



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Development of advanced closed-loop brain electrophysiology systems for freely behaving rodents

Dissertation submitted in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy in Electronic Engineering

Author: **Aarón Cuevas López**

Supervisors: **Prof. Dr. David Moratal Pérez**

Center for Biomaterials and Tissue Engineering
Universitat Politècnica de València
Valencia, Spain

Dr. Santiago Canals Gamoneda

Instituto de Neurociencias
Consejo Superior de Investigaciones Científicas
Universidad Miguel Hernández
San Juan de Alicante, Spain

November 2021

Supervisor	Prof. Dr. David Moratal Pérez Universitat Politècnica de València
Co-Supervisor	Dr. Santiago Canals Gamoneda Instituto de Neurociencias de Sant Joan - CSIC, Universidad Miguel Hernández
Members of the Jury	Vicent Manuel Teruel Martí Universitat de València Francisco Javier García Casado Universitat Politècnica de València Gonçalo Cardoso Lopes NeuroGEARS Ltd.
Members of the reviewers committee	Javier Primitivo Collado Ruiz Instituto de Física Corpuscular - CSIC Universitat de València Oscar Benito Herreras Espinosa Instituto Cajal - Consejo Superior de Investigaciones Científicas Vicent Manuel Teruel Martí Universitat de València

The research described in this thesis was carried out at the Polytechnic University of Valencia (Universitat Politècnica de València), Valencia, Spain in an extremely close collaboration with the Neuroscience Institute - Spanish National Research Council - Miguel Hernández University (Instituto de Neurociencias - Consejo Superior de Investigaciones Científicas - Universidad Miguel Hernández), San Juan de Alicante, Spain. The projects described in chapters 3 and 4 were developed in collaboration with, and funded by, Open Ephys, Cambridge, MA, USA and OEPS - Eléctronica e produção, unipessoal lda, Algés, Portugal.

Acknowledgements

Quisiera comenzar agradeciendo David Moratal y Santiago Canals por confiar en mi y abrirme todas las puertas que me han llevado hasta aquí. Muchas grandes historias comienzan con un "¿y si...?" y el suyo, hace ya más de diez años, acabó llevándome al camino que ahora recorro. Este trabajo no existiría sin ellos.

I would like to sincerely thank the Open Ephys/OEPS team for all the support over the years. Josh and Jakob for starting the project and welcoming me on it. Filipe and Lía, for founding OEPS and their help through bureaucracy. Jon, for being a fellow tech soul and our regular discussions. And Alex, for her help with wiring, her way with people and her weekly rankings.

Me gustaría también agradecer a todas la gente que me ha acompañado todos estos años. A mis amigos LEDC y asociados, Carlotes, Alejandro, Andrés, Paw, Alex, Estefi, Dani, Mayte, Marco, Natalia, Álvar, Roca, Álvaro, Marc, Víctor, Juan, Juanito, Iván y Patri. A aquellos con los que me he inmerso en mundos fantásticos, Jorge, Dani, Dimitri, Iván, Janha, Víctor, Bruji, Amaia... A la escuela Ruz Ryu por todo lo que me ha enseñado y ayudado a mejorar a muchos niveles así como a Toni, su fundador. Y a Javi, incluido en todos los grupos anteriores. Amigo, compañero de historias, aventuras y tecnologías y sensei.

También a todos los que están fuera y no veo tanto como me gustaría. Lucía, Lorena, Pau, Maripaz, Manu, Nieves, Nacho, Jorge, Xermo y Miguel Ángel. A mis amigos de la Kei Party, Iwa, Keikun, Xanko, Vero y Angelique. A

Darío, por años de compañerismo ¡Te debo un ramen! A Régel, con quien he compartido penurias telequiles y alegrías prácticamente desde que empecé la carrera. A Kelly, que me ha enseñado a ser fuerte aún en la cara de la adversidad. Y a Kitsu, que me ha aguantado durante tantísimos años.

Por último, pero no menos importante, a mi familia, que ha estado siempre a mi lado. A mis padres, a los que se lo debo todo. A mis hermanos, Aitor, Tamara y Aida, que bien sabe lo que es la carrera académica. Y en especial a mi sobrinita Atenea.

Abstract

Extracellular electrophysiology is a technique widely used in neuroscience research. It can offer insights on how the brain works by measuring the electrical fields generated by neural activity. This is done through electrodes implanted in the brain and connected to amplification and digitization electronic circuitry. Of the many animal models used in electrophysiology experimentation, rodents such as rats and mice are among the most popular species thanks to their small size, breeding speed and strong social and exploratory behaviors.

Modern electrophysiology experiments seek increasingly complex conditions that are limited by acquisition hardware technology. Two particular aspects are of special interest: Closed-loop feedback and naturalistic behavior. In this thesis, we present developments aiming to improve on different facets of these two problems.

Closed-loop feedback encompasses all techniques in which stimuli is produced in response of an event generated by the animal. Latency, the time between trigger event and stimuli generation, must adjust to the biological timescale being studied. While modern acquisition systems feature latencies in the order of $10ms$, response to fast events such as high-frequency electrical transients created by neuronal activity require latencies under $1ms$. In addition, algorithms for triggering or generating closed-loop stimuli can be complex, integrating multiple inputs in real-time. Integration of algorithm development into acquisition tools becomes an important part of experiment design.

For electrophysiology experiments featuring naturalistic behavior, animals must be able to move freely in ecologically meaningful environments, mimicking natural conditions. Experiments featuring elements such as large arenas, environmental objects or the presence of another animals are, however, hindered by the wired nature of acquisition systems. Other physical constraints, such as implant weight or power restrictions can also affect experiment time, limiting their duration. Beyond the technical limits, complex experiments are enriched when electrophysiology data is integrated with multiple sources, for example animal tracking or brain microscopy. Tools allowing mixing data independently of the source open new experimental possibilities.

The technological advances presented on this thesis addresses these topics. We have designed devices with closed-loop latencies under $200\mu s$ while featuring high-bandwidth interfaces. These allow the simultaneous acquisition of hundreds of electrophysiological channels combined with other heterogeneous data sources, such as video or tracking. The control software for these devices was designed with flexibility in mind, allowing easy implementation of closed-loop algorithms. Open interface standards were created to encourage the development of interoperable tools for experimental data integration.

To solve wiring issues in behavioral experiments, we followed two different approaches. One was the design of light headstages, weighing less than 2 grams, coupled with ultra-thin coaxial cables and active commutator technology, making use of animal tracking. This allowed to reduce animal strain to a minimum allowing large arenas and prolonged experiments with advanced headstages featuring high channel counts and extra features.

A different, wireless headstage was also developed. We created a digital compression algorithm specialized for neural electrophysiological signals able to reduce data bandwidth to less than 65.5% its original size without introducing distortions. Bandwidth has a large effect on power requirements. Thus, this reduction allows for lighter batteries and extended operational time. The algorithm is designed to be able to be implemented in a wide variety of devices, requiring low hardware resources and adding negligible power requirements to a system.

Combined together, the developments we present open new possibilities for neuroscience experiments combining electrophysiology acquisition with natural behaviors and complex, real-time, stimuli.

Resumen

La electrofisiología extracelular es una técnica ampliamente usada en investigación neurocientífica, la cual permite estudiar el funcionamiento del cerebro mediante la medición de campos eléctricos generados por la actividad neuronal. Esto se realiza a través de electrodos implantados en el cerebro y conectados a dispositivos electrónicos para amplificación y digitalización de las señales. De los muchos modelos animales usados en experimentación electrofisiológica, las ratas y los ratones se encuentran entre las especies más comúnmente utilizadas, gracias a su pequeño tamaño, velocidad reproductiva y sus fuertes comportamientos sociales y exploratorios.

Actualmente, la experimentación electrofisiológica busca condiciones cada vez más complejas, limitadas por la tecnología de los dispositivos de adquisición. Dos aspectos son de particular interés: Realimentación de lazo cerrado y comportamiento en condiciones naturales. En esta tesis se presentan desarrollos con el objetivo de mejorar diferentes facetas de estos dos problemas.

La realimentación en lazo cerrado se refiere a todas las técnicas en las que los estímulos son producidos en respuesta a un evento generado por el animal. La latencia, el tiempo transcurrido entre el evento desencadenante y la estimulación, debe ajustarse a las escalas temporales bajo estudio. Los sistemas modernos de adquisición presentan latencias en el orden de los $10ms$. Sin embargo, para responder a eventos rápidos, como pueden ser los transitorios de alta frecuencia creados por la actividad neuronal, se requieren latencias por debajo de $1ms$. Además, los algoritmos para detectar los eventos desencadenantes o generar los estímulos pueden ser complejos, integrando varias entradas

de datos en tiempo real. Integrar el desarrollo de dichos algoritmos en las herramientas de adquisición forma parte del diseño de los experimentos.

Para que experimentos electrofisiológicos incluyan comportamientos naturales, los animales deben ser capaces de moverse libremente en entornos ecológicamente significativos, emulando condiciones naturales. Experimentos de este tipo, que incluyen elementos como espacios amplios, objetos en el entorno o la presencia de otros animales, se ven dificultados por la naturaleza cableada de los sistemas de adquisición. Otras restricciones físicas, como el peso de los implantes o limitaciones en el consumo de energía, pueden también afectar a la duración de los experimentos, limitándola. Más allá de los límites tecnológicos, la experimentación puede verse enriquecida cuando los datos electrofisiológicos se ven complementados con datos procedentes de múltiples fuentes distintas. Por ejemplo, seguimiento de los animales o microscopía. Herramientas capaces de integrar datos independientemente de su origen abren la puerta a nuevas posibilidades experimentales.

Los avances tecnológicos presentados en esta tesis abordan estas limitaciones. Se han diseñado dispositivos con latencias de lazo cerrado inferiores a $200\mu s$. Estos presentan además interfaces de elevado ancho de banda, lo que permite la adquisición de cientos de canales electrofisiológicos combinados con otras fuentes de datos de naturaleza heterogénea, como vídeo o seguimiento. El *software* de control para estos dispositivos se ha diseñado manteniendo la flexibilidad como objetivo, permitiendo una fácil implementación de algoritmos de lazo cerrado. Se han desarrollado interfaces y estándares de naturaleza abierta para incentivar el desarrollo de herramientas compatibles entre ellas, para facilitar la integración de de datos experimentales.

Para resolver los problemas de cableado en experimentos conductuales se siguieron dos métodos distintos. Uno fue el desarrollo de *headstages* ligeros, con pesos inferiores a los 2 gramos, combinados con cables coaxiales ultra finos y conmutadores activos, posibles gracias al seguimiento de animales. Este desarrollo permite reducir el esfuerzo impuesto a los animales al mínimo, permitiendo espacios amplios y experimentos de larga duración, al tiempo que permite el uso de *headstages* con elevado número de canales y características avanzadas.

Paralelamente se desarrolló un tipo diferente de *headstage*, con tecnología inalámbrica. Se creó un algoritmo de compresión digital especializado para señales electrofisiológicas neuronales capaz de reducir el ancho de banda a menos del 65% de su tamaño original, sin introducir distorsiones. Dado que el ancho de banda juega un papel fundamental en los requisitos energéticos, esta

reducción permite baterías más ligeras y mayores tiempos de operación. El algoritmo fue diseñado para ser capaz de ser implementado en una gran variedad de dispositivos, requiriendo pocos recursos de *hardware* y una cantidad nimia de energía.

Combinados, los desarrollos presentados en esta tesis abren la puerta a nuevas posibilidades experimentales para la neurociencia, combinando adquisición electrofisiológica con estudios conductuales en condiciones naturales y estímulos complejos en tiempo real.

Resum

L'electrofisiologia extracel·lular és una tècnica àmpliament utilitzada en la investigació neurocientífica, aquesta tècnica permet estudiar el funcionament del cervell mitjançant el mesurament de camps elèctrics generats per l'activitat neuronal. Això es realitza a través d'elèctrodes implantats al cervell, connectats a dispositius electrònics per a l'amplificació i digitalització dels senyals. Dels molts models animals utilitzats en experimentació electrofisiològica, les rates i els ratolins es troben entre les espècies més utilitzades, gràcies a la seu reduïda grandària, velocitat reproductiva i forts comportaments socials i exploratoris.

Actualment, l'experimentació electrofisiològica busca condicions cada vegada més complexes, limitades per la tecnologia dels dispositius d'adquisició. Dos aspectes són d'especial interès: La realimentació de sistemes de llaç tancat i el comportament en condicions naturals. En aquesta tesi es presenten desenvolupaments amb l'objectiu de millorar diferents aspectes d'aquests dos problemes.

La realimentació de sistemes de llaç tancat es refereix a totes aquestes tècniques on els estímuls es produeixen en resposta a un esdeveniment general per l'animal. La latència, el temps transcorregut entre l'esdeveniment desencadenant i l'estimulació, ha d'ajustar-se a les escales temporals sota estudi. Els sistemes moderns d'adquisició presenten latències en l'ordre dels $10ms$. No obstant això, per a respondre a esdeveniments ràpids, com poden ser els transitoris d'alta freqüència creats per l'activitat neuronal, es requereixen latències per davall de $1ms$. A més a més, els algoritmes per a detectar els esdeveni-

ments desencadenants o generar els estímuls poden ser complexos, integrant varies entrades de dades a temps real. Integrar el desenvolupament d'aquests algorismes en les eines d'adquisició forma part del disseny dels experiments.

Perquè els experiments electrofisiològics incloguen comportaments naturals, els animals han de ser capaços de moure's lliurement en ambients ecològicament significatius, emulant condicions naturals. Experiments d'aquest tipus, que inclouen elements com grans espais, objectes en l'entorn o la presència d'altres animals, es veuen limitats per la natura cablejada dels sistemes d'adquisició. Altres restriccions físiques, com el pes dels implants o les limitacions al consum d'energia, poden també afectar a la duració dels experiments, limitant-los. Més enllà dels límits tecnològics, l'experimentació es pot enriquir quan les dades electrofisiològiques es complementen amb dades de múltiples fonts. Per exemple, el seguiment d'animals o microscòpia. Eines capaces d'integrar dades independentment del seu origen obrin la porta a noves possibilitats experimentals.

Els avanços tecnològics presentats a aquesta tesi tracten aquestes limitacions. S'han dissenyat dispositius amb latències de llaç tancat inferiors a $200\mu s$. Els quals presenten també interfícies d'elevada amplada de banda, la qual cosa permet l'adquisició de centenars de canals electrofisiològics combinats amb altres fonts de dades de naturalesa heterogènia, com vídeo o seguiment. El *software* de control per a aquests dispositius s'ha dissenyat mantenint la flexibilitat com a objectiu, permetent una fàcil implementació d'algorismes de llaç tancat. S'han desenvolupat interfícies i estàndards de naturalesa oberta per a incentivar el desenvolupament d'eines compatibles entre elles, per a facilitar la integració de dades experimentals.

Per a resoldre els problemes de cablejat a experiments conductuals es van seguir dos mètodes diferents. Un va ser el desenvolupament de *headstages* lleugers, amb pesos inferiors als 2 grams, combinats amb cables coaxials ultra fins i commutadors actius, possibles gràcies al seguiment d'animals. Aquest desenvolupament permet reduir al mínim l'esforç imposat als animals, permetent espais amplis i experiments de llarga durada, al mateix temps que permet l'ús de *headstages* amb elevat nombre de canals i característiques avançades.

Paral·lelament es va desenvolupar un tipus diferent de *headstage*, amb tecnologia sense fil. Es va crear un algorisme de compressió digital especialitzat per a senyals electrofisiològiques neuronals capaç de reduir l'amplada de banda a menys del 65% de la seua grandària original, sense introduir distorsions. Atès que l'amplada de banda juga un paper fonamental en els requisits energètics, aquesta reducció permet bateries més lleugeres i majors temps d'operació.

L'algorisme va ser dissenyat per a ser capaç de ser implementat a una gran varietat de dispositius, requerint pocs recursos de *hardware* i una quantitat nímia d'energia.

Combinats, els desenvolupaments presentats en aquesta tesi obrin la porta a noves possibilitats experimentals per a la neurociència, combinant l'adquisició electrofisiològica amb estudis conductuals en condicions naturals i estímuls complexos en temps real.

Contents

Acknowledgements	iii
Abstract	v
Resumen	vii
Resum	xi
Contents	xv
List of Figures	xix
List of Tables	xxi
Abbreviations and Acronyms	xxiii
1 Introduction	1
1.1 Extracellular electrophysiology	2
1.2 Extracellular electrophysiology recording systems	5
1.3 Rodents in electrophysiology research	8
1.4 Closed-loop feedback experiments	9
1.5 Field-Programmable Gate Array (FPGA) devices	10
2 Motivation and Objectives	13
2.1 Motivation	13
2.2 Objectives	15

2.3	Thesis outline	16
3	Open Ephys: Open source, closed-loop electrophysiology	19
3.1	Introduction	20
3.2	Materials and Methods	22
3.2.1	Intan RHD2000 integrated circuits	22
3.2.2	Xilinx FPGA	24
3.2.3	Connectors	24
3.2.4	JUCE Library	25
3.3	Results	26
3.3.1	Headstages	26
3.3.2	Open Ephys acquisition board	28
3.3.3	Open Ephys Software	31
3.3.4	Performance	35
3.4	Discussion	35
3.5	Conclusions	36
4	Open Neuro Interface: High performance acquisition	39
4.1	Introduction	40
4.1.1	High-bandwidth heterogeneous systems	40
4.1.2	Tether issues	41
4.1.3	Latency in closed-loop experiments	42
4.1.4	Overview	42
4.2	Materials and Methods	43
4.2.1	Bus standards	43
4.2.2	FPD-Link III devices	44
4.2.3	FPGA devices	44
4.2.4	3D Tracking	46
4.2.5	Acquisition devices	50
4.2.6	Stimulation devices	51
4.2.7	Software	51
4.3	Results and Discussion	52
4.3.1	ONI specification	52
4.3.2	ONIX hardware	57
4.3.3	Tethers and torque-free commutator	62
4.3.4	ONIX firmware	65
4.3.5	Acquisition performance	71
4.3.6	3D Tracking	72
4.4	Conclusions	73
5	Wireless electrophysiology compression	75

5.1	Introduction	76
5.1.1	Wireless electrophysiology devices	76
5.1.2	Data compression methods	78
5.1.3	Objectives	80
5.2	Materials	80
5.2.1	Huffman Coding	80
5.2.2	Delta compression	83
5.2.3	Low-power FPGA	83
5.2.4	Wireless processor	84
5.2.5	Sample signals and acquisition hardware	85
5.2.6	Development hardware and software	86
5.3	Methods	88
5.3.1	Software model	88
5.3.2	Hardware design and validation	89
5.3.3	In Vivo testing	90
5.4	Results	90
5.4.1	Compression algorithm	90
5.4.2	Low-memory, Low-resource compression	91
5.4.3	Compression performance	94
5.4.4	Effect of dictionary on compression	96
5.4.5	Transmission protocol	96
5.4.6	Wireless prototype	98
5.4.7	Power usage	104
5.4.8	Resource usage	105
5.5	Discussion	105
5.6	Conclusions	107
6	Conclusions and Outlook	109
6.1	Implications for neuroscience research	111
6.1.1	Effect of tools in the experiments	111
6.1.2	Closed-loop and brain timescales	112
6.1.3	Multi-source acquisition	114
6.1.4	Modular approach	115
6.2	Implications for the academic community	115
6.3	Future steps	117
7	Contributions	119
7.1	Collaborations in the scope of the Thesis	119
7.2	Publications	120
7.3	Teaching	120
7.4	Conference posters	121

Bibliography

123

List of Figures

1.1	Neurons visible through Golgi-Cox method	2
1.2	Action potential	4
1.3	Measured potential	5
1.4	Diagram of an extracellular electrophysiology system	6
2.1	Torque, weight and animal mobility	14
2.2	Common failure events that limit experiment duration.	14
3.1	Sharing closed loop algorithms	21
3.2	32-channel Intan headstage	24
3.3	Omnetics connectors	25
3.4	Open Ephys headstages	27
3.5	Torque comparison between headstages	27
3.6	Open Ephys acquisition board and its components	28
3.7	Open Ephys Input/Output (I/O) board	29
3.8	Rhythm firmware diagram	31
3.9	Open Ephys GUI	32
3.10	Class structure of the Open Ephys software. From [96].	33
3.11	Example code for an Open Ephys GUI processor	34
4.1	Rotary commutator internals	41
4.2	Numato board overview	45
4.3	SteamVR overview	47
4.4	Lighthouse plane sweeps	47
4.5	Differential manchester encoding	49

4.6	Example 8-bit LFSR	49
4.7	ONI specification diagram	53
4.8	OSI-Style relation of layers in an ONI system	56
4.9	Overview of all the components developed for the ONIX system	57
4.10	ONIX host board	58
4.11	ONIX 64-channel headstage	59
4.12	ONIX headstages PCBs	60
4.13	ONIX Neuropixels headstage	61
4.14	PCB layout of the breakout board	61
4.15	ONIX and Open Ephys headstage comparison	62
4.16	Commutator actively following headstage orientation.	63
4.17	Weight-torque diagram on ONIX system	63
4.18	Difference in exploratory behavior in mice using standard and ONIX headstages	64
4.19	ONIX firmware diagram	65
4.20	ONIX internal bus chronograms	68
4.21	Communication protocol between serializer and deserializer.	70
4.22	Bus over I^2C chronograms	71
4.23	ONIX system bandwidth	72
4.24	3D experimentation environment	73
4.25	Environment occupancy heat map	73
5.1	Huffman dictionary creation	82
5.2	Huffman dictionary size	83
5.3	Hardware development boards	87
5.4	Generated synthetic signal for testing	90
5.5	Comparison between raw and delta-coded signals	91
5.6	Symbol appearance distribution in delta-coded signals	93
5.7	Degradation of compression efficiency with uncompressed bits	93
5.8	Algorithm diagram	94
5.9	Effect of compression on signal integrity	96
5.10	Transmission protocol structure	98
5.11	Hardware prototype	99
5.12	SPI communication scheme between the FPGA and the wireless processor.	100
5.13	Block diagram of the wireless acquisition prototype FPGA firmware	101
5.14	Wireless processor software block diagram.	103
5.15	Prototype power usage	104

List of Tables

3.1	Characteristics of Intan RHD2000 integrated circuits	23
4.1	Year of introduction and speed of the different current PCI-Express (PCIe) versions	43
4.2	Lighthouse v1 activation Timings	48
4.3	Required registers in the ONI specification	54
5.1	Symbol appearance frequency in delta-coded signals	92
5.2	Compression ratios	95
5.3	Effect of dictionary on compression	97
5.4	FPGA logic usage	105

Abbreviations and Acronyms

ADC Analog-Digital converter. 7, 8, 10, 22, 29, 30, 52, 53, 58, 69, 76, 77, 92

AIS axon initial segment. 3

API Application Programming Interface. 24, 30, 34, 40, 42, 56, 57, 85, 102

ARM A computer processor architecture developed by Arm Ltd. (Cambridge, UK). 85

ASIC Application-Specific Integrated Circuit. 11, 78, 105, 106

C++ A low-level, object-oriented programming language. 25, 34

CMRR Common-Mode Rejection Ratio. 7

CPU Central Processing Unit. 42, 43, 100, 113

DAC Digital-Analog converter. 29, 51, 58, 69

DCT Discrete Cosine Transform. 78

DDR Double Data Rate. 22–24, 45

DMA Direct Memory Access. 46, 85, 102, 103

DSP Digital Signal Processor. 11, 24, 45, 46, 84, 91, 105, 117

- EIB** Electrode Interface Board. 26, 27, 50
- FFT** Fast Fourier Transform. 78
- FIFO** First In, First Out. A type of memory queue in which data can be read in the same order it was written, removing it from the queue once read. 30, 66–68, 70, 101
- FMC** FMC Mezzanine Card. 43, 45, 58
- FPGA** Field-Programmable Gate Array. xv, 10–12, 24, 28–30, 43–46, 51, 58–60, 65, 83, 84, 86–90, 99–105, 117, 119
- FSM** Finite State Machine. 29, 30
- GPIO** General Purpose Input/Output. 44, 46, 68, 85, 86
- GPU** Graphical Processing Unit. 10, 11, 42, 113
- GUI** Graphical User Interface. 19, 26, 31–33, 37, 88, 115, 116
- HDL** Hardware Description Language. 12, 24, 51, 87–90
- I²C** Inter-Integrated Circuit. 44, 50, 67, 68, 70, 71, 85
- I/O** Input/Output. xix, 21, 28, 29, 36, 43, 58, 60, 61, 69, 70, 84, 110, 112, 114
- IC** Integrated Circuit. 7, 11, 12, 22, 23, 40, 51, 84–86, 89, 98, 99
- IDE** Integrated Development Environment. 12, 52, 87
- IMU** Inertial Measurement Unit. 40, 46, 50, 59, 60
- JTAG** Joint Test Action Group. A standard for debugging interfaces. 12, 51
- LFP** Local Field Potentials. 4, 5, 15, 32, 79, 113
- LFSR** Linear Feedback Shift Register. 49
- LVDS** Low-Voltage Differential Signaling. 7, 22, 35, 36, 41

- MCU** MicroController Unit. 10, 85, 99–102
- MEMS** Micro-Electro-Mechanical System. 50
- MISO** Master-In Slave-Out. 100, 101
- MOSI** Master-Out Slave-In. 100
- ONI** Open Neuro Interface. xvi, xx, xxi, 16, 39, 42, 52–59, 61, 66, 68–70, 73, 110, 114–117, 119
- OOP** Object-Oriented Programming. 25, 34
- PCA** Principal Component Analysis. 5
- PCB** Printed circuit board. 24–26, 28, 43–45, 51, 58, 59, 84, 85, 88
- PCIe** PCI-Express. An high-speed evolution of the PCI (Peripheral Component Interconnect) bus used to connect expansion boards in computer systems. xxi, 43, 45, 46, 51, 56–58, 70, 74, 117
- RAM** Random Access Memory. 11, 24, 30, 45, 46, 84, 85
- ROM** Read-Only Memory. 91
- RTOS** Real-Time Operating System. 102
- SoC** System-on-Chip. 85
- SPI** Serial Peripheral Interface. 22, 28, 30, 36, 41, 45, 85, 99–103
- SSD** Solid-State Drive. 35
- TTL** Transistor to Transistor Logic. A standard for digital circuits. When not using to describe a device it refers to the signaling these devices use: logic communication with simple on-off signaling and 5V levels. 29, 30, 33
- USB** Universal Serial Bus. 8, 24, 30, 35, 36, 56, 57, 113, 117
- VR** Virtual Reality. 9, 40, 47, 74, 112

Chapter 1

Introduction

The brain, its functions and how its physiological processes relate to behavior and cognition has been a research topic since ancient times, with multiple techniques and tools created devoted to its study. Even so, no complete theory on how the brain works has yet been formulated [1].

Modern neuroscience understanding can be traced to the late nineteenth century. The Golgi stain method [2], [3] allowed individual neurons to be visualized (Figure 1.1). Santiago Ramón y Cajal created the neuron doctrine [4], which describes the nervous system as a collection of individual neurons. The notion of different areas of the brain being responsible for specific functions was pioneered by Jean Pierre Flourens and Paul Broca, with Korbinian Brodmann publishing the first map of cerebral regions [5]. While some of these hypotheses have been updated, for example with compound neural activity now being considered important, opposed to only individual activation[6], most of these principles hold and have been verified by modern techniques, such as electrophysiology or neuroimaging [7].

The twentieth century gave rise to an increased understanding of the neuron molecular structure and how they communicate. Edgar Adrian was among the first to notice signals transmitted through individual nerves, related to specific stimuli. He measured a series of electric potential changes, called action potentials or *spikes*, and surmised the existence of a common neural

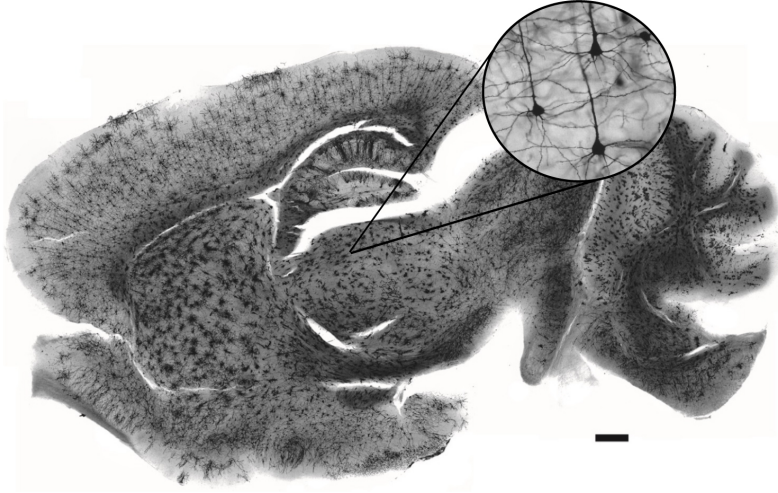


Figure 1.1: Neurons easily visible through the Golgi-Cox staining method. Modified from Zaquot et Al. [3].

communication method [8]. A mathematical model for these action potentials was later presented by Alan Loyd Hodgkin and Andrew Huxley by studying their initiation and propagation in a squid axon, creating the Hodgkin-Huxley model [9].

Research on spike and cell activity in living animals has led to discoveries about brain function [10]–[12] and remains an important tool for neuroscience research.

1.1 Extracellular electrophysiology

Electrophysiology refers to the set of techniques able to measure the electrical properties generated by cell activity. Extracellular electrophysiology in particular measures the electric potential created in the extracellular space by transmembrane ionic flows related to neuron activation.

Neurons in rest state are held in an electrochemical equilibrium governed by the ionic concentrations in the intracellular and extracellular fluids. The resting electric potential across the membrane of a neuron, referred to its exterior, is approximately $-70mV$ [13], meaning that the ionic concentration is more negatively charged in the intracellular side of the membrane than in the extra-

cellular space. This ionic concentration is maintained by a molecular structure present in the membrane called Sodium-Potassium pump which continuously expends energy to ensure that ionic concentrations are not equalized between the intracellular and extracellular mediums [14], creating a chemical gradient in addition to the electrical potential.

Neurons can communicate with each other through the release of chemical messengers called neurotransmitters. These bind to selective receptors on the synapses of the receiving cell, which can open ion channels, a molecular structure in the cellular membrane. These structures allow ions, often Na^+ or Ca^{2+} to flow from the extracellular to the intracellular space, where their concentration is higher. The change on charge distribution provoked by this ion influx causes a local change, or depolarization, of the membrane electric potential. This change in the electrical field propagates through the membrane and can trigger the opening of a different structure, voltage-gated Na^+ channels, present through all the cell membrane. These structures open when exposed to a voltage over a certain threshold, reinforcing the propagation. Na^+ channel distribution is not homogeneous, its density being significantly higher in the axon initial segment (AIS) than in the cell body or dendrites [15].

Potentials initiated in the synapses are attenuated along the way to the AIS [16]. However, multiple simultaneous stimuli can overlap, resulting in a summed potential enough to trigger voltage-sensitive Na^+ channels at the AIS [17]. Due to the higher channel density, this process initiates a fast cascade effect which propagates through the axon, creating a fast influx of Na^+ ions. This flow continues until the membrane potential crosses a positive threshold triggering the opening of a different structure, voltage-dependent K^+ channels open. These allow a flow of K^+ ions from the intracellular medium, where their concentration is higher, to the extracellular space. This results in an electric current in the opposite direction, repolarizing the cell. Repolarization overshoots the original resting potential, creating a period of hyperpolarization. The continuous effect of the Sodium-Potassium pump eventually returns the cell to its resting state, when the process can be triggered again. While this process is considered the main effect of cell activation, other molecular events contribute to the electrical currents flowing through the cellular membrane, such as Ca^{2+} spikes [18].

While the individual transmembrane currents can be measured [19], their compound effect can be observed in the extracellular space as variations on the electric field potential. The effect is described in Equation 1.1 [20]–[22]:

$$V_n(r_m, t) = \frac{1}{4\pi\rho} \sum_{n=1}^N \frac{I_n(t)}{|r_m - r_n|} \quad (1.1)$$

In which V_n is the electric potential, with respect to a point in the infinite according to Coulomb's law, ρ the extracellular medium conductivity value, r_m the position in which the measurement is taking place and I_n and r_n the different individual currents. The resulting waveforms from an action potential have the topology seen in Figure 1.2, where the initial slope is created by the Na^+ ion influx, then followed by the reverse polarity of the outwards K^+ ion flow before finally settling back into the resting potential.

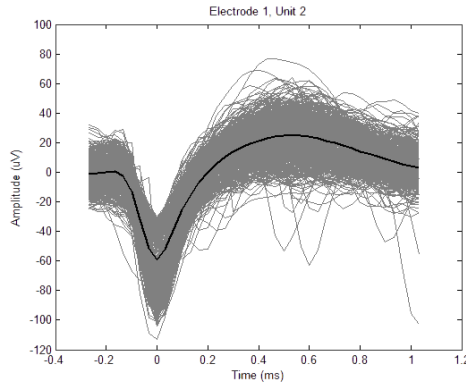


Figure 1.2: Extracellular action potentials from the visual cortex of a rat. The darker line represents the mean waveform.

It is important to note that at any given time, hundreds of neurons are spiking simultaneously. Thus, for any given sensing site, the measured potential is theoretically the sum of the activity of every single cell, resulting in a signal similar to the one seen in Figure 1.3. Due to the inverse distance factor, the conductivity value ρ being dependent on frequency and the interference of different neurons firing off-phase, only the spikes of cells located physically close to the recording site can be extracted. Contrariwise, synchronized activity of multiple neurons in the area surrounding the site result in low-frequency, high-amplitude oscillations, referred to as Local Field Potentials (LFP) [23]. While the former gives information about individual cells, LFPs mostly represent the aggregated synaptic activity and active dendritic currents in the tissue. [23]–[26].

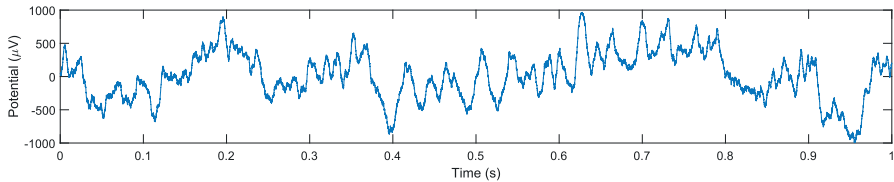


Figure 1.3: Combined measured potential in an extracellular site. Recorded from the hippocampal region on a rat.

Since a single site detects action potentials from different neurons, analyzing spike activity requires multiple steps. First, the signals are high-pass filtered to remove the compound LFP elements, leaving only high frequency spike waveforms. An amplitude threshold method can be used to locate individual events by detecting the large depolarization peaks [27]. While this detects the spikes, extra steps are needed to differentiate events originating in distinct neurons, a process called spike sorting. While there are multiple algorithms to separate spikes into clusters of common neuron origin [28], many of them take advantage of measured waveforms from different neurons being non-identical due by both biological and geometrical differences. Most such algorithms use windowing methods to detect specific waveform features that cross with time-amplitude areas [29] or perform mathematical component extraction such as Principal Component Analysis (PCA) [30] to separate events in a projected space.

1.2 Extracellular electrophysiology recording systems

Since no electronic circuit can be physically tied to infinity, measuring an electric potential implies measuring the potential difference between two points. In the case of electrophysiological signals, a recording probe is inserted next to the cells of interest with the potential V_n created by their transmembrane currents, and a reference electrode is placed far from the region interest, located at a potential V_r . The actual measured voltage V_m is thus the difference $V_m = V_p - V_r$, assuming there is no current flow through the electrodes which would provoke voltage drops resulting in measurement distortions.

The challenge of recording extracellular potentials reside on the small scale required, both physical and electrical. As previously discussed, the signals are in the range of μV , which require amplification to be properly recorded and analyzed. Moreover, to reduce tissue damage, probes inserted into the brain must have a thickness measured in μm [31]. Figure 1.4.B shows the equivalent

circuit of a probe, the input of an amplifier and the wire connecting both. To avoid a potential drop in the order of the interest signal, amplifier input impedance must be as high as possible, while wire and probe impedances must be kept low.

