A multi-modal device for application in microsleep detection

Simon Knopp

A thesis presented for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering
at the
University of Canterbury,
Christchurch, New Zealand.

May 2015

ABSTRACT

Microsleeps and other lapses of responsiveness can have severe, or even fatal, consequences for people who must maintain high levels of attention on monotonous tasks for long periods of time, e.g., commercial vehicle drivers, pilots, and air-traffic controllers. This thesis describes a head-mounted system which is the first prototype in the process of creating a system that can detect (and possibly predict) these lapses in real time. The system consists of a wearable device which captures multiple physiological signals from the wearer and an extensible software framework for implementing signal processing algorithms. Proof-of-concept algorithms are implemented and used to demonstrate that the system can detect simulated microsleeps in real time.

The device has three sensing modalities in order to get a better estimate of the user's cognitive state than by any one alone. Firstly, it has 16 channels of EEG (8 currently in use) captured by 24-bit ADCs sampling at 250 Hz. The EEG is acquired by custom-built dry electrodes consisting of spring-loaded, gold-plated pins. Secondly, the device has a miniature video camera mounted below one eye, providing 320×240 px greyscale video of the eye at 60 fps. The camera module includes infrared illumination so that it can operate in the dark. Thirdly, the device has a six-axis IMU to measure the orientation and movement of the head. These sensors are connected to a Gumstix computer-on-module which transmits the captured data to a remote computer via Wi-Fi. The device has a battery life of about 7.4 h.

In addition to this hardware, software to receive and analyse data from the head-mounted device was developed. The software is built around a signal processing pipeline that has been designed to encapsulate a wide variety of signal processing algorithms; feature extractors calculate salient properties of the input data and a classifier fuses these features to determine the user's cognitive state. A plug-in system is provided which allows users to write their own signal processing algorithms and to experiment with different combinations of feature extractors and classifiers. Because of this flexible modular design, the system could also be used for applications other than lapse detection—any application which monitors EEG, eye video, and head movement can be implemented by writing appropriate signal processing plug-ins, e.g., augmented cognition or passive BCIs. The software also provides the ability to

configure the device's hardware, to save data to disk, and to monitor the system in real time. Plug-ins can be implemented in C++ or Python.

A series of validation tests were carried out to confirm that the system operates as intended. Most of the measured parameters were within the expected ranges: EEG amplifier noise = 0.14 μV_{RMS} input-referred, EEG pass band = DC to 47 Hz, camera focus = 2.4 lp/mm at 40 mm, and total latency < 100 ms. Some parameters were worse than expected but still sufficient for effective operation: EEG amplifier CMRR $\geq$ 82 dB, EEG cross-talk = -17.4 dB, and IMU sampling rate = 10 Hz. The contact impedance of the dry electrodes, measured to be several hundred kilohms, was too high to obtain clean EEG.

Three small-scale experiments were done to test the performance of the device in operation on people. The first two demonstrated that the pupil localization algorithm produces PERCLOS values close to those from a manually-rated gold standard and is robust to changes in ambient light levels, iris colour, and the presence of glasses. The final experiment demonstrated that the system is capable of capturing all three physiological signals, transmitting them to the remote computer in real time, extracting features from each signal, and classifying simulated microsleeps from the extracted features. However, this test was successful only when using conventional wet EEG electrodes instead of the dry electrodes built into the device; it will be necessary to find replacement dry electrodes for the device to be useful.

The device and associated software form a platform which other researchers can use to develop algorithms for lapse detection. This platform provides data capture hardware and abstracts away the low-level software details so that other researchers are free to focus solely on developing signal processing techniques. In this way, we hope to enable progress towards a practical real-time, real-world lapse detection system.

Deputy Vice-Chancellor's Office
Postgraduate Office

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

Co-Authorship Form

This form is to accompany the submission of any thesis that contains research reported in co-authored work that has been published, accepted for publication, or submitted for publication. A copy of this form should be included for each co-authored work that is included in the thesis. Completed forms should be included at the front (after the thesis abstract) of each copy of the thesis submitted for examination and library deposit.

Please indicate the chapter/section/pages of this thesis that are extracted from co-authored work and provide details of the publication or submission from the extract comes:
Section 6.1.1 is adapted from a published conference paper: Knopp, S.J., Bones, P.J., Weddell, S.J., Innes, C.R.H. and Jones, R.D. (2012), 'A miniature head-mounted camera for measuring eye closure', In Proceedings of the 27th Conference on Image and Vision Computing New Zealand, IVCNZ '12, ACM, pp. 313–318.

Please detail the nature and extent (%) of contribution by the candidate:
The PhD candidate did the research and wrote all sections of the published manuscript. The co-authoring supervisory team suggested research directions and offered criticisms which the candidate worked into the manuscript.

Certification by co-authors

If there is more than one co-author then a single co-author can sign on behalf of all. The undersigned certifies that:

- The above statement correctly reflects the nature and extent of the PhD candidate's contribution to this co-authored work
- In cases where the candidate was the lead author of the co-authored work, he or she wrote the text.

Name: *Richard Jones* Signature:  Date: *30 April 2015*

Deputy Vice-Chancellor's Office
Postgraduate Office

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

Co-Authorship Form

This form is to accompany the submission of any thesis that contains research reported in co-authored work that has been published, accepted for publication, or submitted for publication. A copy of this form should be included for each co-authored work that is included in the thesis. Completed forms should be included at the front (after the thesis abstract) of each copy of the thesis submitted for examination and library deposit.

Please indicate the chapter/section/pages of this thesis that are extracted from co-authored work and provide details of the publication or submission from the extract comes:
Section 6.1.2 is adapted from a published conference paper: Knopp, S.J., Bones, P.J., Weddell, S.J., Innes, C.R.H. and Jones, R.D. (2013), A 'wearable device for measuring eye dynamics in real-world conditions', In Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, pp. 6615–6618.

Please detail the nature and extent (%) of contribution by the candidate:
The PhD candidate did the research and wrote all sections of the published manuscript. The co-authoring supervisory team suggested research directions and offered criticisms which the candidate worked into the manuscript.

Certification by co-authors

If there is more than one co-author then a single co-author can sign on behalf of all. The undersigned certifies that:

- The above statement correctly reflects the nature and extent of the PhD candidate's contribution to this co-authored work
- In cases where the candidate was the lead author of the co-authored work, he or she wrote the text.

Name: *Richard Jones* Signature:  Date: *30 April 2015*

CONTENTS

ACKNOWLEDGEMENTS

Firstly, many thanks to my supervisors: Richard Jones, Phil Bones, Carrie Innes and Steve Weddell. Richard, thank you for offering me this project, for your patient help along the way, and for your assiduous attention to commas and hyphens. Phil, thank you for your guidance and wise recommendations; they have been invaluable. Carrie, thanks for sharing your knowledge of all the ins and outs of lapse detection. Steve, thank you for your useful technical pointers. Thanks also to Govinda Poudel for input in the early stages of the project.

To the technical staff that helped with this project, Mike Cusdin and Dave Healy, thank you for your skilled hands and technical know-how.

Thanks to my family and friends, who have supported and encouraged me over the course of this project, especially when things were going two steps forward, one step back, and another step sideways.

To my office-mates and colleagues, Yaqub, John, Sudhanshu, Alex, and Reza, thanks for bringing out the lighter side of being a postgrad (even if the jokes were a bit pun-ishing). Thanks also to Russ Buckley for being such a willing guinea pig.

I am very grateful to the thousands of people that contributed the millions of lines of open-source software that this project builds upon. I have stood on the shoulders of giants.

Most importantly, to the God who both created the human brain and gave us the ability to study it, all thanks are due.

Soli Deo Gloria — to God alone be the glory.

PREFACE

The research for this thesis was completed between March 2011 and December 2014 while enrolled in the Department of Electrical and Computer Engineering at the University of Canterbury. The work was carried out as part of the Christchurch Neurotechnology Research Programme at the New Zealand Brain Research Institute and was supervised by Professor Richard Jones, Professor Phil Bones, Dr Carrie Innes, and Dr Steve Weddell. I was supported by a University of Canterbury Doctoral Scholarship and a NeuroTech Scholarship. The Canterbury Medical Research Foundation and the Canterbury branch of the Royal Society of New Zealand kindly provided travel grants for conference attendance.

PUBLICATIONS

Conference papers

- Knopp, S.J., Bones, P.J., Weddell, S.J., Innes, C.R.H. and Jones, R.D. (2012), 'A miniature head-mounted camera for measuring eye closure', In *Proceedings of the 27th Conference on Image and Vision Computing New Zealand (IVCNZ)*, ACM, pp. 313–318.
- Knopp, S.J., Bones, P.J., Weddell, S.J., Innes, C.R.H. and Jones, R.D. (2013), 'A wearable device for measuring eye dynamics in real-world conditions', In *Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, pp. 6615–6618.

Magazine articles

- Winthrop, M. (2013), 'Wake-up call', *Engineering Insight*, Vol. 14, No. 1, pp. 43–44, IPENZ.

PRESENTATIONS

- August 2012, "Thesis in Three" competition, University of Canterbury finals. Oral presentation.

- November 2012, Image and Vision Computing New Zealand conference (IVCNZ), Dunedin, NZ. Poster presentation.
- July 2013, IEEE Engineering in Medicine and Biology conference (EMBC), Osaka, Japan. Oral presentation.

RELATED WORK

I assisted with the supervision of an undergraduate summer research project to develop dry EEG electrodes and a bioelectrode impedance meter. That work helped to inform the later design of the dry electrodes developed as part of this project.

NOMENCLATURE

Biomedical abbreviations

ASL	Arterial spin labelling
BCI	Brain–computer interface
BM	Behavioural microsleep
BOLD	Blood-oxygen-level dependent
CBF	Cerebral blood flow
ECG	Electrocardiography
EEG	Electroencephalography
EMG	Electromyography
EOG	Electrooculography
ERP	Event-related potential
fMRI	Functional magnetic-resonance imaging
fNIR	Functional near-infrared
MRI	Magnetic-resonance imaging
pBCI	Passive brain–computer interface
PERCLOS	Percentage eye closure
PVT	Psychomotor vigilance task
REM	Rapid eye movement (sleep)
SSVEP	Steady-state visually-evoked potential
VOR	Vestibulo-ocular reflex

Technical abbreviations

ABI	Application binary interface
ADC	Analogue-to-digital converter
AFE	Analogue front-end
API	Application programming interface
ASIC	Application-specific integrated circuit
AUC	Area under the curve

BGA	Ball grid array
CMOS	Complementary metal oxide semiconductor
CMRR	Common-mode rejection ratio
COM	Computer-on-module
CPU	Central processing unit
CSI-1	Camera serial interface
DHCP	Dynamic host configuration protocol
DNS	Domain name system
DSP	Digital signal processor
DVV	Delayed vector variance
EDR	Enhanced data rate
EMI	Electromagnetic interference
ESN	Echo-state network
FFC	Flat flexible cable
FFT	Fast Fourier transform
FIFO	First-in first-out queue
FPGA	Field-programmable gate array
GIL	Global interpreter lock (Python)
GPIO	General-purpose input/output
GUI	Graphical user interface
IC	Integrated circuit
ICA	Independent component analysis
I^2C	Inter-integrated circuit (serial bus)
IMU	Inertial measurement unit
I/O	Input/output
IP	Internet protocol
IPC	Inter-process communication
IR	Infrared
ISP	Image signal processor
LDA	Linear discriminant analysis
LDO	Low-dropout voltage regulator
LED	Light-emitting diode
LiPo	Lithium-ion polymer (battery chemistry)
LSTM	Long short-term memory
MIPI	Mobile industry processor interface
MTU	Maximum transmission unit
NAND	Non-volatile flash memory
NIR	Near-infrared
OS	Operating system
PCA	Principal component analysis

PCB	Printed circuit board
PGA	Programmable-gain amplifier
pimpl	Pointer to implementation
POSIX	Portable operating system interface
PSD	Power spectral density
RAM	Random-access memory
RANSAC	Random sample consensus
RC	Resistor-capacitor
RGB	Red, green, blue (colour space)
RMS	Root mean square
RNN	Recurrent neural network
ROC	Receiver operating characteristic
ROI	Region of interest
RPC	Remote procedure call
RTOS	Real-time operating system
RTP	Real-time transport protocol
SDRAM	Synchronous dynamic random-access memory
SoC	System-on-chip
SPI	Serial peripheral interface
SubLVDS	Sub-low-voltage differential signalling
TCP	Transmission control protocol
TDL-MLP	Tapped delay-line multilayer perceptron
UART	Universal asynchronous receiver/transmitter
UDP	User datagram protocol
USB	Universal serial bus
V4L2	Video for Linux
VGA	Video graphics array (640×480 pixels)
XML	Extensible markup language
YCbCr	Luma + chrominance (colour space)

Other abbreviations

CAMEL	Camera for measuring eye-related lapse indicators
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
NeuroTech	Christchurch Neurotechnology Research Programme
NZBRI	New Zealand Brain Research Institute

Chapter 1

INTRODUCTION

1.1 MOTIVATION

Many occupations and activities require people to maintain a constant state of high alertness or risk serious consequences. For instance, pilots, truck drivers, air traffic controllers, process control operators, and medical workers risk causing fatal mishaps if their level of alertness drops while on the job [Sigurdson and Ayas 2007, Torsvall and Åkerstedt 1987]. As the level of automation in many systems increases, the role of the operator becomes more supervisory. Such jobs can involve extended periods with little happening, but the operator must be ready to respond to abnormal events quickly [Bainbridge 1983]. Compounding the situation, many of these occupations involve night shifts and require prolonged periods of minimal physical activity.

Creating a device capable of detecting when people become unresponsive and alerting them has the potential to avert fatalities. Unfortunately, creating such a device is not trivial. As described in Chapter 2, there is no one measurable signal that has been proven to be a reliable indicator of alertness. A robust measure of alertness, then, must incorporate several distinct signals that in combination give a full picture of the subject's state.

The types of signals that are useful for detecting lapses of responsiveness could also be used for detecting other aspects of the user's cognitive state. If we are going to the effort of developing a device for lapse detection, it would be desirable to make it possible to implement other state/event detection algorithms on the device as well.

1.2 TERMINOLOGY

Before reviewing the relevant literature, it is necessary to define some terminology. Terminology in this field varies widely and there is little standardization between research groups.

Lapses Complete transient disruptions in sensory-motor/cognitive performance from ~0.5–15 s during an active task [Jones et al. 2010]. These can be further categorized as:

Behavioural microsleeps (BMs)
A brief period of suspension of performance and appearing asleep. Also referred to as "microsleeps", though they should not be confused with EEG microsleeps which are short periods of EEG-defined sleep (see *sleep* below) [Peiris et al. 2006]. EEG microsleeps can occur without a drop in performance and BMs can occur without typical EEG sleep markers [Davidson et al. 2007].

Sustained-attention lapses
Commonly called "attention lapses" [Weissman et al. 2006], these are not directly related to the level of arousal and can occur when alert, fatigued, or drowsy.

Diverted-attention lapses
When attention is diverted from the task at hand without a reduction in alertness.

Sleep "Nodding off" for more than 15 s. Sleep is considered distinct from microsleeps because of its duration and because responsiveness is recovered quickly after microsleeps [Jones et al. 2010].

Sleep can also be defined by EEG features. As sleep deepens, EEG activity moves to lower frequency bands, followed by the appearance of sleep spindles and K-complexes [Álvarez-Estévez et al. 2009].

Alertness A measure of how awake someone is [Knipling and Wierwille 1994], that is, the opposite of drowsiness [Berka et al. 2007, Johnson et al. 2011]. A necessary but not sufficient condition for performing a task.

Attention Whether a person is focusing on their assigned task rather than being distracted or absent-minded [Knipling and Wierwille 1994].

Sleepiness It is mostly agreed [e.g., Crummy et al. 2008, Shen et al. 2006, Vanlaar et al. 2008] that sleepiness is the propensity to fall asleep and can be reduced by getting sleep. Levels of sleepiness are affected by sleep deprivation, duration of wakefulness, sleep disorders, circadian rhythm, stimulants/depressants, adrenaline (excitement, fear), and external stimulation (visual, auditory, tactile, etc.).

Drowsiness Some consider drowsiness to be synonymous with sleepiness [Vanlaar et al. 2008], while others consider it a distinct concept. For instance, Shen et al. [2006] define drowsiness (or somnolence) as "the transitional state between wakefulness and sleep associated with a number of subjective feelings and symptoms, sometimes referred to as 'subjective sleepiness'."

Fatigue There are many different definitions of fatigue. Crummy et al. [2008] give a loose definition, calling it a broader term than sleepiness, possibly including physical and psychological effects as well as effects from sleepiness.

Shen et al. [2006] make distinctions between acute and chronic fatigue, physiological and psychological fatigue, and central and peripheral fatigue. Their overarching definition, however, is "an overwhelming sense of tiredness, lack of energy and a feeling of exhaustion, associated with impaired physical and/or cognitive functioning". This definition describes only the effect, making no claims about the cause.

Vanlaar et al. [2008] define fatigue as "a disinclination to continue performing the task at hand, caused by physical labour or repetitive and monotonous activities, such as monitoring a display screen or driving long distances". By this definition, fatigue is not necessarily associated with feeling tired.

May and Baldwin [2009] define sleep-related fatigue as well as active and passive task-related fatigue. Sleep-related fatigue matches the definition of sleepiness given above. Active task-related fatigue is caused by mental overload due to high-demand conditions such as busy traffic or poor visibility. Passive task-related fatigue is caused by mental under-load due to low-demand conditions such as driving on monotonous roads with little traffic.

The precise definition of terms such as drowsiness and fatigue is not required for this project. The effects of drowsiness, sleepiness, and fatigue—by any definition—are similar and detrimental to maintaining attention on a task. All kinds of lapses of responsiveness have the potential to cause accidents.

1.3 OBJECTIVES

A primary aim of the Christchurch Neurotechnology Research Programme (NeuroTech) is to produce a device that can detect lapses in real time in the real world. It is not feasible to complete this whole task within a single PhD project, so the aim of this project is to lay the groundwork for others to build on. Specifically, this project has the following objectives:

- Design and build biosignal acquisition hardware to monitor multiple signals relevant to lapse detection,
- Implement a software structure to enable experimenting with combinations of signal processing and classification algorithms,
- Start implementing some of these algorithms as proof of concept,

- Ensure the system is able to operate in real time in environments typical of real-world usage,
- Design the system to be generalizable to cognitive monitoring applications other than lapse detection.

Note that the project objectives refer to "lapse detection", which is broader than just the "microsleep detection" mentioned in the title. Microsleeps are a subset of lapses, and the ultimate aim of NeuroTech's lapse research programme is to detect as many lapses of responsiveness as possible. Each type of lapse has different underlying physiological and behavioural changes which will require different approaches to their detection. For the purposes of this project, though, as long as each type of lapse can be detected from the same raw signals, all types of lapses (and drowsiness too) can be lumped into one category. The system developed over the course of this project will acquire the raw signals required for lapse detection in general; other projects will investigate how to process those signals to detect each type of lapse.

1.4 STRUCTURE

Chapter 2 provides an overview of relevant literature about devices and techniques for detecting lapses and measuring drowsiness, as well as for other cognitive monitoring applications. Chapter 3 describes the system that has been designed to meet the objectives given above and the rationale for the design decisions. In Chapter 4 the implementation of some signal processing algorithms is presented to operate on data collected from the device. Chapters 5 and 6 describe a series of measurements and experiments designed to test the performance of the hardware, the software, and the system as a whole. A discussion of the outcomes of the project is presented in Chapter 7, along with suggestions for future work.

Chapter 2

BACKGROUND

The dangers of lapses and drowsiness are well recognized, and research into how to prevent, predict, and detect them has been going on for several decades. The following sections outline the current state of this research, first examining the signals that are used and then how those signals are processed. A brief overview of research into other cognitive monitoring applications is also presented.

2.1 SIGNALS FOR DETECTING MICROSLEEPS

Drowsiness and lapses of responsiveness are changes in cognitive state. In the absence of a direct "window into the mind" to measure cognitive state directly, we must rely on measurable signals that are correlated with lapses in order to detect them. These signals can be physiological measures that change with the underlying cognitive state, or behavioural measures that identify the resulting changes in performance. The most commonly used signals for identifying lapses and drowsiness are described below.

2.1.1 EEG

Electroencephalography (EEG) is the measurement of voltages on the scalp generated by ionic currents flowing in the brain's neurons. The firing of neurons generates circulating ionic currents and an associated electric field, and an array of electrodes distributed over the scalp samples the potential in space and time. If enough similarly-aligned neurons fire simultaneously, the voltage between pairs of electrodes will be of a measurable magnitude—the amplitude of normal alert EEG is usually in the range of 40–80 µV [Santamaria and Chiappa 1987].

As a signal that is generated directly by neurons, EEG is the most direct measure we have of brain activity, making it a logical choice when trying to measure changes in cognitive state.

2.1.1.1 Dry electrodes

Given its low amplitude (tens of microvolts), EEG is a difficult signal to acquire. Conventional wet EEG electrodes consist of a silver–silver chloride (Ag–AgCl) plate electrically coupled to the scalp through an electrolyte gel or paste. These electrodes must be placed by a trained technician who can check for correct placement and contact impedance. Also, the scalp must be abraded to remove any dead skin and ensure good electrical contact before recordings.

While the electrolyte solution helps to obtain a clean signal, it does have drawbacks. The solution evaporates over time, reducing its conductivity and leaving a crusty residue in the subject's hair. This limits the length of time for which EEG can be continuously monitored as well as subjects' receptiveness to the process. The electrolyte also requires an ongoing investment of time and money; keeping a continuous supply of solution costs money, and applying it to each electrode before every session takes time.

To eliminate these costs and make EEG suitable for a wider range of applications, several researchers have been investigating dry electrodes. The technologies they have been researching can be categorized as dry contact (with or without skin penetration), capacitive, and electro-optic.

Dry contact electrodes

Dry contact electrodes are much like conventional wet electrodes without the electrolyte gel. For example, Fonseca et al. [2007] developed an active electrode made from TiO_2-coated stainless steel. Each electrode has a unity-gain op-amp with very high input impedance. They found that the contact impedance drops substantially after a layer of perspiration builds up under the sensor surface. In fact, Searle and Kirkup [2000] found that movement artefacts for dry electrodes are less than those for wet electrodes after a layer of perspiration has formed, though they are substantially higher before this.

Achieving reliable contact with the scalp through hair can be very difficult. To address this, Tsai et al. [2009] developed a dry contact electrode with ten protruding arcs of Ag–AgCl that reach through the hair to contact the scalp. They mounted these electrodes around the bottom band of a cap as the basis for a drowsiness detection device. Their paper does not compare the performance of their electrodes to conventional ones but does say that they were able to detect an increase in low-frequency EEG power with drowsiness. Some of their subjects required the electrodes to be adjusted several times to get a good quality signal.

The g.SAHARA active dry electrodes from g.tec[1] have multiple gold-plated pins

[1] http://www.gtec.at

that reach through the hair to contact the scalp, as well as an amplifier built into the electrode [Guger et al. 2011]. Placing an amplifier with a high input impedance directly on the electrode itself makes the design more resistant to movement artefacts and tolerates a higher contact impedance. The electrodes are available with pins either 7 mm or 16 mm long to accommodate a range of hair styles.

Mindo[2] has developed two dry electrode designs. The first has multiple gold-plated pins that are individually sprung and are mounted on a thin flexible plate [Liao et al. 2011]. The springs and plate allow the electrode to conform to the user's head under pressure, making it less likely to be painful than those that have rigid pins. The second design consists of a block of conductive polymer foam wrapped in conductive fabric, making them comfortable and inexpensive [Liao et al. 2012]. This type of electrode is only suitable for use on non-hairy sites like the forehead. The impedance of both of these sensors stayed between 4 kΩ and 14 kΩ over the course of a 2 h recording. Data recorded using the pin-based electrode on the forehead had a correlation of 95.26 % with data from an adjacent wet electrode, and the same experiment with the foam electrode yielded a correlation of 95.56 %. With the pin-based electrode placed on a hairy site, POz, the correlation was 91.47 %.

Electrodes that use multiple metal pins to reach through the hair and push on the scalp can be uncomfortable for the user. Grozea et al. [2011] designed an electrode that uses silver-coated flexible bristles, somewhat reminiscent of a toothbrush. The bristles can reach through the hair, won't penetrate the skin, and distribute the pressure on the scalp "more uniformly and more flexibly" than pin-based electrodes. Using a large number of bristles provides a degree of redundancy to help maintain electrical contact. The authors note that the silver coating starts to wear off the bristles after several months of regular use. A typical skin–electrode impedance of 80 kΩ (at DC) was measured with these electrodes, and when compared to conventional wet electrodes the average coherence between a dry–wet pair was found to be over 80 % that of a wet–wet pair from 7–44 Hz.

Another interesting design which avoids the problems with metal pins is presented by Chi et al. [2013]. They have developed a spider-like active electrode made from a flexible plastic substrate with a conductive coating. As pressure is applied to the electrode, the legs flex outwards, pushing aside the hair to make contact with the scalp. This also makes the design safer than those that use metal pins since the electrode can squash flat under pressure. These electrodes have been shown to have comparable performance to conventional electrodes when measuring event-related potentials (ERPs) [Mullen et al. 2013].

[2] `http://mindo.com.tw`

Penetrating electrodes

The outer layer of the skin, the *stratum corneum*, consists of dry, dead cells with high electrical resistivity. The electrolyte solution and scalp abrasion used with conventional electrodes aims to bridge this layer in order to make contact with the more conductive epidermis. An alternative approach is to construct an electrode that can pierce the dry outer layer and directly contact the epidermis so that a current can be transduced without the need for conductive gel.

This is the approach taken by Silva [2006]. He developed an array of carbon nanotubes bound to a silicon substrate and tipped with silver chloride. This arrangement can penetrate the $\sim$13 µm stratum corneum and provides direct electrical contact to the epidermis. Unfortunately, the carbon nanotubes are weakly bonded to the substrate and can break off, with potential toxicological effects.

Chiou et al. [2006] developed a similar design using etched silicon spikes coated with titanium/platinum. They compared the signals from these electrodes to those from adjacent conventional electrodes and found them to be comparable. Unfortunately, electrodes of this type are limited to use on bare skin since the 200 µm spikes are not long enough to reach through the hair. This is a significant limitation since it means that the majority of the brain cannot be monitored.

Capacitive electrodes

Whereas dry contact electrodes make direct electrical contact with the scalp, capacitive sensors couple to the scalp without direct contact with the skin. Their input stage is a capacitor with one plate formed by the electrode and the other by the scalp. These electrodes work through hair since, by definition, capacitors require separation between the two halves. By the parallel-plate capacitor equation, the coupling capacitance is inversely proportional to the distance between the scalp and the electrode, and directly proportional to the relative permittivity of the intervening material. That is, if the distance between the scalp and the electrode changes over time or as the subject moves, the output signal will be distorted. Such movement artefacts could be minimized by good electrode design, but the relative permittivity is more difficult to control since it will change with levels of perspiration under the electrode [Searle and Kirkup 2000].

To avoid these problems, Chi and Cauwenberghs [2009] developed a non-contact electrode with active input-capacitance cancellation circuitry to make the electrode gain independent of coupling capacitance. Their paper indicates that the gain is indeed almost constant with varying separation. They later improved the design by developing a custom amplifier IC which minimized the problems caused by parasitic capacitance that occur when using discrete off-the-shelf components [Chi et al. 2012].

When tested on 10 subjects, the signals from these non-contact electrodes had a mean correlation of 0.80 with signals from adjacent wet electrodes. They were also able to achieve an information transfer rate of 19 bit/min on a steady-state visually-evoked potential (SSVEP) brain–computer interface (BCI) task using the non-contact electrodes.

Quasar[3] have developed a hybrid capacitive/resistive electrode for measuring EEG [Matthews et al. 2007]. These electrodes have an array of metal pins to reach through the hair, providing a combination of capacitive and high impedance resistive contact to the scalp. In a small study of five participants, Matthews et al. [2007] found a correlation of 0.99 between the power spectral density (PSD) of the dry electrodes and adjacent wet electrodes, though the correlation was lower in the time domain due to skin noise. An independent group [Wilson 2009] did a validation study of these electrodes and also concluded that they provide a signal comparable to conventional electrodes.

Electro-optic sensors

The Photrode, developed by SRICO,[4] is an electro-optic sensor that detects EEG using an optical device known as a Mach-Zehnder interferometer. These devices split a beam of light into two lithium niobate waveguides which pass between pairs of electrodes. The voltage picked up by the electrodes causes the refractive indices of the two waveguides to change slightly, thereby modulating the intensity of the light when it is recombined. The result is a sensor with an extremely high input impedance ($10^{14}\,\Omega$) capable of detecting microvolt signals. Kingsley et al. [2004] compare the Photrode to conventional EEG electrodes, though only by visual comparison of waveforms, and it is difficult to tell from their paper whether the design is as effective as they claim.

2.1.1.2 Commercial wireless EEG devices

A list of some of the commercially-available devices for wireless EEG acquisition is given below. This list does not include every device available for sale but covers a range of devices targeted at a range of markets.

- The Emotiv EPOC[5] is a 14-channel headset designed for gaming and BCI research. Its electrodes are felt pads which must be soaked in saline solution before each use. The device captures 14-bit data at 128 Hz and offers 12 h of battery life. The latest model includes a 9-axis inertial measurement unit.

[3] http://www.quasarusa.com
[4] http://www.srico.com/node/11
[5] http://emotiv.com

- The NeuroSky MindWave[6] is a single-channel headset designed for gaming. It has one dry contact electrode positioned on the forehead at Fp1. The device offers 6–8 h battery life.
- The Holst Centre and Imec[7] have developed an 8-channel headset based around their custom ultra-low-power application-specific integrated circuit (ASIC) for EEG acquisition [Patki et al. 2011]. Its electrodes are a dry Ag–AgCl design which also allow the addition of electrolyte paste if required. The device captures 11-bit data at up to 1024 Hz and offers 45 h of battery life. This system is targeted at health monitoring and BCI applications.
- The Quasar DSI 10/20[8] is a 20-channel headset using the hybrid resistive/capacitive electrodes described earlier in this chapter. The device captures 16-bit data at 240 Hz or 960 Hz and offers 24 h of battery life.
- The B-Alert X24 from Advanced Brain Monitoring[9] is a 20-channel headset aimed at research applications. No information is available about the electrodes used in this system, so presumably they are conventional wet electrodes. The device captures 16-bit data at 256 Hz and offers 6 h of battery life. It also includes a 3-axis accelerometer for monitoring head movement.
- The Mindo-64 Coral[10] is a 64-channel headset using the sprung-pin electrodes described earlier in this chapter. The device captures 12-bit data at 256 Hz and offers 23 h of battery life.
- The Cognionics HD-72[11] is a 64-channel headset using the spider-like electrodes described earlier in this chapter. The device captures 24-bit data at 500 Hz and offers 4 h of battery life. It also includes a 3-axis accelerometer and support for 8 auxiliary analogue channels for, e.g., ECG/EMG/respiration, etc.

As this list shows, there was a variety of wireless EEG acquisition systems available at the time of writing: from 1 to 64 channels, using wet or dry electrodes, for research and recreation. Some have been optimized for ease-of-use, some for low power, and some for data quality. It would have been interesting to compare the cost of each system, too, but prices were not published for most of the devices.

[6] `http://neurosky.com`
[7] `http://www.imec-nl.nl`
[8] `http://www.quasarusa.com`
[9] `http://www.b-alert.com`
[10] `http://www.mindo.com.tw`
[11] `http://www.cognionics.com`

2.1.2 Eye movement

Much like EEG, the eyes provide several useful cues regarding sleep/wake state. The eyes can be monitored with a video camera, mounted either remotely or on the subject's head, or by electrooculography.

2.1.2.1 Remote video of face

One technique for gathering data from the eyes is to use a camera mounted at a distance from the subject. This approach is obviously limited to situations where the subject's head is always facing the camera and cannot leave its field of view. One important such situation is driving, and consequently quite a lot of research has been done into using dash- or visor-mounted cameras to detect driver drowsiness.

The Driver State Sensor from Seeing Machines[12] has a camera mounted on the dashboard which monitors the driver's face. The system measures the position and orientation of the driver's head, their gaze direction, and eye closure. It initializes itself automatically, requiring no interaction from the driver. When the system detects a microsleep (from eye closure) or a lapse of attention (from gaze direction), it sounds an audible alert, vibrates the seat, and sends a notification to the vehicle operator's fleet management system. This system has been adopted in large mining trucks as well as transportation vehicles.

The Smart Eye AntiSleep[13] is a similar device, though targeted more at automotive research applications. It has a camera mounted on the dashboard, along with infrared (IR) illumination so that it is not affected by ambient light levels. The system is capable of tracking the driver's head within a volume of 40 cm $\times$ 40 cm $\times$ 30 cm. Initialization is automatic, requiring no interaction from the driver. The system tracks the driver's head orientation to within 3°, gaze direction to within 1.5°, and the degree of eye closure to within 1 mm.

2.1.2.2 Head-mounted video of eyes

Another technique for gathering data from the eyes is to use a camera mounted on the subject's head. By using a camera in a fixed position relative to the eyes, the face localization step required when processing remote-mounted video is unnecessary. This reduces the computational requirements, thus reducing latency and/or hardware requirements. It also has the added benefit of guaranteeing that the eyes can never leave the camera's field of view.

Remote-mounted camera systems must balance the competing requirements of the subject's range of motion and the apparent size of the face region; if the field of

[12] `http://www.seeingmachines.com`
[13] `http://www.smarteye.se`

view is enlarged to allow tracking the subject over a wider area, the size of the subject's face in the image is reduced. For a driver monitoring system, the eye openings are typically 10–15 pixels high [Hammoud and Zhang 2008, Malla et al. 2010] which limits the accuracy of eye measurement algorithms. Mounting the camera on the subject's head provides a much higher resolution image of the eyes, even if a lower resolution camera is used. Higher resolution images enable more accurate and sophisticated measurements.

On the other hand, head-mounted camera systems require the subject to take the time to put on and remove a device and to put up with any inconvenience or discomfort it may cause them. The device requires either a power cable connected to a base unit or regular battery charging. These factors may make head-mounted devices more suitable for use on the job, where they could be considered part of standard safety equipment, rather than for the general public.

Nishiyama et al. [2007] developed a head-mounted device capable of recording images of both eyes and used it to test subjects in a driving simulator. They confirmed prior research that large, low-frequency fluctuations in pupil diameter are present when the subject feels sleepy [Lüdtke et al. 1998] but, more importantly, they found patterns present before this phenomenon. In 80 % of their subjects, a gradual decrease in pupil diameter occurred over the minutes preceding the large fluctuations. Subjects did not feel sleepy during this period, suggesting that the gradual decrease could be an effective predictor of drowsiness. However, 3 out of 14 subjects displayed the gradual decrease without becoming sleepy afterwards.

Instead of using a video camera to monitor the eyes, Johns et al. [2007] use infrared reflectance oculography. Their device, the Optalert Eagle,[14] consists of a glasses frame with an IR LED and phototransistor mounted below the eyes. This device samples at 500 Hz—much faster than a video camera—which is possible because it only measures a single value; a one-pixel image, as it were. This very high sampling rate allows the velocity of eyelid movements during blinks to be determined accurately. The glasses are connected to an off-body unit, either by a cable or wirelessly, which provides visual feedback of the user's drowsiness level.

Outside of the field of lapse and drowsiness monitoring, there are several commercial head-mounted eye-tracking systems available. These are often used for gaze tracking in psychological research into the way people interact with advertising, user interfaces, and each other.

SensoMotoric Instruments (SMI)[15] has created a pair of eye tracking glasses[16]—a head-mounted, video-based, binocular gaze-tracking system. The device looks like an over-sized pair of ordinary glasses with small cameras mounted in the bottom edge of

[14] `http://www.optalert.com`
[15] `http://www.smivision.com`
[16] `http://www.eyetracking-glasses.com`

the rims facing the eyes and a scene camera mounted in the bridge. It is available in both wired and wireless configurations, although it appears that the wireless version still has a cable between the glasses and an off-head unit. The system provides 60 Hz binocular gaze tracking with no manual calibration required. SMI have recently integrated these glasses with the Emotiv EPOC EEG headset to provide unified access to eye tracking and EEG data with a common time base. This combination comes close to meeting the objectives of this project—it offers real-time EEG and eye video from a wearable device—however it is only suitable for research applications. It requires the user to put on two devices, the glasses and the headset, and to apply saline solution to the EEG electrodes before each use.

The Tobii[17] eye tracking glasses are a similar product, providing head-mounted, video-based, binocular gaze tracking. Unusually, these glasses have four eye-tracking cameras mounted very close to the face, as opposed to the more common design of a single camera per eye. Perhaps using multiple cameras allows them to compensate for the reduced field of view which must result from mounting the cameras so close to the eyes. Like the SMI glasses, this product has a cable connected to an off-head unit which provides the data recording functionality and wireless interface. The system provides 30 Hz gaze tracking and requires a one-point calibration routine. The battery provides 2 h of recording time.

Many other commercial head-mounted eye trackers (e.g., Polhemus VisionTrak,[18] Applied Science Laboratories H7,[19] SR Research EyeLink II[20]) are bulky units with large cameras and partial mirrors mounted in front of the user's eyes. These are clearly limited to laboratory research settings.

Babcock and Pelz [2004] describe how to build a simple monocular eye tracking system from off-the-shelf components. The resulting device is much more bulky than the commercial devices described above but it demonstrates that it is feasible to create such a device from readily available parts. There have been dramatic advances in miniature camera and mobile multimedia-processing technologies since 2004, so even better results should be possible today.

2.1.2.3 Electrooculography

Electrooculography (EOG) is a common technique for monitoring eye movements in a clinical or laboratory setting. The eye forms an electric dipole, the orientation of which can be detected by placing electrodes near the eyes. The application of EOG to drowsiness/lapse detection is generally limited to research settings because it requires electrodes to be attached to the subject's face. Golz et al. [2007] used EOG as part of

[17] `http://www.tobii.com`
[18] `http://www.polhemus.com`
[19] `http://www.asleyetracking.com`
[20] `http://www.sr-research.com`

an automated multi-input system for detecting microsleep events and found vertical EOG to be particularly useful. Conversely, Peiris et al. [2005] found that EOG was of little value to expert raters identifying lapses during a psychomotor vigilance task (PVT). Whether it is useful for detecting lapses or not, the intrusive requirement of fixing electrodes near the subject's eyes limits its suitability for a commercial device. However, it is possible to use independent component analysis (ICA) on the frontal EEG channels to extract much of the EOG signal without the need for electrodes on the face.

2.1.3 Head position/movement

Useful information can be obtained from the orientation and motion of the head in scenarios where the head position is usually fixed. For instance, the head is usually relatively still and facing straight ahead when driving. If the head is detected to be facing down or to the side for some time, this could indicate that the driver is distracted, i.e., having a lapse of task-oriented attention, or is having a long BM. Popieul et al. [2003] add that the head also tends to droop as muscle tone reduces with drowsiness, and the variability in position and speed increases as the subject yawns and moves around to get comfortable. They measured head position using a magnetic head tracker as well as driver performance indicators such as speed, lateral lane position, and steering wheel angle in a driving simulator. They found that the variability in head movements was as useful as driver performance indicators for detecting drowsiness and was less influenced by inter-individual differences and the road environment. While this variability is useful for detecting drowsiness, it does not detect lapse events. As Hartley et al. [2000] state, using head-nod events alone to detect BMs is of limited use since the subject's level of performance has probably decreased markedly by the time their head nods. Nevertheless, head position and movement measures may be useful as part of a multi-modal detection system.

2.1.4 Other physiological signals

Tsuchida et al. [2009] used electrocardiography (ECG) to measure heart rate variability as an indicator of autonomic nerve activity. Feeding this into a multi-class classifier along with eyelid closure, they were able to achieve a drowsiness level classification accuracy of 89 % with a resolution of 10 s. However, they gave no indication of how much heart rate variability contributed to their classifier's accuracy, so we can not tell if it was any improvement over eye closure alone. Additionally, conventional ECG electrodes would be far too intrusive for use in a real-world device, though some advances have been made towards non-invasive signal acquisition through clothes [Leonhardt and Aleksandrowicz 2008]. However, very few researchers have found

compelling reasons to incorporate ECG in drowsiness detection systems over the many years of research in the field.

Kar et al. [2011] measured blood biochemical parameters as part of an EEG study into driver fatigue. Subjects were kept awake for 36 h and completed regular simulated driving tasks. Blood samples were taken five times over the course of the experiment and analysed for glucose, urea, and creatinine. They found a strong correlation between EEG changes and the blood parameters at successive stages of the experiment, suggesting that these blood parameters could indicate fatigue. Although this investigation used invasive methods, it may eventually be possible to use lab-on-a-chip technology to measure blood parameters continuously.

2.1.5 Driver-specific signals

Driving a vehicle is a common and fairly well characterized task: a driver sits down, makes minimal movements, and usually looks straight ahead. Their task is to maintain appropriate vehicle speed and lane position. These clear requirements mean that an appropriately instrumented vehicle may be able to detect changes in performance stemming from driver drowsiness. Knipling and Wierwille [1994] suggested that useful parameters could include corrective steering movements, lateral acceleration, lateral lane position (measured using computer vision), and yaw rate. These parameters are widely used to detect drowsiness in driving simulators where they can be calculated exactly from the computer-generated environment [e.g. Chiou et al. 2006, Popieul et al. 2003, Tsuchida et al. 2009] but measuring them in the real world is much more difficult.

