Bachelor’s Paper

Degree Programm: Biomedical Engineering – Medical and Hospital Engineering

Development of an EOG-device as a sensor-module for the AsTeRICS-project

By: Benedikt Klaus Rossboth
Student Number: 0910227031

Supervisor: Dipl.-Ing. Christoph Veigl

Vienna, 5.12.2011

FACHHOCHSCHULE TECHNIKUM WIEN

Declaration

„I confirm that this paper is entirely my own work. All sources and quotations have been fully acknowledged in the appropriate places with adequate footnotes and citations. Quotations have been properly acknowledged and marked with appropriate punctuation. The works consulted are listed in the bibliography. This paper has not been submitted to another examination panel in the same or a similar form, and has not been published. I declare that the present paper is identical to the version uploaded."

Vienna, 5.12.2011

Signature

Kurzfassung

Diese Arbeit befasst sich mit der Entwicklung eines Messsystems für das elektrische Potential der Augen, einem EOG Sensor. Die ursprüngliche Idee war dieses Signal als Teil eines Musikinstrumentes, zum Auswählen der Note, zu verwenden. Ein solches Instrument würde auch Menschen mit Beeinträchtigungen in den oberen Extremitäten das Musizieren ermöglichen. Jedoch wurde mit Fortschreiten des Projektes mehr und mehr Wert auf die Entwicklung eines universellen EOG-Sensors gelegt. Die Verarbeitung des Signals besteht aus Verstärkung und Filterung der Frequenzen zwischen 0 und 20 Hz. Der EOG Sensor wird außerdem kabellos und mit Batterieversorgung entwickelt, damit keine direkte Verbindung zwischen dem Benutzer und dem Stromnetz besteht.

Außerdem wird dieser EOG Sensor in das AsTeRICS Projekt integriert, welches darauf abzielt, eine Unterstützungsplattform, die die Kommunikation zwischen verschiedenen Sensor- und Aktuatormodulen erleichtert, zu entwickeln.

In dieser Arbeit werden sowohl dieses Projekt als auch die Grundlagen des menschlichen Auges kurz beschrieben. Weiters wird die Verwendung der medizinischen Signalverarbeitung zur Verstärkung und Filterung dieses Signals, die Interrupt-gesteuerte A/D Wandlung und auch die kabellose Kommunikation mit dem PC per Bluetooth detailliert beschrieben. Für diese Übertragung wird das CIM Protokoll, welches vom AsTeRICS Projekt zur Verfügung gestellt wird verwendet.

Ein einkanaliges EOG Sensorsystem mit einer Verstärkung von 68 dB zwischen 0 und 20 Hz wurde auf einem Steckbrett entwickelt. Für die Verstärkung und Filterung wurden aktive Filter verwendet, bestehend aus einem aktiven Hochpass erster Ordnung und einem aktiven butterworth Tiefpass dritter Ordnung im Sallen-Key Layout. Weiters wurde eine driven right leg (DRL) Schaltung hinzugefügt. Die Versorgungsspannung wird von zwei AA-Batterien und einem DC-DC Konverter geliefert. Das System kommuniziert per Bluetooth mit der AsTeRICS Runtime Environment. Außerdem wurden Tests durchgeführt um die Funktion des Systems festzustellen.

Schlagwörter: EOG, HCI, AsTeRICS, ADC, Bluetooth, Medizinische Messtechnik, Assistiertes Leben

Abstract

In this paper a measuring system for the electrical potential of the eyes, an Electro-oculogramm (EOG) system, is devoted. The main idea was to use these electrical signals as part of a musical instrument. Such an instrument would allow people with motor disabilities in their upper limbs to make music by themselves. As the project evolved, nevertheless, this idea was put in the background, but the development of a universal EOG-system became the main goal. The processing of the eyes signal is done by amplifying and filtering out the relevant band of frequencies between 0 and 20 Hz. The EOG-system is realized wireless and with battery supply, so no connection between the user and the power supply system is established and therefore the user's safety is guaranteed. Furthermore, this measuring system is implemented in the project AsTeRICS, which aims for a support platform that will facilitate and improve interaction possibilities for the people mentioned above. An introduction to the AsTeRICS project and to the basics of the human eye, is included in this paper. Furthermore, the use of medical instrumentation to amplify and filter the eyes signal is explained in detail, as well as the microcontroller firmware for the AD conversion and the communication via Bluetooth, using the CIM communication protocol provided by the AsTeRICS project.

A single-channel EOG-sensor system with a gain of 68 dB between the frequency range of 0 and 20 Hz is developed on breadboard. For both, amplifying and filtering, active filters are used, including one active high-pass filter first order and one active butterworth low-pass filter third order in Sallen-Key layout. Furthermore, a right leg driven circuit was realized. The supply voltage is provided by 2 AA batteries and a DC-DC converter. The system is communicating via Bluetooth with the AsTeRICS Runtime Environment. Tests were performed to validate the systems proper working on visualizing the eyes movement.

Keywords: EOG, HCI, AsTeRICS, ADC, Bluetooth, Medical Instrumentation, Ambient Assisted Living

Acknowledgments

At first I want to thank my parents for their financial support, which allows me to take part at this course of study on the University of Applied Sciences Technikum Vienna. Of course, besides of that financial support, the motivated me a lot, by talking about issues mentioned in this paper and by strengthening my mind after undergone failures during the development.

Furthermore, I want to thank my supervisor Dipl.-Ing. Christoph Veigl, who was reachable, as good as always, and who throughout the work tried to help, wherever he could, and wherever I needed help.

Table of contents

1 Introduction

During my community service in a center for handicapped children and adolescents run by the “Gesellschaft für ganzheitliche Förderung und Therapie GmbH” (1) I had experience of working with people with physical and psychic disabilities. As well as the weekly therapeutic program the children undergo, music-therapeutic sessions are held. In these sessions in particular there had always been some children who were not able to contribute in any way, but singing while most of the others were at least able to play the drums, the xylophone or even some notes on the piano. As a result, my colleague Dominik Koller (2) and I had the idea to develop a musical instrument for people with physical disabilities, especially in the upper limbs.

The first plans included a system, measuring the eye-position through the electrical potential by the eye (Electro-oculogramm – EOG), to choose the note by moving your eyes to the left or the right hand side and to trigger the note by blowing through a blow-sensor.

As the blowpipe already exists and the software part was done by my colleague (2), I concentrated on the work for the hardware for “choosing the note”, the EOG-sensor as Human Computer Interface, and furthermore the integration to the AsTeRICS project, which aims at delivering different sensors and actuators, which can easily be adapted to the individual needs of a handicapped person.

With the proceedings in the research in this topic, more and more not only the use as a musical instrument stood in the middle of this thesis’ aim, but developing a simple sensor-tool to help people with physical disabilities to live their daily lives and execute simple tasks like writing on a virtual keyboard.

Furthermore the separation of those two parts, the hard- and software, made it easier to use the single parts for other purposes in the AsTeRICS project. AsTeRICS (Assisted Technology Rapid Integration Construction Set) “is a flexible and affordable construction set for building assistive functionalities which can be highly adapted to individual user’s needs.” (3) The implementation in this project also was the reason to develop an EOG-sensor measuring not only the horizontal axis but also the vertical. More on the AsTeRICS project will be explained in 2.1 The AsTeRICS Project.

1.1 System concept overview

The development of a HCI (Human Computer Interface) mainly contains of three parts which are:

1. The analog electronics
2. The analog-digital conversion
3. The communication with the computer

For the analog signal processing path there are a lot different approaches depending on one’s budget, the demanded accuracy and also the chosen type of electrical supply. Some of

the differences can be seen in comparing the analog block diagram shown in (4) with the analog circuit in (5).
The next part deals with the problem of digitalizing the amplified and filtered body signal. This is done by the use of a micro-processor, which can be programmed to execute this task. This part describes the different registers of the used microcontroller (µC), as well as interrupts, USART and the CIM-Protocol, provided by the AsTeRICS project. Having converted the analog signal, the next step is to send it to the computer. There are also different ways of doing that such as the well known USB or Bluetooth. The choice of the communication is mainly about the patient's safety and about the non-use of wires. In Figure 1 a block diagram of the whole developed system can be seen.

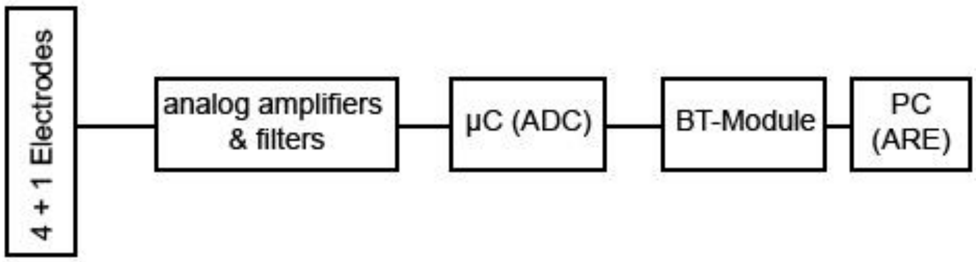


Figure 1. Overall Block diagram, showing the electrodes, amplifiers & filters, µC for ADC, Bluetooth-Module and the ARE running on PC

1.2 State of the art for eye-tracking devices

At the moment there are different approaches towards eye-tracking HCIs, such as wearable EOG-Goggles (6) developed at ETH Zurich. Those goggles can be seen in Figure 2. Furthermore, one of the leading projects in this area working also with EOG-signals is the EagleEyes project by the Boston College (7). This project makes a free mouse replacement available for qualified users from the Opportunity Foundation (8), (9) and it is the only one of the commercial HCIs working with EOG. Nevertheless, there are other eye-trackers available, most of them working with infra-red cameras and dark pupil recognition, like the Intelligaze IG-30 by alea technologies (10) or the EyeTech TM3 by eye tech digital systems. (11) These systems sale for between $4,995 and $14,280 (8) which, in most cases, is too expensive for the users in need. In Figure 3 the EyeTech TM3 tracker is shown and furthermore, in Figure 4 the combination of this tracker with the communication aid Tellus-3 (12) can be seen.

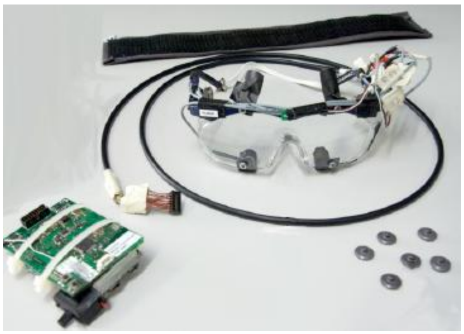


Figure 2. EOG Goggles developed by the ETH Zurich (6)

Figure 3. EyeTech TM3 - infra-red based eye-tracker (13)

Figure 4. EyeTech TM3 + Tellus-3 communication aid (12)

2 Basic concepts

In 2 Basic Concepts, the concepts for the understanding of the idea of this paper are explained. Chapter 2.1 The AsTeRICS project will be an introduction to the AsTeRICS project and furthermore, chapter 2.2 The Human Eye, its physiology and the EOG will give an overview of the human eye.

2.1 The AsTeRICS project

The AsTeRICS project is a Europe-wide project which mainly aims “to develop a support platform that will facilitate and improve communication resources of people with motor disabilities in their upper limbs.” (3) Assistive Technology Rapid Integration and Construction Set, in short AsTeRICS, deals with the topic of Ambient Assisted Living (AAL) and in difference to other AAL solutions its aim is to offer a very flexible and nevertheless affordable system. So the individual users with all their needs and skills take the center stage.

