Report on the integration of a neuromorphic VLSI and a bidirectional BMI

Table of Contents

In this report we present how we implemented the integration of a state-dependent bidirectional BMI with a neuromorphic VLSI chip. In the Introduction section we describe the different steps we made in building a complete interconnected closed loop system as shown Figure 1 In the following section we report a modified and extended version of the join contribution that IIT and UZH partners gave to the 7th International IEEE EMBS Neural Engineering Conference held in Montpellier (France) on April 2015. To describe into details the components of the closed-loop system developed during this project. In the Technical annex we report the technical specifications of the Zynq2Neuro (Z2N), a Printed Circuit Board (PCB) that we developed to interface the evaluation board ZedBoard (ZB) with a custom daughterboard (DTB) for general test of neuromorphic chips (schematics available in folder *HW*).

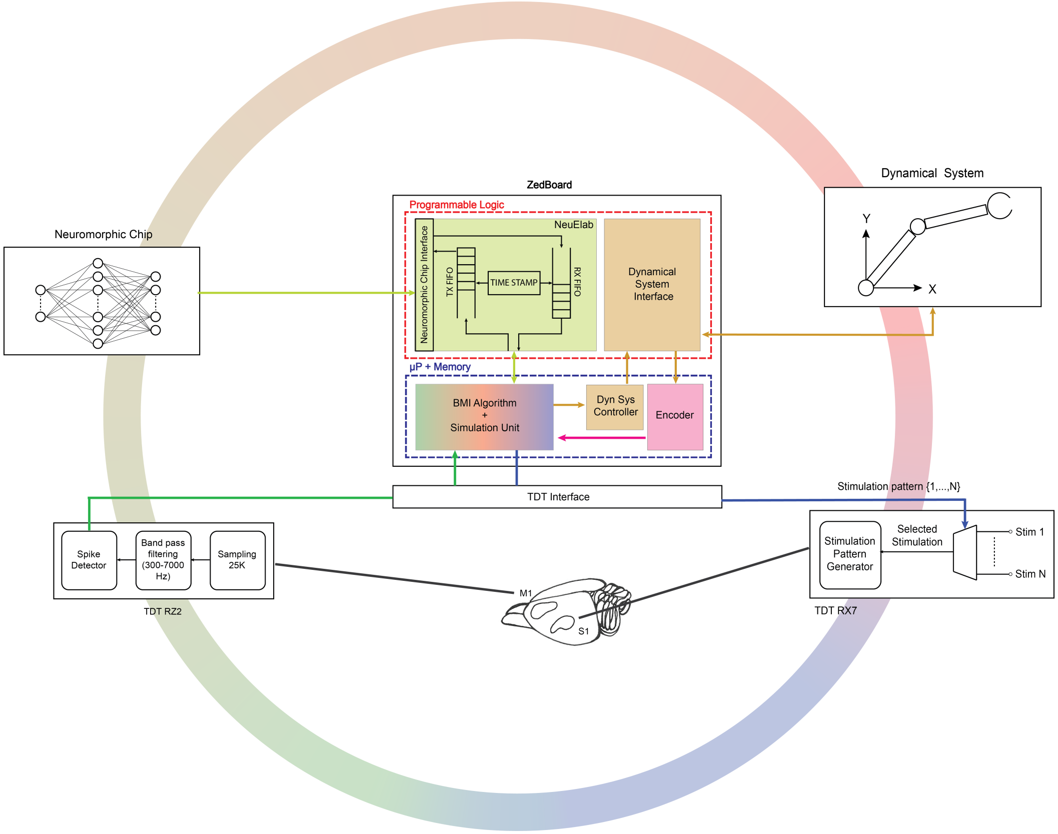


Figure 1: Scheme of the VLSI-based bidirectional Brain-machine interface. The core of the system is represented by a ZedBoard development board that communicates via UDP protocol with a Recording System (TDT RZ2 BioAmp) and a Stimulating System (TDT RX7 Stimulator). On the other side the ZedBoard is connected to the ROLLS neuromorphic processor that implements a perceptron neural network that, properly trained, is able to decode the neural signal coming from rat motor cortex. The result of the decoding stage is translated into a two-dimensional force which is converted into digital signals able to drive the motors installed on the 2 DOF robotic device. This latter communicates back to the software Encoder, implemented on the ZedBoard, its final state which is transformed into a stimulation pattern that is subsequently delivered by the Stimulation System into the somatosensory cortex of the subject.