Several vehicle manufacturers are starting to include advanced driver assistance systems such as lane departure warnings and forward collision warnings. These systems, such as those developed by Mobileye,[21] visually identify the boundaries of the current lane and can apply corrective force to the steering wheel to maintain lane position unless the driver is indicating their intention to change lanes. These features will surely help to reduce on-road fatalities due to lapses of responsiveness. This class of systems detects driver errors as they occur—e.g., beginning to drift out of a lane—but before they become too severe or cause an accident. A lapse detection system based on physiological signals, by contrast, aims to detect the change in cognitive state that caused the behavioural change that caused the change in vehicle dynamics, and therefore to respond earlier than a vehicle-based system possibly could.

Park [2011] proposed driver grip force as a measurable indicator of driver fatigue. They inferred grip force by measuring the bending radiation loss in a plastic optical fibre wound around a steering wheel. While they showed that optical power decreases as a function of the grip force, they gave no evidence to support the claim that driver

[21] `http://www.mobileye.com`

grip strength is correlated with fatigue. Many drivers barely grip the steering wheel at all, making variations in grip force a questionable measure at best.

2.2 SIGNAL ANALYSIS TECHNIQUES

The following sections describe some common techniques for analysing these signals in order to measure drowsiness and detect lapses of responsiveness.

2.2.1 EEG

2.2.1.1 Measuring drowsiness

Observation of EEG has led to activity being divided into five frequency bands (Figure 2.1) [Papadelis et al. 2007] that help to explain common features. Changes in the amplitude of activity in these bands are known to be correlated with, among many other things, drowsiness and sleepiness. Santamaria and Chiappa [1987] describe the following characteristic changes: an increase in alpha during restful wakefulness with eyes closed, followed by a disappearance of alpha as drowsiness deepens; a decrease in amplitude by approximately half; and a reduction in frequency of 3–6 Hz ($\alpha \rightarrow \theta$). They also noted an increase in synchrony between channels as drowsiness increases. These patterns vary significantly between individuals, and even between sleep/drowsiness episodes of the same individual, making it difficult to reliably determine the level of drowsiness from EEG alone.

Rather than trying to develop a single model of EEG activity, Johnson et al. [2011] suggest that there may be three or more "phenotypes" which people's patterns of drowsiness fall into. For instance, some people are barely affected by sleep deprivation, while others become drowsy quickly even when normally rested. Developing models for each of these groups could provide the advantages of having better results than a single model while not requiring the training of individualized models.

To avoid the need to train a classifier on either individuals or large populations, Pal et al. [2008] developed an unsupervised adaptive approach. Their system establishes a baseline alert model by measuring several minutes of EEG when the subject starts driving. It is assumed that they remain alert during this period. When testing subjects in a driving simulator, they observed that deviations in EEG power from the

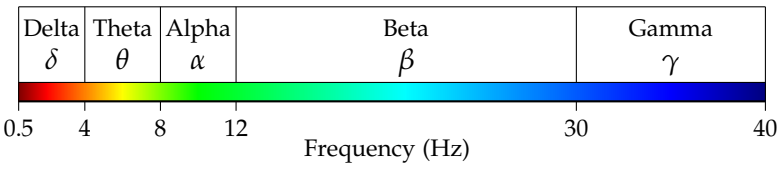


Figure 2.1 Standard EEG frequency bands.

alert baseline measurement corresponded to changes in driving performance. This relationship can be used to detect changes in the subject's alertness. Note that this approach does not actually estimate the level of drowsiness (decreased alertness), but rather *changes* in alertness. For instance, a change could be detected if the subject has an excited conversation with a passenger. Pal et al. claim that it is valid to give a warning in such situations because the driver is distracted from the task of driving.

Golz et al. [2007] compared the periodogram, delayed vector variance (DVV), and a combination of the two for extracting features from EEG. The periodogram is an estimator of the true PSD and assumes linearity and stationarity of the signal. DVV, on the other hand, estimates to what extent the signal has non-linear or stochastic components. The features obtained by these techniques were fed into neural networks (Section 2.2.3) for classification. They found that the periodogram was more effective than DVV for extracting features, and even adding DVV to the periodogram provided only minor improvement.

2.2.1.2 Detecting lapses & microsleeps

Compared to the number of drowsiness estimation approaches in the literature, of which the previous section contains only a small sample, there have been comparatively few attempts at detecting lapses of responsiveness.

Peiris et al. [2011] developed a detector based on spectral power features aimed at detecting lapses with second-scale resolution. This method has comparable performance to the recurrent neural network implementation from their earlier paper [Davidson et al. 2007], indicating that non-linear techniques may not provide any more useful information than simpler linear ones. This, along with the results of Golz et al. [2007] above, seems to imply that traditional linear techniques may be sufficient for extracting features from EEG. It should be noted, however, that their method cannot be applied in real time since the entire EEG feature vector is required for a normalization step. They suggest that it may be possible to remove this limitation by using mean values from previous sessions but that approach has not been tested.

The alertness classification model developed by Berka et al. [2004] uses discriminant function analysis based on a database from a large population but is individualized by taking several baseline readings. This algorithm is used by the B-Alert device mentioned in Section 2.1.1.2. The system classifies each second of EEG into one of four states of alertness: high vigilance, low vigilance, relaxed wakefulness, and sleepy. Berka et al. [2004] claim that the system is capable of identifying second-by-second lapses of alertness that are predictive of errors in a vigilance task. Unfortunately the headset uses wet electrodes, limiting its usefulness outside of a laboratory setting.

2.2.2 Eye video

2.2.2.1 Remote video of face

Wierwille and Ellsworth [1994] defined a set of criteria for manually rating driver drowsiness from video of their face. These criteria consider features such as facial tone, speed of blinks and eye closure, and gestures like rubbing the eyes. Despite rather subjective descriptions (e.g., "normal facial tone"), they found that ratings both between and within raters tend to be consistent and that the ratings consistently discriminate between conditions. A manual approach is not suitable for widespread usage but is useful as a gold standard when evaluating other drowsiness detection techniques in a research setting.

The computer vision technique developed by Malla et al. [2010] automatically measures eye closure in real-time. It uses a cascade of detectors to narrow down the search area for more efficient computation. They begin with a Haar classifier for face detection [Lienhart and Maydt 2002, Viola and Jones 2004] and smooth the result with a Kalman filter. Typical anthropometric proportions are used to place regions of interest around the eyes, and then weighted template matching is applied in these regions to locate the centres of the eyes. The local extrema of the vertical gradient in these regions mark the upper and lower eyelids, from which the eye closure can be measured relative to a known eyes-open condition. This approach had limited success since the cascaded nature of the algorithm compounds the errors in each stage. They were only able to discriminate between three degrees of eye closure, and unreliably at that.

Sigari [2009] developed a similar technique but without an explicit eye-detection stage. After using a Haar classifier to locate the face, he calculates the horizontal projection of the top half of the face image. After 1–2 min of training, the state of the eyes can be calculated by correlating the current projection with the known projection of open eyes. The horizontal projection of open eyes is constantly updated using a fuzzy running average. This adaptive method is reportedly robust to skin color, glasses, and illumination, though no quantitative measures of this robustness are provided. The system provides no external illumination, so one would expect it to perform poorly in low-light conditions.

Grace [2001] takes an entirely different approach, exploiting the so-called red-eye effect. The pupils of human eyes have a distinct glow when lit with IR light at a wavelength of 850 nm but not at 950 nm. By synchronizing two IR sources with a video camera it is possible to obtain frames with alternating light and dark pupils. Subtracting a frame with dark pupils from the previous one (with light pupils) leaves an image with only the pupils visible. The proportion of time that the eyes are closed can be calculated by tracking the visibility of the pupils over time. This technique

completely relies on the red-eye effect, which may be interfered with if there are high levels of ambient IR or the driver is wearing glasses that block infrared.

All of the above eye-based techniques measure, among other things, percentage eye closure (PERCLOS), the percentage of time in a minute that the eyes are at least 80 % closed [Wierwille and Ellsworth 1994]. This measure has been deemed the best predictor of PVT lapses[22] by the US Department of Transportation Federal Highway Administration and National Highway Traffic Safety Administration. Dinges et al. [1998] compared PERCLOS to two EEG algorithms, two eye blink monitors, and a head position monitor. They found that PERCLOS was the only one to correlate highly with PVT lapses both within and between subjects. However, PERCLOS has a temporal resolution of 1 min at best, and that is sometimes increased up to 20 min [Dinges et al. 1998]. This is fine for estimating drowsiness but means it cannot detect BMs as they occur.

In summary, measuring oculometrics with a remote-mounted camera can only be done in situations where the subject is primarily stationary. Complex computation is needed to locate and track the face and eyes, and the output of the system tends to be low resolution. However, a big advantage of such systems is that they are completely non-invasive, requiring no interaction from the subject.

2.2.2.2 Head-mounted video of eyes

Hirata et al. [2009] used a head-mounted eye-video system to measure changes in reflexive eye movements. The vestibulo-ocular reflex (VOR) counter-rotates the eyes to maintain gaze direction during head movement. The performance of this reflex can be measured by comparing head movements to the actual compensatory eye movements. The movement of the eyes is measured by tracking the pupil positions in the eye images, and head movement is detected in six degrees of freedom by accelerometers and gyroscopes. They found that the VOR gain decreases and standard deviation increases before the subject feels drowsy, so these two measures can be used as predictors of drowsiness.

Li et al. [2005] developed a "starburst" algorithm for locating the pupil in video of the eye captured from the Babcock and Pelz [2004] head-mounted eye tracker (Section 2.1.2.2). This process sends out rays from a starting point which stop when the image gradient exceeds a threshold. More rays are then sent back from these end points in the opposite direction, again stopping when the gradient exceeds a threshold. These end points define a set of candidate features for the pupil boundary. An ellipse is then fitted to the feature points using a random sample consensus (RANSAC)

[22] The psychomotor vigilance task (PVT) is commonly used for detecting the effects of drowsiness/fatigue/sleepiness. The subject must respond to a series of visual stimuli presented at random intervals of 2–10 s. A lapse is defined as a response time of more than 500 ms [Drummond et al. 2005].

[Fischler and Bolles 1981], a shape-fitting algorithm which is robust to the presence of outliers (Section 4.1.2).

Świrski et al. [2012] developed a pupil localization algorithm inspired by "starburst" which takes into account the difficulties arising from using video taken at a very oblique angle to the eye. After identifying a region of interest around the pupil using Haar-like features, they use *k*-means clustering on the image histogram to segment the dark pupil from the lighter iris. A Canny edge detector is then applied to locate the edges of the segmented region and a modified version of RANSAC is used to fit an ellipse to the edge points. This algorithm was able to locate the pupil within 5 px of a manually annotated gold standard in 87 % of the 600 frames in their test data set. By contrast, "starburst" achieved 15 % on the same data set.

The Optalert device described in Section 2.1.2.2 infers the degree of eye closure by shining an IR LED at the eye and measuring the reflectance [Johns et al. 2007]. They have developed a drowsiness metric based on the ratio of the amplitude to the velocity of eyelid movements during blinks and eye closure. This ratio is apparently consistent between alert subjects and changes with drowsiness. When tested on sleep-deprived subjects in a driving simulator, the device has been shown to predict that the vehicle will completely leave the road within the next 15 min with sensitivity 83.3 % and specificity 60.9 %. No figures are given for non-sleep-deprived subjects, however, so we cannot know the false alarm rate under normal conditions or how often non-sleep-deprived people have driving errors.

2.2.3 Multi-modal techniques

Several research groups have come to the conclusion that a single signal is not sufficient to identify drowsiness or lapses reliably [Álvarez-Estévez et al. 2009, Golz et al. 2007, Hussain et al. 2008, Kaefer et al. 2003]. Golz et al. [2007] developed a multi-modal classifier for detecting microsleeps from seven channels of EEG, horizontal and vertical EOG, pupil size, and eye-gaze coordinates. Features are extracted from these signals, and a support vector machine is used to decide whether the fused features represent a microsleep event. Using all of these features, test errors (a measure of the classifier's generalizability) were reduced to 9 %. They found that vertical EOG is more useful than horizontal, pupil diameter is more useful than gaze direction, and that Cz is the most useful EEG channel. This group has made the most promising advances towards detecting microsleeps using multiple signal sources.

From the related field of sleep studies, Álvarez-Estévez et al. [2009] developed a fuzzy-logic approach to detect sleep stages from polysomnograms (EEG + ECG + EOG + EMG). The usual method of quantifying sleep stages uses six discrete stages ranging from awake, through S1–4, to REM sleep. To calculate a more useful measure than discrete stages, Álvarez-Estévez et al. used fuzzy logic to determine the degree of

membership of each stage and so obtain a continuous measure of sleep stage. While their work is not directly applicable to identifying lapses, the general approach of using fuzzy rules to classify states may be.

2.3 NEUROTECH'S LAPSE RESEARCH PROGRAMME

To put this project in context, it is useful to review the prior research undertaken as part of NeuroTech's lapse research programme. Over the last eight years the group has carried out three studies to investigate the nature and causes of lapses of responsiveness. Using the data from these studies, a range of techniques to automatically identify lapses from physiological signals have been, and continue to be, developed.

2.3.1 Study 1: EEG

The first study involved 15 participants completing two 1-h sessions of a one-dimensional tracking task. They were seated in front of a computer monitor with a sinusoidal target scrolling down the screen, and used a steering wheel to keep an arrow on top of the target. 16 channels of EEG were recorded, along with video of the subject's face from a remote camera. Subjects were normally rested for both sessions. The gold standard for identifying lapses was generated by manually examining the tracking response and video of the face. A lapse was defined as either an inappropriate flat spot in tracking response (disregarding periods where the target velocity was near zero) or prolonged eye closure.

Peiris et al. [2006] found that 14 of the 15 subjects had at least one lapse, with an overall rate of $39.3 \pm 1.9\,\mathrm{h}^{-1}$ (mean $\pm$ SE) and duration of $3.4 \pm 0.5\,\mathrm{s}$. During lapses, they found an increase in EEG power in the delta, theta, and alpha spectral bands, and a decrease in the beta and gamma bands.

Davidson et al. [2007] developed a long short-term memory (LSTM) recurrent neural network to classify lapses from the log-power spectrum of the EEG with 1 s resolution. Using only two bipolar channels, they were able to achieve similar performance to using all 16 channels—an area under the curve of the receiver operating characteristic (AUC-ROC) of 0.84 ± 0.02. This was the first application of a LSTM to EEG analysis and achieved performance equivalent to the best tapped delay-line multilayer perceptron (TDL-MLP) they compared it to.

Peiris et al. [2008] used linear discriminant analysis (LDA) to develop an individual classifier for each of eight subjects and then used stacked generalization to combine these into an overall detection model. When this overall classifier was run on the concatenated set of data from all eight subjects, it achieved a sensitivity $= 73.5\,\%$, selectivity $= 25.5\,\%$, and accuracy $= 61.2\,\%$. As above, this classifier had a resolution

of 1 s. They later [Peiris et al. 2011] developed a selection of linear and non-linear lapse detectors based on the power spectrum, fractal dimension, approximate entropy, and Lempel-Ziv complexity of the EEG. The best performance was achieved using the detector based on power spectral features, achieving similar performance to the LSTM RNN of Davidson et al. [2007]. This indicates that simpler linear techniques may be sufficient for identifying lapses of responsiveness from EEG.

2.3.2 Study 2: EEG + fMRI

The second study involved 20 participants completing a 50-min two-dimensional tracking task [Poudel et al. 2008] in a 3 T MRI scanner. This study collected functional MRI (fMRI) data, eye video, vertical EOG, and 64 channels of EEG in addition to the tracking data. The participants kept sleep diaries and wore wrist actigraphs for the six days preceding the MRI session to confirm that they were normally rested. The gold standard was defined by manually rating the tracking response and eye video, with BMs defined as poor tracking with full or partial eye closure lasting less than 15 s.

16 of the 20 subjects had BMs during the session, with 14 having more than $43\,\mathrm{h}^{-1}$ [Innes et al. 2010]. 8 subjects had sleep episodes lasting ≥ 15 s. The overall rate of BMs was high but variable, $79.1 \pm 66.2\,\mathrm{h}^{-1}$ (range 0–$225.6\,\mathrm{h}^{-1}$), with each one lasting 3.3 ± 1.6 s. They found a moderate correlation between the number and duration of BMs and self-rated sleep propensity. There was no correlation between the number of BMs and circadian type or the quality, duration, or efficiency of sleep.

In the fMRI data, Poudel et al. [2009, 2014] found a decrease in blood-oxygen-level dependent (BOLD) activity in the thalamus (related to arousal) and the posterior cingulate gyrus and medial frontal cortex (part of the default-mode network which deactivates during the transition to sleep). They also found an unexpected increase in activity in sensory-motor areas, possibly related to some form of compensation mechanism.

Also from this study, Poudel et al. [2010] found that posterior theta-band activity in the EEG is correlated with the occurrence of behavioural microsleeps.

2.3.3 Study 3: EEG + fMRI with sleep restriction

The third study had a similar structure to the second, with the primary difference being that participants completed two sessions: one after a normal night's sleep and one after a restricted sleep period of 4 h. Again, 20 participants completed a 2 D tracking task in an MRI scanner, though this time the protocol included an arterial spin labelling (ASL) sequence.

This study found that participants who were drowsy after the sleep-restricted night had decreased cerebral blood flow (CBF) in both arousal-promoting and at-

tentional brain regions. On the other hand, participants who were not drowsy after sleep restriction were able to maintain the level of activity in attentional regions and increase activity in arousal-promoting regions [Poudel et al. 2012].

After sleep restriction, participants exhibited a decline in tracking performance as a function of time-on-task [Poudel et al. 2013]. Sleep restriction was also related to a time-on-task decrease in BOLD activity in task-related brain regions. Transient tracking errors, though, were associated with BOLD activations in different regions which did not vary with time-on-task, indicating that there are distinct processes governing short- and long-term changes in alertness after sleep restriction.

By comparing tracking performance to actigraphy data and sleep questionnaires, Innes et al. [2013] found that people with regular sleeping patterns are more prone to microsleeps following sleep restriction. Those that went to sleep at a similar time each day, fell asleep quickly, and had few arousals had a greater number of BMs during the sleep-restricted session. Interestingly, they found no correlation between performance on a 10-min PVT task completed directly before each session and the number of microsleeps during the tracking task.

2.3.4 Current and future research

Most current work in the lapse research programme is focused on developing signal processing and classification techniques to identify lapses of responsiveness from physiological signals. Jonmohamadi et al. [2014] developed a technique called source-space independent component analysis (ICA) to locate the source of EEG activity deep within the brain. It performs better than the more common sensor-space ICA because it allows weak signals to be reconstructed, even in the presence of stronger ones.

LaRocco et al. [2014] have designed a feature-reduction technique to select the best spectral features to pass into a classifier. Using this technique to select the single best feature, they were able to create a classifier with a phi correlation of $\varphi = 0.94$ when tested on an artificial data set (2-s periods of a 15 Hz sinusoid overlaid on EEG with a signal-to-noise ratio of 0.3). By contrast, principal component analysis (PCA) achieved $\varphi = 0.00$ on the same data. Ayyagari et al. [2014] have developed an echo-state network (ESN) with a leaky-integrator neuron structure which achieved $\varphi = 0.92$ on the same artificial data set. Further testing is being carried out on real data sets.

It is in this context that the current project is being carried out. Prior research has revealed useful information about the changes that occur in the brain during lapses and progress is being made towards developing automated lapse detection techniques. This project aims to capitalize on the earlier research in order to assist future research, using the understanding of lapses that has been gained to inform the design of hardware and software for lapse detection.

2.4 ALERTING TECHNIQUES

Of course, a device that is capable of detecting lapses in real-time is of little use unless something is done with the information. The device's output needs to either trigger an alert to rouse the subject or intervene in what they are controlling in some way. Several methods of providing alerts have been proposed, including: an audible tone, beep, or buzz, recorded voice messages, a visual gauge of drowsiness, a peppermint scent, vibrations, and temperature changes [Dinges et al. 1998, Grace 2001]. These alerts must be presented in a way that does not startle the user. Alternatively, or in addition, the device could intervene in the task the operator is meant to be performing—bringing a train to a stop, for instance.

Jung et al. [2010] explored the effect of auditory lane-departure warnings on subjects in a driving simulator. The simulated vehicle was subjected to random drifts away from the centre of the lane which the participant was required to correct. In half of the cases in which a participant's reaction time to these events was much longer than an alert baseline response time, the system sounded an audible alert (1750 Hz tone, $\sim$68.5 dB). They found that the reaction time for events immediately following those in which an alert was given were significantly shorter than those for which no alert was given. They also found that EEG power in the alpha and theta bands was closer to the baseline for the events following an alert, indicating that audible alerts are effective at improving responsiveness. This theta-band power suppression lasted for at least 35 s following the audible alert [Lin et al. 2013].

2.5 OTHER COGNITIVE MONITORING APPLICATIONS

In recent years, the number of applications of cognitive monitoring technologies has increased dramatically, driven by the availability of small, low-power, low-cost, biosignal sensors, and advances in signal processing and neuroscience. The term "cognitive state" is a broad term which can be used to refer to almost any aspect of a person's mental state: attention, arousal, alertness, anxiety, drowsiness, fatigue, vigilance, workload, boredom, excitement, etc. The applications below each provide an estimate of one or more of these aspects of the user's cognitive state.

"Augmented cognition" is a form of adaptive automation in which information about the user's cognitive state is used to modify their interaction with a computerized system in real time [Stanney et al. 2009]. The inputs to such a system are taken from one or more sensors—usually a selection of EEG, ECG, EOG, functional near-infrared (fNIR), galvanic skin response, and video oculometry. After processing these signals to determine the user's cognitive state, the information is used to adapt the task that the user is completing with the aim of maximizing their performance. Depending on the situation, this adaptation could involve modifying the level of task automation

or altering the quantity, sequence, timing, or form of the information presented to the user. For example, during periods of high workload some information could be presented as auditory cues instead of visual, or low-priority information could be delayed until the workload decreases. Of the 17 implementations of augmented cognition technology listed by Reeves et al. [2007], the three main areas of application are military command and control, dismounted soldier operations, and vehicle control.

In a slightly different application of the same technology, Stevens et al. [2006] used EEG to quantify the engagement, distraction, and mental workload of students as they solved scientific problems. Neural feedback allows the investigators to track students as they progress through the stages of learning with more accuracy than can be achieved by merely tracking their performance on the task. For example, this information can be used to differentiate between students that perform well as a result of high mental effort and those that perform well with little mental effort. Using this information, the novelty and difficulty of problem solving tasks could be optimized to improve the students' learning outcomes.

Passive brain–computer interfaces (pBCIs) are a related concept to augmented cognition but are not necessarily targeted at improving the user's performance on a task. A typical active BCI such as the P300 speller [Donchin et al. 2000] acts as a user's primary input to a computer and requires the user to actively interact with the device by trying to control their brain activity or responding to a stimulus. By contrast, a passive BCI is intended to complement another input device such as a keyboard or an active BCI. It provides a secondary stream of information by monitoring the user's cognitive state while they interact with the primary input device. This difference also reflects a shift in the target user base from the physically disabled to the able-bodied. Bos et al. [2010] give the example of changing the story-line, presentation, or difficulty of a computer game based on the player's emotional state or mental workload [Saari et al. 2009]. A passive BCI could also enhance interaction with an active BCI by using error-related potentials to detect and correct mistakes [Dal Seno et al. 2010].

In the area of workload monitoring, Borghini et al. [2014] present a review of research into the effect of mental workload, fatigue, and drowsiness on the EEG and task performance of car drivers and airplane pilots. They found that the accuracy of the best currently-available techniques for classifying the cognitive state of drivers and pilots is about 90 %. However, these techniques operate offline and "no device or convincing algorithm has been published or practically applied for a robust online recognition of such mental states to date". Nevertheless, this is an active area of research that will no doubt see advances in the coming years. An interesting suggestion that they raise for future research in this field is monitoring the *collective* cognitive state of an aircraft crew so that human–machine interactions can be optimized over all of the available crew members.

Marshall [2007] describes an index of cognitive activity that uses wavelet analysis to identify small, rapid increases in pupil diameter that occur during periods of high cognitive workload. This index is not affected by changes in lighting conditions; the pupil's response to increased light is rapid constriction, and to decreased light, slow dilation. The index considers only small, rapid increases in diameter, so is unaffected by the pupil's light response. Using this index, along with several others derived from pupil measurements, they were able to develop classifiers that could distinguish a relaxed state from an engaged state, distracted from focused driving, and fatigue from alertness. The accuracy of the three classifiers ranged from 69 % to 92 %. This work is noteworthy because it uses only eye video from head-mounted cameras as input—the majority of other techniques rely, at least in part, on EEG. The SMI eye tracking glasses mentioned in Section 2.1.2.2 include a software module to calculate Marshall's index of cognitive activity.

The fields of augmented cognition and passive BCI are still in their infancy and undergoing active research. There is an overlap between the signals that are used in these fields and those used for detecting lapses. It could be useful, therefore, to take these other applications into account when designing a device for lapse and drowsiness detection. If supported by careful software design, the hardware developed for lapse detection could be used by other research groups for other cognitive monitoring applications.

2.6 SUMMARY

Detecting lapses of responsiveness is difficult. There is no one signal which is a reliable indicator of lapse events, so we will use a combination of signals in an attempt to achieve an acceptable level of accuracy. The two physiological signals used most often in prior research are EEG and video of the eyes, while behavioural signals can be used in specific situations such as driving. There are other areas of research which use these same signals for other purposes, for example, to augment a human–computer interface. If we are going to the effort of developing hardware for lapse detection, it would be sensible to design it to be applicable to these other areas as well.

Chapter 3

THE ELAPSE PLATFORM

In light of the motivation outlined in Chapter 1 and the research presented in Chapter 2, we have developed a platform for cognitive monitoring, with a particular focus on lapse detection. The *Elapse* platform consists of a wearable device which measures multiple physiological signals, and a software framework for analysing these signals to classify the user's cognitive state.

The focus of this project was not to create a system that is itself capable of detecting lapses (or any other cognitive state/events), but rather to develop the infrastructure to enable others to do so. The aim was to produce a system that abstracts away the low-level details and common functionality, allowing other researchers to concentrate on the unique aspects of their application.

This chapter describes the development of the Elapse platform: the requirements for each subsystem, the design decisions and rationale, and details of the implementation.

3.1 SYSTEM DESIGN OVERVIEW

At a high level, the Elapse platform has a two-part architecture: a head-mounted device which acquires and transmits biosignals, and a software framework running on a remote computer which receives and analyses these signals. On top of the software framework are user-supplied plug-ins which implement the actual signal processing required for an application such as lapse detection. This structure is illustrated in Figure 3.1.

The head-mounted device has three sensing modalities: *i)* EEG to measure brain activity, *ii)* video of the eye to measure pupil and eyelid dynamics, and *iii)* an inertial measurement unit (IMU) to measure head movement. This combination of sensors was chosen based on the literature reviewed in Chapter 2 as well as previous experience at the Christchurch Neurotechnology Research Programme (NeuroTech). Eye video alone is insufficient, since not all types of lapses are accompanied by eye closure [Jones et al. 2010] or other visual signs of decreased arousal [Davidson et al. 2007].

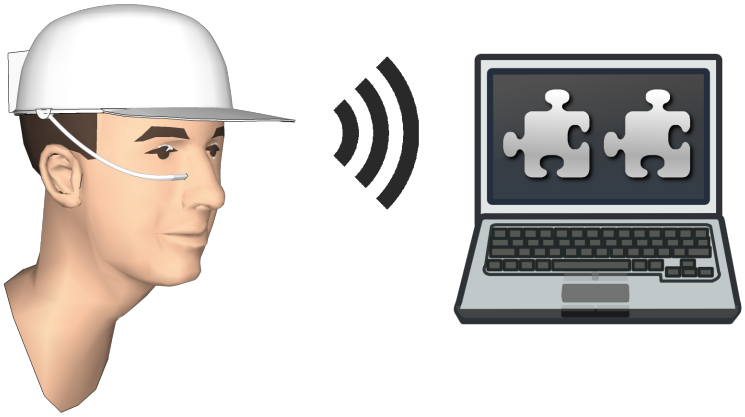


Figure 3.1 Concept drawing of the Elapse platform. A head-mounted device captures biosignals and transmits them to a remote computer where they are analysed by an extensible signal processing application.

Even in the case of microsleeps, which by definition involve eye closure, current eye-video-based systems can only detect microsleeps several seconds after their onset [Peiris et al. 2011]. EEG alone may, in theory, be sufficient to detect (or even predict) lapses, but the current state of the art has room for improvement—for example, not all EEG-defined microsleeps are accompanied by decreased performance. By capturing both eye video and EEG it should be possible to detect lapses with greater accuracy than by either one alone. The IMU was added because the additional effort required to include it was low—it may or may not prove to be useful.

The physical form of the device follows from the choice of sensors. Because it is capturing EEG, at least part of it must be mounted on the head. Using a head-mounted camera also makes it easier to acquire high resolution video of the eyes from a consistent angle compared to a remote-mounted camera.

However, having the device mounted on the head places an upper limit on the allowable size and power consumption of the device's processor. It must be small enough to be contained within the head-mounted device and efficient enough to run on batteries for several hours. By using the head-mounted device solely for data acquisition and off-loading the signal processing to a remote computer, these constraints do not limit the performance of the system as much as if it was a stand-alone device. Separating the acquisition from the analysis minimizes the computation done on the device itself, and therefore reduces the load on the embedded processor. It also provides greater freedom to experiment with computationally-intensive signal processing algorithms since these run on a PC rather than the embedded processor. An additional minor advantage to the two-part design is the relative ease of debugging software running locally rather than on an embedded device.

Although there are several advantages to dividing the system in two, there are also disadvantages. All of the data captured by the device needs to be transmitted

to the remote computer in real time. This must be done over a wireless connection since a cable could interfere with the user's activities. Transmitting data takes time, so this increases the latency of the overall system. Raw video has a very high data rate, so it must be compressed to a rate that is able to be transmitted over the wireless interface. It is possible that the additional power required for video encoding and wireless transmission could negate savings from moving the signal analysis to a remote computer. Doing the signal analysis remotely also adds the cost of the remote computer to the total cost of the system. On balance, however, the flexibility that the two-part design provides to experiment freely with complex signal processing algorithms outweighs these disadvantages.

Using a two-part design now does not eliminate the possibility of moving to a self-contained design in the future. If the system was to be developed into a commercial lapse detector, for example, the flexibility afforded by a two-part design would be less important than the total cost of the system.

3.2 HARDWARE

As described above, the Elapse head-mounted device serves solely as a data acquisition device. It captures signals from its sensor subsystems and streams them to a remote computer for analysis. The following sections describe the design of the Elapse device hardware, beginning with the base board containing the main processor, followed by the sensor subsystems. Complete schematics are provided in Appendix A.

3.2.1 Base board

The Elapse base board hosts the main system processor, the IMU, and ancillary electronics such as power supplies and status LEDs. An annotated photograph of the board is provided in Figure 3.2, shown at actual size.

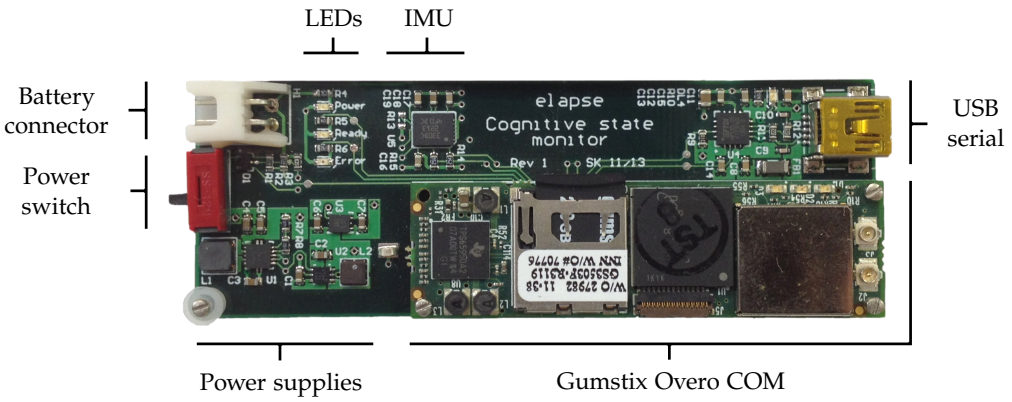


Figure 3.2 The Elapse base board (scale 1:1).

The base board is a four-layer printed circuit board (PCB) with all components and most routing on the top layer. A continuous ground plane runs directly under this layer to minimise electromagnetic emissions. The remaining internal plane is split among supply voltages, and the bottom layer has a connector for a daughter board. With this stack-up, there are two copper planes held at fixed potentials and several millimetres of physical separation between the electrically noisy components on top of the base board and any sensitive components on the daughter board.

Three LEDs provide information about the status of the system: *i)* green—the system is powered on; *ii)* blue—the device has finished booting and the server is running; *iii)* red—an error has occurred (steady) or the battery is low (flashing).

3.2.1.1 Processor

The head-mounted device is based around a Gumstix[1] Overo Fire computer-on-module (COM). The Gumstix is, in turn, based around a Texas Instruments[2] OMAP3530 system-on-chip (SoC). The OMAP3530 is a heterogeneous multi-core SoC combining a 720 MHz ARM Cortex-A8 core, a TMS320C64x+ DSP, and an image signal processor (ISP). To the OMAP3530 the Overo Fire adds 512 MB each of SDRAM and NAND flash, IEEE 802.11g Wi-Fi, Bluetooth, and a µSD card slot. Importantly, the OMAP3530 has a camera interface which the Gumstix COM exposes on a dedicated connector. The overall dimensions of the Gumstix are 58 mm × 17 mm × 4.2 mm.

Several alternatives to the Gumstix Overo were also considered, including single-board computers such as the BeagleBoard[3] and IGEPv2,[4] and COMs such as the Torpedo from Logic PD.[5] The Gumstix was chosen over these options because it is smaller than the BeagleBoard and IGEPv2, and has a more active user community than the Torpedo. The limiting factor in the choice of device was the ability to analyse or encode video in real time, while still being small enough to be wearable. Rather than using a powerful SoC to encode video, it would also be possible to use a less powerful microcontroller with an external video encoder IC such as the ASC8850A from NXP Semiconductors[6] or a field-programmable gate array (FPGA). This approach was not taken because it would have involved more complex hardware design.

3.2.1.2 Power supplies

If the Elapse device is to be wearable, there can be no cables tethering the user to any remote equipment. As a result, the device must be battery powered. A rechargeable

[1] `http://www.gumstix.com`
[2] `http://www.ti.com`
[3] `http://www.beagleboard.org`
[4] `http://www.isee.biz`
[5] `http://www.logicpd.com`
[6] `http://www.nxp.com`

lithium-ion polymer (LiPo) battery was chosen because it has the highest energy density among the widely available battery chemistries. The LiPo cell voltage is nominally 3.6 V, though the usable range is about 3.0–4.2 V. The Gumstix requires a system supply of $V_{\mathrm{SoC}} = 3.3\,\mathrm{V}$. Given the range of battery voltage and accounting for dropout voltage, a step-down voltage regulator cannot effectively supply V_{SoC} from a single LiPo cell—only about half the cell's capacity would be usable. The EEG analogue front-end requires $V_{\mathrm{AFE}} = 5.0\,\mathrm{V}$. Two cells could be connected in series to provide 7.2 V (6.0–8.4 V), which would be high enough for both V_{SoC} and V_{AFE}, however it proved easier to source a 3.6 V battery than a 7.2 V battery of the required capacity. The Elapse device therefore uses a 3.6 V battery with a step-up DC–DC converter to $V_{\mathrm{SYS}} = 5.2\,\mathrm{V}$. This provides some headroom to derive both V_{SoC} and V_{AFE} from V_{SYS}.

The hierarchy of power supply voltages is illustrated in Figure 3.3. Each letter A–D refers to one of the following voltage regulators:

A. A TPS63000 DC–DC converter steps the battery voltage V_{BAT} up to $V_{\mathrm{SYS}} = 5.2\,\mathrm{V}$. All other voltages are derived from V_{SYS}.

B. A TPS62291 buck converter steps V_{SYS} down to $V_{\mathrm{SoC}} = 3.3\,\mathrm{V}$. V_{SoC} is the main supply for the Gumstix COM and the camera board.

C. An LP5900 linear regulator steps V_{SoC} down to $V_{\mathrm{IO}} = 1.8\,\mathrm{V}$. All digital input/output (I/O) on the device is at V_{IO}.

D. This is a voltage regulator dedicated to the EEG module and is described in Section 3.2.2.2.

Soft shutdown

The base board has a soft-shutdown circuit that completely powers off the device when switched off, but allows the Gumstix to complete its shutdown sequence first.

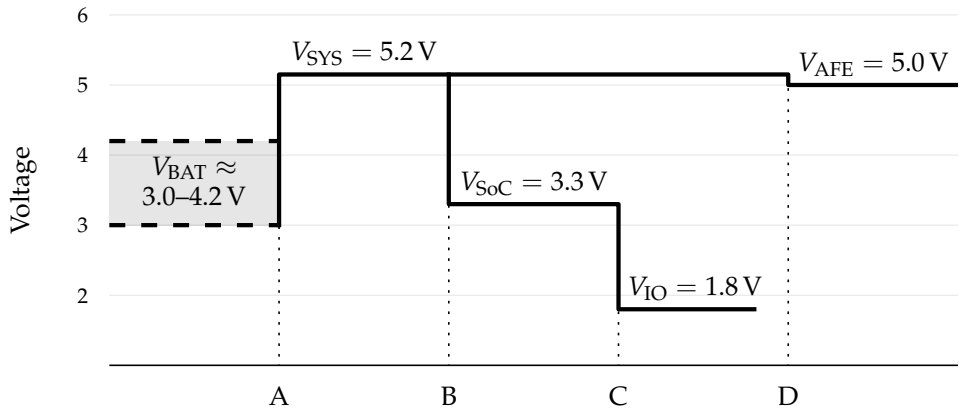
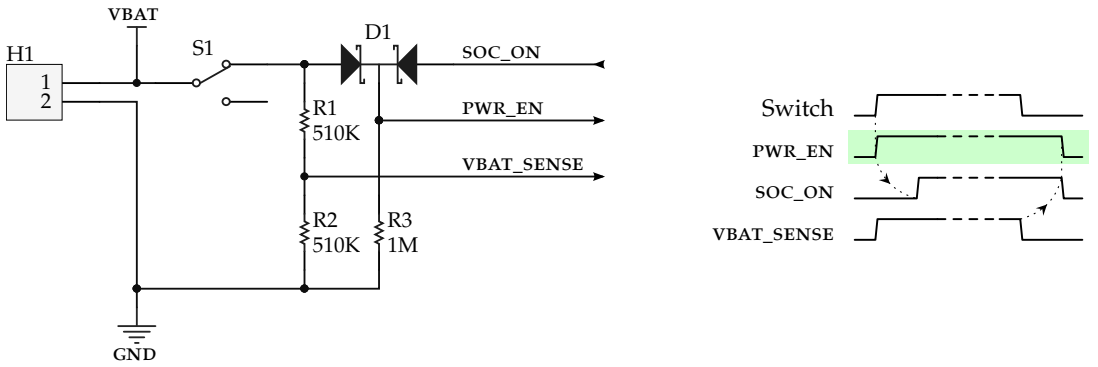


Figure 3.3 Power supply voltages. Each letter A–D corresponds to a voltage regulator.



(a) Soft shutdown circuit.

(b) Soft shutdown timing.

Figure 3.4 Soft shutdown circuit design. PWR_EN is the logical OR of the power switch and SOC_ON.

The schematic for this circuit is shown in Figure 3.4a, where switch S1 and connector H1 are the power switch and battery connector visible on the left side of the PCB in Figure 3.2. The circuit's functionality is based around a common-cathode diode pair acting as an OR-gate. Figure 3.4b shows the operation: when the switch is closed, PWR_EN goes high which enables the main system regulator. When the Gumstix boots, it drives the SOC_ON line high. The VBAT_SENSE line is connected to an analogue-to-digital converter (ADC) on the Gumstix which allows the Elapse software to monitor the battery voltage. When the switch is opened, the software detects that VBAT_SENSE is pulled to GND and triggers a system shutdown. At this point, SOC_ON is still driving PWR_EN high. The last instruction executed by the processor is to drive SOC_ON low, disabling the system regulator and powering off the system.

The common cathode diode pair in this circuit are Schottky diodes because they must have a low forward voltage drop, V_f. The Gumstix uses 1.8 V I/O and the system regulator's "enable" input high threshold $V_{\mathrm{IH}} = 1.2\,\mathrm{V}$, requiring $V_f < 0.6\,\mathrm{V}$. The forward voltage of the Schottky diodes is less than 0.2 V.

The power dissipated in this circuit is negligible in the context of the device as a whole. In the worst case, with a fully charged battery ($V_{\mathrm{BAT}} \approx 4\,\mathrm{V}$), the circuit draws approximately 8 µA.

3.2.1.3 USB console

The Elapse base board includes a USB interface to the serial console of the Gumstix. This provides a reliable connection to the SoC which is particularly useful for debugging issues with the bootloader or device drivers. The interface is based around the FTDI[7] FT230X USB to UART IC in a bus-powered configuration. Powering the interface circuit over USB ensures that it only draws power when needed—when connected to a USB host.

