEG, in: [ORT05]

Abb. 3.25: Bluetooth Modul der Firm Promiserial, Online Ressource (12.7.2006): http://www.promiserial.de/promiesd02_e.htm

Abb. 3.26: VDrive, USB-Flash-Drive mit UART Schnittstelle, Online Ressource (12.7.2006): <http://www.vinculum.com>

Abb. 4.1: Stationäre und nichtstationäre Signale, in: [SMI01]

Abb. 4.2: Histogramm und Wahrscheinlichkeitsdichte, in: [SMI01]

Abb. 4.3: Frequenzantwort von Chebyshev-, Butterworth- und Bessel-Filtern, in: [SMI01]

Abb. 4.4: Step-Antwort von Chebyshev-, Butterworth- und Bessel-Filtern, in: [SMI01]

Abb. 4.5: Auswirkungen von Tiefpass- und Hochpassfiltern, in: [SMI01]

Abb. 4.6: Die Darstellung eines Signales im Frequenzbereich durch die DFT, in: [SMI01]

Abb. 4.7: Frequenzantwort eines Tiefpass-Filters in der Polardarstellung, in: [SMI01]

Abb. 4.8: Aufbau eines neuronalen Netzes mit 3 Schichten bzw. einer einzelnen Unit, in: [SMI01]

Abb. 4.9: Klassifikation mit Support-Vektoren, in: [BUR98]

Abb. 4.10: Einzelne Merkmale eines Harr-Classifiers, in: [BRA05]

Abb. 4.11: Analyse von EEG-Daten mit EEGLab, in: [DEL06]

Abb. 4.12: Scilab, eine freie Alternative zu Matlab, Online Ressource (1.8.2006): <http://www.scilab.org>

Abb. 4.13: 3D-Darstellung von EEG-Daten mit TEMPO, [SAM05]

Abb. 4.14: Frequenzdarstellung mit BioExplorer, Online Ressource (12.6.2006): <http://cyberevolution.com>

Abb. 4.15: FidView - Filter Designer von Jim Peters, [PET03]

Abb. 4.16: Wavelet Analyse von EEG-Daten mit BwView, [PET02]

Abb. 4.17: BioEra-Applikation für Biofeedback, Online-Ressource (1.7.2006): <http://www.bioera.net>

Abb. 5.2: Grafisches Design einer Signalverarbeitung in Simulink, [MAT06]

Abb. 5.5: Windows Message Loop, Online ressource (1.11.2006): http://www.microsoft.com/taiwan/msdn/columns/DoNet/images/MessageLoop_1.gif

Bibliographie

[ALL03] Brendan Allison: P3 or not P3: Toward a Better P300 BCI, Dissertation, University of California, San Diego (2003), Online Ressource (18.8.2006): <http://cogsci.ucsd.edu/~pineda/PAPERS/bza-thesis.pdf>

[BAK02] S. Baker, I. Matthews: Lucas-Kanade 20 years on: A unifying Framework, Robotic Institute, Carnegie Mellon University (2002), Online Ressource (23.9.2006): http://www.ri.cmu.edu/pub_files/pub3/baker_simon_2002_3/baker_simon_2002_3.pdf

[BAR00] Armando B. Barreto, PhD; Scott D. Scargle: A practical EMG-based human-computer interface for users with motor disabilities, Journal of Rehabilitation Research and Development, Vol 37, No1, Jan/Feb 2000, 53 - 64

[BAY00] J.D. Bayliss, D.H. Ballard, et al. (2000). "A virtual reality testbed for brain-computer interface research." IEEE Trans Rehabil Eng. 8 (2):188-90.