To reach this aim, the project consists of different modules and platforms, which are:

- Sensor Modules
- Communication Modules
- Actuator Modules
- AsTeRICS personal platform

- Configuration Suite
- Environment and Services

The sensor modules are on the user-side and can be every different kind of getting information from the user. This can range from an accelerometer to notice if the user stumbles to the measurement of body-signals, like EMG or EOG. The measured values than are passed on through the communication modules, which can, for example, be USB or Bluetooth. For the easy integration of sensors and individual adaptations, the CIM-protocol was developed. Via those communication modules actuators, mostly implemented as software plug-ins, are controlled, which then directly collaborate with the user's environment, such as the mobile phone or the personal computer.

Furthermore, the different sensor-, communication- and actuator modules can be fit together in various ways so they satisfy the user's needs best. This adjustment is done by the configuration suite and the result is the AsTeRICS personal platform, which is an embedded computing system adapted to the individual user. Figure 5. Project Concept of AsTeRICS showing the connections between sensor, communication and actuator modules. Also the AsTeRICS Personal Platform and the Configuration Suite can be seen.Figure 5 shows the basic concept of the project.

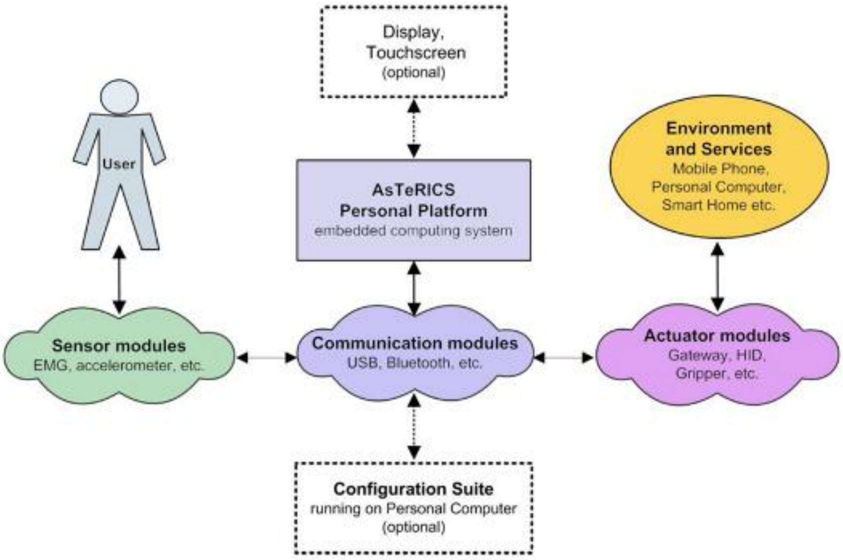


Figure 5. Project Concept of AsTeRICS showing the connections between sensor, communication and actuator modules. Also the AsTeRICS Personal Platform and the Configuration Suite can be seen.

2.2 The human eye, its physiology and the EOG

As one of the sensory organs the human eye is a very important one in order to get in contact with a person's environment. The main function of course is to perceive one's environment visually. But as can be seen in the following chapters, the eyes can not only be used to see, but also to carry out some of everyday actions. In this chapter, a very short introduction to the eyes anatomical and physiological functions is given to be able to know the basics of an EOG.

2.2.1 The anatomy and physiology of the human eye

The eyeball – bulbus oculi – is, as the name says, an organ similar to a ball with a diameter of about 24 mm and a weight of about 7.5 g (14). The musculature of the eye allows a movement of the eyeball in three degrees of freedom (15) caused by six muscles which can be seen in Figure 6.

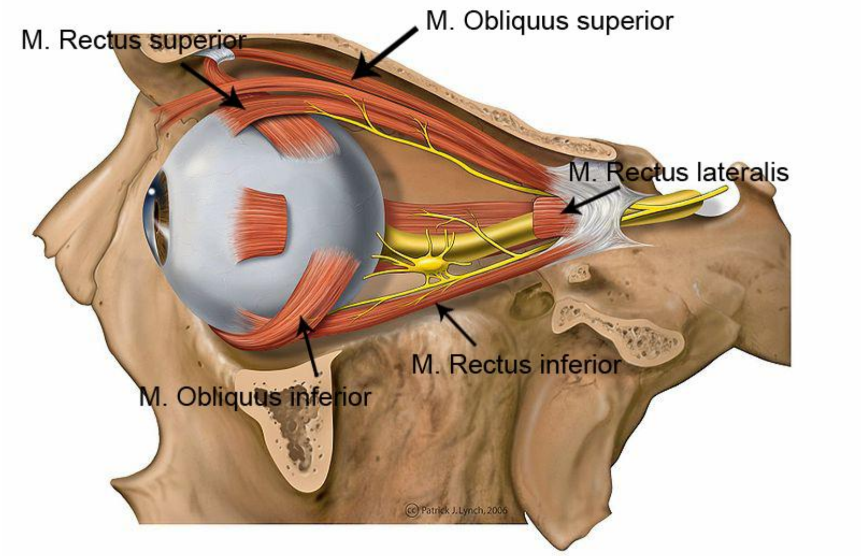


Figure 6. Lateral Eye and orbit anatomy with nerves (16)

In particular these six muscles and their functions are:

- *M. Rectus medialis*
  This muscle is the strongest of these six. It is responsible for the eye movement to the inner side towards the nose.
- *M. Rectus lateralis*
  Musculus Rectus lateralis is the antagonist of M. Rectus medialis. It assesses on the outer side of the eyeball and moves the eye towards the external side.
- *M. Rectus superior*
  M. Rectus superior is located on the top of the eyeball. This muscle is responsible for the elevation of the eye.
- *M. Rectus Inferior*
  Musculus Rectus Inferior is the antagonist of the M. Rectus Superior. Its task is the depression of the eye.

- *M. Obliquus superior*

  This muscle supports the M. Rectus inferior in the depression of the eye. Furthermore, the roll-movement of the eye towards the inner direction is done by it.

- *M. Obliquus inferior*

  M. Obliquus inferior supports the M. Rectus superior in elevating the eye. Also the roll-movement towards the outer side is performed by this muscle.

Furthermore, for the use of the EOG the binocular visual field is important. It is the total range a person can perceive by just moving the eyes. The binocular visual field, which means the visual field of both eyes together, of the human eyes is about +-100° (17), as seen in Figure 7, of which according to (18) up to +-80° can be measured.

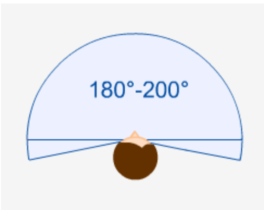


Figure 7. The Horizontal Binocular Visual Field showing a range of 180 - 200° (19)

The visual image starts to be created in the retina, which is located on the backside of the eyeball, in the way of converting the electromagnetic waves into membrane potentials. The optical signals are received by the photoreceptor cells, the cones, for colours, and the rods, for black/white and less intense light. (20)

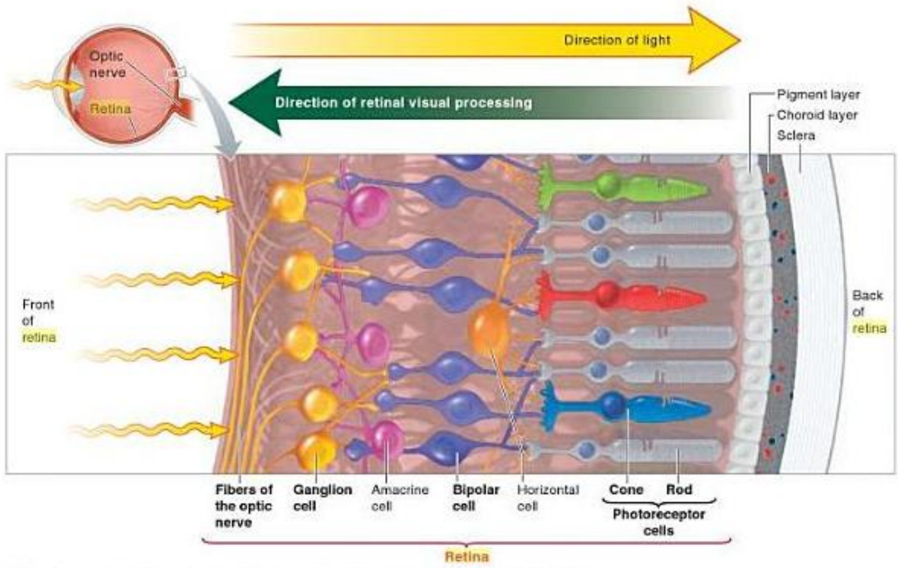


Figure 8. Parts of the Retina, showing the mentioned cones and reds, and further cells involved in the transport of visual signals (21)

Figure 8 shows the parts of the retina, and all the different cells involved in the processing of a visual incoming signal. Then, the action potentials are transported through the nervus opticus, which leads through the lamina cribosa from the nasal cavity into the cranial cavity. Furthermore, half of the nerve fibres cross at the chiasma opticum to the tractus opticus of the other side, which enables the best possible three-dimensional vision. The information of the optical signals then reaches the diencephalon and furthermore, the visual cortex, where the information finally is perceived.

2.2.2 The charge of the eye

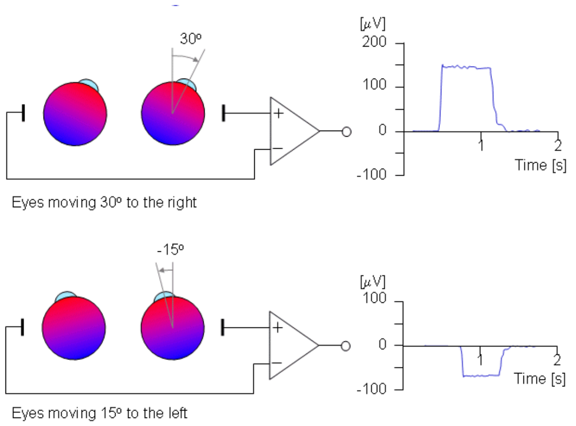


Figure 9. The Magnitude of a horizontal EOG (22)

In 1848 Emil du Bois-Reymond discovered that the eye-ball is a dipole (23). The cornea in the front is positively charged to the back of the eye, the retina. It first was thought of being induced by the action potentials occurring in the retina, but as the dipole charge of the eye continued to exist with closed eyes and therefore no action potentials at all, it was described as the resting potential of the eye. Those resting potentials, induced by the movement of the eye respectively the dipole can be deduced with electrodes. Figure 9 illustrates the derived electrical charge with two horizontal placed electrodes (22).

Regarding (24) the linearity of the measurement decreases for angles wider than 30°. The typical magnitude of this voltage is at about 5-20 µV/°.

3 System development

3.1 The analog electronics

This chapter deals with the analog signal path until it reaches the µC. The ideas of the circuit are explained and supported by some simulations. The all-in-all task of the analog circuit is to amplify and filter the eyes bio-signal and provide it in a voltage range between 0 and 5 V, so the µC can process the signal further on and do the A/D conversion. In Figure 10 the block diagram of the analog circuit is shown.