[7] `http://www.ftdichip.com`

It is important to note that operating the device with the USB cable connected introduces a small potential for an electrical hazard. Connecting the cable to a grounded computer ties the "ground" of the device to earth. Although there is no exposed ground on the device, it is possible that if a fault developed within the device and the user was exposed to an external voltage, the USB cable could provide a path to ground via the user's head. As unlikely as this may be, it could be possible if, for instance, a damaged active electrode shorted the exposed electrode to the device ground. To be sure of the user's safety, the device should not be worn when the USB debugging interface is attached.

3.2.1.4 Inertial measurement unit

A six-axis inertial measurement unit (IMU) is mounted on the Elapse base board to track the orientation of the user's head. The IMU is an LSM330DLC from STMicroelectronics[8] which integrates a 3D accelerometer and a 3D gyroscope with an I^2C interface. Using an IC which includes both accelerometers and gyroscopes in a single package avoids the need to compensate for axis misalignment caused by inaccuracies during PCB assembly.

3.2.1.5 Daughter board interface

On the bottom side of the Elapse base board is a connector for interfacing with a daughter board. This connector carries power, I^2C, serial peripheral interface (SPI) with two chip selects, and eight general-purpose input/output (GPIO) lines, making it suitable to connect to a wide variety of peripherals. The connector is placed near the centre of the board and the board has screw holes in the corners, providing a secure mechanical connection.

3.2.2 EEG module

3.2.2.1 Requirements

Choosing the number of EEG channels to include on the device is a trade-off between spatial resolution and the practicalities of a wearable device. The device must have enough channels to acquire a meaningful representation of neural activity, but having a large number of electrodes increases the time and effort required to put the device on and ensure all electrodes have low contact impedance. This is a significant drawback for a device that is intended to be used in real-world situations. On the other hand, having more electrodes provides a degree of redundancy if the contact impedance of one or more electrodes degrades over time. With this in mind, and from

[8] `http://www.st.com`

previous experience at NeuroTech, a target of 16 channels was chosen as a reasonable compromise.

The device should be able to acquire EEG up to and including the gamma band, i.e., $f \leq 40\,\text{Hz}$. It must therefore have a sampling rate $f_s > 80\,\text{Hz}$ to satisfy the Nyquist criterion, along with appropriate anti-aliasing filters.

The device is intended to operate in unshielded real-world environments, so it should have sufficiently high common-mode rejection to acquire usable data in the presence of external electromagnetic interference (EMI).

3.2.2.2 Implementation

The EEG module is based around a pair of ADS1299 analogue front-ends (AFEs) from Texas Instruments. The ADS1299 features eight low-noise programmable-gain amplifiers (PGAs) and eight simultaneous-sampling $\Delta\Sigma$ ADCs (Table 3.1). Although the PGAs have a maximum gain of only 24, in combination with the 24-bit ADCs this equates to a resolution of 22 nV when using the internal voltage reference.

The Elapse device uses two ADS1299s to provide a total of 16 input channels. One of the AFEs is configured as a master and provides the ADC clock and reference voltage to the other, ensuring they share common voltage and time references. The inputs are wired in a referential configuration—each channel has an electrode connected to its positive input and all negative inputs are connected to one reference electrode.

All electrodes have single-pole RC filters with a cut-off frequency of 72 Hz. Such simple analogue filters can be used because the digital decimation filters in the AFEs also provide anti-alias filtering. The RC filter cut-off was chosen so that EEG bands up to and including gamma have minimal attenuation. This cut-off is also comparable to the typical low-pass frequency of 70 Hz used in clinical EEG recordings [Duffy et al. 1989, p. 35].

Table 3.1 ADS1299 EEG analogue front-end specifications.

Feature	Description
Channels	8 (simultaneously sampled)
ADC resolution	24 bit
ADC data rate	250 Hz to 16 kHz
Amplifier gain	1 to 24
Input-referred noise	$1.0\,\mu V_{\text{p-p}}$
CMRR	-110 dB
Power dissipation	5 mW/channel
Digital interface	SPI
Package	TQFP-64

For maximum interference rejection, inputs to differential amplifiers should have equal impedances. This way, any interfering electromagnetic field induces the same voltage in both conductors and the differential amplifier can reject the common-mode voltage. In the referential EEG montage used here, however, the reference electrode is connected to 16 amplifiers in parallel, reducing the impedance on that line. To counteract this effect, a unity-gain buffer amplifier is placed between the reference electrode and the AFEs to provide the same input impedance as for the measurement electrodes. This buffer can be enabled or disabled by changing jumpers in order to test how much of an effect it has on common-mode rejection. When enabled, the buffer acts as an extra noise source in the signal path, so any improvement in common-mode rejection must be balanced against an increased noise level.

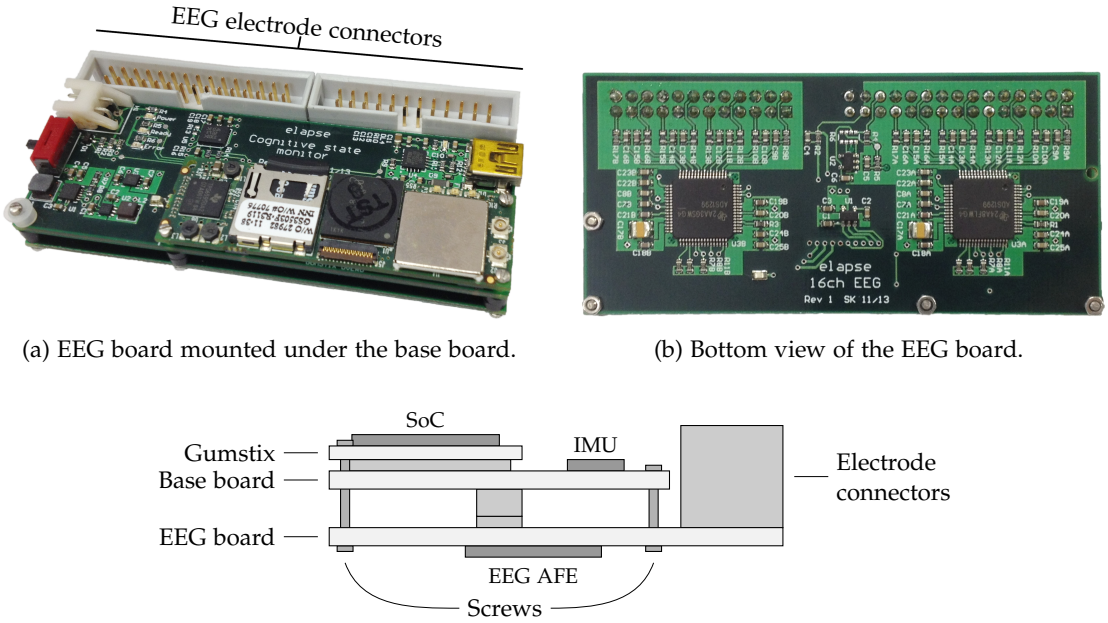
Clinical EEG systems usually ground the patient in order to reduce the amplitude of 50 Hz mains interference present on the body [Duffy et al. 1989, p. 56]. This is, of course, not possible in a battery-powered device; the whole system is floating relative to ground. The ADS1299 provides a way to emulate this technique by using a patient bias amplifier, similar to the right-leg-drive circuit used in ECG systems. This approach uses another electrode to drive the body with an inverted common-mode signal from the measurement electrodes. There is provision for a bias electrode on the Elapse device, though it is not used in the current configuration.

A TPS73250 low-dropout voltage regulator (LDO) is used to provide a low-noise analogue supply for the AFEs ($V_{\mathrm{AFE}} = 5.0\,\mathrm{V}$). The system supply voltage $V_{\mathrm{SYS}} = 5.2\,\mathrm{V}$ was chosen so that the difference between it and V_{AFE} is slightly greater than the worst case dropout voltage of the TPS73250. Using a linear regulator in this way filters out the switching noise generated by the V_{SYS} boost converter, while minimizing power dissipation by using the lowest possible input voltage to the LDO.

PCB layout

Like the Elapse base board, the EEG board is a four layer PCB. It is attached to the base board's daughter board connector and oriented so that the sensitive EEG AFEs are on the side furthest from the base board (Figure 3.5). A dedicated analogue ground plane is placed directly underneath the AFEs and star-connected to the system ground near the output of the LDO. The remaining internal plane is split among supply voltages and digital ground. Most digital signals are routed on the back side of the board, over a digital ground plane and away from the analogue components. The input traces between the electrode connectors and the AFEs are surrounded by a copper fill tied to analogue ground to minimize cross-talk between channels.

The electrode connectors were designed with future extensibility in mind. The connectors accept a ribbon cable with conductors split into groups of three, one group going to each electrode. Each group carries 5 V and ground as well as the electrode



(a) EEG board mounted under the base board.

(b) Bottom view of the EEG board.

(c) End view of the assembled boards.

Figure 3.5 Assembled Elapse electronics. The EEG board is mounted under the Elapse base board, with the electrode connectors positioned along one side.

signal, allowing for possible future expansion to an active electrode design. Having power and ground conductors interleaved between the signals on the ribbon cable also provides some shielding, reducing the capacitive coupling between adjacent channels.

Alternative AFE

The ADS1299 was the only commercially-available single-IC solution specifically designed for EEG acquisition at the time the EEG module was being developed. Another product, the RHD2000 from Intan Technologies,[9] was released shortly afterwards and is comparable in many ways. It is better than the ADS1299 in that it has 32 channels and programmable on-chip analogue filters. It is worse in that it has a single multiplexed 16 bit ADC and a higher noise floor. One very favourable point, though, is that it is available on a small, self-contained board that has electrode connectors on one side and SPI out the other. Had it been available at the time, the RHD2000 may have been a better choice than the ADS1299.

3.2.2.3 Safety

The Elapse device is a wireless, battery-powered system that has no connection to ground. The highest voltage present on the device is 5.2 V. The EEG electrodes are connected to differential amplifier inputs and so are not even referenced to the device's

[9] `http://www.intantech.com`

battery. As a result, there is no risk of electric shock from this device unless the USB cable is connected, as described in Section 3.2.1.3.

3.2.3 Dry electrodes

3.2.3.1 Requirements

Because the Elapse device is intended for use outside of the laboratory, it cannot rely on conventional wet electrodes that require skin preparation, electrolyte gel, and placement by a trained operator. The device, including electrodes, must be able to be taken on and off by the user with minimal training.

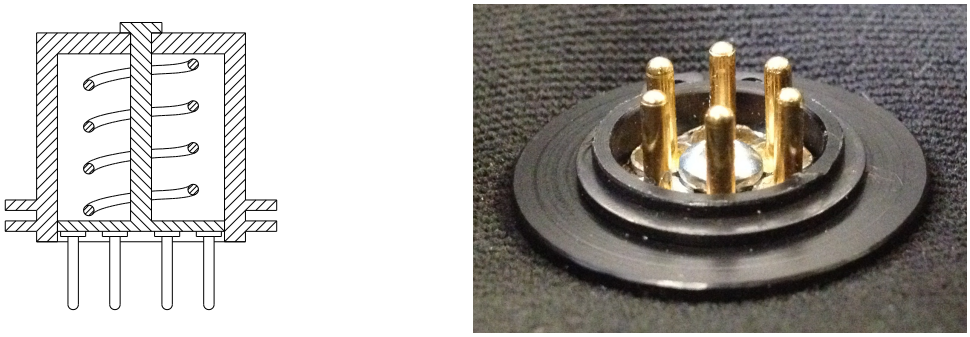
The electrodes must be able to make contact with the scalp for a wide variety of hair styles. That is, whether the user is bald or has thick curly hair, the electrodes should still be able to make electrical contact with the scalp.

The device may be worn for several hours at a time, so the electrodes must not cause discomfort to the user when worn for extended periods. Also, the contact impedance should not degrade over time.

EEG electrodes are, of course, mounted on the head, so it is important to ensure that their design does not introduce any hazard to the user's head. The design of the electrodes and their mounting hardware should ensure that if the user bumps their head while wearing the device the electrodes will not cause any harm.

3.2.3.2 Implementation

As described in Section 2.1.1.1, a number of companies and research groups are developing dry electrodes, though none have yet emerged as a clear competitor to conventional wet electrodes. Until a quality commercial electrode is available, the Elapse device uses dry electrodes of my own design.



(a) Cross-section of the dry electrode (scale 3:2). (b) An electrode mounted in the fabric cap.

Figure 3.6 The dry electrode design. Gold-plated, spring-loaded prongs reach through the hair to contact the scalp.

As depicted in Figure 3.6 the dry electrodes have 2 mm diameter prongs with rounded ends, mounted on a sprung platform, which push through the hair. The prongs are made of brass with a gold-over-nickel plating to provide good electrical conductivity and biocompatibility. The sprung platform allows the prongs to retract fully into the plastic housing. This design bears some similarity to the g.SAHARA electrodes from g.tec, except those are not retractable.

On the current prototype of the Elapse device there are eight electrodes positioned at F3, F4, C3, C4, P3, P4, O1, and O2, plus a reference electrode at Cz (Figure 3.7). This reduced set of electrodes was chosen, rather than using all of the device's 16 channels, because this electrode design is a temporary solution. It allows us to capture EEG in the early stages of development, but it is likely that the design will need to be either revised or swapped for a commercial design if a suitable one becomes available. In future revisions it may be useful to add frontal channels to capture some EOG-like information.

The electrodes are mounted in a stretchable fabric cap (modified NRS Mystery[11] kayak helmet liner), allowing the device to conform to a range of head shapes and sizes. The elastic fabric pulls the electrodes towards the head so that the electrode prongs reach through the hair to contact the scalp.

3.2.3.3 Safety

As mentioned earlier, given the intended real-world applications of the Elapse device, it is important that the EEG electrodes do not present a hazard to the user's head. To ensure this, the fabric cap in which the electrodes are mounted is suspended inside a standard safety helmet. If the user receives a blow to the head, the fabric cap is pushed against the scalp and the electrode prongs retract into their housings. Because

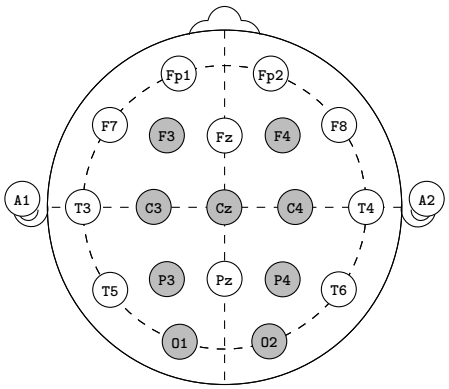


Figure 3.7 10/20 system EEG electrode positions.[10] Electrodes used on the Elapse device are shaded.

[10] Adapted from `https://commons.wikimedia.org/wiki/File:21_electrodes_of_International_10-20_system_for_EEG.svg`.

[11] `http://www.nrs.com`

of the way the cap is suspended inside the helmet, no external force can be transmitted directly through the electrode prongs to the head.

3.2.4 Camera module

3.2.4.1 Requirements

Since the device is intended to monitor people in real-world situations, it must be as unobtrusive to the wearer as possible. The camera module must therefore be small ($\sim$10 mm wide) and positioned in such a way that it does not impair their vision.

The video frame rate must be sufficiently high to differentiate between normal blinks and slower drowsy eye closure. The mean time that the eyes remain completely closed in the middle of a blink increases from 0.3 ms when alert to 144 ms when drowsy, and also becomes more variable [Tucker and Johns 2005]. The target frame rate is 60 fps, which gives a span of $\sim$9 frames over an average eye closure when drowsy.

The camera should be positioned close to the face, preferably no further away than the tip of the nose, to make the device as compact as possible. This requires the camera lens to focus at a distance of less than 40 mm.

The device must operate under a wide range of lighting conditions, from office lighting to direct sunlight to a dark night. The camera subsystem must be able to capture video across this dynamic range, providing illumination if necessary.

3.2.4.2 Implementation

An extensive search revealed no commercially available camera modules that satisfied all of these requirements. Most of the cameras that met the size requirement had a maximum frame rate of 30 fps. The cameras that supported higher frame rates were too large or had auto-focus mechanisms that cannot focus at such a short object distance. Therefore, there was little choice but to develop a new camera module to meet all of these requirements.

The Elapse camera module is based around the OV7735 image sensor from OmniVision Technologies.[12] The key specifications of the OV7735 are given in Table 3.2 [OmniVision 2011a]. Note that the sensor is capable of producing 60 fps at its full resolution (and even higher rates at lower resolutions). It outputs data over a 10-bit parallel interface at 1.8 V logic levels, which is compatible with the camera interface on the Gumstix. The sensor has good low-light sensitivity which is important for capturing low-noise video at night.

[12] `http://www.ovt.com`

Table 3.2 OV7735 image sensor specifications.

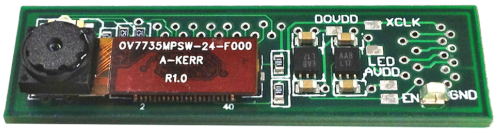
Feature	Description
Technology	CMOS
Chroma	Colour
Array size	640×480 (VGA)
Frame rate	60 fps at VGA
Output formats	8/10 bit RGB
	8 bit YCbCr 4:2:2
Output interface	Parallel
Sensitivity	$3300\,\mathrm{mV\,lx^{-1}\,s^{-1}}$
Active power	100 mW
Package size	$3.99\,\mathrm{mm} \times 3.64\,\mathrm{mm}$

A colour sensor is not necessary for this device. In fact, it would be preferable to have a greyscale sensor because a coloured Bayer filter over the pixels reduces the amount of light reaching the sensor. Also, under monochromatic illumination the colour information has little meaning so bandwidth could be saved by only capturing greyscale video. Manufacturers, however, produce a much wider range of colour image sensors than greyscale ones, and often at a lower price. In this case, the sensor that best matched the requirements was a colour one.

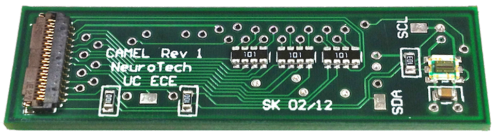
The OV7735 has an on-chip image signal processor (ISP) which implements several useful image processing algorithms in hardware. It supports automatic exposure, gain, white balance, flicker, and black level controls. It allows control of frame rate, mirror and flip, scaling, cropping, and subsampling. It also has some image quality controls that allow adjustment of colour saturation, hue, gamma, sharpness, and lens shading correction. The automatic exposure and gain controls are particularly useful for the Elapse device since it needs to operate under a wide range of ambient illumination levels.

Another reason for choosing the OV7735 was that it is available with an integrated lens assembly on a small module ($6.0\,\mathrm{mm} \times 20.5\,\mathrm{mm}$) with a board-to-board connector. Designing an interface board to accommodate such a module seemed like an easier prospect than placing a ball grid array (BGA) component by hand and manually aligning the lens mount. However, these modules must be specially ordered from OmniVision, as they are built to order. A month after placing the order, OmniVision's contract manufacturers informed us that the module could not be manufactured because the lens it used was no longer available. OmniVision had to re-design the module to use a different lens, so the total time from order to delivery came to almost five months. In hindsight, it would probably have been easier and faster to design for the BGA part directly, even considering the difficulty of manual assembly.

The complete camera module is shown in Figure 3.8. For the sake of having a



(a) Top side of the CAMEL module. The camera is on the left with the edge of the IR LED just visible behind it.



(b) Bottom side of the CAMEL module. The ambient light sensor is on the right and the cable connector on the left.

Figure 3.8 The CAMEL module: a camera for measuring eye-related lapse indicators.

convenient name, it is known as "CAMEL": a CAmera for Measuring Eye-related Lapse indicators. The CAMEL module incorporates the OV7735 image sensor, an IR LED, an ambient light sensor, and associated voltage regulators, etc. It measures 11 mm $\times$ 41 mm; the minimum width is limited by the size of the cable connector, as seen in Figure 3.8b.

The IR LED has a peak wavelength of 850 nm and is positioned so that it illuminates the eye area. Under near-infrared (NIR) illumination, the iris appears lighter than under visible light [Daugman 2004] which has the desirable effect of increasing the contrast between the pupil and the iris (Figure 3.9). NIR illumination also allows the camera to capture video in the dark, where using visible light would otherwise interfere with the wearer's vision.

The ambient light sensor is positioned on the back side of the CAMEL module, facing away from the face. This sensor has two sensing elements which allow separate measurements of visible and infrared irradiance. Communication is done over an I^2C interface which allows it to connect to the camera control bus, avoiding the need for additional I/O lines. The sensor was initially intended to be used as part of an adaptive illumination system, altering the power of the IR LED according to ambient light levels, though this was ultimately found to be unnecessary. The sensor may still be useful to compensate for lighting conditions when monitoring pupil diameter.

(a) Ambient lighting only, with low-contrast pupil.

(b) Ambient + NIR with improved contrast between pupil and iris.

Figure 3.9 The effect of NIR illumination on pupil–iris contrast.

Infrared illumination

Since CMOS image sensors are naturally sensitive to NIR, colour cameras usually have an IR filter in the lens assembly to avoid colour distortion. This was the case with the camera module from OmniVision. Of course, this filter renders the IR illumination ineffective. The glued-in lens assembly had to be carefully removed and the IR filter pried off so that the sensor could capture infrared images.

Eyes under infrared illumination can exhibit either the bright- or dark-pupil effect depending on the distance between the IR source and the optical axis of the camera [Ji and Yang 2002]. The bright-pupil effect occurs when the source is close to the camera and IR enters the pupil, reflects off the retina, and passes back through the pupil to the camera. The dark-pupil effect occurs when the IR source is positioned away from the camera. In this case the narrow aperture of the pupil prevents any reflected IR from reaching the camera. A dark pupil arrangement is easier to implement since there is no need for ring lights or beam splitters to provide co-axial illumination [Babcock and Pelz 2004]. On the CAMEL module, the IR LED was initially mounted close to the camera in an effort to induce the bright pupil effect, but at the typical camera–eye distance of $\sim$35 mm only about one third of the pupil appeared bright. Instead, the LED was moved further away from the camera so that the pupil appeared uniformly dark.

Electrical interface

The cable between the CAMEL module and the Gumstix is a 27-conductor, 0.3 mm pitch, flat flexible cable (FFC). It carries the parallel data interface, the I^2C control bus, and the power supply for the camera. Besides being difficult to source because of its unusual pin count and pitch, this type of cable is not well suited to carrying video data over a long distance. To reach from below the eye to the side of the head, the cable must be at least 150 mm long. Having many parallel conductors carrying high-frequency single-ended signals over long distances without shielding goes against good electronics design. Unfortunately, there is not a lot of room to work around these short-comings because the choice of cable is constrained by the camera interface connector on the Gumstix.

A much better solution would be to use an interface that makes use of differential signalling. In fact, both the OV7735 image sensor and the OMAP3530 SoC support the MIPI CSI-1 serial camera interface. The CSI-1 interface is designed specifically for interfacing cameras to host processors and carries the data and clock over two SubLVDS differential pairs. At the time that the CAMEL module was being developed, however, Linux platform driver support was not available for the CSI-1 interface on the OMAP3530, and the OV7735 datasheet provided almost no technical information

about the CSI-1 interface.

3.2.4.3 Safety

To ensure that the infrared radiation from the CAMEL module will not cause harm to the wearer, the following calculations were carried out to compare the theoretical infrared irradiance to standardized exposure limits. These calculations are based on an application note from the IR LED manufacturer [Jäger 2010], which in turn is based on IEC standards 62471:2006 and 60825-1:2007.

There are three hazards to account for when irradiating the eye area: damage to the cornea, the skin, and the retina. This damage is caused by overheating the tissue, destroying cells. Because different tissues respond to IR differently, these hazards must be assessed separately.

Cornea hazard

The maximum allowable irradiance[13] of the cornea for exposure times exceeding 1000 s is $E_{e,\max} = 100\,\mathrm{W\,m^{-2}}$. The irradiance E_e from the IR LED can be found from the inverse square law

$$E_e = \frac{I_e}{d^2} \tag{3.1}$$

where the radiant intensity I_e is calculated from the datasheet [Osram 2012] to be $I_e \leq 4.38\,\mathrm{mW\,sr^{-1}}$ given a forward current $I_f = 25\,\mathrm{mA}$. Assuming a worst-case eye–LED distance of $d = 30\,\mathrm{mm}$, the irradiance is

$$E_e = \frac{4.38\,\mathrm{mW\,sr^{-1}}}{(30\,\mathrm{mm})^2} = 4.87\,\mathrm{W\,m^{-2}} \tag{3.2}$$

Clearly $E_e < E_{e,\max}$; the corneal irradiance is less than one twentieth of the exposure limit.

The maximum allowable irradiance of the skin is slightly higher than that of the cornea, so we are well within the limit here also.

Retina hazard

Since the lens of the eye focuses light onto the retina, the radiance[14] limit is a function of the angular subtense α of the light source.

$$\alpha = \frac{Z}{d} \quad \text{with} \quad Z = \frac{l + w}{2} \tag{3.3}$$

[13] Radiant power per unit area
[14] Radiant power per unit solid angle per unit projected source area

where $l = w = 0.2\,\text{mm}$ are the dimensions of the emitter. There is a minimum effective value imposed on α due to physical characteristics of the eye, so for exposure times exceeding 10 s this becomes

$$\alpha_{\text{eff}} = \max(\alpha, 0.011) \quad \Longrightarrow \quad \alpha_{\text{eff}} = \max\left(\frac{0.2}{30}, 0.011\right) = 0.011\,\text{rad} \tag{3.4}$$

The maximum allowable radiance is then given by

$$L_{\text{IR,max}} = \frac{6000}{\alpha_{\text{eff}}} = 545\,\text{mW}\,\text{mm}^{-2}\,\text{sr}^{-1} \tag{3.5}$$

The radiance of the IR LED is approximated by

$$L_{\text{IR}} \approx \frac{I_e \cdot R(\lambda)}{\left(\frac{l+w}{2}\right)^2} \tag{3.6}$$

where

$$R(\lambda) = 10^{(700-\lambda)/500} \tag{3.7}$$

is a wavelength-dependent burn hazard weighting function. Using the value of I_e from earlier and $\lambda = 850\,\text{nm}$, we find that the radiance of the IR LED is

$$L_{\text{IR}} \approx \frac{4.38\,\text{mW}\,\text{sr}^{-1} \times 0.5}{(0.2\,\text{mm})^2} = 54.8\,\text{mW}\,\text{mm}^{-2}\,\text{sr}^{-1} \tag{3.8}$$

Again, we see that $L_{\text{IR}} < L_{\text{IR,max}}$ for a factor of safety of 10.

These calculations indicate that the IR exposure from the CAMEL module is well below safety limits for the cornea, the skin, and the retina.

3.2.5 Complete wearable device

Now that the electronics have been described, this section describes how all of the subsystems are combined into a complete wearable device, along with the requirements for the device as a whole.

3.2.5.1 Requirements

The connection between the Elapse device and the remote computer must be wireless. A cable would interfere with the user's ability to carry out their tasks in some occupations.

For the device to be useful for monitoring people as they go about their jobs, it should be able to run off batteries for several hours—ideally for a typical eight-hour shift.

There are other "soft" requirements which are difficult to measure but which are nevertheless important for the device to succeed practically. For example, if it is to be worn for hours at a time, it should not cause discomfort to the user. It should also not be difficult or time-consuming to put on and take off, perhaps after some initial training on how to achieve good electrode and camera placement.

3.2.5.2 Implementation

The Gumstix has two wireless interfaces which can be used to transmit data to a remote computer: IEEE 802.11g Wi-Fi and Bluetooth v2.0+EDR. The Elapse device uses the Wi-Fi interface. Wi-Fi has the higher data rate of the two interfaces, at $54\,\mathrm{Mbit\,s^{-1}}$, but also uses significantly more power. By contrast, Bluetooth EDR has a nominal rate of about $3\,\mathrm{Mbit\,s^{-1}}$, and on the Gumstix the maximum rate is even lower ($921\,\mathrm{kbit\,s^{-1}}$) due to limitations in the Bluetooth driver. Higher rates are available in recent versions of the Linux kernel but the device is bound to an older kernel for the reasons outlined in Section 3.3.1. It would be preferable to use the Bluetooth interface because it uses less power, but because the data rate of video is very high, even when compressed (Section 3.3.1.3), the Elapse device uses the faster Wi-Fi interface for communication.

The first prototype of the Elapse device is constructed around a standard safety helmet, Figure 3.10. This provides a convenient sturdy base to build on, while also representing one form that the final device could take. Some occupations for which a lapse detection device would be beneficial require wearing a safety helmet anyway—e.g., crane operators, some truck drivers—so building the device into something the user must already wear makes sense. The structure of the helmet has been left intact as much as possible in an effort to keep the device functional as a safety helmet.

The camera is mounted on an adjustable arm which positions it under the right eye near the end of the nose, looking up at the eye. It is positioned as low as possible to avoid encroaching on the user's field of view while still providing a clear view of the eye. It would be easier to mount the camera above the eye looking down, e.g., on the brim of a hat, but that position was found to be unsuitable for users with bushy eyebrows or long eyelashes. The lower position also provides a better view of the eye when the user is looking down or their eyelids start to droop.

The device is powered by a 3.6 V, 6 Ah LiPo battery, providing a power budget of 2.7 W if the battery is to last for 8 h. The battery measures $54\,\mathrm{mm} \times 60\,\mathrm{mm} \times 19\,\mathrm{mm}$ and weighs 110 g. This is a similar size to the electronics module. The electronics module is mounted on the right side of the helmet and the battery is mounted on the left, balancing the design both physically and visually. The battery can be removed from the device, allowing a fresh one to be swapped in so that the device can be used while the other battery recharges.

Figure 3.10 The current prototype of the Elapse device.

The total cost of the parts required to build the Elapse device was about $650. Most of these components cost significantly less when ordered in volume, so the price per device when assembling multiple devices would be less. In addition to the Elapse device, a laptop is required to run the signal processing software which should be considered as part of the total cost of the system.

Besides the safety helmet, other form factors were also considered for the device. The most promising alternative was a glasses-based design which mounted the camera in the frame of a pair of glasses and had EEG electrodes positioned on semi-flexible arms, much like the Emotiv EPOC headset. While this would have made it easier to mount the camera in a good position, it provides less space to mount electronics and would be more difficult to manufacture.

3.3 SOFTWARE

The Elapse device hardware provides the necessary electronics to acquire and transmit biosignals. The next requirement is for software to both manage the hardware and process the acquired signals. As described in Section 3.1, the functionality of the Elapse platform is split between the wearable device and a remote computer. Given this structure, it follows that the software on the device needs to capture data, encode or compress it if necessary, and transmit it to the remote computer. It should also provide a configuration and control interface. This is described in Section 3.3.1. The software on the remote computer, in turn, needs to receive the data transmitted from the device, decode or decompress it, and analyse the resulting signals, as well as providing a user interface. This is described in Section 3.3.2.

Programming languages

Before describing the design of the Elapse software, it is worth mentioning the rationale behind the choice of programming language. The first prototype of the software was written in Python[15] because it is succinct and expressive and well suited to rapid application development. After implementing some proof-of-concept code, however, it became clear that the Python interpreter's lack of true multithreading was a significant bottleneck. Internally, the Python interpreter is not thread-safe; it uses a global interpreter lock (GIL) to ensure that only one thread executes at a time. Paradoxically, the GIL can cause multithreaded Python code to perform slower on multi-core processors than on single-core processors [Beazley 2010]. Since the goal was to develop a framework that involves both a lot of I/O and a lot of computation—a combination which particularly stresses the GIL—it seemed unwise to limit the design to a single thread of execution when multi-core processors are so readily available.

Consequently, a different language was chosen—C++ with the Qt libraries. Qt[16] is a cross-platform framework which provides several useful application development features, including a graphical user interface (GUI) toolkit, support for dynamically-loadable plug-ins, and high-level networking classes. Both the server running on the device and the client running on a laptop are implemented in C++ with Qt, allowing some code reuse between the two.

C++ was chosen over other languages because it is more expressive than C, faster than Python and other interpreted languages, and has direct interoperability with system libraries, unlike Java. The structure of this software—a chain/pipeline of signal processing functions—would be well suited to implementation in a functional language like Scala,[17] however for the sake of maintainability it seemed more important to choose a language that is widely known.

3.3.1 Embedded software

The embedded software running on the device consists of several layers. At the lowest level is the operating system (OS) and the device drivers for interacting with hardware. On top of that are the system libraries, the Qt framework, and finally the server application which implements the device's functionality. Of these, I customized the operating system, implemented two device drivers, and wrote the server application.

A simplified block diagram of the server application is shown in Figure 3.11. Data is read from each sensor via the device drivers described in Section 3.3.1.2 and transmitted to a remote computer as described in Section 3.3.1.3. A control interface is

[15] `http://www.python.org`
[16] `http://qt-project.org`
[17] `http://www.scala-lang.org`

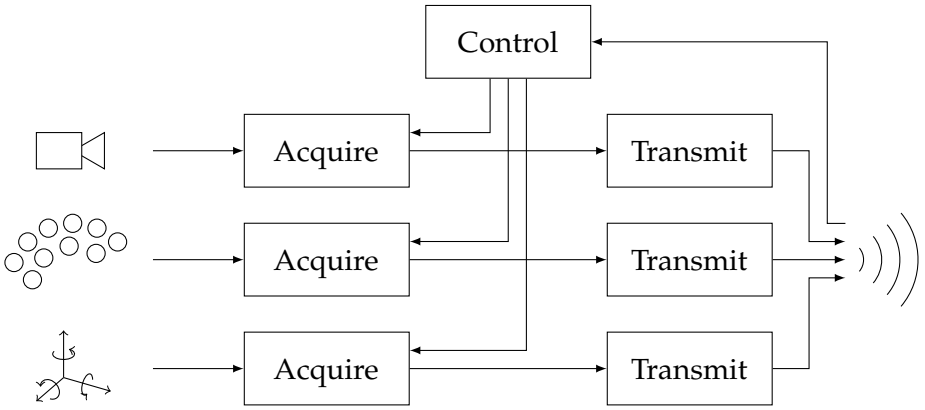


Figure 3.11 Server application architecture.

also provided which allows the remote computer to configure the sensors as necessary (Section 3.3.1.4).

3.3.1.1 Operating system

The Elapse device uses the Linux operating system. The kernel has been customized to remove all unnecessary features and modules in order to decrease boot time and memory utilization. Similarly, the set of software included on the root filesystem has been reduced to the Elapse server and its dependencies, along with some debugging tools.

The device is running the Ångström Linux distribution, based on version 2.6.34 of the Linux kernel, and built by the OpenEmbedded-Classic build system. This is in contrast to the more recent combination of the Poky distribution, based on version 3.x of the Linux kernel, built by OpenEmbedded-Core from the Yocto Project.[18] Development of the Elapse device began during the transition from the 2.6 to the 3.x kernel series. The older and more stable kernel and build tools were chosen, despite being obsolescent, because they were more widely supported at the time. The pace of embedded Linux development is such that it is impractical to keep up with all upstream changes; it is better to reach a working configuration and build upon that.

As a data acquisition system, the device may be more suited to a real-time operating system (RTOS) than a traditional OS, but the range of RTOSs supporting the OMAP3530 is very limited. Given the abundance of existing software available for Linux and minimal support for other OSs, Linux was the clear choice. It may be possible to use the PREEMPT_RT patch to enable hard real-time support for the Linux kernel, though it is unclear whether this would provide any measurable benefit.

[18] `http://www.yoctoproject.org`

3.3.1.2 Device drivers

Two of the key components of the Elapse device, the ADS1299 EEG AFE and the OV7735 image sensor, did not have existing Linux device drivers. I implemented these two drivers myself.

Camera driver

The OV7735 image sensor driver is modelled on an existing driver for the Aptina MT9V032. The MT9V032 is the sensor used on the Gumstix Caspa camera board and is therefore well supported within the Gumstix community.

The structure of camera drivers within the Linux kernel is complex and was undergoing changes during development of the OV7735 driver. The kernel's Video4Linux (V4L2) application programming interface (API) has had to adapt in order to better support the complex video processing hardware present on modern SoCs. In more recent kernels ($\geq$ 2.6.37), the media controller framework and the `v4l2_subdev` API are used to control the interaction of image sensors with the video capture and processing hardware modules on SoCs. At the time the CAMEL was designed, however, these systems were still in active development and not stable. Instead, the OV7735 driver uses the older `v4l2_int_device` API in the same way as the MT9V032 driver it is based on. This provided a stable target at the time, but this API has been declared obsolete by the V4L2 community. If, in the future, the device needs to be updated to use a newer kernel for any reason, the OV7735 driver will need to be ported to the `v4l2_subdev` API.

To be able to use a particular image sensor with a particular SoC, the kernel requires two drivers—a platform driver for the SoC's camera interface and a device driver for the image sensor (Figure 3.12). Each of these abstracts away hardware-specific details and conforms to the generic V4L2 API. On the Gumstix, the `omap34xxcam` platform driver supports the OMAP3530's ISP [OMAPpedia 2012]. The platform driver handles the actual capture of frame data, since that is specific to the SoC and not affected by the type of image sensor producing the data. The device

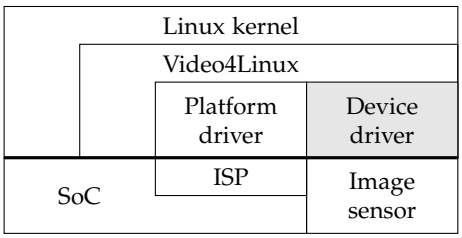


Figure 3.12 Video4Linux camera driver structure. The device driver provides a V4L2-compatible interface to the OV7735 image sensor.

driver for the OV7735 is responsible for configuring the sensor to produce data in the format requested by an application.

The OV7735 device driver implements the minimum functionality necessary to acquire video in the required format. As described in Section 3.3.1.3, the captured video is currently scaled on the SoC. The OV7735 supports these operations directly on the sensor, which would reduce the load on the host processor but, given the time constraints on the project, we chose to leave the driver at that point and only come back to it if further optimization proved necessary.

Although OmniVision provided a datasheet for the OV7735 [OmniVision 2011a], after configuring the sensor as the datasheet described, it did not function as expected. Upon request, OmniVision provided a full table of register values to achieve the configuration required for the Elapse device. Interestingly, about 48 % of these registers were not documented in the datasheet. This influenced the decision to not implement more functionality in the driver than absolutely necessary.

EEG analogue front-end driver

The driver for the ADS1299 EEG AFE was written from scratch, with some reference to existing drivers for SPI devices. It was designed to work with an arbitrary number of ADS1299 ICs configured as a single multi-channel device as shown in Figure 3.13. The primary IC is configured to provide the ADC clock and reference voltage to the other ICs so that they all operate from a common time and voltage reference. Each IC has its own SPI chip select so that it can be individually addressed by the host processor.

A GPIO line on the host processor is connected to the START input of all of the ADS1299s to trigger conversions simultaneously. The ADS1299s are configured for continuous conversion, so they sample data continuously until START is driven low. When a sample is ready, the primary AFE triggers an interrupt on the host processor. The corresponding interrupt handler immediately begins an asynchronous SPI read from each of the AFEs. The completion handler for the SPI transaction pushes the received data into a first-in first-out queue (FIFO). In the meantime, the driver's `read()` function blocks pending data. When the FIFO is full, `read()` wakes up, copies the contents of the FIFO to user-space, and goes back to waiting for data.

In addition to the blocking `read()` described above, the driver supports non-blocking `read()` and implements the `poll()` function. This gives user-space code the flexibility to interact with the device file by any of the standard POSIX methods: using `poll()` or `select()` to watch for readability, and using `read()` with or without blocking the calling thread.

Several samples are buffered in the FIFO before waking up `read()` to pass the data to user-space. There is some overhead associated with the context switch between

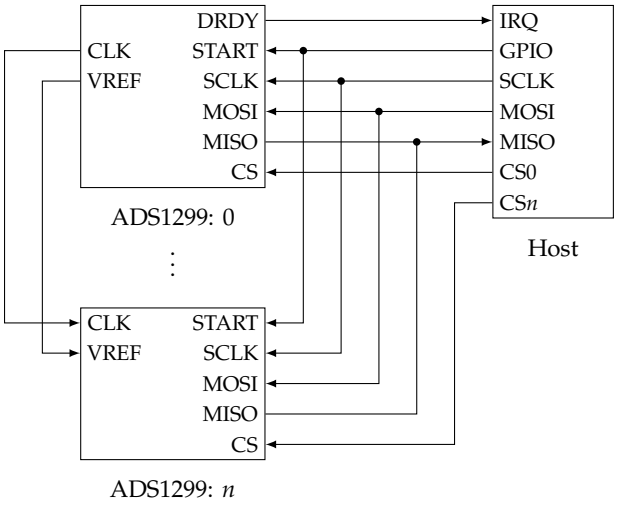


Figure 3.13 Multi-chip configuration supported by the ADS1299 driver. The first ADS1299 provides a reference voltage and clock to the other AFEs and sends a data-ready interrupt to the host. Data is read out over SPI with one chip select per AFE.

the kernel and user-space, so returning "blocks" of samples is more efficient than returning one at a time. The number of samples in each block can be configured through `sysfs` or when loading the driver into the kernel, and Section 3.3.1.3 explains how the block size was optimized for the Elapse device.