[BIT06] Sebastian Bitzer, Patrick van der Smagt: Learning EMG-control of a robotic hand: Towards Active Prosthesis, IEEE International Conference on Robotics and Automation, Orlando, Florida, 2006, IEEE 0-7803-9505-0/06

[BET02] Margrit Betke, James Gips, Peter Fleming: The Camera Mouse: Visual Tracking of Body Features to Provide Computer Access for People With Severe Disabilities, IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol. 10, No. 1, March 2002

[BET05] Michael Chau and Margrit Betke: Real Time Eye Tracking and Blink Detection with USB Cameras, Boston University Computer Science Technical Report No. 2005-12

[BIR00] Niels Birbaumer, Andrea Kübler, Nimr Ghanayim, Thilo Hinterberger: The Thought Translation Device (TTD) for Completely Paralyzed Patients, IEEE Transactions on Rehabilitation Engineering, Vol.8, No.2, June 2000, Online Ressource (12.10.2006): <http://ieeexplore.ieee.org/iel5/86/18398/00847812.pdf>

[BIR99] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kübler, J. Perelmouter, E. Taub, and H. Flor: A spelling device for the paralysed, Nature, vol. 398, pp. 297-298, 1999

[BRA05] G. Bradsky, A. Kaehler, V. Pisarevsky: Learning-Based Computer Vision with Intel's Open Source Computer Vision Library (2005), Online Ressource (12.9.2006): http://www.intel.com/technology/itj/2005/volume09issue02/art03_learning_vision/vol09_art03.pdf

[BUR98] C. J. Burges: A Tutorial on Support Vector Machines for Pattern Recognition, Bell Laboratories, Lucent Technologies, in: Data Mining and Knowledge Discovery 2, 121-167, 1998, Online Ressource (10.11.2006): <http://research.microsoft.com/~cburges/papers/SVMTutorial.pdf>

[CAR06] EKG in Wort und Bild, 2care.at: Freies Pflegelexikon, Online-Ressource (10.9.2006): <http://www.2care.at/Artikel/Grundlagen/EKG>

[COG05] Bates, R., Istance, H., Oosthuizen, L., and Majaranta, P.: Survey of De-Facto Standards in Eye Tracking, Communication by Gaze Interaction (COGAIN), IST-2003-511598, 2005, Online Ressource (13.10.2006): <http://www.cogain.org/results/reports/COGAIN-D2.1.pdf>

[COL01] Coleman, Krista: Electromyography Based Human-Computer-Interface to Induce Movement in Elderly Persons with Movement Impairments, EC/NSF Workshop on Universal Accessibility of Ubiquitous Computing: Providing for the Elderly, 2001, Alcácer do Sal, Portugal

[CON] J. Conrad: Electromagnetic Compatibility of Medical Electrical Equipment: IEC 60601-1-2 Online Ressource (4.10.2006): <http://www.abinee.org.br/informac/arquivos/emc8.pdf>