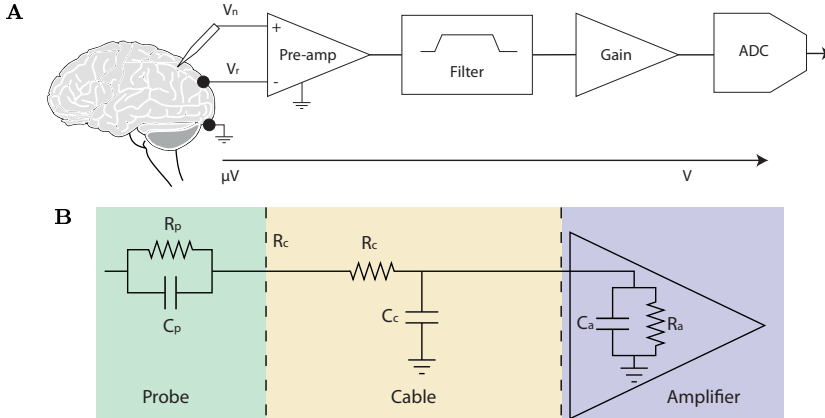


Figure 1.4: **A:** Diagram of an extracellular electrophysiology circuit. **B:** Equivalent circuit of the probe-wire-amplifier path

The first neural potentials were recorded from individual neurons using a single ultra-thin wire, or microwire [32]. Over time, as amplifier technology was refined, channel count increased in the form of microwire bundles [33]. Progress in semiconductor manufacturing processes led to silicon probes. These feature long shafts with multiple electrode sites [34]–[36]. Both types of probes are in use nowadays, with microwires offering flexibility on precise electrode position while silicon probes offer higher channel count and the ability to reach deeper parts of the brain. Recently CMOS semiconductor technology has been used to develop a new family of probes, featuring even higher channel densities [37]–[41]. As previously stated, probe impedance has to be kept low, even with an increased electrode density. Thus, extensive research has been done to develop electrodes with lower impedance [42]–[45] and techniques, such as electroplating [46]–[48] exist to lower electrode impedance by coating them with a different material.

While the increasing electrode density allows recording from different neuron groups with a single probe, it offers other advantages. Since signal amplitude is proportional to the distance between the neuron that produced it and the electrode, by grouping electrodes in pairs, called stereotrodes [33] or groups of four referred as tetrodes [49], [50] physically close it is possible to use the small

differences in amplitude of otherwise identical spike waveforms to differentiate the individual neurons that produced each spike through simple triangulation methods.

Amplifier technology has also improved from the valve amplifiers used in the first electrophysiology recordings [8]. Modern bioamplifiers are differential amplifiers that apply a gain to only the difference between the probe potential and the reference electrode, thus removing any common noise that could be received by the wires or electrode, for example 50/60Hz noise from the electric installation. This ability to only amplify the differential signals, referred as Common-Mode Rejection Ratio (CMRR), has been accompanied by and increasingly high input impedance and noise reduction [51]. Thanks to the semiconductor industry, Integrated Circuits (ICs) packing multiple low-power, high-performance amplifiers are possible [52].

A modern extracellular acquisition system is described in Figure 1.4.A. In general, the μV potentials are preamplified by a high input impedance amplifier to a manageable range, then filtered to remove all unwanted frequencies and finally amplified to a tension range able to be captured by an Analog-Digital converter (ADC), which then sends the data to a recording device. As discussed and shown in Figure 1.4.B, wire impedance has to be kept as low as possible. However, this value is proportional to wire length, with longer wires providing larger impedances. Moreover, long wires can act as antennas, receiving noise from nearby electrical fields or even radiofrequency emissions. As such, it is important to keep wires carrying analog signals, specially the ones connected to the brain, sensing low-voltage signals.

To keep noise-sensitive wiring short, there has been a tendency to place electronics as close to the brain as possible. The hardware located at the head of the experimental subjects is what is commonly referred as the headstage. Originally, only the smaller preamplifier could be located there, with wires carrying analog signals routed to a larger amplifier and recording system. Technology and miniaturization advances have allowed to move more elements to the headstage [36], with modern devices featuring amplification, filtering and digitization for all channels on the same headstage [53]. Newer CMOS probes go one step further, integrating some circuitry into the probe itself, reducing impedance and noise of the probe-amplifier electrical interface even further [38], [40], [54]. Having the ADC in the headstage offers two big advantages: Since its outputs are digital signals, longer wires become a less prominent issue, thanks to the natural noise resilience of digital communication. This is specially true if a differential pair signaling method, such as Low-Voltage Differential Signaling (LVDS), is used. Also, since data from multiple channels

can be multiplexed into a high-speed digital link, communication can be performed over a dozen or less wires, instead of requiring one wire per channel, reducing tether complexity.

Digitizing neural data offers many advantages, allowing more effective storage and analysis. However, sampling frequency must be carefully selected. A neural spike has a duration of 1-2ms, its waveform containing high-frequency components. Due to the Nyquist theorem, sampling frequency must be double the desired signal bandwidth. In order to successfully record spikes without losing detail, modern recording systems acquire at a rate of 20KS/s or higher [27], with 30KS/s being the norm.

The growing interest in higher channel counts translates into an increasing need for data bandwidth. At 16 bit resolution, typical for modern electrophysiology ADCs, a 32 channel signal sampled at 30KS/s requires near 2 MB/s. With newer system potentially reaching thousands of channels, fast communication links, such as Universal Serial Bus (USB) 3.0, as well as fast storage systems are needed.

1.3 Rodents in electrophysiology research

While the first action potentials were recorded in squid axons, thanks to their big size facilitating direct membrane readings [55], advances in extracellular electrophysiology have allowed to observe neural activity in a wide variety of living animals.

Cats [10], [56] and monkeys [11] were among the first animals used for *in vivo* brain electrophysiology experiments. Nowadays, thanks to the miniaturization of electrodes, experimentation is possible with not only mammals but birds [57], [58], fish [59], [60] or insects [61].

Of all the animal models for research rodents, specifically rats and mice, remain among the most used for electrophysiology research. Both species feature short gestation cycles, which facilitates breeding the animals in laboratory conditions. Over the years and with the advent of genetic techniques, this has resulted on a number of genetically-controlled strains [62], [63].

Both species offers different advantages for research. For example, genetic techniques are more available for mice, allowing faster breeding of specific models. On the other hand, rat brains have some similarities with human brains that mice lack [64]. While the larger size of rats makes surgery and

long-term recordings easier, the smaller size of mice facilitates the handling of large colonies.

Given the strong social and exploratory behaviors of rats and mice, it is usual for experiments to explore this tendencies by featuring animals freely moving in enclosed spaces [65]–[67] or small maze structures [68]–[71], providing insights in topics such as location encoding or information retrieval. Wiring can be a challenge on these kind of experiments, as cables can become tangled or twisted due to animal movement. To alleviate these issues, experiments with freely moving animals often feature a rotary commutator between the headstage and the recording system. These devices separate cables in two sections able to rotate independently while maintaining electrical connection in the wires.

A different approach to this issue has been the development of Virtual Reality (VR) environments with head-fixed animals [72]. While this allows to simulate some conditions normally impossible with tethered systems, VR environments are not a complete substitute for real spaces. They lack vestibular inputs [73] and small delays in visual stimuli can led to experimental errors. Motion-wise, while 3D location and navigation [74] is an integral part of natural animal behavior, VR spaces are limited to 2D movements. In addition, VR environments are isolated by nature, making any kind of social interaction impossible. Thus, truly free movement and big, complex arenas are still highly desirable.

1.4 Closed-loop feedback experiments

Traditional electrophysiology experiments consist on presenting a series of stimuli or tasks to an animal and passively record brain activity in response. While this approach has yielded many valuable results, it does not take into account the inherent recursive nature of the brain. Not only neural networks can loop over themselves, but the brain of an animal is embedded in a feedback loop with the environment: Any action of the animal results in a change of sensory inputs that enter the brain and, in turn, affect posterior actions [75].

Closed-loop feedback is a term stemming in engineering referring to any system whose output is directly tied to the input, usually feeding back an error signal between the expected and real outputs so the system can stabilize to a desired state. In neuroscience research, this concept can be applied by stimulating the brain in response to its own activity, thus allowing a greater level of control over the neuronal network and the isolation of events to prove causality [76], [77]. This method has lead to insights on neural plasticity [78], neuronal response latency [79], spatial learning [80] or neural circuit adaptation [81].

Multiple stimulation methods can be used. Sensory feedback, by itself, translates into brain inputs, so visual [82] or auditory [83] stimulation can be modulated by brain readings to achieve closed-loop feedback. Cells near an implanted electrode can also be activated by electrical stimulation [79], [84], by controlling an externally supplied current flow. Optogenetic stimulation [62], [85], a technique in which neurons are genetically modified to react to specific light wavelengths, is gaining popularity nowadays. Neurons can be modified to express photosensitive receptors that either activate or inhibit membrane ion channels, thus allowing not only to externally trigger an action potential but to completely block the ability of a neuron to naturally fire one while illuminated by a specific wavelength. By the use of fine-tuned lasers, this control can be targeted to individual neurons, allowing for precise feedback to specific neural circuits.

One crucial parameter for closed-loop systems is feedback latency. Since the objective is to create a correlation between the output and the feedback input, if latency between the source event and stimulation were too high they would become decoupled, losing correlation and behaving as simply two independent events. Timescales depend on the underlying biological system that is being manipulated. For example action potentials have timescales under 1ms, stimulation-induced neural plasticity requires scales under 10ms [78], [86] while auditory stimuli can be effective with latencies in the hundreds of ms [83]. Closed-loop latency, the total delay between source event and stimulation, is a sum of the time required to perform acquisition, data transmission, digital processing and stimulus generation. Thus, faster systems imply reduced latency, which in turn allows for its use in experiments requiring a wider variety of timescales.

1.5 Field-Programmable Gate Array (FPGA) devices

With the rising interest in higher channel counts at high sampling speeds and lower closed-loop latencies [87], hardware requirements on acquisition system also increase. While modern personal computers have powerful processors and Graphical Processing Units (GPUs) able to process big volumes of data, an acquisition system must be able to provide such data by driving all the ADCs and sensors, packing the data and sending it over a transmission link to the computer, all with strict timing constraints.

Classical, sequential MicroController Units (MCUs) are limited in the number of operations per second they can run. Since they can only execute one

command at a time, performing multiple, synchronized actions can only be done by running the steps sequentially at high speeds. However, processor speed is limited by power and thermal considerations [88]. Parallel computing, performing multiple actions simultaneously, is the solution to this problem, as being able to perform acquisition for each sensor, packing and transmission in an independent, parallel but synchronized way alleviates the need to perform these steps at higher speeds. There are different devices able to perform such parallel operations. Multicore processors lack enough parallelism, as they can only perform one simultaneous action per core. GPUs can run hundreds of parallel operations, but while they are widely used for processing [89] they lack the hardware resources needed for driving sensors. Application-Specific Integrated Circuits (ASICs) are custom hardware devices created for a specific task, so they can achieve the desired parallelism with very high performance. However, they lack flexibility, once made any change requires a new fabrication batch. Field-Programmable Gate Arrays (FPGAs) offer the parallel capabilities of an ASIC, albeit with slightly reduced performance, but with the ability of being reconfigured at no cost.

FPGAs are ICs featuring configurable digital electronics. A FPGA, on its basic form, is comprised of a number of logical gates and memory elements whose inputs and outputs can be freely rearranged through an interconnection matrix. This makes it possible to configure a FPGA to act as any digital circuit, as long as there are sufficient logic elements available. Unlike programmable devices, such as microprocessors, which run a single program step by step, the circuits configured into a FPGA can run in parallel and independently. The specific organization of the logic elements is vendor-dependent, which makes it challenging to compare devices from different manufacturers.

The drawbacks of a FPGA over a specialized circuit or programmable device are size and speed. Due to the configurable nature of the logic, the internal components that comprise it use more silicon area in the IC than those of a fixed circuit. Due to this complexity and the limits of the interconnection matrix, clock speeds used to drive the logic is limited in comparison with specialized circuitry. To work around these limitations, many FPGA integrate what are called *hard blocks*: circuitry specialized on a single task that can be routed to the configurable logic of the device. Some common examples are Random Access Memory (RAM) blocks to allow for increased storage capacity or Digital Signal Processor (DSP) units, combinations of multipliers and accumulators. Newer devices include a microprocessor connected to the FPGA fabric to combine the advantages of both programmable and configurable devices. Commercial FPGAs range from big devices with thousands of elements

and plenty of integrated hard blocks to small, compact ICs with fewer resources and little to no hard blocks but designed for small size and reduced power consumption.

FPGA functionality is usually designed using a Hardware Description Language (HDL), the most common being VHDL and Verilog. These kind of languages allow for a functional description of the hardware, which is then compiled into the required logic circuitry by a synthesizer software and mapped into the FPGA hardware by a router program. Both synthesizer and router are usually part of an Integrated Development Environment (IDE) provided by the device manufacturer. The resulting product is a file, called *bitfile*, which contains the FPGA configuration. Most FPGAs require an external flash memory chip to store the bitfile. However, some devices designed for embedded application can feature internal flash storage for this purpose. Bitfiles are usually loaded by the device at power-on from a the flash memory, but can also be used to manually configure a powered FPGA through debug interfaces, such as JTAG.

Motivation and Objectives

2.1 Motivation

The contributions of electrophysiology experimentation in small spaces or mazes are undeniable, and can still provide new insights [90]. However, research into areas such as complex behavior or social interactions are hindered by the artificial nature of those environments. More natural experimental conditions, with bigger areas and physically unburdened animals are required to create meaningful environments [91]. The growing interest on combining electrophysiology records with behavior [92] requires technological improvements on acquisition systems. These advances must not be limited to improving specifications of existing hardware but to develop new architectures able to go beyond the limits that prevent current electrophysiology devices to be viable in truly natural and complex environments.

One of the main limitations on natural movement is the burden the electrophysiology equipment itself causes on the animals. Headstages add unnatural weight to their heads, while tethers provide tension when applying torque to turn a commutator, or extra weight and drag when the cable is hanging from the head itself. Figure 2.1 shows a graph of how the combined forces of tether torque and headstage weight can hinder movement in mice. Although rats,

due to their bigger size, can withstand these conditions better, it is still an added burden on the animals.

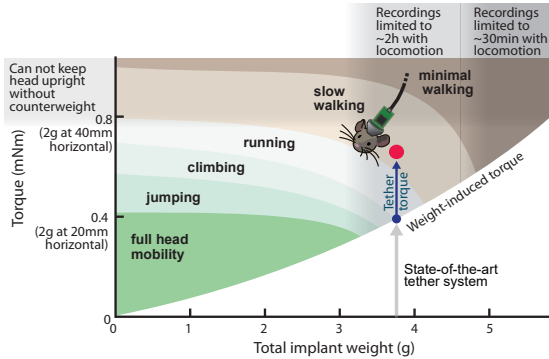


Figure 2.1: Mice mobility based on headstage weight and tether torque. Other animals follow similar distributions with altered scales.

Movement capacity is not the only parameter hindered by the hardware. Behavior experimentation can require up to several days [92]. However, cables can easily break or burden in the animals can accumulate until their behavior becomes unnatural. Figure 2.2 shows an example of typical study duration and common failure events on electrophysiology systems. Even wireless approaches [93] are limited by their battery capacity.

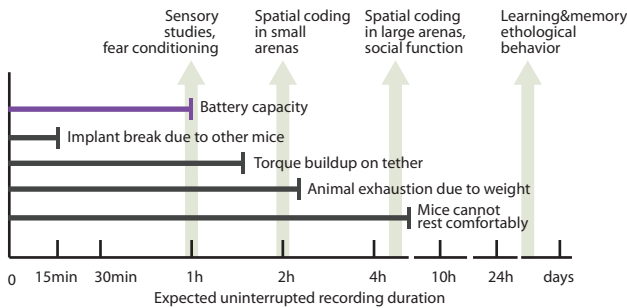


Figure 2.2: Common failure events that limit experiment duration.

Thus, to enable true behavior experiments combined with electrophysiology recordings to unveil knowledge on how brain activity relates to natural and social behavior, the issues concerning headstage weight and tether torque must

be minimized, if not removed, while operational time of the equipment must extend to days.

As discussed on section 1.4, closed-loop feedback systems are an important research tool, offering more experimental possibilities. This is still true in behavioral studies [94]. For this reason, an ideal electrophysiology system for ethology experiments should also have the ability to provide real time data for feedback based on neural signals.

2.2 Objectives

The objective of this thesis is the development of novel electrophysiology recording systems that could expand the range of possible experiments with freely moving rodents. Five specific objectives for system improvement are targeted:

- **Headstage reduction:** Headstage weight and size must be reduced while maintaining or improving acquisition capabilities
- **Tether impact reduction:** Issues caused by tethers must be minimized or even removed by improving wiring management, reducing tether weight, eliminating torque issues or eliminating their need through wireless alternatives.
- **Experiment time extension:** Running time of experiments must be improved by minimizing or removing limitations caused by the hardware.
- **Closed-loop improvement:** Closed-loop capabilities must be improved. This can include experiment design or technical improvements. The former can be achieved by standardizing interfaces with different equipment and increasing the researchers flexibility to define triggers and responses. Technical characteristics can be improved by minimizing latency to allow closed-loop capabilities for events in the sub-millisecond range, such as spikes.
- **Tool flexibility:** New tools must allow researchers to build different experiments suited to their needs, including complex arenas or specific closed-loop algorithms, avoiding hard limitations.

Any development can not compromise the ability to perform multichannel acquisition in the LFP and spike ranges. Optional improvement areas include increasing channel count or introducing other measurement devices useful for

behavior research, such as animal tracking, without compromising the fundamental objectives of this work.

2.3 Thesis outline

This thesis is presented in 6 chapters, following a development progression through different electrophysiology acquisition systems, each offering distinct improvements. While each chapter can be read independently, chapter order shows a natural evolution from a more classical approach to electrophysiology system to high-performance acquisition and finally wireless devices.

Chapter 1 presents an introduction to the neuroscience field and electrophysiology hardware.

Chapter 2 exposes the motivations behind this work and its objectives.

Chapter 3 describes the Open Ephys system, an open source electrophysiology acquisition hardware and software. This is the first development related to this work and the project that provided a starting point for the next developments. This system features a digital headstage and was designed with closed-loop flexibility in mind, allowing researchers to define their own triggers and offering communication capabilities with external hardware.

Chapter 4 is dedicated to the Open Neuro Interface (ONI) specification and its implementation, the ONIX acquisition system, a high-performance electrophysiology recording device with zero-torque headstages. The ONI specification defines a standard set of interfaces and protocols for high-speed transmission of data from multiple heterogeneous acquisition devices, able to mix sources such as neural data, positional tracking and imaging in a seamless way. ONIX is an implementation of this standard, an electrophysiology acquisition system with support for thousands of channels and a closed-loop latency below the millisecond mark. ONIX headstages communicate through single, ultra-thin coaxial cables, greatly minimizing wiring issues. Additionally, an active commutator was developed, which effectively makes it a zero-torque system.

In Chapter 5 wiring limitations are removed by developing a completely wireless system. In this chapter a device-agnostic, low-footprint and low-power compression algorithm is presented, able to greatly reduce the bandwidth needs of wireless electrophysiology data transmission with a negligible overhead. A hardware prototype using this algorithm was created, performing wireless elec-

trophysiology acquisition and demonstrating reduced power needs thanks to bandwidth reduction.

Finally, in Chapter 6 an overall conclusion and future prospects are provided.

Chapter 3

Open Ephys: Open source, closed-loop electrophysiology

Open Ephys is a community-based, open-source project for closed-loop electrophysiology. It was born from the need to share closed-loop algorithms and techniques, which are required to enable experiment replication. The project includes a multichannel electrophysiology acquisition hardware system and a Graphical User Interface (GUI) software. The hardware features state-of-the-art characteristics and digital headstages, making them noise-resistant. The software follows a modular approach, allowing easy construction of signal chains.

The open source nature of the system allows researchers and developers to create new processing modules for the software, adding support for multiple acquisition and stimulation systems or adding support for the Open Ephys hardware to third-party software. Being able to share modules and algorithms with the community allows researchers to easily reuse technical work and share their methods with the community.

3.1 Introduction

Closed-loop electrophysiology introduces a series of challenges to acquisition system development not traditionally considered. Traditional, or open-loop, acquisition needs to be concerned only with maintaining signal fidelity and recording the data, which is stored and analyzed offline. Stimulation or other events are regarded as external systems working independently. In this case, only consideration synchronizing methods to show event timing in the recordings is considered with respect to the interaction between stimulation and recording.

With closed-loop, however, the need for online analysis arises [76]. To generate the appropriate feedback stimuli, data must be actively monitored and analyzed by an algorithm to detect when and how to stimulate. These algorithms and its parameters are heavily experiment-dependent and must be able to be adjusted to fit the research.

Most commercial systems feature closed-source software, which can not be modified beyond the vendor intentions. This becomes an issue when the possibilities allowed by the manufacturer limit or disallow the creation of specific algorithms required by the nature of an experiment. In these cases, researchers are faced with the options of modifying their experiment around the capabilities of the tool or purchase a different system that allows their specific experiment. Moreover, often vendors tie their hardware to their software and vice versa, offering little compatibility with external systems.

Closed-source software pose another issue for closed-loop scientific research. In open-loop electrophysiology only the experimental conditions and the data itself is shared, while the actual hardware and recording process can be omitted and substituted for a different system with similar characteristics. Closed-loop feedback, on the other hand, includes an algorithm that is an integral part of the scientific process (Figure 3.1). Sharing this algorithm is required for externally reproducing the experiments [95]. However, algorithms or analysis processes created in closed-source tools can be difficult to share, might include details hidden by their source tool and are often tied to the specific system and not reproducible with different hardware or software.

Open-source projects are those allow code or any equivalent to be freely shared, inspected and modified [97]. There exists a wide variety of open-source licenses, ranging from those that allow creation of derived closed-source products to those that force any derivative work to share the open license of the original work [98].

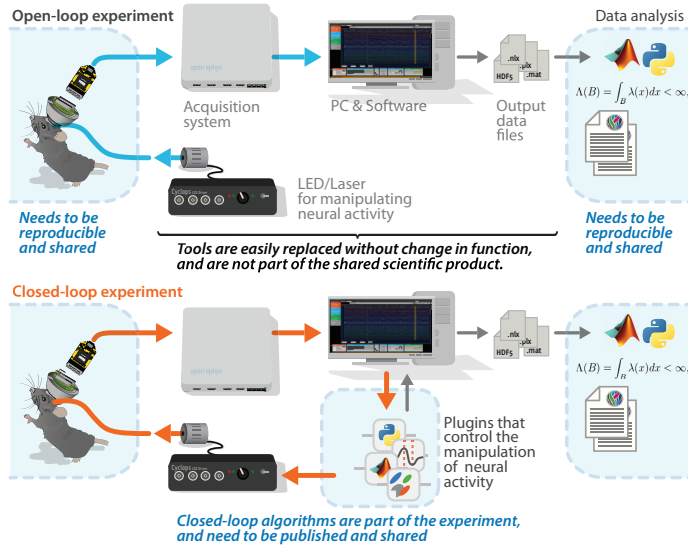


Figure 3.1: Algorithms governing closed loop feedback are integral part of the methodology and need to be shared for reproducing the experiments. From [96].

Both issues regarding algorithm creation and sharing can be solved by an open source tool for electrophysiology research. Having the code accessible would allow researchers to modify it in case the system would not meet their experimental needs, adapting the tool to the experiment instead of the opposite. An open source philosophy eases closed-loop algorithm sharing [99]. Being able to share not only the theory behind the algorithm but the code itself allows external reviewers to easily replicate the experiment or even adapt the algorithm to their own tools. Moreover, an open-source software would not be limited to a single hardware, but could be extended to support multiple devices as long as their interfaces are public. In the same way, an open-source acquisition hardware would allow modifications and its compatibility with different software suites.

For such a project to be successful, some conditions must be met [95]. The system has to be well maintained and easy to use. For an open-source project to grow, the community must be active, with users contributing their own upgrades and sharing technical discussions.

This chapter presents an open-source electrophysiology acquisition system fulfilling these characteristics. It is composed of hardware allowing multichannel acquisition, with I/Os for external device synchronization, as well as a soft-

ware with recording, visualization and online processing capabilities, designed for its use in closed-loop systems.

3.2 Materials and Methods

3.2.1 *Intan RHD2000 integrated circuits*

Intan Technologies (Los Angeles, CA, USA) RHD2000 series are ICs featuring both analog and digital stages specially designed for brain electrophysiology [100]. While traditional acquisition hardware has separate stages for amplification and digitalization this chip series, released in 2012, allows building very compact headstages. Since the ADC is close to the brain, data can travel to other parts of the acquisition hardware digitally encoded, thus being resilient to electrical noise.

The chip family is comprised of three devices, varying by their inputs:

- **RHD2216:** 16-channel device with differential inputs for each channel.
- **RHD2132:** 32-channel device with unipolar inputs, with a single common reference.
- **RHD2164:** 64-channel device with unipolar inputs and a single common reference.

All ICs of the family feature 3 auxiliary inputs, a digitally-configurable analog input bandpass filter and an optional digital high-pass filter for offset removal. They also feature internal voltage and temperature sensors that can be sampled in a manner similar to the input channels. A complete set of characteristics for the RHD2000 devices is listed in Table 3.1.

Communication with RHD2000 chips is done through a standard 16-bit, 4-wire Serial Peripheral Interface (SPI) bus, with both single-ended or LVDS electrical interfaces. Commands exist to trigger acquisition for each channel as well as configuring all their internal options. Acquisition frequency is determined by SPI command rate. Both 32 and 64 channel versions share the same command set, i.e., both have sampling commands for 32 channels. For each command, the 32-channel headstage just samples the selected channel and returns the corresponding bits on the rising edge of the SPI clock. The 64-channel variant, however, samples channels both n and $32 + n$ simultaneously. It then returns the bits for both values in each SPI clock using a Double Data Rate (DDR)

Characteristic	
Max. acquisition rate (for all channels)	30KS/s
Data output width	16 bit
Input range	$\pm 5mV$
Gain	192 V/V
Bit resolution	$0.195\mu V$
Input-referred noise	$2.4\mu V$
Analog bandpass filter low cutoff frequency	0.2Hz-1KHz
Analog bandpass filter high cutoff frequency	10KHz-20KHz
Digital high-pass filter cutoff frequency	$4.8\mu Hz$ -0.1Hz
Required power supply	3.3V

Table 3.1: Characteristics of Intan RHD2000 integrated circuits

scheme. This entails a data signal able to switch at twice the speed of the clock, including two bits per clock cycle. Bits in this signaling are captured at both rising and falling edges, as opposed to standard digital signaling where only one edge is used to sample data bits. In the case of RHD devices, the 32-channel variants use standard signaling with only one bit per clock cycle, captured on the rising edge. The 64-bit variant, using DDR and sending two bits per clock cycle, synchronizes the first 32 channels to be captured on the rising edge, and channels 33-64 on the falling edge.

Intan Headstages

Intan offers already-assembled headstages featuring their ICs in 16, 32 and 64 channel configurations. The devices optionally include a 3-axis accelerometer connected to the auxiliary inputs of the RHD chips, with the biggest headstage measuring 24mm x 15.5mm. They also provide thin cables for digital interfacing.

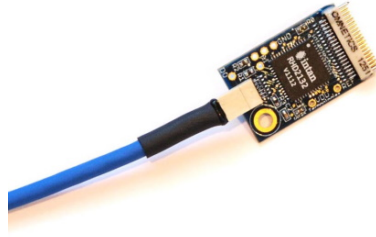


Figure 3.2: 32-channel headstage with cable. From Intan Technologies (http://intantech.com/products_RHD2000.html).

3.2.2 Xilinx FPGA

At the Open Ephys hardware core lies a FPGA driving acquisition and packing the data for transmission to the computer. The specific device was the XC6SLX45-2C chip, a FPGA pertaining to the Spartan-6 series [101], from Xilinx (San José, CA, USA). It features 43661 Xilinx Logic Cells, 2088 Kbit of RAM and 38 DSP units.

A commercially available module featuring this device was used. It was a XEM6310 module [102] from Opal Kelly (Portland, OR, USA). In addition to the FPGA it features 512MB of DDR RAM memory and a USB port. It can be plugged to a Printed circuit board (PCB) using a connector exposing most of the FPGA pins. Although the XEM6310 required power supply is 5V, the FPGA itself is powered by 3.3V lines. As a consequence, logic connected to any pin of the device must be limited to this voltage.

A set of HDL modules are available to facilitate communication through the USB interface. These are easily accessible from software using an Application Programming Interface (API) provided by Opal Kelly called *FrontPanel*. This API provides functions to set or read individual signals inside the FPGA as well as methods for high-speed transfer through USB.

3.2.3 Connectors

Omnetics (Minneapolis, MN, USA) connectors were used for connecting the acquisition chips with both the electrodes and the FPGA. The company specializes in miniature connectors, some of them widely used for neuroscience. 18 and 36-pin Nano Strip connectors were used for 16 and 32 channel electrodes respectively. These connectors are a *de facto* standard, used by a variety of probe manufacturers. For the digital interface, 12-pin Polarized Nano connec-

tors were used. These ensure that cables can not be connected in an incorrect orientation, as well as allow daisy-chaining multiple cables for longer lengths. An example of these connectors can be seen in Figure 3.3.

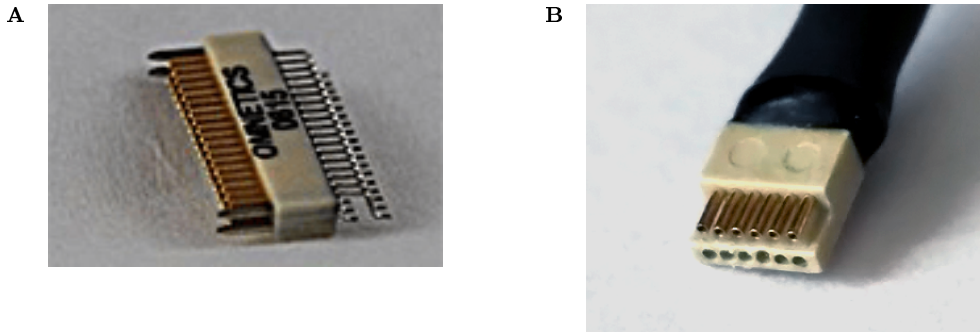


Figure 3.3: Omnetics connectors. **A:** Omnetics 36-pin Nano Strip. From Omnetics (<https://www.omnetics.com/products/neuro-connectors/nano-strip-connectors>). **B:** Omnetics 12-pin Polarized nano.

Additionally, Hirose Electric (Yokohama, Japan) DF40 connectors were used for PCB-to-PCB connection due their low profile.

3.2.4 JUCE Library

The Open Ephys software is coded in C++, a Object-Oriented Programming (OOP) language. This kind of programming languages are characterized for the existence of structures called *classes* which encapsulate a specific functionality and exposes interfaces to it through *methods*. Classes can be expanded by a process called *inheritance* in which a child class inherits all the parent class functionality and can add new methods as well as modify some behavior of the parent ones. This way, a class can expose a generic, common functionality that can be easily expanded and customized in child classes. In this paradigm, an *object* is a particular instantiation of a class. Different objects expose the same class functionality with differences on the data they hold.

The backbone of the Open Ephys software a C++ library called JUCE, originally designed for audio processing software. As such, it provides a powerful set of classes and methods to manage signal graphs. Those are comprised of nodes which can be created to perform some processing to an input and present the result as an output. Such nodes, or *processors*, can be connected in different ways, allowing a flexible method to create multiple signal processing chains.

The JUCE framework also provides utility classes and methods to handle the creation of a Graphical User Interface (GUI), data collections, file handling or network communication, making it possible to construct a rich application using its functionality.

3.3 Results

3.3.1 Headstages

The first headstages developed within the Open Ephys project were similar to those directly manufactured by Intan, targeted to silicon probes. They consisted in a simple PCB with the chip, an Omnetics Nano Strip connector for the electrodes and an Omnetics Polarized Nano connector for the digital interface. Since their design proved effective, with very few areas to improve upon, efforts were shifted into microwire electrodes, with special interest in tetraode recordings.

To that avail, a chronic drive implant was developed, the ShuttleDrive [103]. This is a device designed to easily guide microwire electrodes to the desired areas of the brain while protecting them. Electrodes are connected to the acquisition headstage through a Electrode Interface Board (EIB) which sits on top of the drive, fully enclosing the assembly. A 64-channel drive is 15mm tall, with a weight of 2 grams, making it possible to be used with both rats and mice. The EIB has gold plated holes already grouped to facilitate the construction of tetrodes.

The first EIB version featured two Omnetics connector for interfacing with either a 64-channel or two 32-channel Intan-style headstages. While weight was kept small, the height of the assembly proved detrimental to the ability for mice to freely move their heads. To improve into this aspect, a different, low-profile headstage was developed. This headstage lies flat over the EIB interfacing through a low-profile Hirose connector. Figure 3.5 shows a detailed comparison of both profiles while pictures of the different headstages can be seen in Figure 3.4. The lower torque point reduces mechanical stress on the animals, allowing for more natural movement.

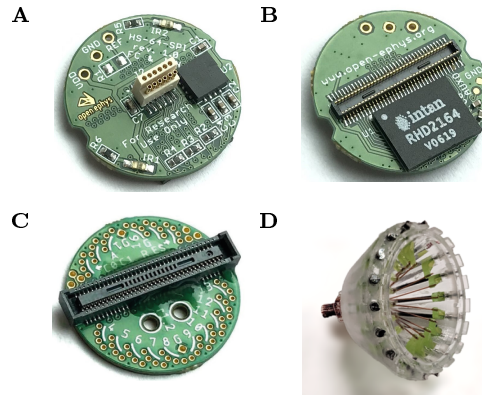


Figure 3.4: Open Ephys headstages. **A:** Flat headstage (top). **B:** Flat headstage (bottom). **C:** 64 channel EIB. **D:** ShuttleDrive.
From Open Ephys (<https://open-ephys.org/>).

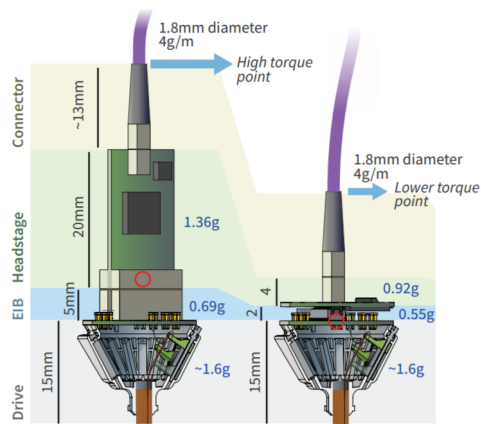


Figure 3.5: Comparison of sizes and torque point between Intan headstage and low-profile headstage when interfaced with a ShuttleDrive. From Open Ephys (<https://open-ephys.org/>).

3.3.2 Open Ephys acquisition board

The Intan RHD chips in the headstages need to be driven by a SPI controller. To that avail, an acquisition board was created. It does not only control the headstages but includes analog and digital I/Os, all hardware synchronized with the neural acquisition.

3.3.2.1 Hardware

Figure 3.6 shows the Open Ephys acquisition board. Controlling the system is the Opal Kelly XEM6310 FPGA board, with the main PCB acting as an interface for the external devices. It features 4 ports able to drive 2 64-channel headstages each. Independent power supplies are provided for each port to ensure minimal losses through the cabling.