Most of the configuration registers on the ADS1299 are exposed through a `sysfs` interface. Simple, readable statements can be used to configure the AFEs, for example to set the gain of channel 1: "`echo 24 > /sys/class/eeg/ads1299/channel/1/gain`". When compiled in debug mode, the driver also supports direct access to individual registers through the `ioctl()` system call.

IMU driver

STMicroelectronics provides a Linux device driver for the LSM330DLC IMU. It exposes the accelerometer and gyroscope as two input devices and provides a `sysfs` configuration interface. However, this driver is poorly implemented, almost to the point of being unusable. Indeed, when sampling at 10 Hz it uses approximately 50 % of the CPU. Time constraints precluded fixing or writing a replacement for this defective driver. After some experimentation, however, it was found to be possible to configure the driver with very reduced functionality to avoid the extreme CPU load. The only properly-functioning configuration is with the accelerometers sampling at 10 Hz and the gyroscopes completely disabled. Increasing the sampling rate beyond this does not result in an increase in the actual output data rate, and enabling the gyroscopes loads the CPU as described above. While this configuration is far from ideal, it does provide *some* inertial data to work with until the driver can be fixed or replaced.

3.3.1.3 Data transmission

The Elapse device's Wi-Fi is configured to make it simple to connect to the device. Each device provides an ad hoc Wi-Fi network and runs a DHCP server. With this arrangement, connecting to the device from the remote computer is as simple as selecting the device's name from the list of available Wi-Fi networks. No manual configuration is required—the DHCP server on the device will assign the remote computer an IP address on the device's network. Every Elapse device is configured with the same static IP address, so no DNS server or multicast DNS is required to resolve the address of the device.

All data from the sensors is transmitted over UDP. Unlike TCP, UDP has no acknowledge/retransmit mechanism to handle dropped packets. While this means that any dropped packets are irrecoverable, it also means that it has less protocol overhead and lower latency. In a real-time situation, retransmitting data would simply delay the next packet waiting to be transmitted. In the Elapse platform, dropped packets can be detected at the application level by inspecting the sequence numbers or timestamps of the samples once they have been received by the client. With that knowledge, signal processing algorithms may interpolate or otherwise handle any missing data.

Video compression

Video comprises 99.9 % of the data captured by the device; the uncompressed video has a bit rate of $184\,\mathrm{Mbit\,s^{-1}}$. This is more than three times the nominal bit rate of the device's Wi-Fi interface, so the bit rate of the video needs to be reduced significantly before transmission.

The first step of the bit rate reduction is to scale the video down by a factor of two in each dimension, from its original 640×480 to 320×240 pixels. This is done efficiently in hardware by the OMAP's ISP. The scaled video is then encoded to a compressed format. Texas Instruments provides an encoder for the H.264 format that is optimized to run on the DSP core of the OMAP3530, reducing the load on the CPU. As a whole, this video compression pipeline reduces the video bit rate to approximately $960\,\mathrm{kbit\,s^{-1}}$—an overall reduction of 192:1 resulting in an average of 0.2 bits per pixel. The encoded video is transmitted to the remote computer using the real-time transport protocol (RTP), a UDP-based protocol for transmitting real-time video over IP networks.

In the version used on the Elapse device, GStreamer,[19] the multimedia processing library with which the video compression pipeline is implemented, has a bug in its handling of video timestamps. It ignores the original timestamp which gets applied

[19] `http://gstreamer.freedesktop.org`

to each frame when it is captured and instead generates a new one. Additionally, the H.264 encoder ignores the timestamps of the input frames and generates new timestamps for the encoded output. For the Elapse device, accurate timestamps are important for maintaining synchronization between the signals captured from the different sensors. To work around this bug, the video source element in GStreamer had to be patched to extract the correct timestamps from the captured frames and then re-apply these to the encoded video. This patch is not good software design but it is required to ensure that each frame of video has an accurate timestamp.

EEG block size

As described in Section 3.3.1.2, the number of samples returned by each `read()` from the EEG driver is configurable. This parameter has been tuned for the Elapse device so that each block of data that is read can be sent directly as a UDP packet of the optimum size. It is desirable to put as much data as possible into each packet to reduce network overhead, but packets larger than the maximum transmission unit (MTU) will be split into multiple fragments at the IP layer, adding more overhead. The default MTU for the Wi-Fi adapter on the Gumstix is 1500 B. By configuring the EEG driver to return 20 samples from each `read()`, and adding 42 B for the UDP and IP headers, the packet size comes to 1362 B.

The maximum MTU for Wi-Fi is actually 7981 B but the Wi-Fi adapter on the Gumstix uses the maximum MTU of Ethernet by default. This makes sense in most situations since Wi-Fi is usually used as a wireless access point to a larger Ethernet network. In the case of the Elapse platform, however, there is a direct Wi-Fi link between the device and the remote computer with no Ethernet involved. Because of this, it may be possible to increase the MTU of the device's Wi-Fi adapter and thus allow larger packets to be transmitted, reducing overhead.

3.3.1.4 Control interface

The server application exposes a configuration and control interface to the remote computer. This allows the client software to configure the device's sensors, monitor hardware state such as battery voltage, and trigger actions such as starting and stopping data capture.

This interface is implemented using DBus over TCP. DBus is a message-passing inter-process communication (IPC) and remote procedure call (RPC) mechanism that is widely used in Linux software. The Qt framework has good support for DBus, including automatically generating C++ interface code from XML interface descriptions. Using a well-established RPC system makes it easy to test the interface from shell scripts or DBus debuggers since the tools already exist. DBus is usually used

for communication between processes on the same computer, but all communication is done over sockets so it works equally well communicating with remote processes over TCP.

The server is configured to launch automatically at boot, so the remote interface is available as soon as the device has finished booting. This state is indicated to the user by turning on the device's blue "ready" LED.

3.3.2 Client software

As described so far, we have a wearable device that acquires biosignals and embedded software to transmit those signals to a remote computer. All that remains to complete the platform is some client software running on that computer to receive and analyse the incoming signals.

The purpose of the client software is to do everything that a user of the Elapse platform may need to do except for the actual signal processing. That is, it needs to handle communication with the device, decoding of the incoming data, the ability to save and load data, and provide a graphical user interface (GUI). It should also provide a framework into which the user can load signal processing plug-ins for their specific application.

As a point of clarification, the client software is described as a "framework" throughout this document. To quote Pree [1994, p. 152]:

> Application frameworks consist of ready-to-use and semi-finished building blocks. The overall architecture is predefined as well. Producing specific applications usually means to adjust building blocks to specific needs by overriding some methods in subclasses.

In our case, the Elapse framework provides the core application functionality and connects several signal processing "blocks" together to produce a cognitive state/event detection application. The user of this framework can implement a specific application (e.g., lapse detection) by providing suitable signal processing blocks.

3.3.2.1 Signal processing pipeline

At the centre of the Elapse framework is a configurable signal processing pipeline designed to encapsulate a wide variety of signal processing and classification algorithms. The pipeline has five stages, illustrated in Figure 3.14:

Data source Receives the data transmitted from the device and outputs a byte stream for each signal type. As described later, it may alternatively load saved data from a file.

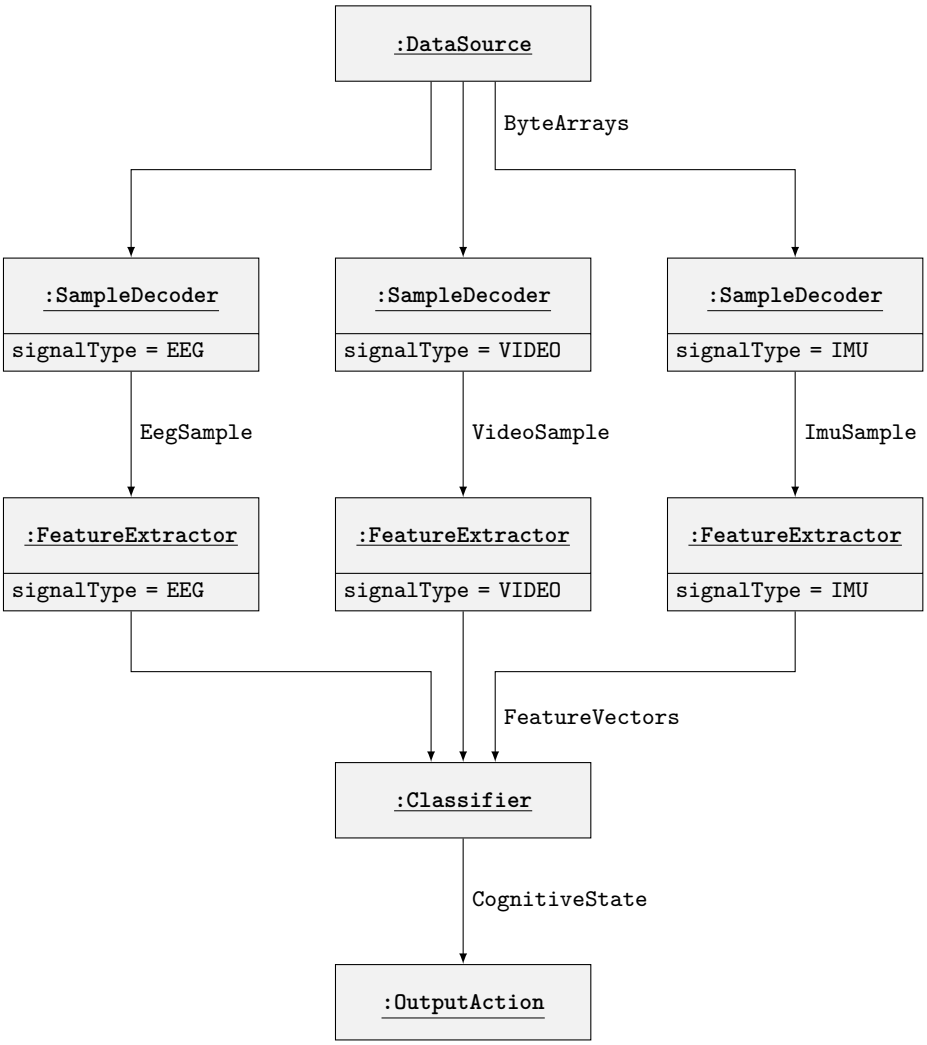


Figure 3.14 Object diagram of the Elapse signal processing pipeline. Arrows represent data flow and are annotated with the data types being passed (cf. Figure 3.15).

Sample decoder Decodes a byte stream into `Samples`. For example, the video sample decoder decodes the compressed video byte stream into video frames. There is one `SampleDecoder` per signal type.

Feature extractor Extracts meaningful features from a sequence of `Samples`. For example, an EEG feature extractor could calculate the power in the standard EEG bands over a window of `Samples`. There is one `FeatureExtractor` per signal type.

Classifier Classifies the user's cognitive state based on the latest set of `FeatureVectors`. There is one `Classifier` that operates on the `FeatureVectors` from all of the signal types.

Output action Takes some action based on the classified cognitive state. For example, this could sound an alarm or send a message to a fleet management system.

In addition to these elements there is a `DataSink` which can capture data at any point in the pipeline and save it to disk.

Each element in the pipeline knows nothing about any of the other elements. All inter-element interactions are mediated by the `Pipeline` object. This loose coupling is achieved using Qt's "signals and slots" mechanism. Signals and slots are essentially a generalization of callbacks or the observer pattern [Gamma et al. 1994]; a signal is an observable event and a slot is an event handler. Signals may be connected to slots so that when an object emits a signal all connected slots are executed. In this case, each element in the pipeline implements an input slot and an output signal and the `Pipeline` object connects these together per Figure 3.14. The connections between pipeline elements are made indirectly by queueing signals in Qt's event loop. This way, elements are free to use background worker threads internally and emit output signals from those threads, but the input slot to the following element will always be called from the main thread.

The following sections describe each element of the pipeline in more detail.

Data source

The purpose of the `DataSource` element is to push data into the pipeline. The default implementation listens for incoming data on three UDP ports, one per biosignal type as described in Section 3.3.1.3. When data is received on any port, the `DataSource` emits the received array of bytes via the Qt signal corresponding to the biosignal type, e.g., `eegReady(bytes)`. It is also possible to implement offline data sources which, instead of receiving data from the device, load previously saved data from disk. This is particularly useful for research situations, where data can be saved during a study and "re-played" later to experiment with different signal processing algorithms.

Sample decoder

Each `SampleDecoder` receives the byte arrays emitted by the `DataSource` corresponding to one signal type. Its purpose is to decode those bytes into structures representing the samples captured by the device. These structures are defined in Figure 3.15. All `Samples` have a timestamp recording the time at which that sample was captured by the device, though the other fields are specific to each signal type. The `EegSample` contains a vector of EEG channel readings in microvolts, as well as a sequence number which is added by the EEG AFE device driver for detecting missed samples. The `VideoSample` contains the dimensions of the video frame as well as the 8-bit greyscale frame data. The `ImuSample` contains two three-dimensional vectors of the accelerometer readings in metres per second squared and gyroscope readings in degrees per second. Depending on how the device encodes each type of data, a `SampleDecoder` may emit multiple `Samples` for each byte array it receives from the `DataSource`.

Feature extractor

The `FeatureExtractor` for each signal type must extract meaningful features from a sequence of `Samples` emitted by a `SampleDecoder`. Most interesting features in the signals we are using involve changes over time, so the `FeatureExtractor` must be able to analyse a sequence of several `Samples`. To achieve this, a `BaseFeatureExtractor` class is provided for users to inherit from that implements a windowing algorithm internally. When the input slot of a subclass of `BaseFeatureExtractor` is called with a new sample, the sample gets pushed into an internal queue. When a sample arrives with a timestamp beyond the end of the current window, the virtual `analyseWindow()` method is called with the contents of the queue. This method must extract features from the queued data (e.g., PERCLOS, blink duration) and emit them

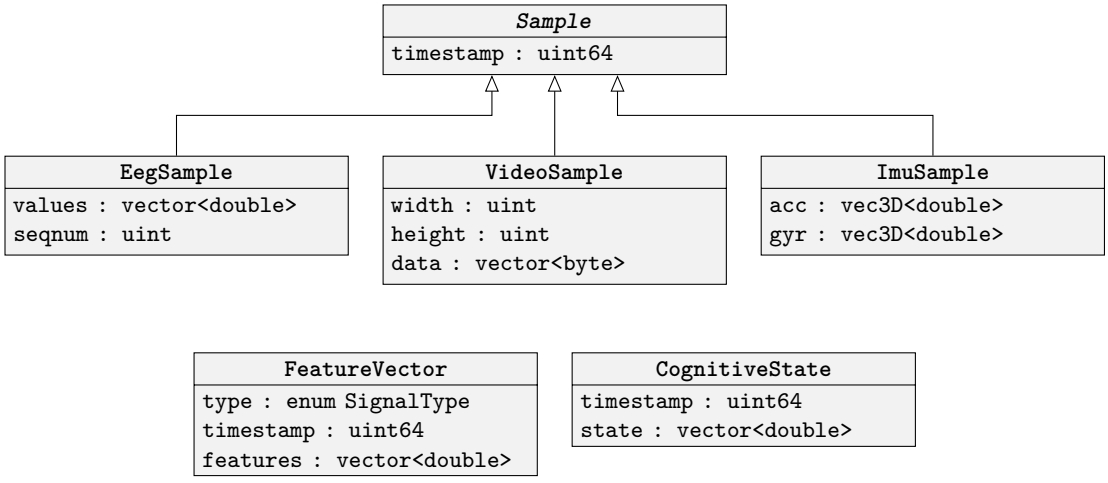


Figure 3.15 Class diagram of the types passed through the pipeline.

Algorithm 3.1 `BaseFeatureExtractor` windowing algorithm.

```
function BaseFeatureExtractor::onSample(sample)
    windowEnd ← windowStart + windowLength
    nextWindowStart ← windowStart + windowStep

    » If we've reached the end of the current window
    if sample.timestamp > windowEnd then
        » then extract features from the window of stored samples
        features ← analyseWindow(window)
        emit features
        » and update the window start point
        windowStart ← nextWindowStart
        RemoveDataBefore(window, windowStart)
    end if

    » otherwise store the sample for later.
    Append(window, sample)
end function
```

as a `FeatureVector`. The starting point of the window is then updated, data in the queue from before that point is removed, and analysis continues. The length and step size of the window are independently configurable so it is possible to have, for example, a 2 s window that moves in steps of 500 ms. This process is perhaps easier to understand with reference to the pseudocode in Algorithm 3.1.

The `FeatureVector` class contains a vector of floating-point values. Both the number and meaning of these values is up to the implementor of the `FeatureExtractor` to decide. Additionally, a `FeatureVector` contains the timestamp of the first sample in the window from which it was extracted and a tag identifying the type of signal it was extracted from (Figure 3.15).

Classifier

The `Classifier` receives the `FeatureVectors` emitted by all of the `FeatureExtractors` and classifies the `CognitiveState` of the wearer. The `FeatureExtractors` may take different amounts of time to analyse their windows of samples, so a `BaseClassifier` class is provided which hides this complexity from anyone implementing a `Classifier`. When the input slot of a `BaseClassifier` is called with a new `FeatureVector`, it matches up that `FeatureVector` with others that it has received recently. If it has received one `FeatureVector` for each signal type in a given time window, the virtual `classify()` method is called with those `FeatureVectors`.

Much like a `FeatureVector`, the `CognitiveState` class contains a vector of floating-point values, with the number and meaning left up to the implementor

of the Classifier (Figure 3.15). It also contains the timestamp of the first sample in the window from which the corresponding FeatureVectors were extracted.

Output action

The OutputAction can take some action based on the classified CognitiveState of the user. This could range from displaying a message, to sounding an alarm, to disabling machinery, or whatever is appropriate for a given application.

Data sink

The DataSink observes the connections between all of the other elements in the pipeline. DataSink implementations may choose to save data at any of these points—raw bytes, Samples, FeatureVectors, CognitiveState—in some meaningful format on disk. With this mechanism it is possible to, for example, dump all raw data, save EEG in a Matlab file, or log a text file of classified CognitiveStates. While this is useful for keeping logs and doing further offline analysis, it is particularly useful when combined with a DataSource that is able to read this data back into the pipeline again to enable further investigation within the Elapse framework itself.

3.3.2.2 Plug-in architecture

All of the elements in the signal processing pipeline are provided by plug-ins. Plug-ins are discovered dynamically at run-time and the user can select which elements to use interactively through the GUI. A core plug-in is provided with the framework which provides standard DataSources and SampleDecoders. These elements should not generally need to be overridden by users since they are tied to the implementation of the device's embedded software. The core plug-in also provides placeholder implementations for the other elements which simply ignore their inputs and produce dummy outputs.

Using plug-ins to provide signal processing algorithms is the key to the flexibility of the Elapse platform. It allows the framework to provide all of the common functionality—device connectivity, user interface, data I/O, etc.—while users of the platform need only implement the signal processing code specific to their application. Additionally, it makes experimenting with different combinations of feature extractors and classifiers easy.

For maximum flexibility in the implementation of plug-ins, the plug-in management code has been designed to also support plug-ins written in languages other than C++. This means that, given appropriate language bindings for the Elapse base classes, it would be possible to write signal processing algorithms in high-level languages such

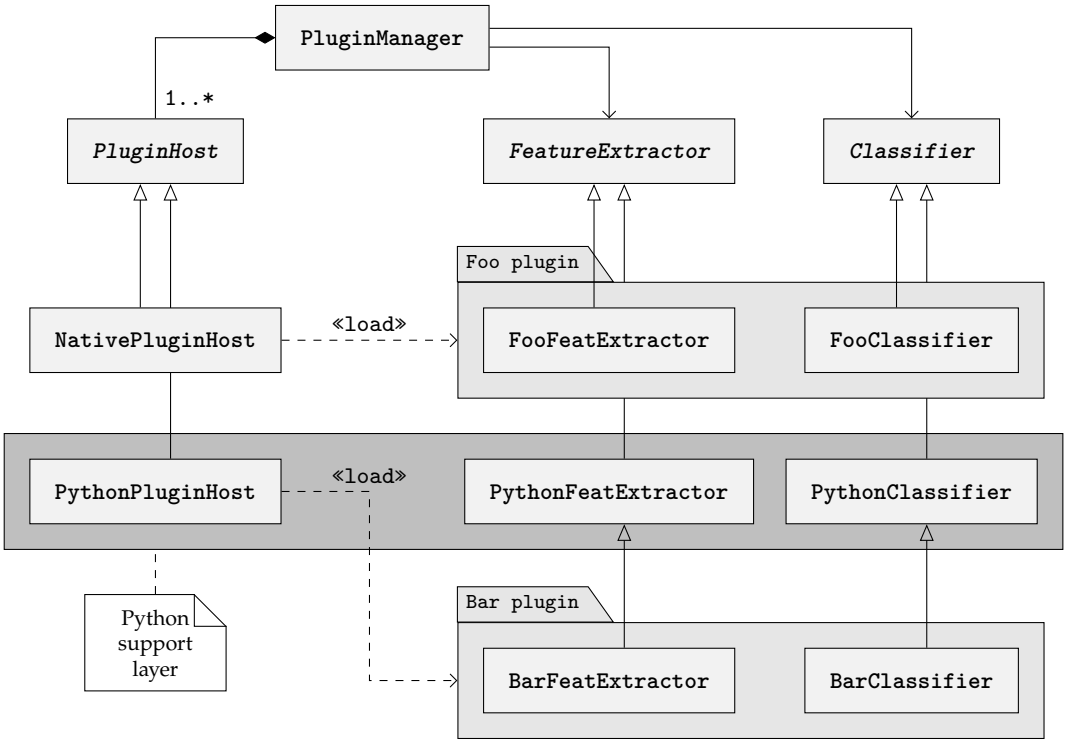


Figure 3.16 Class diagram of the Elapse plug-in manager. `PluginHost` is an abstract factory that instantiates classes provided by plug-ins. "Foo plugin" is implemented in C++ and "Bar plugin" is implemented in Python.

as Python, Matlab, or Julia.[20] The `PluginManager` class uses the abstract factory pattern [Gamma et al. 1994] to create instances of classes provided by plug-ins. A diagram of this structure is given in Figure 3.16. The plug-in manager has one or more plug-in hosts, each of which is able to load plug-ins written in a particular language and instantiate the classes provided by those plug-ins. For example, the `NativePluginHost` loads C++ shared libraries and instantiates C++ classes, and the `PythonPluginHost` starts a Python interpreter and imports Python modules. Each plug-in contains one or more implementations of Elapse base classes, e.g., `FeatureExtractor`, `Classifier`, as well as some metadata identifying those classes. For C++ plug-ins, the classes inherit directly from the Elapse base classes, while other languages require some glue code (bindings) to translate to and from C++. To implement support for a new language, it is only necessary to implement a new plug-in host and bindings for the Elapse base classes, as the shaded region in Figure 3.16 shows for Python. At the time of writing, the `NativePluginHost` and `PythonPluginHost` are implemented, supporting plug-ins written in C++ and Python respectively. The Python host makes it possible to write signal processing algorithms using the extensive SciPy[21] numerical computing libraries.

[20] `http://julialang.org`
[21] `http://www.scipy.org`

Binary compatibility

When designing the API for a shared library, it is important to consider the issue of binary compatibility. That is, if a new version of the library is released, will software compiled against the previous version work with the new version without needing to be recompiled? In our case, it is desirable that plug-ins compiled against one version of the Elapse framework continue to work without recompilation if the framework is updated. To maintain a stable application binary interface (ABI), there are several types of changes that must not be made to public interfaces, such as changing method parameters and adding class members. One common technique to help with achieving this, and what is used in the Elapse element base classes, is the pointer-to-implementation (pimpl) idiom. Using this approach, all private data and methods are moved to a private class which is hidden behind an opaque pointer in the base class. Data members and methods can then be freely added, removed, or changed in the private class while the public interface remains unchanged. External code can never instantiate or directly access the modified private class, so ABI compatibility is maintained.

3.3.2.3 Core application

The Elapse client application ties together the signal processing pipeline and plug-in management code, handles communication with the device, and provides a graphical user interface (GUI). A diagram of the structure of the application is given in Figure 3.17. The following paragraphs describe a typical session using the program, outlining both the user's perspective and how it operates internally.

When the user first launches the application they must choose a set of signal processing elements to work with. Clicking the *Load Elements* button launches the `PluginManager`'s plug-in selection dialog (Figure 3.18). This window displays a list of the available implementations for each type of pipeline element and allows the user to select which ones to load from plug-ins. Once the user has made their selection, the `PluginManager` creates an instance of each selected element class (Section 3.3.2.2) and returns them as an `ElementSet`. This is simply a container holding a complete set of pipeline elements. The `ElementSet` is passed to the `Pipeline` object which connects the corresponding signals and slots between the elements (Section 3.3.2.1). The selected set of elements is also saved to the application's configuration file so that it can be loaded automatically the next time the application is launched.

Having loaded a set of elements from plug-ins, the user can click *Connect* to connect to the device. The client application has a `DeviceProxy` which is a transparent proxy object for the configuration interface on the device (Section 3.3.1.4). That is, when a method is called on the `DeviceProxy`, the call is passed to the device over

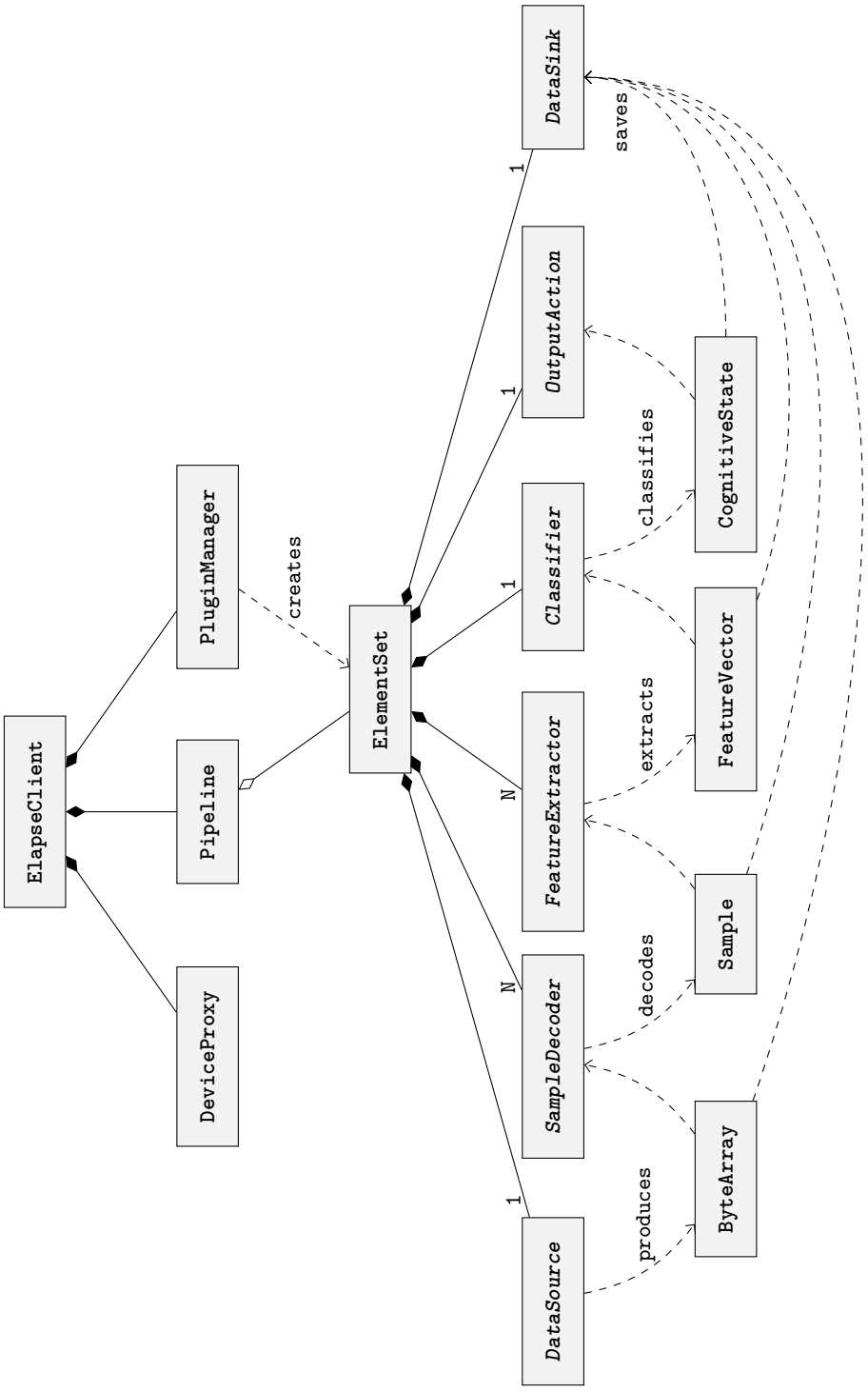


Figure 3.17 High-level class diagram of the Elapse framework.

DBus and the result returned to the caller as if it had been a normal method call. When the user clicks *Connect*, this proxy makes its initial connection to the device.

If the user chose to use an `OfflineDataSource`, this connection happens a little differently to normal. There are two types of data that must be passed from the device to the client: the hardware configuration and the captured biosignals. Ordinarily, the `DeviceProxy` reads the configuration from an Elapse device over DBus and the device transmits data from its sensors to a `DataSource`, as in Figure 3.20a. When using an `OfflineDataSource`, however, both configuration data and signal data are loaded from a file. There is no device to connect to in this case. Instead, the `OfflineDataSource` has an adapter which makes it look like a `Device`. It exposes this adapter on the local DBus session bus so that when the `DeviceProxy` attempts to read the hardware configuration, it actually gets read from the `OfflineDataSource`'s data file (Figure 3.20b). By using this approach, the only special case that needs to be handled to differentiate between online and offline `DataSources` is within the `DeviceProxy`—it needs to know whether to connect to a remote or a local DBus server. No other code within the Elapse framework needs to know whether the `DataSource` is online or offline because the same interfaces are implemented in both cases.

If the application successfully connects to the device, the GUI shows something similar to Figure 3.19. All of the widgets in the main area of the window are provided by the pipeline elements loaded from plug-ins. Pipeline elements may choose to implement the `Displayable` interface which simply exposes a GUI widget representing the state of the element in some way. In Figure 3.19, the EEG strip-chart, eye video, and 3D head widget are provided by the default `SampleDecoders` in the core plug-in, and display the raw data as it is decoded. The battery gauge widget is a special case—it is provided by the application itself. All of these widgets may be rearranged and resized within the window to suit the user's taste.

Depending on the `DataSink` that the user selected, it may be necessary to take one more step before beginning data capture. `DataSinks` support the concept of "capture info"—arbitrary metadata associated with a captured data set. This could be as simple as selecting a filename for the saved data, or as complex as recording the test conditions and participant ID for a research study. The capture info system is flexible, supporting `DataSinks` that never require capture info, that only require info once, that require new info for every capture, and that are able to load info from a configuration file. The user can click the *Set capture info* button to enter any required information, or if they attempt to start data capture without entering required information they will be prompted for it then.

At this point the device's hardware gets configured. This happens in two stages. Firstly, any hardware settings that the user wishes to override are read from the application's configuration file. These are used to configure the corresponding properties

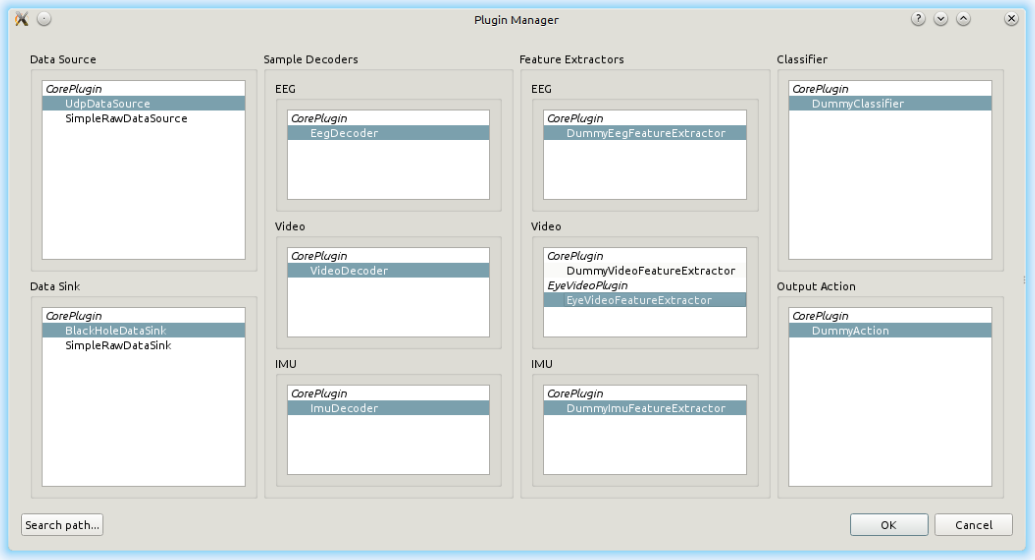


Figure 3.18 The Elapse plug-in selection window.

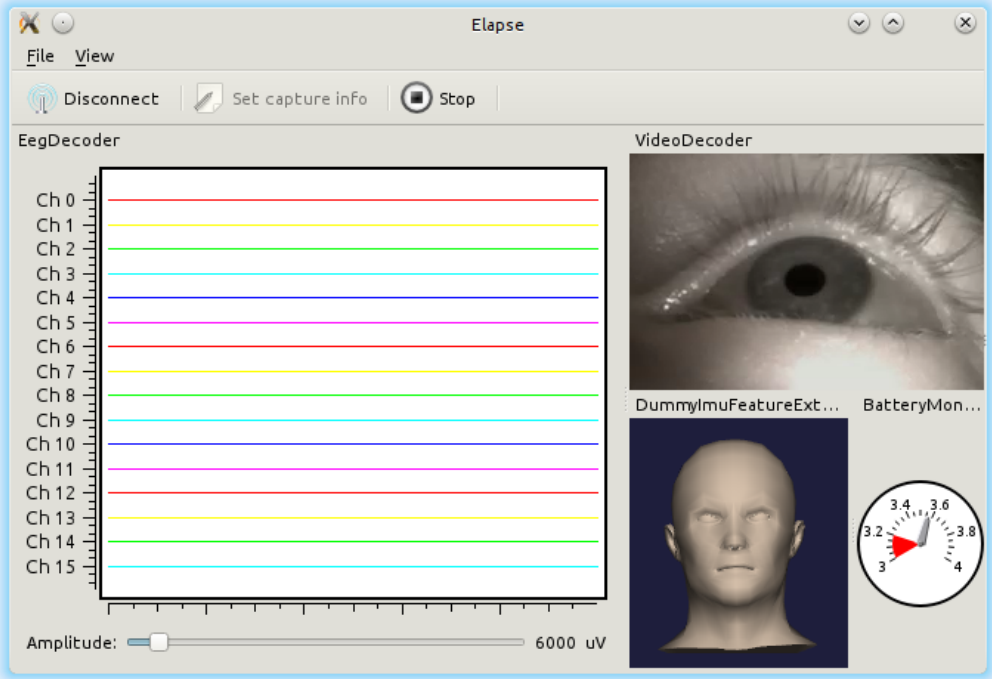
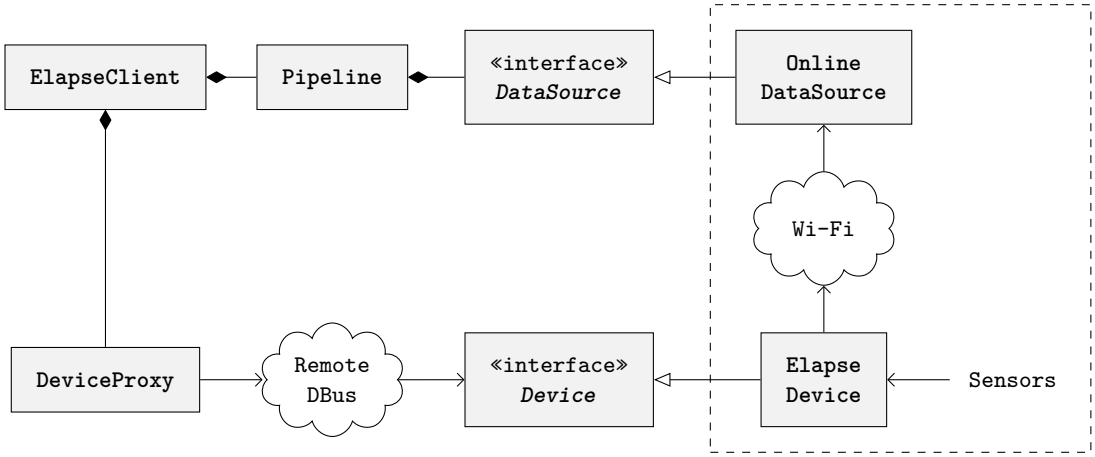
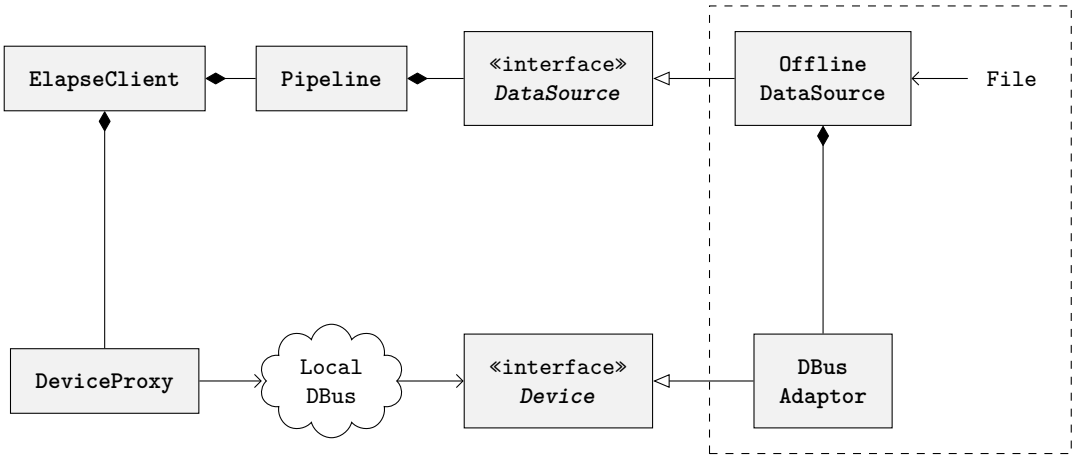


Figure 3.19 The Elapse main window during operation.



(a) Normally the `DeviceProxy` reads the hardware configuration from an Elapse device over DBus. The device reads data from its sensors and transmits it to the `DataSource`.



(b) When using an `OfflineDataSource`, both the configuration and sensor data are read from a file. The `OfflineDataSource` exposes the configuration to the `DeviceProxy` via a DBus adaptor.

Figure 3.20 Comparison of online and offline data sources. Most of the Elapse framework is oblivious to whether the data is received from a device or loaded from a file.

of the device's sensors. Secondly, the complete hardware configuration is read back from the device. This is passed to each element in the `Pipeline` so that they can configure themselves as necessary. For example, the EEG `SampleDecoder` needs to know the gain of each channel in order to correctly decode the measured voltages from the incoming byte stream. Similarly, the `DataSink` can save the configuration to disk alongside the captured data for later reference.

To start capturing data, the users clicks the *Capture* button. This triggers the server to start acquiring and transmitting data, and starts the `Pipeline` receiving and processing data. The `Pipeline` object presents a simplified facade for managing the pipeline elements. It provides the application with simple methods like `start()` and `stop()` which internally handle starting/stopping/resetting the individual pipeline elements. During data capture the `DataSource` receives data from the device and feeds it through the signal processing pipeline as described in Section 3.3.2.1. As data passes through each element it can update its GUI widget if it is `Displayable`, for example, overlaying the detected pupil region on the eye video or plotting a spectrogram of the EEG. When the data capture has finished, the user can click *Stop* and either begin a new capture or *Disconnect* from the device.

3.3.2.4 Developer tools

Particular effort was put into providing tools to assist new developers working on both the Elapse framework and signal processing plug-ins. Comprehensive API documentation is extracted from the source code as browsable HTML, including class and state diagrams. Example code is also provided for implementing plug-ins in both C++ and Python. Much of the Elapse framework has unit tests for regression testing. Time constraints have prevented better test coverage, but the most important classes have associated test suites.

3.4 SUMMARY

The Elapse platform provides an extensible framework for developing real-time cognitive monitoring applications. The head-mounted device captures EEG, eye video, and head motion, and transmits these signals to a remote computer. Software on the remote computer receives the data and feeds it through a signal processing pipeline. Users of the framework may implement custom feature extraction and classification algorithms as plug-ins, along with custom file formats for saving data, and actions to take based on the classified cognitive state. Documentation and examples are provided to assist those developing signal processing plug-ins.

Chapter 4

LAPSE DETECTION SOFTWARE

Although the primary focus of this project was on developing the Elapse platform itself and not on the actual detection of lapses or any other specific cognitive states, some signal processing and classification algorithms needed to be implemented as proof of concept. Other students in NeuroTech are working on EEG feature extraction and classification algorithms, so this project focused primarily on developing an eye-video feature extraction algorithm. The EEG and IMU feature extractors described here are deliberately simple, intended only to demonstrate that is is possible to extract meaningful information from the data that the device acquires.