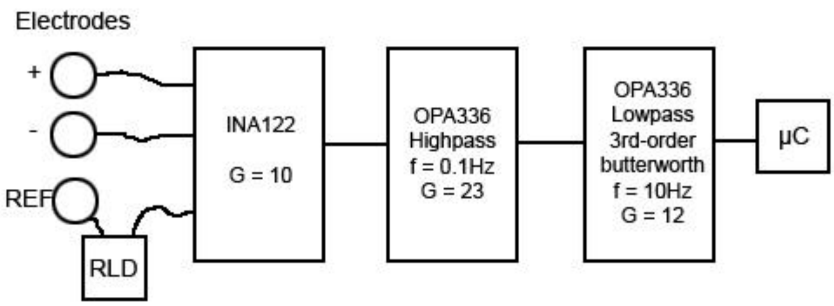


Figure 10. Block diagram of the analog circuit of one channel, showing the electrodes, the instrumentational amplifier, the filters and the µC

3.1.1 The electrodes and the wiring

Due to the easy availability and cheapness of adhesive electrodes for one-time ECG-use, it was decided to use the AgCl ECG electrodes by Skintact (25).
As connection between those electrodes and the analog circuit for the first tests, common crocodile clips were used.

3.1.2 The voltage supply

The first idea regarding the voltage supply was to decouple the whole sensor of the power supply system to a) ensure the patients safety and b) remove all the bindings due to any wires. For this reason, the voltage supply was realized by simple AA-batteries.

Furthermore, the chosen instrumentation amplifier, operation amplifiers and µC require a supply voltage of +5 V. For this reason, a voltage regulator and stabilizer is used. An easy way to realize this, was the RN-Batt DC-DC converter (26), which is offered by robotikhardware.de. This is a circuit based on the LT1302 DC/DC converter, which transforms voltages between 2 and 8 V to a constant voltage of 5 V. The circuit of this converter can be seen in Figure 11 and the values for the particular components in Table 1.

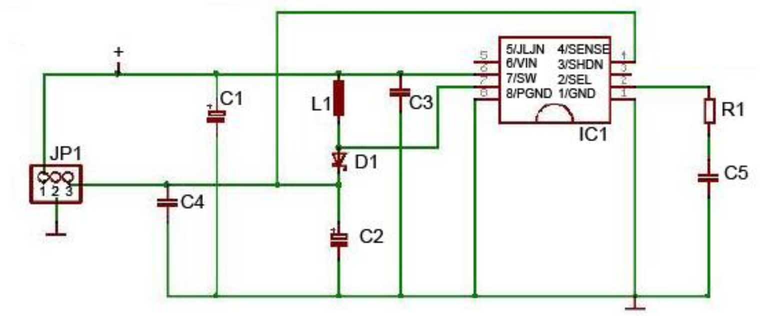


Figure 11. DC converting circuit, based on LT1302 (IC1), with three connectors (JP1) to respectively V_{IN}, GND and +5 V

Component	Value	Device
C1	100 µF	
C2	100 µF	
C3	100 nF	
C4	100 – 220pF	
C5	10 nF	
D1	MBRS240	Schottky Diode
IC1	LT1302	Step-up DC/DC converter
L1	10 µH	
R1	20 kΩ	
JP1		3-pole header

Table 1. Values and components for Figure 12.

Furthermore, for the application of the two active filters with single supply a virtual ground has to be added. This part of the analog circuit was realized by a buffered resistor divider, based on the operational amplifier uA741 (27). The circuit and values used for this purpose can be seen in Figure 12 respectively Table 2 (28).

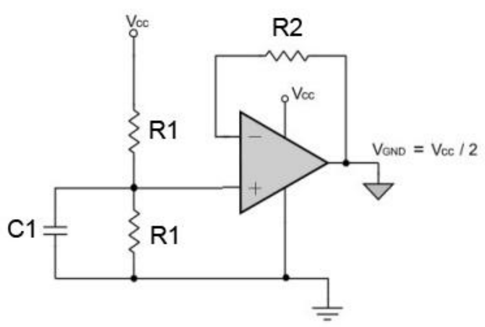


Figure 12. Circuit delivering a virtual ground at 2.5 V, by a voltage divider

Component	Value	Device
C1	0.1 µF	
R1	220 kΩ	
R2	110 kΩ	
	UA 741	General Purpose Single OpAmp

Table 2. Values and Components for Figure 12

3.1.3 The instrumentation amplifier

The INA122 by Texas Instrument (29) suits the requirements for the first stage of amplification perfectly. The low quiescent current of 60 µA makes the INA122 a good choice for portable and battery operated systems. Furthermore, a low offset voltage, offset drift and noise are very appropriate for physiological amplifiers like this EOG.

The task of this first amplification stage is to do the differential amplification between the two electrodes used. As can be found in the datasheet (29), the gain of this stage is calculated by:

$$G = 5 + \frac{200k}{R_G}, with\ R_G = 40k\ \rightarrow G = 10$$

Furthermore, the reference voltage of 2.5 V was realized by a voltage divider with $R_{1/2}$ = 1kΩ.

3.1.4 Active high-pass

The active high-pass filter and the low-pass filter described in 3.1.5 Active Low-pass, are both realized with the OPA2336 by Texas Instruments (30). This two-channel operation amplifier is optimized for uses in battery powered systems, with a quiescent current of 20 µA per channel and furthermore, the low offset voltage qualifies it for medical instrumentations.

For the elimination of the DC-drift due to the electrodes an active high-pass filter was installed. The cut off frequency was calculated at 100 mHz. There is also another step of amplification of 25 included in this active high-pass.

A first order active high pass was realized with the following circuit:

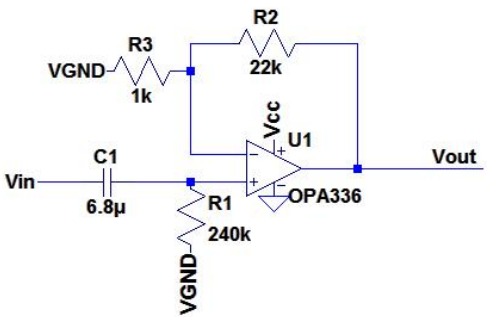


Figure 13. Implemented 1st-order Active Highpass Filter

The values for the particular components were calculated by comparison of the normalized transfer function of a high pass filter with the one for the active high pass filter first order.

$$F(j\omega) = \frac{K\frac{j\omega}{\omega_0}}{1 + j\frac{\omega}{\omega_0}} \leftrightarrow \frac{V_o}{V_i} = \frac{\left(1 + \frac{R_2}{R_3}\right) j\omega R_1 C_1}{1 + j\omega_0 R_1 C_1}$$

1. $\boldsymbol{K = (1 + \frac{R_2}{R_1})}$

So for the desired gain of 25 the values for R_1 and R_2 are:

$$R_1 = 1\ k\Omega$$
$$R_2 = 24\ k\Omega$$

Due to availability in the laboratory for R_2 a value of 22 kΩ was chosen, which reduced the gain of this stage to 23.

2. $\boldsymbol{\frac{1}{\omega_0} = R_1 C_1}$

The desired cut-off frequency of this high-pass filter is 100 mHz and the capacitor was chosen to 6.8 µF.

This leads to a value of $R_1 = \frac{1}{\omega_0 C_1} = 234\ k\Omega$. Again due to availability a 240 kΩ resistor was used, so the cut-off frequency changed to $97.5\ mHz$.

The realized active high-pass filter results in the following amplitude plot simulated with LTSpice IV (31). Small variations to the amplification can exist due to the non-availability of a Spice model for the OPA336 and so a standard model of the operational amplifier LT1001 was used.

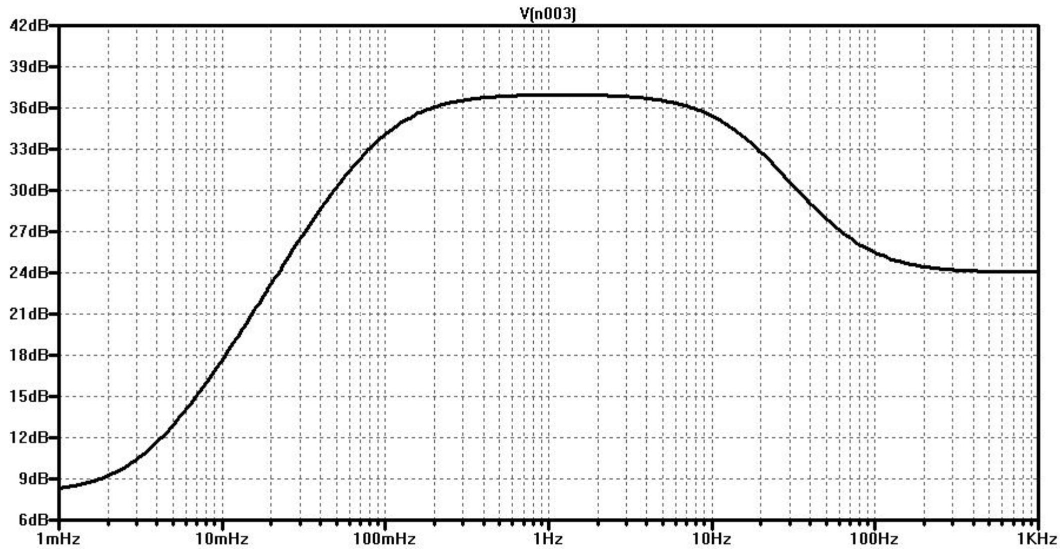


Figure 14. Amplitude attenuation after the active highpass filter

3.1.5 Active Low-pass

The final stage of the analog signal pathway is an active low-pass filter for filtering out the higher frequencies, which are not relevant for the use of this EOG-sensor. As in 2.2 The human eye, its physiology and the EOG already described, the relevant bio-signal of the eye is between 0 and 20 Hz (15) and therefore the cut off frequency was set to 10 Hz. As in both previous stages, there is also an amplification done at this stage.

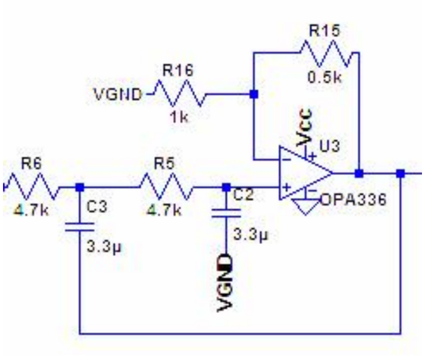


Figure 15. Stage 1 of Active Low-pass Filter

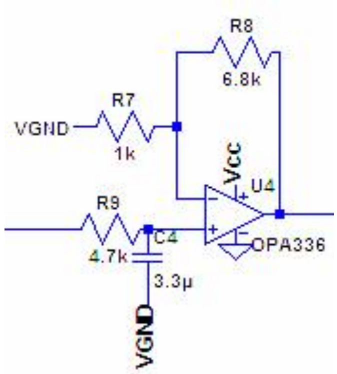


Figure 16. Stage 2 of the Active Low-pass Filter

As can be seen in the two figures above, the low-pass filter was divided into two stages, one second order filter with a gain of 1.5 and one first order filter with a gain of 7.8 V/V. This ends up in an overall amplification of 11.7 V/V. For the calculation, the normalized low-pass filter transfer function again was compared with the particular ones of the implemented Sallen-Key low-pass ending in:

Stage 1:

$$F(j\omega) = \frac{K}{1 + 2aj\frac{\omega}{\omega_0} + (j\frac{\omega}{\omega_0})^2} \leftrightarrow \frac{V_o}{V_i} = \frac{1 + \frac{R_{15}}{R_{16}}}{1 + \left(3 - \left(1 + \frac{R_{15}}{R_{16}}\right)\right) j\omega_0 R_{5/6} C_{2/3} + (jR_{5/6}\omega_0 C_{2/3})^2}$$

1. $\boldsymbol{a = \frac{3-K}{2}}$**, with a = 0.707 (defined by the design structure of a butterworth filter)**

$$K = 1.586$$

2. $\boldsymbol{\omega_c = \frac{1}{R_{5/6}C_{2/3}} = 20\,\pi}$

The cut-off frequency is set to 10 Hz, and the capacitors are again chosen to 3.3 µF.

$$R_{5/6} = \frac{1}{20\pi C_{2/3}} = 4.8k\Omega$$

3. $\boldsymbol{K = \left(1 + \frac{R_{15}}{R_{16}}\right) = 1.586}$ **, (defined by 1.)**

$$0.586\, R_{16} = R_{15}$$
$$R_{16} = 1\, k\Omega$$
$$R_{15} \approx 586\, \Omega$$

Stage 2:

$$F(j\omega) = \frac{K}{1 + j\frac{\omega}{\omega_0}} \leftrightarrow \frac{V_o}{V_i} = \frac{1 + \frac{R_8}{R_7}}{1 + j\omega_0 R_3 C_4}$$

1. **$K = (1 + \frac{R_8}{R_7})$, see chapter 3.1.4**

   To reach the desired amplification of the whole system of 2500 - 3000, R_8 was chosen to 6.8 kΩ and R_7 set to 1 kΩ, which equals an amplification of this stage of 7.8.