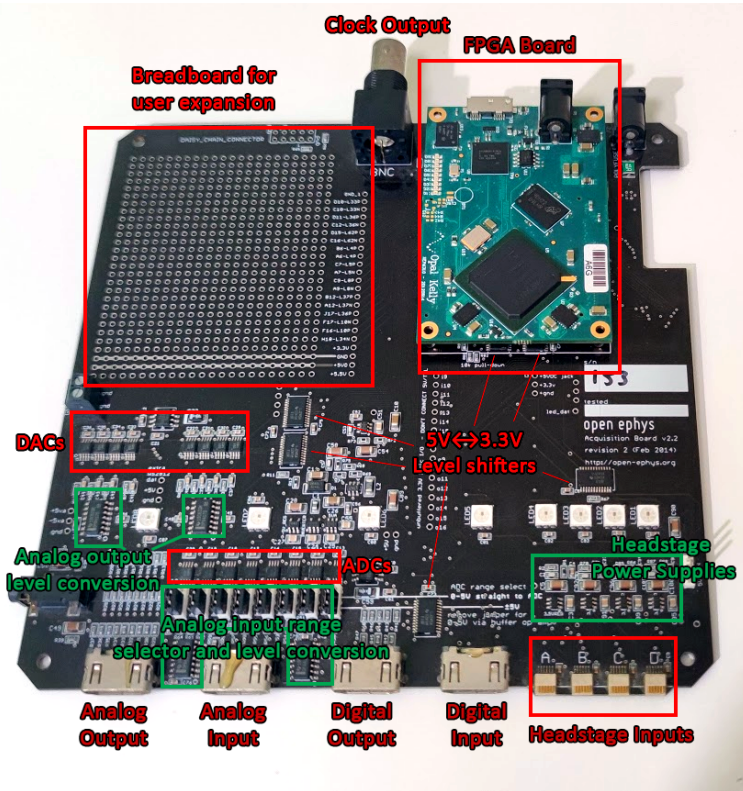


Figure 3.6: Open Ephys acquisition board and its components

The acquisition board also features ports to connect I/O boards, shown in Figure 3.7, for digital or analog auxiliary signaling. Digital lines, 8 input and 8 output lines, are connected to the FPGA through level-shifters to adapt the 3.3V FPGA requirement to 5V, which corresponds to widely-used TTL signaling levels. Analog output is provided through 8 Digital-Analog converters (DACs) driven by the FPGA, followed by level conversion circuitry to provide a $\pm 5V$ output. Likewise, 8 ADCs connected to the FPGA allow for analog inputs. The input levels are configurable through individual jumpers to be $\pm 5V$ or $0 - 5V$. The acquisition board also features a BNC connector which outputs the sampling clock.

Finally, to make it easier for users to expand the hardware in custom ways, a breadboard section was included, with lines connected directly to the FPGA.



Figure 3.7: Open Ephys I/O board

3.3.2.2 Firmware

The firmware configured into the acquisition FPGA, codenamed Rhythm, is responsible for driving the headstages, reading the state of the digital and analog inputs, generating the outputs and synchronizing all the data. The core of the firmware is a Finite State Machine (FSM) driving the SPI bus for the Intan RHD chips in the headstages. It is comprised of 4 states per bit, two for the high clock state and two for the low state, for 20 bits, 16 with data and 4 extra bits required by RHD timings. The FSM loops then for a total of $(16_{data_bits} + 4_{wait_bits}) \times 4_{states_per_bit} = 80$ states. Each 80-state loop corresponds to a single SPI command, which is sent to all headstages simultaneously, sampling their data in the same manner.

Having a FPGA clock 4 times faster than the SPI clock driving the acquisition chip allows to compensate for cable propagation delays. Digital systems capture bit values on clock edges. However, since the clock is generated in the FPGA, signals originating in the remote RHD device can suffer a delay due to electrical propagation over long wires and arrive off-phase with the clock. By having extra granularity of 4 cycles per bit makes it possible to finely select the instant in which incoming data is captured, thus being able to compensate for this delay.

To sample all channels, another loop of 35 cycles is done on top of the 80-state loop. Of these, the first 32 correspond to the sampling commands for the Intan RHD chips. The following three are used for auxiliary commands, such as configuration, reading the internal sensors or sampling the auxiliary signals. Each of these commands have a memory associated in which a variable-length list of commands can be filled. For each cycle of the 35-channel loop the next command for each of the three lists will be send in the appropriate slot, increasing the counter. All connected RHD chips are commanded simultaneously, so a sample for every headstage is collected for each of the 35 cycles. An overview of the command structure can be seen in Figure 3.8.B. External lines, such as TTL or ADC inputs, are sampled during the 35th cycle, along the last neural channel. This makes all signals, neural and external, synchronized by hardware, with a single timestamp.

Since the FSM has a total of $35 \times 80 = 2800$ unique states, the acquisition clock is set appropriately for the desired sampling rate. For example, for the maximum of $30KS/s$ a clock of $3 \times 10^4 Hz \times 2800 = 84MHz$ is needed. Sampling of the digital and analog inputs as well as setting the outputs is done at specific states, all following the main acquisition clock. For each complete $35 * 80$ cycle a single sample of data, containing the value of all 32 channels as well as the 8 digital and 8 analog inputs is timestamped with an unique sample counter and sent to a FIFO using the external RAM of the XEM6310 module. Data from the FIFO is then read by the software through the FrontPanel API and transmitted via USB. The FrontPanel interface is also used to configure acquisition parameters as well as signaling start and stop commands to the FSM. Additionally, for each complete sample, a pulse is sent to the external clock output. This output can be configured to pulse once every n samples instead. A block diagram of the firmware can be seen in Figure 3.8.A.

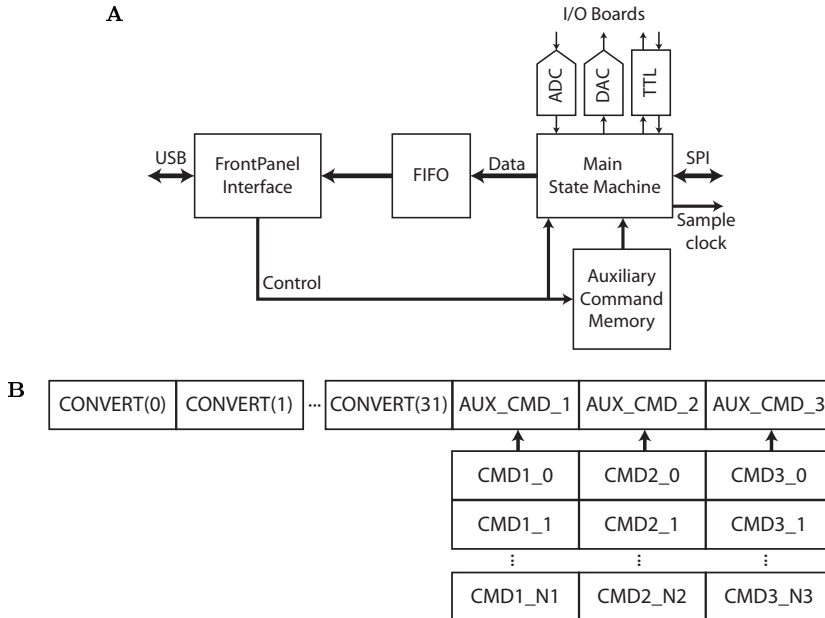


Figure 3.8: Rhythm firmware diagram. **A:** Block diagram of the firmware. **B:** Command loop structure.

3.3.3 Open Ephys Software

The Open Ephys GUI is capable of acquiring data from multiple sources, including the Open Ephys hardware, perform signal processing in real time and record data to disk. It is based around the concept of a configurable signal chain. This is comprised of a set of nodes, called Processors, connected between them to perform a specific set of operations depending on their configuration and the order they are connected. Different processors are able to acquire data (*sources*), process it (*filters*) or visualize or store it to disk (*sinks*). The software allows the creation of any possible chain by means of simple drag-and-drop procedure, instantiating nodes from a processor list into the signal chain, where node-dependent parameters can be set. Processors, specially visualizers, can display data in a visualization area. Figure 3.9 shows the application and its areas.

A key aspect of the software is the plugin architecture. Processors are not built into the main program, but are developed and compiled separately. This allows for any developer to expand the functionality of the application. While the software is distributed with a set of basic plugins, community developed

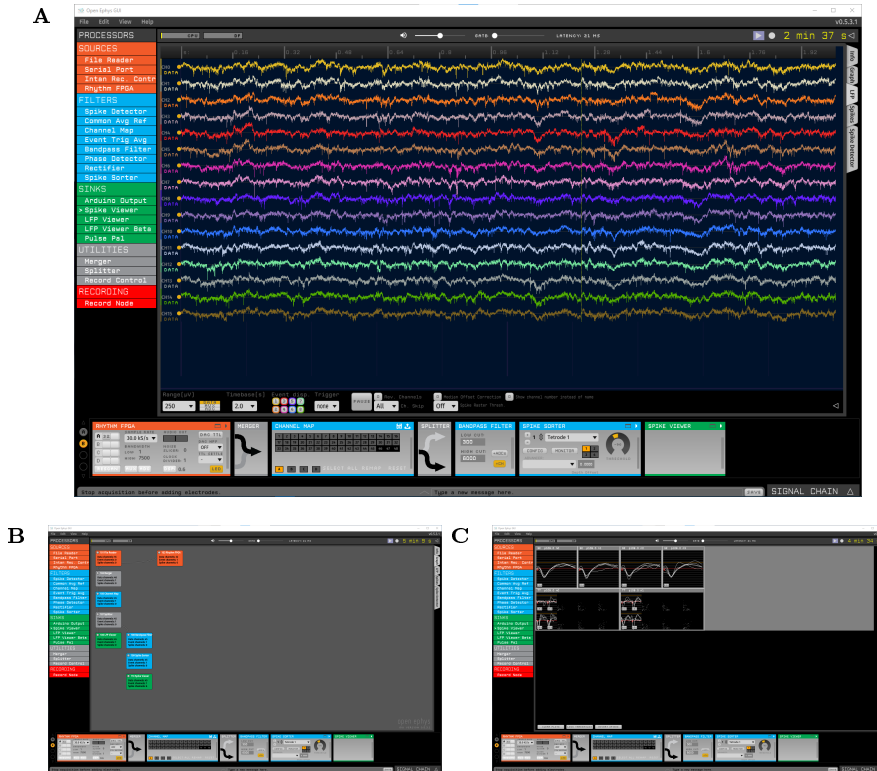


Figure 3.9: Different views in the Open Ephys GUI software. **A:** LFP continuous display. **B:** Signal chain overview window. **C:** Neural spike viewer, displaying both single units and tetrodes

processors can be installed. There are four different kind of plugins that expand the functionality of the GUI:

- **DataThread:** Implement acquiring data from hardware, such as the Open Ephys acquisition board.
- **FileSource:** Make it possible to play back recorded data from disk in a particular format provided by the plugin.
- **Processor:** The most common plugin, these can implement any kind of processing to perform with the data.
- **RecordEngine:** Provide a file format to record the data to disk.

Since the software is based on audio processing libraries, plugins can easily perform any kind of digital signal processing, allowing online analysis. Some examples of commonly used plugins, included in the base distribution of the software, are digital filters, phase detectors or spike processing. Spikes, due to their particular importance, are featured as a special data event in the GUI data structures. As such, the software features an online spike sorter able to use windows [29] and PCA [50], [104], [105] methods. Spikes can be visualized and saved independently, alongside the continuous data.

Aside from spikes, the software can handle other types of non-continuous events, such as changes on digital lines, also called TTL events. These can be inputted into the signal chain, for example through the digital input of the acquisition board or a network plugin, or generated inside the signal chain through a plugin, such a phase or threshold detector. Input events can be used to trigger specific functions within the processors. An example is the record control module, able to start and stop disk recording through an external trigger. Both input and generated events can be outputted through a number device sinks. These can be used to trigger external stimulation hardware for closed-loop control [70].

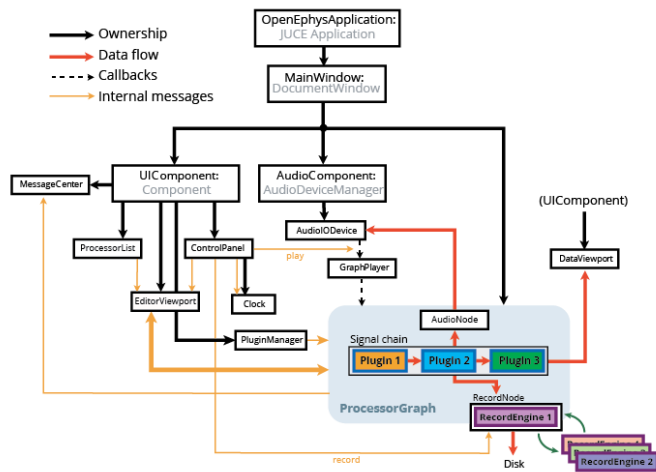


Figure 3.10: Class structure of the Open Ephys software. From [96].

Data can be recorded to disk in multiple formats, which can be expanded with plugins. Some examples included with the base distribution are a plain binary files with a separate JSON including metadata information or Neurodata Without Borders [106], an open format for scientific data. Thanks to the

modular nature of the software, data can be recorded at multiple points of the signal chain, allowing to simultaneously record, for example, raw electrophysiology data along with the filtered and processed signals used for closed-loop feedback.

The application was developed in C++ using the JUCE library. Following the OOP paradigm, it consists of a hierarchy of objects, most of them derived from classes provided by the JUCE API. Figure 3.10 shows the dependency structure of this class hierarchy. Under this structure, plugins are created as classes that inherit from one of the four base classes provided by the plugin API that the software implements. Creating a basic plugin requires filling a simple function, available from a template, while more complex plugins can be created by using more advanced interfaces. An example of a simple rectifier plugin can be seen in Figure 3.11

```
rectifier.h

#include <ProcessorHeaders.h>
class Rectifier : public GenericProcessor
{
public:
    Rectifier() {}
    ~Rectifier() {}
    void process(AudioSampleBuffer &buffer) override;
}

rectifier.cpp

#include "Rectifier.h"
Rectifier::process(AudioSampleBuffer &buffer)
{
    for (int ch = 0; ch < nChannels; ++ch)
    {
        const int nSamples = getNumSamples(ch);
        float* bufPtr = buffer.getWritePointer(ch);
        for (int n = 0; n < nSamples; ++n)
        {
            *(bufPtr + n) = fabsf (*(bufPtr + n));
        }
    }
}
```

Figure 3.11: Example code for a simple processor

The software is heavily parallelized, with different threads for acquisition, visualization, recording to disk and signal chain processing. This last thread in particular is triggered in regular intervals by a high-precision timer. Data is processed in buffers whose size is dependent of the processing interval and the sampling rate. This kind of block processing introduces a variable latency with a maximum value of the buffer length.

3.3.4 Performance

Using all the neural inputs, the Open Ephys acquisition board is able to acquire from 512 channels at 30KS/s. Thanks to LVDS communication and the delay compensation features of the firmware, tethers can be as long as 10m without any signal distortion.

The amount of channels supported by the software is only limited by the characteristics of the computer running it as well as the complexity of the signal chain. With fast Solid-State Drives (SSDs) counts of over 8192 channels, from high-density probes [41], have been successfully recorded to disk.

Acquisition latency is introduced by two factors: USB transmission and software block processing. The former creates a maximum of 10ms delay, while the second is configurable, with 20ms default but able to be as low as 5ms. Total mean latency with default settings is measured as 20 ms.

3.4 Discussion

Closed-loop electrophysiology introduces an active feedback component to the classical acquisition and analysis approach. This element, the algorithms to trigger and create stimuli, are tied to the acquisition setup, which often makes them difficult to share and analyze by the community, one of the pillars of scientific knowledge. An open source approach solved this, as algorithms and procedures can be closely inspected and reproduced, not only for the original system, but can be adapted to alternative devices or software if needed.

The Open Ephys system facilitates this process with its modular architecture, by making it easy to create and share plugins independently of the rest of the software. Its growing adoption, with citations in more than 200 papers, has resulted in many laboratories developing plugins for various online analysis and event detection tasks, of which over a dozen are now featured in the official plugin repositories of the Open Ephys organization. Thanks to its open-source nature, third party projects, such as Bonsai [107], now support the Open Ephys hardware, while the software has been used to support other hardware as well, such as Neuropixels [41] or the Neuralink interface [108].

Developed amongst the first generation of digital-headstage devices, the Open Ephys acquisition system has three main paths for improvement: latency, asynchronous signal management and wiring.

Due to the buffering architecture used in the software, coupled with the bulk transfer characteristics of USB the minimal achievable latency is 5ms, with the mean being 20ms. While this allows for closed loop experiments based on slow events, such as LFP rhythms [70] it is not enough to respond to faster signals, such as action potentials. A maximum latency of 1ms would be required for spike-based closed-loop feedback.

Neuroscience research is steadily moving from head-fixed electrophysiology experiments to more complex environments, integrating multiple elements such as animal tracking or video feeds alongside the neural data. All these diverse data sources, of different nature and sampling rates, need to be perfectly aligned and synchronized for analysis and their use in closed-loop feedback algorithms. The Open Ephys system, however, has been developed following traditional signal processing methods which are designed for synchronous signals. While it can handle thousands of channels from the same source, mixing independent sources can lead to alignment problems. A completely different architecture is needed to properly process asynchronous sources.

Lastly, while wiring has been greatly improved from analog systems, with tethers being as thin as 1.8mm with a mass of 4.1g/m, they still entail difficulties for animal movement range. Moreover, a SPI cable for driving an Intan RHD2000 chip in LVDS mode requires 10 wires. As such, even with the number of wires reduced from analog systems, multi-wire rotative commutators are still complex to manufacture and prone to mechanical failure.

All these concerns are addressed in posterior hardware iterations, as presented in following chapters.

3.5 Conclusions

The Open Ephys project is a complete open-source electrophysiology acquisition system developed with the ideas of flexibility and community sharing.

Its hardware is able to acquire up to 512 channels at 30KS/s with a 16bit resolution and multiple I/O lines, both digital and analog, for interfacing with external devices. Neural data is digitized at the headstage level, allowing long cables without inducing noise into the signal.

On the software side, the Open Ephys GUI is a plugin-based application able to handle thousands of channels, both for recording and signal processing, including online spike sorting. Its modular nature makes it possible for re-

searchers to develop their own algorithms, integrate them into an experiment and share not only the results but the process with the community. Designed with closed-loop experimentation in mind, the GUI is able to process events and trigger stimulation devices through output interfaces.

The open source nature of the project allows for researchers to fully share their entire workflow, including the algorithms, triggers and stimuli used for closed-loop experimentation. This information can be used to replicate an experiment with the Open Ephys system, or studied and adapted for other existent systems.

Chapter 4

Open Neuro Interface (ONI): High performance acquisition

This chapter presents the Open Neuro Interface (ONI) standard and a high-performance hardware implementation, ONIX. ONI is a specification designed for communication between a computer and an acquisition system able to acquire data from multiple, heterogeneous sources while ensuring synchronization. ONIX is an electrophysiology acquisition system following the ONI specification. It features high-bandwidth and low latency, able to acquire from hundreds of channels and perform closed-loop feedback in the sub-millisecond range. It includes lightweight headstages with 3D-tracking and stimulation capabilities. Headstage tracking allow the use of an active torque-free commutator. Thanks to the reduced strain on the animal and the advanced capabilities, the ONIX system can be used in complex, long-term experiments.

4.1 Introduction

4.1.1 *High-bandwidth heterogeneous systems*

Modern neuroscience is trending towards integration of multiple data and stimulation sources of diverse nature. Animal tracking and computer vision, widely used in behavioral experiments [109], [110] are being combined with electrophysiology acquisition [92]. Many systems also include stimulation both environmental [111] and neuronal [76]. In parallel, technological advances in consumer electronics has brought a plethora of off-the-shelf sensor devices of interest for neurophysiology and behavioral studies, such as Inertial Measurement Units (IMUs), VR tracking, small high-power LEDs or miniaturized cameras and microphones. Both developments have opened the door to complex experiments [112] featuring composite devices [113], [114] and multiple, independent sources of data.

However, no standardized way to access the variety of available sensors exists. This becomes a roadblock for the development of new acquisition hardware. While it is inevitable to deal with low-level IC communication when creating new hardware, developers need to also create high-level interfaces for data transfer to the computers. Due to time constraints, these can end being simple ad-hoc data links, only useful for a particular project and not reusable. Some devices have developed complex and mature APIs to access their data [41], but they are geared toward those specific devices and feature little to no way to extend their capabilities.

On the other hand, advances in electrophysiology technology have increased the number of channels able to fit in a single probe exponentially, with devices featuring hundreds of channels per shank being available [36], [115]. Bandwidth requirements for current data sources can range from a few KB/s of tracking data to dozens of MB/s from high channel count electrophysiology or real-time video stream devices, such as head-mounted microscopes [114].

A standardized interface able to bidirectionally access a variety of devices, independently of their nature, would be of great interest. Such an interface would free developers from the need to create data transfer interfaces and focus on the specifics of their device only. Such an interface should support a wide range of data rates and hide all specificity behind a set of common and accessible API.

4.1.2 Tether issues

There is a growing interest on neuroscience experiments featuring bigger, more natural and meaningful spaces [91], [92]. Wiring, however, becomes a burden to animal movement.

While the move to digital headstages has reduced the number of wires per cable, multi-wire tethers are still the norm. For example, a SPI bus for full-duplex communication such as the used in RHD2000 acquisition chips [100] require 4 lines. LVDS communication, required for high speeds over long cables, double the amount to 8, which then totals to 10 when including power wires. Thus, cables of ranging from 10 to 14 wires, for multi-headstage support, are common. More wires in a cable results in an increase to the total weight to the tether.

Weight and thickness are not the only issues caused by multi-wire cables. For an animal to be able to move freely in the experimental space, the tether must be able to rotate to avoid becoming twisted, which could lead to a broken cable or reduce the range of movement of the animal. This is achieved through the use of rotary commutators, devices that divide a cable in two segments able to rotate independently.

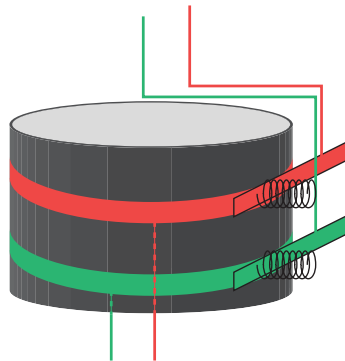


Figure 4.1: Diagram of the internals of a rotary commutator

For non-coaxial cables, a rotary commutator uses a slip-ring structure (Figure 4.1). Wires from one cable are connected to rings in a cylindrical structure. Connectivity to wires on the second cable is performed through brushes, laminar metal pads held against the rings by a spring mechanism ensuring continuous contact even when rotating. This structure is complex to manufacture and prone to failure, these difficulties increasing with the number of wires.

4.1.3 Latency in closed-loop experiments

Feedback stimulation in a closed-loop experiment must be performed within the timescale of the biological event of origin. Temporal association in behavioral tasks can take seconds, while neural plasticity occurs in a few ms [86], to cite some examples. Action potentials, on their part, have a duration of 1-2ms [13], so being able to act in that timeframe would open new experimental possibilities.

The main sources of closed loop latency are data transmission from the acquisition system and processing times. This gets aggravated by newer high-density probes with hundreds or even thousands of channels [87]. While algorithms using the computation capabilities of newer Central Processing Units (CPUs) and GPUs are being developed [89] to reduce processing time, latency due to data transmission from the acquisition system to the computer is still a limitation.

4.1.4 Overview

This chapter presents the Open Neuro Interface (ONI) specification and API as well as an ONI-compliant acquisition hardware, ONIX.

ONI is a standard for heterogeneous neuroscience experimentation. It is designed to abstract data from multiple asynchronous, independent devices into streams that can be accessed by any software by the use of a simple API. Data framing with timestamping ensures that synchronization between devices is possible. ONI is device-agnostic, so as long as any hardware complies with the specification it can communicate with software using the API.

ONIX is a specific implementation of the ONI standard. It is an acquisition device featuring sub-millisecond latencies and bandwidths of up to 500MB/s. Various headstages for electrophysiology, featuring 3D tracking capabilities, were created. Headstages or other devices communicate with the host system through ultra-thin, coaxial cables.

4.2 Materials and Methods

The two main hardware components of the ONIX system, which will be presented in detail in section 4.3, are the host card and the headstages. The former includes all devices required for intercommunication with both the computer and the headstages, while the latter include multiple sensor and stimulation components. This section includes the description of some elements that conform those parts.

4.2.1 Bus standards

Two main interconnection standards were used to connect hardware to the computer and between different circuit boards: PCIe and FMC.

PCIe is a bus standard designed for high-speed data transfers between computer peripherals. It is based on point-to-point communication, connecting devices directly to a root complex, usually located in the CPU. Data is transmitted serially through differential pairs, or *lanes*, which can be aggregated into single links, multiplying bandwidth. Thus links referred as x1, x4, x8 or x16 refer to one, four, eight or sixteen differential lanes dedicated for a single peripheral. Line speed is determined by the version, or generation (abbreviated Gen.), of the protocol. Table 4.1 shows the bandwidth for each current standard, with Gen. 3 and Gen. 4 being, at the time of this writing, the most common in commercial computers.

Gen.	Year	x1 Bandwidth	x4 Bandwidth
1	2003	250 MB/s	1 GB/s
2	2007	500 MB/s	2 GB/s
3	2010	958 MB/s	3.938 GB/s
4	2017	1.969 GB/s	7.877 GB/s
5	2019	3.938 GB/s	15.754 GB/s

Table 4.1: Year of introduction and speed of the different current PCIe versions

FMC Mezzanine Card (FMC), or ANSI/VITA 57.1, is a standard defining daughter boards connected to a PCB featuring a FPGA or similar device. It allows a fixed main board to change its I/O capabilities by replacing the mezzanine modules. The standard defines a specific form factor as well as two types of connection density. LPC (Low Pin Count) connections provide user 68 lines. The HPC (High Pin Count) standard provide 160 user lines as well as 10 dedicated differential pairs for serial transceivers and additional clocks. The

standard allows for the user lines to work as single-ended lines or coupled in differential pairs, 34 for LPC and 80 for HPC. Both densities share the same physical, 400-pin connector, the difference referring only to the connectivity between it and the FPGA present in the main board.

4.2.2 *FPD-Link III devices*

FPD-Link III [116] is a protocol designed by Texas Instruments (Dallas, TX, USA) to transmit high-speed digital data over a single differential pair or coaxial cable. A FPD-Link system is comprised of two elements: a serializer, which packs and transmits a data bus and a digital clock, and the deserializer, able to receive and reconstruct them. The FPD-Link protocol is designed for transmission of video data, transmitting pixel color values, vertical and horizontal synchronization signaling (V-SYNC and H-SYNC) and the pixel clock (PCLK). As such, the data bus is comprised of 12 or 24 bits, which are common pixel color widths, as well as the two synchronization bits.

In addition to the high-speed data channel FPD-Link III devices also feature a bidirectional configuration channel, also called backchannel. Through it, communication using an Inter-Integrated Circuit (I²C) bus is possible between devices at both ends of the link. General Purpose Input/Output (GPIO) lines are also provided and transparently connected between the deserializer and the serializer through this backchannel.

The devices selected for its use in the ONIX system were the ds90ub933 serializer and ds90ub934 deserializer, used to transmit data from the headstages to the host. They are able to transmit 12-bit data, plus sync signals, up to 100MHz, feature a 400KHz I²C bus and include 4 GPIO lines. They were selected over 24-bit variants due to power and PCB layout considerations as well as compatibility with existing devices such as Miniscope [114]. The devices were configured to transmit the data over a coaxial cable.

4.2.3 *FPGA devices*

To drive the different acquisition devices, coordinate communication through the FPD-Link interface and transfer data to the host computer two different kind of FPGAs were used. Designs for both of them were developed in VHDL.

4.2.3.1 Kintex-7 FPGA

A Xilinx FPGA from the Kintex-7 series, the model xc7k160t, was used in the ONIX host for gathering data from the deserializers and transmitting it to the computer. This is a mid-end device featuring 162240 Xilinx Logic Cells, 11700 Kbit of RAM and 600 DSP units.

Additionally, FPGAs of the Kintex-7 family include integrated transceivers for PCIe bus communication. The devices are able of a Gen2 x8 interface, allowing a maximum bandwidth of 4GByte/s.

Numato Nereid board

A commercially available PCB was used, the Nereid board from Numato Lab (Bangalore, India), which features the FPGA as well as a 128Mbit SPI flash memory to store the bitfile. It has a form factor able to fit into a normal computer PCIe slot with a x4 interface and includes a SO-DIMM slot populated with 4GB of DDR RAM. Custom hardware is implemented in a daughter boards that are plugged into the Nereid through a standard HPC FMC interface. The location of this connector is such that, when the main board is connected into a computer, the custom daughter board can expose connection ports through the back of the computer. Figure 4.2 shows a detailed description of this board.

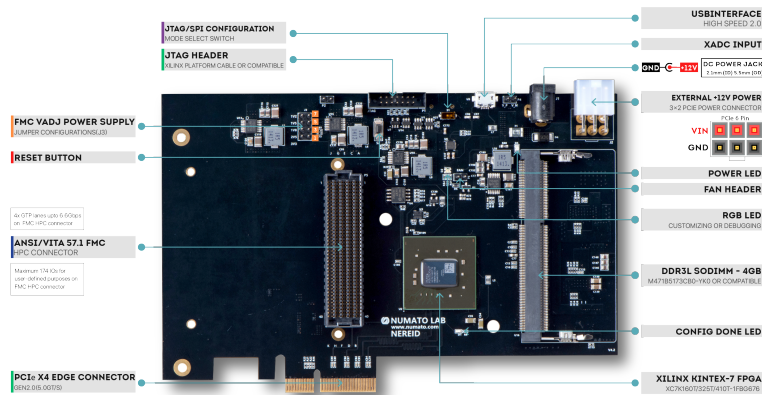


Figure 4.2: Overview of the Numato Nereid board. From Numato (<https://numato.com/product/nereid-kintex-7-pci-express-fpga-development-board/>).

PCIe interface module

To take advantage of the PCIe interface on the Kintex-7 FPGA an existing communication module was used. RIFFA (Reusable Integration Framework for FPGA Accelerators) [117] is a framework for communicating a computer with a FPGA board. It can be configured to create a number of transmission channels, able to receive or send data. Each channel is completely independent and can be accessed in parallel with all others. RIFFA makes use of Direct Memory Access (DMA) and interruptions to achieve high bandwidth over the PCIe link.

RIFFA does not implement the low-level interface with the PCIe bus. For that, a module offered by Xilinx to use with its FPGA was used.

4.2.3.2 MAX-10 FPGA

Since size and weight are important priorities for a headstage a different, smaller device was used in them. The MAX-10 FPGA series from Intel (Mountain View, CA, USA) are designed for small, embedded applications. As such, they feature a small footprint and reduced power requirements, in comparison with bigger devices such as the Kintex-7. Following this specialization on small size, MAX-10 FPGAs do not require an external memory for storing the configuration bitfile. Instead, they embed flash storage inside the same chip, reducing the component count needed for a design featuring these chips.

In particular, the 10M08DF device was used. It contains 8000 Intel Logic Elements, 378Kbit of RAM and 24 DSP units. Regarding its internal flash storage, it features 312KByte which can be distributed between one or two bitfiles and user memory. Two packages were selected, the V81 variant measuring 4x4mm and housing 56 GPIO lines and the M153 with 8x8mm footprint and 112 GPIOs.

4.2.4 3D Tracking

All ONIX headstages feature 3D tracking to accurately follow the animals. This is accomplished using two complementary systems: SteamVR tracking and a 9-axis Inertial Measurement Unit (IMU).

4.2.4.1 SteamVR Tracking

SteamVR is a VR platform developed by Valve (Bellevue, WA, USA) featuring highly accurate object positioning in 3D space. A SteamVR tracking system is composed of two or more basestations, called *lighthouses* and a number of trackable devices, such a headsets, controllers, etc. as showcased in Figure 4.3.

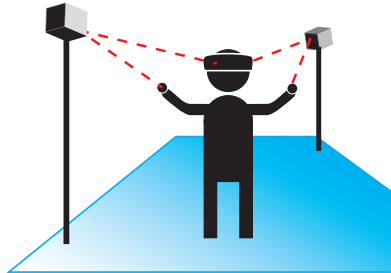


Figure 4.3: SteamVR overview

Two versions of the tracking system exist, sharing the basic working principle: Base stations, positioned in fixed and known positions in space, emit either an horizontal or vertical laser plane that sweeps vertically or horizontally, respectively, as shown in Figure 4.4. Trackable devices contain infrared photodiodes able to precisely detect when the light from the base stations touch them. Knowing the precise angle in which light from the vertical and horizontal sweeps are detected by a photodiode makes it possible to trace a straight line from a base station to it. Multiple base stations create different lines whose intersection point can then be calculated, determining the spatial position of the photodiode. By having more than one sensor in an object and knowing their relative position, orientation can be calculated as well.

The difference between versions 1 and 2 of SteamVR lies in detection of the precise sweep angles by the trackable objects.

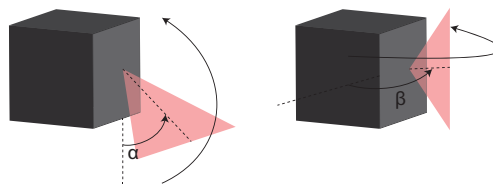


Figure 4.4: Lighthouse plane sweeps

SteamVR version 1

Version 1 of the tracking protocol requires precise synchronization between base stations. As a consequence, it is limited to only two lighthouses. In addition to the laser plane sweeps version 1 stations also make use of light pulses covering the entire space to synchronize. These flashes are detected by both the base stations themselves and the trackable objects, allowing all involved devices to be in synchrony.

At system setup, base stations are assigned as lighthouse A or B. Each cycle starts with lighthouse A emitting a pulse, followed by a pulse produced by lighthouse B. The duration of these two pulses indicate which of the stations will proceed to sweep and whether it will be a vertical or horizontal sweep. These pulses are followed by the sweep and a known length end of cycle period. Timing information can be seen in Table 4.2. Full tracking requires of four cycles, a total of 39.6ms, to cover both sweeps of both base stations.

Pulse start, μs	Pulse length μs	Source station	Meaning
0	65 - 135	A	Sync pulse
400	65 - 135	B	Sync pulse
1222 - 6777	~ 10	A or B	Laser plane sweep
8333	1556		End of cycle

Table 4.2: Lighthouse v1 activation Timings

With this schema, incidence angles are calculated by precise timing synchronization, as the device is able to count, for each individual photodiode, the time difference between the synchronization pulses and the sweep detection.

SteamVR version 2

Version 2 of SteamVR tracking addresses the limit of two base stations and improves accuracy by encoding the sweep angle into the laser itself, thus removing the need to synchronize elements.

Data is sent through amplitude modulation of the laser beam. A series of bits at 6MHz is transmitted using differential manchester encoding (Figure 4.5). This method allows accurate clock reconstructions by ensuring that there is always a transition between bits, while the bit value is encoded by the presence or absence of an additional transition mid-bit.

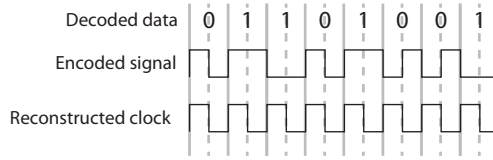


Figure 4.5: Differential manchester encoding

The data stream is created by the output of a 17 bit Linear Feedback Shift Register (LFSR). This is a special kind of binary shift register in which the input is created by summing the values of specific bits in the register, as shown in Figure 4.6. The selected bits can be expressed in polynomial form. For example, a LFSR in which the input is comprised of the sum of the 1st, 4th and 8th bit would correspond to the polynomial $x^8 + x^4 + x + 1$. All LFSR return to their initial value a set number of cycles. Those polynomials that ensure a LFSR of width W have a period of $2^W - 1$ cycles are called maximal-length polynomials [118]. 32 different 17 bit polynomials are defined by the SteamVR standard, all being maximal-length, thus having $2^{17} - 1 = 131071$ possible values.

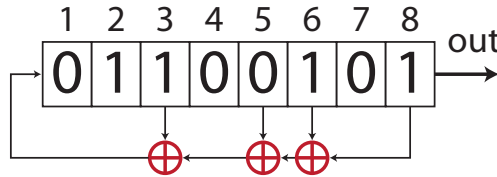


Figure 4.6: An example 8-bit LFSR with polynomial $x^8 + x^6 + x^5 + x^3 + 1$

In contrast with version 1 base stations, version 2 lighthouses are continuously performing sweeps, controlled by a single mechanical rotor with slanted mirrors. A complete rotor revolution performs one full horizontal and one full vertical sweep. This sequence is synchronized with the LFSR in a way that a complete revolution coincides with a full period of the shift register. This way, by receiving 17 consecutive bits the receiver is able to decode the state of the LFSR. This informs of the precise rotor position and, with it, the horizontal or vertical angle that corresponds. This requires knowing which base station, and by extension which polynomial, corresponds to the signal being decoded. This is achieved by placing two infrared receivers in close proximity for each sensing point. These devices will detect light from the same lighthouse at a very small offset, resulting in two 17-bit sequences from the same polynomial. Since the basestations polynomials are designed so their sequences are completely differ-

ent for each one, this can be used to locate the correct one and identify the base station. Knowing base station and sweep angles, triangulation is performed with the same process as with version 1.