- [COO04] T.F. Cootes, C.J.Taylor: Statistical Models of Appearance for Computer Vision, Imaging Science and Biomedical Engineering, University of Manchester (2004), Online Ressource (10.10.2006): http://www.isbe.man.ac.uk/~bim/Models/app_models.pdf
- [CUN00] Ulrich Cuntz, Ruth Rauh, Winfried Rief: Biofeedback und Beckenbodenbehandlung von Inkontinenz und Obstipation; in: Winfried Rief, Niels Birbaumer: Biofeedbacktherapie: Grundlagen, Indikation und praktisches Vorgehen, Schattauer, 2000, ISBN 3-7945-1968-X
- [CIL03] R. Cilibrasi: The Neuroserver Software Framework (2003), Online Ressource (4.7.2006): <http://openeeg.sourceforge.net/doc/sw/NeuroServer>
- [DAH05] Bernhard Dahme: Methoden der Psychophysiologie, Vorlesung an der Universität Hamburg (2005), Online-Ressource (1.8.2006): www1.uni-hamburg.de/psych-3/seminar/dahme/ss05/PsPh8.ppt
- [DAV06] J.M Davis, EKG Praktikum Herz / EKG in der vegetativen Physiologie (2006), Online Ressource (13.10.2006): <http://www.lrz-muenchen.de/~jmd/index.htm>
- [DEL06] A. Delorme, T. Fernsler, H. Serby, S Makeig: The EEGLab-Tutorial, University of San Diego, California (2006), Online Ressource (13.7.2006): <http://www.sccn.ucsd.edu/eeglab/eeglabtut.html>
- [DEM05] John N. Demos: Getting Started with Neurofeedback, W.W. Norton & Company, New York, 2005, ISBN 0-393-70450-5
- [DUR04] Durka Piotr J., Zygierewicz Jaroslaw, Klekowicz Hubert, Ginter Josef, Blinowska Katarzyna J.: On the statistical significance of event-related EEG desynchronization and synchronization in the time-frequenye plane. IEEE Transactions and Biomedical Engineering, July 2004, Online Ressource (17.10.2006): <http://eeg.pl/Members/durka/Article.2004-03-12.2521/getFile>
- [EDF92] The European Data Format: EDF (1992); Online Ressource (2.11.2006): <http://www.edfplus.info/>
- [ERL37] Erlanger, Joseph, Gasser, Herbert S. Electrical Signs of Nervous Activity. Eldridge Reeves Johnson Foundation Lectures. University of Pennsylvania Press, Philadelphia. 1937. Second edition, 1968.
- [ERX1797] Erxleben, J. Chr. P.: Anfangsgründe der Naturlehre, 6. Aufl., mit Zusätzen von G.C. Lichtenberg, Göttingen 1794
- [FAR88] L. A. Farwell, E. Donchin (1988). "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potenzials.", *Electroencephalography and Clinical Neurophysiology* 70(6): 510-23.
- [FEH89] L.G. Fehmi, A. Sundor: The Effects of Electrode Placement Upon EEG Biofeedback Training: The Monopolar-Bipolar Controversy; *INTERNATIONAL JOURNAL OF PSYCHOSOMATICS* 36(1-4):23-33,1989
- [FER06] Univ.-Prof. Dr. H.-J. Ferenz: Physiologie der Tier und des Menschen, Martin Luther Universität Halle-Wittenberg (2006), Online Ressource (20.8.2006): <http://www2.biologie.uni-halle.de/zool/physiol/Materialien/Vorlesung%208.PDF>
- [FIN05] Carsten Finke: Okulomotorische Untersuchungen zur Modulation visueller Aufmerksamkeit durch räumliches Arbeitsgedächtnis beim Menschen, Dissertation, Medizinische Fakultät der Charite, Universitätsmedizin Berlin (2005), Online Ressource (8.10.2006): <http://edoc.hu-berlin.de/dissertationen/finke-carsten-2005-12-16/PDF/finke.pdf>
- [FRI98] Stephan Frings und Dieter Grammig: Vorlesungsskripte Zoophysiologie, Zur Struktur und Funktion von Ionenkanälen, November 1998, Online Ressource (6.10.2006): <http://www.physiology.de/kanal/ionoin.htm>,