2. **$\frac{1}{\omega_0} = R_3 C_4$**

   The desired cut-off frequency of this low-pass filter is 20 Hz and C was chosen to 3.3 µF. Therefore the resistor R was calculated to 4.8 kΩ, as can be seen in Stage 1.

The amplitude over the frequency after this low pass filter can be seen in Figure 17.

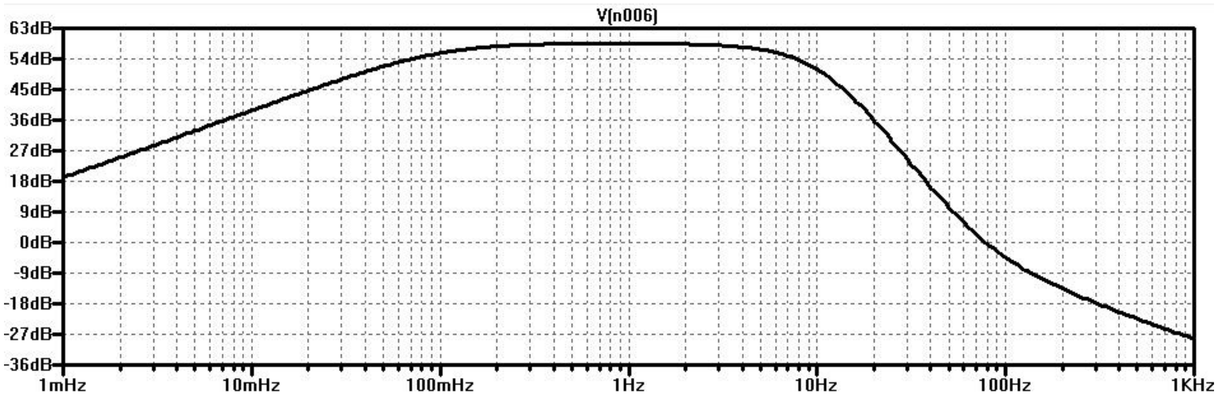


Figure 17. Amplitude attenuation after active low- and high-pass filtering

3.1.6 Minimizing the noise

The human body acts as a big antenna and in this way captures electromagnetic interference. These interferences are mainly caused by the 50 Hz hum produced by the electrical power lines.

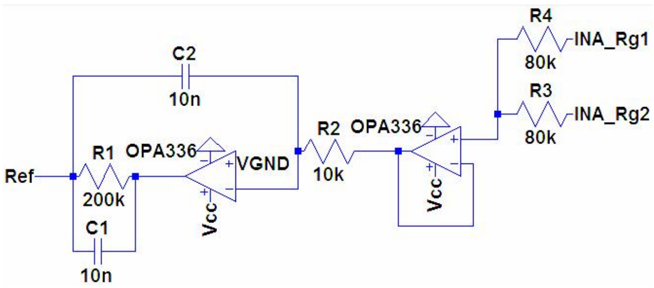


Figure 18. RLD circuit for one channel, leading from reference electrode to the INA122

A first reduction of this signal is done by the active low pass filtering in 3.1.5 Active Low-pass. In order to reduce the 50Hz hum even more, a driven right leg circuit (DRL) was added. Therefore, the proposed circuit by the openEEG (32) project was used. The idea of this circuit is to minimize the common-mode signal. This is done by sending the common mode signal back in the patient and so the patient is driven to the amplifier common voltage. In this way the voltage difference between patient and amplifier common is reduced (33).

This circuit can be seen in Figure 18. Furthermore, the dividing of R_G for setting the gain of the INA in two 80kΩ in parallel can be seen. This is necessary to get the whole amplifier common to the RLD circuit.

3.2 The µC computing ADC and usage of CIM

3.2.1 Teensy++ 2.0

The Teensy++ 2.0 is a complete USB-based microcontroller development system by PJRC (34). This board is based on the AT90USB1286 (35) microcontroller by Atmel with a clock frequency of 16 MHz. It has 8 analog input pins, can be programmed in C per USB and also makes use of an external voltage source of 5 V.

A picture of the Teensy++ 2.0 USB Development Board is shown in Figure 19.

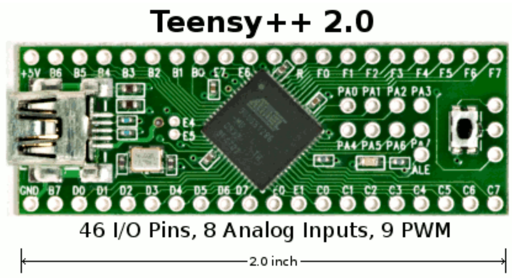


Figure 19. Teensy++ 2.0 USB Development Board by PJRC (34)

For testing the µC Code a TTL-232R cable by FTDI Chip (36) was used. Furthermore, to receive and visualize the data, the freeware HTerm (37) was used.

3.2.2 µC firmware

An interrupt driven AD converting software was implemented on the µC which, furthermore, provides the communication with the AsTeRICS Runtime Environment (ARE) using the CIM-protocol.

CIM-protocol (38)

For an easy adaptation of the different AsTeRICS sensors (CIMs) with the ARE a communication protocol, the CIM protocol, was determined. To fit the developed EOG-sensor into the AsTeRICS project the CIM protocol was implemented on the Teensy++ 2.0.
The CIM protocol consists of a minimum of 11 byte with additional 2048 bytes of data.
In this protocol ARE runs as master and the CIM acts as slave. So the µC will only send EOG-data, if it is once asked by ARE. The format of a data packet can be seen in the following table.

Packet ID	2 bytes	"@T" (0x4054)
ARE ID or *CIM ID*	2 bytes	
Data Size	2 bytes	0x0000 – 0x0800
Packet Serial Number	1 byte	0x00 – 0x7f or 0x80 – 0xff (replies from CIM)
CIM Feature address	2 bytes	
Request or *Reply Code*	2 bytes	
Optional data	0 – 2048 bytes	
Optional CRC checksum	0 or 4 bytes	CRC32 checksum

Table 3. Components of a CIM data packet - data sent from the CIM to ARE is italic, explanation to the table see text below.

The Packet ID identifies the beginning of a data packet. The ARE ID equals the software version of the ARE. Each implemented CIM gets its own ID, consisting of 2 bytes, e.g. 0xA101 for the EOG CIM. The data size specifies whether data is sent, and if so, how many bytes the data consists of. It furthermore defines the size of the total CIM data packet. The maximum size for the optional data is given by 0x0800 (= 2048 bytes).
Each packet sent from ARE to CIM is identified with a serial number reaching from 0x00 to 0x7f (0-127). The answering packet sent by CIM has got the same serial number, so it is easy to observe, which packet has been answered. If CIM sends a packet which is not requested by the ARE, the serial number will differ and start therefore at 0x80 (128).
For each CIM there are features defined which specify the different tasks the CIM can handle.

For the implementation of the CIM to the EOG-Sensor, three basic features were defined:

CIM-ID	Feature-address	Access	Descritption	Data
0xa101: EOG version 1 sensor/ actuator	0x0000	read	Unique serial number	4 bytes
	0x0001	write	Activate Periodic Value Reports	2 bytes: bytes 0,1: period time 0 (off) to 200 milliseconds
	0x0002	read	Channel Value Report	4 bytes: 2 channels of ADC values Byte 1: chn1 low byte Byte 2: chn1 high byte Byte 3: chn2 low byte Byte 4: chn2 high byte

Table 4. Feature List of EOG-sensor showing the three different features of the EOG-CIM. 1) Read the uniqe serial number 2) Activate Periodic Value Reports and 3) Channel Value Report

The Request Code sent from ARE to CIM represents in the MSB the transmission mode (0 = CRC checksum disabled) and in the LSB the Request Code, like request feature list (0x00) or request write feature (0x10), shown in Table 5. The Reply Code sent in the opposite direction consists of an error or status code in the MSB and the reply code, like reply feature list (0x00) or reply write feature (0x10), in the LSB.

Request / ***Reply*** **code**	Direction	Description
0x00	ARE→CIM	request feature list
0x00	*CIM→ARE*	*reply feature list*
0x10	ARE→CIM	request write feature
0x10	*CIM→ARE*	*reply write feature*
0x11	ARE→CIM	request read feature
0x11	*CIM→ARE*	*reply read feature*
0x20	*CIM→ARE*	*event reply*
0x80	ARE→CIM	request reset CIM
0x80	*CIM→ARE*	*reply reset CIM*
0x81	ARE→CIM	request start CIM
0x81	*CIM→ARE*	*reply start CIM*
0x82	ARE→CIM	request stop CIM
0x82	*CIM→ARE*	*reply stop CIM*

Table 5. Request and Reply Codes with their direction and description

In the optional data bytes of the packet, the data of the in “Data Size” predefined size is stored. On request, also a CRC32 checksum can be evaluated. This increases the data packet size for 4 bytes.
The request code is checked first, so in case the different features shall be addressed, it is needed to select the request read/write feature in the request code first. If not, the request code equals 0x00 and the feature list is requested.
So if the request code is set properly, by sending the feature address 0x0000 from ARE to CIM, the unique serial number of the sensor is responded. By sending 0x0001, the sent value equals the period time in milliseconds and furthermore, the channel value report feature is initialized.

The overall µC-firmware

After having explained the tasks, the µC firmware performs, this subchapter describes the most important parts of the software, which are:

- Initialization of UART
  At first the UART connection has to be initialized. Receiving and transmitting on the µC is enabled, as well as the reception interrupt. Furthermore, the baud-rate is set to 115200.

- CIM protocol parser
  After enabling the receive data interrupt, the µC is waiting for any packets to arrive from the ARE. When a packet is received, it is stored into a ring buffer. As soon as the packet ends, the function parse_CIM_protocol() is executed. In this method, the received packet is divided byte by byte and stored into the respective elements of a structure variable.