Since base stations are not synchronized, there is a risk of beam collisions. This is minimized by the devices having slightly different revolution speeds, so collisions are never in synchrony. Possible values for a full revolution are 21.85ms and 19.98ms. Operating in an individual manner, more than one base station can be used with version 2, allowing for larger areas.

4.2.4.2 *Inertial Measurement Unit (IMU)*

Although SteamVR tracking is highly precise, the rate it produces full positional data is approximately 25Hz for version 1 and 50Hz for version 2. In some cases, it might be necessary to detect fast movements between full tracking samples. This can be achieved by the use of a 9-axis IMU.

This device is a Micro-Electro-Mechanical System (MEMS) integrated into a chip offering three different types of measurement, each monitoring 3 spatial axes: An accelerometer is able to detect linear movement as well as orientation relative to the ground, by detecting the acceleration of gravity; a gyroscope is able to measure rotations along any of the axes; and a magnetometer is able to act as a compass, measuring the heading of the device. Of these measurements, only data from the first two modes were used as a complement for tracking data.

The particular device used was the BNO055 from Bosch (Gerlingen, Ludwigsburg, Germany). It can provide accelerometer and gyroscope data at 100Hz. It can measure accelerations up to 16g and maximum rotation speeds of $2000^{\circ}/s$. Digital communication is performed through an I²C bus.

4.2.5 *Acquisition devices*

For neural data acquisition, Intan Technologies RHD2164 chips were used. These devices have the ability to acquire 64 channels of neural data at 30KSamples/s, with an input range of $\pm 5mV$ and $2.4\mu V_{rms}$ of input-referred noise.

Three kind of probes were used. Microwire probes, including stereotrodes and tetrodes, were interfaced through the use of Open Ephys EIBs boards. For silicon multichannel probes, adapters with multiple Omnetics 36-pin Nano Strip connectors were used.

The third type of probe used were Neuropixels probes [40], [41]. There are high-density, low-noise probes manufactured using Integrated Circuit lithography techniques. These devices provide an array of 960 electrodes located in a 10mm long, $70 \times 24 \mu\text{m}$ cross-section shank. Of those, 384 channels can be selected to record simultaneously. Filters to separate LFP and Spike bands are integrated into the same silicon body of the probe, along the electrodes.

4.2.6 Stimulation devices

Electrical stimulation was provided in current-controlled form, generated by a Howland current pump [119]. This circuit acts as a transconductance amplifier generating a fixed current, independent on the output impedance, defined by the input voltage. This tension was controlled by an Analog Devices (Norwood, MA, USA) AD5683 DAC.

For optical stimulation, a CAT4016 LED Driver from ON Semiconductor (Phoenix, AZ, USA) was used. This device provides constant current, controlled by an analog pin, to up to 16 output channels specially designed for LED operation which can be enabled individually by a digital bus. To have control over the stimulation current, a TPL0501 digitally-controlled potentiometer from Texas Instruments (Dallas, TX, USA) was connected to the current-sensing pin of the LED Driver. To allow for higher LED currents, the 16 output channels were tied together in two 8-channel groups, each connected to a single output, thus enabling a LED current up to 8 times the one set by the potentiometer.

4.2.7 Software

- **EAGLE** (Autodesk, San Rafael, CA, USA) is a electronic design suite including schematic and PCB design. It was used to develop the different pieces of custom ONIX hardware.
- **Vivado Design Suite** (Xilinx, San José, CA, USA) and **Quartus Prime** (Intel, Santa Clara, CA, USA) are the design softwares for Xilinx and Intel FPGAs, respectively. They are required to create the bitfiles for their respective devices from HDL code such as VHDL or Verilog, each taking into account the specific basic hardware elements for each vendor. The suites include simulation tools to test the designs before transferring them to the actual hardware, programmers to upload the bitfiles to the FPGAs as well as capabilities able to debug the physical devices through a JTAG interface. Vivado was used to create bitfiles for the host PCIe

board, while Quartus was used for the headstages. Most of the designs were created with VHDL, with a few modules made in Verilog.

- **Bonsai** [107] is an open source visual programming language capable of handling asynchronous data streams of different natures, with an ample library of predefined functions for vision and data processing. These capabilities have made it popular as a control framework for complex experiments, specially those including behavior study. Input nodes for the ONIX hardware were made for the Bonsai language and used to construct basic modules providing neural and tracking data.
- **Visual Studio** (Microsoft, Redmond, WA, USA) is a IDE for C/C++ and .NET applications, including coding assistant tools and a powerful debugger. It was used to build the low-level ONI library in C, as well as the Bonsai ONIX nodes using C#.

4.3 Results and Discussion

4.3.1 ONI specification

Open Neuro Interface (ONI) is an open standard describing a high-speed interface between a computer and a collection of devices, which can be of different nature. Its goal is to provide a single, unified protocol to communicate with the variety of instruments widely used in neuroscience such as electrophysiology acquisition devices, tracking systems, cameras or stimulators. It defines both a logical structure for the different elements as well as the format in which the data is transferred to the computer. It, however, does not define a specific hardware transport layer, leaving that open to different implementations, but only how the data should be organized.

Figure 4.7 shows a diagram of the ONI standard. The most basic elements of the structure are *streams*, *registers* and *devices*.

A stream represents a unidirectional flow of data. Direction is named from the computer perspective, input streams being data going into the computer and output streams data originating from it. Stream data can be continuous with a fixed rate, e.g., data from an ADC or a camera, or sporadic, e.g., the interaction of a button or a multichannel digital trigger.

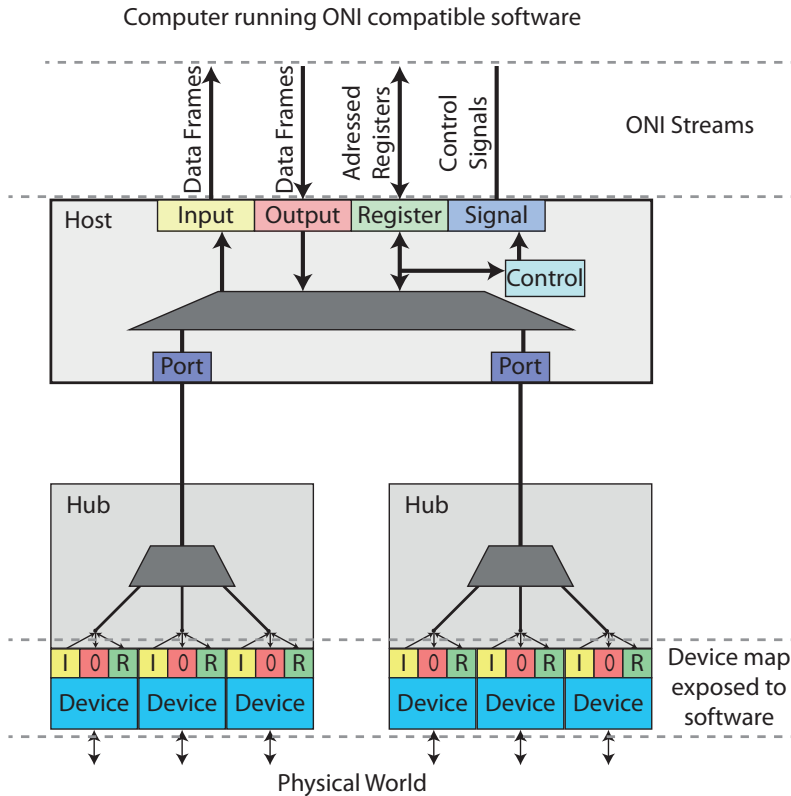


Figure 4.7: ONI specification diagram

Registers are a map of values referenced by an index, or address. Those are always passive, accessed only by request of the computer. A single register can be either read-only, write-only or accept both write and read operations.

Devices are producers or consumers of data. The term usually refers to a physical device, such as an ADC or a stimulation circuit, with access to the physical world. However, a device can also be a virtual element, e.g., a timer or a control interface. All ONI-compliant devices provide three interfaces: a mandatory register map, an optional input stream and an optional output stream. The collection of all available devices is called the register map.

Some extra structural elements are defined in the standard: A hub represent a physical or logical group of devices. For example, a headstage with neural acquisition, tracking and stimulation devices. A host is the element that acts

as an interface with the computer, aggregating data from the devices and transmitting it. A host contains a number of ports that connect to one hub each through a link, which can be physical, i.e., a cable, or virtual, for example to a logical hub comprised of devices physically existing in the same hardware as the host.

Regarding the computer interface and data format, the ONI standard defines four channels that must be implemented.

A register interface must be present with, at least, the registers shown in Table 4.3. They are divided into two groups: Those pertaining to the register access interface allow interfacing with the register channel of any particular device. Global registers offer control and information over the acquisition process. The standard allow specific implementations to add their own registers to the map.

Register	Type	Description
Device ID	Device register interface	Target device unique identifier
Register Address		Register address to access
Register Value		Value to write in register or read value from register
Read/Write		Set register operation to read or write
Trigger		Start register access operation
Running	Global	Start and stop acquisition
Reset		Reset the system
System Clock		Frequency in Hz of the main system clock
Acquisition Clock		Frequency in Hz of the clock driving acquisition and generating host timestamps
Reset acquisition counter		Reset the host timestamps to 0

Table 4.3: Required registers in the ONI specification

A low speed, unidirectional signal stream is used to notify the computer of the completion and result of asynchronous events, particularly device register access and system reset. It is also responsible of transferring the whole device map, containing details about every connected device. Specifically, each entry of the map contains:

- A unique identifier, created by the hub index and the device index within the hub.
- An identifier indicating the device type.
- A number indicating the version of the device, or its driver firmware
- Device input sample size in bytes
- Device output data size in bytes

High speed data transmission between the computer and the devices is done through the input and output streams. Data is transferred in frames, each consisting of

- A 64-bit timestamp, created by the host hardware, representing the moment the data arrived to it. For output frames this value is not used.
- A 32-bit value with the identifier of the origin or destination device
- A 32-bit value with the size of the payload
- The data payload

Output payloads are completely device dependent. Data originating from the devices is organized in samples, each frame containing a single sample in its payload. A sample contains a 64-bit timestamp created by the hub containing the device and the sample data. Having two timestamps, one from the hub and another from the host, allows hubs to run independently with different clocks and still be able to synchronize the data.

ONI Library

LibONI, a library written in the C programming language was created to provide easy access to any ONI-compliant system. It manages data to and from the different streams and provides simple C functions to perform operations such as frame reading and writing, system configuration and register access. To support multiple implementations, a simple modular driver system was developed. Different implementations must only provide small low-level functions to write and read raw data to and from the four streams.

Bindings in different programming languages, such as Python and C#, are provided to allow for easy interoperability.

Structure of ONI-based systems

A typical ONI system is comprised of four main layers, as described in Figure 4.8, where the top elements are the extremes of the system, individual devices and user data, and bottom ones the interconnection between the system and the host computer.

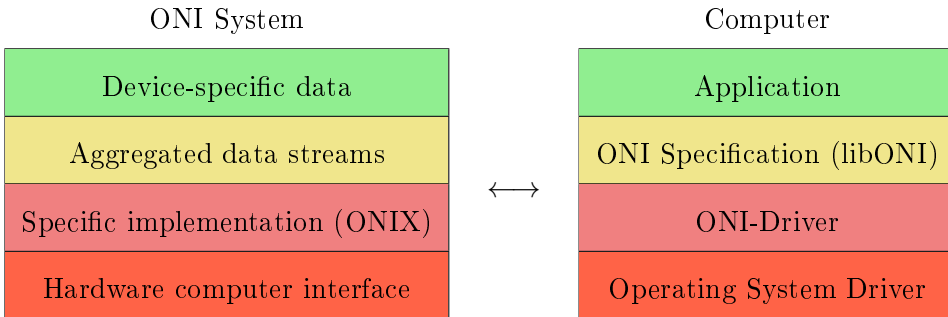


Figure 4.8: OSI-Style relation of layers in an ONI system

Device data, at the top layer, is produced by a data source. It is sent transparently to the application through the ONI specification and API, so the software only needs to know which type of data it is and act accordingly. For example image data would be displayed as a video feed while electrophysiology data would be shown as a waveform and used for spike sorting.

The next layer is the core of the ONI specification. On the hardware side, it defines how device data must be packed and send through a predefined stream, with a clear frame structure. On the software side, the ONI API, using the knowledge of this structure, is able to decode the streams and present device data as independent packets to the application. The actual methods to communicate with the devices are not specified in the standard, as it is a transparent process that gets packed through the ONI specification.

The following two layers are tightly related and worth explaining in opposite order. Bottom layer is the actual interconnection between the ONI-based system and the host computer. This might be any interface such as PCIe, USB or even network. On the hardware side, this is the device that performs actual communication. While on the software side is the low-level operating system driver for said device. This might be an off-the-shelf part, and no control over it is specified in the standard.

The second to bottom layer represents the actual acquisition system. On the hardware side, it represents the system communicating with the different devices, framing it following the ONI standard and sending it through the data link. Since the actual link might not feature the virtual streams required by the specification, for example a USB chip featuring a single bidirectional data stream, it is the responsibility of the hardware implementation to pack the ONI streams in a way fitting its physical data link. On the software side, a lightweight module for the API, provided by the system developer, translates the hardware-based link into ONI streams, for the API to interpret.

4.3.2 ONIX hardware

ONIX is a hardware neural electrophysiology acquisition system implementing the ONI specification. Figure 4.9 shows all its components. It is comprised of a PCIe board acting as an ONI host and different types of headstages, taking the role of hubs in ONI terminology. Communication between the host and each headstage is done using the FPD-Link III protocol, thus allowing for bidirectional communication and power transmission over a single coaxial cable.

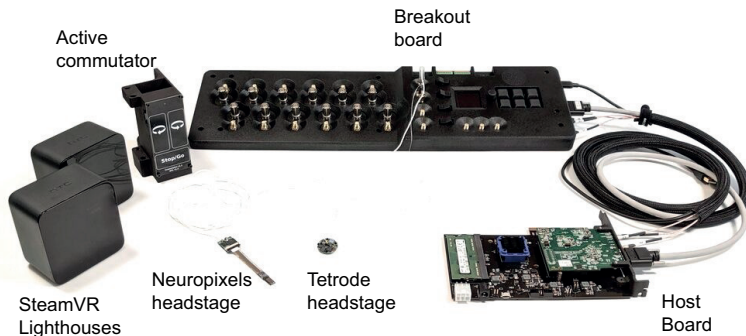


Figure 4.9: Overview of all the components developed for the ONIX system

4.3.2.1 Host board

Following the ONI specification, the ONIX host board was designed to aggregate data from different hubs and devices and interface with the computer through a high-speed PCIe bus. The Numato Nereid board with a Kintex-7 FPGA at its core was used as the base of the host system. A FMC daughter card was made containing all custom electronics for interfacing with external devices. Figure 4.10 shows the complete board as well as the PCB layout of the FMC card, which is comprised of a 6-layer stack-up.

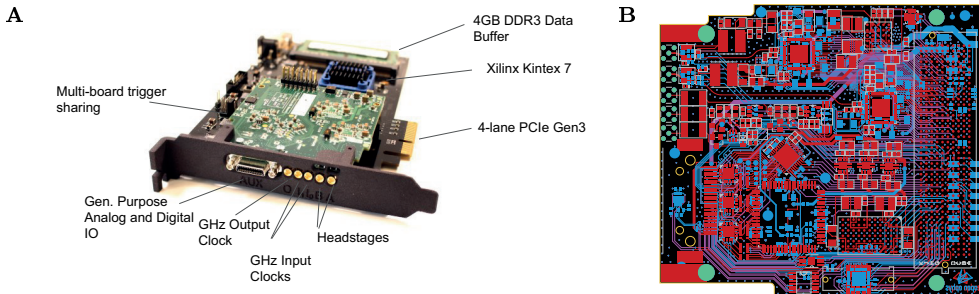


Figure 4.10: ONIX host board. **A:** Overview of the full host board, with the FMC card assembled over the Numato Nereid. **B:** PCB layout of the FMC card.

The main components of the FMC board are two ds90ub934 deserializers in charge to communicate through a coaxial link with different hubs. External connection to the deserializers is made via MMCX connectors. These devices require two different power inputs require, 3.3V for I/O lines to the FPGA and 1.8V for the core. Those are provided through 3.3V pins in the FMC connector, originating in the Nereid board, and a DC-DC Buck converter to efficiently derive the 1.8V lines. Power for the external devices, transmitted through coaxial cables, is derived from a 12V source also provided through the FMC connector. Instead of being converted to a fixed voltage, a combination of configurable step-down converters and digitally-controlled potentiometers was used to digitally control the link voltage for each port. This allow to compensate for larger cables or adjust for different devices.

The host board also contains interfaces for general purpose analog and digital signals. It features 12 analog lines that can be, through digitally controlled analog switches, independently routed to either a multichannel ADC or DAC. The board does not feature direct digital lines, however, but 5 high-speed differential pairs, 2 output and 3 input, which can be used to interface with a variety of digital systems.

For extra synchronization capabilities, the host board contains 2 buffered high-speed clock inputs and 1 clock output, accessible through coaxial MMCX connectors. It also features an internal connector with 4 differential pairs of configurable direction, designed to connect multiple boards to work in a synchronized manner.

4.3.2.2 Headstages

The main source of data on the ONIX system are the headstages, that fulfill the role of hub on the ONI specification. While the specification allows for any kind of data, ONIX headstages are specially designed for brain electrophysiology acquisition, neural stimulation and animal tracking. Two different types of headstages were developed: those for microwire and silicon probes using Intan RHD2164 chips, in both 64 and 256 channel variants, and a headstage designed to drive neuropixel probes. All headstages communicate with the ONIX host board through a coaxial cable, carrying bidirectional communication originating from a ds90ub933 FPD-Link III serializer as well as power and feature a MAX10 FPGA at their core. There are complex devices with a high density of components requiring PCBs with tight tolerances, featuring a trace separation of 0.05mm and buried and blind vias in a 8-layer stack-up.

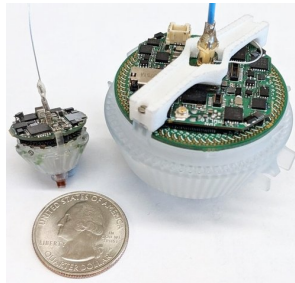


Figure 4.11: RHD-based headstages mounted in drives for tetrode recording, 64 and 256 channel variants. Size comparison with a quarter US\$ coin (24mm diameter)

Figure 4.11 shows the microwire and silicon headstages, mounted on drives for tetrode acquisition, which make use of the smaller V81 FPGA packaging. The whole assembly weights 3.65g in the case of the 64 channel variant, while the 256 version totals 12,5g. The bare PCBs for the headstages weights are 1.5g and 4,7g respectively.

Functionality-wise, both variants include full 3D tracking capabilities using a combination of SteamVR sensors and IMU devices as well as optical and

electrical stimulation circuits. The main difference being the channel count and more stimulation channels in the bigger device. Figure 4.12 shows a detailed component description of the devices. The 256-channel headstage has some circuitry that would allow it a certain degree of autonomy without the need of a host for low-bandwidth operations, but actual development on those capabilities is still pending.

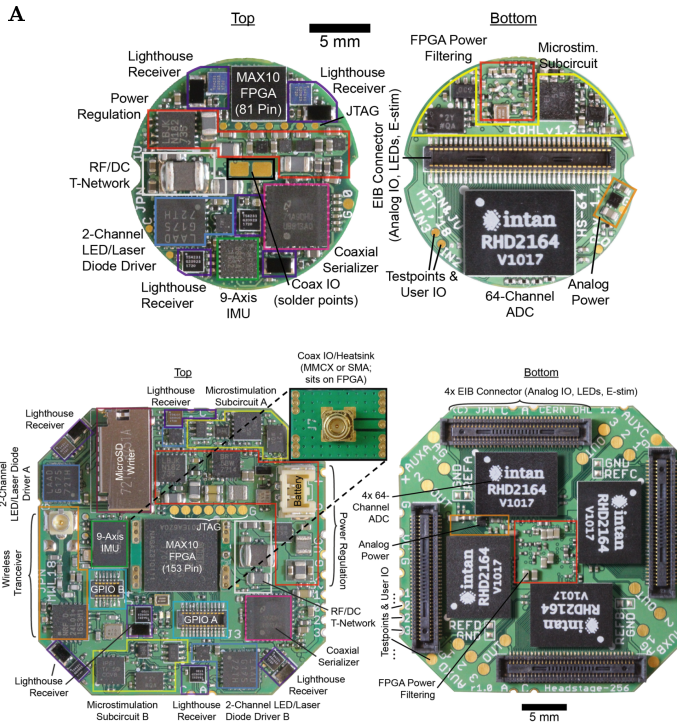


Figure 4.12: Detailed component description of RHD-based headstages. **A:** 64-channel version. **B:** 256-channel version.

The Neuropixels headstage allows the connection of two probes. Due to the higher I/O needs of Neuropixels hardware, the bigger M153 package was used for its FPGA. Contrary to the previously described headstages, it does not contain stimulation circuitry. However, the headstage still retains 3D tracking capabilities, through both SteamVR and IMU devices.

Although not an ONIX headstage, Miniscope devices, integrated lightweight head-mounted microscopes [114] are supported by the system and can be

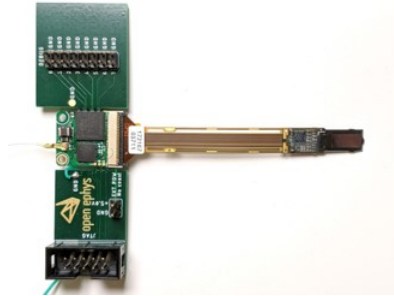


Figure 4.13: Neuropixels headstage with one probe attached. Includes programming extensions that can be broken after production.

plugged using the same coaxial interfaces than the headstage, with pixel data being enclosed into standard ONI data frames.

4.3.2.3 Breakout board

The ONIX host board (Figure 4.14) includes a port for analog and digital I/O. To facilitate their use, a breakout, or connection board was designed. It features direct access to all 12 analog channels through BNC and SMA coaxial connectors. 16 digital lines are provided, 8 input and 8 output, their values communicated to the host board through a high-speed differential interface. This communication channel is also used to provide user signaling through the use of user-configurable buttons on the breakout board. Finally, to facilitate cable management, it can connect to the coaxial ports on the ONIX host so headstages and external clocks can be connected to the breakout board instead, featuring radio frequency switches to safely isolate the coaxial ports if needed.

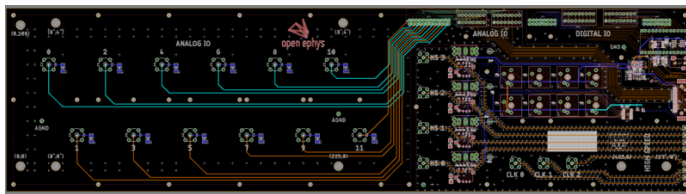


Figure 4.14: PCB layout of the breakout board

4.3.3 Tethers and torque-free commutator

The use of a single coaxial cable between headstage and host system offers numerous advantages over classical multi-wire tethers. Since a coaxial requires a single filament and its shielding, it can be made much more thinner and lighter, helping to reduce strain on the animal. Connectors can also be smaller, which reduces the center of mass and torque point on headstages, as shown in Figure 4.15. Cable width is only limited by the electrical resistance it provides to power transmission. In the case of the ONIX headstages, a coaxial cable of 0.4mm in diameter has been successfully tested for up to 2m, while 0.8mm work for lengths exceeding 10m, resulting in thin and lightweight tethers.

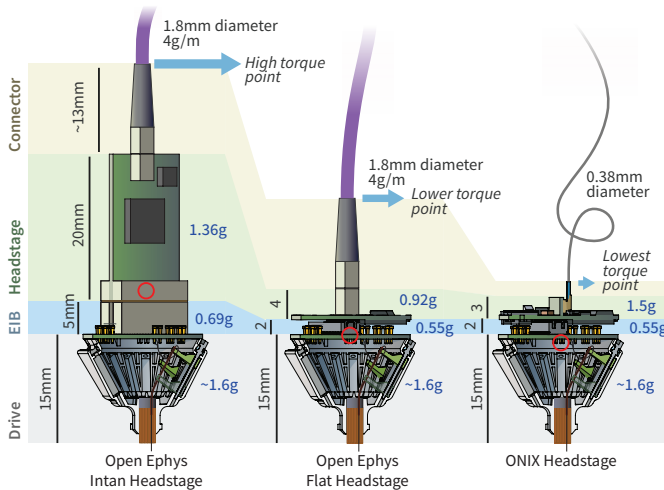


Figure 4.15: Comparison between different Open Ephys classic and ONIX headstages. Red circle represents the center of mass. Blue arrow the approximate point at which tethers become flexible.

Another advantage of coaxial wiring is the availability and reliability of rotary commutators. While those designed for multi-wire cables are complex mechanisms, the radial nature of coaxial cables allows more robust construction, by replacing spring-loaded brushes by ball bearing or liquid metal contacts. Commercial commutators for coaxial, designed for high frequency operations, are widely available.

The ONIX system includes such a commutator with an added advantage. Thanks to the 3D tracking capabilities of the headstages, their position and orientation are always known. This allows to motorize the commutator, mak-

ing it pro-actively follow the animal movements, as depicted in Figure 4.16, instead of turning mechanically by the tether torque as is the case with regular commutators, thus reducing strain on the animal head. An active commutator also facilitates wiring management. Since no mechanical force stemming from the animal is needed, long cables can be kept out of their reach by tying them up with elastic strings, allowing the tether to be kept high when the animal is near the commutator and extend when it moves outward. This also negates the effect of wiring weight on the animal, reducing strain even further.



Figure 4.16: Commutator actively following headstage orientation.

This torque-canceling effect, coupled with the low-weight headstages described in section 4.3.2.2 makes the system suitable for long-running experiments featuring freely-moving animals, as it does not hinder movement or supposes a significant hurdle even to mice (Figure 4.17).

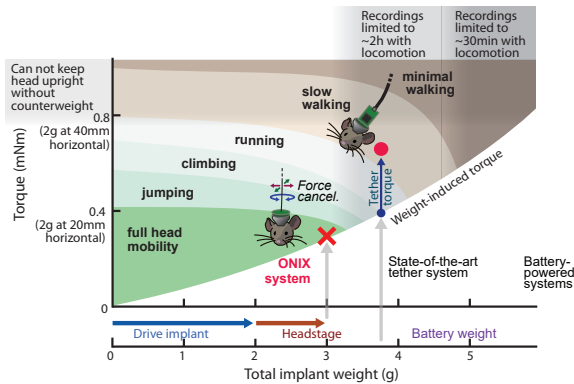


Figure 4.17: The combination of lightweight headstage and force-canceling active commutator allows for little to no strain in mice, allowing for free range of movement.

An experiment was performed to demonstrate the effect of headstage weight and tether torque on animal behavior, shown in Figure 4.18. A 3D environment was constructed and a mouse with no previous experience with this space was left

to explore. The paths traversed by the animal during its exploratory behavior were precisely followed using a computer vision tracking system. Four different trials of 2 hours were recorded, alternating torque-free ONIX headstages and standard Intan headstages using a passive commutator, to avoid any bias caused by exhaustion or familiarity. A separate, 4-hour trial was performed by a non-implanted animal, acting as a control path for naturalistic behavior.

This experiment shows clearly distinct behavior patterns. Exploratory paths with a standard headstage are shorter, with the animal favoring being immobile in specific areas and avoiding those requiring high jumps. Behavior with the ONIX system, however, is more regular across all the available space, closely resembling the path distribution followed by the control trial with no implant.

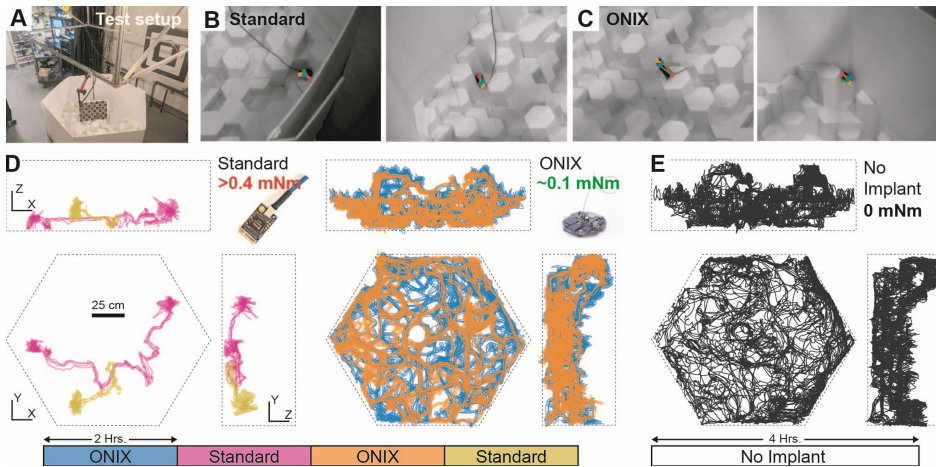


Figure 4.18: Difference in exploratory behavior in mice using standard and ONIX headstages during 2-hour trials. **A:** Experimental setup and calibration. **B:** Video-tracked mouse with a standard, Intan headstage. **C:** Video-tracked mouse with a torque-free ONIX headstage. **D:** Exploratory paths during four 2-hour trials alternating standard headstages, in pink and yellow, and torque-free ONIX headstages, in blue and orange. **E:** Control exploratory path during a 4-hour trial with no implant.

4.3.4 ONIX firmware

The ONIX firmware, coded mostly in VHDL, can be divided in two big blocks. The Hub, or remote side, implemented in the MAX10 FPGAs present in the headstages, and the host side, running in the Kintex-7 device present in the host board, directly plugged to the PC. Data is transferred between both parts through a logic interface using the FPD-Link III serializer/deserializer devices. Figure 4.19 shows an overview of the firmware for one example link.

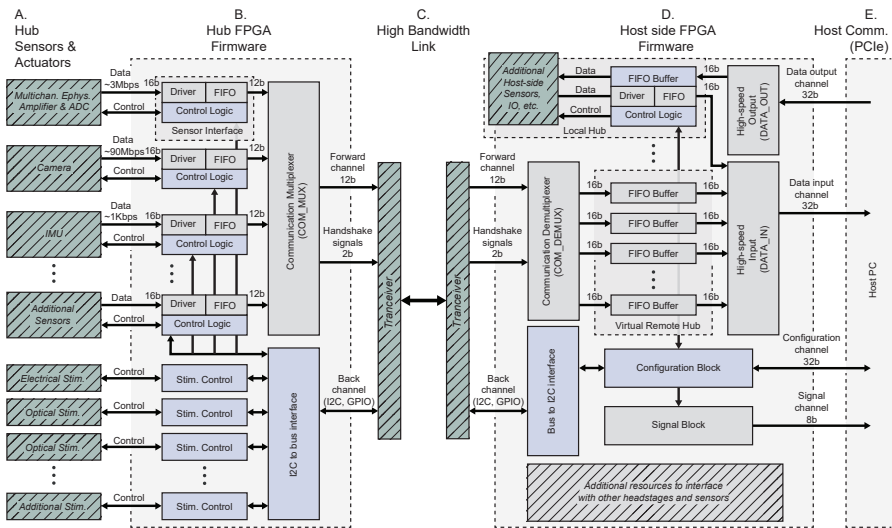


Figure 4.19: Block overview of the ONIX firmware. Green boxes represent physical hardware, gray blocks handling data from devices to the computer and blue both bidirectional data or data from the computer to the devices.

4.3.4.1 Hub firmware

The hub firmware, running in ONIX headstages, has four functions:

1. Control and acquire data from sensor devices
2. Control and drive stimulation and actuator devices
3. Mux sensor data for serializer transmission
4. Receive and route command and configuration data to the devices

The firmware is designed to be modular, so different device combinations can be easily achieved. All the firmware functions run concurrently, while devices might be running asynchronously, potentially even with different clocks. The firmware synchronizes and multiplexes all the data for transmission over the ONI streams.

To achieve this, each device has an interface module in the firmware in charge of controlling the device, which can be in its own clock domain. The interface is comprised of two main parts: one dedicated to streaming data from the device and other for control registers. Devices on an external ONIX hub do not feature a high-speed output stream, one of the optional features on the ONI standard. This is due to a limitation the FPD-Link III transceivers used in the ONIX system, which have only one high-speed channel, used for the acquisition data stream, and a low-speed bidirectional channel, for control signals. All device interface modules export a compound signal with the required values for its entry on the global device table:

- Hardware identifier
- Implementation version
- Sample size, including timestamp
- Write data size. As mentioned before, this value is 0 for hub devices connected through a FPD-Link III interface.

The data stream path starts with a driver module inside the device interface reading and decoding data from the sensor. Although device data can be of varied size, for consistency it is packed in multiples of 16-bit words in what are called samples. A sample is always comprised of:

- A 64-bit timestamp generated by the local clock
- The sample data

The next step in the data path is a FIFO buffer. This serves a triple function: First, it allows data to be stored while waiting to be transmitted. Second, if the device interface uses a different clock from the core section, a dual-port FIFO helps crossing the data between domains safely. And third, since the serializer chip has a 12-bit interface, the FIFO has asymmetrical ports, performing 16 to 12 bit conversion. Due to this conversion, every three 16-bit written words require 4 read cycles to be fully read.

Data from all device FIFOs is gathered in a communication multiplexer, in charge of data transmission over the high-speed channel of the serializer. This includes both device data and the device map. After a reset, the module proceeds to send the number of devices on the hub followed by the map entry for each one. After the device map is completed, the multiplexer enters monitor mode, waiting until there is at least one sample worth of data on any of the FIFOs and then sending their contents following the protocol described in section 4.3.4.3. The module sends data blocks from different devices in a round-robin schedule.

Control data is provided by a bidirectional register interface accessed through a simplified version of the Wishbone bus. It is a master-slave bus topology whose chronogram can be seen in Figure 4.20. Its signals are:

■ Master signals:

- **cyc**: (1 bit) Signals start of a transaction.
- **idx**: (32 bits) Indicates the target device. The higher 16 bits are reserved, the following 8 bits represent the hub, and the lower 8 bits the device index within the hub.
- **addr**: (32 bits) Indicates the target register address. Only the lower 16 bits are used.
- **we**: (1 bit) 0 indicates a read operation. 1 indicates a write operation.
- **val**: (32 bits) Data for writing operations.

■ Slave signals:

- **val**: (32 bits) Data returned in read operations.
- **ack**: (1 bit) Signals the end of a transaction.
- **err**: (1 bit) Asserted at the same time as *ack*, indicates an error on the transaction.

Devices act as slaves, with their bus ports connected into a demultiplexer that routes its input to the appropriate device based on the *idx* field. Bus commands from the host to the hubs are encapsulated using a bus over I²C protocol described in detail in section 4.3.4.3. A module connected to the I²C

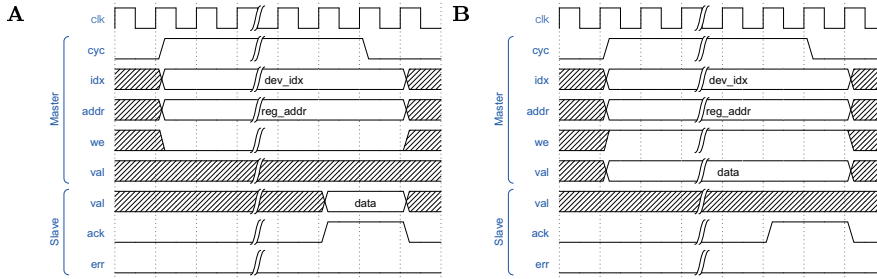


Figure 4.20: Chronographs of the bus protocol for successful operations. For unsuccessful operations, *err* will assert at the same time as *ack*. **A:** Read operation. **B:** Write operation

port of the serializer decodes the commands and drives the bus demultiplexer input as the local bus master.

GPIOs are also used as control signals. Specifically, one of the lines available in the serializer is used to transmit a reset signal from the host, while other lines can be used for stimulation triggers, as they are faster to issue than I²C commands.