The first step consisted in developing the C++/Python code on a Linux based PC that regulates the entire bBMI data flows. In particular in this first phase this algorithm loads from local memory a pre-elaborated neural dataset and, by means of a software emulator of a neuromorphic VLSI, it decodes the incoming activity into a force that is sent to a simulated Dynamical System. After a set amount of time the dynamical system communicates its final internal states to the bBMI that, according to this information, selects the next neural from the preloaded dataset.

In the second step we moved the above described code from the PC to a developmental board (i.e. ZedBoard – Zynq 7000). We reviewed all the scripts by cleaning and optimizing them in relation to the limited resource of the Zynq 7000 and we also build up the communication channels between the ZedBoard and both the physical dynamical system and the recording/stimulation apparatus. The two-dimensional commands to control the dynamical system are generated from the FPGA inside the Zynq and delivered from two different ZedBoard analog I/O pins to the system's motors in form of continuous Pulse-width modulation signals. On the other part the recording and stimulation systems (TDT systems) are connected by means of UDP communication to acquire the neural data and to transmit the stimulation patterns to be delivered to the brain. To allow the data exchange between ZedBoard and TDT devices we provided a software interface developed in Matlab language able to acquire/send data from/to the optic fiber of the acquisition/stimulation system rerouting them to/from its Ethernet port connected to the developmental board.

In the third phase of this process, we built the Zynq2Neuro (Z2N) hardware interface (see Technical Annex - Zynq2Neuro) that interconnects the ZINQ to the ROLLS neuromorphic chip. The Zynq FPGA embeds a dual-core ARM processor with 512MB RAM and 16GB SD Hard Drive and light Linux distribution called "Linaro" is installed. A component named *NeuElab* has been designed and upload on the Zynq in order to interface the ARM processor to the neuromorphic chip. A detailed user manual of the NeuElab software has been uploaded and it is available at the Si-Code Project web site (HTTP://www.sicode.eu/results/software.html).

Simultaneously it has been developed a driver for the NeuElab device able to manage the communication between the ZedBoard and the neuromorphic chip: by means of this tool we are able to set, stimulate and collect data from the spiking neural networks using high level C/C++ functions (drivers available in folder *SW/ZedBoardDrivers*). At the end of this stage we also wrote additional C/C++ library that further simplify the user to implement neural topography, perform training and test on the ROLLS chip (tools available in folder *SW/ZedbordTestTools*). NeuElab is interconnected to the serial biases programmer bus and to the bidirectional AER bus of the ROLLS. We also interfaced the biases programmer with the bias graphical interface and the AER bus with the *pyNCS viewer*, both belonging to the *pyNCS python package* developed by UZH partner.

The last step regarded the implementation of the state dependency algorithm described in the deliverable D4.3 inside the bBMI loop. It has been embedded directly in the TDT interface such that all the transformations performed by this process were transparent to the ZedBoard. In this way we can reduce the Ethernet bandwidth making our BMI faster and more responsive, giving in input to the

ZedBoard just the data strictly necessary for the decoding operations. In deliverable D8.3 we reported to the benefits of this strategy.

1 A modular configurable system for closed-loop bidirectional brain-machine interfaces

Extending bidirectional brain-machine interfaces (BMI) tailored for specific experiments with additional software and hardware tools can be very onerous, if not impossible. To overcome this problem, we developed a modular configurable system by modifying the architecture of an existing bidirectional BMI. This modular system enables the seamless and efficient inclusion of new features and the integration of new protocols without changing the native system's overall structure. By introducing a platform for the implementation of BMI algorithms on neuromorphic chips, this method represents a step towards the development of low-power, compact and computationally powerful tools for clinical applications.