- Protocol processor
  The protocol processor then looks for the different request codes received from ARE and the respective tasks to do. If, for example, the request code equals the one of “request write feature” (0x0010), the CIM-feature address is furthermore checked. Above, the feature address for “activate periodic values” has already been mentioned, therefore, if the feature address equals 0x0001, a timer and the ADC are started.

- Timer, ADC and UART
  The 16-bit Timer of the µC is then initialized with a preload value, derived by the data sent with the ARE packet. Hence, 0x1000 would equal a periodic update time of 4096 ms. On the overflow interrupt on this timer, the ADC for both channels is started and finally the whole packet is fitted together and sent back to ARE by replayValues().

The above mentioned parts of the code of the µC firmware can be seen in 5.5 Appendix at the end of this document.

3.2.3 AsTeRICS EOG plug-in

To include the EOG in the AsTeRICS project a plug-in had to be implemented. This was done by the AsTeRICS plug-in Creation Wizard. Furthermore, the handlePacketReceived() method was implemented to process different data-packets received from the µC, as the unique serial number or the converted values. Also the further processing of the received values is done by the plug-in within the method handleEogInputValuePacket(). These two methods mentioned can be found in 5.5 Appendix.

3.3 The Bluetooth communication

There are different ways to ensure a wireless connection between the µC and a PC, such as ZigBee (39), UltraWideband and Bluetooth. For an easy implementation a Bluetooth module was installed and configured to connect the µC with the PC. Furthermore, by enabling a wireless communication the patients decoupling of the power supply system was realized. In 3.3.1 Bluetooth and 3.3.2 BTM-222 a short introduction to Bluetooth and the used module is given. A more detailed description would go beyond the scope of this work.

3.3.1 Bluetooth

In 1998 the five companies Ericsson, Nokia, IBM, Toshiba and Intel founded the Bluetooth Special Interest Group (SIG) (40), which, since then, has been working on the Bluetooth Standards and growing fast in number of participating companies. Up to now over 15.000 companies have contributed to this interest group (41). The latest specification, Bluetooth 4.0, was released in 2009 and in summer 2011 the first hardware including Bluetooth 4.0 was produced. SIG developed the industrial standard regarding IEEE802.15.1 for Wireless Personal Area Network (WPAN) (42).

Due to their maximum permitted power, Bluetooth devices can be divided into three different classes:

	Max. Permitted power [dBm]	**Range [m]**
Class 1	20	~100
Class 2	4	~10
Class 3	0	~5

Table 6. Different Bluetooth Classes

The frequency band of the Bluetooth communication is between 2.402 and 2.480 GHz. For an increased stability frequency hopping is used, which means the frequency band is divided into 79 1 MHz steps and the communication is changed between those up to 1600 times a second (43).
In this paper, the Bluetooth module is used to provide a connection between the EOG-system and the PC. Therefore, the PC is acting as master and the Bluetooth module has to be configured as slave.

3.3.2 BTM-222

Due to availability, the Bluetooth Module BTM-222 by Rayson (45) was used. The processing of this module was done by Peter Plischka (44), who provided a basic board for this module, offering pins for Input, Output, Supply and Ground. The BTM-222 is a class 1 module based on Bluetooth 2.0 + EDR certification.

Figure 20. BTM-222 based Bluetooth Module designed and provided by Peter Plischka (44)

The Bluetooth 2.0 + EDR certification was released in November 2004, and due to the Enhanced Data Rate (EDR), an increased bit-rate, up to 2.1 MBit/s, was enabled. The module had to be reconfigured to a baud-rate of 115200. This was done by sending AT commands via the TTL-232R cable by FTDI Chip (36) and the terminal freeware PuTTY (46).
Following commands were sent to configure the module for the desired use:

- ATR1 – sets the module as slave, ARE as master
- ATL5 – sets the baud-rate to 115200, after this command a reconnection at this baud-rate is necessary
- ATE0 – disables echoing to the host
- ATQ1 – disables responses to other commands

3.4 Tests during the development

In this chapter the different tests which have been taken during the development to verify the proper working of each parts will be illustrated.

3.4.1 Development and testing of the EOG system

In Figure 21, the breadboard development of the horizontal EOG-channel, including the power supply and the Teensy++ 2.0 development board, can be seen.

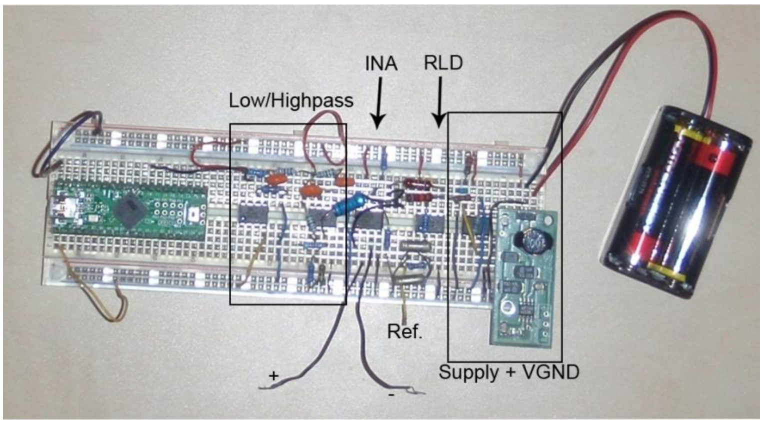


Figure 21. Breadboard Development Result of one EOG channel, showing all components of the analog circuit, including the DC/DC converter, 2xAA batteries and the Teensy. +, - and Ref. mark the connections to the electrodes.

Furthermore, a first recording of the horizontal EOG, analog-digital converted by the Teensy++ 2.0 board, and visualized by MatLAB is shown in Figure 22. This data was recorded with a sampling frequency of 256Hz and the mean of each ten samples was visualized.

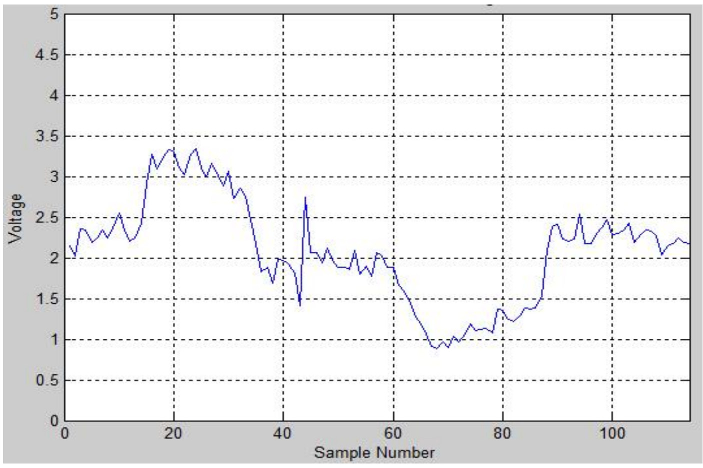


Figure 22. First horizontal recording of an EOG signal, recorded with a sampling frequency of 256 Hz. A movement of the eyes to the right side can be seen starting at sample number 15, and to the left starting at sample number 60.

3.4.2 Implementation of the CIM-protocol and the µC firmware

The data shown above was recorded before the implementation of the CIM protocol. For the implementation of the protocol, testing data packets were sent to and received from the Teensy using HTerm. The sent and received packets can be seen in Table 7.

	ARE to CIM	CIM to ARE
Request Feature List	4054 0100 0000 0D 0000 0000	4054 01A1 0600 0D 0000 0000 0000 1000 0200
Read Serial No.	4054 0100 0000 0E 0000 1100	4054 01A1 0400 0E 0000 1100 4030 2010
Start CIM	4054 0100 0000 11 0000 8100	4054 01A1 0000 11 0000 8100
Write PeriodicTime	4054 0100 0200 12 0100 1000 0010	4054 01A1 0000 12 0100 1000
		4054 01A1 0400 80 0200 2000 0xxx 0xxx
		...

Table 7. CIM-packets sent and received by the µC - This table shows the different commands sent to the Teensy and the corresponding answers. The starting bytes 4054 can be seen as well as the identifying bytes of the EOG-CIM 01A1. Request feature list is answered by 0000, 0001 and 0002, which equals the three features of the EOG-CIM, mentioned in 3.2.2. Read Serial Number is answered by sending the hardcoded serial number of the sensor module: 4030 2010. Furthermore, Write PeriodicTime is answered with an acknowledgement packet and the AD conversion is started and sent automatically.

3.4.3 Realization and test of the BTM-222

For the test of the BTM-222 module, the same packets as above mentioned were sent and the same results as in Table 7 were achieved.

3.4.4 Implementation and test within the AsTeRICS project

For the final implementation in and the tests within the AsTeRICS project, the following model was configured in the AsTeRICS Configuration Suite (ACS). This model is shown in Figure 23 and consists of the EOG sensor plug-in and two oscilloscope plug-ins to visualize the converted digital values. This model was tested two times:

1. electrode arrangement for the horizontal EOG signal
2. electrode arrangement for the vertical EOG signal

On both arrangements the other channel of the Teensy was used by a 2xAA battery cell.
The output to the AsTeRICS Runtime Environment GUI is shown in Figure 24 and Figure 25, respectively.

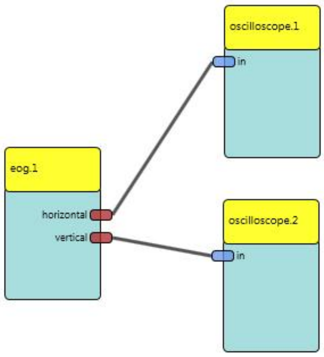


Figure 23. Testing ACS Model for the CIM Implementation, containing of the EOG-Sensor plugin with two channels, and one oscilloscope for each of these channels

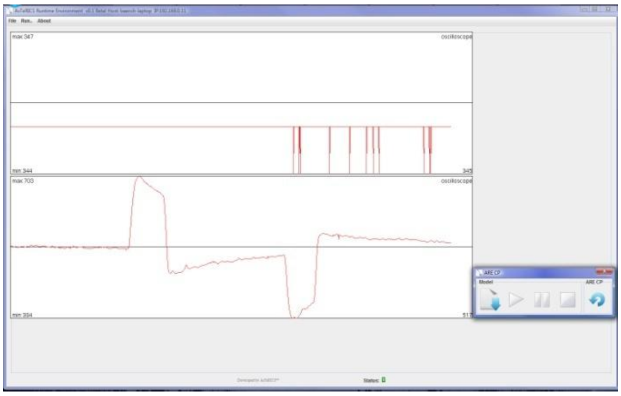


Figure 24. running ACS-Model with a 2xAA battery voltage on channel 1 and the horizontal EOG signal on channel 2, showing a look to the right and left side, respectively

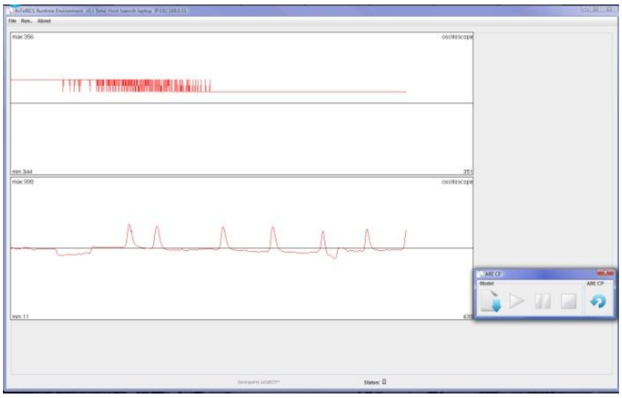


Figure 25. running ACS-Model with 2xAA battery voltage on channel 1 and the vertical EOG signal on channel 2, showing 6 blinkings of the eyes

3.4.5 Test of the whole system (EOG-Sensor, Teensy-Board and AsTeRICS plug-in)

For the final test, the EOG sensor system was connected with an ACS model, which can be seen in Figure 26. This served as a test to examine the usability of the system as a musical instrument, as mentioned in 1.Introduction. Therefore the horizontal channel of the EOG plug-in was connected again to an oscilloscope, to visualize the signal, and to the MIDI-plug-in, created by Mr. Koller (2), to define the pitch of the note played. Furthermore, a text-field reader was used to trigger the note, by pressing "<" on the PC keyboard, what was visualized as a bar-graph. The visual result can be seen in Figure 27.