4.1 EYE VIDEO

Many eye-related parameters have been found to relate to drowsiness and lapses. PERCLOS, the percentage of time that the eyes are more than 80 % closed during a one-minute window, is correlated with lapses of responsiveness on the psychomotor vigilance task [Wierwille and Ellsworth 1994]. The durations of the closing, dwell, and reopening phases of eye closure increase and become more variable with drowsiness [Schleicher et al. 2008, Tucker and Johns 2005]. Similarly, the ratio of the amplitude of eye closure to eyelid speed is correlated with drowsiness [Johns et al. 2007], as is the proportion of blinks that are "long" (300–500 ms) [Caffier et al. 2003]. Lüdtke et al. [1998] found large low-frequency fluctuations in the diameter of the pupil when a subject feels sleepy, and these are possibly preceded by a gradual decrease in pupil diameter occurring over several minutes [Nishiyama et al. 2007]. A decrease in the gain of the vestibulo-ocular reflex (reflexive compensatory eye movement in response to head movement) has been observed before subjects reported feeling drowsy [Hirata et al. 2009]. For detecting non-drowsiness-related lapses, gaze direction may be an indicator of diverted attention if the subject is carrying out a task that requires visual attention in a known direction. Measurements of gaze direction also provide information about the duration of visual fixations which may be related to drowsiness [Schleicher et al. 2008].

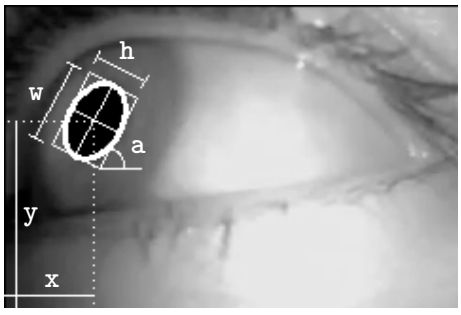
Some of these parameters cannot be measured solely from video of one eye. For example, measuring VOR gain requires knowledge of the relative motion of the visual target, and gaze direction can only be measured accurately by observing both eyes and knowing the orientation of the head.

Most of the remaining parameters, however, can be calculated by locating the boundary of the visible pupil region in video of the eye. The proportion of the pupil that is visible can be used to calculate PERCLOS, the number of frames in which the pupil is partly covered can be used to approximate the speed of eye closure, and the size of the visible pupil region can be used to calculate the pupil diameter. The eye-video processing algorithm described in this section aims to locate the boundary of the pupil accurately so that all of these parameters can be measured from the eye video.

The following section describes the pupil localization algorithm which has been developed, followed by a comparison to alternative methods.

4.1.1 Pupil localization algorithm

The algorithm developed for locating the pupil is based on flood-filling about a dark seed point. Flood-filling is a common process for locating connected components in an image which we have applied to pupil localization. The output of the process is an instance of the data structure given in Figure 4.1b—the boundary of the pupil is represented as a rotated ellipse (Figure 4.1a) and the state of eye closure is classified as open/partial/closed. The parameters defining the pupil ellipse are given in image coordinates and measured in pixels. The criteria for assigning each of the eye states are explained later in this section.

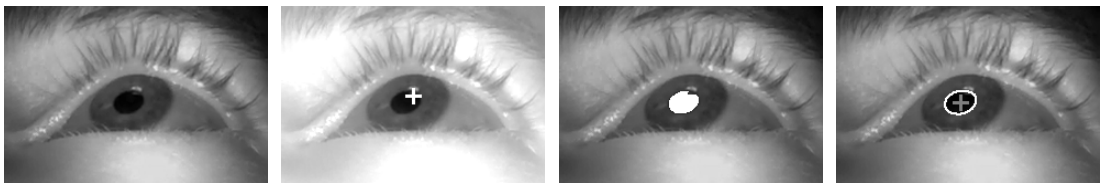


(a) Definition of pupil ellipse parameters.

```
struct Pupil {
    struct Ellipse {
        int x, y;
        int w, h;
        float angle;
    } boundary;
    enum { Open, Partial, Closed } state;
};
```

(b) Software representation of a pupil.

Figure 4.1 Representation of a pupil. The image on the left illustrates the definition of the parameters in the code on the right.



(a) The captured frame. (b) Weighted by distance from the centre, global minimum marked. (c) Flood-filled pupil region. (d) Pupil ellipse and seed point for next frame.

Figure 4.2 Pupil localization algorithm: flood-filling about a dark seed point.

Initialization

In the very first frame of video, a dark point near the centre of the frame is defined as the seed point. To exclude dark areas which occur, for instance, where the face curves back away from the IR LED, the brightness of each pixel is linearly weighted by its distance from the centre of the image. That is, for each pixel a value proportional to its distance from the centre of the frame is added to its intensity. The seed point is defined as the global minimum of the resulting weighted image (Figure 4.2b).

The gradient of this weighting function is defined such that at the edge of the frame half of the full-scale brightness range is added. This value was experimentally determined to provide a balance between brightening dark areas near the edge of the frame enough so that they do not pose a problem and avoiding saturating too much of the image.

Threshold

To determine whether the eye is completely closed, the brightness at the seed point is compared to a fixed threshold. If the brightness at the seed point exceeds the threshold, we conclude that it cannot be part of the pupil. Further, because the seed point is the darkest point near where the pupil is likely to be, we conclude that the pupil is not visible. In this case, the eye is considered to be closed and the rest of the pupil localization process is skipped.

It is possible to use a fixed threshold for this comparison because the camera's automatic black-level, exposure, and gain controls ensure that the dynamic range of every frame is within fixed limits. That is, the illumination arrangement ensures that the pupil appears black, the camera's automatic controls ensure that "black" lies within a fixed range of digital brightness values, and so the image processing algorithms can use a fixed threshold. The actual value of the threshold was chosen by manually examining several frames captured in a range of lighting conditions and selecting a threshold based on the average brightness of the skin.

Pupil shape

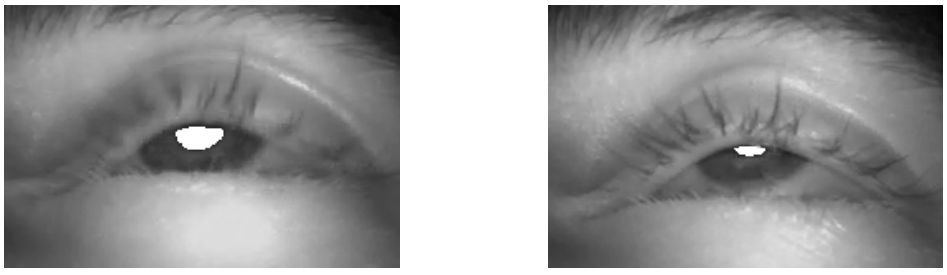
The shape of the pupil region is defined by flood-filling about the seed point (Figure 4.2c). This process starts at a point and recursively fills all adjacent pixels which have a brightness within a specified range. This range can be fixed, in which case the brightness of the adjacent pixel is compared to the seed point, or floating, in which case it is compared to the current pixel. In this case, a fixed range relative to the seed point is used so that any blurring of the pupil–iris boundary resulting from a slightly out of focus image will not affect the extent of the fill. If a floating range was used, the flood could continue too far in a de-focused image because the difference between the brightness of adjacent pixels is reduced when the image is blurred. By using a fixed range flood fill, obtaining a correctly filled area relies on there being some contrast between the pupil and the iris but does not rely on a sharp edge delineating the two.

The pupil is considered to be partly covered under two conditions: *i)* if the widest point is close to the top of the filled region, or *ii)* if the width of the region is more than twice its height. An example of each of these cases is shown in Figure 4.3. If, by these criteria, the pupil is not partly covered, the boundary of the pupil is approximated by fitting an ellipse to the boundary of the flood-filled region (Figure 4.2d). This reduces the description of the pupil to five parameters: coordinates of the centre, lengths of major and minor axes, and rotation angle. The ellipse is fitted to the filled region using a direct least-squares method [Fitzgibbon et al. 1999].

Finally, the bounding ellipse and closure state of the pupil are returned. This information is passed back into the algorithm to be used as a starting point for the next frame.

Adjustment

When the next frame is ready for processing, an adjustment step takes place to optimize the location of the seed point before beginning the flood-fill. Since the eye can rotate between frames, the centre of the pupil ellipse from the previous frame may not be the best seed point from which to begin the flood-fill in the current frame.



(a) A partly covered pupil during a blink. (b) A proportionally wide pupil region.

Figure 4.3 Criteria for classifying the pupil as partly covered.

Instead, a window around the previous pupil centre is examined and the seed point is defined to be the darkest point therein. In this way, if the eye moved between frames such that the previous pupil centre now lay on a brighter area—perhaps from reflected ambient light or a stray eyelash—the flood-fill will still begin from a dark point. A seed point which accurately represents the brightness of the region of interest is important since the extent of the flood-fill is defined by brightness relative to the seed point. Without this adjustment, an unfortunately-placed glint could throw off the flood-fill process entirely. The size of the window that is searched is fixed at approximately half of the average size of the pupil.

Once the position of the seed point has been adjusted, the process continues from "Pupil shape" above. A pseudo-code representation of this process is given in Algorithm 4.1. Section 6.1 describes the methods and results of two experiments designed to test the performance and robustness of this pupil localization algorithm.

Algorithm 4.1 Pupil localization.

```
function LocatePupil(frame, prev) → pupil
    if prev = NULL then                                  ▷ This is the first frame
        seed ← FindCentralDarkPoint(frame)
    else
        seed ← FindNearbyDarkPoint(frame, prev.centre)
    end if

    if frame[seed] > T_closed then                ▷ Seed point is too bright to be the pupil
        pupil.state ← CLOSED
        pupil.boundary ← NULL
        return pupil
    end if

    region ← FloodFill(frame, seed, T_pupil)

    if HasFlatTop(region) or IsWide(region) then          ▷ Pupil is partly covered
        pupil.state ← PARTIAL
        pupil.boundary ← prev.boundary
    else                                                  ▷ Pupil is fully visible
        pupil.state ← OPEN
        pupil.boundary ← FitEllipse(region)
    end if

    return pupil
end function
```

4.1.2 Comparison with other approaches

The algorithm described in the previous section is the result of several iterations of development. In this section it is compared to some of the alternative approaches that were explored during that process, as well as some techniques from the literature.

The design of the CAMEL module introduces some constraints on the techniques that can be used to process the video that it produces. Because it is mounted well below the natural gaze direction of the eye, any approach that assumes that the pupil appears circular cannot be used [e.g., Lüdtke et al. 1998, Pérez et al. 2003]. Techniques originating from the field of biometrics assume high resolution images taken in controlled lighting conditions from directly in front of the eye [Daugman 2004]. Clearly the CAMEL module does not satisfy these assumptions.

Likewise, techniques relying on the bright pupil effect cannot be used because the CAMEL cannot produce a bright pupil. To induce the bright pupil effect, IR illumination must be provided coaxial with the camera (Section 3.2.4.2). This generally requires either a ring of IR LEDs around the camera lens [Ji and Yang 2002] or a beam splitter in front of the lens. The CAMEL module was designed to be as small as possible and so uses a single IR LED. This LED was initially mounted close to the camera lens in an attempt to induce the bright pupil effect but, because the module is mounted close to the eye, even this small distance limited the effect to only about one third of the pupil. It is not feasible to mount multiple IR LEDs around the camera, at least on the current version of the hardware, so the LED was moved further away from the camera to avoid the bright pupil effect entirely and instead rely on the dark pupil effect.

Gaze tracking is usually done by supplying an external IR source which causes a specular reflection on the surface of the eye. The orientation of the eye is measured by comparing the positions of either the reflections off the outer surface of the cornea and the inner surface of the lens (first and fourth Purkinje images), or of the first Purkinje image relative to the centre of the pupil [Crane and Steele 1985]. The relative position of these points is invariant to translation but changes with eye rotation. In the Elapse device, however, the lower eyelid blocks the reflection of the IR LED most of the time, except when looking sharply downwards. Measuring gaze by tracking Purkinje images, then, is not an option. It may be possible to approximate gaze direction by comparing the position of the pupil to the corners of the eye, the canthi, though this is not without difficulty—the canthi deform when the eye moves, so make poor reference points. Alternatively, it may be possible to back-project the detected pupil ellipse in two dimensions to the original circle in three dimensions to identify the orientation of the eye relative to the camera and so infer gaze direction. This is the approach taken by Nishino and Nayar [2004], although they use the limbus rather than the pupil. The geometry involved has inherent ambiguities but these could perhaps be overcome

with some well-chosen heuristics. This geometrical approach is probably the most promising for future development of gaze estimation from pupil shape.

Many pupil detection techniques treat eye closure as noise to be filtered out or model it very simply as a binary open/closed state. For example, Li et al. [2005] attempt to fit an ellipse to the pupil in every frame and if the fit fails to converge they assume the eye is closed. Świrski et al. [2012] simply exclude frames in which the pupil is covered from their dataset. Without explicitly taking eye closure into account in the pupil detection process, it is difficult to differentiate between a change in pupil size and a change in the degree of eye closure [Lüdtke et al. 1998]. Because the Elapse device is intended to be used for lapse detection, eye closure is an important parameter and must be measured as accurately as possible. While there is certainly room for improvement over the algorithm described in the previous section, the criteria for detecting partial eye closure shown in Figure 4.3 at least provide some more information than just whether the eye is open or closed.

The pupil localization algorithm developed here is simple. The algorithm was developed before the two-part device + remote computer design (Section 3.1) was chosen for the Elapse platform. At that stage we were still considering the possibility of running the video processing software on the head-mounted device. Because of the limited capacity of the device's processor, the algorithm was designed to be as simple as possible while still producing acceptable results. This simplicity is justified below by comparing the flood-fill approach to two other classes of shape-fitting algorithms—searching and voting.

Search-based shape fitting algorithms include the Hough transform [Duda and Hart 1972] and variations of Daugman's integro-differential operator [Daugman 2004, Nishino and Nayar 2004]. The characteristic feature of these techniques is the process of searching for the combination of parameters which produce the best fitting shape. Take, for instance, a circle given by the equation $(x - x_0)^2 + (y - y_0)^2 = r^2$. The Hough transform takes a binary edge image and, for each edge pixel (x, y), searches the (x_0, y_0, r) parameter space for triples that would produce a circle passing through (x, y). These triples are added to an accumulator and, after all edge points have been processed, the parameters defining the "best" circle are given by the coordinates of the largest accumulated value. The integro-differential operator searches for the (x_0, y_0, r) triple that maximizes

$$\max_{r,x_0,y_0} \left| G * \frac{\partial}{\partial r} \oint_{r,x_0,y_0} \frac{I(x,y)}{2\pi r} ds \right| \tag{4.1}$$

This finds the mean integral of the image I along a circular arc ds of radius r about (x_0, y_0), differentiates to find the radial gradient, smooths the result by convolution with a Gaussian kernel G, and then searches for the peak gradient. In other words, it finds the strongest circular edge in the image. Nishino and Nayar [2004] generalize this operator to ellipses. Both of these techniques involve an exhaustive search over the

n-dimensional parameter space which becomes prohibitively expensive as n increases; for an ellipse, $n = 5$. It is possible to reduce the necessary computation by limiting the span of each dimension, reducing the quantization resolution of each dimension, or doing both by iteratively refining the scope of the search. Nevertheless, these techniques are computationally intensive.

Voting-based shape fitting algorithms include those from Li et al. [2005] ("Starburst") and Świrski et al. [2012]. These are both based on the random sample consensus (RANSAC) method [Fischler and Bolles 1981]. RANSAC takes a small random sample of edge pixels from a binary edge image and fits the desired shape model to them. The "consensus set" is then formed by finding the remaining edge points that match the fitted model well according to some support function. This function could be as simple as the distance between the point and the fitted shape, or it may be something more complex that takes into account other properties of the original image like brightness or gradient. This process is repeated until a sufficiently large consensus set is found or a limit on the number of iterations is reached. This is more efficient than performing an exhaustive search of the parameter space although still involves fitting a potentially large number of shape models to edge points and evaluating support functions. Since it uses random sampling, the time it will take to satisfactorily fit a shape, or indeed whether it will manage to fit one at all, is non-deterministic. RANSAC's main advantage is its robustness to outliers in the edge image that it takes as input.

The pupil localization process described earlier does not use either of these computationally intensive techniques, instead using a simple least-squares fit. This is possible because of the nature of the edge image that flood-filling produces compared to the usual Canny edge detector or Starburst technique used in the previous examples. Typical edge detectors locate strong gradients in the image. These gradients may occur at the pupil boundary, or may be parts of eyelashes, eyelids, reflections, and so on. To locate shapes in such an edge image requires a technique that is robust to these outliers, such as RANSAC. The flood-fill process, however, produces a region with a continuous outline; there cannot be any outliers. This makes it possible to directly fit an ellipse to the boundary of the filled region using a least-squares approach, with no need for complex voting or searching algorithms. Of course, if the filled region is not elliptical this will still produce a poor fit. A non-elliptical region can occur for one of two reasons: the pupil is partly covered, or the threshold was inappropriate and the fill extended to non-pupil regions or did not fill the whole pupil. The first is accounted for by examining the shape of the filled region to detect partial eye closure. The second has not proven to be much of a problem in practice because the automatic image controls on the camera maintain the contrast between the pupil and the iris.

To summarize, by providing external IR illumination, exploiting the automatic image adjustment controls on the camera to maintain constant contrast, and using

an edge detection technique that avoids outliers, we have developed a simple and comparatively efficient method for locating the pupil.

4.2 OTHER FEATURE EXTRACTORS

As mentioned at the beginning of this chapter, the remaining signal processing elements are deliberately simplistic, intended only to demonstrate that it is possible to extract some meaningful information from the captured signals.

4.2.1 EEG

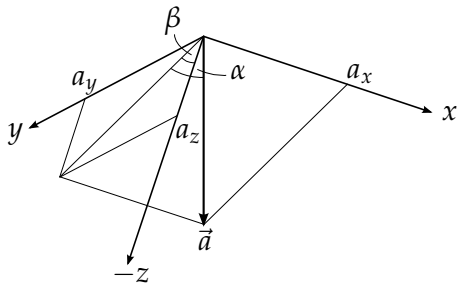
The EEG feature extractor calculates the power in each of the standard frequency bands described in Section 2.1.1. In each window of EEG data that the feature extractor receives (Section 3.3.2.1), the PSD is estimated by calculating the squared magnitude of the fast Fourier transform (FFT) of each channel. The mean power in each frequency band is calculated by integrating the estimated PSD over the relevant frequencies and dividing by the width of the band.

Changes in these features—the power in each band for each channel—are known to be correlated with cognitive changes. For example, Lin et al. [2005] found that the power in the 10–14 Hz band had a positive correlation with increased driving errors in a simulator. They also found strong correlations in more complex features, such as the log power spectrum of ICA components. Future development of the EEG feature extractor could explore the use of more complex features like these, but simple band power features suffice for the purposes of this project.

4.2.2 IMU

The purpose of the IMU is to measure the orientation of the head and its variation over time. Because of the problems with the IMU driver outlined in Section 3.3.1.2, the gyroscopes are currently disabled, leaving only the accelerometers active. Since accelerometers are unable to measure rotation about the gravitational field vector, it is only possible to measure the "tilt" of the head. That is, it is possible to measure rotations about the axes parallel to the ground plane (pitch and roll) but not about the vertical axis (yaw).

To estimate tilt from acceleration, it is necessary to assume that the measured acceleration is entirely due to gravity and points straight down, i.e., $\vec{a} \equiv [0\ 0\ -1]^T\,\mathrm{g}$ in world coordinates. This is a crude approximation, especially when the device is used in a moving vehicle, but is necessary for the mathematics to work. Figure 4.4 illustrates how the tilt is calculated from the measured acceleration vector. The angle



$$\tan\alpha = \frac{a_x}{\sqrt{a_y^2 + a_z^2}}$$

$$\tan\beta = \frac{a_y}{a_z}$$

Figure 4.4 Calculating head tilt from the acceleration vector using $Y_\alpha X_\beta$ extrinsic angles.

convention used is a rotation of α about the y axis followed by a rotation of β about the x axis.

The IMU feature extractor exposes a GUI widget to the Elapse application which depicts the measured head orientation as the orientation of a 3D model head (Figure 3.19).

4.3 SUMMARY

This chapter has presented a simple and computationally efficient method for locating the pupil in video of the eye. The shape of the pupil can be used to measure several parameters that are known to correlate with lapses and drowsiness. The method takes eye closure into account in its calculations in order to differentiate between changes in pupil diameter and changes in eye closure.

This chapter has also demonstrated simple EEG and IMU feature extractors as proof of concept. The EEG feature extractor calculates the power in each frequency band for each channel. The IMU feature extractor is currently limited to measuring head rotation about two axes because of hardware limitations.

Chapter 5

SYSTEM EVALUATION

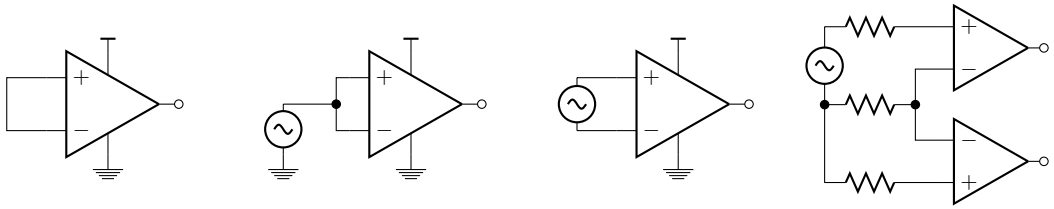
Having presented the development of the Elapse platform, it is necessary to demonstrate that the system meets the requirements set out in Chapter 3. This chapter describes the methods and results of several measurements taken to evaluate the performance of each of the device's subsystems, as well as the integration of these systems into a complete device. These measurements are intended to quantify the device's performance and demonstrate that it meets the requirements.

5.1 EEG SUBSYSTEM CHARACTERISTICS

5.1.1 Noise

When developing an acquisition system for signals with very low magnitude such as EEG, it is important to consider how much noise the acquisition system itself adds to the measurements. The majority of the noise in the system comes from thermal and flicker noise within the amplifiers and passive components, and quantization noise from the ADC. This noise can be measured by shorting the amplifier's inputs and measuring the output voltage (Figure 5.1a).

The EEG analogue front-end (AFE) noise was measured in two ways: *i)* the input multiplexer of the ADS1299 was configured to short both amplifier inputs to the mid-rail voltage (2.5 V), and *ii)* a measurement electrode was shorted to the reference electrode on the device itself. The first of these configurations measures the noise



(a) Noise (b) CMRR (c) Frequency response (d) Cross-talk

Figure 5.1 Circuits for measuring EEG AFE performance parameters.

contributed by the ADS1299 and the second includes any noise and interference from the rest of the system as well. Three 10 s periods of data were captured for every combination of the two input conditions and amplifier gains $G \in \{1, 2, 4, 12, 24\}$. The noise for each combination was found by calculating the root mean square (RMS) of the measured voltage for each of the three periods and selecting the median. All measurements were taken at room temperature.

Figure 5.2 shows the measured RMS voltage as a function of PGA gain. (Note that the gain is on a log scale for clarity.) As expected, the noise measured with the inputs internally shorted to mid-rail is close to the values given in the datasheet, differing by less than 6 % across the full range of gains. The expected decrease in input-referred noise with increasing gain is also clearly visible.

The measurements with an external short show more noise, most likely due to mains interference picked up over the length of the electrode cables. The magnitude of any mains interference would be constant, independent of the amplifier gain, since it is caused by an unchanging external source. RMS noise from multiple sources is not additive but combines as the root sum of squares, so the total noise tends to be close to the noise from the largest source. Thus, in this case, at low gains the amplifier noise dominates and the total noise is close to that measured with an internal short. At higher gains, the input-referred amplifier noise decreases and the 50 Hz interference dominates.

In summary, the EEG AFEs contribute 0.14 μV_{RMS} of input-referred noise to the system at $G = 24$, compared to a typical EEG signal amplitude of tens of microvolts. With the current arrangement of electrodes and cables on the device, mains interference raises the noise floor up to 0.36 µV.

5.1.2 Common-mode rejection

The common-mode rejection of a differential amplifier is a measure of how well it amplifies only the difference between its inputs and rejects any signal that is common to both inputs. This is an important parameter for EEG amplifiers because EEG typically has an amplitude of a few tens of microvolts, while the common-mode signal on the body from 50 Hz mains interference can be several orders of magnitude larger.

To measure the common-mode rejection ratio (CMRR) of the ADS1299, both inputs of one channel were driven with a 2 $V_{p\text{-}p}$ sinusoid centered about the mid-rail voltage of 2.5 V (Figure 5.1b). Ten seconds of data were captured for each combination of input frequencies $f \in [1, 100]$ Hz and amplifier gains $G \in \{1, 12, 24\}$. This was repeated with the buffer amplifier on the reference electrode both disabled and enabled. The CMRR was calculated as $20 \log_{10}(V_{out}/V_{in})$, where V_{out} is the peak-to-peak voltage of the captured signal and V_{in} the input from the signal generator. The results are plotted in Figure 5.3.

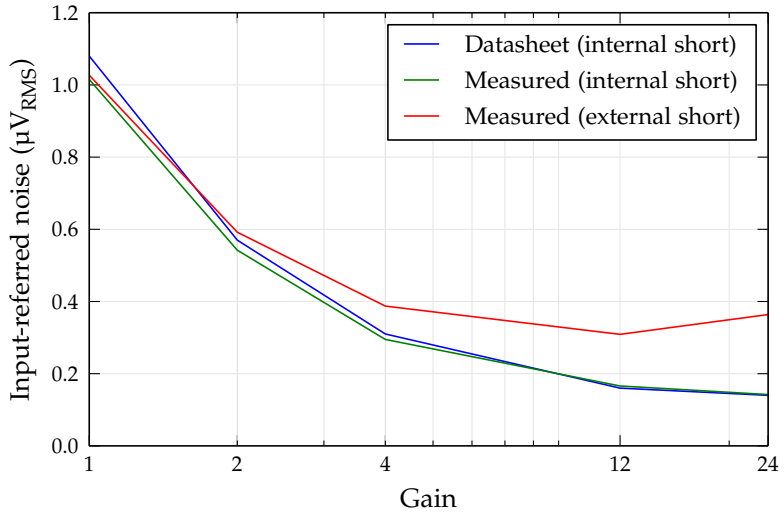


Figure 5.2 Measured EEG AFE noise compared to specified values.

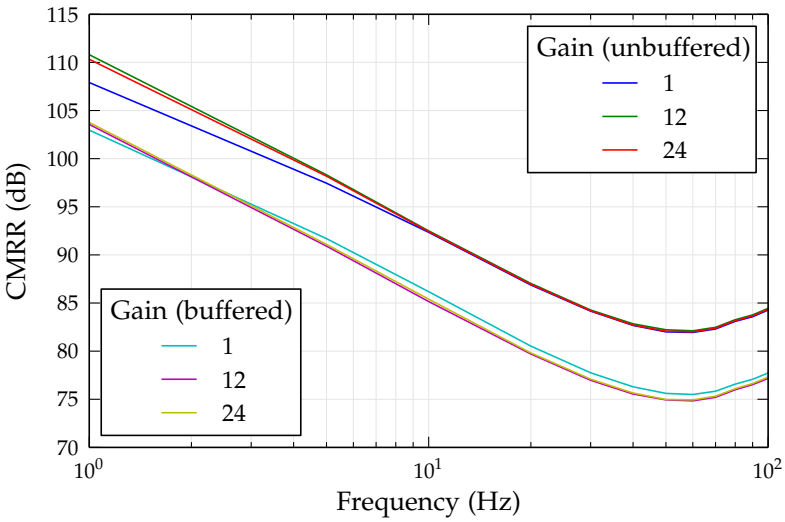


Figure 5.3 Measured EEG AFE common-mode rejection, with and without a buffer on the reference electrode.

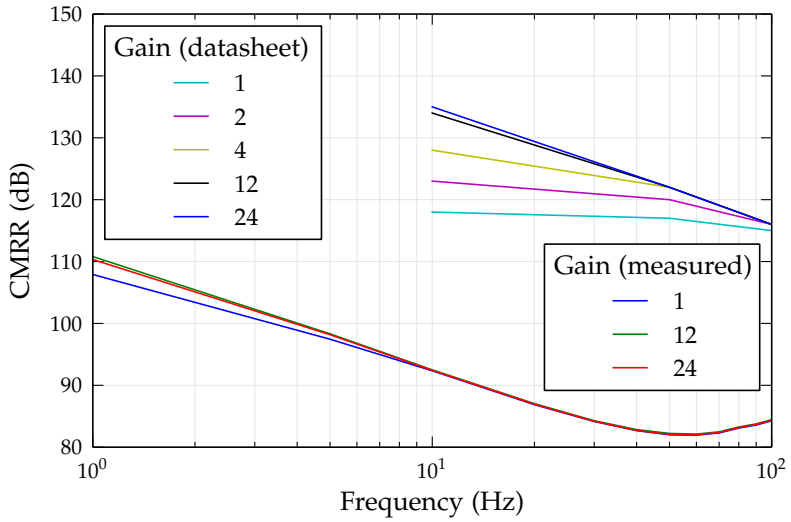


Figure 5.4 Measured common-mode rejection compared to the values specified in the ADS1299 datasheet.

With the reference buffer disabled, the CMRR spanned the range 82–110 dB depending on frequency and gain. With the buffer enabled it spanned 75–104 dB, a decrease of 7 dB. The decrease in CMRR when enabling the reference buffer was unexpected; enabling the buffer should increase the CMRR. That is, after all, the reason for putting a buffer on the reference electrode (Section 3.2.2.2). The cause of this unexpected behaviour is likely to be an error in the circuit design—the buffer amplifier was mistakenly placed before the input RC filter instead of after it. Because of this mistake, enabling the buffer actually increases the impedance mismatch between the reference electrode input and the measurement electrode inputs. Fixing this error should stop the CMRR decreasing when the buffer is enabled and hopefully cause it to increase as intended.

Figure 5.4 shows the difference between the CMRR values specified in the ADS1299 datasheet and those measured in this experiment. If these values are directly comparable then the CMRR is ∼32 dB lower than expected at $G = 24$ and $f = 10$ Hz, or equivalently, the amplitude of the common-mode signal in the output is 40× higher than expected. However, it is likely that the values can not be meaningfully compared. The circuit used to measure the CMRR values in the datasheet is not specified but it is safe to assume that it was designed to avoid any input impedances and imbalances. By contrast, when measuring the CMRR on the Elapse device, the input RC filters, electrode cables, and PCB layout have an effect on the measured values. These effects can not be avoided entirely in real systems, only minimized. Using a right-leg-drive circuit to bias the subject with an inverted common-mode signal (Section 3.2.2.2) can help to minimize the effect of a reduced CMRR by reducing the magnitude of the common mode signal, although this is not currently implemented on the device.

Two features of Figure 5.4 are unexplained: *i)* the upward trend in CMRR at frequencies above 50–60 Hz, and *ii)* the lack of spread in CMRR as a function of gain compared to the datasheet. Regarding the upward trend in CMRR, it is possible that the factors that are causing the CMRR to be worse than specified have some frequency-dependent effect and start to dominate at higher frequencies. The reason for the lack of spread in measured CMRR compared to the values specified in the datasheet is unknown.

5.1.3 Cross-talk

Cross-talk is an unwanted phenomenon where a signal in one channel, the "driven channel", is coupled into another channel, the "susceptible channel". It is quantified by the ratio of the amplitude of the unwanted signal in the susceptible channel to the amplitude of the original signal in the driven channel. The circuit shown in Figure 5.1d was used to measure the cross-talk of the Elapse device. One channel was driven with a 100 $\mathrm{mV_{p\text{-}p}}$, 10 Hz sinusoid, and the voltage induced in another channel was measured. 10 kΩ resistors were placed in series with the electrodes to simulate a typical electrode impedance and ensure that the driven and susceptible channels had matching impedances.

Most of the cross-talk was expected to occur along the length of the ribbon cable that connects the electrodes to the AFEs. To verify this, the signal generator was connected to channel 2 and channels 3 to 7 were measured in turn. Channel 2 was chosen as the driving channel to avoid any difference in cross-talk that may occur for the outermost conductor in a ribbon cable. The gain of the amplifiers for the driven and susceptible channels were set to $G = 1$ and all of the other channels were internally shorted to mid-rail and disabled. The results of these measurements are given in Table 5.1.

The ADS1299 datasheet states that the cross-talk is −110 dB at 50 Hz. This is very different to the values measured here. However, the datasheet gives no information about the circuit used to measure the cross-talk, the amplitude of the driving signal, or the physical arrangement of the components in the circuit. Without this information, we cannot know whether the values in the datasheet and these measured values are

Table 5.1 Measured EEG cross-talk.

Channel separation	Cross-talk (dB)
1	−17.53
2	−17.27
3	−17.32
4	−17.37
5	−17.36

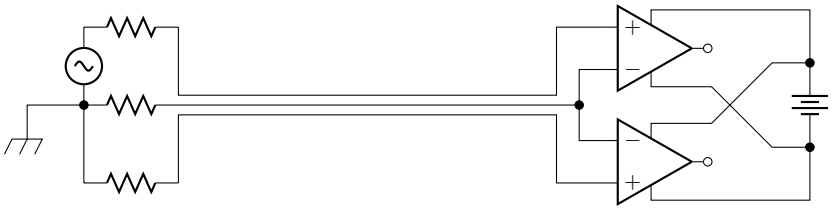


Figure 5.5 More accurate cross-talk measurement circuit diagram.

directly comparable. Figure 5.5 shows a more realistic diagram of the circuit used to measure the cross-talk which makes it easier to see where some differences may arise. Firstly, because the driving signal and the input resistors were connected to the electrodes on the helmet, there was about 30 cm of ribbon cable between the signal source and the amplifier inputs, creating some capacitive coupling between the input conductors. The ribbon cable has power and ground lines interspersed between each electrode conductor to reduce this coupling as much as possible. Secondly, it is possible that using a mains-powered signal generator with a battery-powered device contributed to the difference.

It is surprising that the measured cross-talk between channels did not change as a function of the separation between the channels. Despite each increase in channel separation representing a distance of three conductors on the ribbon cable (3.8 mm), the cross-talk remained constant at approximately -17.4 dB. The phase shift between the driven and induced voltages was also constant at $2\pi/5$ rad (72°), indicating a constant coupling capacitance. This may indicate that the cross-talk is not primarily occurring in the ribbon cable; perhaps it could be occurring on the PCB where there traces run close together near the AFE inputs. Although the measured cross-talk is considerably higher than the value stated in the datasheet, it is probably still a useful separation.

5.1.4 Frequency response

The frequency response of the EEG AFE was measured by connecting a signal generator between the reference electrode and one measurement electrode (Figure 5.1c). The signal generator was configured to output a 100 $\mathrm{mV_{p\text{-}p}}$ sinusoid at a range of frequencies between 1 Hz and 100 Hz. Ten seconds of data were captured at each frequency with PGA gain $G = 1$. The peak-to-peak voltage V_{out} was calculated by finding the difference between the median of the local maxima and the median of the local minima of the captured signal. A plot of magnitude ($V_{\mathrm{out}}/V_{\mathrm{in}}$) vs. frequency is given in Figure 5.6.

The theoretical frequency response can be calculated by taking the product of the transfer functions of the two filters in the system. The decimation stage of the $\Delta\Sigma$

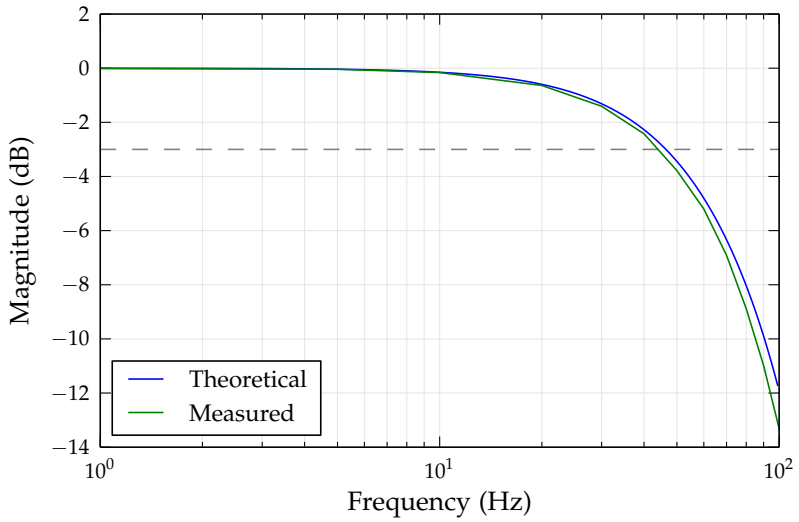


Figure 5.6 EEG AFE frequency response at a sampling rate of 250 Hz.

ADC consists of a third-order sinc filter with the frequency response

$$|H_{\text{sinc}}(f)| = \left| \frac{\sin \frac{N\pi f}{f_{\text{mod}}}}{N \sin \frac{\pi f}{f_{\text{mod}}}} \right|^3 \tag{5.1}$$

where $f_{\text{mod}} = 1024\,\text{kHz}$ is the modulator frequency and N is the decimation ratio. For an output data rate of 250 Hz, $N = 4096$ and the cut-off frequency $f_c = 65\,\text{Hz}$. The inputs have single-pole RC filters with frequency response

$$|H_{\text{RC}}(f)| = \frac{1}{\sqrt{1 + (2\pi f RC)^2}} \tag{5.2}$$

and cut-off frequency $f_c = 72\,\text{Hz}$. The combination of these two filters,

$$|H(f)| = |H_{\text{sinc}}(f)| \cdot |H_{\text{RC}}(f)| \tag{5.3}$$

has a cut-off frequency $f_c = 47\,\text{Hz}$ and is plotted in Figure 5.6.

The measured frequency response is close to the theoretical value; the cut-off frequency differs by only 2.3 Hz. The higher frequencies are approaching the Nyquist frequency, which explains the increasing negative error since there is less chance that a sample will be taken near the peak of the input waveform.

5.1.5 Electrode impedance

The contact impedance of EEG electrodes needs to be as low as possible in order to capture a good signal. In a clinical setting using wet electrodes, an EEG technician

will usually aim for an impedance of 5 kΩ or less at frequencies of 10–30 Hz [Duffy et al. 1989, p. 49]. When using dry contact electrodes, though, an impedance on the order of 100 kΩ is often considered acceptable.

To ensure that the electrodes on the Elapse device make good contact with the scalp, it would be useful to measure the electrode impedance while the user is donning the helmet. The ADS1299 AFE has a hardware module to enable just this; by driving a known alternating current through two electrodes and measuring the voltage across them, the impedance can be inferred. Unfortunately this feature does not function correctly on the Elapse hardware—connecting a resistor between two electrodes and measuring its impedance gave nonsense results. With a 10 kΩ resistor the measured impedance was correct but both larger and smaller resistors also measured close to 10 kΩ. One possible explanation for this error is that the current source has been damaged by unintentionally short-circuiting the two driven electrodes.

As a work-around, an external impedance meter was used instead. To measure the electrode impedance the meter was connected between a pair of the dry electrodes while a subject wore the device. Over five trials in which the subject simply pulled the helmet on and gave it a wiggle to seat the electrodes, measuring between the same pair of electrodes each time, the measured impedance ranged from 340 kΩ to greater than 1 MΩ at 10 Hz. By pushing down very hard on the electrode housings it was possible to reduce the impedance to 100 kΩ, though the force required to do so was painful where the electrode prongs pushed on the scalp. When the electrodes were released after having been pushed down hard, the impedance gradually increased to ∼500 kΩ over the course of about 30 s.

The impedance of the electrodes on the Elapse device is significantly higher than is typically required for capturing EEG, and is unpredictable. If a user takes the device off and puts it on again, even when trying to replicate their technique, the impedance can be markedly different each time. It is no surprise that we have had difficulty achieving low-impedance contact with dry electrodes—as described in Sections 2.1.1.1 and 3.2.3.2, the dry electrode design currently used on the device is an interim solution until an effective electrode is found. It is possible that one of the electrodes from Cognionics [Chi et al. 2013, 2012] or Quasar [Matthews et al. 2007], for example, would be suitable.

5.1.6 Sampling rate

The ADS1299 EEG AFEs in the Elapse device are configured for an output data rate of 250 Hz. The highest frequency of interest in the EEG is approximately 40 Hz, so this is well above the Nyquist rate. It is also the lowest data rate for which the ADS1299 can be configured. While it is possible to sample at a faster rate, doing so would require transmitting at least twice as much EEG data yet would provide no clear benefit.

5.2 CAMERA SUBSYSTEM CHARACTERISTICS

5.2.1 Frame rate and image size

The speed of the device's processor and the bandwidth of its Wi-Fi interface place an upper limit on the data rate of the video that the device captures. If the data rate is too high then the video encoder will not be able to encode each frame before the next one is captured and the Wi-Fi interface will not be able to transmit the encoded video as fast as it is produced. The data rate of the video stream is given by:

$$\frac{\text{bits}}{\text{second}} = \frac{\text{bits}}{\text{pixel}} \times \frac{\text{pixels}}{\text{frame}} \times \frac{\text{frames}}{\text{second}} \times \text{compression ratio} \tag{5.4}$$

This means that in order to achieve a given bit rate there is a trade-off to be made between the bit depth, the frame size, and the frame rate of the video.