1.1 Introduction

During the late nineties the technological progress for the interconnection of neural tissue with artificial devices was one of the main factors that led to a rapid growth of the research field known as brain-machine interfaces (BMIs) [1]. After almost twenty years, the unquestionable scientific results achieved in this frontier research also highlighted the hurdles of technological transfer of such results into clinical applications [2]. This is also due to the fact that BMIs comprise several dedicated modules, each tackling specific research and technological challenges. The coupling of the electrodes with the neural tissue, the computational capabilities of the processing unit, the performances of the decoding algorithms and the generation of motor commands to control external devices represent the main topics of different research fields. The algorithmic, software and hardware heterogeneity of the modules assembled together by the researchers in developing BMIs makes it difficult to ex- change knowledge or include additional modules developed by different laboratories and, at the same time, to compare the performances of these systems. In this paper, we present a new modular and configurable implementation of a closed loop bidirectional BMI system first tested with neural data from anesthetized animals [3] and later improved and tested with simulated data [4]. The modular configuration of this system creates a flexible structure in which new or updated versions of individual modules, even based on different communication protocols, can be inserted or replaced without having to change the whole system.

1.2 System Architecture

The architecture of a bidirectional BMI usually comprises five main components: an *Acquisition System* to record the neural signals, a *Decoder* to decode the recorded activity to be translated into motor commands, an external *Dynamical System*, an *Encoder* and *Stimulation System* with the goal of encoding the information collected from the environment and of translating it into stimuli to be delivered directly into the brain [5]. Despite the main components being similar throughout all of the existing closed-loop BMIs proposed so far, each of them depends on specific and mutually dependent hardware, software and communication modules, with low compatibility and, hence, flexibility. This is one of the main reasons why each lab usually built form scratch its own BMI and the technology-sharing rate is really low.

1.2.1 An existing bidirectional BMI

In this paper we consider a BMI that uses two microwire electrode arrays inserted into different regions of the cortex as described in [6]. The Acquisition System records, amplifies and performs an A/D conversion of the neural signal recorded from the electrodes placed in the motor cortex and it might also perform further signal processing, such as spike detection or sorting. Spike occurrences and other information are then sent to the Decoder that interprets and transforms them into proper commands to drive the movement of a Dynamical System that evolves for a certain time interval. Its corresponding final state is converted by the Encoder into an appropriate electrical stimulation pattern. The Stimulation System delivers it directly to the rat's sensory cortex by using a second microwire electrode array giving information about the actual status of the Dynamical System. The stimulation pattern is usually composed by a bi-phasic current pulse train of given amplitude and duration.

1.2.2 A modular bidirectional BMI

We propose a flexible, modular implementation based on a core, the *Managing Unit*, that coordinates the information flow across a series of modular satellites that complete/enhance the BMI itself as shown in Figure 2.

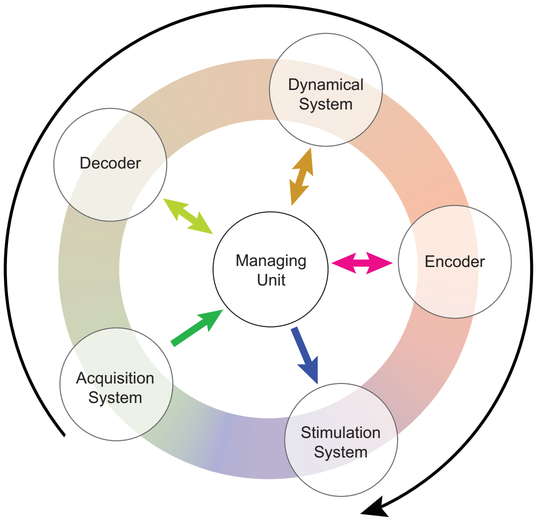


Figure 2 General scheme of the modular configurable system for bidirectional BMIs. On the rainbow donut are depicted a series of different satellites that communicates, in mono or bi directional way, with the central MU which manages their data flow. In particular the AS catches the neural signal that is collected from MU, which sends the rehashed data to the DE. It returns to the MU the result of the decoding policy that is translated into a command that drives the DS. When necessary the MU can retrieve the current state of the DS and through the EN it can decide for the most appropriate stimulation that SS will deliver to the brain.