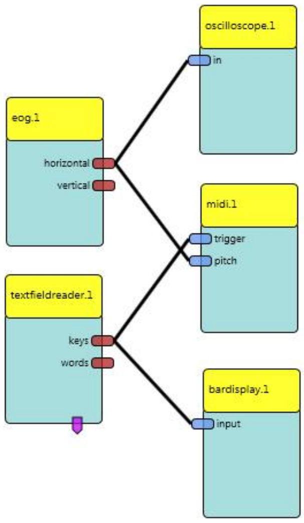


Figure 26. ACS-Model for testing of the whole system

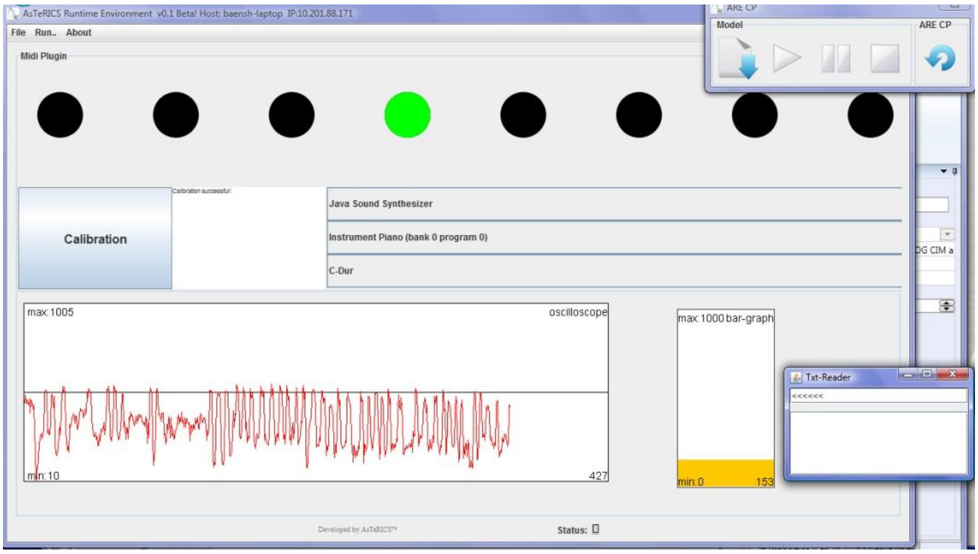


Figure 27. running ACS-Model with horizontal EOG signal, MIDI-plug-in, TXT-Reader and bar-graph, showing the use of the EOG to select the played note. After calibration only the most right and the most left notes could be played intentionally. The signal in red shows the highest possible left and right movement done by the users eyes.

4 Conclusion

4.1 Results

A wireless, battery supplied, one channel EOG-sensor system was developed, which is able to detect if the user is looking to the left-, the middle or the right-hand side, as shown in the chapter before. Also depending on the arrangement of the electrodes, this sensor system can be used for the vertical axis of the eye movement. The system provides an amplification of 2691 which equals about 68 dB. While the frequencies below 100 mHz are less amplified, frequencies higher than 80 Hz are not amplified anymore, but attenuated.
Furthermore, the implementation of the system within the AsTeRICS project was realized, so that the system is able to communicate and interact with the AsTeRICS Runtime Environment via the CIM-protocol properly.

4.2 Considered improvements and future work

There are some goals which were not reached within the extent of this paper, and would not just improve the measuring, but also the handling of the developed system.

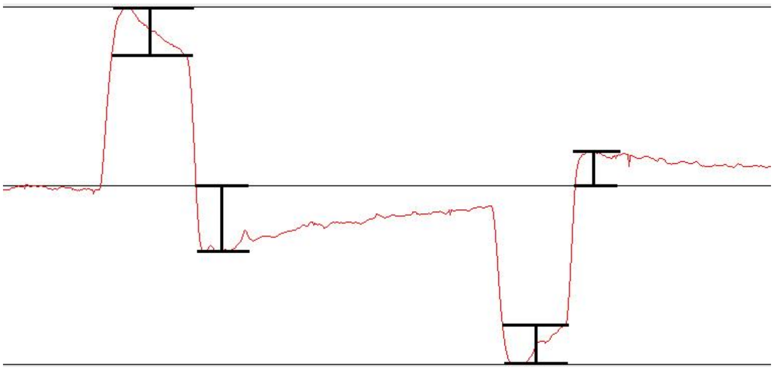


Figure 28. horizontal EOG record showing the attenuation of the EOG signal and the offset shift caused by that attenuation (marked)

As it can be seen in Figure 28, the signal measured, if looking to the right or left side is slowly going back to the predefined offset voltage. Furthermore, the signal, if looking back to the mid is displaced approximately by the part it went back earlier and slowly moves back to the offset voltage. This behaviour of the analog circuit is due to the active high pass, as the direct voltage is filtered out. When there is no eye movement, there is no voltage change. Therefore, a direct voltage exists, which is, however, filtered out. Because of this, the signal decreases slowly until it reaches the offset voltage. As a solution to this problem, another approach to the analog circuit and system design can be used. For example, the high pass can be omitted. Therefore, a µC with an ADC-resolution of 24-bit would be necessary to visualize the signal detailed enough and to be able to differentiate between the offset drift and the direct voltage with steady eyes.

The developed system is a single-channel one, but the realization of this second channel is not a big challenge, as the existing analog circuit would just have to be realized a second time. This was not done due to the missing space on the breadboard, but planned to be included on the soldered grid board.

These issues all refer to the measurement of the electrical signal of the eyes. Further improvements are given by the handling of the system. As the EOG-system is developed on breadboard, the wires and passive components are attached poorly, and can easily be distracted. This can be handled with the development of a grid board, or further on with the etching of a SMD board. This would not only improve the handling of the system, but also lead to a smaller system. When developed on board, a connection other than crocodile wires, between the board and the electrodes has to be installed. Therefore, the implementation of DIN 1.5mm connectors, which are commonly used in bio-signal instrumentation, and Clip-to-DIN 1.5mm wires are recommended. In Figure 29 an example of such a wire is shown.

Figure 29. Clip-to-DIN 1.5mm Wire (47)

4.3 Conclusion

As in 4.1 Results and 4.2 Considered improvements and future work explained, some of the initially planned tasks were reached and some not. To conclude the development done during writing this thesis, the results, including improvements and proposed changes, now are summarized, in Table 8.

Results	**Improvements/Changes**
basic single-channel wireless EOG sensor system	omit high-pass and use 24-bit µC for ADC
AsTeRICS plug-in	solder to grid board / SMD board
CIM protocol implementation on µC	add a second channel
ADC implementation on µC	add DIN 1.5mm connectors and wires

Table 8. Results of this paper and possible improvements and changes

At the end of this thesis, thus, a working single-channel EOG sensor system, which can be used either for the horizontal or for the vertical derivation, was developed on breadboard. The communication with the PC is realized using Bluetooth, and based on the CIM-protocol.
Nevertheless, further developments such as soldering, or another approach to the analog circuit can improve the EOG system.
A practical usage as a musical instrument could not be prooven, but the sensor-system is able to distinguish left-right or up-down eye movement reliable, what, however, can be used in AAL issues.

5 Appendix and indexes

5.1 Bibliography

1. **Gesellschaft für ganzheitliche Förderung und Therapie GmbH.** [Online] [Cited: October 2nd, 2011.] www.gfgf.at.

2. **Koller, D.** *BSc., personal communication, dominik.koller@gmx.at* Vienna, November 22nd, 2011.

3. **AsTeRICS.** [Online] [Cited: October 2nd, 2011.] http://www.asterics.eu/.

4. **Usakli, A.B., et al.** On the Use of Electrooculogram for Efficient Human Computer Interfaces. *Computational Intelligence and Neuroscience.* 2010.

5. **Bárcia, J. C.** *Human electrooculography interface.* Lisbon : Technical University of Lisbon, 2010.

6. **Bulling, A., et al.** *It's in Your Eyes - Towards Contex-Awareness and Mobile HCI Using Wearable EOG Goggles.* Zurich : ETH Zurich, 2008.

7. **Boston College.** About EagleEyes. [Online] [Cited: October 2nd, 2011.] http://www.bc.edu/schools/csom/eagleeyes/about-real.html.

8. **Communication by Gaze Interaction.** Eye Trackers. [Online] [Cited: October 2nd, 2011.] http://www.cogain.org/wiki.

9. **The Opportunity Foundation of America.** [Online] [Cited: October 2nd, 2011.] http://www.ofoa.net/.

10. **Alea Technologies.** Intelligaze IG-30. [Online] [Cited: October 2nd, 2011.] http://www.alea-technologies.de/pages/en/products/intelligaze/ig-30-system.php.

11. **EyeTech Digital Systems.** EyeTech TM30. [Online] [Cited: October 2nd, 2011.] http://www.eyetechds.com/assistive-technology/tm3.

12. **Techcess.** Techcess Ltd - Tellus 3+, Fujitsu tablet computer, wheelchair mountable communication aid, DaeSSy Compatible Mounting Plate, MindExpress, Eurovocs suite. [Online] [Cited: November 17th, 2011.] http://www.techcess.co.uk/3_1_tellus.php .

13. —. Techcess Ltd - EyeTech TM3, Tellus Communication Aid, eye tracker tool, Tellus 3+ communication aid. [Online] [Cited: November 17th, 2011.] http://www.techcess.co.uk/3_7_eyetech_tm3.php.

14. **Schmidt, R. F., Lang, F. and Thews, G.** *Human Physiology.* New York : Springer Verlag, 1989.

15. **Veigl, C.** *Ein universelles System zur Anwendung von Biosignalen im, Biofeedback und als Human Computer Interface.* Vienna : Institut "integriert studieren", 2007.

16. **Lynch, P. J.** *Lateral eye and orbit anatomy with nerves.* 2009.

17. **Speckmann, E.-J., Hescheler, J. and Köhling, R.** *Physiologie.* Munich : Elsevier GmbH, 2008.

18. **Joos, M., Rötting, M. and Velichkovsky, B. M.** *Bewegungen des menschlichen Auges: Fakten, Methoden und innovative Anwendungen.* Dreden : TU Dresden, 2002.

19. **SpoMedial - Ruhr University Bochum.** Gesichtsfeld und periphäres Sehen. [Online] [Cited: October 6th, 2011.] http://vmrz0100.vm.ruhr-uni-bochum.de/spomedial/content/e866/e2442/e8554/e8574/e8610/e8656/index_ger.html.

20. **Huppelsberg, J. and Walter, K.** *Kurzlehrbuch Physiologie.* Stuttgart : Georg Thieme Verlag KG, 2005.