- [FRI98a] Stephan Frings und Dieter Grammig: Vorlesungsskripte Zoophysiologie, Primärprozesse bei der Muskelkontraktion, November 1998, Online Ressource (6.10.2006): <http://www.sinnphysiologie.de/muskel/muskelin.htm>
- [GAO03] Xiaorong Gao, Dingfeng Xu, Ming Cheng, and Shangkai Gao: A BCI-Based Environmental Controller for the Motion-Disabled, IEEE Transactions on neural systems and rehabilitation engineering, Vol. 11, No. 2, June 2003
- [GIP96] Gips J, Di Mattia P, Curran FX, Olivieri P (1996) Using EagleEyes to help people with special needs, In: Klaus J, Auff E, Kremser W, Zagler W (eds), Interdisciplinary aspects on computers helping people with special needs. R. Oldenbourg, Vienna
- [GOR02] D. O. Gorodnichy, S. Malik and G. Roth. Nouse 'Use Your Nose as a Mouse' - a New Technology for Hands-free Games and Interfaces, Proceedings of International Conference on Vision Interface (VI'2002), pp. 354-361, Calgary, May 27-29, 2002
- [GRI03] Dan Griffiths, Nelo, Jim Peters, Andreas Robinson, Jack Spaar, Yaniv Vilnai, Jörg Hansmann: The Modular EEG Design; Online Ressource (3.6.2006): http://openeeg.sourceforge.net/doc/modeeg/modeeg_design.html
- [GUG99] C. Guger, W. Harkam, C. Hertnaes, and G. Pfurtscheller, "Prosthetic control by an EEG-based brain-computer interface (BCI)," in Proc. AATE 5th European Conference for the Advancement of Assistive Technology. Düsseldorf, Germany, Nov. 3-6, 1999
- [GUG06] Guger Technologies: Produktbeschreibung der Biosignalverstärker gUSB-Amp und gMobi-Lab, Online Ressource (20.8.2006): <http://www.gtec.at>
- [HAN03] Jörg Hansmann: The Modular EEG, Online Ressource (8.7.2006): <http://openeeg.sourceforge.net/doc/modeeg/modeeg.html>
- [HEL06] Ulrich Helmich, Skriptum zur Neurobiologie, 18.7.2006, Online Ressource (22.10.2006): <http://www.drd.de/helmich/bio/neu/reihe1/ur12/ruhepotenzial.html>
- [HIL02] S.A. Hillyard, F. Di Russo, W.A. Teder-Sälejärvi (2002). Steady-state VEP and Attentional Visual Processing. In: A. Zani & A.M. Proverbio (Eds.), The Cognitive Electrophysiology of Mind and Brain, Academic Press, pp. 259-274.
- [HEU00] Jörg Heuser: Biofeedback bei chronischen Rückenschmerzen; in: Winfried Rief, Niels Birbaumer: Biofeedbacktherapie: Grundlagen, Indikation und praktisches Vorgehen, Schattauer, 2000, ISBN 3-7945-1968-X
- [HEU00a] Jörg Heuser, Winfried Rief: Biofeedback bei Spannungskopfschmerz und Migräne; in: Winfried Rief, Niels Birbaumer: Biofeedbacktherapie: Grundlagen, Indikation und praktisches Vorgehen, Schattauer, 2000, ISBN 3-7945-1968-X
- [INS06] Insight Instruments Inc., Biofeedbackgeräte, Online Ressource (6.11.2006): <http://www.insight.co.at>
- [JEO04] Hyuk Jeong, Jong-Sung Kim, Jin-Seong Choi: A study of an EMG-controlled HCI method by clenching teeth, Lecture notes in computer science, ISSN 0302-9743 Asia Pacific conference on computer human interaction No6, Rotorua, 2004 (APCHI 2004)
- [KIM04] Jong-Sung Kim, Hyuk Jeong, Wookho Son: A new means of HCI: EMG-Mouse, International IEEE Conference on Systems, Man and Cybernetics, 2004
- [KEN87] P.R. Kennedy, R.A. Bakey, M.M. Moore, K. Adams, J. Goldwaithe. "Direct control of a computer from the human central nervous system." (1987), IEEE Trans Rehabil Eng. 8 (2):198-202.
- [KRE01] Alois Krenn: Informationen zur Krankenpflege: Die Körpertemperatur Online Ressource (20.9.2006): <http://www.passail.eu/krankenpflege/koerpertemperatur.htm>

[KRO00] R. Kroymann: Biofeedback in der Therapie von Angststörungen; in: Winfried Rief, Niels Birbaumer: Biofeedbacktherapie: Grundlagen, Indikation und praktisches Vorgehen, Schattauer, 2000, ISBN 3-7945-1968-X