Thanks to the Managing Unit, each satellite can be in- dependently instantiated on different platforms and during the development and prototyping phases, the corresponding algorithms can be explored and tested on software running on general purpose CPUs. At a later stage they can be progressively moved to more specialized hardware platforms, up to the implementation of dedicated modules such as the neuromorphic chips. In the prototyping phase the system trades off flexibility for power consumption and compactness that are instead optimized with the use of dedicated HW devices.

1.3 System Implementation

Figure 1 shows a possible instantiation of the described system architecture. It is composed by a ZedBoard hosting both the Managing Unit and the Encoder, a RZ2 BioAmp Processor (Tucker-Davis Technology, TDT, Gainesville, FL) as Acquisition System, a RX7 Stimulator Base Station (TDT) as Stimulation System, the Reconfigurable On-Line Learning Spiking (ROLLS) neuromorphic processor chip as Decoder and a planar two degrees of freedom robotic device as Dynamical System.

1.3.1 Managing and Encoder Unit

The Managing Unit represents the core of our system and we choose a development board named *ZedBoard* based on a Xilinx Zynq-7000 which is composed by a fully programmable logic (FPGA) and a Dual ARM Cortex™ A9 MPCore™ (667MHz). It has a 512 MB DDR3 memory and a 16GB SD card flash storage unit. Its huge connectivity (Ethernet, UART, several I/O ports and many others) and the possibility to have a microprocessor (μP) and a programmable logic on the same device makes the ZedBoard a perfect candidate to be the main core of a bidirectional BMI. The ZedBoard can be logically subdivided in two main parts: a microprocessor with a memory block and a Programmable Logic Module (FPGA). A lighter and faster version of the commercial operating system Linux Ubuntu runs on the μ P providing a series of C/C++ and Python built-in libraries and compilers for software development. We used it to host the core of the Managing Unit that regulates the flow of information across all of the satellites and the FPGA and that we called *BMI Algorithm* (C/C++ and Python code available in *SW/ZedBoardCode/BMIALgorithm*).

As the μ P is a general purpose hardware, it is also used to run some of the software modules of the brain- machine interface such as the Dynamical System Controller, the Encoder and the Simulation Unit, as shown in Figure 2.

a) The *Dynamical System Controller* translates the desired state of the Dynamical System into a proper command through the Dynamical System Interface that handles the communication to the specific robotic system.

b) The *Encoder* retrieves the final state of the Dynamical System and, through a look-up table, selects the most appropriate stimulation pattern that has to be delivered by the Stimulation System. In this first implementation we built the encoder such that there is a correspondence between each position of the controlled device with a stimulation pattern, as described in [3].

c) The *Simulation Unit* is composed by a series of Python and C/C++ functions for the simulation of the Decoder and Dynamical System modules; this modularity allows for flexible prototyping and algorithm development before porting the closed loop BMI to dedicated hardware. We implemented a software version of the Decoder module based on python scripts simulating a spiking neural networks for learning and classification, specifically designed to be hardware compliant. This setup allows us to test the behavior of the neuromorphic chip in a closed-loop system via a software simulation before connecting the hardware.

The μ P can also simulate a Dynamical System even in absence of the external robotic actuator, by directly calculating the temporal evolution of the system after the application of the decoded input.

Finally, the μ P imports and exports data from the Zed- Board Ethernet port using the User Datagram Protocol (UDP). This allows to run the entire closed loop process off line by importing a representative neural dataset and by disconnecting the Acquisition and Stimulation Systems.

The Programmable Logic Module is programmed by the μ P and comprises the NeuElab and the Dynamical System Interface modules.