4.3.4.2 Host firmware

Similarly to the hubs, the host contains a main data path for high speed data and a bus-based control path. It acts as a bridge between the protocols transmitting data through the coaxial link and the protocols and data formats described on the ONI specification for communication with a computer.

The host data path stems from the deserializer interface, which receives raw data from the hub through the FPD-Link III connection. A communication demultiplexer receives the data, decodes it and fills a FIFO buffer for each device. These FIFOs serve as means to transfer data from the deserializer clock domain to the host clock domain, as well as to store data waiting to be sent to the computer. This is similar to the case of the hub and, in fact, from the host point of view, the communication demultiplexer acts as a local copy of the remote hub. This demultiplexer also receives, after reset, the complete device map and exposes it to the host through the signal channel, as described on the ONI specification. Multiple demultiplexers allow for multiple ports, each creating its own virtual hub in the host, reflecting the connected headstages.

A high speed input controller monitors the FIFOs until any of them has a full sample worth of data. Once this happens, it reads the full contents, packs it

into a ONI frame and sends it through the input channel to the computer. If multiple devices have data at the same time, the input controller sends one frame from each device in a round-robin manner. As explained in section 4.3.1, a frame has the following structure:

- A 64-bit timestamp, created by the host hardware, representing the moment the data arrived to it.
- A 32-bit value with the identifier of the origin or destination device
- A 32-bit value with the size of the payload
- The sample data

The host timestamp is created by the input controller based on the host clock. This allows all samples from different devices to be timestamped by a local counter, simplifying synchronization procedures.

The ONI register map is interpreted by a module connected to the configuration channel. It manages all global states, such as acquisition start/stop or reset requests from the computer software. Device register interface commands are translated into bus transactions, with the configuration block acting as the main master of the bus structure. The module is connected to a demultiplexer that routes transactions to the appropriate hub, based on the provided device identifier. Each hub port, with its communication demultiplexer, also features a bus over i2c controller that translates the bus requests and sends them through the coaxial link so they can reach the appropriate device on the remote hub.

Following the ONI specification, the result of device register access is notified to the computer through the signal channel. A signal controller reads the status from the configuration block and generates the appropriate responses. This signal block is also responsible for gathering the device maps from all hubs and sending a consolidated full map after a reset procedure.

Finally, the host firmware also contains a local hub. This is, following the definition, a collection of devices located on the host board, or virtual ones implemented within the firmware. Some of such devices are:

- A heartbeat device implemented in firmware which produces a simple timestamp every fixed period of time.
- The device in charge of communicating with the breakout board and accessing the digital I/Os.
- A device able to drive the host board ADCs and DACs to make use of the analog I/Os.

- Devices connected to the communication demultiplexer to report the physical status of the links as well as any communication errors.

As with the virtual hubs, the local hub is connected to the input controller with a series of FIFO buffers, one per device. Similarly, the control logic of the devices is connected to the bus network with the same demultiplexer structure. The only difference, from an architectural point of view, between the local hub and the remote ones is that the former has access to the high speed output channel driven by the computer. This is used in the I/O devices.

Communication with the computer is done through the PCIe bus using the RIFFA framework. Four different endpoints were created, one per ONI stream. A wrapper module translates the RIFFA-specific protocol into ONI signaling, which is performed through the use of FIFOs, except for the configuration channel, for which a register interface was developed.

4.3.4.3 Link protocols

Since the FPD-Link III devices are designed for their use with video sensors, which produce a constant rate of pixels, they do not offer ways to delimit valid data. A framing protocol was devised to ensure data transmission, shown in Figure 4.21. The H- and V-Sync signals are used to delimit the start and end of a valid transmission. Block of data sent from a device FIFOs starts with a header containing the device index in the hub, is followed by sample data and ends with a checksum to detect possible communication errors. The device map that is sent after reset follows a similar structure but with no device index, sending the number of devices, the map entry for each one and a checksum of the whole map. An invalid checksum on device map will disable the entire hub. The serializer pixel clock is driven by the hub communication multiplexer, usually at the same rate as the hub clock.

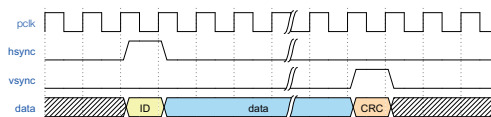


Figure 4.21: Communication protocol between serializer and deserializer.

While the ONIX devices operate under a complex bus structure, able to access any register from any device on all hubs, the serializer/deserializer link only offers a I²C bus as bidirectional communication. Moreover, due to how the link devices work, the number of accessible addresses is low, not enough for

the amount of devices present in some hubs. To work around these restrictions, a bus over I²C protocol was developed.

With this encapsulation protocol, all I²C requests are directed towards a single device address, which corresponds to the I²C transceiver programmed into the hub firmware. Instead of performing traditional I²C transactions, the master device in the host and the slave in the hub make alternative use of the register address and data parts of a transaction to send a series of commands to perform bus writes or reads, as shown in Figure 4.22. Using those fields, full information such as target device id and register address can be sent to initiate a transaction. Since bus transactions are not atomic, but can include waiting cycles until they are completed, a status polling mechanism is integrated into the encapsulation protocol. While flexible and able to address all the device map, the downside of this method is speed, since it adds a considerable overhead over the already slow I²C protocol.

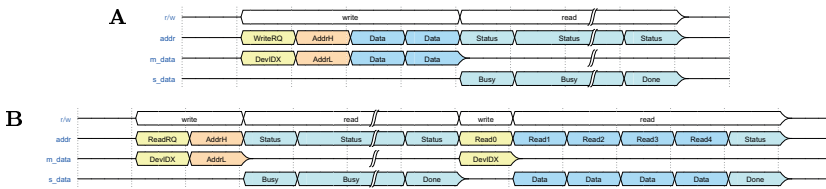


Figure 4.22: Simplified chronograms of the bus over I²C. **A:** Write operation. **B:** Read operation.

4.3.5 Acquisition performance

Performance of any acquisition system is measured in bandwidth, indicating the amount of data it can acquire per second, and latency, which measures the time between an actual event and the ability to act on it.

The maximum bandwidth allowed by the x4 PCIe Gen2 interface is 2GB/s. However, the ONIX system operates under a 250MHz clock, with 16-bit for device data and 32-bit for the main, aggregated bus. This yields a maximum theoretical bandwidth of 500MHz per device, or 1GHz total.

Data transfer is not realized continuously, but in blocks, with each individual transfer operation featuring a small overhead. As such, bigger block size increases actual bandwidth by reducing overheads, at the cost of increased data latency. Figure 4.23 shows measured bandwidth using a load-testing device integrated in the host firmware. With a 8KB block size the measurements are

close to saturate the 500MHz limit of a single device. With a block size of 16KB, the 1GB/s limit of the host system is achieved.

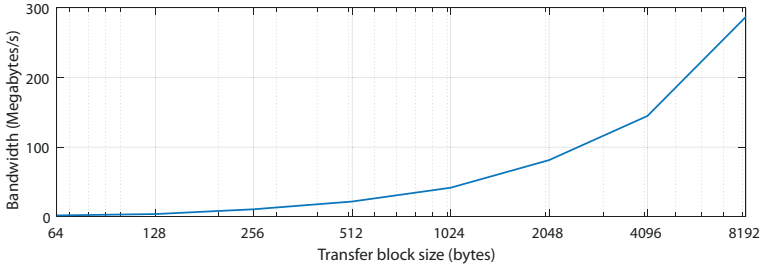


Figure 4.23: Measured bandwidth from data generated in a test device in relation with transfer block size.

While latency, measured in time, is related to block size, its actual impact is dependent on the data origin. This effect is more detrimental for devices generating small loads at high frequencies, which are rarely the case in neuroscience, as opposed to big sample sizes at the KHz range. For example, a 8192-byte block size would introduce a latency of 1024 samples on a simple, 8-byte timestamping device. However, for the case of 4 neuropixel probes, which a sample size of 480bytes each, a 8192-byte block transfer would imply a latency of less than 5 samples.

Although total latency is dependent on block size, there is a fixed, minimum latency associated to the processes of transfer initiation. For the ONIX system, transmission latency was measured with the smallest block size and a simple C program responding to a digital event. Under these conditions, $150\mu\text{s}$ of maximum transmission latency were measured.

4.3.6 3D Tracking

The headstages were able to track animals with millimetric precision. A 3D experimental environment, shown in Figure 4.24 was made, in which a mouse could move freely. The mouse was implanted with the 64-channel headstage connected through the active commutator. A 8-hour experiment was conducted, in which position and electrophysiology data was continuously acquired. Figure 4.25 shows accumulated occupancy heat maps at different points in time.

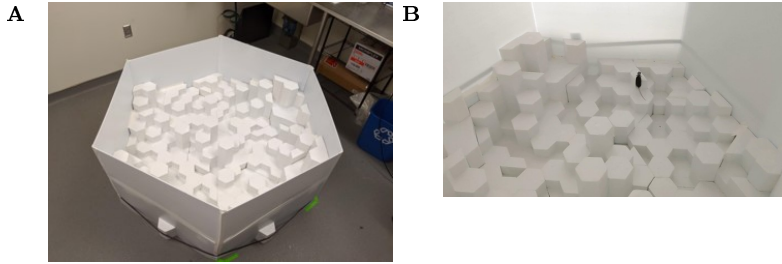


Figure 4.24: 3D environment created for experimentation. **A:** Full structure. **B:** mouse inside the environment.

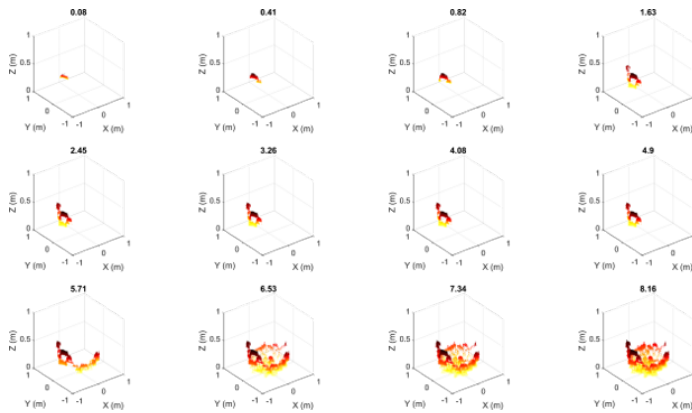


Figure 4.25: Occupancy heat map over 8 hours.

4.4 Conclusions

The ONI specification was designed to facilitate access to a diverse range of acquisition devices working asynchronously, independently of the data nature and the sampling speed. By defining a series of streams with a standardized frame structure, any ONI-Compliant hardware can communicate with ONI-based software no matter the kind of devices it exposes. Thanks to the libONI library, created to take care of the low-level communication interfaces, an application only has to define functions treating the specific device data it needs to handle.

An electrophysiology acquisition hardware based on the ONI specification, the ONIX system, was created. It was designed to improve upon the limitations

traditional electrophysiology hardware suffer in regards of animal movement and closed-loop experiments.

The system is composed of a host board and different headstages. The host is connected to the acquisition computer via the PCIe bus, allowing high bandwidth transfer with low latencies. The maximum bandwidth of the system is 1GB/s while minimal closed-loop latency has been measured as $150\mu s$. This allows its use for closed-loop experiments requiring sub-ms reaction, such as those based on firing spikes.

Headstages are connected through a single, thin coaxial cable, of less than 0.8mm in diameter, resulting in lightweight tethers. This interface has been designed to work with different device configurations. Two headstages were built, one featuring 64 channel neural acquisition and optical and electrical stimulation capabilities and another for Neuropixels probes, supporting up to two of them. Both headstages feature full 3D tracking using VR technology with sub-mm precision.

Thanks to the tracking capabilities of the headstage, an active commutator was developed, able to follow the movements of the experimental animals, completely removing torque, thus easing free movement in the animals.

Chapter 5

Wireless electrophysiology compression

Wireless systems are limited by energy requirements. Data bandwidth is directly related with transmission power, one of the biggest contributors to total energy consumption. This chapter presents a lossless digital compression algorithm for electrophysiological signals. It requires negligible extra power, reducing energy needs on wireless systems. The algorithm is designed to be device-agnostic and require little hardware resources, enabling its use in a wide variety of acquisition devices, with a specialized transmission protocol enabling data integrity in wireless systems susceptible to packet losses. A hardware prototype was created using this algorithm to demonstrate its effect on power consumption.

5.1 Introduction

Studies combining electrophysiology and behavior have provided insights on topics such as spatial navigation [65], [120], [121], memory formation [70], [122] and decision making [68], to cite some examples.

However, electrophysiological studies in behaving animals have been traditionally performed in well-controlled but severely constrained laboratory conditions, in relatively reduced size arenas or task apparatus and involving a limited, and often artificial (i.e. pressing a lever bar), repertoire of behaviors. Therefore, more natural and elaborated experimental conditions in ecologically meaningful contexts are required [91].

The need for open and meaningful spaces conflicts with the tethered nature of most electrophysiology systems, as the physical connection between the implanted electrodes and the recording equipment introduces mobility and distance restrictions. While this does not pose an issue for small spaces and simple maze topologies [31], [67], [69], [70], [121], [123] it entails difficulties for large arenas with enriched environments and social experiments with complex interactions [112].

While some approaches, like reducing cable weights or remove torque (see Chapter 4) can alleviate some issues, obstacles still remain. Wiring limits the maximum distance the experimental subjects are able to travel, make them unable to access enclosed areas such as burrows, can become tangled with environmental objects or be damaged by animal action. To solve these shortcomings, developments have been made towards wireless electrophysiology systems [93], [124].

5.1.1 *Wireless electrophysiology devices*

There exist two main approaches to wireless acquisition system: dataloggers and radio transmission. Both incorporate the same elements as a digital electrophysiology headstage, from the analog amplifiers to the ADCs, but differ in the treatment of the digital data.

Dataloggers are autonomous devices featuring a local, non-volatile storage medium, such as flash memory. Data is acquired and immediately stored. Data retrieval is done after the intended experiment has ended, by physically accessing the device and downloading the storage contents into a computer for its posterior analysis. Due to their full autonomous nature, they are specially suited for areas in which other approaches are not feasible, such as birds in

flight [125] or, by being possible to fully isolate the electronics, freely-moving fish [126].

The major disadvantage of dataloggers is its retrospective nature. Data can only be accessed after the experiment has run its course and cannot be monitored online. Any possible issue is only apparent after data retrieval, preventing any possible fix during the experiment itself. Moreover, the lack of online data makes it possible to perform any closed-loop experiment based on data acquired by the device.

Devices using radiofrequency links are able to transmit data to a remote receiver in real-time. Multiple methods exist for encoding and sending data through a radio stream. Analog neural data can be modulated, with multiple channels merged via time multiplexing, and sent over a carrier frequency [127], [128]. While an analog transmitter requires less energy [127], analog signals are more susceptible than digital signaling and the absence of an arbitration protocol prevents multiple devices sharing the same frequencies.

Digital transmission can be as simple as their analog counterpart, just modulating the ADC output instead of the raw analog signals [93], which alone adds noise resistance to the transmission. However, more complex protocols can be used that add synchronization and arbitration features, as well a bidirectional control [129]. Some examples of widespread digital protocols are Bluetooth [130], a low power protocol designed for data rates up to 2Mbit/s, Bluetooth Low-Energy [131], a slightly slower (up to 1.37Mbit/s) version with reduced power needs or WiFi 802.11b/g [132], a high-speed protocol with data rates of up to 54Mbit/s and advanced arbitration capabilities, but with higher power requirements. Some projects have developed a custom protocol, being able to fine-tune the power-performance trade-off [133].

Power is the main bottleneck of wireless devices, limiting data rates and device operating life. Batteries are the most common power source, followed by radiofrequency power transmission [134], [135]. Other alternative sources have been researched [136], although the energy they can provide is far lower. Some examples are deriving power from light [137], [138], using kinetic energy from the animal movement [139], body heat [140], or even chemically from the blood [141].

Lowering power consumption allows for longer operational time, increased data rates and reduced battery weights. As such, minimizing power requirements is a goal for every wireless device. In the case of radiofrequency systems, the power bottleneck is derived by the power needs of high-bandwidth data trans-

mission [130]. Different approaches exist to reduce energy consumption in these devices. For example, developing the core hardware as a custom-made ASIC can help by developing electronics highly optimized for the task [93], [127] at the expense of increased development and production costs. Another line of improvement, since the bulk of power requirements stem from radiofrequency transmission, is the development of specialized protocols which can yield improvements over generalist, commercial ones [142]. Research is also being made in fields like antenna optimization [143], [144], to further reduce power needs of radiofrequency signals, as well as optimize wireless power transmission.

A different approach, compatible with the previous ones, and applicable to both data loggers and radiofrequency transmission, is to reduce the bandwidth needs of the data. A neural recording including fast activity transients, like spikes, require a sampling rate of at least $20KS/s$ [27], [145]. This combined with multichannel acquisition, typical on modern high-density electrophysiology recordings, results in bandwidths of tens of megabits per second [144]. Compression techniques can be used to reduce bandwidth needs which, in turn, decreases the power consumption of the wireless transmitter.

5.1.2 Data compression methods

Data compression algorithms decrease the size of a dataset by finding redundant components of the signals in a particular domain and removing them. All compression method fall into one of two categories: lossless or lossy algorithms [146]. The former produce, once decompressed, a signal identical to the one being compressed. Lossy methods eliminates not only pure redundant data but other components as well. As such, compression rate is higher than their lossless counterparts but introduce distortions in the signal. As long as the errors introduced by a particular method are below the margin desired by the application, lossy methods are a powerful alternative.

Basic compression methods, which then an algorithm can mix, are based on the following principles [146]:

- **Domain transformation:** By transposing the signal from the time domain to a transformed domain through operations such as the Discrete Cosine Transform (DCT) or Fast Fourier Transform (FFT), components in this domain might be sparser, with zero elements that can be removed, thus decreasing signal size. If only zero components are eliminated, then the compression is lossless. However, small components can also be re-

moved, introducing small errors in the signal, for a lossy method with better compression ratio. [147], [148].

- **Run-length encoding:** This is a lossless method that works by replacing repetitions of a symbol, or a sequence of them, by a single occurrence followed by the number of repetitions. Its performance is dependent on the sequence redundancy of the dataset [149].
- **Entropy coding:** Another lossless method that uses variable bit coding to encode different symbols. These methods require sets in which some symbols appear more frequently than others. In this case, symbols with higher probability can be coded with less bits than the less frequent ones, resulting in reduced overall data size. [150]
- **Compressed sensing:** Although sharing similarities to domain transformations, compressed sensing is a recent lossy method based on sampling a signal below the Nyquist frequency, thus reducing the data size [151]. To do so, the signal must be sparse in some transformed domain and be sampled at an irregular interval, incoherent with the signal itself. Data reconstruction involves a complex mathematical operation. [152], [153].

Compressed sensing has gained popularity in recent years thanks to its asymmetric nature with the computational bulk moved to the decompression stage, resulting in a negligible amount of complexity and power consumption on compression. This makes compressed sensing a very good candidate for wireless acquisition devices [154]–[156]. However, its lossy nature limits the range of experiments in which it might be applicable. Moreover, both its compression efficiency and signal distortion is affected by acquisition noise [157].

Domain transform algorithms, such as wavelet compression [113] can yield excellent compression ratios, but require circuitry capable of handling advanced mathematical operations. This translates in higher power needs, thus defeating the usefulness of compression as a power-saving technique. On the other hand, algorithms with lesser computational requirements are often used on only a particular part of the signal spectrum. For example, lower-frequency LFPs tend to have high inter-channel redundancy, making high compression ratios with simple techniques possible [158]. High-frequency spikes, in contrast, are sparse events, so it is possible to use spike-detection algorithms and only perform compression for the discrete, individual events [154]. Both techniques can be combined, compressing and sending both LFPs and spikes separately by the same device [156]. These approaches, however, are not able to provide a complete, continuous view of the entire acquired signal.

5.1.3 Objectives

This chapter describes a compression algorithm for brain electrophysiology , able to reduce bandwidth and, by extension, transmission power requirements, along with a novel hardware implementation.

Its main requirements are:

- Has to be functionally lossless. Noise is permitted as long as it is below the natural noise margin of the acquisition device.
- Must not require specialized circuitry. Hardware requirements and power usage must be kept at a minimum.
- Must be device-agnostic. It must not be tied to a specific device or technology.
- Its compression ratio on electrophysiology data must be enough to cause a noticeable decrease in transmission-related power consumption in the case of wireless transmission.
- Has to be flexible, adaptable to multiple sample rates and channel counts.

In addition to the algorithm, a hardware prototype implementing it was developed and built to test compression and power performance.

5.2 Materials

This section describes materials used in both the research and design of the compression algorithm as well as in the development and construction of the hardware prototype. Figures 5.8 and 5.11 of the Results section can be used as reference of the usage of each item described in this section.

5.2.1 Huffman Coding

Huffman coding is an entropy-based method to encode information devised in 1952 by David Huffman [159]. The basic precept behind Huffman coding is that, in any set of symbols, the appearance frequency of every different symbol might not be the same. Under that case, a usual fixed-length coding is highly redundant, as defined by Shannon's theorems [160] and, thus, not optimal. Huffman coding, instead, codes each symbol with a different bit

length, depending on their appearance frequency. This way, symbols that appear more frequently are coded with fewer bits than less frequent symbols, reducing the overall bit size of the set. A natural consequence of this encoding is that compression rates are higher when symbols follow a steep distribution, i.e., a few subset symbols conform the majority of the set, while flat symbol distributions result in low compression rates.

The unique relationship between each symbol in a dataset and its variable-length code is referred as a dictionary. For any given set of symbol, an optimal dictionary exists that minimizes the size of the coded set. Huffman coding proposes an algorithm to create such a dictionary in the form of a binary tree, with each symbol in its end points, or leafs, and each *left* or *right* branch representing a binary 0 or 1. The algorithm constructs each node to create the full tree in the following way:

1. Create a list containing one node for each symbol, including its appearance probability.
2. Sort the list based on the nodes probability field of each node.
3. Retrieve and remove from the list the two nodes with lower probability.
4. Create an intermediate node with. Connect the two retrieved nodes as children, left branch being the node with lower probability.
5. Update the probability of the intermediate node to be the sum of the probabilities of its two children nodes.
6. Insert the new node into the list.
7. As long as there are two or more nodes, repeat from step 2.

Figure 5.1 shows an example of a dictionary created from a simple dataset with 2-bit symbols, including the binary tree representation. For this example, a dataset of size L_{dset} would take $2L_{dset}$ bits of memory space, while the Huffman-coded version, with each symbol having a probability P_{symbol} and a coded width of W_{symbol} would take $L_{dset} \sum P_{symbol} W_{symbol} = 1.7L_{dset}$.

The binary tree format makes it trivial for decoding Huffman-coded data into their original symbols, as it is only a matter of traversing the tree depending on each received bit until reaching a leaf node. Coding data, which involves backwards-traversing the tree adds a level of difficulty, involving either searching the full tree or creating a parallel index. The latest option being the fastest.

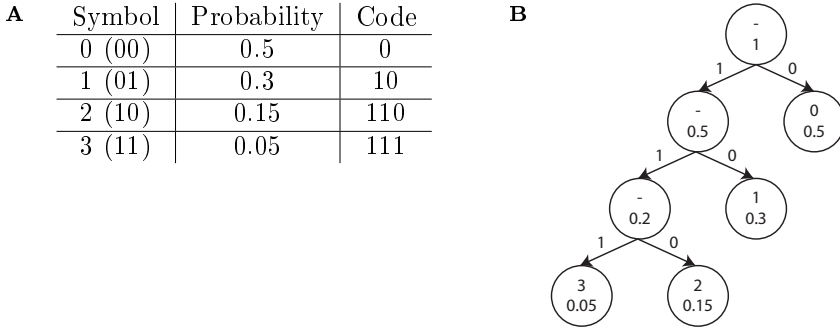


Figure 5.1: Dictionary creation example. **A:** fixed 2-bit symbols to compress, their probabilities and the resulting variable-length code. **B:** Binary tree representation. Top value in each node represents a target symbol, or dash for intermediate node. Bottom value represents node probability. Bit values on arrows represent code creation.

Creating an optimal dictionary requires previous knowledge of the full dataset in order to compute the appearance probabilities of each symbol. This is not possible for streaming data that must be encoded in real time. In this case, a dictionary can be created with a sample data set approximating the symbol distribution expected in the online data, with better results the closer this approximation is to the actual symbol appearance rate. In this case the optimal size ratio provided by the dictionary represents an average result.

Due to the variable nature of Huffman coding, dictionary size for encoding, including a quick search index, can vary depending on the datasets and the resulting codes, with a maximum possible size of 2^{2n} bits, for a collection of n bits symbols [161]. Different algorithms exist to reduce this size. Some force a maximum code length [162] which reduces the algorithm efficiency. Others work by rearranging the dictionary after it has been created [161], which requires it to be known in advance to measure the memory needed.

The algorithm designed on this work is based on [163], which minimizes dictionary size while ensuring only the word width of the dataset has effect on said size, and without limiting code length. Using this algorithm, a collection of n bit symbols requires 2^{n+1} bits of memory, as shown in Figure 5.2

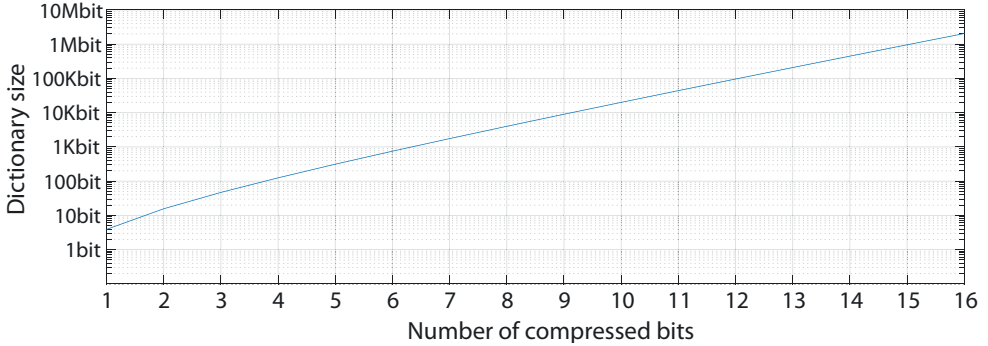


Figure 5.2: Dictionary size needed by the selected variation of the Huffman algorithm [163] for datasets of word lengths of 1 to 16 bit

5.2.2 *Delta compression*

Delta compression, or delta encoding, is a very simple method in which each symbol is represented with the difference with the preceding one, i.e., $y_i = x_i - x_{i-1}$. Decoding can be done by cumulative addition of the received values $x_i = x_{i-1} + y_i = \sum_{n=0}^i y_n$.

This coding is specially suitable for signals that follow a smooth progression, with high-frequency components having low amplitude, such as those of biological origin [164]. For signals with these characteristics, the resulting difference vector $y = \Delta x$ is composed of a majority of low values which can be encoded with fewer bits, as opposed to the more even symbol distribution that the raw signals have.

5.2.3 *Low-power FPGA*

An ultra-low-power FPGA with a small footprint was used. The selected device is the AGLN250 IGLOO nano FPGA (Microsemi, Alto Viejo, CA, USA) [165].

The IGLOO family of FPGAs differ from other devices in the way their configuration is loaded. The bitfile is stored in an internal FLASH memory, similar to other devices such as the MAX10. The difference lies in that these devices have storage flash memory as a single block. As any regular FPGA, they need to load the bitfile from it into its internal registers and use power to keep its configuration. IGLOO devices, however, feature their flash distributed through the chip, being an integral part of its logic elements. As such, there is no need

to initialize the FPGA or maintain a loaded configuration, as the bitfile storage itself is integral part of the configuration logic.

IGLOO devices are designed for low power in mind, with the nano subfamily being the ones with lower power and smallest footprints. The trade-off is reduced availability of hardware resources, far inferior than other commercial FPGAs. IGLOO nano devices lack advanced hardware resources, such a differential I/Os, DSP or multiplier units. Logic resources are also reduced, with the selected AGLN250 device featuring 3000 IGLOO Logic Elements and 36Kbit of RAM.

Two different packagings were considered for this project. The AGLN250 VQG100 with a 14x14mm footprint with a 0.5mm-pitch Quad Flat Pack format, i.e.: external pins in all four sides of the die, and 68 I/O pins was used on the prototype. A smaller package, AGLN250 CSG81 of 5x5mm is available, with 60 I/O pins and a 0.5mm-pitch Ball Grid Array connection format. This package requires high-density PCBs with small trace separations and blind microvias, which make it unsuitable for prototyping. Another difference considered is that the VQG100 variant features a very low power, optimized internal configurable clock generation circuit, while the CSG81 would either be locked to the main clock or require an external configurable generator which adds another IC to the footprint.

5.2.4 *Wireless processor*

Wireless communication in the hardware prototype was performed through the WiFi IEEE 802.11g protocol. Although several studies have demonstrated how custom protocols can offer very efficient wireless transmission [133], [142] the use of a standard, widely available protocol allows for an easy way of testing the efficiency of the compression algorithm without adding any external bandwidth constraints. Moreover, the wide availability of commercial devices for both transmission and reception as well as the interference protection features of the protocol make it a perfect candidate for prototype building.

The major downside of the protocol in the context of this work is that it is not natively designed for low power applications. However, commercial devices exist that reduce the power requirements to a minimum and are able to operate on batteries. While the 802.11g protocol has little provision for reducing its power levels in relation to the required bandwidth, these devices can send data in bursts at full speed and power down the transmitter circuitry when not in

use. This means that a reduced data rate, as achieved by compression, still translates as lower power usage even with a non-optimal protocol.

A transceiver device with integrated network processor was used, specifically the CC3320SF IC (Texas Instruments, Dallas, TX, USA) [166]. This device is a System-on-Chip (SoC) embedding:

- A wireless transceiver, with all the required analog circuitry.
- An ARM-based network processor independently driving WiFi operation and in charge of data transmission.
- An ARM Cortex-M4 MCU for custom application programming, featuring 256KB of RAM and up to 27 GPIO pins, some of which can act as dedicated transceivers for buses such as I²C or SPI.
- 1MB of flash memory for the application program

The Cortex-M4 MCU can be programmed in C using Texas Instruments Code Composer Studio. Access to the network processor is done through an API denominated SimpleLink, which offers functions for all required network operations. The MCU features multiple DMA modules that are able to transfer data quickly from and to the different data buses and the network processor.

The wireless circuitry required an external antenna tuned to the specific parameters of IEEE 802.11g transmission. Texas Instruments provides, however, an assembled module with a built-in PCB planar antenna, denominated CC3320MODASF, which is the final device used in the project.

5.2.5 *Sample signals and acquisition hardware*

The compression algorithm implementation was designed to be optimized, both in performance and resource efficiency, for brain electrophysiology signals. Since different devices have unique properties that can affect the resulting Huffman dictionary, it is important that the same device is used for dictionary creation and compression.

The device used during development is the Intan Technologies RHD2132 acquisition chip, which offers a maximum rate of 30KS/s for 32 channels, with a 16-bit output of 0.195 μ V resolution, 2.4 μ V_{rms} input noise and a \pm 5mV input range. These parameters affect algorithm tuning and Huffman dictionary

but not the algorithm design itself, so it can be tuned for other devices with different characteristics.

Sample data sets, acquired with the RHD IC, were used for different development and design stages. In particular, 10 minute recordings from the hippocampus of two different rats and a 5 minute recording from the visual cortex of a third rat were provided by the Alicante Neuroscience Institute (San Jose de Alicante, Alicante, Spain) and the Open Ephys project (Cambridge, MA, USA), respectively. These sample datasets were used for the initial measurements and design testing. A base dictionary was created from these signals as well, after being processed by the modified algorithm presented in this work. (section 5.4.1 and 5.4.2).

To test algorithm performance on signals not related to dictionary creation, data from two sets of animals was used. Five minute recordings from the retrosplenial cortex of five different mice were provided by Jakob Voigts, from Harnett lab, at Massachusetts Institute of Technology (Cambridge, MA, USA). Canals lab, at Alicante Neuroscience Institute, provided data from an independent experiment involving three long-evans rats implanted in the hippocampal region. From these animals, 30 minutes of data was recorded daily for 4 consecutive days. Both these datasets were processed offline by the algorithm to measure compression ratios.

5.2.6 Development hardware and software

5.2.6.1 Hardware

Development boards for both the wireless module and the FPGA were used. These boards feature a version of the target device built with all the electronic components needed for them to work. These boards also contain a variety of switches or jumpers to select different operation modes, monitor power consumption or control optional features as well as connection pins to expose the GPIO ports present in the device. Power is provided through USB connectors and a programmer is embedded on the boards for ease of development. In particular the CC32200SF-LAUNCHXL development kit (Figure 5.3.A) was used for the network processor while a AGLN-NANO-KIT (Figure 5.3.B)) was used for the FPGA.

For early development and testing stages, the designs were implemented on the Open Ephys acquisition hardware. This device features a mid-range Spartan-6 FPGA (Xilinx, San Jose, CA, USA) and the RHD2132 acquisition chip. The

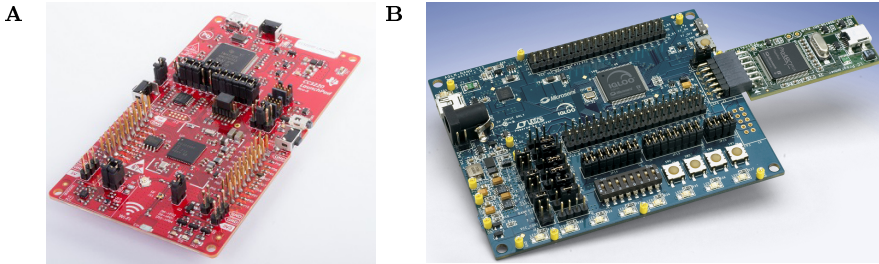


Figure 5.3: Development boards. **A:** CC3220SF-LAUNCHXL. From Texas Instruments (<https://www.ti.com/tool/CC3220SF-LAUNCHXL>). **B:** AGLN-NANO-KIT. From Microsemi (<https://www.microsemi.com/existing-parts/parts/144014>).

configuration file normally running on the FPGA was replaced by a custom version including the compression algorithm, running in the same manner it would run in an IGLOO device, thus taking advantage of the existing acquisition and communication hardware present on the Open Ephys board.

5.2.6.2 Software

- **MATLAB** (The MathWorks Inc. Massachusetts, USA) is a mathematical software featuring its own programming language. In addition of offering the possibility of writing custom functions, it features built-in for common processing and filtering operations, offering good performance in treatment of big data matrices. This software was used for researching the mathematical aspects of the algorithm prior to its implementation, as well as all the result analysis.
- **Visual Studio** (Microsoft, Redmond, WA, USA) is a IDE for C/C++ and .NET applications. It was used for the creation of a software model.
- **Libero SoC** (Microsemi, Alto Viejo, CA, USA) is the design suite for IGLOO FPGAs, required to create the bitfiles for the devices from HDL code such as VHDL or Verilog and upload it into the IGLOO embedded flash memory. Was used to develop the HDL modules and to program the prototype.
- **ModelSim** (Siemens EDA, Plano, TX, USA) is a HDL simulation software able to inspect designs in detail. It was used to test the developed modules, debug them and compare the results to the software model.

- **Xilinx ISE** (Xilinx, San José, CA, USA) is the design suite for Spartan-6 FPGAs, required to create the bitfiles to configure these devices. Was used to build the HDL design in a way compatible with the Open Ephys System.
- **EAGLE** (Autodesk, San Rafael, CA, USA) is a electronic design suite including schematic and PCB design. It was used to develop the hardware prototype.
- **Open Ephys GUI** is the graphical interface of the Open Ephys acquisition system. It was used in the *in vivo* experiments to acquire data, show it on screen and record it to disk. Two plugins were created for it able to receive compressed data streams and uncompress them in real time: one that drove the Open Ephys board modified to perform acquisition and compression and another able to receive compressed neural data through a network interface, for use with the wireless prototype.

5.3 Methods

5.3.1 Software model

The first design step was done in MATLAB, in which a mathematical model of the algorithm was tested to verify its viability, using standard functions built in into the software.

Once the algorithm was deemed viable, a software model in C++ was developed using Visual Studio. This model was created programming the exact same algorithm that would be later implemented in hardware and was used as a reference for the latter. A collection of software utilities were created:

- A program to read a collection of data and create a modified Huffman dictionary from it, following the requirements of the compression algorithm.
- A compression program, able to read a multichannel dataset and compress it. The compressed file it creates has a identical format as a compressed data stream generated by the hardware.
- A decompression program, able to read a compressed data file and reconstruct the original multichannel file.