21. **Sherwood, L.** *Human Physiology: From Cells to Systems.* Belmont : Brooks/Cole, Cengage Learning, 2010.

22. **Malmivuo, J. and Plansey, R.** *Bioelectromagnetism - Principles and Applications of Bioelectric and Biomagnetic Fields.* New York : Oxford University Press, 1995.

23. **Bois-Reymond, E.d.** *Untersuchungen Ueber Thierische Elektricität.* Berlin : G Reimer, 1848.

24. **Young, L.R. and Sheena, D.** Eye-movement measurement techniques. *Encyclopedia of Medical Devices and Instrumentation.* J.G. Webster, 1988.

25. **Skintact.** Connect with Quality. [Online] [Cited: October 16th, 2011.] http://www.skintact.com/296.0.html.

26. **robotikhardware.de.** DC-DC Wandler. [Online] [Cited: November 22nd, 2011.] http://www.shop.robotikhardware.de/shop/catalog/product_info.php?products_id=194.

27. **Texas Instruments.** Operational Amplifier uA741. [Online] [Cited: November 22nd, 2011.] http://www.ti.com/product/ua741.

28. **Schematica Software.** Single Supply Active Filter Design. [Online] Schematica Software. [Cited: October 21st, 2011.] http://www.schematica.com/resources/op_amp_virtual_ground_circuits.htm.

29. **Texas Instruments.** Instrumentational Amplifiers - Single Supply - INA122. [Online] [Cited: November 22nd, 2011.] http://www.ti.com/product/ina122.

30. —. Precision Amplifiers - OPA2336. [Online] [Cited: November 22nd, 2011.] http://www.ti.com/product/opa2336.

31. **Linear Technology.** LT - Design Simulation and Device Models. [Online] [Cited: November 22nd, 2011.] http://www.linear.com/designtools/software/.

32. **openEEG.** Amplifier Schematic - The ModularEEG. [Online] [Cited: October 21st, 2011.] http://openeeg.sourceforge.net/doc/modeeg/modEEGamp-v1.0.png.

33. **Metting van Rijn, A. C., Peper, A. and Grimbergen, C. A.** High-quality recordings of bioelectric events - Part 1 Interference reduction, theory and practice. *Med. & Bio. Eng. & Comput.* 1990.

34. **PJRC.** Teensy USB Development Board. [Online] [Cited: November 22nd, 2011.] http://www.pjrc.com/teensy/.

35. **Atmel Corporation.** Atmel AVR 8- and 32-bit Microcontrollers - AVRmega. [Online] [Cited: November 22nd, 2011.] http://www.atmel.com/dyn/products/product_card.asp?part_id=3874.

36. **FTDI Chip.** USB TTL Serial. [Online] [Cited: November 22nd, 2011.] http://www.ftdichip.com/Products/Cables/USBTTLSerial.htm.

37. **Hammer, T.** der-Hammer: HTerm - A Terminal Program For Windows and Linux. [Online] [Cited: November 22nd, 2011.] http://www.der-hammer.info/terminal/.

38. **AsTeRICS.** *Communication Protocol for CIMs - AsTeRICS Internal Document.* 2010.

39. **ZigBee Alliance.** ZigBee Alliance Home. [Online] [Cited: November 22nd, 2011.] http://www.zigbee.org/.

40. **Demand Media, Inc.** Bluetooth History | eHow.com. [Online] [Cited: November 22nd, 2011.] http://www.ehow.com/info_8065916_bluetooth-history.html.

41. **Bluetooth Special Interest Group.** SIG Membership. [Online] [Cited: November 22nd, 2011.] http://www.bluetooth.com/Pages/SIG-Membership.aspx.

42. **NetworkDictionary.** [Online] [Cited: November 29th, 2011.] http://www.networkdictionary.com/wireless/WPAN.php.

43. **Gadatsch, A.** *Grundkurs Geschäftsprozess-Management: Methoden und Werkzeuge für die IT-Praxis.* Wiesbaden : Driedr. Vieweg & Sohn, 2006.

44. **Plischka, P.** BTM-222 Class1 Modul mit SPP-Firmware. [Online] [Cited: November 22nd, 2011.] http://plischka.at/Funk_Bluetooth.html.

45. **Rayson Technology.** Bluetooth® Dongle / Module. [Online] [Cited: November 22nd, 2011.] http://www.rayson.com/btm220.html.

46. **Tatham, S., Dunn, O., Harris, B., Nevins, J.** Download PuTTY - a free SSH and telnet client for Windows. [Online] [Cited: November 29th, 2011.] http://www.putty.org/.

47. **Schuler Medizintechnik.** Elektroden für die neurophysiologische Funktionsdiagnostik. [Online] [Cited: October 16th, 2011.] http://www.schulermed-tech.de/index.php?id=21.

5.2 List of Figures

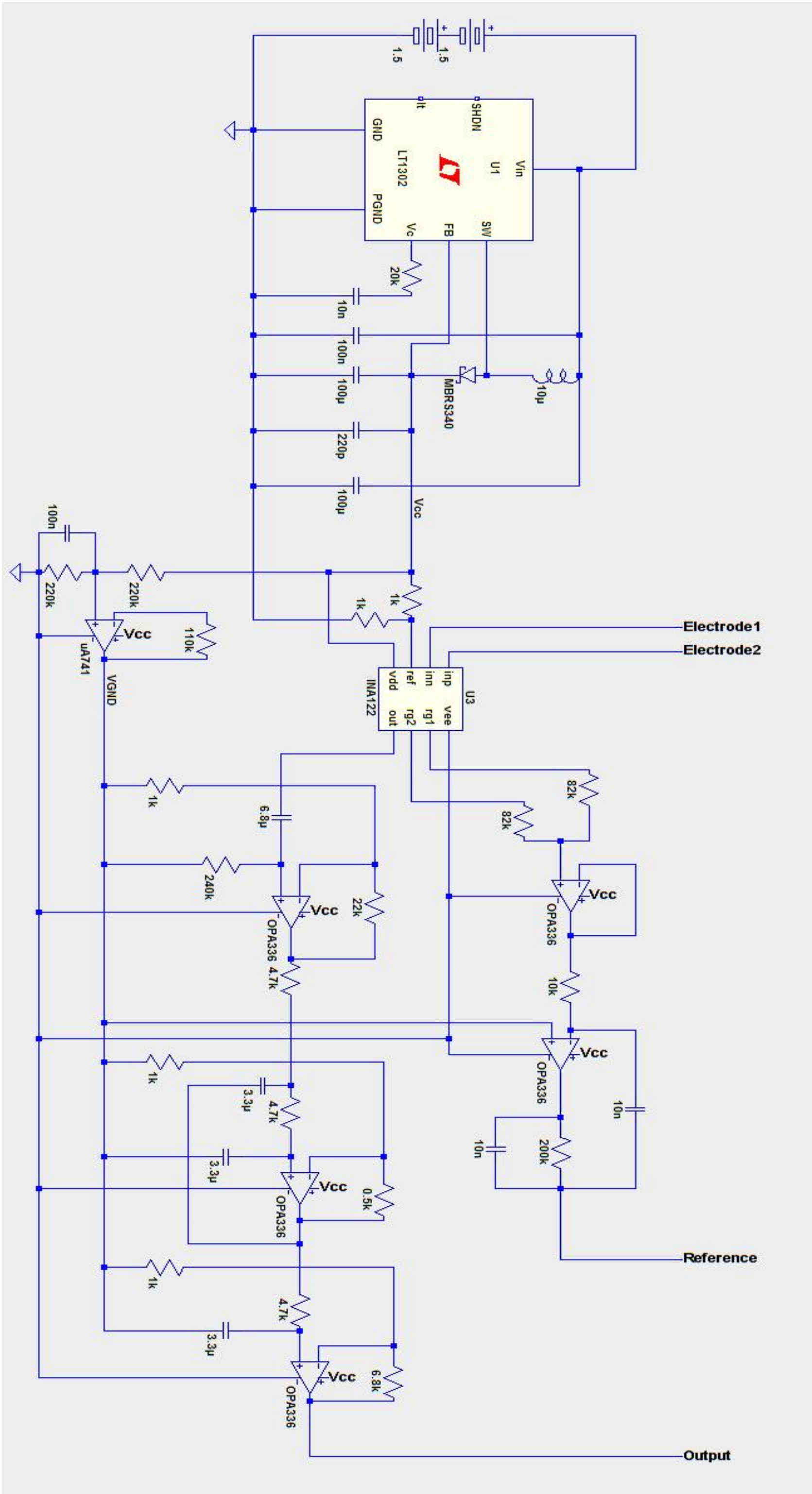
5.3 List of Tables

5.4 List of abbreviations

EOG	Electro-occulogram
AsTeRICS	Assistive Technology Rapid Integration Construction Set
DRL	Driven right leg
ARE	AsTeRICS Runtime Environment
HCI	Human Computer Interface
µC	Microcontroller
BT	Bluetooth
AAL	Ambient assisted living
EMG	Electro-myogram
ADC	Analog – digital conversion
EDR	Enhanced Data Rate
ACS	AsTeRICS Configuration Suite
MIDI	Musical Instrument Digital Interface
SMD	Surface-Mounted Device
DIN	Deutsches Institut für Normung

5.5 Appendix

5.5.1 Total Circuit Schematic



5.5.2 µC – Code for A/D conversion and CIM implementation

UARTinit()

```
void UART_Init(long baudrate)
{
       init_InBuf(&uart_in);
       init_OutBuf(&output_buffer);

       UBRR1 = F_CPU / baudrate / 8 - 1;

       UCSR1A |= (1<<U2X1);
       UCSR1B = (1<<RXEN1) | (1<<TXEN1) | (1<<RXCIE1);
       UCSR1C =  ( 1<<UCSZ10 | 1<<UCSZ11 );
}
```