The NeuElab module acquires the rehashed brain signals from the BMI Algorithm and routes them, in the proper way, to the Decoder and vice versa. The spikes are encoded with the Address Event

Representation (AER) protocol and sent to the Neuromorphic chip (Decoder), conversely, the chip output AER spikes are acquired and the resulting decoded response is sent to the Dynamical System Interface. The spikes are digital pulses generated (and sent) by the neurons asynchronously using the AER protocol. The information is in the identity of the firing neuron (address) and in the implicit timing between the spikes. When acquired on a clocked system, the time is explicitly associated to the address of the spike by the TimeStamp block. NeuElab is composed by two different FIFOs that drive the data flow from/to the ROLLS neuromorphic chip, the RX and TX Fifo. a) The *TX Fifo* is filled with the address of the neuron that shall receive the spike and the time relative to the other spikes, by associating a delay time value by the TimeStamp block. NeuElab will send each spike in the TX Fifo to the correct neuron at the correct time. b) The *RX Fifo* is filled with the spikes from the neurons of the ROLLS chip. The received couples of address and relative time stamp are then sent to the BMI algorithm that translates the recorded neural activity in a command for the Dynamical System.

The Dynamical System Interface module acquires from the Dynamical System Controller the commands to be translated into digital signals to drive the external device. If the Dynamical System is able to return feedback about its current state the Dynamical System Interface has to collect and redirect it to the Encoder. In the case in which the Dynamical System is driven in an open-loop fashion the current state will be communicated to the Encoder directly from the BMI Algorithm itself.

1.3.2 Acquisition and Stimulation Units

The TDT RZ2 BioAmp Processor band pass filters (300- 7000 Hz) the neural signal and performs an on-line spike detection based on the standard deviation of the signal RMS. Spike occurrences can be sent to the Managing Unit by using two different modalities: a) a direct UDP connection between the ZedBoard and the RZ2 b) a middleware composed by a Windows PC that receives the data coming from the RZ2 via optic fiber and routes it via UDP to the ZedBoard.

The TDT RX7 Stimulator Base Station can store several stimulation patterns and it is possible to associate for each of them a different stimulating electrodes. The TDT interface allows to setup the stimulation pattern parameters as duration, frequency and amplitude of the biphasic stimulation pulse train. During the closed-loop BMI functioning, the stimulation code sent by the Managing Unit is able to recall one of the stored stimulation patterns that is subsequently D/A converted into a biphasic current stimulus and directly delivered to the stimulating electrodes. The TDT interface allows to run built-in tools in order to supervise and manage the RZ2 and RX7 functioning.

In the proposed implementation we use a Tucker Davis system because of its high efficiency and reliability, how- ever it can be easily replaced by other electrophysiological devices systems either by plugging them into the Ethernet port of the ZedBoard (with the only workload of converting the neural data to the right format) or by adding a module to the middleware libraries that handles the compatibility of the communication and data format. This is generally true for all the satellites, not only for the Acquisition and Stimulation Systems.

1.3.3 Decoder Unit

Neuromorphic chips are dedicated devices that reproduce some of the key properties of neural computation such as event-driven, cooperative, context-dependent processing, learning, and adaptation [7]. They are best suited for low- power applications where noisy and uncontrolled signals need to be decoded in real-time, for example to control external devices. In the proposed architecture, a multi-neuron neuromorphic chip, the ROLLS neuromorphic processor [8], is deployed for decoding the neural signals recorded from the rat's brain and producing the signals that will drive the robotic actuator . The ROLLS processor is a full- custom mixed signal VLSI chip that uses low-power sub- threshold analog circuits to implement spiking neurons and biophysically realistic synapse dynamics, and asynchronous digital circuits to communicate the neurons action potentials via fast digital pulses. Its architecture comprises 256 adaptive exponential Integrate and Fire neurons [9] each receiving input currents from 520 synapses Out of all the synapses afferent to a neuron, 8 have digital programmable weights, 256 implement short term plasticity [10] and 256 spike-based learning circuits based on the model proposed in [11]. All synapses and neurons comprise additional digital circuits that can be used to configure both their individual properties (such as time constants, leak conductance, etc.) and the network internal connectivity. The on-chip programmable all-to-all connectivity allows the configuration of a wide range of neural networks topologies. The activity of the spiking neurons can be transmitted to off-chip devices in the form of prototypical address-events using the AER protocol [12]. In our setup we use the Zynq programmable logic to manage the flow of spikes from and to external devices (see Section III-A). In the proposed closed-loop BMI, the ROLLS neuromorphic processor is used for the classification and decoding of spike patterns recorded from the rat's motor cortex, either online from the TDT recording unit, or offline, loading stored data sets from the Managing Unit, for tuning and characterization of the decoding algorithm.