The bit rate of the encoded video is some fraction of the bit rate of the raw video. From the perspective of the Elapse software, the video encoder is a black box with no way to choose the balance between the output data rate and the CPU/DSP utilization required to achieve it. This means that the compression ratio is essentially fixed and the only way to change the bit rate of the encoded video is to change the bit rate of the raw video according to (5.4).

The actual rate at which frames arrived at the laptop was measured while the camera was configured to output full-resolution 640×480 frames at 60 fps. The measured rate was less than 60 fps, indicating that the video encoder or the Wi-Fi interface or both were unable to handle the full video data rate and that the rate therefore had to be reduced. The requirements for the video subsystem given in Section 3.2.4.1 state that the minimum frame rate is 60 fps, so reducing the frame rate is not an option. This leaves the options of reducing the number of bits per pixel and the number of pixels per frame.

First, consider the number of bits per pixel. The camera outputs a raw Bayer pattern with 10 bit/px. The ISP captures this, reduces the bit depth to 8 bit/px, de-mosaics it to interpolate RGB colour for each pixel, and converts the colour space to YCbCr. The YCbCr colour space has a luma (brightness) channel and two chrominance channels. The luma channel contains the greyscale image and each chrominance channel contains the difference between a colour channel and the luma channel. Because the scene is lit with monochromatic illumination, the video contains little meaningful colour information. It would be preferable to make the most of this by configuring the ISP to output greyscale which would reduce the bit rate further, but the H.264 encoder requires YCbCr input. The unavoidable overhead of including the chrominance channels is minimized by the use of "4:2:0" chroma sub-sampling—the chrominance channels have half the horizontal and vertical resolution of the luma

channel.

The second part of reducing the bit rate is to reduce the number of pixels per frame. The ISP is configured to downsample the original image from 640×480 to 320×240 pixels. A reduction of $2\times$ in each dimension was chosen because it allows efficient resampling and still provides ample resolution for locating eye features. Measuring the actual received frame rate with this configuration demonstrated that the encoder is able to operate at the full 60 fps at this resolution and that the Wi-Fi interface is able to transmit the encoded video along with the data from the other sensors. As stated in Section 3.3.1.3, the total bit rate of the encoded video is approximately $960\,\mathrm{kbit\,s^{-1}}$.

5.2.2 Focus

As described in Section 3.2.4.1, the camera is mounted close to the eye—approximately in line with the end of the nose. This means that the camera must be able to focus at a distance of about 40 mm. Most small camera modules with fixed lenses such as the one used on the CAMEL are designed to focus from $\sim$300 mm to infinity. It is possible to adjust the focus of these units by adjusting how far the lens module is screwed into its threaded mount, though the amount of adjustment is limited.

Theoretical calculations

Some definitions of optical terms are required before calculating the theoretical focus limits of the camera. The near and far *depth-of-field* limits, D_N and D_F, are the lens-to-object distances between which all objects appear acceptably sharp. The *hyperfocal distance*, H, is the near depth-of-field limit when the lens is focused at infinity. Both of these quantities are defined in terms of the *circle of confusion* limit, c, which is the maximum diameter that the image of a point may have while still being considered "acceptably sharp". The *field of view* is the angular extent of the scene visible to the camera, governed by the size of the image sensor and the focal length of the lens.

The closest that the lens on the CAMEL module can be focused is at $d \approx 52$ mm, given the physical constraints of the lens mount. Using the lens parameters in Table 5.2 [OmniVision 2011b], we can calculate the near and far depth-of-field limits D_N and

Table 5.2 CAMEL lens specifications.

Parameter	Symbol	Value
Effective focal length	f	1.92 mm
f-number	N	2.4
Hyperfocal distance	H	300 mm
Diagonal field of view	θ	63.9°

D_F as

$$D_N = \frac{Hd}{H+d} = 44.3\,\text{mm} \tag{5.5a}$$

$$D_F = \frac{Hd}{H-d} = 62.9\,\text{mm} \tag{5.5b}$$

respectively [Merklinger 1992, p. 15]. That is, objects between 44.3 mm and 62.9 mm from the camera will be in focus.

To calculate what spatial resolution this corresponds to, consider the circle of confusion limit, c. We can calculate what value of c OmniVision used to design the camera module by the definition of the hyperfocal distance [Merklinger 1992, p. 14]

$$H = \frac{f^2}{Nc} + f \quad \Longrightarrow \quad c = \frac{f^2}{N(H-f)} = 5.15\,\mu\text{m} \tag{5.6}$$

This is slightly larger than the diagonal length of a pixel, which is reasonable. Using similar triangles, we can calculate the size C of the object at distance d which will produce an image of size c:

$$\frac{C}{d} = \frac{c}{f} \quad \Longrightarrow \quad C = \frac{cd}{f} = 0.14\,\text{mm} \tag{5.7}$$

This corresponds to a spatial resolution of 3.58 line pairs per millimetre (lp/mm). Combined with the depth-of-field limits calculated in (5.5), this means that the camera should be able to resolve features as small as 3.58 lp/mm and that objects at distances of 44.3–62.9 mm will appear sharp.

The near depth-of-field limit calculated here, $D_N = 44.3$ mm, is greater than the required object distance of 40 mm. To calculate whether this matters in practice, we need to consider the minimum size of the features of interest in the scene. Supposing that the camera can be considered to be in focus if it can resolve individual eyelashes, then we can repeat the calculations using the diameter of an eyelash as the circle of confusion limit. Elder [1997] reports that the mean width of an upper eyelash is 205 ± 28 µm. Substituting this value for C above, we find that the camera can resolve 205 µm features (2.44 lp/mm) between $D_N = 41.5$ mm and $D_F = 69.7$ mm. This is close to the 40 mm requirement.

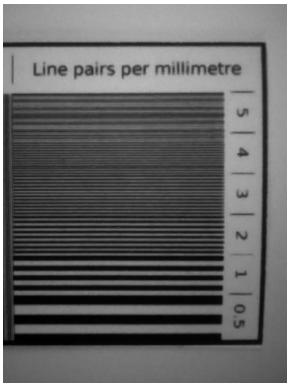
Spatial resolution measurements

To verify these calculations, a test pattern consisting of parallel horizontal black and white lines with a range of widths was used. Only horizontal lines were used for measurement; the lens is circularly symmetric and the pixels are square, so the results should generalize to vertical lines. The test pattern was positioned in front of the

camera at distances between 30 mm and 100 mm in increments of 10 mm. At each distance an image was captured with the pattern centred in the frame.

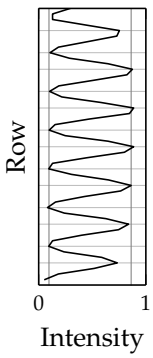
For each frame, a region of interest (ROI) was defined around each band of line pairs between 1 and 4 lp/mm. The median of each row of the ROI was computed, reducing it to a column vector. The contrast of the ROI was then defined as the difference between the median of the local maxima and the median of the local minima of the column vector. This process is summarized in Figure 5.7. (The 5 lp/mm band visible in Figure 5.7a was not analysed because it could not be printed with sufficient accuracy.)

Figure 5.8 plots the measured contrast of each ROI as a function of distance from the camera and line density. As expected, the contrast between black and white lines is highest near the focal distance of ∼52 mm and decreases as the distance changes. The contrast also decreases as line density increases and the lines blur together. If we define the threshold of what is considered "in focus" to be a contrast of half the full-scale range, we see that the 3 lp/mm band is in focus between 45–63 mm which is the same result as the theoretical calculations in (5.5). We also see that near the focal distance $d \approx 52$ mm the camera can resolve about 3.4 lp/mm which is within 5 % of the theoretical value from (5.7). Using the same focus threshold, we can see from Figure 5.8 that the camera is able to resolve about 2.4 lp/mm at a distance of 40 mm. This equates to a feature size of 210 µm, which is close to the eyelash diameter stated above. That is, the camera can almost resolve individual eyelashes at a distance of 40 mm, which comes very close to meeting the focus requirement. It may be possible to improve the focus slightly further in software using some kind of deconvolution process if necessary.



(a) Captured image of the test pattern.

(b) Extracted ROI around the 2 lp/mm band.



(c) Median of the ROI rows with extrema marked.

Figure 5.7 Example analysis of a test pattern ROI.

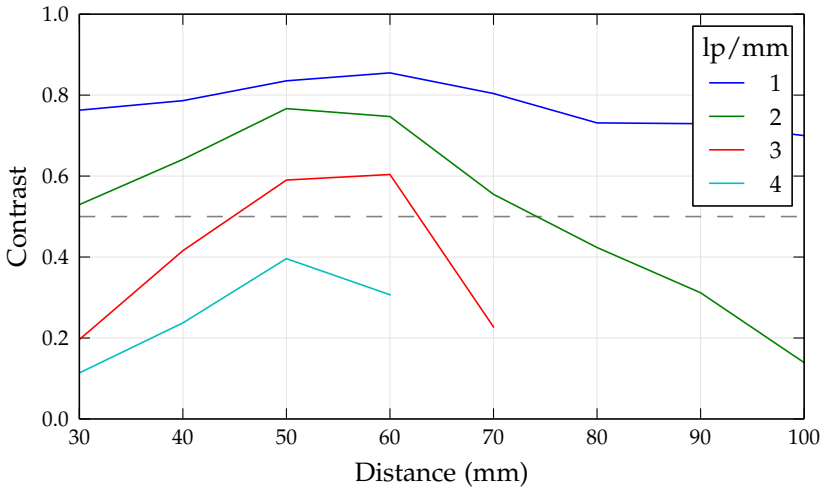


Figure 5.8 Measured spatial resolution of the CAMEL. Contrast is given as a proportion of the full-scale brightness range.

5.3 IMU SUBSYSTEM CHARACTERISTICS

The problems with the IMU device driver described in Section 3.3.1.2 made it difficult to test the performance of the IMU. As mentioned in that section, the driver only operates correctly with the gyroscopes disabled and the IMU sampling rate set to 10 Hz. Increasing the data rate beyond 10 Hz did not result in an increase in the actual output data rate from the driver.

The driver that STMicroelectronics provides for the LSM330DLC ignores the data-ready interrupts from the IMU and instead polls the device at its output data rate. That is, the driver reads a sample from the device, then sleeps for the sampling period before reading the next sample. This assumes that it takes no time to wake up the process, read a sample, pass the data to user-space, schedule the next read, and return the process to sleep. That assumption is, of course, false, meaning that the driver's sampling rate is lower than the hardware's sampling rate. This is likely to be part of the reason why the driver performs so poorly.

5.4 SYSTEM-LEVEL CHARACTERISTICS

The following experiments were done to evaluate the performance of the system as a whole.

5.4.1 Synchronization

The Elapse device has multiple sensors. To be able to implement a classifier which uses data from multiple sensors, the data must be synchronized. That is, two samples

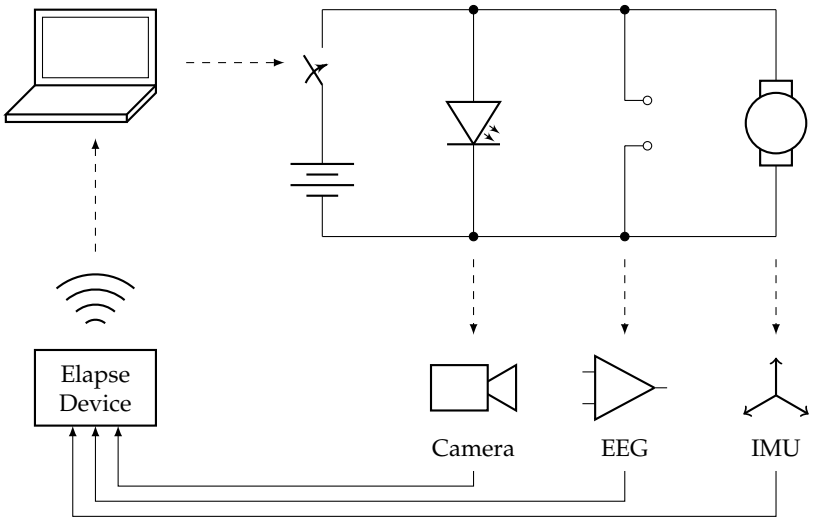


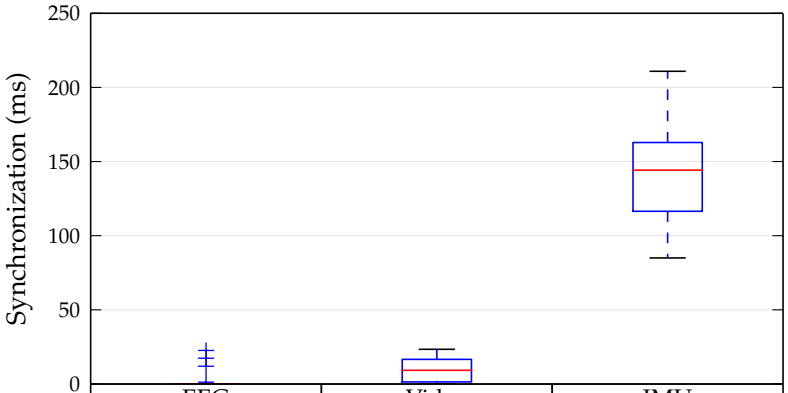
Figure 5.9 Synchronization and latency measurement apparatus. The laptop triggers an event that is visible to all of the sensors, and measures the time until that event appears in the captured data.

captured by different sensors at the same moment in time must have the same timestamp. The experimental apparatus depicted in Figure 5.9 was used to measure the synchronization of the sensors on the Elapse device.

The illustrated circuit generates an event which is visible to all of the device's sensors at the same time: *i)* an LED positioned in front of the camera lights up, *ii)* a voltage pulse is applied between two EEG electrodes, and *iii)* a servo rotates to tap the IMU. As the device reads data from each sensor, each sample is timestamped with the current value of the system clock. The signals are transmitted to a laptop as usual and stored to disk for later analysis. This system was used to generate and record 20 events.

For each recorded event, the signal from each sensor was thresholded to locate the sample in which the event occurred, and the timestamp of that sample was noted. We will call this the "event timestamp". The synchronization of each sensor was then defined as the difference between the event timestamp for that sensor and the earliest event timestamp from all three sensors. A box plot of the measured synchronization values is given in Figure 5.10. Box plots have boxes indicating the inter-quartile range, a line indicating the median, and whiskers extending to the most extreme data point within 1.5× the inter-quartile range.

The measured synchronization was expected to be within the sampling period of each sensor, i.e., 4 ms for EEG, 16.7 ms for video, and 100 ms for the IMU. The median synchronization for EEG was zero because it has the highest sampling rate and therefore is usually the first sensor to detect the event. The median values for EEG and video are both less than their respective sampling periods, as expected, though the maxima are greater. This is likely to be due to the non-deterministic interrupt

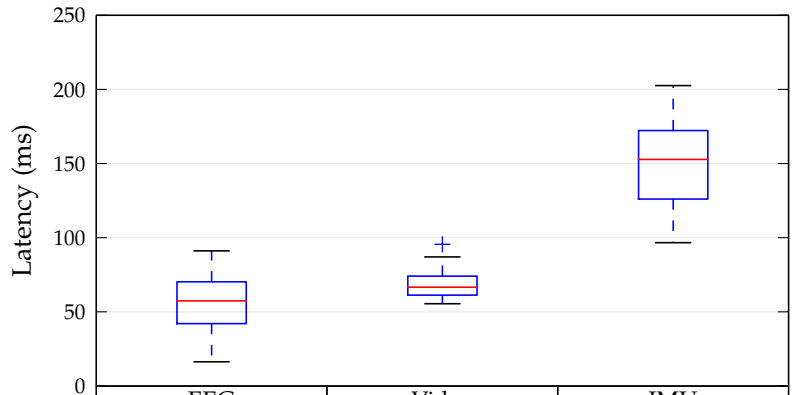


	EEG	Video	IMU
Minimum	0.0	0.0	85.0
Median	0.0	9.2	144.2
Maximum	22.6	23.4	210.9

Figure 5.10 Measured synchronization.

latency of the non-real-time Linux kernel. The synchronization delay of the IMU is much longer than its sampling period. Much of this delay can probably be explained by limitations in the design of the event trigger—it takes time for the servo to rotate and physically move the device, meaning that the event actually occurs slightly later for the IMU. Adding to this delay is the fact that it can take up to 40 ms between commanding a movement and the servo beginning to move because of the servo controller's update rate. The span of the IMU synchronization values is reasonably close to its sampling period, so it is likely that the vertical offset is mainly due to this delay between triggering the event and the detectable motion occurring. To get more accurate synchronization values for the IMU it would be necessary to use a mechanism with a shorter response time—perhaps a solenoid. To get a smaller range of IMU synchronization values it would be necessary to increase its sampling rate, though this is not possible with the current driver.

The measured synchronization of the EEG and video are acceptable—few samples are delayed for longer than the sampling period of each sensor. If we assume that the vertical offset of the measured IMU synchronization in Figure 5.10 is due to shortcomings of the experiment design (slow servo response), then the IMU also has acceptable synchronization for the same reason. It is important that the synchronization error is less than the sampling period for each sensor, otherwise samples from different sensors that were captured at the same time may not have the same timestamp, which could be problematic for a classifier.



	EEG	Video	IMU
Minimum	16.3	55.5	96.6
Median	57.4	66.6	152.8
Maximum	91.1	95.5	202.6

Figure 5.11 Measured latency.

5.4.2 Latency

The second important timing-related parameter for the Elapse system is latency, the time between when an event occurs and when it is processed. The system latency was measured at the same time as the synchronization using the same experimental apparatus (Figure 5.9). The latency for each sensor is defined as the time between the laptop triggering the event and the sample containing that event being emitted by the corresponding `SampleDecoder` in the Elapse client software. That is, it is the total round-trip time for the laptop to trigger the event, the sensor to sample the event, the server to capture and transmit the sample containing the event, and the client to receive and decode that sample. It does not include the time taken to process the sample through the `FeatureExtractors` and `Classifier`.

To measure latency, the laptop recorded the time according to its system clock immediately before triggering the event. Similarly, it recorded the time at which each `Sample` was emitted by the `SampleDecoders`. The captured samples were thresholded to locate the event in the same way as for the synchronization measurements in the previous section. The latency is the difference between these two times. A box plot of the measured latency values is given in Figure 5.11.

The range of EEG latencies is related to the block size of the ADS1299 driver (Section 3.3.1.2). Because the driver captures several samples before passing them to user-space, several sampling periods can pass between the sample containing the event being captured and it being transmitted to the laptop. As configured for this experiment, the EEG sampling rate was 250 Hz and the block size was 20, giving a block duration of 80 ms. The measured EEG latencies fall within this expected value, spanning a range of 75 ms. The minimum EEG latency of 16.3 ms indicates the

minimum round-trip time taken to capture, transmit, receive, and decode an EEG sample.

Compared to EEG, the video latency has a higher minimum latency and smaller range, but similar worst case performance. The increased minimum latency is due to the time it takes to encode and decode each frame of video before and after transmission. The video latency spans a smaller range than that for EEG because each frame is transmitted individually as soon as it is encoded, rather than being aggregated into blocks. What variation there is may be explained by the difference in the time that the H.264 encoder requires to encode intra-frames (whole images) versus inter-frames (difference images).

The measured latency of the IMU is similar to its synchronization. That is, it is poor and can probably be explained by a combination of limitations in the experiment design and the buggy driver implementation that can't exceed 10 Hz.

The classifier requires data from all three of the sensors, so the latency of the whole system is limited by the worst latency of the three. Assuming that the latency of the IMU could be improved with a replacement device driver, the total worst-case latency of the system is approximately 100 ms. In practice, when simultaneously watching a user wearing the device and the real-time data display on the laptop, the offset between the two is barely perceptible to the human eye. A lapse detection device needs to detect and respond to events lasting > 500 ms, so a latency of 100 ms allows detection soon after lapse onset.

5.4.3 Power

To measure the total power usage of the device, it was connected to a bench-top power supply through an ammeter. The supply voltage was set to 3.6 V—the nominal battery voltage—and the supply current was measured during three stages of operation. During the boot process, the peak power draw was 2.53 W. When sitting idle with the server running but no data capture in process, the average power was 2.35 W. When fully operational, with data being captured and transmitted to the remote computer, the power draw was 2.92 W. Given these measurements, and simplistically assuming a flat discharge curve, the 6 Ah battery should be able to power the device for approximately $6\,\text{Ah} \times 3.6\,\text{V}/2.92\,\text{W} = 7.4\,\text{h}$ while continuously capturing data. This falls slightly short of the target run time of 8 h stated in Section 3.2.5.1.

The power drawn by the laptop was not measured because it was assumed that an external power source would always be available. The situations in which a lapse and drowsiness detector would be useful all involve the subject sitting in a fixed position. Whether this is in the cab of a truck or an industrial control room, there is generally a power source available within the range of the device's wireless interface.

5.4.4 Weight

The total weight of the Elapse device is 1020 g, while the unmodified helmet upon which it is built weighs 600 g. The battery contributes 110 g of this and the enclosures and mounting hardware contribute the majority of the remainder. Weight was not considered during the design stage, though it may have some effect on how comfortable the device is to wear for long periods of time.

5.4.5 Data rate

The Elapse device transmits a total of $\sim$3.2 MB min^{-1} (195 MB h^{-1}). This was measured by capturing 5 min of data, saving it to disk using the default `SimpleRawDataSink`, and examining the size of the resulting file. At this rate, hundreds of hours of data could be recorded on a typical hard drive. Note that this measure is only important in a research setting when data is saved for later analysis—a commercial lapse detector would not save the raw data to disk.

5.4.6 Boot time

The time it takes to boot the Elapse device is defined as the time from when the power switch is turned on to when the server is ready to accept a connection. This was measured using a stopwatch, starting when the switch was closed and stopping when the "ready" LED was lit. The average of three such trials was 25 s, with all three falling within a 1 s range. It is likely that this time could be reduced by further optimizing the boot process—statically linking all kernel modules, removing kernel features that are only useful during development, disabling the bootloader timeout, and so on. These optimizations have not yet been made because the time it takes to boot the device is negligible compared to the length of time the device is typically operated for.

5.5 SUMMARY

Most of the subsystems of the Elapse device operate as expected and meet the requirements laid out in Chapter 3. The EEG front-end has less than 1 $\mu V_{p\text{-}p}$ of noise and a 3 dB bandwidth of 47 Hz. The CMRR was not as good as expected, though was still 82 dB at worst. The impedance of the dry electrodes was several hundred kΩ—too high to be useful for a passive electrode—so a different electrode design will need to be sourced.

The camera is able to resolve features as small as 2.4 lp/mm at a distance of 40 mm. When down-sampled to 320 $\times$ 240 px and encoded, 60 fps of video can be transmitted to the remote computer.

The overall system has a latency of less than 100 ms (disregarding the defective IMU). Its battery should be able to provide approximately 7.4 h of operation.

Chapter 6

EXPERIMENTAL EVALUATION

The previous chapter examined the performance of the Elapse device hardware in isolation. This chapter describes experiments that quantify the performance of the device when in use on a person. The first two of these evaluate the eye video processing software. The third is a brief integration test to ensure that the system as a whole can capture data from which the user's cognitive state could be inferred.

6.1 EYE VIDEO ANALYSIS

The following sections describe two small-scale experiments undertaken as a proof of concept to determine whether the eye-video processing algorithm developed in Section 4.1 can identify eye features as accurately as a human can.

6.1.1 Measuring PERCLOS

As discussed in Sections 2.1.2 and 4.1, a commonly-used indicator of drowsiness is PERCLOS, the percentage of time that the eyes are $> 80\,\%$ closed in a one minute window. PERCLOS is easy (albeit time consuming) to measure manually and simple to generate from pupil location data. For these reasons, the first experiment aimed to compare PERCLOS values generated from the output of the pupil localization algorithm to manually rated values.

6.1.1.1 Experiment design

Two sessions lasting 15 min each were recorded with a single subject. The tests were carried out at 1:00 pm and midnight after 6 and 17 hours of wakefulness respectively. The subject was seated in front of a computer screen and instructed to visually track a dot moving around the screen for the duration of the session, similar to the 2D tracking task of Poudel et al. [2008].

The pupil localization algorithm was run on the resulting video to extract the location and size of the pupil and the eye closure state (Section 4.1.1). The algorithm

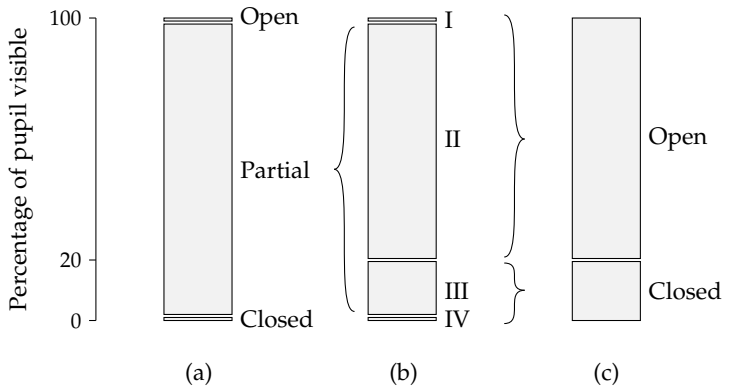


Figure 6.1 Percentage eye closure categories: (a) shows the categories assigned by the pupil localization algorithm, (c) the categories used for PERCLOS calculations, and (b) the categories used to map between the two.

uses a different definition of "open" and "closed" to that used for PERCLOS, so further sub-categories were introduced in order to map between the two systems. These categories are illustrated in Figure 6.1b. Categories I and IV correspond directly to the "open" and "closed" states from the pupil localization algorithm. Frames with "partial" pupils were assigned to category II or III according to whether the height of the visible pupil region was greater than 20 % of the height of the pupil region in the most recent frame in which the pupil was not partly covered. For PERCLOS calculations, categories I and II are considered "open" and III and IV "closed":

$$\text{PERCLOS} = \frac{\text{III} + \text{IV}}{\text{I} + \text{II} + \text{III} + \text{IV}} \times 100\,\% \tag{6.1}$$

PERCLOS is measured over a one-minute window, so the assigned category for each frame was pushed into a buffer of 60 s × 60 fps = 3600 elements. The PERCLOS value is then simply the percentage of values in the buffer belonging to categories III or IV.

Four one-minute windows evenly spaced throughout each session were manually rated as a gold standard against which the output of the software could be compared. Unfortunately it was not feasible to manually rate the whole data set because of the time required to do so. Using four windows was a more achievable target, though still required manually classifying 2 sessions × 4 windows/session × 60 s/window × 60 fps = 28800 frames.

6.1.1.2 Results

The categories assigned by the pupil localization software were compared to those assigned manually in two ways. Firstly, for each frame that was manually classified, the automatically-assigned category was compared to the manually-assigned gold

standard. Secondly, the PERCLOS values for the four manually-rated one-minute windows were compared to the automatically-rated values.

Frame-by-frame comparison

Each element of the confusion matrix in Table 6.1 contains the number of frames manually categorized as being in that row and automatically categorized as being in that column. That is, if the algorithm produced the same results as a human, all off-diagonal entries would be zero. The table includes the categories of all frames from the eight one-minute periods.

Several of the frames manually classified as category I (open) were automatically classified as category II (partly covered) and vice versa. Many of these errors are due to the ambiguity around which dark pixels belong to the pupil and which to the dark line between the eyelid and the cornea. Note, though, that these errors have no effect on the PERCLOS value since categories I and II are both "open".

Many of the errors for frames manually classified as category III (mostly covered) are due to having to estimate whether the visible pupil region is less than 20 % of the full pupil height when manually assigning a category. The frames manually classified as category I and automatically classified as category IV are those in which the algorithm failed to detect a pupil that was actually visible.

The process of assigning these categories to video frames can be considered a binary classification problem where the two classes are the PERCLOS open/closed states. Using this idea, and considering "closed" to be a positive result, the pupil localization software has an accuracy = 0.998, precision = 0.815, and recall = 0.918. The accuracy is very high because the large imbalance in class sizes obscures the errors. The precision is lower, pulled down by the frames where the algorithm failed to locate the pupil despite it being visible (manual I/II, auto IV). The reasonably high recall indicates that most frames in which the pupil was mostly covered were identified as such.

Table 6.1 Comparison of manual vs. automatic pupil classification.

		Automatic			
		I	II	III	IV
Manual	I	27 673	375	0	42
	II	190	285	0	6
	III	0	17	2	15
	IV	0	2	0	195

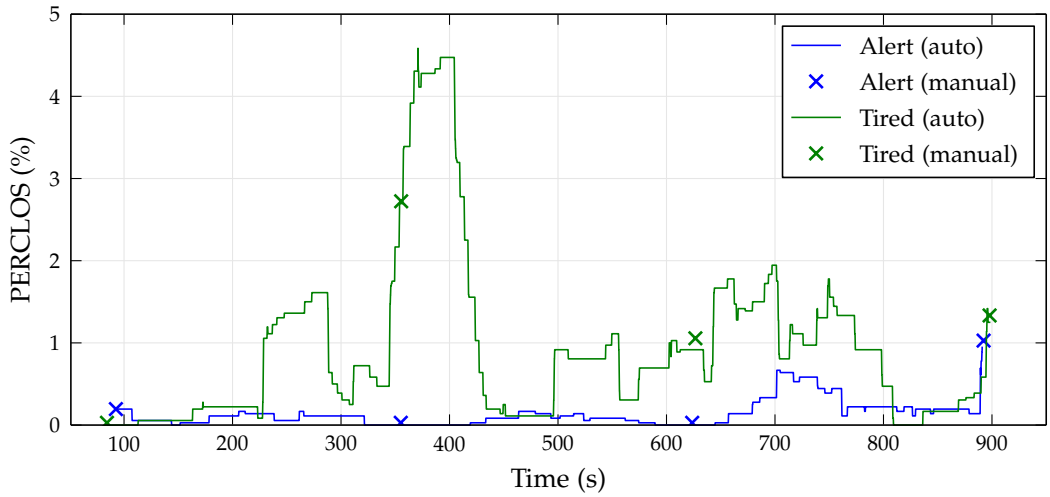


Figure 6.2 Measured PERCLOS.

PERCLOS comparison

Figure 6.2 plots the automatically measured PERCLOS values over time for the two sessions. As expected, PERCLOS was both substantially higher and more variable in the second session when the subject was drowsy. Also visible in the second session is a large spike in PERCLOS centred at about 380 s. This was caused by a microsleep. A period of eye closure lasting 2.5 s caused a jump of 4 % in PERCLOS, and 60 s later, when the 1-min window moved past that period of eye closure, the PERCLOS dropped back down to its previous level.

The values for the four manually classified windows for each session are marked with crosses on Figure 6.2. Note that they appear as points because only at the end of the window is there enough data to generate a PERCLOS value. If the windows for manual classification had been longer than 60 s then the results would appear as lines rather than points. As the figure shows, there is a reasonable level of agreement between the manually and automatically measured values.

6.1.2 Robustness to environment

The second experiment was designed to test the robustness of the pupil localization algorithm to several external factors that could be expected in typical real-world usage.

6.1.2.1 Experiment design

A total of eight input conditions were tested, representing every combination of three binary variables:

- eye colour — one subject with light irises, one with dark,
- ambient lighting — office lighting and a darkened room,
- glasses — with and without prescription eye glasses.

From previous experience, these three variables were expected to have the most effect on the system's performance.

Subjects completed a task for each set of conditions to emulate a typical range of eye movements—saccades, tracking, and eye closure. They were seated in front of a computer screen and instructed to watch a dot as it moved around the screen. Initially the dot jumped to each corner of the screen at 1 s intervals. This was followed by a random 2D tracking task similar to that of Poudel et al. [2008] for 3.5 s. At that point a beep sounded, instructing the subject to close their eyes. After 1.5 s another beep sounded, the subject opened their eyes and resumed tracking for the remaining 3 s. This task induces a wider range of eye movements than the purely smooth tracking used in the experiment of Section 6.1.1.

Every frame of video from the eight 12 s sessions (5760 frames) was manually annotated with the position and diameter of the pupil. Each frame was also assigned one of four categories: "invalid", if the data was visibly corrupted, "open" if the pupil was completely visible, "partly closed" if the pupil was partly covered by an eyelid, or "closed" if the pupil was not visible at all. Because of the way the data capture software was implemented for this experiment, recording was not synchronized to the video encoder and could therefore start at any point during an encoded group of pictures. This caused several frames at the start of each session to contain meaningless data until the first intra-frame was received, and it is these meaningless frames that were classified as "invalid". The remaining three categories correspond directly to those used by the pupil localization software. This experiment did not involve calculating PERCLOS, so there was no need to introduce the 20 % eye closure threshold as for the previous experiment.

6.1.2.2 Results

As before, box plots in this section have boxes indicating the inter-quartile range, a line indicating the median, and whiskers extending to the most extreme data point within 1.5× the inter-quartile range. For sizes given in pixels, 1 px $\approx$ 0.15 mm. Subject 1 had light-coloured irises, subject 2 dark.

Pupil position

The error in the pupil position is defined as the Euclidean distance between the manually annotated centre of the pupil and the centre of the pupil detected by the

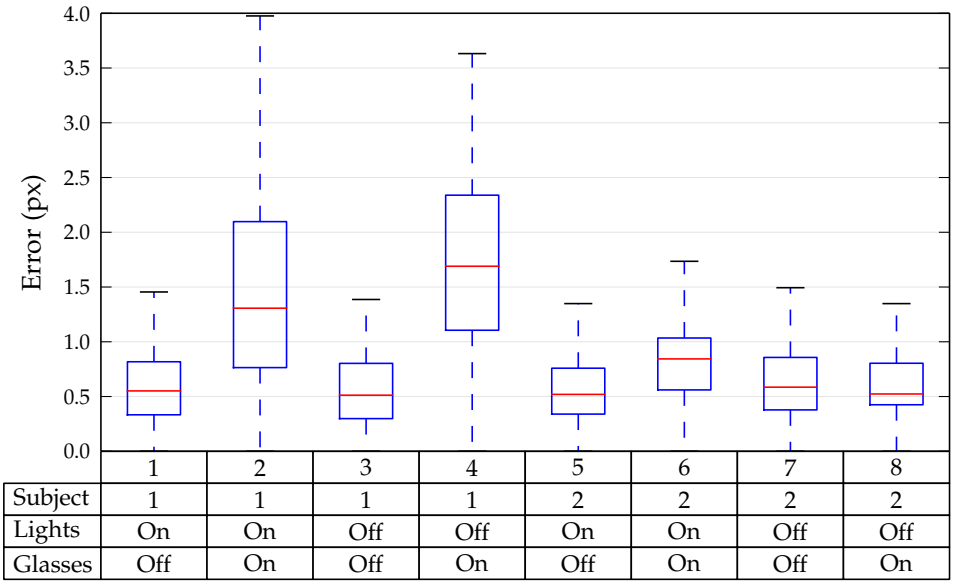


Figure 6.3 Measured pupil position error.

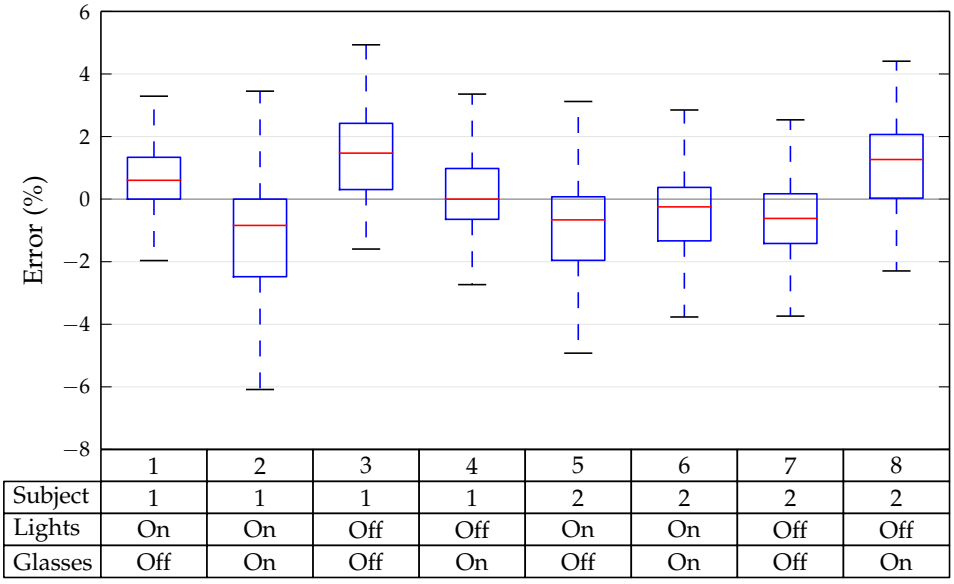


Figure 6.4 Measured pupil diameter error.

software. For six of the eight sets of input conditions the median pupil position error was less than 0.7 pixels (Figure 6.3). The remaining two were both for subject 1 with glasses. This could be because subject 1's glasses affect the accuracy of the pupil detection algorithm or because the camera was poorly positioned for those measurements—further investigation would be needed to determine the cause.

Note that measurements of gaze derived from the pupil location are unlikely to be useful when detecting microsleeps but may be useful for detecting diverted attention.

Pupil diameter

Figure 6.4 shows the differences between the pupil diameter measured by the software and that measured manually for each set of input conditions. Since the projection of the pupil is elliptical, the diameter is equivalent to the length of the major axis of the ellipse. For all of the eight combinations of input conditions, the median absolute error is less than 1.8 %.

Marshall [2007] describes an index of cognitive activity that uses wavelet analysis to identify small, rapid changes in pupil diameter that occur during periods of high cognitive workload. Without further testing, it is unclear whether such a technique could be used on the output of our pupil localization algorithm or whether the errors shown in Figure 6.4 would obscure the important high-frequency features.

Eye closure

For lapse detection, the accuracy with which the software detects eye closure is the most important performance metric for the eye video subsystem. The confusion matrix in Table 6.2 compares the category that each frame was assigned manually (Section 6.1.2.1) against the category assigned by the image processing software (Section 4.1). Of the 5706 valid frames, the software categorized 38 incorrectly (0.7 %). Most of these errors, though, occurred during eye closure, which is when accurate classification is most important. 33 of these 38 errors occurred when the software classified a "partly closed" frame as either "open" or "closed". That is, during a blink the software tends to keep classifying frames as "open" for longer, and then jumps straight to "closed". In practical terms, if the length of time that the eyes are closed

Table 6.2 Comparison of manual classification to software output.

		Automatic		
		Open	Partial	Closed
Manual	Open	5016	4	0
	Partial	26	26	7
	Closed	1	0	626

is being used as a drowsiness indicator, categorizing a "partly closed" eye as "open" has little effect because the number of frames categorized as "closed" remains the same. As can be observed in Figure 6.4, the errors were evenly distributed across the combinations of input conditions—no condition or combination of conditions was found to increase the error rate.

6.2 INTEGRATION TEST

This final experiment aimed to demonstrate that all of the parts of the Elapse system work together as a whole and that the system is capable of detecting events in the captured signals.

6.2.1 Experiment design

The experiment was carried out with a single subject. The subject completed a two-part task while wearing the device. In the first part of the task, the subject looked at a fixed point in front of him while keeping his eyes open for 15 s, followed by closing his eyes for 15 s. This sequence was repeated three times. In the second part of the task, the subject performed an exaggerated simulation of a microsleep: slowly closing his eyes and allowing his head to droop forward before quickly opening his eyes and jerking his head back to a neutral position. This was performed three times, separated by 15 s intervals in which he maintained gaze on a fixed point. The simulated microsleep task was designed to induce measurable changes in all of the signals that the Elapse device measures: eye closure in the eye video, a spike in the angle of the head from the IMU, and changes in the power of the EEG frequency bands.

We initially tried to use the dry electrodes on the Elapse device to capture EEG while conducting this experiment. As described in Section 5.1.5, however, it is very difficult to achieve reliable, low-impedance contact with the scalp using these electrodes. In an attempt to improve the chances of achieving adequate contact, a bald subject was recruited for the experiment. Even though the subject had no hair under the device's electrodes, we were unable to achieve a sufficiently low contact impedance and so were forced to conduct the experiment using wet electrodes instead. Two of the dry electrodes on the device—the reference electrode and one measurement electrode—were disconnected and replaced with gold disc electrodes. Because only two disc electrodes were connected to the Elapse device, the experiment described in the previous section was carried out twice, once with the measurement electrode at O1 and once at Fp1. The reference electrode was placed at Cz in both cases. In all of these positions, the scalp was prepared using an abrasive gel (Nuprep, Weaver & Co.[1]), the electrode was filled with electrolyte paste (Ten20 conductive paste, Weaver & Co.[1])

[1] `http://www.doweaver.com`

and taped in place, and then the helmet was put on over the top. The impedance of the gold disc electrodes with electrolyte paste was measured to be 7.7 kΩ which is reasonably close to the usual target of 5 kΩ used in clinical practice.

The feature extractors described in Chapter 4 were used to process the captured data. The window size was set to 2 s with a step size of 0.5 s. The pupil localization algorithm from Section 4.1 was used to calculate a variant of PERCLOS in each 2 s window of data (not the usual one minute window) which we will denote as PERCLOS_2. The EEG power-band feature extractor from Section 4.2.1 was used to calculate the total power between 4 Hz and 40 Hz. The IMU feature extractor from Section 4.2.2 was used to calculate the cumulative change in the pitch (nod) angle of the head in each window.