parse_CIM_protocol()

```
void parse_CIM_protocol(void)
{
    uint32_t checksum=0;
    static uint8_t transmission_mode;
       static uint8_t reply_status_code;
       static uint8_t last_serial;
       uint8_t actbyte;

    while (keys_in_InBuf(&uart_in)) {
     actbyte=read_InBuf(&uart_in);

    switch (readstate)
       {
                case 0: // first sync byte
                              reply_status_code=0;
                              if (actbyte=='@') readstate++;
                              break;
                case 1: // second sync byte
                              if (actbyte=='T')  readstate++;
                              else readstate=0;
                              break;

                // packet in sync !

                case 2: // ARE-ID: SW-version low byte
                              ARE_frame.are_id=actbyte;
                              readstate++;
                              break;
                case 3: // ARE-ID: SW-version high byte
                              ARE_frame.are_id+=((uint16_t)actbyte)<<8;
                              if (ARE_frame.are_id < ARE_MINIMAL_VERSION)    //
outdated ARE ?
                                 reply_status_code |= CIM_ERROR_INVALID_ARE_VERSION;
                              readstate++;
                              break;
                case 4: // data length low byte
                              ARE_frame.data_size=actbyte;
                              readstate++;
                              break;
                case 5: // data length high byte
                              ARE_frame.data_size+=((uint16_t)actbyte)<<8;
```

```
                              if (ARE_frame.data_size > DATABUF_SIZE)  { // dismiss
packets of excessive length
                                   readstate=0;
                                      //ARE_frame.data_size=0;
                                   //reply_status_code |= CIM_ERROR_INVALID_FEATURE;
                              }
                              else readstate++;
                              break;
             case 6: // serial_number
                              ARE_frame.serial_number = actbyte;
                              if (first_packet)    // don't check first serial
                                  first_packet=0;
                              else if (actbyte != (last_serial+1)%0x80) // check
current serial number
                                 reply_status_code |= CIM_ERROR_LOST_PACKETS;

                              last_serial=actbyte;
                           readstate++;
                              break;
                case 7: // CIM-feature low byte
                              ARE_frame.cim_feature= actbyte;
                              readstate++;
                              break;
                case 8: // CIM-feature high byte
                              ARE_frame.cim_feature+=((int)actbyte)<<8;
                              readstate++;
                              break;
                case 9: // Request code low byte ( command )
                              ARE_frame.request_code=actbyte;
                              readstate++;
                              break;
                case 10:// Request code high byte (transmission mode)
                              transmission_mode=actbyte;  // bit 0: CRC enable
                              // reply_status_code|=(actbyte & CIM_STATUS_CRC);
                              // remember CRC state for reply

                              if (ARE_frame.data_size>0) {
                                   readstate++;
                                   ARE_frame.data= databuf;
                                   datapos=0;
                              }
                              else {  // no data in packet
                                if (transmission_mode & CIM_STATUS_CRC)
                                      readstate+=2;      // proceed with CRC
                                   else readstate=FRAME_DONE;  // frame is finished
here !
                              }
                              break;

                case 11: // read out data
                              ARE_frame.data[datapos]=actbyte;
                              datapos++;
                              if (datapos==ARE_frame.data_size)
                              {
                                 if (transmission_mode & CIM_STATUS_CRC)  // with CRC:
get checksum

                                    readstate++;
                                   else  readstate=FRAME_DONE; // no CRC:  frame is

finished here !
                              }
```

```
                                break;
                case 12: // checksum byte 1
                                checksum=actbyte;
                                readstate++;
                                break;
                case 13: // checksum byte 2
                                checksum+=((long)actbyte)<<8;
                                readstate++;
                                break;
                case 14: // checksum byte 3
                                checksum+=((long)actbyte)<<16;
                                readstate++;
                                break;
                case 15: // checksum byte 4
                                checksum+=((long)actbyte)<<24;

                                // check CRC now (currently not used):
                                // if (checksum != crc32(ARE_frame.data,
ARE_frame.data_size))
                                reply_status_code |= CIM_ERROR_CRC_MISMATCH;

                                // frame finished here !
                                readstate=FRAME_DONE;
                                break;
                default: readstate=0; break;
              }

        if (readstate==FRAME_DONE)  {  // frame finished: store command in ringbuffer
                      process_ARE_frame(reply_status_code);
                      readstate=0;
              }
        }
}
```

process_ARE_frame()

```
uint8_t process_ARE_frame(uint8_t status_code)
{
    uint8_t ack_needed, i;
       uint8_t data_size=0;
       uint8_t command;
       struct ringbuf_i * new_insert;

       command=(uint8_t)ARE_frame.request_code;

       CIM_frame.cim_feature=ARE_frame.cim_feature;
       CIM_frame.serial_number=ARE_frame.serial_number;
       CIM_frame.reply_code=(((uint16_t)status_code)<<8) + command;
       data_size=(uint8_t)ARE_frame.data_size;

       ack_needed=1;
       new_insert=0;

       if ((status_code & CIM_ERROR_INVALID_ARE_VERSION) == 0)
       {
                 // no serious packet error
             switch (command)   {  // process requested command

                    case CMD_REQUEST_FEATURELIST:
```

```
                            if (data_size==0) {
                                 reply_FeatureList();  // reply requested feature list
                                   ack_needed=0;
                                 } else status_code |= CIM_ERROR_INVALID_FEATURE;
                                 break;

                     case CMD_REQUEST_RESET_CIM:
                     case CMD_REQUEST_START_CIM:
                                 if (data_size==0) {
                                 reset_EOG_CIM();  // reset first frame indicator etc.
                                 } else status_code |= CIM_ERROR_INVALID_FEATURE;
                              break;



                     case CMD_REQUEST_STOP_CIM:
                            if (data_size==0) {
                                 reset_EOG_CIM();  // reset first frame indicator etc.
                                 } else status_code |= CIM_ERROR_INVALID_FEATURE;
                              break;

                     case CMD_REQUEST_READ_FEATURE:  //  read feature from CIM
                              switch (ARE_frame.cim_feature) {
                                 case EOG_CIM_FEATURE_UNIQUENUMBER:   // read unique
serial number
                                 if (data_size==0) {
                                        reply_UniqueNumber();
                                          ack_needed=0;
                                        } else status_code |=
CIM_ERROR_INVALID_FEATURE;
                                 break;

                                 default:
                                        status_code |= CIM_ERROR_INVALID_FEATURE;
                               }
                               break;

                     case CMD_REQUEST_WRITE_FEATURE:  //  write feature to CIM
                                 switch (ARE_frame.cim_feature) {  // which feature
address ?
                                        case
EOG_CIM_FEATURE_ACTIVE_PERIODIC_VALUE_REPORTS :   // activate timer reg. periodic
value + adc + uart

                                          if (data_size==2)
                                          {

                                                 reply_Acknowledge();

                                                 reload =
init_timer((uint16_t)ARE_frame.data[0]);
                                                 init_adc();

                                                 CIM_frame.serial_number=127;
                                                 CIM_frame.cim_feature=0x0002;
                                                 TIMSK1 |= (1<<TOIE1);
                                                 sei();
                                                 ack_needed=0;

                                          } else status_code |=
```

```
CIM_ERROR_INVALID_FEATURE;
                                            break;

                                          default:
                                                 status_code |=
CIM_ERROR_INVALID_FEATURE;
                                   }
                                   break;

                     default:
                        status_code |= CIM_ERROR_INVALID_FEATURE;
              }
    }

    if (new_insert)
         if (free_in_InBuf(new_insert) < INBUF_WARNING) {  // indicate input buffer
warning
               status_code |= CIM_ERROR_CIM_NOT_READY;
         }

       if (ack_needed) {
           reply_Acknowledge();
       }

    return(1);
}
```

init_timer()

```
uint16_t init_timer(uint16_t periodTime)
{
       uint16_t RELOAD = (65536 - (( F_CPU / 1024) / (1000/periodTime)));

       TCNT1 = RELOAD;                      //  Set Timer1 Counter to reload
    TCCR1B = (1<<CS12) | (1<<CS10);      //  Set Prescaler to  1024
                                                 //  16000000 / 1024 = 15625
counts per second
       return RELOAD;
}
```

init_adc()

```
void init_adc()
{
    ADCSRA =  (1<<ADPS2) | (1<<ADPS1);  // Prescaler = 64, free running mode = off,
interrupts off.
    ADCSRA |= (1<<ADIF);                // Reset any pending ADC interrupts
    ADCSRA |= (1<<ADEN);                // Enable the ADC
}
```

ISR(TIMER1_OVF_vect)

```
ISR(TIMER1_OVF_vect)   // Sampling timer (timer 1) interrupt service routine
{
       TCNT1= reload;     // reload the timer counter for 256 Hz sampling frequency
```

```
	if (CIM_frame.serial_number==255)		//get serial-nr. between 0x80
and 0xFF
	{
		CIM_frame.serial_number=128;
	}
	else CIM_frame.serial_number++;

    UCSR1B &= ~(1<<UDRIE1);				//  Ensure UART IRQ's are
disabled.
	ADCSRA |= (1<<ADIE);				// Enable ADC interrupts.
    ADMUX = (1<<REFS0);					//  start sampling
with ADC chn 0, use internal 5V reference
    ADCSRA |= (1<<ADIF);				//  Reset any pending ADC
interrupts
    ADCSRA |= (1<<ADSC);				//  Start the ADC
}
```

ISR(ADC_vect)

```
ISR(ADC_vect)	//  AD-conversion-complete interrupt service routine
{

			low = ADCL;					// read ADC value
(low byte first !)
			high = ADCH;				// read ADC value high
byte

			TXBuf[1+channel*2]= low;		// fill transmit-buffer with
ADC values
			TXBuf[0+channel*2]= high;

			channel++;					// next channel

			if (channel==2)				// ADC finished

			{
				channel=0;
				reply_Values(&TXBuf);		// send packet
			}
			else
			{
				ADMUX = (1<<REFS0)+channel;	// select the next channel,
use internal 5V reference
				ADCSRA |= (1<<ADSC);		// start ADC of next channel
			}
}
```

replyValues()

```
void reply_Values(char *values)
{
    uint8_t free_bytes = free_in_OutBuf(&output_buffer);

	if (free_bytes < OUTBUF_WARNING) {
		CIM_frame.reply_code |= (CIM_ERROR_CIM_NOT_READY<<8) ;
```

```
    }
       if (free_bytes > 15) {
              CIM_frame.data_size=4;
              CIM_frame.reply_code=0x0020;
              UART_Send_NonBlocking ((char *) &CIM_frame, CIM_HEADER_LEN);
              UART_Send_NonBlocking ((char *) &TXBuf, CIM_frame.data_size);
       }
}
```

5.5.3 EOG plug-in Code for AsTeRICS implementation

handlePacketReceived()

```
public void handlePacketReceived(CIMEvent e)
{

      CIMEventPacketReceived ev = (CIMEventPacketReceived) e;
      CIMProtocolPacket packet = ev.packet;

      short featureAddress = 0;
      featureAddress=packet.getFeatureAddress();

      switch(packet.getRequestReplyCode())
      {
            case CIMProtocolPacket.COMMAND_REPLY_START_CIM:
                  System.out.println ("Reply Start.");
                  break;

            case CIMProtocolPacket.COMMAND_REPLY_STOP_CIM:
                  System.out.println ("Reply Stop.");
                  break;

            case CIMProtocolPacket.COMMAND_REPLY_RESET_CIM:
                  System.out.println ("Reply Reset.");
                  break;

            case CIMProtocolPacket.COMMAND_REPLY_READ_FEATURE:
                  System.out.print ("Reply Read: ");

                  if (featureAddress == EOG_FEATURE_SERIAL_NUMBER )
                  {
                        System.out.println ("UniqueNumber");
                        handleEogSerialNumber(packet);
                  }
                  break;

            case CIMProtocolPacket.COMMAND_EVENT_REPLY:
                  System.out.print ("...");

                  if (featureAddress == EOG_FEATURE_CHANNEL_VALUE_REPORT)
                  {
                        handleEogInputValuePacket(packet);
                  }
                  break;

            case CIMProtocolPacket.COMMAND_REPLY_WRITE_FEATURE:
```

```
                    System.out.print ("Replay Write: ");

                    if (featureAddress ==
EOG_FEATURE_ACTIVATE_PERIODIC_VALUE)
                    {
                          System.out.println ("Set AdcPeriod.");
                    }
                    break;
        }
}
```

handleEogInputValuePacket()

```
        private void handleEogInputValuePacket(CIMProtocolPacket packet)
        {
              byte [] b = packet.getData();
              int horizontal = 0;
              int vertical = 0;

              horizontal =  ((int) b[0]) & 0xff;
              horizontal =  horizontal | ((((int) b[1]) & 0xff) << 8);


              vertical =  ((int) b[2]) & 0xff;
              vertical =  vertical | ((((int) b[3]) & 0xff) << 8);

              opHorizontal.sendData(ConversionUtils.intToBytes(horizontal));

              opVertical.sendData(ConversionUtils.intToBytes(vertical));
        }
```