1.4 Conclusion

The way in which the technology advances is a great opportunity for neuroscientists as it allows them to easily access the latest available technological tools and overcome existing bottlenecks and limitations [13]. In this scenario, it is crucial to develop systems that are flexible enough to easily include the latest technologies and tools in the part of the system already developed, without changing the entire architecture. An optimal solution is represented by modular systems, in which each element can be developed and modified independently from the others and a central unit manages the exchange of information across each module. In this paper we presented an example of such a configurable architecture for developing closed-loop bidirectional brain- machine interfaces. We reconfigured an existing BMI system into a new architecture that allows us to modify, substitute or add new features without having to restart from the beginning and without giving up performances. Notably when we configured the system to reproduce the experiment presented in [3] obtaining results comparable with the previous work. The whole concept of the modular BMI has been developed *around* a central core that it is capable of managing the information flow across several modules that can be added to the main systems, managing also different protocols. In this way it is easy, for example, to simulate the behavior of the dynamical system, or of the decoding algorithm, before plugging inserting them into the closed-loop. For the first time we showed a BMI closed-loop system capable of integrating modules based on the AER protocol and, hence, exploiting the emergent neuromorphic technology. The proposed system represents a powerful developmental platform for BMI research, specifically for testing the implementation of BMI algorithms on neuromorphic hardware. This approach has the potential of delivering compact, low- power and computationally powerful devices essential for transferring the BMI research into clinical products.

These features are essential for those clinical applications that need to use devices implanted in proximity of the brain under the skull that use wireless communication protocols to transfer the data outside the body [14].

Eventually, this approach will be useful for researchers that want to use updated technology and, more interestingly, it represents a good solution which increases the possibility of exchanging knowledge between different laboratories working in the same field.

1.5 References

[1] J. Wander and R. Rao, “Braincomputer interfaces: a powerful tool for scientific inquiry,” 2014.

[2] G. Baranauskas, “What limits the performance of current invasive brain machine interfaces?” *Frontiers in systems neuroscience*, vol. 8, 2014.

[3] A. Vato, M. Semprini, E. Maggiolini, F. D. Szymanski, L. Fadiga, S. Panzeri, and F. A. Mussa-Ivaldi, “Shaping the dynamics of a bidirectional neural interface,” *PLoS computational biology*, vol. 8, no. 7, 2012.

[4] A. Vato, F. D. Szymanski, M. Semprini, F. A. Mussa-Ivaldi, and S. Panzeri, “A bidirectional brain-machine interface algorithm that approximates arbitrary force-fields,” *PloS one*, vol. 9, no. 3, 2014.

[5] E. C. Leuthardt, G. Schalk, D. Moran, and J. G. Ojemann, “The emerging world of motor neuroprosthetics: a neurosurgical perspec- tive.” *Neurosurgery*, vol. 59, no. 1, pp. 1–14; discussion 1, 2006.

[6] F. A. Mussa-Ivaldi, S. T. Alford, M. Chiappalone, L. Fadiga, A. Karniel, M. Kositsky, E. Maggiolini, S. Panzeri, V. Sanguineti, M. Semprini *et al.*, “New perspectives on the dialogue between brains and machines,” *Frontiers in neuroscience*, vol. 4, 2010.