A basic classifier was implemented which simply thresholds these three features and considers an event to have occurred if all three features exceed their respective thresholds. That is, it classifies a window of data as containing a simulated microsleep if there is sufficient eye closure and increase in total EEG power and forward head tilt. This is a very simplistic design and does not reflect the subtleties of detecting real microsleeps, yet it is sufficient to demonstrate the concept of classifying some state based on a set of features from multiple modalities.

6.2.2 Results

The first part of the experiment, in which the subject alternately opened and closed his eyes, was designed to test that the Elapse device can capture clean EEG. In the majority of people (∼80 %), an increase in posterior alpha activity occurs during restful wakefulness with the eyes closed compared to the eyes-open condition [Santamaria and Chiappa 1987]. Additionally, some EOG is usually visible in anterior channels during eye movement. Capturing data at O1 and Fp1 while alternately opening and closing the eyes was intended to show the presence of these features and so demonstrate that the captured data is actually EEG.

Figure 6.5 shows a spectrogram of the EEG captured at O1 during 2.5 periods of alternating eyes open and closed. The increase in alpha-band power (8–12 Hz) during eye closure is clearly visible. Figure 6.6 shows the same data in the time domain, bandpass filtered between 4 Hz and 40 Hz, as two 10 s windows centred at the onset of each of the periods of eye closure in Figure 6.5. Again, the increase in alpha activity is clearly visible when the eyes are closed. In Figure 6.7, 25 s of data captured at Fp1 is displayed, bandpass filtered between 1 Hz and 40 Hz. Eye movement artefacts are visible as the eyes open and close, as are smaller transients during blinks.

The second part of the experiment, with the simulated microsleeps, was designed to test the operation of the system as a whole. As described in the previous section, features were extracted from each captured signal and a classifier used these features to

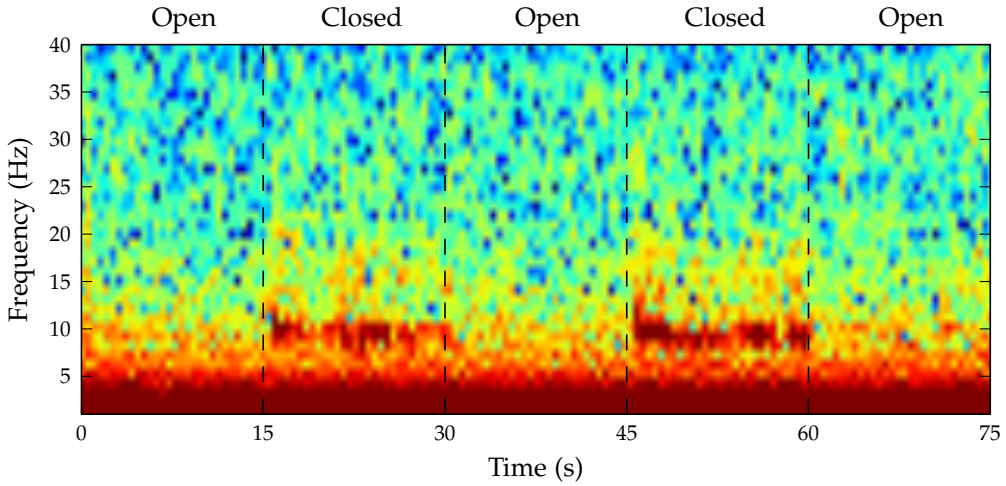


Figure 6.5 Spectrogram of occipital EEG (O1), showing increased alpha power during eye closure.

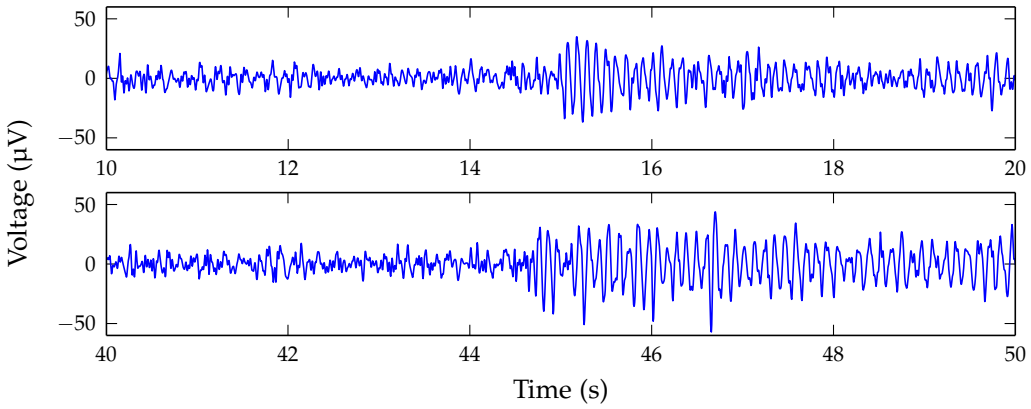


Figure 6.6 Increase in occipital alpha during eye closure. The two traces show 10 s windows around the beginning of the periods of eye closure shown in Figure 6.5.

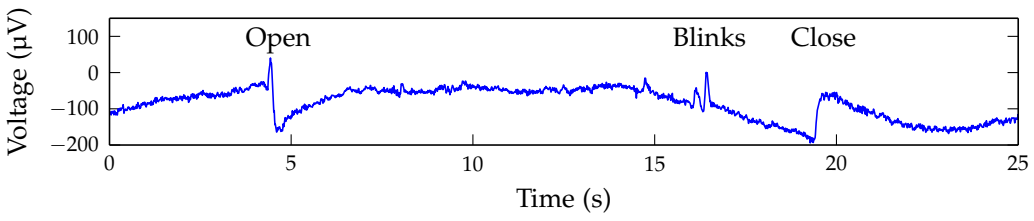


Figure 6.7 Eye blink artefacts in a frontal EEG channel (Fp1).

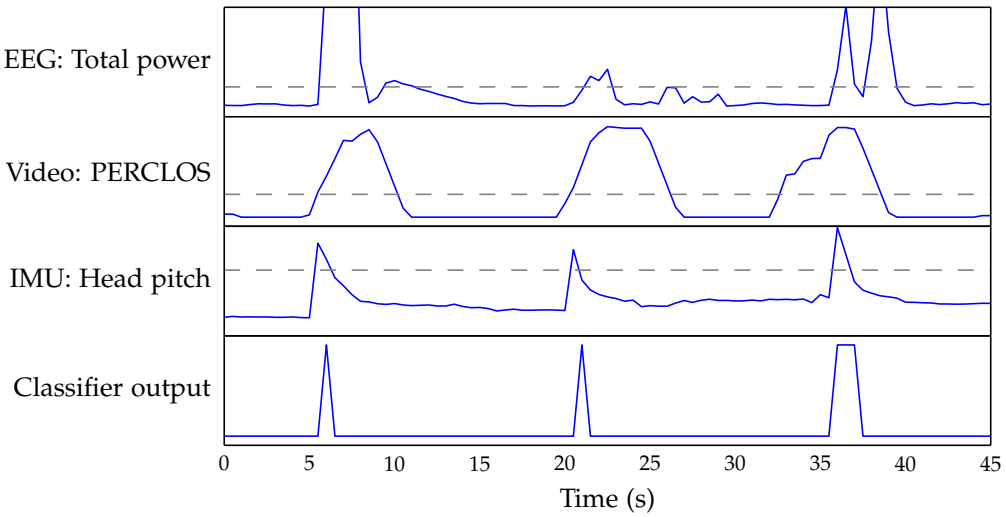


Figure 6.8 Detecting simulated microsleeps. The upper three plots show the features extracted from each signal. The classifier thresholds each feature and considers an event to have occurred when all features exceed their respective thresholds.

identify the simulated microsleeps. In Figure 6.8 the three features—PERCLOS$_2$, EEG power, and head pitch—are plotted over the course of three simulated microsleeps. The threshold that the classifier used for each signal is overlaid on each with a dashed line. The output of the classifier is illustrated in the bottom trace, measuring '1' when it identifies a simulated microsleep, defined as all three of the features exceeding their thresholds. While this is not a compelling demonstration of biosignal classification, it does demonstrate that the system as a whole operates as intended. The hardware captures signals from the user's body, the captured data is transmitted to a remote computer, salient features are extracted from each signal, and a multi-modal classifier combines the features to produce an output that indicates the state of the user.

Careful inspection of Figure 6.8 shows that the head pitch measured by the IMU peaks at the onset of the lapse, rather than at the end as would be expected from the experiment description. As a result of either a poor description of the task or a poor understanding of what was required, the subject actually nodded his head sharply while closing his eyes and then straightened up slowly while opening his eyes. Any future experiments involving simulated microsleeps should ensure that subjects understand the required sequence of events in order to avoid this happening.

6.3 SUMMARY

The output of the pupil localization algorithm described in Section 4.1 can be used to classify the PERCLOS open/closed states with precision $= 0.815$, and recall $= 0.918$. For six of the eight tested combinations of input conditions—ambient lighting, eye colour, and presence of glasses—the median pupil position error was less than 0.7 px

($\sim$0.1 mm); further investigation is needed to determine the cause of the slightly higher error in the other two cases. The median absolute error in the pupil diameter was less than 1.8 % for all combinations of input conditions.

The first part of the integration test demonstrated that the dry electrodes on the Elapse device can not achieve the reliable low-impedance contact necessary for capturing EEG, as outlined in Section 5.1.5. When conventional wet electrodes are used, though, the captured EEG clearly shows frontal eye blink artefacts and increased posterior alpha during eye closure, as expected. This shows that the EEG acquisition electronics operate as expected but that the dry electrodes do not.

The second part of the integration test showed that the system as a whole can capture multiple biosignals, transmit them to a remote computer, extract features from the data, and combine multiple features to classify the cognitive state of the user. Creating a system with this functionality was the primary objective of this project. With the foundation now in place, it remains for others to implement the more complex feature extraction and classification algorithms necessary to detect lapses of responsiveness.

Chapter 7

DISCUSSION & FINAL REMARKS

7.1 DISCUSSION

This thesis has presented the development of a multi-modal device for application in lapse detection. The wearable device captures EEG, eye video, and head movement and transmits the data to a remote computer in real time. It has also presented the development of associated software that incorporates an extensible framework for real-time, multi-modal signal processing and classification. The result, the Elapse platform, is the only system existing at this time with this combination of features.

7.1.1 Hardware

To the best of our knowledge, the combination of sensors present on the Elapse device is unique. SensoMotoric Instruments' pairing of their eye-tracking glasses with the Emotiv EPOC EEG headset (Section 2.1.2.2) is the most similar system that is currently available (although it wasn't available when this project began). That system is only suitable for use in a research setting; it requires the user to don both the glasses and the EEG headset as well as the separate wireless interface module for the glasses. By contrast, the Elapse system requires the user to wear a single device which they can put on without assistance.

7.1.1.1 EEG

Physically, the dry electrode design used on the Elapse device resembles aspects of g.tec's g.SAHARA electrode, Mindo's pin-style electrode, and Quasar's dry electrode (Section 2.1.1.1). It has an array of rigid pins, like the g.SAHARA, mounted on a sprung base, somewhat like Quasar's. It is a passive design, like Mindo's, lacking the active buffer amplifier that the other two have. Unlike all of the other three, the pins on the Elapse electrodes can retract fully into their housing, making it a safer design.

Unfortunately, a clean EEG recording was not able to be obtained using these electrodes. From the research reviewed in Section 2.1.1.1, the most successful dry

contact electrodes have amplifiers mounted directly on the electrodes with careful shielding. Without this buffering and shielding, the high contact impedance makes the electrode too susceptible to movement artefacts and electromagnetic interference to be useful. There is some evidence that if a delay of 10 min or more had been introduced before taking measurements then the impedance would have been lower [Liao et al. 2011, Searle and Kirkup 2000]. Nevertheless, it will be necessary to find replacement dry electrodes in order for the Elapse device to be useful.

On the other hand, the EEG acquisition electronics were demonstrated to function correctly. The measured CMRR and cross-talk values were poorer than expected, but the experiment in Section 6.2 showed that the device can capture clean EEG containing typical spectral features (when using appropriate electrodes).

The EEG subsystem of the Elapse device offers similar features to the commercial EEG headsets listed in Section 2.1.1.2. It has 16 channels (8 currently in use) and captures 24-bit data at 250 Hz. Data is transmitted wirelessly to a remote computer in real time.

7.1.1.2 Eye video

The Elapse device has a camera mounted below one eye, capturing 320×240 px greyscale video of the eye at 60 fps and streaming it to the remote computer in real time. This is a subset of the functionality offered by the commercial head-mounted eye trackers listed in Section 2.1.2.2. The SensoMotoric Instruments and Tobii glasses both do binocular gaze tracking and include an outward-facing scene camera. The Elapse camera operates at the same frame rate as the SMI glasses and twice the rate of the Tobii glasses. The Elapse device has the video capture and wireless interface hardware integrated into an on-head unit rather than using a cable to connect to a unit on the user's belt like the two commercial devices.

Compared to the head-mounted eye tracker that Babcock and Pelz [2004] developed, the Elapse device is very compact. Where their device had a backpack full of electronics to save video for later offline analysis, the Elapse device has a small module on the side of the head that transmits the video for real-time analysis.

Although intended for a similar purpose to the Elapse device, the Optalert Eagle uses a different paradigm to the Elapse eye-video camera. Optalert's infrared reflectance oculometry measures eye closure with high temporal resolution; the eye-video camera measures more information—pupil location and size as well as eye closure—but at a lower temporal resolution.

7.1.1.3 IMU

The problems with the device driver for the IMU outlined in Section 3.3.1.2 have prevented extensive testing of the IMU hardware. However, the IMU does currently provide an estimate of the head orientation with two degrees of freedom at 10 Hz. It will be necessary to fix the driver or write a replacement in order to make better use of the IMU in the future.

7.1.1.4 Wearable device

The current prototype of the Elapse device is built around a safety helmet. This provided a convenient rigid platform for developing the first generation of the hardware. If the device is eventually developed into a commercial lapse detector, it will not necessarily take this form. However, given some of the potential application areas for a lapse detector, building it into a safety helmet may be advantageous (Section 3.2.5.2). As it stands, the device, the wearable portion of the system, is a single unit that the user can don without external assistance.

The battery life of the Elapse device (7.4 h) is not as good as the best of the commercial wireless EEG headsets listed in Section 2.1.1.2 (6–45 h). The inclusion of the eye video camera increases the power requirements substantially above what is required for EEG alone. Conversely, the battery life of the Elapse device is longer than the commercial eye tracking glasses listed in Section 2.1.2.2 (2–3 h). To achieve this battery life, though, the device has a relatively large and heavy battery which may make it less comfortable to wear for extended periods.

The current dry electrodes are moderately uncomfortable where the pins push on the scalp. In a sense, this issue is moot since the electrodes need to be replaced anyway, but it is a factor to consider when selecting a replacement. Electrodes made from bristles [Grozea et al. 2011] or a flexible polymer [Chi et al. 2013] or that don't contact the scalp at all [Chi et al. 2012] show promise on this front.

7.1.2 Software

The Elapse client software takes care of all the non-application-specific details related to using the device, while allowing users to supply their own application-specific signal processing code. The client software handles all of the communication with the device: it can read hardware configuration from a file and configure the device's sensors accordingly; it can start and stop data capture; it monitors the device's battery voltage. The signal processing pipeline was designed to encapsulate a range of algorithms—almost anything that can be represented as feature extraction followed by classification. The plug-in system allows users to write signal processing code to fill the slots in the pipeline and to choose which plug-ins to load at runtime. Plug-ins

can be written in C++ or Python and future extension to other languages is possible. The data flowing between elements at any point in the pipeline can be saved to disk for later analysis and this data can also be re-loaded back into the pipeline to "replay" it. The client software also provides a graphical user interface to load plug-ins and monitor the system.

7.1.2.1 Signal processing

The pupil localization algorithm presented in Section 4.1 is computationally simpler than many alternatives [Daugman 2004, Li et al. 2005, Świrski et al. 2012] while still providing reasonable performance. Initial experiments indicate that it is robust to eye colour, ambient lighting, and the presence of glasses, and that it can be used to measure PERCLOS accurately. However, the system needs to be tested on a wider range of people in a wider range of environments to confirm these results. In particular, it has not yet been tested in sunlight or rapidly changing lighting conditions such as may be encountered when driving. The pupil localization algorithm currently provides no estimation of gaze direction which could be useful for detecting diverted attention lapses.

Since signal processing was not the focus of this project, only simple proof-of-concept feature extractors have been implemented for the EEG and IMU. Initial experiments indicate that it is possible to extract meaningful features from both of these signals (given suitable EEG electrodes) which is promising for future development.

7.1.3 Usefulness

The Elapse platform will enable NeuroTech to further progress their research into detecting and predicting lapses of responsiveness. It will enable others to focus on developing signal processing algorithms by providing a complete data acquisition and analysis platform to test their designs.

The Elapse device captures EEG so that lapses can be detected earlier than is possible with a dash-mounted camera system. It combines the EEG with eye video so that lapses can be detected with potentially higher accuracy than by either one alone. It uses a fast update rate in order to detect short lapses rather than just long-term drowsiness. Hopefully it is the first step towards a better real-world lapse detector than what is currently available.

7.2 REVIEW OF OBJECTIVES

Design and build biosignal acquisition hardware to monitor multiple signals relevant to lapse detection.

The Elapse device captures EEG, video of the eye, and head movement, and streams the data to a remote computer wirelessly in real time. These three signals have been shown to be useful for lapse detection in the prior research reviewed in Chapter 2. To the best of our knowledge, no other single device exists that captures this combination of biosignals.

Implement a software structure to enable experimenting with combinations of signal processing and classification algorithms.

The Elapse client software provides a framework for implementing a wide range of signal processing algorithms. It is structured in a way which is suitable for encapsulating a wide variety of algorithms: a feature extraction stage for each signal type followed by a classifier to combine the features and produce a meaningful output. The software also handles all of the communication with the device, the receiving, saving, and loading of data, and provides a graphical user interface.

Start implementing some of these algorithms as proof of concept.

A pupil localization algorithm has been developed which identifies the location and size of the pupil and whether it is partially or completely covered by the eyelids. Simple feature extractors have been written for the EEG and IMU which extract the power in the standard frequency bands and the head orientation respectively. A simple classifier has also been written which identifies simulated microsleeps by thresholding these features.

Ensure the system is able to operate in real time in environments typical of real-world usage.

The Elapse system has been tested on people with different eye colours, with and without glasses, in light and dark ambient conditions. It successfully detected simulated microsleeps, and the latency of the system was measured to be less than 100 ms. These positive results were only able to be obtained when using wet EEG electrodes; the current dry electrodes are inadequate and will need to be replaced for the device to be usable in the real world.

Design the system to be generalizable to cognitive monitoring applications other than lapse detection.

The signals that the device acquires can be used for applications other than lapse detection, such as augmented cognition, workload monitoring, and passive BCIs. The software framework makes it equally possible to implement algorithms for these applications as for lapse detection.

7.3 FUTURE WORK

There is plenty of scope for future work to build on the results of this project. There are opportunities to improve the system itself and opportunities to apply the system to new areas.

7.3.1 Fixes and improvements

The Elapse device has two problems that need to be corrected in order for it to be of practical use: *i)* the dry EEG electrodes need to be replaced with a better-performing design, and *ii)* the IMU device driver needs to be fixed or replaced in order to enable the gyroscopes and increase the sampling rate.

In addition to these fixes, there are a number of improvements that could be made to the system. As far as the design of the signal processing pipeline is concerned, it could be useful to provide coupling between feature extractors. This could enable an EEG feature extractor to use the output of the eye video feature extractor to help remove eye blink artefacts, for example. This is difficult to implement when the user is allowed to select any combination of plug-ins, though, because of the possibility of circular dependencies. It may also be useful to allow the user to select multiple output actions to execute in parallel—e.g., display an alert and sound an alarm and activate a vibration motor.

Regarding the signal processing code that has been implemented so far, it would be useful to explicitly track the position of the eyelids in the eye video. This would provide more accurate measurement of eye closure and eyelid speed, rather than relying only on the size and shape of the visible pupil region. Once the IMU driver has been fixed, it would also be useful to implement a better orientation estimation algorithm, e.g., Madgwick et al. [2011].

To improve the usability of the Elapse client software it would be useful to be able to load complete system configurations from a file, including hardware configuration and plug-in selection. This would allow a project to define all of its settings in one place and conveniently load them in one step. Currently only the most recently used settings are loaded from file. For developers that are unfamiliar with C++ and Python, it could be useful to implement a Matlab plug-in host to enable them to write plug-ins in Matlab. It may also be useful to provide a scripting interface to the client so that, for example, external stimulus presentation software could start and stop data capture.

If it becomes necessary to update the operating system on the Elapse device, the OV7735 camera driver will need to be ported to the new `v4l2_subdev` API, the ADS1299 driver may need to be updated, and the video encoding pipeline will need to be adjusted to use the updated GStreamer DSP H.264 encoder element. The server software could also be updated to use Qt 5 instead of Qt 4. Most of the work to implement these changes is likely to be in understanding the differences between the current and updated systems; the actual implementation should be fairly straightforward once the required changes are understood.

7.3.2 Extensions and applications

The primary purpose of this project was to develop a system that can be used for detecting lapses. The most important extension of this project, then, is to implement the feature extractors, classifier, and output actions necessary to make a complete lapse detection device. This work will be carried on by students in NeuroTech. One of the important questions that they will need to investigate is whether all of the sensors are required for detecting lapses. It may turn out that using ICA to extract EOG from frontal EEG is sufficient for measuring eye closure, making the eye video camera unnecessary. It may also turn out that a reduced set of EEG electrodes is sufficient which could affect the form of the device on the head. If this happened, the Elapse device would continue to be useful as a research tool and for non–lapse-detection applications, even if the ultimate form of a commercial lapse detection device ended up being different.

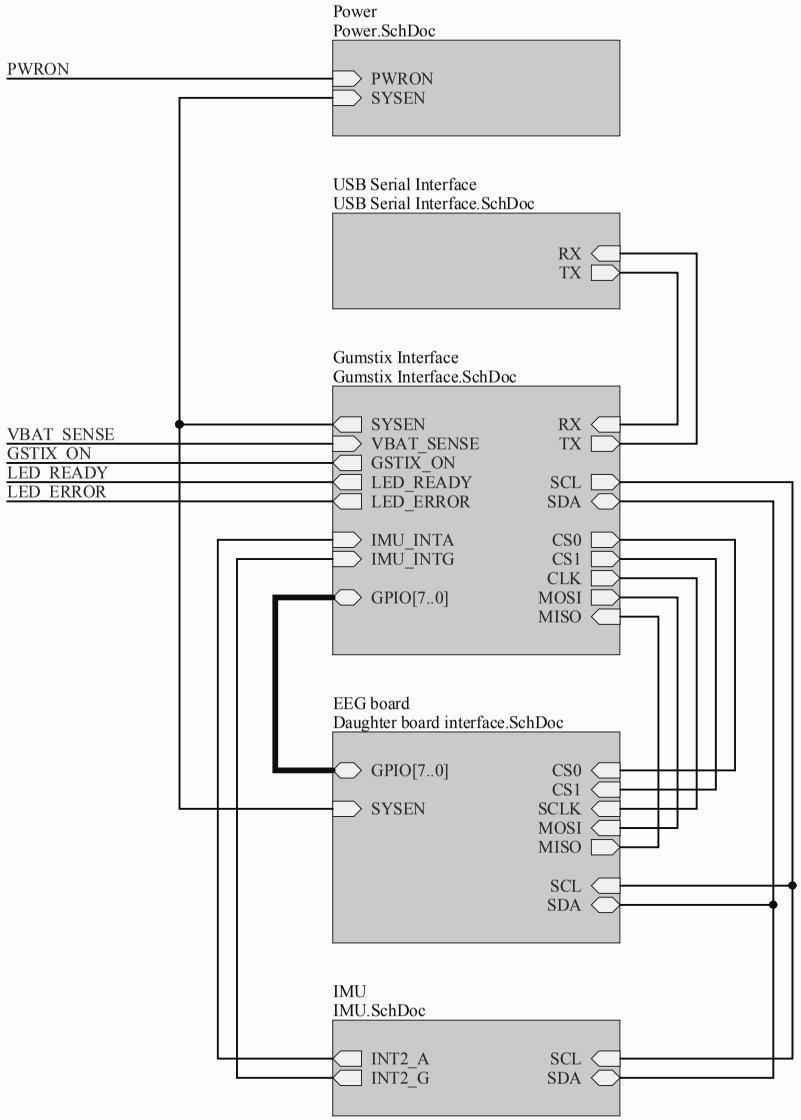
A secondary aim of this project was to design the system to be useful for applications other than lapse detection. The Elapse device and signal processing pipeline design are sufficiently general to allow this, so it would be possible to use the system for the applications listed in Section 2.5—e.g., workload monitoring or augmented cognition.

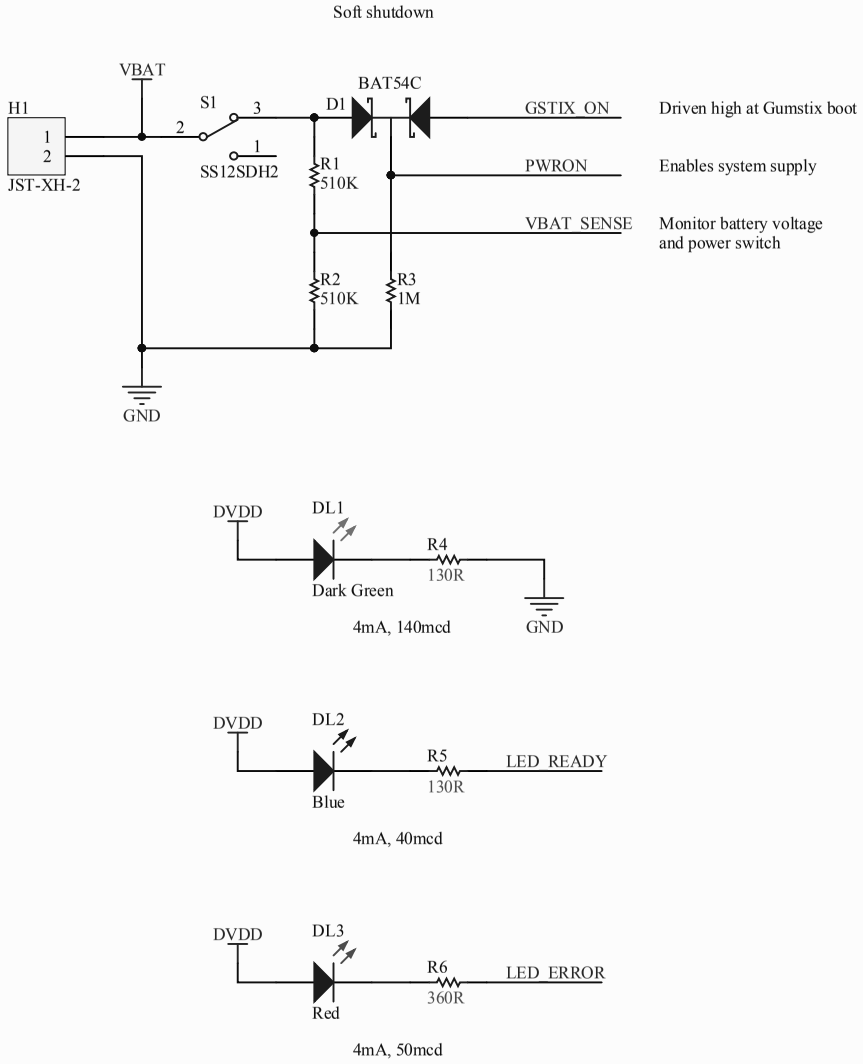
Appendix A

ELECTRONICS SCHEMATICS

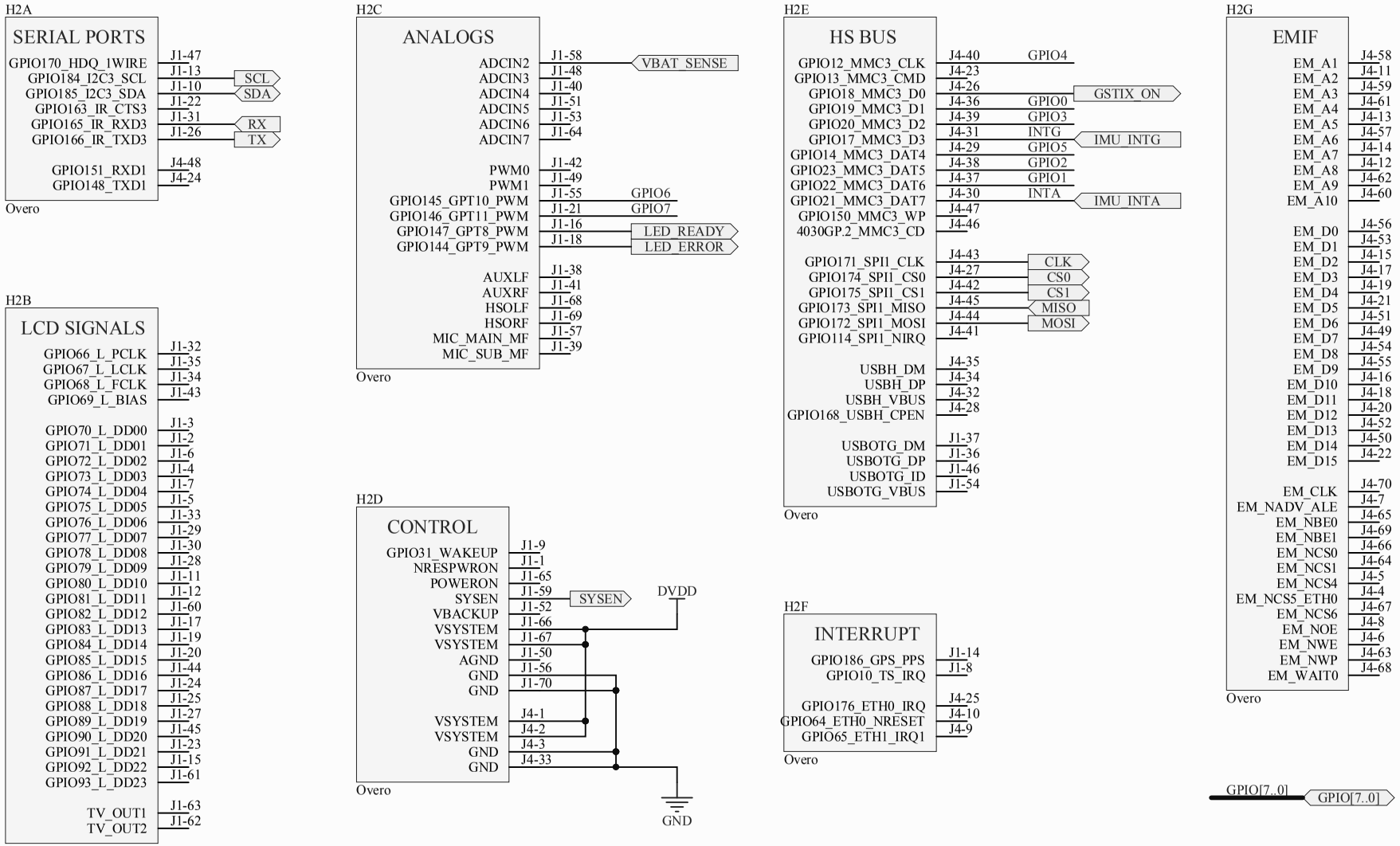
The schematics for all of the electronics developed for this project are included in this appendix. For a detailed description of the hardware functionality and design rationale, see Section 3.2.

Contents



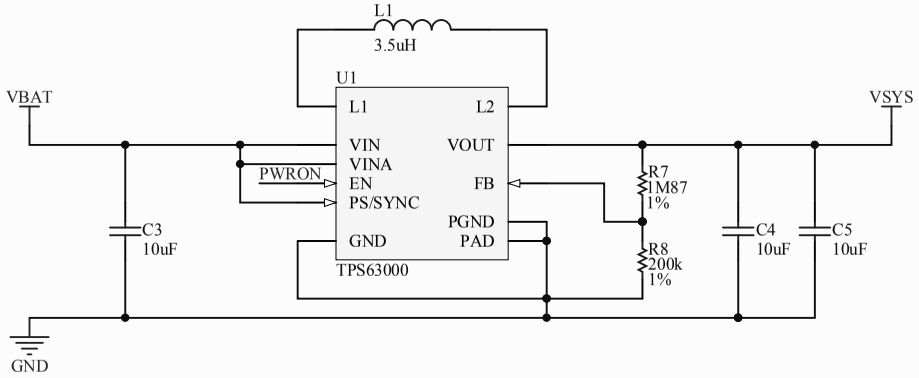


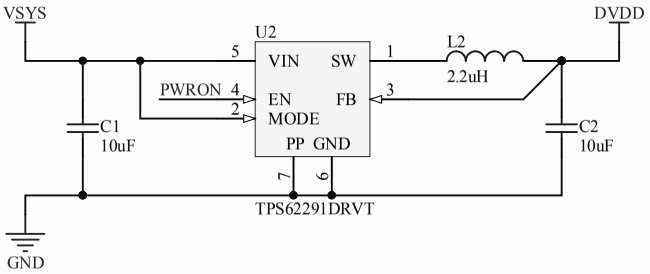
Title ***Elapse base board***		*NeuroTech & ECE Department* *University of Canterbury* *Christchurch 8041* *New Zealand*	UNIVERSITY OF CANTERBURY Te Whare Wānanga o Waitaha CHRISTCHURCH NEW ZEALAND
Drawn by: Simon Knopp	Revision: 1		
Date: 21/11/2013 Time: 1:02:25 p.m.	Sheet 1 of 6		
File: Elapse base.SchDoc			



Title *Elapse base: Gumstix interface*		*NeuroTech & ECE Department University of Canterbury Christchurch 8041 New Zealand*
Drawn by: Simon Knopp	Revision: 1	
Date: 21/11/2013 Time: 1:02:25 p.m.	Sheet 2 of 6	
File: Gumstix Interface.SchDoc		

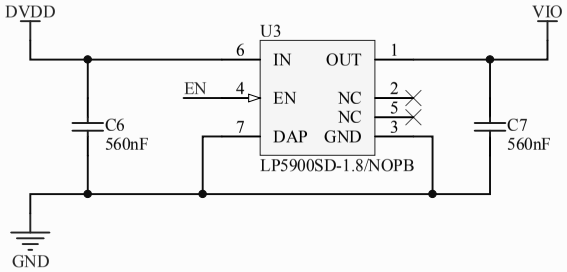
UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND





PWRON — From power switch: enables switching regulators

SYSEN — From Gumstix PMIC: enables linear regulators



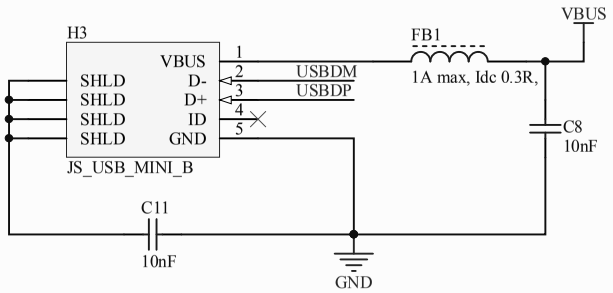
Power supplies:

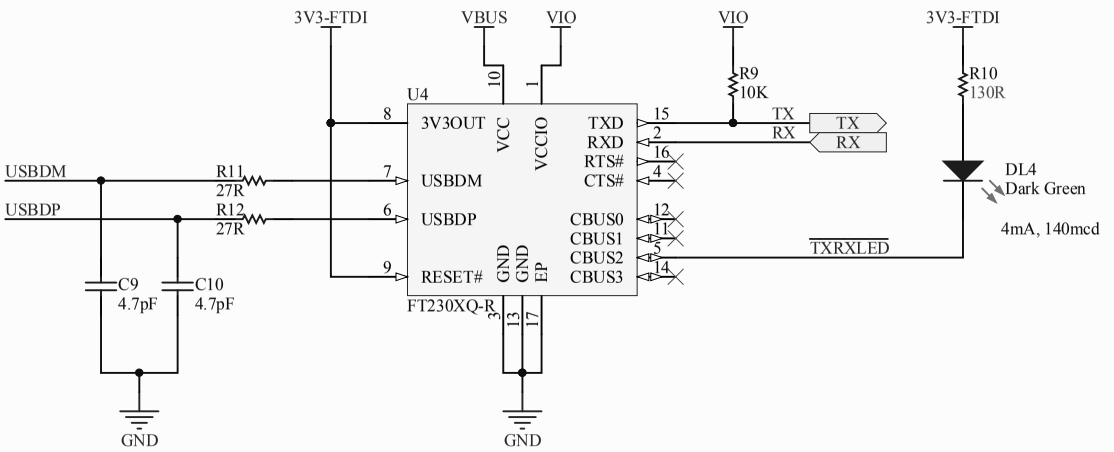
VBAT is the raw battery voltage
VSYS is the 5.2 V system supply
DVDD is the main 3.3 V supply for the Gumstix
VIO is a 1.8 V supply for logic that interfaces with the Gumstix

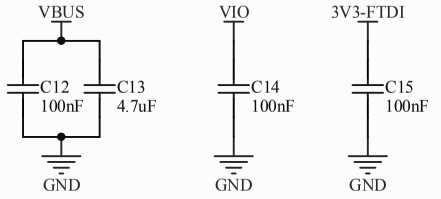
Battery is a single-cell Li-Ion. Nominal 3.6 V. Actual 2.7 -- 4.0 V.

Title ***Elapse base: Power supply***		*NeuroTech & ECE Department* *University of Canterbury* *Christchurch 8041* *New Zealand*
Drawn by: Simon Knopp	Revision: 1	
Date: 21/11/2013 Time: 1:02:25 p.m.	Sheet 3 of 6	
File: Power.SchDoc		

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND







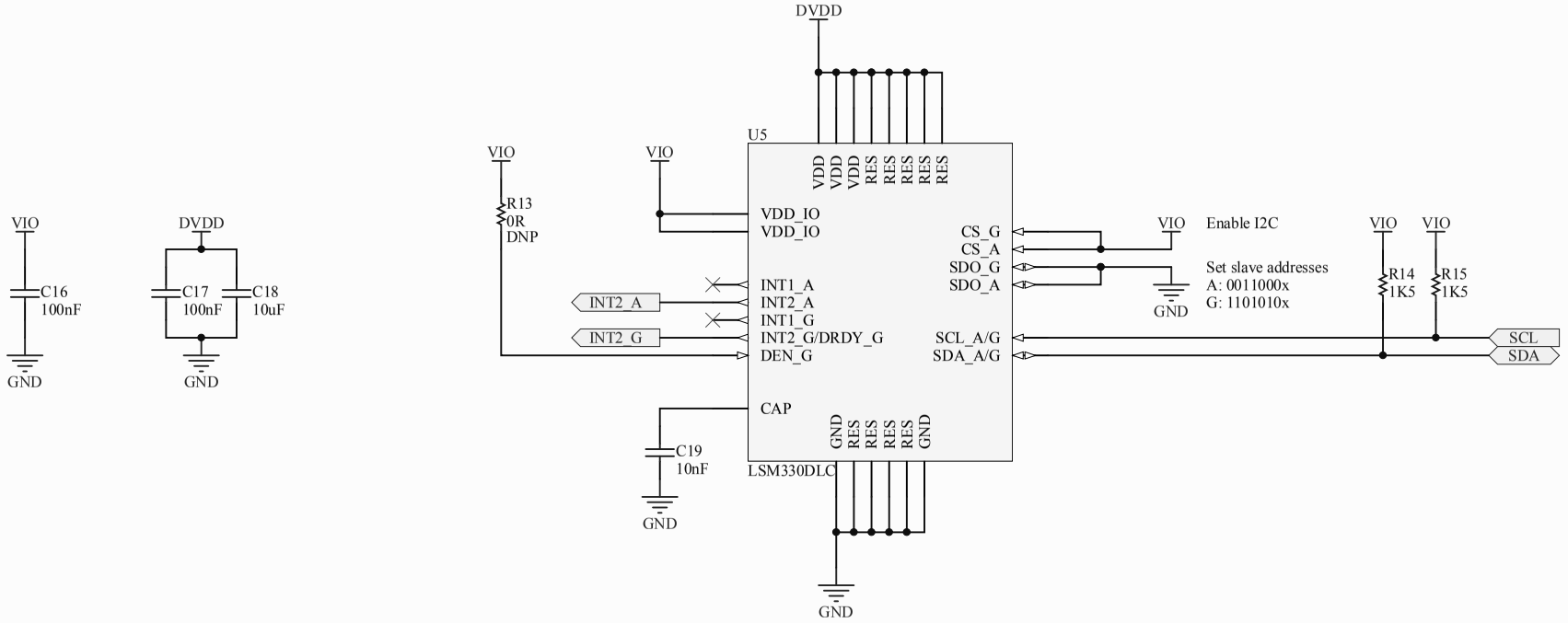
NOTES:

Use a bus-powered configuation so that the FT230X is only powered up when there is a USB connection.

CBUS2 defaults to TXLED. The configuration needs to be changed to TXRXLED using the FT_PROG tool (available from FTDI).