- A decompression program, similar to the previous one, but handling framed data blocks (as described on section 5.4.5) instead of a raw compressed stream
- Created during the hardware development phase: A utility to convert the Huffman dictionary into a format fit for the FPGA design.

Using this model, the base dictionary using the sample datasets was created. The model allowed for quickly adjustment of some parameters in the compression algorithm to optimize performance. Since the compressed data created by the hardware algorithm is identical as the one produced by the software model, it was used to perform performance measurements on offline data.

5.3.2 Hardware design and validation

A full hardware design, able to acquire from the RHD IC, compress the data and send it to the wireless processor was created in Verilog Hardware Description Language (HDL). Using simulation software, signals from the sample datasets were processed by this implementation, the resulting compressed data saved to a file and compared with the software model output, the design being valid when both results matched.

A signal generator was also developed to be able to test the design on actual hardware. Due to the limited resources available on the target FPGA, generated signals could only follow very simple patterns. The signal is a repeating combination of two arithmetic series (Figure 5.4)

$$\begin{aligned}
 a_{i+1} &= a_i + i, \quad a_0 = 0 \\
 b_{i+1} &= b_i - i, \quad b_0 = a_{2^8-1} \\
 i &\in [0, 2^8 - 1]
 \end{aligned}
 \tag{5.1}$$

with an offset in both value and time that could be different per channel. While the flat distribution of post-delta values on this signal is suboptimal for Huffman compression performance (see section 5.4.1), it allows to test the algorithm with a rich range of values.

Results with this signal were generated with the software model, the simulated implementation and in the final hardware prototype, which allowed to verify the different implementations.

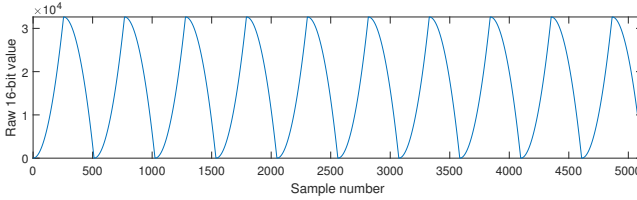


Figure 5.4: Generated synthetic signal for testing

5.3.3 *In Vivo* testing

Online compression with the HDL implementation of the algorithm was tested *in vivo* with animals provided by the Alicante Institute of Neurosciences, where the experiments were performed. The Open Ephys hardware was used for this, which allowed for a full test of the algorithm in similar hardware before building the final prototype. The device usual FPGA firmware was completely replaced by one created from the HDL modules developed for this project in their final low-resource, low-power implementation, including acquiring from the RHD chip and compressing the signal.

20-minute recordings were obtained from three different animals using this method, with the software doing online decompression of the signal. Both the compressed and the decompressed signals were stored to disk for comparison.

5.4 Results

5.4.1 *Compression algorithm*

Neural compression is achieved in this work by the combination of both delta and Huffman encoding. Huffman compression efficiency depends on the dataset featuring few symbols with much higher appearance probability than others. While neural signals are not optimal for Huffman compression in raw form, this can be improved by the derivative transformation caused by delta encoding.

Figure 5.5 shows how delta compression can optimize a electrophysiological recordings for its use with Huffman encoding. It can be seen how, after performing delta compression, the distribution becomes steep, with low value symbols being orders of magnitude more frequent than higher value ones. By using delta encoding on a neural signal it becomes optimal for further compress-

sion using the Huffman method. The Huffman dictionary is thus elaborated from delta-compressed values.

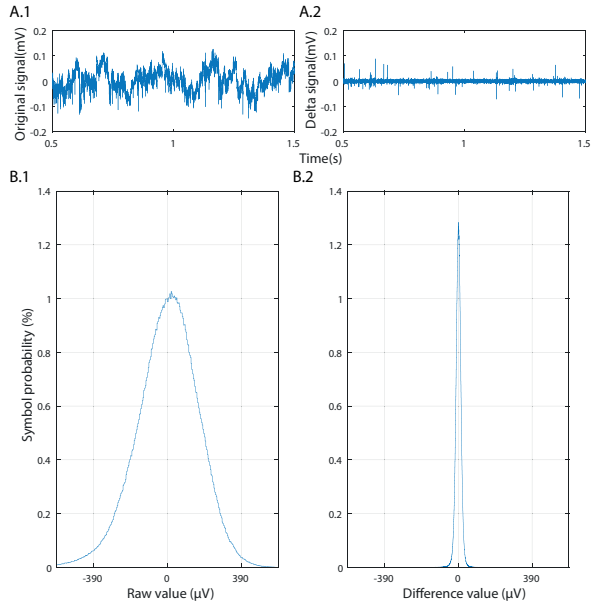


Figure 5.5: Comparison between a 20 minute neural signal in raw form (A.1, B.1) and after delta-encoding (A.2, B.2). **A:** 1 second time-domain sample of the raw signal (A.1) and the delta-encoded signal (A.2). **B:** Symbol probability of the full recording in raw form (B.1) and after delta-encoding (B.2).

While methods exist that take advantage of the usual spatial proximity of recording sites [145], this work treats each channel of the multichannel recording separately to achieve the maximum possible compression while being tolerant to as many electrode configurations as possible.

5.4.2 Low-memory, Low-resource compression

While neither delta compression or Huffman coding requires specialized DSP or multiplier circuitry, which would limit the range of low-power devices they could be implemented on, Huffman coding requires Read-Only Memory (ROM) memory storing the symbol dictionary. The algorithm version used in this work is already designed to minimize memory needs [163]. However, as seen in Figure 5.2, for 16-bit words, which are typical in neural acquisition [27], this results in 2Mbit of memory, limiting the available devices able to run

this algorithm. A number of ways were devised to reduce the word width and thus the dictionary size, while minimizing the impact on compression ratio and signal integrity.

Delta encoding, performing a binary subtraction, already trims one bit, leaving 15-bit words to be compressed. To further reduce the amount of memory needed by the Huffman dictionaries, which are determined by bit number, not all bits are coded using that process. Huffman algorithm efficiency relies on the appearance probability of a small subset of symbols being higher than the rest. However, as evidenced by Table 5.1, not all bits of a delta-compressed signal follow the same distribution, with only higher bits contributing to the steepness of the distribution, as shown in Figure 5.6.

Bit	0	1
0	50%	50%
1	50.72%	49.28%
2	52.19%	47.81%
3	55.15%	44.85%
4	61.36%	38.64%
5	77.18%	22.82%
6	95.51%	3.49%
7	99.81%	0.19%
8	99.99%	0.01%
9	~100%	<0.01%
Sign	49.76%	50.24%

Table 5.1: Frequency of each bit of a delta-coded sample signal having a value of '0' or '1', when taken as an absolute value. Bits 9 to 14 are not shown as the probability of them being '1' is exponentially reduced each step. For the Signal bit only nonzero values were counted, with '0' meaning a positive signal and '1' negative.

It is possible then to only use Huffman encoding on the higher bits, while appending the lower bits of the delta-compressed signal without further processing. Figure 5.7 shows how compression efficiency is affected by this approach. Moreover, the probability distribution is symmetrical, which allows to create a Huffman dictionary for only absolute values and appending the sign bit unprocessed as well.

Two extra steps are incorporated into the algorithm to improve the compression ratio even further. Since the sign bit has only relevance for nonzero values, it is not transmitted when the decoded value is zero. Additionally, although ADC circuits usually have 16bit outputs, the conversion process produces a lower

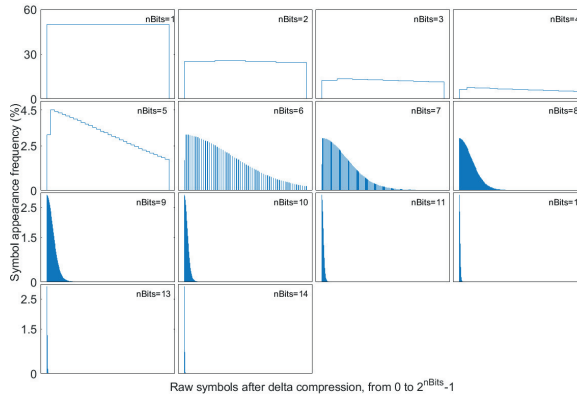


Figure 5.6: Symbol distribution of a delta-coded sample signal, in absolute value, for different amount of masked bits. The lower $nBits$ are kept and the other discarded before plotting the probability distributions. The X-axis of each plot are the different symbols, from 0 to 2^{nBits} .

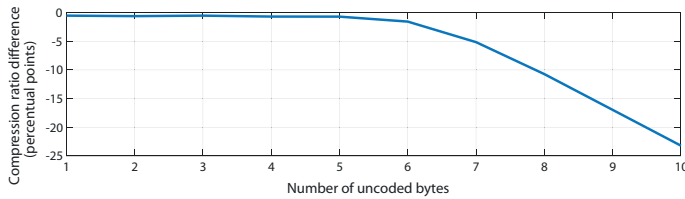


Figure 5.7: Degradation, in percentual points, of the compression efficiency when different amount of bits are transmitted without being coded by the Huffman algorithm

number of relevant bits, with the less significant bits being electrical noise. Those can safely be omitted, as they contain no useful data by design. In the case of the RHD chip used in this work, with $0.195\mu V$ resolution and $2.4\mu V_{rms}$, it is possible to calculate that the output contains $\log_2(2.4/0.195) = 3.6$ bits of noise. Thus, the 3 lower bits can be completely discarded instead of being sent uncompressed. For other acquisition devices, the number of discarded bits can be adjusted so they are always below the input noise level, resulting in small differences on compression ratio.

A complete block diagram of the algorithm can be seen in Figure 5.8. The Huffman dictionary needed for the algorithm is, thus, made from the sample dataset after it has been treated by this process, considering only the bits that are to be compressed by the Huffman method.

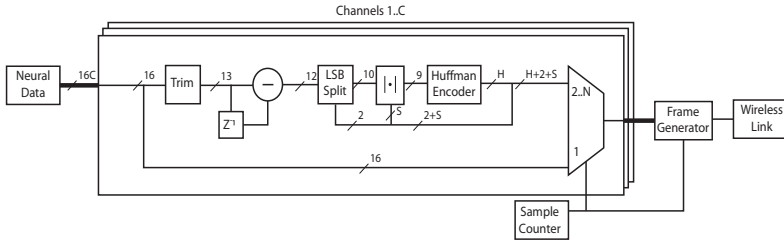


Figure 5.8: Block diagram of the complete in-system algorithm, detailing the compression algorithm. H stands for the variable bit count of a Huffman-coded word, while S can be 1 bit for sign coding, or 0 bits for 0-value words.

5.4.3 Compression performance

5.4.3.1 Compression ratios

A Huffman dictionary provides the optimal compression ratios for the data used to create it. However, actual experimental situations require the algorithm compressing a stream of animal electrophysiology data not known in advance with a dictionary made with a sample dataset. Thus, to test for algorithm performance, two sets of unrelated data were used. A sample dictionary was created from 25 minutes of data, originating from three different animals. A second dataset of 385 minutes of available offline data, from eight animals unrelated to the ones used for dictionary creation, was processed through the software model, with a resulting average ratio of 47.94% of the original signal size.

In vivo real-time compression, using the hardware implementation of the algorithm, yielded a mean ratio of 65.58%. It is worth noting that one of the three experimental animals, which will be called *Rat 3* from now on, had a uncommonly high amount of acquisition artifacts. As a result, the performance of the algorithm was slightly affected. Removing the data from this animal results in a mean compression ratio of 62.64%.

Table 5.2 shows the detailed ratios for each of the animals and setups.

Data	Compression ratio	recording time
Offline compression		
Mouse 1	33.86%	5 min.
Mouse 2	33.72%	5 min.
Mouse 3	33.37%	5 min.
Mouse 4	33.25%	5 min.
Mouse 5	38.44%	5 min.
Rat 1	51.37%	30 min. x 4 days
Rat 2	44.60%	30 min. x 4 days
Rat 3	59.31%	30 min. x 4 days
Mean	47.94%	Weighted Average
<i>In vivo</i> online compression		
Rat 1	62.99%	20 min.
Rat 2	62.29%	20 min.
Rat 3	71.45%	20 min.
Mean	65.58%	Average

Table 5.2: Compression ratios, in percentage of the original signal size, for the different datasets not related to dictionary creation.

5.4.3.2 Signal integrity

The combination of delta compression and Huffman coding in their original forms is completely lossless, introducing no alteration to the input signal in the process of compression and decompression. However, the implementation described in this work (section 5.4.2) alters the input signal by removing the trailing bits of the input signals, corresponding to the input noise of the acquisition circuit.

The only effect this procedure has on signal integrity is a distortion below the noise floor of the acquisition chip itself, thus not affecting the actual acquired data. Figure 5.9 show a comparison between an original and a processed signal. The measured error is $0.21\mu V_{rms}$, while the maximum possible error introduced by the current implementation of the compression algorithm is $1.56\mu V_{rms}$, all below the $2.4\mu V_{rms}$ noise of the neural acquisition chip itself.

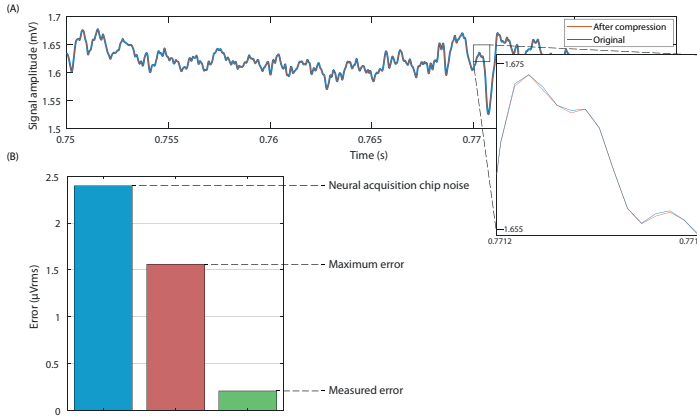


Figure 5.9: Effect of compression on signal integrity. **A:** Compressed and original signals. The error is indistinguishable without magnification. **B:** Error introduced by the algorithm compared with the acquisition chip noise floor.

5.4.4 Effect of dictionary on compression

Since the Huffman dictionary was created with a sample set of signals, it was of interest to ascertain whether creating dictionaries using some datasets from the same animals in the experiment provided any performance variation. This was tested using the 4-day dataset. Dictionaries were made from the data acquired during the first day. The whole set was then compressed offline using the base dictionary, the dictionary made from each of the rats and combinations of these dictionaries with the original.

Table 5.3 shows the obtained compression ratio for different combinations. A slight improvement from the base results can be observed when using dictionaries including data from the experimental dataset. As expected, removing the data from the noisy *Rat 3* improves the results.

5.4.5 Transmission protocol

When using the algorithm in a hardware system, the device running neural data compression might be different from the one performing wireless transmission. In addition, many low-power wireless protocols lack mechanisms to ensure reception, thus being susceptible to packet loss [132]. This is especially problematic for delta coding, since each lost incremental value introduces a permanent error in the signal which increases with each consecutive missed

		Animal data	Animal + base
Base	w. Rat3	51.87%	N/A
	w/o Rat3	48.15%	N/A
All	w. Rat3	50.76%	50.59%
	w/o Rat3	47.42%	47.96%
Self	w. Rat3	48.8%	49.42%
	w/o Rat3	46.42%	47.44%
Others	w. Rat3	49.37%	51.16%
	w/o Rat3	47.67%	46.39%

Table 5.3: Mean sizes of the compressed signals relative to the original data for *in vivo* tests. Columns for dictionaries using experimental data alone or added to the base dictionary. Rows for base dictionary, dictionary from data from all the animals (All), dictionaries from data from each individual rat (Self) or dictionaries from data from all animals except the one being tested (Others). Data shown including and excluding the anomalous rat labeled "Rat3"

value. A protocol was designed to transmit data between devices, including information enabling the wireless processor to pack the data in a way able to recover from packet losses. This protocol utilizes few hardware resources, has no RAM requirements and adds a low overhead to the transmission, maintaining the reduced bitrate achieved by the compression.

To recover from the errors introduced by packet losses on delta coding, data is packed in blocks of N samples, with the first sample for each channel being the raw, uncompressed values followed by the compressed remaining samples. Since this results in a variable number of bits, due to the compression method, the last word is zero-filled after the last sample data to ensure an integer number of 16-bit words per block. If network packet aligns with block boundaries, in the case of a packet loss the receiver could recover at the start of the next block. Huffman coding introduces an additional challenge due to the resulting samples having a variable bit length. The network processor must be able to track of block boundaries, independently of their actual byte size.

Data from the compressing device to the wireless processor is packed in fixed-length frames of M words, with the current implementation using 16-bit words. The first $M-1$ words are comprised of block data. The last word is a block boundary indicator. If the frame contains the start of a new block, the indicator is an index pointing to the specific word of the frame in which the new block starts. If the whole frame contains data from the same block, the indicator is a value greater than M . This transmission protocol allows the wireless

processor to be fully aware of delta-coded block boundaries without requiring the compressor device to store them in memory. With this information, the wireless protocol can add simple indexed markers to the data packets, allowing the receiver to detect a packet loss and to wait for the next delta-coded block to start.

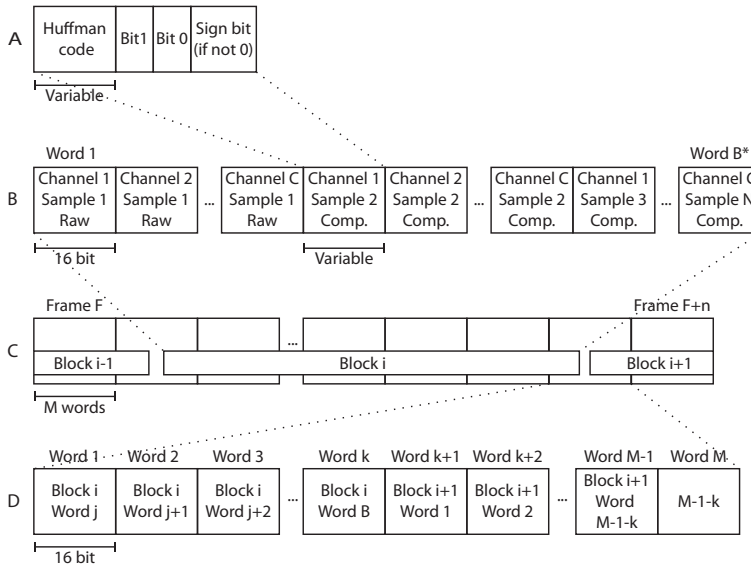


Figure 5.10: Data transmission structure. **A:** Compressed sample. **B:** Compressed block of N samples for C channels. Total size of B word can vary depending on compression. **C:** A block spans several frames, while a single frame can include the boundary between two blocks, **D:** Frame of M words sent to the transmitter, with index to detect block boundaries.

Figure 5.10 shows the complete structure of compressed data packed for transmission, including the composition of a compressed sample (section 5.4.2).

5.4.6 Wireless prototype

A prototype was created, as described in Figure 5.11, integrating acquisition, compression and wireless transmission. While a smaller version, using compact packages of the ICs was designed, the produced unit featured bigger versions, for ease of prototyping, manufacture and testing, as well as several headers for programming and debugging.

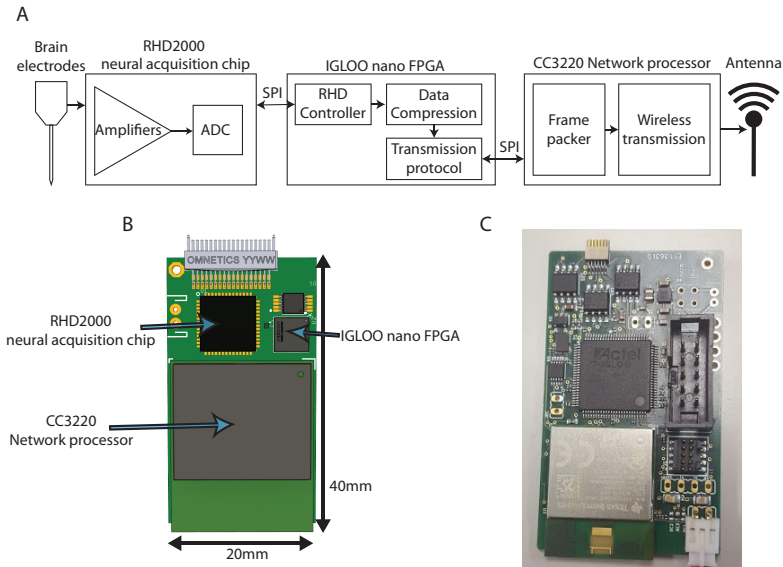


Figure 5.11: **A:** Functional diagram of the developed prototype. **B:** Sketch of a complete device. **C:** Picture of the built prototype. For development purposes, the layout differs from what would be a finished unit, including the addition of debug headers and the use of bigger versions of the integrated circuits.

Beyond the storage required for its configuration, the IGLOO FPGA only features a few bits of flash memory, not enough for Huffman dictionary data. Since the CC3220 wireless processor features 1MB of persistent memory it can be used to store this data, along with the MCU program. The software then uploads this data to the FPGA at startup, along with the command list required to configure the RHD2132 acquisition IC.

Communication between the IGLOO FPGA and the wireless MCU is done through a full-duplex SPI bus, transferring the data in blocks for best performance. Since compressed data is not produced at a constant rate, due to the variable bit width of the samples, the FPGA acts as the bus master, able to momentarily pause transmission between samples until a full word has been produced. Using the MCU as a master device would be possible only by having the FPGA store a buffer of an entire transfer block, which would require a big amount of memory, defeating the purpose of the low-resource, low-memory approach of the compression system.

The SPI transceiver in the MCU, however, can not be continuously active. It has to be configured to perform a fixed-size receive transaction, notifying the program running in the processor at the end, after it will configure the transceiver for the next block. Due to the sequential nature of a CPU, however, this reconfiguration time could cause a missing transaction if the bus master is not aware. To solve this, a protocol was devised with an extra *READY* line, set by the MCU, which allows the SPI master in the FPGA when the slave is ready to receive data. This solution has the downside of requiring a buffer on the master side to account for data arriving while the slave is not ready. By ensuring the time between SPI slave reconfigurations is short, however, this buffer can be only a few samples deep, instead of requiring to store a whole block.

The protocol, shown in Figure 5.12.A, consists of 16-bit transfers. The bus master listens for the *READY* line to be asserted, then initiates a single transfer containing a command sent by the MCU through the Master-In Slave-Out (MISO) line and a status word sent by the FPGA over the Master-Out Slave-In (MOSI) line. The status word contains information about the last performed command, bits reporting if it was successful and information about the configuration state of the device. After the command is issued another multi-word transfer is initiated containing the data from either of the devices. The size of this block is known to both the devices and dependent on the command.

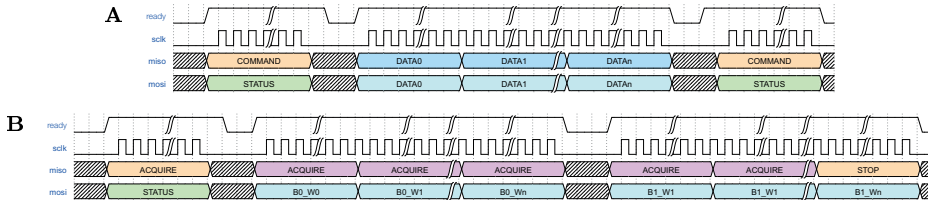


Figure 5.12: SPI communication scheme between the FPGA and the wireless processor. **A:** General command structure. **B:** Acquisition command structure.

Possible commands are:

- **Program:** to transfer data to FPGA memory, such as the Huffman tree of the acquisition chip configuration data.
- **Configure:** to start the configuration process of the RHD2132 acquisition chip. Sets a *configured* bit in the status word after a successful operation.

- **Acquire:** to start the acquisition process. Requires the RHD2132 to be properly configured.
- **Idle:** does not perform any action, but reports the status word.

The Idle and Acquire commands have a slightly different transfer structure. The Idle command does not require any extra data, so only the command is issued, with no follow up data transaction. The Acquire command, shown in Figure 5.12.B performs a continuous operation, so the standard rule of known data length can not apply. Instead, after an Acquire command has been issued, a continuous series of transactions containing a single data frame, as described in section 5.4.5 and Figure 5.10.D are performed. The value of the MISO line is largely ignored, until a stop word is issued. Once the FPGA receives a stop word, it will finish transferring the current frame, to successfully complete a full transaction, and stop acquiring after its completion, returning to the idle state and awaiting for the next command.

5.4.6.1 Firmware

The firmware for configuring the IGLOO FPGA, whose block diagram can be seen in Figure 5.13 was written in Verilog. A block-throttled SPI master handles communication with the MCU, with signals originating on the buffer FIFO and the different command state machines to regulate the output and ensure that transmission is done following the protocol.

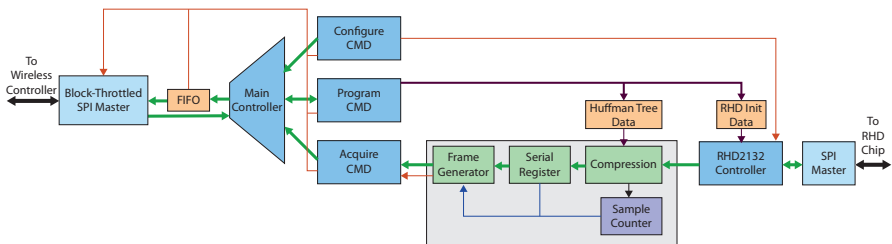


Figure 5.13: Block diagram of the wireless acquisition prototype FPGA firmware. Marked in a gray square are the core components of the algorithm.

A main controller is in charge of reading the commands and passing control to the appropriate block, except for Idle, of which the main controller takes care of, generating the status word. The different command controllers generate the appropriate data streams, with the Configure block sending a signal to the RHD controller to start configuration and the Program block filling the Huffman and RHD configuration memories.

The acquire block starts acquisition and manages data sent to the wireless processor. A controller block for the RHD2132 is in charge of sending the appropriate commands through a SPI master and retrieving raw neural data. The compression and framing process is comprised of three main blocks: A compressor block performing delta coding and Huffman compression through the methods described in section 5.4.1 and 5.4.2. A serial register with zero-filling capabilities packs the variable bit output of the compression block into 16-bit words, while a frame generator block packs the data into the protocol described in section 5.4.5. A sample counter keeps track of block boundaries.

5.4.6.2 Software

Software for the CC3220 network processor was developed in the C programming language. Texas instruments offers a framework to simplify network operation and peripheral access through the networking API *SimpleLink* and a low-footprint Real-Time Operating System (RTOS) called *TI-RTOS*, specialized for embedded devices and offering threading capabilities as well as functions to access the MCU peripherals, such as SPI transceivers.

Although the compression algorithm and transmission protocol are designed to be able to stream data to any wireless link without the need of extra memory, the CC3220 network processor requires data blocks to be loaded in its memory prior to transmission. Thus, a combination of double-buffering, threading and DMA transactions were used to allow the software to receive data from the FPGA and transmit over the wireless link simultaneously. Since the MCU sends data in full packets, it was decided that every transmitted packet would consist of a single block of compressed samples. This way, a missing packet would be easily recoverable by the receiving software. Thus, the algorithm for frame-decoding was integrated into the wireless processor software.

The threads created in the software were:

- A thread required by the background functions of the SimpleLink API.
- The main thread, handling SPI operations, decoding the frame structure and signaling the network thread when a whole block is present.
- The networking thread, that sleeps until notified by the main thread and sends the block over the network.

- Although not a software thread, SPI transactions are performed autonomously by the hardware and stored through DMA, only notifying the main thread when the transaction is finished.

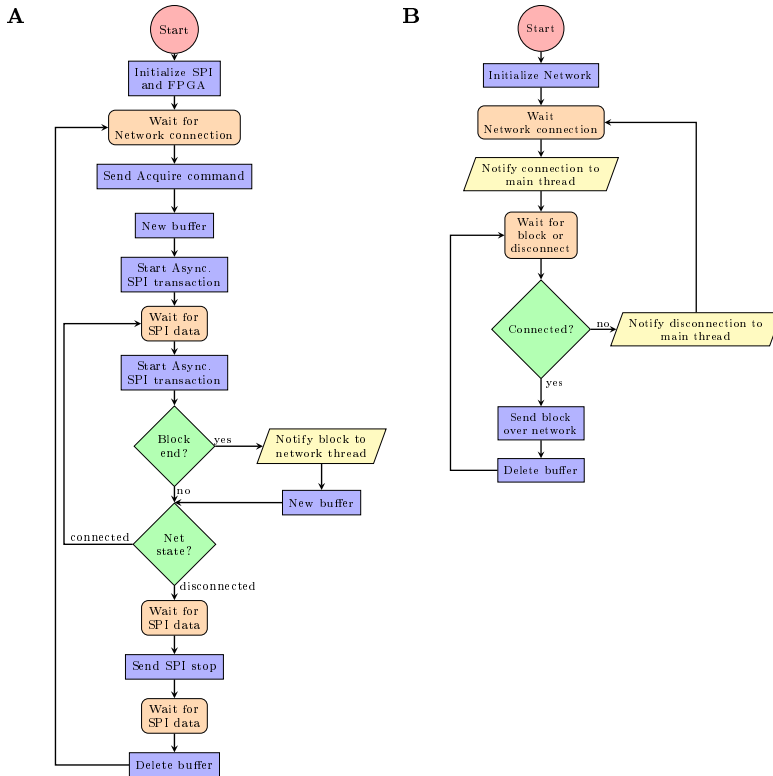


Figure 5.14: Wireless processor software block diagram. **A:** Main thread. **B:** Network thread

The main thread initializes the SPI bus, prepares the FPGA by uploading Huffman tree data and configuring the RHD chip and waits until there the network thread detects a data connection. Once connected, starts acquiring compressed neural data, notifying the network thread when a block is available. The main thread starts a new SPI DMA transaction the moment the previous one is finished and a block is received, so data transfer can occur in parallel through the hardware transceivers. Two buffers are used so SPI data can be acquired into one while the network thread sends the contents of the other. Once the wireless link is disconnected, the stop sequence is sent to the FPGA, the contents of the received block discarded, and the complete system returns to

the idle state. The network thread has a simpler loop, waiting for connections, informing the main thread of the network state and sending data. A complete flowchart of the two primary threads is shown in Figure 5.14.

5.4.7 Power usage

Bandwidth reduction, which compression achieves, can reduce power in two main ways: by allowing the usage of low-power protocols which often have a lower bandwidth associated or by enabling a higher bandwidth protocol in small bursts, increasing the time the device is not transmitting. In the case of the CC3220 network device used, it can be configured so the wireless circuitry enters a lower-power state between operations. This way less data to transmit translates to smaller bursts and longer sleep times for the wireless circuitry, thus reducing transmission power accordingly.

To measure power usage on the prototype, test points were added to independently measure current consumption of the FPGA, wireless processor and acquisition device. Accurately measuring the specific effect of compression required a known and noise-free signal to be transmitted both raw and compressed and transmission power be measured in both. To this avail, the synthetic signal described in section 5.3.2, generated inside the FPGA was used, by simply creating an alternative firmware replacing the compression scheme for a simple bypass.

Figure 5.15 shows the power usage of the wireless prototype. It can be seen how the amount of extra power used by the compression algorithm, measured at 2.7mW, is negligible. Since the Wi-Fi protocol is not designed specifically for low power, it features a high, static consumption dedicated to maintaining the link, even when it is not transmitting. However, even in this non-optimal case, the measurements demonstrate a clear reduction on transmission power, directly related to the decrement of required bandwidth.

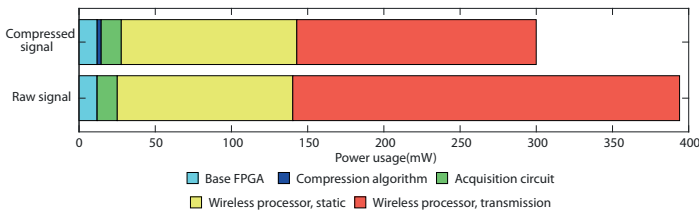


Figure 5.15: Power usage of the sample hardware implementation transmitting the compressed signal and the raw, uncompressed signal.

5.4.8 Resource usage

Minimizing hardware resources was an important objective, as this allows the algorithm to be used in a higher variety of existing devices, or makes it more efficient to be integrated into an ASIC. This includes both memory and logical requirements. In the case of the former, data size of the Huffman dictionary was reduced. Instead of the 2Mbit a 16-bit dataset would need, only 9Kbit were required, resulting on lesser memory blocks. Logic resource usage in the FPGA was kept minimal as well, with no need of DSP or any other specialized hardware block. Table 5.4 shows the FPGA cell usage of the different modules for both Xilinx and IGLOO nano FPGAs, as well as the percentage of the device used in the prototype.

	Xilinx Cells	IGLOO cells	Prototype usage
Compression	60	585	9.20%
Transmission protocol	22	210	3.42%
Data acquisition	54	495	8.08%

Table 5.4: Prototype usage percentage measured for the Microsemi AGLN250 device

5.5 Discussion

Studying complex and ecologically meaningful behaviors in animals is necessary to move experimental cognitive neuroscience forward [167], but requires experimental spaces closer to the natural conditions or even experiments in the real world. This often implies large spaces filled with elements like uneven terrain, obstacles, hiding places or even burrows, and environments shared by multiple animals. These elements render tethered devices impractical as the wiring, no matter its length or weight, would limit mobility and animal to animal interactions.

Wireless implants able to record brain activity during extended periods of time allow free movement of animals in complex environments, opening the possibility to a new generation of neurophysiological investigations in behaving animals.

For a wireless device, autonomy is crucial, with power usage being often the most limiting factor. Wireless data transmission has large power requirements, which are closely tied to bandwidth, with higher data rates requiring more power.

Compression is an efficient technique to reduce data rate, but only if the power needed for compression is lower than the power saved by rate reduction. This is the case for the algorithm presented in this chapter. Power reduction was demonstrated on a regular Wi-Fi IEEE 802.11g chip, designed for high-bandwidth and not optimized for low power, with a sizable percentage of its energy needs originating in static link usage. Using custom wireless protocols or newer low power devices, yet in development at the time of this writing, will reduce transmission power needs, in particular static requirements, further reducing total power. Specially interesting are the developments on IoT-related wireless protocols and devices, such as IEEE 802.11AH [168]], designed for low-power transmission while allowing a variety of different data rates.

Although this compression scheme was originally designed for wireless transmission, it can be of use to other electrophysiology technologies. For example, data loggers can increase the amount of stored data without physically increasing memory capacity while wired technologies could fit more channels into a single link.

This flexibility of usages is reinforced by the low-resource nature of the development. Minimal hardware needs results in an algorithm easy to fit in existing designs, being able to be implemented in a variety of devices. This is also important for power consumption as, unless highly optimized custom ASICs are used, devices with more hardware resources tend to be bigger and with more power requirements. This low-resource design makes it possible to be implemented in simple, low-power, commercial chips. The developed transmission protocol further reinforces this flexibility with the ability to maintain long-term signal integrity in the cases where data losses are expected. This might be the case for ultra-low-power wireless transmission protocols, as the drawback of expending less energy on link maintenance is the possibility of short interruptions on transmission, with their related packet losses. Being able to recover from such events makes the complete design suitable for almost any situation.

Data integrity and compression efficiency are two elements that must always be balanced. In this work the compression algorithm was developed with the former in mind, being virtually lossless and with compression noise below the noise floor of the acquisition chip. There are methods in which the compression ratio can be increased, while introducing noise into the signal. One such way is in the delta coding step. As seen in Fig 5.5.B.2, large delta values are rare and often the result of acquisition artifacts. Those uncommon, large values could be removed by trimming the most significant bits, further reducing word width [150]. In this case, any time such a large jump occurred, either naturally or by

an acquisition artifact, the DC offset of signal would drift from its real value, while maintaining most of its characteristics. In this case the signal would be corrected at the start of the following block. Another way to increase compression would be to trim even more bits before delta coding. This would result in a loss of resolution, with an equivalent noise of $V_{LSB} * 2^{n_{RemovedBits}}$. Conversely, if an acquisition chip with a lower noise floor were used, the number of discarded bits could be lowered, albeit with a slight impact on compression ratios.