[KRO00a] Reiner Kroymann, Horst Zittlau, Gerhard Goebel: Biofeedback in der Therapie des chronischen Tinnitus; in: Winfried Rief, Niels Birbaumer: Biofeedbacktherapie: Grundlagen, Indikation und praktisches Vorgehen, Schattauer, 2000, ISBN 3-7945-1968-X

[KUP05] Tony R. Kuphaldt: All about circuits, E-books Volume I - IV, Online Ressource (17.10.2006): <http://www.allaboutcircuits.com>

[LAU06] Stefan Lautenbacher, Lehrstuhl für physiologische Psychologie, Universität Bamberg, Skriptum Biologische Psychologie II, Online Ressource (2.6.2006): http://www.unibamberg.de/fileadmin/uni/fakultaeten/ppp_professuren/physiologische_psychologie/Biologische_Psychologie_II/M-edA25.01.2006.pdf

[LEI05] Denis Leimberg, Martin Vester-Christensen: Heuristics for Speeding Up Gaze Estimation, Swedish Symposium on Image Analysis (SSBA 2005), Malmö, March 2005

[LI06] Dongheng Li, Derrick Parkhurst: OpenEyes - an open-hardware open-source system for low-cost eye tracking, Journal of Modern Optics, Volume 53, Number 9, Taylor and Francis Ltd., June 2006, pp. 1295-1311

[LEV00] Simon P. Levine, Jane E. Huggins: Direct Brain Interface Project, University of Michigan Online Ressource (3.7.2006): <http://www.engin.umich.edu/dbi/nih2000/dataacq.html>

[LIS05] Lutz Lißbeck: Entwicklung und Aufbau eines Systems zur Erfassung und Echtzeit-Übertragung biometrischer Daten zur Unterstützung neurologischer Therapiemaßnahmen im Einsatz an Kindern beim Aufenthalt im Wasser, Diplomarbeit an der Fakultät für Ingenieurwissenschaften, Fachgebiet Elektromechanische Konstruktion, Universität Duisburg-Essen, 2005, Online Ressource (2.9.2006): <http://www.l-lisseck.de>

[MAG04] John J. Magee, Matthew R. Scott, Benjamin N. Waber and Margrit Betke: EyeKeys: A Real-time Vision Interface Based on Gaze Detection from a Low-grade Video Camera, Computer Science Department, Boston University, IEEE 0-7695-2158-4/04, 2004

[MAH06] Sasha Mahlke, Michael Minge: Emotions and EMG measures of Facial Muscles in Interactive Contexts, Conference on Human Factors in Computing Systems (CHI 2006), Montreal, Quebec, Canada, 2006

[MAK00] S. Makeig, T. Jung, C. Humphries, T. Lee, M. J. McKeown: Removing electroencephalographic artifacts by blind source separation; Psychophysiology, 37 (2000), 163–178, Cambridge University Press

[MAL95] Jaakko Malmivuo, Robert Plonsey: Bioelectromagnetism - Principles and Application of Bioelectric and Biomagnetic Fields, Oxford University Press, 1995

[MAR00] Michael Marwitz: Ein neuer Weg zur Behandlung der essentiellen Hypertonie: Integrative Biofeedback-Therapie; in: Winfried Rief, Niels Birbaumer: Biofeedbacktherapie: Grundlagen, Indikation und praktisches Vorgehen, Schattauer, 2000, ISBN 3-7945-1968-X

[MAT06] Mathworks Inc.: Matlab and the Matlab Engine, Online Ressource (23.5.2006): <http://www.mathworks.com>.