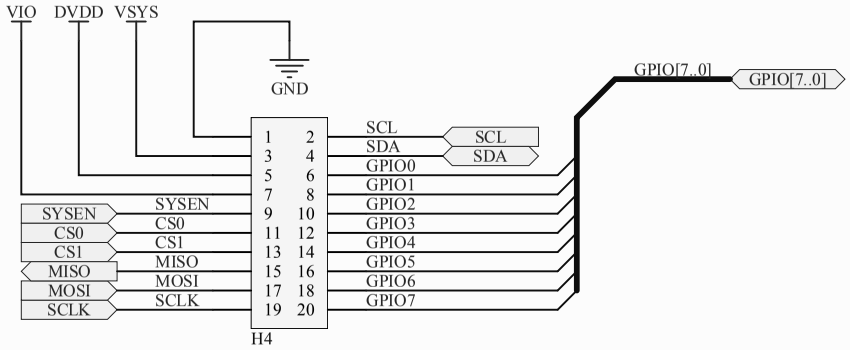
Title ***Elapse base: USB-UART interface***		*NeuroTech & ECE Department University of Canterbury Christchurch 8041 New Zealand*
Drawn by: Simon Knopp	Revision: 1	
Date: 21/11/2013 Time: 1:02:25 p.m.	Sheet 4 of 6	
File: USB Serial Interface.SchDoc		

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

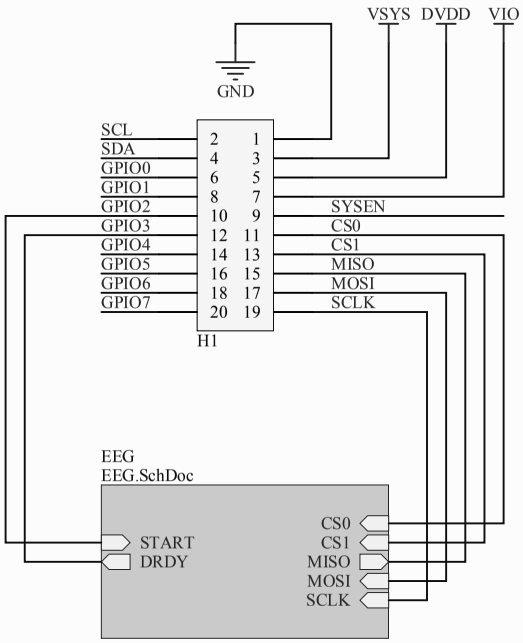


Title	***Elapse base: Inertial measurement***		*NeuroTech & ECE Department University of Canterbury Christchurch 8041 New Zealand*
Drawn by: Simon Knopp		Revision: 1	
Date: 21/11/2013 Time: 1:02:26 p.m.		Sheet 5 of 6	
File: IMU.SchDoc			

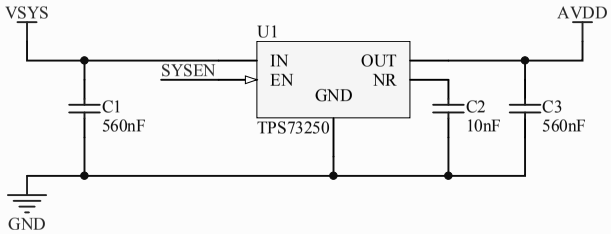
UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

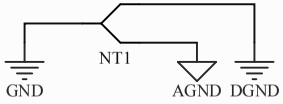


Title	***Elapse base: Daughter board interface***	*Neurotech & ECE Department University of Canterbury Christchurch 8041 New Zealand*	UNIVERSITY OF CANTERBURY Te Whare Wānanga o Waitaha
Drawn by: Simon Knopp	Revision: 1		
Date: 21/11/2013 Time: 1:02:26 p.m.	Sheet 6 of 6		
File: Daughter board interface.SchDoc			

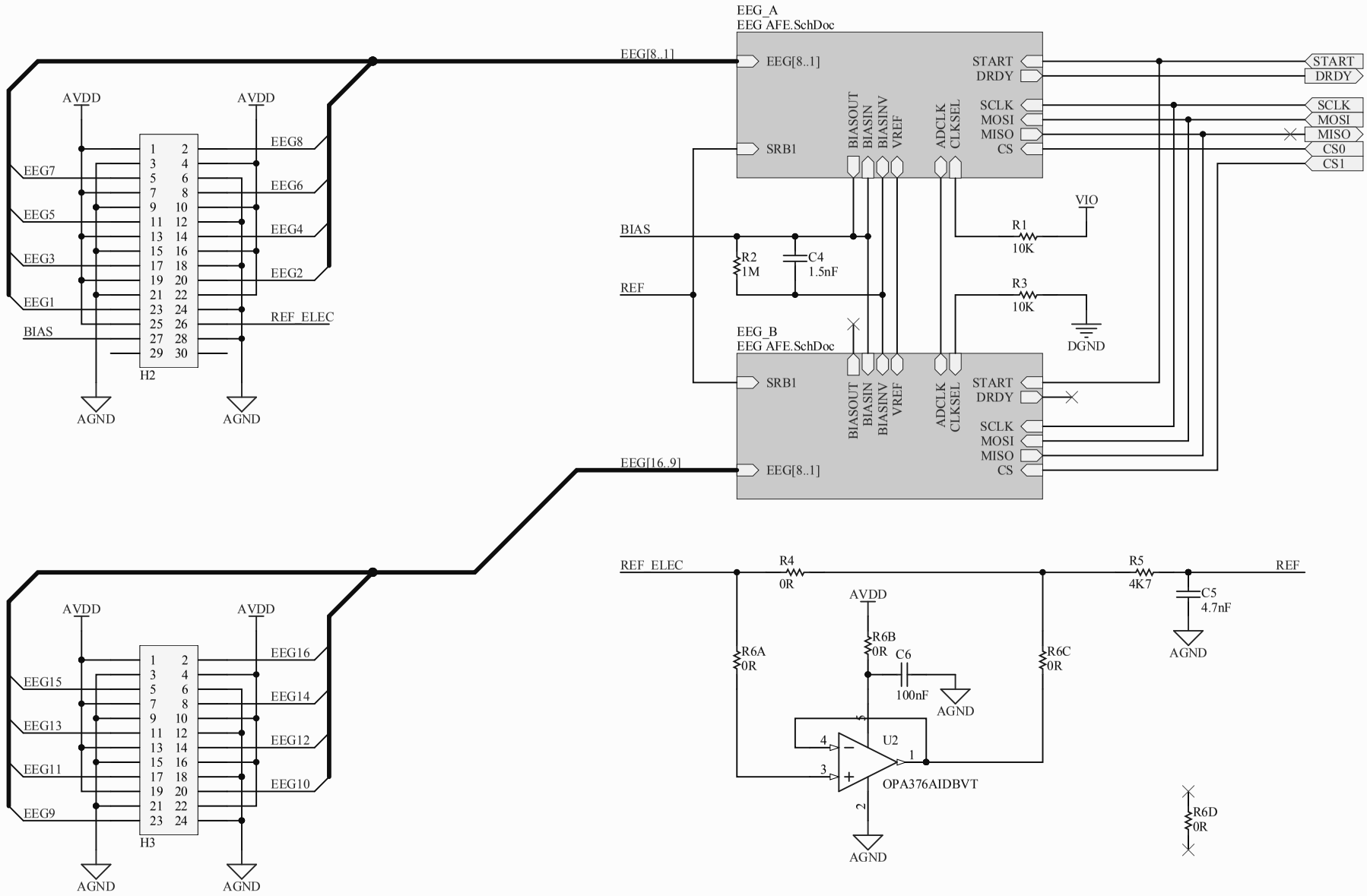


AVDD is a 5V supply for the EEG AFEs



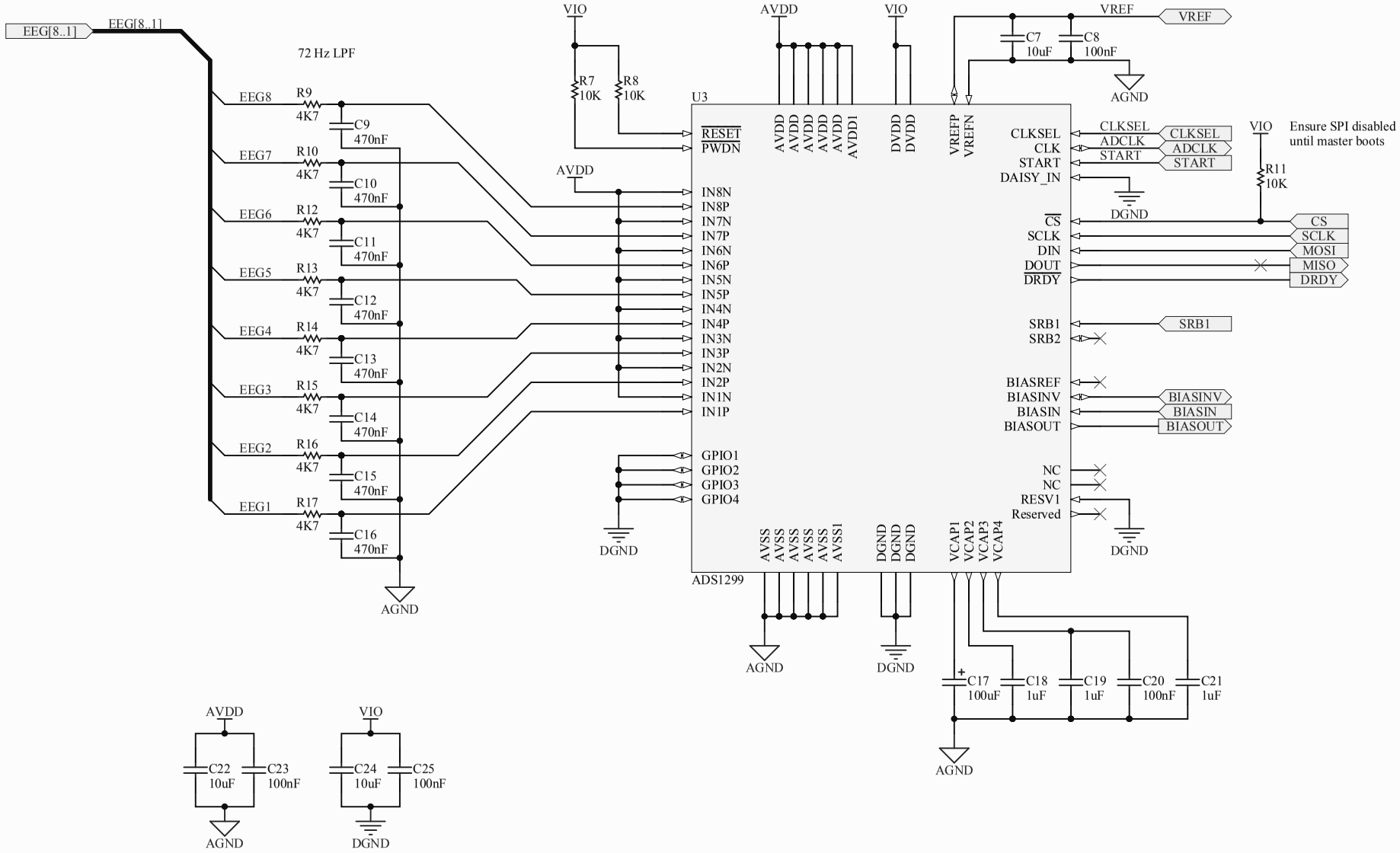


Title ***Elapse base board***		*NeuroTech & ECE Department University of Canterbury Christchurch 8041 New Zealand*	UNIVERSITY OF CANTERBURY Te Whare Wānanga o Waitaha CHRISTCHURCH NEW ZEALAND
Drawn by: Simon Knopp	Revision: 1		
Date: 21/11/2013 Time: 1:01:23 p.m.	Sheet 1 of 3		
File: Elapse EEG.SchDoc			



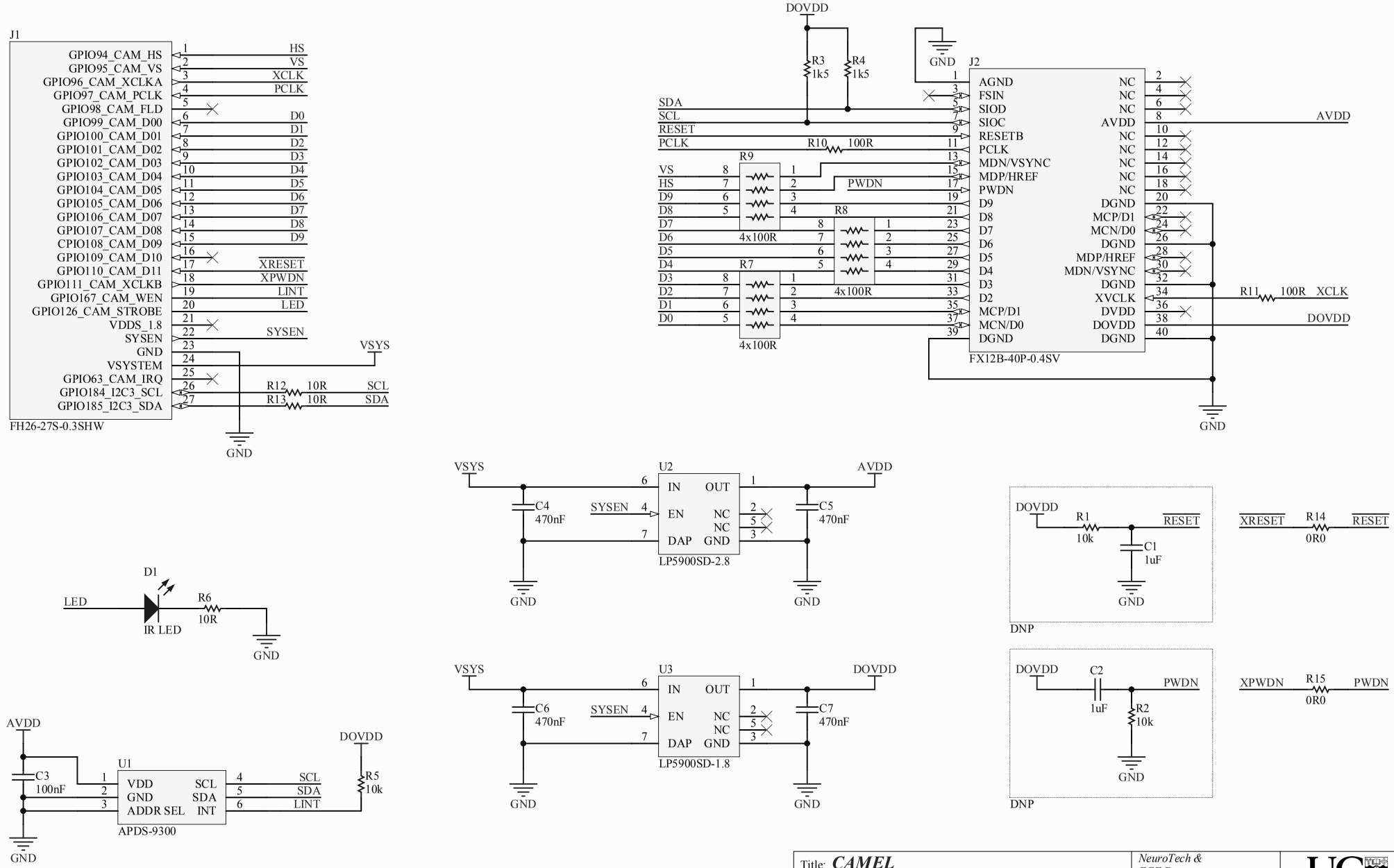
Title ***Elapse EEG board***		*Neurotech & ECE Department University of Canterbury Christchurch 8041 New Zealand*
Drawn by: Simon Knopp	Revision: 1	
Date: 21/11/2013 Time: 1:01:23 p.m.	Sheet 2 of 3	
File: EEG.SchDoc		

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha



Title *Elapse EEG: Analogue front end*		*Neurotech & ECE Department University of Canterbury Christchurch 8041 New Zealand*
Drawn by: Simon Knopp	Revision: 1	
Date: 21/11/2013 Time: 1:01:23 p.m.	Sheet 3 of 3	
File: EEG AFE.SchDoc		

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND



Title: ***CAMEL***		*NeuroTech & ECE Department*	UC UNIVERSITY OF CANTERBURY
Drawn by: Simon Knopp	Revision: 1	*University of Canterbury*	
Date: 2/3/2012 Time: 12:38:48 PM	Sheet 1 of 1	*Christchurch 8041*	
File: \\maxwell\sjk114$\MyDocuments\elapse\CAMEL\CAMEL.SchDoc		*New Zealand*	

REFERENCES

Álvarez-Estévez, D., Fernández-Pastoriza, J. and Moret-Bonillo, V. (2009), 'A continuous evaluation of the awake sleep state using fuzzy reasoning', In *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5539–5542.

Ayyagari, S., Jones, R. and Weddell, S. (2014), 'EEG-based event detection using optimized echo state networks with leaky integrator neurons', In *Proceedings of the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5856–5859.

Babcock, J.S. and Pelz, J.B. (2004), 'Building a lightweight eyetracking headgear', In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*, ETRA '04, ACM, pp. 109–114.

Bainbridge, L. (1983), 'Ironies of automation', *Automatica*, Vol. 19, No. 6, pp. 775–779.

Beazley, D. (2010), 'Understanding the Python GIL', In *Proceedings of the Python Conference (PyCon)*, Python Software Foundation.

Berka, C., Levendowski, D.J., Cvetinovic, M.M., Petrovic, M.M., Davis, G., Lumicao, M.N., Zivkovic, V.T., Popovic, M.V. and Olmstead, R.E. (2004), 'Real-time analysis of EEG indexes of alertness, cognition, and memory acquired with a wireless EEG headset', *International Journal of Human-Computer Interaction*, Vol. 17, No. 2, pp. 151–170.

Berka, C., Levendowski, D.J., Lumicao, M.N., Yau, A., Davis, G., Zivkovic, V.T., Olmstead, R.E., Tremoulet, P.D. and Craven, P.L. (2007), 'EEG correlates of task engagement and mental workload in vigilance, learning, and memory tasks', *Aviation, Space, and Environmental Medicine*, Vol. 78, No. Supplement 1, pp. B231–B244.

Borghini, G., Astolfi, L., Vecchiato, G., Mattia, D. and Babiloni, F. (2014), 'Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness', *Neuroscience & Biobehavioral Reviews*, Vol. 44, No. 0, pp. 58–75.

Bos, D.P.O., Reuderink, B., van de Laar, B., Gürkök, H., Mühl, C., Poel, M., Nijholt, A. and Heylen, D. (2010), 'Brain–computer interfacing and games', In D.S. Tan and A. Nijholt (editors), *Brain–Computer Interfaces*, Human–Computer Interaction Series, pp. 149–178, Springer, London.

Caffier, P.P., Erdmann, U. and Ullsperger, P. (2003), 'Experimental evaluation of eye-blink parameters as a drowsiness measure', *European Journal of Applied Physiology*, Vol. 89, No. 3-4, pp. 319–325.

Chi, Y. and Cauwenberghs, G. (2009), 'Micropower non-contact EEG electrode with active common-mode noise suppression and input capacitance cancellation', In *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, pp. 4218–4221.

Chi, Y.M., Wang, Y., Wang, Y.T., Jung, T.P., Kerth, T. and Cao, Y. (2013), 'A practical mobile dry EEG system for human computer interfaces', In D.D. Schmorrow and C.M. Fidopiastis (editors), *Foundations of Augmented Cognition*, Vol. 8027 of *Lecture Notes in Computer Science*, pp. 649–655, Springer, Berlin/Heidelberg.

Chi, Y.M., Wang, Y.T., Wang, Y., Maier, C., Jung, T.P. and Cauwenberghs, G. (2012), 'Dry and noncontact EEG sensors for mobile brain–computer interfaces', *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 20, No. 2, pp. 228–235.

Chiou, J., Ko, L., Lin, C., Hong, C., Jung, T., Liang, S. and Jeng, J. (2006), 'Using novel MEMS EEG sensors in detecting drowsiness application', In *Proceedings of the IEEE Biomedical Circuits and Systems Conference*, IEEE, pp. 33–36.

Crane, H.D. and Steele, C.M. (1985), 'Generation-V dual-Purkinje-image eyetracker', *Applied Optics*, Vol. 24, No. 4, pp. 527–537.

Crummy, F., Cameron, P.A., Swann, P., Kossmann, T. and Naughton, M.T. (2008), 'Prevalence of sleepiness in surviving drivers of motor vehicle collisions', *Internal Medicine Journal*, Vol. 38, No. 10, pp. 769–775.

Dal Seno, B., Matteucci, M. and Mainardi, L. (2010), 'Online detection of P300 and error potentials in a BCI speller', *Computational Intelligence and Neuroscience*, Vol. 2010, pp. 11:1–11:1.

Daugman, J. (2004), 'How iris recognition works', *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14, No. 1, pp. 21–30.

Davidson, P.R., Jones, R.D. and Peiris, M.T.R. (2007), 'EEG-based lapse detection with high temporal resolution', *IEEE Transactions on Biomedical Engineering*, Vol. 54, No. 5, pp. 832–839.

Dinges, D.F., Maislin, G., Powell, J.W. and Mallis, M.M. (1998), 'Evaluation of techniques for ocular measurement as an index of fatigue and the basis for alertness management', Tech. Rep. DOT HS 808 762, National Highway Traffic Safety Administration (USA).

Donchin, E., Spencer, K.M. and Wijesinghe, R. (2000), 'The mental prosthesis: assessing the speed of a P300-based brain–computer interface', *IEEE Transactions on Rehabilitation Engineering*, Vol. 8, No. 2, pp. 174–179.

Drummond, S.P., Bischoff-Grethe, A., Dinges, D.F., Ayalon, L., Mednick, S.C. and Meloy, M. (2005), 'The neural basis of the psychomotor vigilance task', *Sleep*, Vol. 28, No. 9, pp. 1059–1068.

Duda, R.O. and Hart, P.E. (1972), 'Use of the Hough transformation to detect lines and curves in pictures', *Communications of the ACM*, Vol. 15, No. 1, pp. 11–15.

Duffy, F.H., Iyer, V.G. and Surwillo, W.W. (1989), *Clinical electroencephalography and topographic brain mapping*, Springer-Verlag.

Elder, M.J. (1997), 'Anatomy and physiology of eyelash follicles: relevance to lash ablation procedures', *Ophthalmic Plastic & Reconstructive Surgery*, Vol. 13, No. 1, pp. 21–25.

Fischler, M.A. and Bolles, R.C. (1981), 'Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography', *Communications of the ACM*, Vol. 24, No. 6, pp. 381–395.

Fitzgibbon, A., Pilu, M. and Fisher, R.B. (1999), 'Direct least square fitting of ellipses', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 5, pp. 476–480.

Fonseca, C., Cunha, J., Martins, R., Ferreira, V., de Sá, J., Barbosa, M. and da Silva, A. (2007), 'A novel dry active electrode for EEG recording', *IEEE Transactions on Biomedical Engineering*, Vol. 54, No. 1, pp. 162–165.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1994), *Design patterns: elements of reusable object-oriented software*, Pearson Education.

Golz, M., Sommer, D., Chen, M., Trutschel, U. and Mandic, D. (2007), 'Feature fusion for the detection of microsleep events', *Journal of VLSI Signal Processing*, Vol. 49, No. 2, pp. 329–342.

Grace, R. (2001), 'Drowsy driver monitor and warning system', In *International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, pp. 201–208.

Grozea, C., Voinescu, C.D. and Fazli, S. (2011), 'Bristle-sensors — low-cost flexible passive dry EEG electrodes for neurofeedback and BCI applications', *Journal of Neural Engineering*, Vol. 8, No. 2, p. 025008.

Guger, C., Krausz, G. and Edlinger, G. (2011), 'Brain–computer interface control with dry EEG electrodes', In *Proceedings of the 5th International Brain–Computer Interface Conference*, p. 312.

Hammoud, R.I. and Zhang, H. (2008), 'Alertometer: Detecting and mitigating driver drowsiness and fatigue using an integrated human factors and computer vision approach', In R.I. Hammoud (editor), *Passive Eye Monitoring*, Springer Series on Signals and Communication Technology, chap. 14, pp. 301–321, Springer-Verlag.

Hartley, L., Horberry, T., Mabbott, N. and Krueger, G.P. (2000), 'Review of fatigue detection and prediction technologies', Tech. rep., National Road Transport Commission (Australia).

Hirata, Y., Nishiyama, J. and Kinoshita, S. (2009), 'Detection and prediction of drowsiness by reflexive eye movements', In *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4015–4018.

Hussain, A., Bais, B., Samad, S. and Hendi, S. (2008), 'Novel data fusion approach for drowsiness detection', *Information Technology Journal*, Vol. 7, No. 1, pp. 48–55.

Innes, C.R.H., Poudel, G.R., Signal, T.L. and Jones, R.D. (2010), 'Behavioural microsleeps in normally-rested people', In *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, pp. 4448–4451.

Innes, C.R., Poudel, G.R. and Jones, R.D. (2013), 'Efficient and regular patterns of nighttime sleep are related to increased vulnerability to microsleeps following a single night of sleep restriction', *Chronobiology International*, Vol. 30, No. 9, pp. 1187–1196.

Jäger, C. (2010), *Eye safety of IREDs used in lamp applications*, Osram Opto Semiconductors.

Ji, Q. and Yang, X. (2002), 'Real-time eye, gaze, and face pose tracking for monitoring driver vigilance', *Real-Time Imaging*, Vol. 8, No. 5, pp. 357–377.

Johns, M., Tucker, A., Chapman, R., Crowley, K. and Michael, N. (2007), 'Monitoring eye and eyelid movements by infrared reflectance oculography to measure drowsiness in drivers', *Somnologie-Schlafforschung und Schlafmedizin*, Vol. 11, No. 4, pp. 234–242.

Johnson, R.R., Popovic, D.P., Olmstead, R.E., Stikic, M., Levendowski, D.J. and Berka, C. (2011), 'Drowsiness/alertness algorithm development and validation using synchronized EEG and cognitive performance to individualize a generalized model', *Biological Psychology*, Vol. 87, No. 2, pp. 241–250.

Jones, R.D., Poudel, G.R., Innes, C.R.H., Davidson, P.R., Peiris, M.T.R., Malla, A.M., Signal, T.L., Carroll, G.J., Watts, R. and Bones, P.J. (2010), 'Lapses of responsiveness: Characteristics, detection, and underlying mechanisms', In *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, pp. 1788–1791.

Jonmohamadi, Y., Poudel, G., Innes, C. and Jones, R. (2014), 'Source-space ICA for EEG source separation, localization, and time-course reconstruction', *NeuroImage*, Vol. 101, pp. 720–737.

Jung, T.P., Huang, K.C., Chuang, C.H., Chen, J.A., Ko, L.W., Chiu, T.W. and Lin, C.T. (2010), 'Arousing feedback rectifies lapse in performance and corresponding EEG power spectrum', In *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, pp. 1792–1795.

Kaefer, G., Prochart, G. and Weiss, R. (2003), 'Wearable alertness monitoring for industrial applications', In *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, Citeseer, pp. 254–255.

Kar, S., Routray, A. and Nayak, B.P. (2011), 'Functional network changes associated with sleep deprivation and fatigue during simulated driving: Validation using blood biomarkers', *Clinical Neurophysiology*, Vol. 122, No. 5, pp. 966–974.

Kingsley, S.A., Sriram, S., Pollick, A. and Marsh, J. (2004), 'Photrodes for physiological sensing', In *Proceedings of SPIE*, Vol. 5317, pp. 158–166.

Knipling, R. and Wierwille, W. (1994), 'Vehicle-based drowsy driver detection: Current status and future prospects', In *Proceedings of the Intelligent Vehicle Highway Systems (IVHS) America Fourth Annual Meeting*, pp. 245–256.

LaRocco, J., Innes, C.R., Bones, P.J., Weddell, S. and Jones, R.D. (2014), 'Optimal EEG feature selection from average distance between events and non-events', In *Proceedings of the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, pp. 2641–2644.

Leonhardt, S. and Aleksandrowicz, A. (2008), 'Non-contact ECG monitoring for automotive application', In *Proceedings of the 5th International Summer School and Symposium on Medical Devices and Biosensors*, pp. 183–185.

Li, D., Winfield, D. and Parkhurst, D.J. (2005), 'Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches', In *Proceedings of the IEEE Vision for Human–Computer Interaction Workshop at CVPR*, pp. 1–8.

Liao, L.D., Wang, I.J., Chen, S.F., Chang, J.Y. and Lin, C.T. (2011), 'Design, fabrication and experimental validation of a novel dry-contact sensor for measuring electroencephalography signals without skin preparation', *Sensors*, Vol. 11, No. 6, pp. 5819–5834.

Liao, L.D., Chen, C.Y., Wang, I.J., Chen, S.F., Li, S.Y., Chen, B.W., Chang, J.Y. and Lin, C.T. (2012), 'Gaming control using a wearable and wireless EEG-based brain–computer interface device with novel dry foam-based sensors', *Journal of neuroengineering and rehabilitation*, Vol. 9, No. 1, p. 5.

Lienhart, R. and Maydt, J. (2002), 'An extended set of Haar-like features for rapid object detection', In *Proceedings of the International Conference on Image Processing*, Vol. 1, pp. I–900–I–903.

Lin, C.T., Wu, R.C., Liang, S.F., Chao, W.H., Chen, Y.J. and Jung, T.P. (2005), 'EEG-based drowsiness estimation for safety driving using independent component analysis', *IEEE Transactions on Circuits and Systems—Part I: Regular Papers*, Vol. 52, No. 12, pp. 2726–2738.

Lin, C.T., Huang, K.C., Chuang, C.H., Ko, L.W. and Jung, T.P. (2013), 'Can arousing feedback rectify lapses in driving? Prediction from EEG power spectra', *Journal of Neural Engineering*, Vol. 10, No. 5, p. 056024, (10 pp.).

Lüdtke, H., Wilhelm, B., Adler, M., Schaeffel, F. and Wilhelm, H. (1998), 'Mathematical procedures in data recording and processing of pupillary fatigue waves', *Vision Research*, Vol. 38, No. 19, pp. 2889–2896.

Madgwick, S.O., Harrison, A.J. and Vaidyanathan, R. (2011), 'Estimation of IMU and MARG orientation using a gradient descent algorithm', In *Proceedings of the IEEE International Conference on Rehabilitation Robotics (ICORR)*, IEEE, pp. 1–7.

Malla, A.M., Davidson, P.R., Bones, P.J., Green, R. and Jones, R.D. (2010), 'Automated video-based measurement of eye closure for detecting behavioral microsleep', In *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, pp. 6741–6744.

Marshall, S. (2007), 'Identifying cognitive state from eye metrics', *Aviation, Space, and Environmental Medicine*, Vol. 78, No. Supplement 1, pp. B165–B175.

Matthews, R., McDonald, N., Anumula, H., Woodward, J., Turner, P., Steindorf, M., Chang, K. and Pendleton, J. (2007), 'Novel hybrid bioelectrodes for ambulatory zero-prep EEG measurements using multi-channel wireless EEG system', In D. Schmorrow and L. Reeves (editors), *Foundations of Augmented Cognition*, Vol. 4565 of *Lecture Notes in Computer Science*, pp. 137–146, Springer, Berlin/Heidelberg.

May, J. and Baldwin, C. (2009), 'Driver fatigue: The importance of identifying causal factors of fatigue when considering detection and countermeasure technologies', *Transportation Research Part F: Traffic Psychology and Behaviour*, Vol. 12, No. 3, pp. 218–224.

Merklinger, H. (1992), *The ins and outs of focus: an alternative way to estimate depth-of-field and sharpness in the photographic image*, `http://www.trenholm.org/hmmerk/TIAOOFe.pdf`.

Mullen, T., Kothe, C., Chi, Y.M., Ojeda, A., Kerth, T., Makeig, S., Cauwenberghs, G. and Jung, T.P. (2013), 'Real-time modeling and 3D visualization of source dynamics and connectivity using wearable EEG', In *Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2184–2187.

Nishino, K. and Nayar, S.K. (2004), 'Eyes for relighting', In *ACM Transactions on Graphics*, Vol. 23, ACM, pp. 704–711.

Nishiyama, J., Tanida, K., Kusumi, M. and Hirata, Y. (2007), 'The pupil as a possible premonitor of drowsiness', In *Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1586–1589.

OMAPpedia (2012), 'Camera-ISP driver', Retrieved from `http://omappedia.org/wiki/Camera-ISP_Driver`, accessed 22 July 2014.

OmniVision (2011a), *OV7735 Datasheet*, OmniVision Technologies, obtained under NDA.

OmniVision (2011b), *OV7735-MPSW Camera Module Datasheet*, OmniVision Technologies, obtained under NDA.

Osram (2012), *SFH4058 High Power Infrared Emitter Datasheet*, Osram Opto Semiconductors.

Pal, N., Chuang, C., Ko, L., Chao, C., Jung, T., Liang, S. and Lin, C. (2008), 'EEG-based subject- and session-independent drowsiness detection: an unsupervised approach', *EURASIP Journal on Advances in Signal Processing*, Vol. 2008, p. 192.

Papadelis, C., Chen, Z., Kourtidou-Papadeli, C., Bamidis, P.D., Chouvarda, I., Bekiaris, E. and Maglaveras, N. (2007), 'Monitoring sleepiness with on-board electrophysiological recordings for preventing sleep-deprived traffic accidents', *Clinical Neurophysiology*, Vol. 118, No. 9, pp. 1906–1922.

Park, J. (2011), 'Plastic optical fiber sensor for measuring driver-gripping force', *Optical Engineering*, Vol. 50, No. 2, p. 020501, (3 pp.).

Patki, S., Grundlehner, B., Nakada, T. and Penders, J. (2011), 'Low power wireless EEG headset for BCI applications', In J. Jacko (editor), *Human-Computer Interaction. Interaction Techniques and Environments*, Vol. 6762 of *Lecture Notes in Computer Science*, pp. 481–490, Springer, Berlin/Heidelberg.

Peiris, M.T.R., Jones, R.D., Davidson, P.R., Carroll, G., Signal, T., Parkin, P., van den Berg, M. and Bones, P.J. (2005), 'Identification of vigilance lapses using EEG/EOG by expert human raters', In *Proceedings of the 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, pp. 5735–5737.

Peiris, M.T.R., Jones, R.D., Davidson, P.R., Carroll, G.J. and Bones, P.J. (2006), 'Frequent lapses of responsiveness during an extended visuomotor tracking task in non-sleep-deprived subjects', *Journal of Sleep Research*, Vol. 15, No. 3, pp. 291–300.

Peiris, M.T.R., Davidson, P.R., Bones, P.J. and Jones, R.D. (2011), 'Detection of lapses in responsiveness from the EEG', *Journal of Neural Engineering*, Vol. 8, p. 016003.

Peiris, M., Jones, R., Davidson, P.R. and Bones, P. (2008), 'Event-based detection of lapses of responsiveness', In *Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4960–4963.

Pérez, A., Córdoba, M., García, A., Méndez, R., Muñoz, M., Pedraza, J. and Sánchez, F. (2003), 'A precise eye-gaze detection and tracking system', In *11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, Citeseer.

Popieul, J., Simon, P. and Loslever, P. (2003), 'Using driver's head movements evolution as a drowsiness indicator', In *Proceedings of the Intelligent Vehicles Symposium*, IEEE, pp. 616–621.

Poudel, G.R., Jones, R.D. and Innes, C.R.H. (2008), 'A 2-D pursuit tracking task for behavioural detection of lapses', *Australasian Physical & Engineering Sciences in Medicine*, Vol. 31, No. 4, pp. 528–529.

Poudel, G.R., Jones, R.D., Innes, C.R.H., Watts, R., Signal, T.L. and Bones, P.J. (2009), 'fMRI correlates of behavioural microsleeps during a continuous visuomotor task', In *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, pp. 2919–2922.

Poudel, G.R., Innes, C.R.H., Bones, P.J. and Jones, R.D. (2010), 'The relationship between behavioural microsleeps, visuomotor performance and EEG theta', In *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, pp. 4452–4455.

Poudel, G.R., Innes, C.R. and Jones, R.D. (2012), 'Cerebral perfusion differences between drowsy and nondrowsy individuals after acute sleep restriction', *Sleep*, Vol. 35, No. 8, pp. 1085–96.

Poudel, G.R., Innes, C.R. and Jones, R.D. (2013), 'Distinct neural correlates of time-on-task and transient errors during a visuomotor tracking task after sleep restriction', *NeuroImage*, Vol. 77, pp. 105–113.

Poudel, G.R., Innes, C.R., Bones, P.J., Watts, R. and Jones, R.D. (2014), 'Losing the struggle to stay awake: Divergent thalamic and cortical activity during microsleeps', *Human Brain Mapping*, Vol. 35, No. 1, pp. 257–269.

Pree, W. (1994), 'Meta patterns—a means for capturing the essentials of reusable object-oriented design', In *Object-oriented programming*, pp. 150–162, Springer.

Reeves, L.M., Schmorrow, D.D. and Stanney, K.M. (2007), 'Augmented cognition and cognitive state assessment technology – near-term, mid-term, and long-term research objectives', In *Foundations of Augmented Cognition*, pp. 220–228, Springer.

Saari, T., Turpeinen, M., Kuikkaniemi, K., Kosunen, I. and Ravaja, N. (2009), 'Emotionally adapted games – an example of a first person shooter', In J.A. Jacko (editor), *Human–Computer Interaction. Interacting in Various Application Domains*, Vol. 5613 of *Lecture Notes in Computer Science*, pp. 406–415, Springer, Berlin/Heidelberg.

Santamaria, J. and Chiappa, K.H. (1987), 'The EEG of drowsiness in normal adults', *Journal of Clinical Neurophysiology*, Vol. 4, No. 4.

Schleicher, R., Galley, N., Briest, S. and Galley, L. (2008), 'Blinks and saccades as indicators of fatigue in sleepiness warnings: looking tired?', *Ergonomics*, Vol. 51, No. 7, pp. 982–1010.

Searle, A. and Kirkup, L. (2000), 'A direct comparison of wet, dry and insulating bioelectric recording electrodes', *Physiological Measurement*, Vol. 21, No. 2, p. 271.

Shen, J., Barbera, J. and Shapiro, C.M. (2006), 'Distinguishing sleepiness and fatigue: focus on definition and measurement', *Sleep Medicine Reviews*, Vol. 10, No. 1, pp. 63–76.

Sigari, M. (2009), 'Driver hypo-vigilance detection based on eyelid behavior', In *Proceedings of the 7th International Conference on Advances in Pattern Recognition*, IEEE, pp. 426–429.

Sigurdson, K. and Ayas, N.T. (2007), 'The public health and safety consequences of sleep disorders.', *Canadian Journal of Physiology & Pharmacology*, Vol. 85, No. 1, pp. 179–183.

Silva, R. (2006), 'Nanotechnology based dry sensors for the recording of electrophysical signals', *Journal of Sleep Research*, Vol. 15, No. Supplement 1, pp. 16–17.

Stanney, K.M., Schmorrow, D.D., Johnston, M., Fuchs, S., Jones, D., Hale, K.S., Ahmad, A. and Young, P. (2009), 'Augmented cognition: An overview', *Reviews of human factors and ergonomics*, Vol. 5, No. 1, pp. 195–224.

Stevens, R., Galloway, T. and Berka, C. (2006), 'Integrating EEG models of cognitive load with machine learning models of scientific problem solving', *Augmented Cognition: Past, Present and Future*, Vol. 2, pp. 55–65.

Świrski, L., Bulling, A. and Dodgson, N. (2012), 'Robust real-time pupil tracking in highly off-axis images', In *Proceedings of the 2012 Symposium on Eye Tracking Research & Applications*, ETRA '12, ACM, pp. 173–176.

Torsvall, L. and Åkerstedt, T. (1987), 'Sleepiness on the job: continuously measured EEG changes in train drivers', *Electroencephalography and Clinical Neurophysiology*, Vol. 66, No. 6, pp. 502–511.

Tsai, P., Hu, W., Kuo, T. and Shyu, L. (2009), 'A portable device for real time drowsiness detection using novel active dry electrode system', In *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, pp. 3775–3778.

Tsuchida, A., Bhuiyan, M. and Oguri, K. (2009), 'Estimation of drowsiness level based on eyelid closure and heart rate variability', In *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2543–2546.

Tucker, A.J. and Johns, M.W. (2005), 'The duration of eyelid movements during blinks: changes with drowsiness', *Sleep*, Vol. 28, p. A122.

Vanlaar, W., Simpson, H., Mayhew, D. and Robertson, R. (2008), 'Fatigued and drowsy driving: A survey of attitudes, opinions and behaviors', *Journal of Safety Research*, Vol. 39, No. 3, pp. 303–309.

Viola, P. and Jones, M.J. (2004), 'Robust real-time face detection', *International Journal of Computer Vision*, Vol. 57, pp. 137–154.

Weissman, D.H., Roberts, K.C., Visscher, K.M. and Woldorff, M.G. (2006), 'The neural bases of momentary lapses in attention', *Nature Neuroscience*, Vol. 9, No. 7, pp. 971–978.

Wierwille, W. and Ellsworth, L. (1994), 'Evaluation of driver drowsiness by trained raters', *Accident Analysis & Prevention*, Vol. 26, No. 5, pp. 571–581.

Wilson, G.F. (2009), 'Validation of a dry electrode system for EEG', In *Proceedings of the Human Factors and Ergonomics Society 53rd Annual Meeting*, pp. 1171–1175.

elapse