Compression efficiency can also be improved without degrading signal quality by the optimization of the Huffman dictionary. section 5.4.4 shows how creating a customized dictionary with data previously recorded from the same experimental animals can increase compression. Understanding the specific factors that lead to these improvements could help further improve the performance. Current suspicions point to them being related to the physical properties of the experiment, such as electrode impedance and acquisition rate, which affect how the signal varies over time, and such the result of delta coding. More research on this topic needs to be done to further understand procedures to optimize the compression dictionary.

5.6 Conclusions

In order to reduce data bandwidth requirements for digitized brain electrophysiology signals, a low power compression algorithm was developed. It combines delta compression and Huffman code to compress neural data to nearly half its original size in a lossless manner, without adding any distortion beyond the acquisition circuit natural noise floor. Compression efficiency can be slightly improved by customizing the dictionaries using data from the same experimental animals.

This algorithm uses minimal hardware resources, making it possible to be implemented in low-power devices. A protocol for packing the compressed signals with little overhead and the capability to recover from packet losses was also developed for its use with wireless transmission. The compression algorithm and the transmission protocol add negligible extra power usage to the system, favoring the implementation of the algorithm in a variety of wireless electrophysiology acquisition systems.

Reducing bandwidth naturally reduces the power needed for a wireless transmission protocol. This was verified in a prototype wireless acquisition system created using commercially available, low resource and low-footprint devices.

Although the transmission protocol utilized in this work was not designed for low power, a sizable reduction in power consumption was achieved due to data compression.

Conclusions and Outlook

Brain extracellular electrophysiology is a powerful tool for neuroscience research. However, modern experiments are becoming more complex. They require arenas involving bigger and more ecologically meaningful spaces as well as multiple animal interactions. Closed-loop feedback, on the other hand, is moving towards responses to fast biological events, such as neural spikes. Under all these conditions, the technological limitations imposed by current tools become evident.

This thesis addresses these limitations through the development of novel hardware systems for data acquisition. All designs improve upon the characteristics offered by state-of-the-art systems, which offer multichannel acquisition of 16-bit data at 30KS/s. Improvements include technical characteristics, such as increased bandwidth allowing more than a thousand channels, and reduced latency, below $200\mu s$, but are not limited to those. New architectures and communication standards were developed to allow simultaneous acquisition from multiple, heterogeneous data sources such as video or animal tracking. Flexibility was a fundamental design constraint, with all systems allowing the creation of limitless experimental configurations and feedback algorithms.

Tether issues were also addressed through two different means. One is the design of ultra-light headstages coupled with torque-free, active commutators, reducing animal strain to a minimum. This allows long-running experiments in

large arenas with free 3-D movement. The second approach is the development of a wireless solution, able to acquire even in complex environments or in the presence of other animals. For this approach a custom compression algorithm was designed, able to reduce data bandwidth below 65.5% of its original size with no loss of signal integrity, allowing for extended battery life or reduced weight.

Combined, these developments allow for complex experiments, integrating multiple data sources with natural behavior and low-latency stimulation. The possibility to combine all these elements open new experimental possibilities not possible with current acquisition systems.

Three particular developments, improving on different aspect, are presented in this work:

Open Ephys is an open-source electrophysiology system designed with closed-loop experimentation in mind, able to acquire up to 512 channels. While sharing many similarities, and limitations, with traditional tools, its open source nature and modular design facilitates the creation and sharing of diverse closed-loop algorithms, with no hard limitations imposed by the tool. It features a digital headstage for noise tolerance and plenty I/Os for external hardware synchronization and communication.

The ONI specification defines an interface between acquisition hardware and a computer designed to enable the development of systems with multiple, heterogeneous and asynchronous sensor devices that can be effortlessly accessed by any compliant software. An acquisition tool implementing this standard, the ONIX system, was created. It features high-bandwidth multichannel acquisition, able to acquire simultaneously from over a thousand electrophysiological channels combined with other sources of data. The system features sub-millisecond acquisition latencies, which can be as low as $150\mu s$, allowing for complex, spike-based, closed-loop feedback. Its headstages are lightweight, less than 4g for a whole assembly, and connected through thin, coaxial cables using an active, torque-free commutator. This, coupled with 3D tracking of millimetric precision, allows its use on long-term experiments with freely-moving animals, including mice.

Lastly, a compression algorithm designed for wireless transmission of electrophysiology signals was devised, able to compress data below 65.5% of its original size with no distortion. This algorithm is designed to be device-agnostic and low-resource, allowing its implementation on any small, low-power device. Power required by the algorithm itself is negligible. Thanks to it, data rates of

any system can be nearly doubled or transmission-related power consumption considerably lowered. While developed for wireless applications, its device-agnostic nature allows it to be used in wired or logger systems. A hardware system performing neural acquisition, data compression and wireless transmission was created to demonstrate the algorithm effects.

6.1 Implications for neuroscience research

6.1.1 *Effect of tools in the experiments*

An ideal research tool should be transparent and offer faithful data without disturbing the experimental subjects. While this is not fully achievable in practice, developments have been made towards that goal.

Data accuracy needs to be high and not distorted by the acquisition process. All presented systems achieve this by using state-of-the-art probes and amplifier chips. Brain electrophysiology data is acquired at 16bits and 30KS/s independently of channel count. Electrical interference is kept as a minimum by digitizing data near the brain, while transmission protocols do not introduce any signal distortion. This includes the wireless compression algorithm developed in chapter 5, which is lossless, beyond the noise floor of the analog amplifier itself.

Channel count can become an important factor, as well, as with more channels available a higher number of neurons or brain regions can be monitored simultaneously, which can help to understand complex brain networks. Channel count, however, is limited by available bandwidth. Thus, while developed wired systems easily allow hundreds of channels wireless acquisition is still limited. Increasing bandwidth, by improving either efficiency of transmission protocols or compression ratios of compression algorithms, will result in an increase of available bandwidth and, by extension, channel count on wireless systems.

Beyond data accuracy, experimental equipment must disturb animals as little as possible. Neuroscience research is trending towards electrophysiology recordings with freely moving animals in complex spaces with social interaction [91]. In these conditions, headstage weight and tension due to torque hinder the movement range of the animals and cause fatigue which limit experiment duration while wires limit maximum area and arena complexity, as wires can become tangled with environmental object or be damaged by animals.

Some of these issues are addressed by employing head-fixed animals immersed in a VR environment [72]. This approach is specially useful with big, high-density probes that require support structures [169]. These kind of experiments are fully supported by both the Open Ephys GUI and the ONIX system, as both have support for high-density probes such as Neuropixels and feature I/Os that can be used to synchronize with animal movement in the VR space, such as treadmill motors or sensors. However VR environments are limited in their possibilities, as they do not fully and realistically represent a natural space. Thus, the ability to perform experiments in big arenas with complex elements and social interaction, with full freedom of movement, is highly desired. This thesis presents two developments towards this.

The ONIX system features lightweight (3.65g) headstages with ultra-thin coaxial tethers that ease mechanical restrictions on the animals. Moreover, the tethers can be connected through an actively driven commutator, which rotates automatically following the movement of the experimental subjects, thus completely negating the effect of torque. Thanks to this torque-free approach, cables can be hung using light elastic bands so they are retracted, out of reach for the animal, when it is near the commutator and fully extended when it is in the arena borders. This facilitates the use of longer cables and by extension larger arenas. Thanks to the combination of reduced animal strain and a tether less prone to failure thanks to the active commutator eliminating torque stress and animal-related damages, experiment time can be extended to even days.

Wireless transmission, on the other hand, removes any wiring limitation, enabling arenas of arbitrary size, multiple animals and any kind of environmental element, such as vegetation or burrows. The limitation of wireless solution comes in the form of batteries which limit their operating time and add weight, with bigger-capacity batteries being heavier. Compression helps with these issues by reducing data bandwidth and, by extension, transmission-related power consumptions, which translates to smaller batteries or longer operating times.

6.1.2 Closed-loop and brain timescales

The brain operates in multiple, different timescales. Some events, like neuronal action potentials, occur in the scale of one millisecond. Behavior-related patterns, however, are measured in ranges for seconds to days. Some events can produce different effects at different timescales. For example, external stimulation of serotonin-producing neurons through optogenetic means has opposite effects in the seconds scale when compared with weeks of experimentation [170]. Extrapolating results from short-term experiments into the long-term, then, is

not trivial and can result in erroneous conclusions. Studying such long-term effects require long-running experiments, with the associated hurdles due to electrophysiology tools which the presented developments have alleviated.

In the shorter scale, issues arise when closed-loop feedback is involved. Closed-loop experiments are a great tool for experimentation, allowing refined control over the animal and its brain. Stimulation, which can be of multiple natures, needs to be performed in the timescale of the event which the feedback loop is intended to modulate. This becomes challenging for fast events, as there is always a latency between the origin event and the stimuli due to the delays required for acquisition, data transmission and event processing. With the use of powerful computers with high-speed CPUs and computing GPUs [89], processing time is becoming a minor component, with data transmission representing the bulk of closed-loop latency.

The Open Ephys system, through its USB 3.0 interface, achieves mean latencies between 10 and 20ms. This is enough for LFP-based event detection and experiments involving synaptic plasticity [86] but limits research into faster events.

Events in the millisecond and sub-millisecond range are one of the design objectives of the ONIX system, which achieves a transmission-based closed-loop latency of $150\mu s$. This allows to quickly react to events such as action potentials, initiating stimulation even before the event has completed, as long as there is enough information for detection.

Latency in wireless links is a complex issue [171]. It is highly dependent on the wireless protocol and, in cases where the radioelectric spectrum is saturated anti-interference features from the transmission protocol can cause big uncertainties. In general, protocols with less arbitration mechanisms tend to have lower latencies, but at the risk of packet losses [132], while more complex protocols can assure no-loss transmission at the cost of higher and more uncertain latencies. If for this reason that the wireless algorithm designed in this work is accompanied by a custom framing protocol developed to take into account possible packet losses, assuring the possibility to recover from them.

6.1.3 Multi-source acquisition

Electrophysiology is not the only tool used in neuroscience research. Other tools, such as microscopy [114] can be used to inspect brain activity, stimulation can be achieved through a multitude of different techniques such as position or postural tracking [172].

While each tool provides different insights, combining them can allow to research relations that could not be studied independently. For example, while there is little doubt that behavior and brain activity are linked, timescales are different, so a composite view of both could help understand their relations. Even different versions of the same measurement can be present in parallel, such as high-resolution video or 3D tracking alongside radio-based location for movement inside burrows [173], [174].

The Open Ephys system has partial support for multi-source acquisition, with multiple I/Os in its hardware and the ability to create modules for different inputs in its software. However, due to being primarily designed for electrophysiology signal processing, the Open Ephys system has limitations when dealing with asynchronous, independent sources with heterogeneous clocks, formats and sampling rates.

To solve the problem of heterogeneous multi-source acquisition, the Open Neuro Interface (ONI) specification was created. The standard specifies a set of protocols for communication between acquisition hardware and software, independently of the nature of the data. It is designed to handle simultaneous streams from multiple devices with independent clock sources and data rates as well as bidirectional communication. Thus, it is possible for a single acquisition hardware to mix, for example, video, electrophysiology and optogenetics stimulation and transmit the data to a processing software with no synchronization issues.

The ONIX acquisition implements the ONI standard, exemplifying its multiple-source approach. It features headstages with high-density electrophysiology, 3D tracking and stimulation capabilities, both optical and electrical. The system includes 12 analog and 16 digital high-speed multipurpose I/Os and is compatible with miniaturized microscopy headstages. A library for the Bonsai visual programming language [107], designed for asynchronous data processing, was made.

6.1.4 Modular approach

Traditional, monolithic tools are limited to a fixed set of capabilities, making it difficult, if not impossible, to perform any experiment outside of them. Modular tools, on the other hand, can give more flexibility to researchers, able to reconfigure them to adjust to their particular needs. This philosophy is embraced by all projects described on this thesis.

The Open Ephys GUI features a fully-configurable signal chain based on a drag-and-drop interface. Any modules can be linked in any order, enabling the creation of a multitude of different processing chains, each apt for a particular analysis or closed-loop feedback stimuli generation scheme. Recording itself can be done from multiple parts of the chain simultaneously and in a variety of formats. Moreover, the plugin architecture of the software allows researchers to create their own processing modules, as well as plugins to accessing different acquisition or stimulation hardware, giving full freedom to the researcher.

The ONIX system goes a step further by incorporating this flexibility into the hardware itself thanks to the standardized nature of the ONI specification. Multiple, different headstages can be swapped in a plug-and-play manner, including switching from electrophysiology headstages to miniature microscopes.

In the case of the wireless system and compression algorithm, it has been developed as a module on itself. The device-agnostic and low-resource natures of the algorithm mean that it can be easily implemented in any kind of acquisition system. The presence of the transmission protocol helps with this, as allows the algorithm to be safely used with any kind of data link, regardless of its reliability.

6.2 Implications for the academic community

The tools described in this thesis have all been designed with modularity and flexibility in mind, allowing researchers to configure the tools around their intended experiments, instead of having to adapt their experiments to the tools available to them. Different approaches are able to fulfill distinct needs. For example, while wireless transmission gives the most environmental flexibility and freedom of movements, it is limited in time. However, the lightweight, torque-free headstages sacrifices a small amount of movement freedom, while still allowing more than traditional tools, in exchange for long-term experiment duration and high channel counts.

This flexibility is improved by the development of open interfaces and standards. In the case of most commercial tools the software and hardware are tied together, incapable of working independently. This is not the case with the presented tools. The Open Ephys hardware can be used with a variety of software beyond the Open Ephys GUI, while the GUI itself can support multiple acquisition and stimulation devices.

In the case of the ONIX system, the existence of a formal standard, in the form of the ONI specification, makes this compatibility granted as any hardware following the specification will be able to communicate with any compliant software. Moreover, the serializer interface in the ONIX system allows any manufacturer to create headstages that could be easily plugged into the system and immediately used by the software. Data types for newly used devices could easily be implemented in Bonsai.

The open-source nature of the software facilitates this flexibility by allowing researchers to modify it to suit their needs. Although their modular nature often means that new algorithms can be implemented and used in an individual manner, it is impossible to account for any circumstance. Being open-source avoids the possibility of an oversight preventing a particular experiment, allowing any researcher to alter the required parts. In the same ways, researchers can discover issues, oversights or shortcomings and either make them known or suggest a fix. Thanks to this, a tool can grow beyond the mindset of a single development team, becoming more complete and useful for the diverse members of the neuroscience community, as each one can contribute their specific knowledge.

This collaborative nature helps sharing knowledge, one of the pillars of science. Any developed algorithm can be shared and published along with the experimental results. The open nature of the standards and software guarantees that said algorithms do not depend on any undocumented piece, allowing their implementation on not only the original system, but any alternative, facilitating experimental replication.

6.3 Future steps

Future steps move towards the improvement of the presented tools. Probe density, channel count and overall data bandwidth of sensors will increase over the years, so acquisition technology must follow the same path. On the hardware level, for example, it is planned to eventually design a second version of the ONIX system featuring higher-bandwidth coaxial links, allowing for headstages with more channels or high-resolution video feeds.

Make the ecosystem grow is another path to follow. A USB-based ONIX system is planned to allow its use on computers without available PCIe slots, such as laptops. More hubs, with different sensors and acquisition capabilities could to be designed, even by different companies or development teams.

Regarding the wireless system, the natural next step is the creation of a complete, user friendly wireless headstage. It could use the ONI interface, being immediately included into the growing ecosystem and ensuring compatibility with all existing tools. This headstage would discard the WiFi protocol for a lower-power transmission method.

The algorithm itself can still be improved. For example, neural signals present high inter-channel redundancy, specially on lower frequencies. This could be exploited to achieve even better compression ratios. While the algorithm was originally designed to require little resources, a version using DSP modules, widely available in many FPGAs could be developed, taking advantage of filtering and transform features to further increase compression ratio. This approach, however, would require careful study, since power used by compression must always remain lower than power saved by bandwidth reduction. A big focus on compression complexity can lead to diminished returns.

Ideally, efforts should head towards a wireless headstage with the same capabilities of a wired one, featuring acquisition of high channel counts, tracking and stimulation while allowing free, unconstrained movement through a complex, and naturalistic environment. This device would communicate through a standardize interface such as ONI, allowing its use with any analysis software and enabling its integration into complex setups with controlled arenas and closed-loop feedback. Such as headstage would require efforts in further reducing power consumption of the system, not only compression but acquisition itself as well as stimulation.

As long as a battery is required, however, its weight and volume will have to be added to the headstage, hindering animal movement. Research into

headstages powered by wireless energy transfer methods [175], coupled with the advances already presented or discussed in this thesis could lead to the creation of such an ideal electrophysiology system: A lightweight, fully autonomous and feedback-capable implantable wireless headstage for long term, high bandwidth electrophysiological acquisition in complex environments.

Chapter 7

Contributions

7.1 Collaborations in the scope of the Thesis

Some of the developments described in this thesis have been the result of collaborations with external projects. This section details the specific contributions for each chapter.

While the Open Ephys project is a community-based effort, the bulk of the development of the acquisition system and software lies within the core team of the Open Ephys organization (Cambridge, MA, USA). This team is comprised of an international group of people of which the author of this document is part. In this context, the work performed in the scope of this thesis includes major developments in the software and FPGA firmware originally developed by the Open Ephys founders, as well as minor revisions to the hardware.

The work on the ONI specification and ONIX system presented on this thesis represents a full collaboration from the design phases of the project. Developed by a small team within the Open Ephys organization, including the author of this thesis, the most notable contributions to the project lie in system design and firmware and software development. Specifically, approximately a 40% of the ONI specification, 40% of the FPGA firmware and 30% of the interface software can be attributed to the author of this document. Additionally, the

participation has featured full involvement in design decisions of core structures and communication protocols.

Finally, the developments described in chapter 5 have been fully and independently performed within an academic scope at the Universitat Politècnica de València by the author, with no external collaboration.

7.2 Publications

- A. Cuevas-López, E. Pérez-Montoyo, V. J. López-Madrona, S. Canals, and D. Moratal, “Low-power lossless data compression for wireless electrophysiology acquisition”. Submitted.
- J. H. Siegle, A. C. López, Y. A. Patel, K. Abramov, S. Ohayon, and J. Voigts, “Open ephys: An open-source, plugin-based platform for multi-channel electrophysiology,” *Journal of Neural Engineering*, vol. 14, no. 4, p. 045 003, Jun. 2017. DOI: 10.1088/1741-2552/aa5eea
- D. R. Quiñones, A. Cuevas, J. Cambra, S. Canals, and D. Moratal, “RATT: RFID assisted tracking tile. preliminary results,” *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, ISSN: 1558-4615, Jul. 2017, pp. 4114–4117. DOI: 10.1109/EMBC.2017.8037761

7.3 Teaching

- “The Cajal NeuroKit Course: Extracellular Electrophysiology Acquisition”, CAJAL Training Programme, Online Course, Feb.-March. 2021. URL: <http://cajal-training.org/neurokit/electrophysiology-0321/>
- “36th Microelectrode Techniques for Cell Physiology Workshop”, Marine Biological Association, Plymouth, UK, Nov. 2019.
- “Instituto de Neurociencias Bonsai Course”, Instituto de Neurociencias de Alicante CSIC-UMH, San Juan de Alicante, Spain, Nov. 2017. URL: <https://neurogears.org/news/2017/10/12/inbc-2017.html>

7.4 Conference posters

- A. Cuevas-Lopez, D. R. Quiñones, E. Pérez-Montoyo, V. J. López-Madrona, J. Voigts, J. H. Siegle, S. Canals, and D. Moratal, “A low-power wireless transmission system of neural data by hardware compression,” 49th SfN Annual Meeting, Chicago, IL, USA, Oct. 22, 2019
- P. Kulik, A. Doshi, A. Cuevas-Lopez, J. Voigts, and J. H. Siegle, “Real-time processing and visualization of high-channel-count electrophysiology data with the open ephys GUI,” 49th SfN Annual Meeting, Chicago, IL, USA, Oct. 22, 2019
- J. P. Newman, J. Zhang, J. Voigts, A. Cuevas-Lopez, and M. A. Wilson, “An open-source PCIe based electrophysiology system for high data rate, low-latency closed-loop experiments,” 47th SfN Annual Meeting, Washington, DC, USA, Nov. 14, 2017
- A. Cuevas-Lopez, J. Voigts, and J. H. Siegle, “Open ephys: A flexible and affordable data acquisition system for extracellular electrophysiology,” 10th FENS Forum of Neuroscience, Copenhagen, Denmark, Jun. 7, 2016
- A. Cuevas-Lopez, Y. Patel, J. Voigts, and J. H. Siegle, “The open ephys gui: Plugin-based software for high-channel count,” 45th SfN Annual Meeting, Chicago, IL, USA, Oct. 21, 2015

Bibliography

- [1] D. Krioukov, “Brain theory,” *Frontiers in Computational Neuroscience*, vol. 8, 2014. DOI: 10.3389/fncom.2014.00114 (cit. on p. 1).
- [2] C. Golgi, “Sulla sostanza grigia del cervello,,” *Gazetta Medica Italiana*, vol. 33, pp. 244–246, 1873 (cit. on p. 1).
- [3] S. Zaqout and A. M. Kaindl, “Golgi-cox staining step by step,” *Frontiers in Neuroanatomy*, vol. 10, 2016. DOI: 10.3389/fnana.2016.00038 (cit. on pp. 1, 2).
- [4] Ramón y Cajal, S., “Estructura de los centros nerviosos de las aves,,” *Rev. Trim. Histol. Norm.*, 1888 (cit. on p. 1).
- [5] Brodmann, K., *Vergleichende Lokalisationslehre der Großhirnrinde : in ihren Prinzipien dargestellt auf Grund des Zellenbaues / von K. Brodmann*. 1909 (cit. on p. 1).
- [6] R. Yuste, “From the neuron doctrine to neural networks,” *Nature Reviews Neuroscience*, vol. 16, pp. 487–497, Aug. 2015. DOI: 10.1038/nrn3962 (cit. on p. 1).

- [7] E. R. Kandel, S. Mack, T. M. Jessell, J. H. Schwartz, S. A. Siegelbaum, and A. J. Hudspeth, *Principles of Neural Science, Fifth Edition*. McGraw Hill Professional, 2013, 1761 pp. (cit. on p. 1).
- [8] E. D. Adrian, *The basis of sensation*, ser. The basis of sensation. New York, NY, US: W W Norton & Co, 1928, 122 pp. (cit. on pp. 2, 7).
- [9] B. Hassard, “Bifurcation of periodic solutions of the Hodgkin-Huxley model for the squid giant axon,” *Journal of Theoretical Biology*, vol. 71, pp. 401–420, Apr. 6, 1978. DOI: [10.1016/0022-5193\(78\)90168-6](https://doi.org/10.1016/0022-5193(78)90168-6) (cit. on p. 2).
- [10] D. H. Hubel and T. N. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex,” *The Journal of Physiology*, vol. 160, pp. 106–154, 1962. DOI: <https://doi.org/10.1113/jphysiol.1962.sp006837> (cit. on pp. 2, 8).
- [11] D. H. Hubel and T. N. Wiesel, “Receptive fields and functional architecture of monkey striate cortex,” *The Journal of Physiology*, vol. 195, pp. 215–243, 1968. DOI: [10.1113/jphysiol.1968.sp008455](https://doi.org/10.1113/jphysiol.1968.sp008455) (cit. on pp. 2, 8).
- [12] J. O’Keefe and J. Dostrovsky, “The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat,” *Brain Research*, vol. 34, pp. 171–175, Nov. 12, 1971. DOI: [10.1016/0006-8993\(71\)90358-1](https://doi.org/10.1016/0006-8993(71)90358-1) (cit. on p. 2).
- [13] C. Gold, D. A. Henze, C. Koch, and G. Buzsáki, “On the origin of the extracellular action potential waveform: A modeling study,” *Journal of Neurophysiology*, vol. 95, pp. 3113–3128, May 1, 2006. DOI: [10.1152/jn.00979.2005](https://doi.org/10.1152/jn.00979.2005) (cit. on pp. 2, 42).
- [14] H. G. Glitsch, “Electrophysiology of the sodium-potassium-ATPase in cardiac cells,” *Physiological Reviews*, vol. 81, pp. 1791–1826, Jan. 10, 2001. DOI: [10.1152/physrev.2001.81.4.1791](https://doi.org/10.1152/physrev.2001.81.4.1791) (cit. on p. 3).

-
- [15] M. H. P. Kole, S. U. Ilshner, B. M. Kampa, S. R. Williams, P. C. Ruben, and G. J. Stuart, “Action potential generation requires a high sodium channel density in the axon initial segment,” *Nature Neuroscience*, vol. 11, pp. 178–186, Feb. 2008. DOI: 10.1038/nn2040 (cit. on p. 3).
- [16] N. Spruston, G. Stuart, and M. Häusser, “Principles of dendritic integration,” *Dendrites*, G. Stuart, N. Spruston, and M. Häusser, Eds., Oxford University Press, Mar. 1, 2016, pp. 351–398. DOI: 10.1093/acprof:oso/9780198745273.003.0012 (cit. on p. 3).
- [17] A. L. Hodgkin and A. F. Huxley, “Action potentials recorded from inside a nerve fibre,” *Nature*, vol. 144, pp. 710–711, Oct. 1939. DOI: 10.1038/144710a0 (cit. on p. 3).
- [18] G. Buzsáki, C. A. Anastassiou, and C. Koch, “The origin of extracellular fields and currents — EEG, ECoG, LFP and spikes,” *Nature Reviews Neuroscience*, vol. 13, pp. 407–420, Jun. 2012. DOI: 10.1038/nrn3241 (cit. on p. 3).
- [19] B. Sakmann, E. Neher, and P. P. Corporation, *Single-Channel Recording*. Springer US, 1983, 536 pp. (cit. on p. 3).
- [20] Neto, Joana P., “Materials and neuroscience: Validating tools for large-scale, high-density neural recording,” Ph.D. dissertation, Universidade NOVA de Lisboa, Lisbon, Portugal, Mar. 2018, 137 pp. (cit. on p. 3).
- [21] G. T. Einevoll, C. Kayser, N. K. Logothetis, and S. Panzeri, “Modelling and analysis of local field potentials for studying the function of cortical circuits,” *Nature Reviews Neuroscience*, vol. 14, pp. 770–785, Nov. 2013. DOI: 10.1038/nrn3599 (cit. on p. 3).
- [22] P. L. Nunez and R. Srinivasan, *Electric Fields of the Brain: The neurophysics of EEG*. Oxford University Press, 2006 (cit. on p. 3).
- [23] L.-W. S. Leung, “Field potentials in the central nervous system,” *Neurophysiological Techniques: Applications to Neural Systems*, ser. Neu-

- romethods, A. A. Boulton, G. B. Baker, and C. H. Vanderwolf, Eds., Totowa, NJ: Humana Press, 1990, pp. 277–312. DOI: 10.1385/0-89603-185-3:277 (cit. on p. 4).
- [24] J. Liu and W. T. Newsome, “Local field potential in cortical area MT: Stimulus tuning and behavioral correlations,” *Journal of Neuroscience*, vol. 26, pp. 7779–7790, Jul. 26, 2006. DOI: 10.1523/JNEUROSCI.5052-05.2006 (cit. on p. 4).
- [25] A. Berényi, Z. Somogyvári, A. J. Nagy, *et al.*, “Large-scale, high-density (up to 512 channels) recording of local circuits in behaving animals,” *Journal of Neurophysiology*, vol. 111, pp. 1132–1149, Mar. 1, 2014. DOI: 10.1152/jn.00785.2013 (cit. on p. 4).
- [26] C. Lewis, C. Bosman, and P. Fries, “Recording of brain activity across spatial scales,” *Current Opinion in Neurobiology*, Large-Scale Recording Technology (32), vol. 32, pp. 68–77, Jun. 1, 2015. DOI: 10.1016/j.conb.2014.12.007 (cit. on p. 4).
- [27] J. Navajas, D. Y. Barsakcioglu, A. Eftekhar, A. Jackson, T. G. Constantinou, and R. Quian Quiroga, “Minimum requirements for accurate and efficient real-time on-chip spike sorting,” *Journal of Neuroscience Methods*, vol. 230, pp. 51–64, Jun. 15, 2014. DOI: 10.1016/j.jneumeth.2014.04.018 (cit. on pp. 5, 8, 78, 91).
- [28] H. G. Rey, C. Pedreira, and R. Quian Quiroga, “Past, present and future of spike sorting techniques,” *Brain Research Bulletin*, Advances in electrophysiological data analysis, vol. 119, pp. 106–117, Oct. 1, 2015. DOI: 10.1016/j.brainresbull.2015.04.007 (cit. on p. 5).
- [29] R. Q. Quiroga, “Spike sorting,” *Scholarpedia*, vol. 2, p. 3583, Dec. 21, 2007. DOI: 10.4249/scholarpedia.3583 (cit. on pp. 5, 33).
- [30] M. Abeles and M. Goldstein, “Multispikes train analysis,” *Proceedings of the IEEE*, vol. 65, pp. 762–773, May 1977. DOI: 10.1109/PROC.1977.10559 (cit. on p. 5).

-
- [31] G. Buzsáki, “Large-scale recording of neuronal ensembles,” *Nature Neuroscience*, vol. 7, pp. 446–451, May 2004. DOI: 10.1038/nn1233 (cit. on pp. 5, 76).
- [32] D. H. Hubel, “Tungsten microelectrode for recording from single units,” *Science*, vol. 125, pp. 549–550, Mar. 22, 1957. DOI: 10.1126/science.125.3247.549 (cit. on p. 6).
- [33] B. L. McNaughton, J. O’Keefe, and C. A. Barnes, “The stereotrode: A new technique for simultaneous isolation of several single units in the central nervous system from multiple unit records,” *Journal of Neuroscience Methods*, vol. 8, pp. 391–397, Aug. 1, 1983. DOI: 10.1016/0165-0270(83)90097-3 (cit. on p. 6).
- [34] K. Najafi, K. Wise, and T. Mochizuki, “A high-yield IC-compatible multichannel recording array,” *IEEE Transactions on Electron Devices*, vol. 32, pp. 1206–1211, Jul. 1985. DOI: 10.1109/T-ED.1985.22102 (cit. on p. 6).
- [35] K. Takahashi and T. Matsuo, “Integration of multi-microelectrode and interface circuits by silicon planar and three-dimensional fabrication technology,” *Sensors and Actuators*, vol. 5, pp. 89–99, Jan. 1, 1984. DOI: 10.1016/0250-6874(84)87009-2 (cit. on p. 6).
- [36] J. Du, T. J. Blanche, R. R. Harrison, H. A. Lester, and S. C. Maniatis, “Multiplexed, high density electrophysiology with nanofabricated neural probes,” *PLOS ONE*, vol. 6, e26204, Oct. 12, 2011. DOI: 10.1371/journal.pone.0026204 (cit. on pp. 6, 7, 40).
- [37] C. M. Lopez, A. Andrei, S. Mitra, M. Welkenhuysen, W. Eberle, C. Batic, R. Puers, R. F. Yazicioglu, and G. Gielen, “An implantable 455-active-electrode 52-channel CMOS neural probe,” *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb. 2013, pp. 288–289. DOI: 10.1109/ISSCC.2013.6487738 (cit. on p. 6).

- [38] P. Ruther and O. Paul, “New approaches for CMOS-based devices for large-scale neural recording,” *Current Opinion in Neurobiology, Large-Scale Recording Technology* (32), vol. 32, pp. 31–37, Jun. 1, 2015. DOI: 10.1016/j.conb.2014.10.007 (cit. on pp. 6, 7).
- [39] M. Fiscella, K. Farrow, I. L. Jones, *et al.*, “Recording from defined populations of retinal ganglion cells using a high-density CMOS-integrated microelectrode array with real-time switchable electrode selection,” *Journal of Neuroscience Methods*, vol. 211, pp. 103–113, Oct. 15, 2012. DOI: 10.1016/j.jneumeth.2012.08.017 (cit. on p. 6).
- [40] J. J. Jun, N. A. Steinmetz, J. H. Siegle, *et al.*, “Fully integrated silicon probes for high-density recording of neural activity,” *Nature*, vol. 551, pp. 232–236, Nov. 2017. DOI: 10.1038/nature24636 (cit. on pp. 6, 7, 51).
- [41] J. Putzeys, B. C. Raducanu, A. Carton, *et al.*, “Neuropixels data-acquisition system: A scalable platform for parallel recording of 10 000+ electrophysiological signals,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, pp. 1635–1644, Dec. 2019. DOI: 10.1109/TBCAS.2019.2943077 (cit. on pp. 6, 35, 40, 51).
- [42] T. D. Y. Kozai, Z. Du, Z. V. Gugel, M. A. Smith, S. M. Chase, L. M. Bodily, E. M. Caparosa, R. M. Friedlander, and X. T. Cui, “Comprehensive chronic laminar single-unit, multi-unit, and local field potential recording performance with planar single shank electrode arrays,” *Journal of Neuroscience Methods*, vol. 242, pp. 15–40, Mar. 15, 2015. DOI: 10.1016/j.jneumeth.2014.12.010 (cit. on p. 6).
- [43] T. Chung, J. Q. Wang, J. Wang, B. Cao, Y. Li, and S. W. Pang, “Electrode modifications to lower electrode impedance and improve neural signal recording sensitivity,” *Journal of Neural Engineering*, vol. 12, p. 056 018, Sep. 2015. DOI: 10.1088/1741-2560/12/5/056018 (cit. on p. 6).

-
- [44] G. Baranauskas, E. Maggiolini, E. Castagnola, *et al.*, “Carbon nanotube composite coating of neural microelectrodes preferentially improves the multiunit signal-to-noise ratio,” *Journal of Neural Engineering*, vol. 8, p. 066 013, Oct. 2011. DOI: 10.1088/1741-2560/8/6/066013 (cit. on p. 6).
- [45] A. Ansaldo, E. Castagnola, E. Maggiolini, L. Fadiga, and D. Ricci, “Superior electrochemical performance of carbon nanotubes directly grown on sharp microelectrodes,” *ACS Nano*, vol. 5, pp. 2206–2214, Mar. 22, 2011. DOI: 10.1021/nn103445d (cit. on p. 6).
- [46] G. Márton, I. Bakos, Z. Fekete, I. Ulbert, and A. Pongrácz, “Durability of high surface area platinum deposits on microelectrode arrays for acute neural recordings,” *Journal of Materials Science: Materials in Medicine*, vol. 25, pp. 931–940, Mar. 1, 2014. DOI: 10.1007/s10856-013-5114-z (cit. on p. 6).
- [47] S. Arcot Desai, J. D. Rolston, L. Guo, and S. M. Potter, “Improving impedance of implantable microwire multi-electrode arrays by ultrasonic electroplating of durable platinum black,” *Frontiers in Neuroengineering*, vol. 3, 2010. DOI: 10.3389/fneng.2010.00005 (cit. on p. 6).
- [48] J. E. Ferguson, C. Boldt, and A. D. Redish, “Creating low-impedance tetrodes by electroplating with additives,” *Sensors and Actuators A: Physical*, vol. 156, pp. 388–393, Dec. 1, 2009. DOI: 10.1016/j.sna.2009.10.001 (cit. on p. 6).
- [49] C. M. Gray, P. E. Maldonado, M. Wilson, and B. McNaughton, “Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex,” *Journal of Neuroscience Methods*, vol. 63, pp. 43–54, Dec. 1, 1995. DOI: 10.1016/0165-0270(95)00085-2 (cit. on p. 6).
- [50] K. D. Harris, D. A. Henze, J. Csicsvari, H. Hirase, and G. Buzsáki, “Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements,” *Journal of Neurophysiol-*

- ogy, vol. 84, pp. 401–414, Jul. 1, 2000. DOI: 10.1152/jn.2000.84.1.401 (cit. on pp. 6, 33).
- [51] T. C. Ferree, P. Luu, G. S. Russell, and D. M. Tucker, “Scalp electrode impedance, infection risk, and EEG data quality,” *Clinical Neurophysiology*, vol. 112, pp. 536–544, Mar. 1, 2001. DOI: 10.1016/S1388-2457(00)00533-2 (cit. on p. 7).
- [52] R. Harrison and C. Charles, “A low-power low-noise CMOS amplifier for neural recording applications,” *IEEE Journal of Solid-State Circuits*, vol. 38, pp. 958–965, Jun. 2003. DOI: 10.1109/JSSC.2003.811979 (cit. on p. 7).
- [53] R. R. Harrison, “The design of integrated circuits to observe brain activity,” *Proceedings of the IEEE*, vol. 96, pp. 1203–1216, Jul. 2008. DOI: 10.1109/JPROC.2008.922581 (cit. on p. 7).
- [54] J. Ji and K. Wise, “An implantable CMOS circuit interface for multiplexed microelectrode recording arrays,” *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 433–443, Mar. 1992. DOI: 10.1109/4.121568 (cit. on p. 7).
- [55] A. L. Hodgkin and A. F. Huxley, “A quantitative description of membrane current and its application to conduction and excitation in nerve,” *The Journal of Physiology*, vol. 117, pp. 500–544, 1952. DOI: 10.1113/jphysiol.1952.sp004764 (cit. on p. 8).
- [56] D. H. Hubel and T. N. Wiesel, “Receptive fields of single neurones in the cat’s striate cortex,” *The Journal of Physiology*, vol. 148, pp. 574–591, Oct. 1959 (cit. on p. 8).
- [57] Y. Yang, P. Cao, Y. Yang, and S.-R. Wang, “Corollary discharge circuits for saccadic modulation of the pigeon visual system,” *Nature Neuroscience*, vol. 11, pp. 595–602, May 2008. DOI: 10.1038/nn.2107 (cit. on p. 8).