[MUE05] Reiner Münch: The MonolithEEG Amplifier Design (2005), Online Ressource (17.10.2006): <http://www.people.freenet.de/moosec/projekte/simpleeeg>

[MUP05] G. Müller-Putz, R. Scherer, Ch. Neuper., G. Pfurtscheller: EEG-based control of functional electrical stimulation (FES) in tetraplegic patients, Laboratory for Brain-Computer Interfaces, Graz University of Technology, Online Ressource (11.10.2006): http://www.bci-info.tugraz.at/groups/Graz-BCI/Graz-BCI/EEG-based_control_of_functional_electrical_stimulation.pdf

[MÜL00] Friedemann Müller, Ingo Keller, Marguerite Ledches, Jörg Wissel: Biofeedback bei Lähmungen und anderen neurologischen Erkrankungen; in: Winfried Rief, Niels Birbaumer: Biofeedbacktherapie: Grundlagen, Indikation und praktisches Vorgehen, Schattauer, 2000, ISBN 3-7945-1968-X

[NAN00] Alexandra Nanke, Winfried Rief: Biofeedback bei Somatisierungspatienten: Die Brücke zwischen organmedizinischem und psychosomatischem Krankheitsbild; in: Winfried Rief, Niels Birbaumer: Biofeedbacktherapie: Grundlagen, Indikation und praktisches Vorgehen, Schattauer, 2000, ISBN 3-7945-1968-X

[OPC06] The Intel Open Computer Vision Library: Online Ressource (4.10.2006): <http://www.intel.com/technology/computing/opencv>

[OPE04] Jörg Hansmann, Andreas Robinson, Jim Peters, Yaniv Vilnay, Moritz v. Buttlar a.o.: The Open EEG Project, Online Ressource (4.7.2006): <http://openeeg.sf.net>

[ORT05] Pedro Ortega: OpenEEG Active Electrodes Building Guide, Online-Ressource (10.10.2006): <http://www.dcc.uchile.cl/~peortega/ae>

[PET03] J. Peters, T. Fisher: FidView: an interactive Filter Designer, Online Ressource (8.6.2006): <http://uazu.net/fiview>

[PET02] J. Peters: BwView: a wavelet-based EEG viewing tool, Online Ressource (8.6.2006): <http://uazu.net/bwview>

[PFU98] G. Pfurtscheller, 1998, "EEG event-related desynchronization (ERD) and event-related synchronization (ERS)". In: E. Niedermeyer, F. H. Lopes da Silva, (Eds.), Electroencephalography: Basic Principles, Clinical Applications and Related Fields. 4th Edition, Williams and Wilkins, Baltimore, pp. 958-967, 1998

[RIE00] Winfried Rief, Niels Birbaumer: Biofeedbacktherapie: Grundlagen, Indikation und praktisches Vorgehen, Schattauer, 2000, ISBN 3-7945-1968-X

[RIJ94] A. C. Metting Van Rijn, A. Peper, C. A. Grimbergen: Amplifiers for Bioelectric events - A Design with a minimal number of parts (1994), Me. % Biol. Eng. & Comput., 1994, 32, 305-310

[SAM04] A. B. Samardzic: The GNU LIBMatheval Online Ressource (3.5.2006): <http://www.gnu.org/software/libmatheval>

[SAM05] A. B. Samardzic: TEMPO - Topographic EEG Mapping Program, Online Ressource (1.6.2006): <http://tempo.sourceforge.net/>

[SCH04] Gerwin Schalk, Dennis J. McFarland, Thilo Hinterberger, Niels Birbaumer, Jonathan R. Wolpaw: BCI2000: A General-Purpose Brain-Computer Interface (BCI) System, IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 51, NO. 6, JUNE 2004

[SCL02] A. Schloegl: BIOSIG - An Open Source Library for bioelectric data processing (2002), Online Ressource (28,7,2006): <http://www.dpmi.tu-graz.ac.at/schloegl/matlab>, <http://biosig.sourceforge.net>

[SCN00] Gerd Schneider, Joachim Kurt: Technische Prinzipien zur Messung der Augenbewegungen, (2000), Humboldt Universität Berlin, Institut für Rehabilitationswissenschaften, Online Ressource (14.9.2006): <http://www2.hu-berlin.de/reha/eye/Studie2000/tech.pdf>