-
- [58] T. S. Okubo, E. L. Mackevicius, and M. S. Fee, “In vivo recording of single-unit activity during singing in zebra finches,” *Cold Spring Harbor protocols*, vol. 2014, pp. 1273–1283, Oct. 23, 2014. DOI: 10.1101/pdb.prot084624 (cit. on p. 8).
- [59] J. G. Canfield and S. J. Y. Mizumori, “Methods for chronic neural recording in the telencephalon of freely behaving fish,” *Journal of Neuroscience Methods*, vol. 133, pp. 127–134, Feb. 15, 2004. DOI: 10.1016/j.jneumeth.2003.10.011 (cit. on p. 8).
- [60] L. Johnston, R. E. Ball, S. Acuff, J. Gaudet, A. Sornborger, and J. D. Lauderdale, “Electrophysiological recording in the brain of intact adult zebrafish,” *Journal of Visualized Experiments : JoVE*, Nov. 19, 2013. DOI: 10.3791/51065 (cit. on p. 8).
- [61] R. Delventhal, A. Kiely, and J. R. Carlson, “Electrophysiological recording from drosophila labellar taste sensilla,” *Journal of Visualized Experiments : JoVE*, Feb. 26, 2014. DOI: 10.3791/51355 (cit. on p. 8).
- [62] P. J. Danneman, M. A. Suckow, and C. Brayton, *The Laboratory Mouse*. Boca Raton: CRC Press, Dec. 27, 2000, 184 pp. DOI: 10.1201/9780849376276 (cit. on pp. 8, 10).
- [63] M. A. Suckow, S. H. Weisbroth, and C. L. Franklin, *The laboratory rat*. Amsterdam; Boston: Elsevier, 2006 (cit. on p. 8).
- [64] B. Ellenbroek and J. Youn, “Rodent models in neuroscience research: Is it a rat race?” *Disease Models & Mechanisms*, vol. 9, pp. 1079–1087, Oct. 1, 2016. DOI: 10.1242/dmm.026120 (cit. on p. 8).
- [65] R. G. M. Morris, “Spatial localization does not require the presence of local cues,” *Learning and Motivation*, vol. 12, pp. 239–260, May 1, 1981. DOI: 10.1016/0023-9690(81)90020-5 (cit. on pp. 9, 76).
- [66] C. V. Vorhees and M. T. Williams, “Morris water maze: Procedures for assessing spatial and related forms of learning and memory,” *Nature*

- protocols*, vol. 1, pp. 848–858, 2006. DOI: 10.1038/nprot.2006.116 (cit. on p. 9).
- [67] J. S. Taube, R. U. Muller, and J. B. Ranck, “Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis,” *Journal of Neuroscience*, vol. 10, pp. 420–435, Feb. 1, 1990. DOI: 10.1523/JNEUROSCI.10-02-00420.1990 (cit. on pp. 9, 76).
- [68] M. Rosenberg, T. Zhang, P. Perona, and M. Meister, “Mice in a labyrinth: Rapid learning, sudden insight, and efficient exploration,” *Neuroscience*, preprint, Jan. 15, 2021. DOI: 10.1101/2021.01.14.426746 (cit. on pp. 9, 76).
- [69] V. J. López-Madrona, E. Pérez-Montoyo, E. Álvarez-Salvado, D. Moratal, O. Herreras, E. Pereda, C. R. Mirasso, and S. Canals, “Different theta frameworks coexist in the rat hippocampus and are coordinated during memory-guided and novelty tasks,” *eLife*, vol. 9, M. Vinck, L. L. Colgin, and A. Fernandez-Ruiz, Eds., e57313, Jul. 20, 2020. DOI: 10.7554/eLife.57313 (cit. on pp. 9, 76).
- [70] J. H. Siegle and M. A. Wilson, “Enhancement of encoding and retrieval functions through theta phase-specific manipulation of hippocampus,” *eLife*, vol. 3, e03061, Jul. 29, 2014. DOI: 10.7554/eLife.03061 (cit. on pp. 9, 33, 36, 76).
- [71] D. S. Olton, C. Collison, and M. A. Werz, “Spatial memory and radial arm maze performance of rats,” *Learning and Motivation*, vol. 8, pp. 289–314, 1977. DOI: 10.1016/0023-9690(77)90054-6 (cit. on p. 9).
- [72] K. Thurley and A. Ayaz, “Virtual reality systems for rodents,” *Current Zoology*, vol. 63, pp. 109–119, Feb. 2017. DOI: 10.1093/cz/zow070 (cit. on pp. 9, 112).
- [73] J. Voigts and M. T. Harnett, “An animal-actuated rotational head-fixation system for 2-photon imaging during 2-d navigation,” *Neuroscience*, preprint, Mar. 1, 2018. DOI: 10.1101/262543 (cit. on p. 9).

-
- [74] M. M. Yartsev and N. Ulanovsky, “Representation of three-dimensional space in the hippocampus of flying bats,” *Science*, vol. 340, pp. 367–372, Apr. 19, 2013. DOI: 10.1126/science.1235338 (cit. on p. 9).
- [75] S. M. Potter, A. El Hady, and E. E. Fetz, “Closed-loop neuroscience and neuroengineering,” *Frontiers in Neural Circuits*, vol. 8, Sep. 23, 2014. DOI: 10.3389/fncir.2014.00115 (cit. on p. 9).
- [76] L. Grosenick, J. H. Marshel, and K. Deisseroth, “Closed-loop and activity-guided optogenetic control,” *Neuron*, vol. 86, pp. 106–139, Apr. 2015. DOI: 10.1016/j.neuron.2015.03.034 (cit. on pp. 9, 20, 40).
- [77] J. P. Newman, M.-f. Fong, D. C. Millard, C. J. Whitmire, G. B. Stanley, and S. M. Potter, “Optogenetic feedback control of neural activity,” *eLife*, vol. 4, M. Bartos, Ed., e07192, Jul. 3, 2015. DOI: 10.7554/eLife.07192 (cit. on p. 9).
- [78] A. Jackson, J. Mavoori, and E. E. Fetz, “Long-term motor cortex plasticity induced by an electronic neural implant,” *Nature*, vol. 444, pp. 56–60, Nov. 2006. DOI: 10.1038/nature05226 (cit. on pp. 9, 10).
- [79] A. Wallach, D. Eytan, A. Gal, C. Zrenner, and S. Marom, “Neuronal response clamp,” *Frontiers in Neuroengineering*, vol. 4, 2011. DOI: 10.3389/fneng.2011.00003 (cit. on pp. 9, 10).
- [80] S. P. Jadhav, C. Kemere, P. W. German, and L. M. Frank, “Awake hippocampal sharp-wave ripples support spatial memory,” *Science*, vol. 336, pp. 1454–1458, Jun. 15, 2012. DOI: 10.1126/science.1217230 (cit. on p. 9).
- [81] M. B. Ahrens, J. M. Li, M. B. Orger, D. N. Robson, A. F. Schier, F. Engert, and R. Portugues, “Brain-wide neuronal dynamics during motor adaptation in zebrafish,” *Nature*, vol. 485, pp. 471–477, May 2012. DOI: 10.1038/nature11057 (cit. on p. 9).
- [82] M. B. Reiser and M. H. Dickinson, “A modular display system for insect behavioral neuroscience,” *Journal of Neuroscience Methods*, vol. 167,

- pp. 127–139, Jan. 30, 2008. DOI: 10.1016/j.jneumeth.2007.07.019 (cit. on p. 10).
- [83] H.-V. V. Ngo, T. Martinetz, J. Born, and M. Mölle, “Auditory closed-loop stimulation of the sleep slow oscillation enhances memory,” *Neuron*, vol. 78, pp. 545–553, May 8, 2013. DOI: 10.1016/j.neuron.2013.03.006 (cit. on p. 10).
- [84] G. Kozák and A. Berényi, “Sustained efficacy of closed loop electrical stimulation for long-term treatment of absence epilepsy in rats,” *Scientific Reports*, vol. 7, p. 6300, Jul. 24, 2017. DOI: 10.1038/s41598-017-06684-0 (cit. on p. 10).
- [85] J. P. Newman, “Optogenetic feedback control of neural activity,” Ph.D. dissertation, Georgia Institute of Technology, Atlanta, Georgia, USA, Nov. 18, 2013, 269 pp. (cit. on p. 10).
- [86] P. J. Drew and L. F. Abbott, “Extending the effects of spike-timing-dependent plasticity to behavioral timescales,” *Proceedings of the National Academy of Sciences*, vol. 103, pp. 8876–8881, Jun. 6, 2006. DOI: 10.1073/pnas.0600676103 (cit. on pp. 10, 42, 113).
- [87] F. Franke, D. Jäckel, J. Dragas, J. Müller, M. Radivojevic, D. Bakkum, and A. Hierlemann, “High-density microelectrode array recordings and real-time spike sorting for closed-loop experiments: An emerging technology to study neural plasticity,” *Frontiers in Neural Circuits*, vol. 6, 2012. DOI: 10.3389/fncir.2012.00105 (cit. on pp. 10, 42).
- [88] S. H. Fuller and L. I. Millett, “Computing performance: Game over or next level?” *Computer*, vol. 44, pp. 31–38, Jan. 2011. DOI: 10.1109/MC.2011.15 (cit. on p. 11).
- [89] M. Pachitariu, N. Steinmetz, S. Kadir, M. Carandini, and H. K. D., “Kilosort: Realtime spike-sorting for extracellular electrophysiology with hundreds of channels,” *bioRxiv*, p. 061481, Jun. 30, 2016. DOI: 10.1101/061481 (cit. on pp. 11, 42, 113).

-
- [90] K. Juczewski, J. A. Koussa, A. J. Kesner, J. O. Lee, and D. M. Lovinger, “Stress and behavioral correlates in the head-fixed method: Stress measurements, habituation dynamics, locomotion, and motor-skill learning in mice,” *Scientific Reports*, vol. 10, p. 12 245, Jul. 22, 2020. DOI: 10.1038/s41598-020-69132-6 (cit. on p. 13).
- [91] J. W. Krakauer, A. A. Ghazanfar, A. Gomez-Marin, M. A. MacIver, and D. Poeppel, “Neuroscience needs behavior: Correcting a reductionist bias,” *Neuron*, vol. 93, pp. 480–490, Feb. 2017. DOI: 10.1016/j.neuron.2016.12.041 (cit. on pp. 13, 41, 76, 111).
- [92] E. J. Dennis, A. E. Hady, A. Michaiel, A. Clemens, D. R. G. Tervo, J. Voigts, and S. R. Datta, “Systems neuroscience of natural behaviors in rodents,” *Journal of Neuroscience*, vol. 41, pp. 911–919, Feb. 3, 2021. DOI: 10.1523/JNEUROSCI.1877-20.2020 (cit. on pp. 13, 14, 40, 41).
- [93] M. Yin, D. A. Borton, J. Komar, *et al.*, “Wireless neurosensor for full-spectrum electrophysiology recordings during free behavior,” *Neuron*, vol. 84, pp. 1170–1182, Dec. 17, 2014. DOI: 10.1016/j.neuron.2014.11.010 (cit. on pp. 14, 76–78).
- [94] A. Nourizonoz, R. Zimmermann, C. L. A. Ho, *et al.*, “EthoLoop: Automated closed-loop neuroethology in naturalistic environments,” *Nature Methods*, vol. 17, pp. 1052–1059, Oct. 1, 2020. DOI: 10.1038/s41592-020-0961-2 (cit. on p. 15).
- [95] M.-O. Gewaltig and R. Cannon, “Current practice in software development for computational neuroscience and how to improve it,” *PLOS Computational Biology*, vol. 10, e1003376, 2014. DOI: 10.1371/journal.pcbi.1003376 (cit. on pp. 20, 21).
- [96] J. H. Siegle, A. C. López, Y. A. Patel, K. Abramov, S. Ohayon, and J. Voigts, “Open ephys: An open-source, plugin-based platform for multichannel electrophysiology,” *Journal of Neural Engineering*, vol. 14, p. 045 003, Jun. 2017. DOI: 10.1088/1741-2552/aa5eea (cit. on pp. 21, 33, 120).

- [97] S. WEBER, *The Success of Open Source*. Harvard University Press, Jun. 30, 2009, 321 pp. (cit. on p. 20).
- [98] L. Rosen, *Open source licensing*. Prentice Hall, 2005, vol. 692 (cit. on p. 20).
- [99] E. De Schutter, “Collaborative modeling in neuroscience: Time to go open model?” *Neuroinformatics*, vol. 11, pp. 135–136, Apr. 1, 2013. DOI: 10.1007/s12021-013-9181-6 (cit. on p. 21).
- [100] “Intan technologies RHD electrophysiology amplifier chips,” Intan Technologies. (), [Online]. Available: http://intantech.com/products_RHD2000.html (visited on 05/25/2020) (cit. on pp. 22, 41).
- [101] “Spartan-6 FPGA family,” Xilinx. (), [Online]. Available: <https://www.xilinx.com/products/silicon-devices/fpga/spartan-6.html> (visited on 01/05/2021) (cit. on p. 24).
- [102] opalkelly.com. “XEM6310,” Opal Kelly. (), [Online]. Available: <https://opalkelly.com/products/xem6310/> (visited on 02/02/2021) (cit. on p. 24).
- [103] J. Voigts, J. P. Newman, M. A. Wilson, and M. T. Harnett, “An easy-to-assemble, robust, and lightweight drive implant for chronic tetrode recordings in freely moving animals,” *bioRxiv*, p. 746 651, Aug. 24, 2019. DOI: 10.1101/746651 (cit. on p. 26).
- [104] M. S. Lewicki, “A review of methods for spike sorting: The detection and classification of neural action potentials,” *Network: Computation in Neural Systems*, vol. 9, R53–R78, Jan. 1, 1998. DOI: 10.1088/0954-898X_9_4_001 (cit. on p. 33).
- [105] S. Shoham, M. R. Fellows, and R. A. Normann, “Robust, automatic spike sorting using mixtures of multivariate t-distributions,” *Journal of Neuroscience Methods*, vol. 127, pp. 111–122, Aug. 15, 2003. DOI: 10.1016/S0165-0270(03)00120-1 (cit. on p. 33).

-
- [106] J. L. Teeters, K. Godfrey, R. Young, *et al.*, “Neurodata without borders: Creating a common data format for neurophysiology,” *Neuron*, vol. 88, pp. 629–634, Nov. 18, 2015. DOI: 10.1016/j.neuron.2015.10.025 (cit. on p. 33).
- [107] G. Lopes, N. Bonacchi, J. Frazão, *et al.*, “Bonsai: An event-based framework for processing and controlling data streams,” *Frontiers in Neuroinformatics*, vol. 9, 2015. DOI: 10.3389/fninf.2015.00007 (cit. on pp. 35, 52, 114).
- [108] E. Musk and Neuralink, “An integrated brain-machine interface platform with thousands of channels,” *bioRxiv*, p. 703801, Aug. 2, 2019. DOI: 10.1101/703801 (cit. on p. 35).
- [109] A. E. X. Brown and B. de Bivort, “Ethology as a physical science,” *Nature Physics*, vol. 14, pp. 653–657, Jul. 1, 2018. DOI: 10.1038/s41567-018-0093-0 (cit. on p. 40).
- [110] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge, “DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning,” *Nature Neuroscience*, vol. 21, pp. 1281–1289, Sep. 1, 2018. DOI: 10.1038/s41593-018-0209-y (cit. on p. 40).
- [111] A. T. Schaefer and A. Claridge-Chang, “The surveillance state of behavioral automation,” *Current Opinion in Neurobiology, Neurotechnology*, vol. 22, pp. 170–176, Feb. 2012. DOI: 10.1016/j.conb.2011.11.004 (cit. on p. 40).
- [112] A. S. Reinhold, J. I. Sanguinetti-Scheck, K. Hartmann, and M. Brecht, “Behavioral and neural correlates of hide-and-peek in rats,” *Science*, vol. 365, pp. 1180–1183, Sep. 13, 2019. DOI: 10.1126/science.aax4705 (cit. on pp. 40, 76).
- [113] G. Gagnon-Turcotte, Y. LeChasseur, C. Bories, Y. Messaddeq, Y. De Koninck, and B. Gosselin, “A wireless headstage for combined optoge-

- netics and multichannel electrophysiological recording,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, pp. 1–14, Feb. 2017. DOI: 10.1109/TBCAS.2016.2547864 (cit. on pp. 40, 79).
- [114] D. Aharoni, B. S. Khakh, A. J. Silva, and P. Golshani, “All the light that we can see: A new era in miniaturized microscopy,” *Nature Methods*, vol. 16, p. 11, Jan. 2019. DOI: 10.1038/s41592-018-0266-x (cit. on pp. 40, 44, 60, 114).
- [115] N. A. Steinmetz, C. Aydin, A. Lebedeva, *et al.*, “Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings,” *bioRxiv*, p. 2020.10.27.358291, Oct. 28, 2020. DOI: 10.1101/2020.10.27.358291 (cit. on p. 40).
- [116] Texas Instruments, “An introduction to FPD-link (application note),” p. 9, 1998 (cit. on p. 44).
- [117] M. Jacobsen, D. Richmond, M. Hogains, and R. Kastner, “RIFFA 2.1: A reusable integration framework for FPGA accelerators,” *ACM Transactions on Reconfigurable Technology and Systems*, vol. 8, 22:1–22:23, Sep. 13, 2015. DOI: 10.1145/2815631 (cit. on p. 46).
- [118] S. W. Golomb, *Shift register sequences*, Rev. ed. Laguna Hills, Calif: Aegean Park Press, 1982, xvi, 247 (cit. on p. 49).
- [119] A. Mahnam, H. Yazdanian, and M. Mosayebi Samani, “Comprehensive study of howland circuit with non-ideal components to design high performance current pumps,” *Measurement*, vol. 82, pp. 94–104, Mar. 1, 2016. DOI: 10.1016/j.measurement.2015.12.044 (cit. on p. 51).
- [120] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser, “Microstructure of a spatial map in the entorhinal cortex,” *Nature*, vol. 436, pp. 801–806, Aug. 2005. DOI: 10.1038/nature03721 (cit. on p. 76).
- [121] E. I. Moser, E. Kropff, and M.-B. Moser, “Place cells, grid cells, and the brain’s spatial representation system,” *Annual Review of Neuroscience*,

- vol. 31, pp. 69–89, 2008. DOI: 10.1146/annurev.neuro.31.061307.090723 (cit. on p. 76).
- [122] D. J. Cai, D. Aharoni, T. Shuman, *et al.*, “A shared neural ensemble links distinct contextual memories encoded close in time,” *Nature*, vol. 534, p. 115, May 23, 2016 (cit. on p. 76).
- [123] A. F. Meyer, J. O’Keefe, and J. Poort, “Two distinct types of eye-head coupling in freely moving mice,” *Current Biology*, vol. 30, 2116–2130.e6, Jun. 8, 2020. DOI: 10.1016/j.cub.2020.04.042 (cit. on p. 76).
- [124] B. Massot, S. Arthaud, B. Barrillot, J. Roux, G. Ungurean, P.-H. Luppi, N. C. Rattenborg, and P.-A. Libourel, “ONEIROS, a new miniature standalone device for recording sleep electrophysiology, physiology, temperatures and behavior in the lab and field,” *Journal of Neuroscience Methods*, Methods and models in sleep research: A Tribute to Vincenzo Crunelli, vol. 316, pp. 103–116, Mar. 15, 2019. DOI: 10.1016/j.jneumeth.2018.08.030 (cit. on p. 76).
- [125] N. C. Rattenborg, B. Voirin, S. M. Cruz, R. Tisdale, G. Dell’Omo, H.-P. Lipp, M. Wikelski, and A. L. Vyssotski, “Evidence that birds sleep in mid-flight,” *Nature Communications*, vol. 7, pp. 1–9, Aug. 3, 2016. DOI: 10.1038/ncomms12468 (cit. on p. 77).
- [126] E. Vinepinsky, O. Donchin, and R. Segev, “Wireless electrophysiology of the brain of freely swimming goldfish,” *Journal of Neuroscience Methods*, vol. 278, pp. 76–86, 2017. DOI: 10.1016/j.jneumeth.2017.01.001 (cit. on p. 77).
- [127] A. Borna and K. Najafi, “A low power light weight wireless multichannel microsystem for reliable neural recording,” *IEEE Journal of Solid-State Circuits*, vol. 49, pp. 439–451, Feb. 2014. DOI: 10.1109/JSSC.2013.2293773 (cit. on pp. 77, 78).

- [128] D. Fan, D. Rich, T. Holtzman, *et al.*, “A wireless multi-channel recording system for freely behaving mice and rats,” *PLOS ONE*, vol. 6, e22033, Jul. 12, 2011. DOI: 10.1371/journal.pone.0022033 (cit. on p. 77).
- [129] Y. Su, S. Routhu, K. S. Moon, S. Q. Lee, W. Youm, and Y. Ozturk, “A wireless 32-channel implantable bidirectional brain machine interface,” *Sensors*, vol. 16, p. 1582, Oct. 2016. DOI: 10.3390/s16101582 (cit. on p. 77).
- [130] A. Ghomashchi, Z. Zheng, N. Majaj, M. Trumpis, L. Kiorpes, and J. Viventi, “A low-cost, open-source, wireless electrophysiology system,” *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug. 2014, pp. 3138–3141. DOI: 10.1109/EMBC.2014.6944288 (cit. on pp. 77, 78).
- [131] Y. Jia, W. Khan, B. Lee, B. Fan, F. Madi, A. Weber, W. Li, and M. Ghovanloo, “Wireless opto-electro neural interface for experiments with small freely behaving animals,” *Journal of Neural Engineering*, vol. 15, p. 046032, Jun. 2018. DOI: 10.1088/1741-2552/aac810 (cit. on p. 77).
- [132] M. R. Mukati, S. Kocatürk, M. Kocatürk, and T. Baykaş, “A microcontroller-based wireless multichannel neural data transmission system,” *2017 21st National Biomedical Engineering Meeting (BIYOMUT)*, Nov. 2017, pp. i–iv. DOI: 10.1109/BIYOMUT.2017.8479160 (cit. on pp. 77, 96, 113).
- [133] S. Brenna, F. Padovan, A. Neviani, A. Bevilacqua, A. Bonfanti, and A. L. Lacaita, “A 64-channel 965- μm neural recording SoC with UWB wireless transmission in 130-nm CMOS,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, pp. 528–532, Jun. 2016. DOI: 10.1109/TCSII.2016.2530882 (cit. on pp. 77, 84).
- [134] R. Ameli, A. Mirbozorgi, J.-L. Néron, Y. LeChasseur, and B. Gosselin, “A wireless and batteryless neural headstage with optical stimulation and electrophysiological recording,” *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*

-
- (*EMBC*), Jul. 2013, pp. 5662–5665. DOI: 10.1109/EMBC.2013.6610835 (cit. on p. 77).
- [135] K. L. Montgomery, A. J. Yeh, J. S. Ho, *et al.*, “Wirelessly powered, fully internal optogenetics for brain, spinal and peripheral circuits in mice,” *Nature Methods*, vol. 12, pp. 969–974, Oct. 2015. DOI: 10.1038/nmeth.3536 (cit. on p. 77).
- [136] S. M. Won, L. Cai, P. Gutruf, and J. A. Rogers, “Wireless and battery-free technologies for neuroengineering,” *Nature Biomedical Engineering*, pp. 1–19, Mar. 8, 2021. DOI: 10.1038/s41551-021-00683-3 (cit. on p. 77).
- [137] S. Lee, A. J. Cortese, A. P. Gandhi, E. R. Agger, P. L. McEuen, and A. C. Molnar, “A $250\ \mu\text{m} \times 57\ \mu\text{m}$ microscale opto-electronically transduced electrodes (MOTEs) for neural recording,” *IEEE transactions on biomedical circuits and systems*, vol. 12, pp. 1256–1266, Dec. 2018. DOI: 10.1109/TBCAS.2018.2876069 (cit. on p. 77).
- [138] H. Ding, L. Lu, Z. Shi, *et al.*, “Microscale optoelectronic infrared-to-visible upconversion devices and their use as injectable light sources,” *Proceedings of the National Academy of Sciences*, vol. 115, pp. 6632–6637, Jun. 26, 2018. DOI: 10.1073/pnas.1802064115 (cit. on p. 77).
- [139] G.-T. Hwang, Y. Kim, J.-H. Lee, *et al.*, “Self-powered deep brain stimulation via a flexible PIMNT energy harvester,” *Energy & Environmental Science*, vol. 8, pp. 2677–2684, Aug. 26, 2015. DOI: 10.1039/C5EE01593F (cit. on p. 77).
- [140] K. T. Settaluri, H. Lo, and R. J. Ram, “Thin thermoelectric generator system for body energy harvesting,” *Journal of Electronic Materials*, vol. 41, pp. 984–988, Jun. 1, 2012. DOI: 10.1007/s11664-011-1834-3 (cit. on p. 77).

- [141] R. A. Bullen, T. C. Arnot, J. B. Lakeman, and F. C. Walsh, "Biofuel cells and their development," *Biosensors & Bioelectronics*, vol. 21, pp. 2015–2045, May 15, 2006. DOI: 10.1016/j.bios.2006.01.030 (cit. on p. 77).
- [142] Y. Jia, B. Lee, F. Kong, Z. Zeng, M. Connolly, B. Mahmoudi, and M. Ghovanloo, "A software-defined radio receiver for wireless recording from freely behaving animals," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, pp. 1645–1654, Dec. 2019. DOI: 10.1109/TBCAS.2019.2949233 (cit. on pp. 78, 84).
- [143] B. Lee, Y. Jia, F. Kong, M. Connolly, B. Mahmoudi, and M. Ghovanloo, "Toward a robust multi-antenna receiver for wireless recording from freely-behaving animals," *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, Oct. 2018, pp. 1–4. DOI: 10.1109/BIOCAS.2018.8584800 (cit. on p. 78).
- [144] S. B. Lee, M. Yin, J. R. Manns, and M. Ghovanloo, "A wideband dual-antenna receiver for wireless recording from animals behaving in large arenas," *IEEE Transactions on Biomedical Engineering*, vol. 60, pp. 1993–2004, Jul. 2013. DOI: 10.1109/TBME.2013.2247603 (cit. on p. 78).
- [145] M. Pagin, M. Haas, J. Becker, and M. Ortmanns, "Delta compression in time-multiplexed multichannel neural recorders," *2016 12th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, Jun. 2016, pp. 1–4. DOI: 10.1109/PRIME.2016.7519487 (cit. on pp. 78, 91).
- [146] M. Alsenwi, T. Ismail, and H. Mostafa, "Performance analysis of hybrid lossy/lossless compression techniques for EEG data," *2016 28th International Conference on Microelectronics (ICM)*, Dec. 2016, pp. 1–4. DOI: 10.1109/ICM.2016.7847849 (cit. on p. 78).
- [147] C. H. Chung, L.-G. Chen, Y.-C. Kao, and F.-S. Jaw, "Multichannel evoked neural signal compression using advanced video compression algorithm," *2009 4th International IEEE/EMBS Conference on Neural*

-
- Engineering*, Apr. 2009, pp. 697–701. DOI: 10.1109/NER.2009.5109392 (cit. on p. 79).
- [148] M. A. Shaeri and A. M. Sodagar, “A method for compression of intracortically-recorded neural signals dedicated to implantable brain–machine interfaces,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, pp. 485–497, May 2015. DOI: 10.1109/TNSRE.2014.2355139 (cit. on p. 79).
- [149] S. Akhter and M. A. Haque, “ECG compression using run length encoding,” *2010 18th European Signal Processing Conference*, Aug. 2010, pp. 1645–1649 (cit. on p. 79).
- [150] U. Bihr, H. Xu, C. Bulach, M. Lorenz, J. Anders, and M. Ortmanms, “Real-time data compression of neural spikes,” *New Circuits and Systems Conference (NEWCAS), 2014 IEEE 12th International*, Jun. 2014, pp. 436–439. DOI: 10.1109/NEWCAS.2014.6934076 (cit. on pp. 79, 106).
- [151] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, pp. 1289–1306, Apr. 2006. DOI: 10.1109/TIT.2006.871582 (cit. on p. 79).
- [152] F. Chen, A. P. Chandrakasan, and V. Stojanovic, “A signal-agnostic compressed sensing acquisition system for wireless and implantable sensors,” *Custom Integrated Circuits Conference (CICC), 2010 IEEE*, IEEE, 2010, pp. 1–4 (cit. on p. 79).
- [153] S. Aviyente, “Compressed sensing framework for EEG compression,” *Statistical Signal Processing, 2007. SSP’07. IEEE/SP 14th Workshop on*, IEEE, 2007, pp. 181–184 (cit. on p. 79).
- [154] N. Li and M. Sawan, “High compression rate and efficient spikes detection system using compressed sensing technique for neural signal processing,” *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*, Apr. 2015, pp. 597–600. DOI: 10.1109/NER.2015.7146693 (cit. on p. 79).

- [155] X. Liu, M. Zhang, T. Xiong, A. G. Richardson, T. H. Lucas, P. S. Chin, R. Etienne-Cummings, T. D. Tran, and J. Van der Spiegel, "A fully integrated wireless compressed sensing neural signal acquisition system for chronic recording and brain machine interface," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, pp. 874–883, Aug. 2016. DOI: 10.1109/TBCAS.2016.2574362 (cit. on p. 79).
- [156] Â. C. Lapolli, B. Coppa, and R. Héliot, "Low-power hardware for neural spike compression in BMIs," *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jul. 2013, pp. 2156–2159. DOI: 10.1109/EMBC.2013.6609961 (cit. on p. 79).
- [157] C. Bulach, U. Bihl, and M. Ortmanns, "Evaluation study of compressed sensing for neural spike recordings," *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE, IEEE*, 2012, pp. 3507–3510 (cit. on p. 79).
- [158] T. Wu, W. Zhao, H. Guo, H. Lim, and Z. Yang, "A streaming PCA based VLSI chip for neural data compression," *2016 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, Oct. 2016, pp. 192–195. DOI: 10.1109/BioCAS.2016.7833764 (cit. on p. 79).
- [159] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, pp. 1098–1101, Sep. 1952. DOI: 10.1109/JRPROC.1952.273898 (cit. on p. 80).
- [160] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, Jul. 1948. DOI: 10.1002/j.1538-7305.1948.tb01338.x (cit. on p. 80).
- [161] R. Hashemian, "Memory efficient and high-speed search huffman coding," *IEEE Transactions on Communications*, vol. 43, pp. 2576–2581, Oct. 1995. DOI: 10.1109/26.469442 (cit. on p. 82).

-
- [162] L. L. Larmore and D. S. Hirschberg, “A fast algorithm for optimal length-limited huffman codes,” *Journal of the ACM*, vol. 37, pp. 464–473, Jul. 1990. DOI: 10.1145/79147.79150 (cit. on p. 82).
- [163] R. A. Freking and K. K. Parhi, “Low-memory, fixed-latency huffman encoder for unbounded-length codes,” *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers (Cat. No.00CH37154)*, vol. 2, Oct. 2000, 1031–1034 vol.2. DOI: 10.1109/ACSSC.2000.910670 (cit. on pp. 82, 83, 91).
- [164] J. N. Y. Aziz, K. Abdelhalim, R. Shulyzki, R. Genov, B. L. Bardakjian, M. Derchansky, D. Serletis, and P. L. Carlen, “256-channel neural recording and delta compression microsystem with 3d electrodes,” *IEEE Journal of Solid-State Circuits*, vol. 44, pp. 995–1005, Mar. 2009. DOI: 10.1109/JSSC.2008.2010997 (cit. on p. 83).
- [165] “Microsemi IGLOO series low-power FPGAs.” (), [Online]. Available: <https://www.microsemi.com/product-directory/fpgas/1689-igloo> (visited on 05/25/2020) (cit. on p. 83).
- [166] “CC3220sf data sheet, product information and support | TI.com.” (), [Online]. Available: <https://www.ti.com/product/CC3220SF> (visited on 06/23/2020) (cit. on p. 85).
- [167] A. Gomez-Marin and A. A. Ghazanfar, “The life of behavior,” *Neuron*, vol. 104, pp. 25–36, Oct. 9, 2019. DOI: 10.1016/j.neuron.2019.09.017 (cit. on p. 105).
- [168] T. Adame, A. Bel, B. Bellalta, J. Barcelo, and M. Oliver, “IEEE 802.11ah: The WiFi approach for m2m communications,” *IEEE Wireless Communications*, vol. 21, pp. 144–152, Dec. 2014. DOI: 10.1109/MWC.2014.7000982 (cit. on p. 106).
- [169] C. S. Mallory, K. Hardcastle, M. G. Campbell, A. Attinger, I. I. C. Low, J. L. Raymond, and L. M. Giocomo, “Mouse entorhinal cortex encodes a diverse repertoire of self-motion signals,” *Nature Communications*,

- vol. 12, p. 671, Jan. 28, 2021. DOI: 10.1038/s41467-021-20936-8 (cit. on p. 112).
- [170] P. A. Correia, E. Lottem, D. Banerjee, A. S. Machado, M. R. Carey, and Z. F. Mainen, “Transient inhibition and long-term facilitation of locomotion by phasic optogenetic activation of serotonin neurons,” *eLife*, vol. 6, N. Uchida, Ed., e20975, Feb. 14, 2017. DOI: 10.7554/eLife.20975 (cit. on p. 112).
- [171] C. Bossetti, J. Carmena, M. Nicolelis, and P. Wolf, “Transmission latencies in a telemetry-linked brain-machine interface,” *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 919–924, Jun. 2004. DOI: 10.1109/TBME.2004.827090 (cit. on p. 113).
- [172] M. W. Mathis and A. Mathis, “Deep learning tools for the measurement of animal behavior in neuroscience,” *Current Opinion in Neurobiology*, vol. 60, pp. 1–11, Feb. 2020. DOI: 10.1016/j.conb.2019.10.008 (cit. on p. 114).
- [173] D. R. Quiñones, A. Cuevas, J. Cambra, S. Canals, and D. Moratal, “RATT: RFID assisted tracking tile. preliminary results,” *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jul. 2017, pp. 4114–4117. DOI: 10.1109/EMBC.2017.8037761 (cit. on pp. 114, 120).
- [174] D. R. Quiñones, “Developing preclinical devices for neuroscience research in the fields of animal tracking, fMRI acquisition, and 3d histology cutting,” Ph.D. dissertation, Universitat Politècnica de València, Valencia, Spain, Jan. 16, 2019 (cit. on p. 114).
- [175] Uei-Ming Jow, P. McMenamin, M. Kiani, J. R. Manns, and M. Ghovanloo, “EnerCage: A smart experimental arena with scalable architecture for behavioral experiments,” *IEEE Transactions on Biomedical Engineering*, vol. 61, pp. 139–148, Jan. 2014. DOI: 10.1109/TBME.2013.2278180 (cit. on p. 118).