[SEE06] Dr. Gunnar Seemann: Bioelektrische Signale und Felder, Institut für biomedizinische Technik Universität Karlsruhe (2006), Online Ressource (20.10.2006): <http://www.ibt.uni-karlsruhe.de/signaleundfelder.php>

[SHE05] Richard A. Sherman: The Use of Electromyographic and Temperature Feedback for Treatment of Phantom Limb Pain (2005), Online Ressource (20.9.2006): <http://www.bfe.org/protocol/pro05eng.htm>

[SIT98] R. Sitzmann: Elektromagnetische Verträglichkeit in der Medizintechnik, Siemens AG, Bereich Medizinische Technik, Erlangen, Online Ressource (3.11.2006): http://www.healthcare.siemens.com/medroot/en/news/electro/issues/pdf/heft_2_98_d/08sitzma.pdf

[SMI01] Steven W. Smith: The Scientist and Engineer's Guide to Digital Signal Processing (2001), California Technical Publishing, 1st edition 1997, ISBN 0966017633
Online Ressource (1.10.2006): <http://www.dspguide.com>

[STR00] Ute Strehl, Boris Kotchoubey, Niels Birbaumer: Biofeedback von Hirnaktivität bei epileptischen Anfällen; in: Winfried Rief, Niels Birbaumer: Biofeedbacktherapie: Grundlagen, Indikation und praktisches Vorgehen, Schattauer, 2000, ISBN 3-7945-1968-X

[SUT92] E.E. Sutter, The brain response interface: communication through visually induced electrical brain responses. Journal of Microcomputer Applications, v. 15, pp. 31-45, 1992

[TEI00] Malvin C. Teich, Steven B. Lowen, Bradley M. Jost, and Karin Vibe: Heart Rate Variability: Measures and Models, Department of Electrical and Computer Engineering, Boston University, Online Ressource (14.8.2006): http://arxiv.org/PS_cache/physics/pdf/0008/0008016.pdf

[VOB06] Open Source Lernplattform Vorarlberg, Online Ressource (18.7.2006): <http://www.vobs.at/Bio/physiologie/a-neuro-4.htm>

[VID73] J.J. Vidal, "Toward direct brain-computer communication," Annu. Rev. of Biophys. Bioeng., pp. 157-180, 1973

[WHE03] Kevin R. Wheeler, Charles C. Jorgensen: Gestures as Input: Neuroelectric Joysticks and Keyboards, IEEE Pervasive Computing, 2003, IEEE 1536-1268/03

[WIE03] Peter Wiebe: Biosignale für die Kommunikation bei Mensch-Maschine-Systemen, Dissertation and der Universität Duisburg-Essen, 2003, Online-Ressource (10.10.2006): http://deposit.ddb.de/cgi-bin/dokserv?idn=970509480&dok_var=d1&dok_ext=pdf&filename=970509480.pdf

[WOL91] J. Wolpaw, D. McFarland, G.W. Neat, and C.A. Forneris, "An EEG-based brain-computer interface for cursor control," Electroenceph. Clin. Neurophys., vol. 78, pp. 252-259, 1991

[WOL04] J. Wolpaw, D. McFarland: Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans, PNAS, Vol 101, No 51, Dec. 2004, Online Ressource (5.10.2006): <http://www.pnas.org/cgi/doi/10.1073/pnas.0403504101>

[YAM99] Hiroki Yamaguchi, Daisuke Nishikawa, Wenwei Yu: EMG control switching System for FES, IEEE International Workshop on Robot and Human Interaction, Pisa, Italy, IEEE 1999, 0-7803-5841-4/99

[ZAG04] Wolfgang Zagler: Elektronische Hilfen für behinderte und alte Menschen - Rehabilitationstechnik 1; Arbeitsunterlage für Lehrzwecke an der Technischen Universität Wien, Institut "integriert studieren", Forschungsgruppe für Rehabilitationstechnik, 2004