[7] C. Bartolozzi and G. Indiveri, “Synaptic dynamics in analog VLSI,” *Neural Computation*, vol. 19, no. 10, pp. 2581–2603, Oct 2007.

[8] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, and G. Indiveri, “A reconfigurable on-line learning spiking neuromorphic processor,” *Frontiers in Neuroscience*, 2015, submitted.

[9] R. Brette and W. Gerstner, “Adaptive exponential integrate-and-fire model as an effective description of neuronal activity,” *Journal of Neurophysiology*, vol. 94, pp. 3637–3642, 2005. [10] F. Chance, S. Nelson, and L. Abbott, “Synaptic depression and the temporal response characteristics of V1 cells,” *The Journal of Neuroscience*, vol. 18, no. 12, pp. 4785–99, 1998.

[11] J. Brader, W. Senn, and S. Fusi, “Learning real world stimuli in a neural network with spike-driven synaptic dynamics,” *Neural Computation*, vol. 19, pp. 2881–2912, 2007. [12] E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri, “Neuromorphic electronic circuits for building autonomous cognitive systems,” *Proceedings of the IEEE*, vol. 102, no. 9, pp. 1367–1388, Sep 2014.

[13] L. F. Nicolas-Alonso and J. Gomez-Gil, “Brain computer interfaces, a review,” *Sensors*, vol. 12, no. 2, 2012.

[14] G. N. Angotzi, F. Boi, S. Zordan, A. Bonfanti, and A. Vato, “Aprogrammable closed-loop recording and stimulating wireless system for behaving small laboratory animals,” *Scientific reports*, vol. 4, 2014.

2 Technical Annex - Zynq2Neuro

2.1 Description

The Zynq2Neuro (Z2N) is a Printed Circuit Board (PCB) that interfaces the evaluation board ZedBoard (ZB) (http://www.zedboard.org/product/zedboard) with custom daughterboard (DTB) for general test of neuromorphic chips, with the aim of supporting the characterization and development of such devices for custom embedded applications. The aim is to manage, program and interface up to 2 DTBs. The supported features are:

a. Program on-chip bias generator
b. Program external DAC to bias the neuromorphic chips (only supported for one DTB)
c. Power supply the chip
d. Send and receive AER data
e. Support additional chip functionality by IO expanders and I2C protocol

The board has been designed for maximum flexibility and possibility of expansion, in order to accommodate different flavors of neuromorphic chips. The pinout supports different kinds of Bias Generators and the AER addressing space is expanded up to 30 bits (configurable as inputs or outputs).

The Z2N can support logical levels, power supply and biases from Digital to Analog Converters of 3.3V OR 1.8V, as selected from the first DTB. This means that the DTB will host chips that are homogeneous for the logical levels. In general the Z2N can support chips fabricated on the 350nm process (3.3V) and 180nm process of the latest generation (1.8V and mixed 1.8V/3.3V).

To optimize the design, AER address lines are shared among the DTBs, that share a common and some bits of the Bias Generator programming, I2C and I/O expander. The sharing of the AER address lines is based on the assumption that they are put in tri-state when the chip is not sending or receiving an event. This is guaranteed by the chips that support the SCX protocol, but can be supported also by the chips supporting the P2P protocol by adding buffers on the DTB driven by the handshake signals (ACK) from the ZedBoard. The correct addressing of the event to the input of the chip is guaranteed by the reserved handshake signal (REQ) line that targets only the receiving chip.

The Z2N specifically targets compatibility with the chips used within the Si-Code project (such as ROLLS and SpikeBetter), but is a more general tool for most of existing neuromorphic chips. This system therefore supports at the hardware level the possibility to interface with such devices. The NeuElab FPGA IP allows for the configuration and AER interfacing of the different devices with the Zynq embedded system. Such IP has been designed to allow as well the interfacing to different software infrastructures for interfacing with neuromorphic chips, that can now be interfaces with PyNCS, YARP and standard C++.

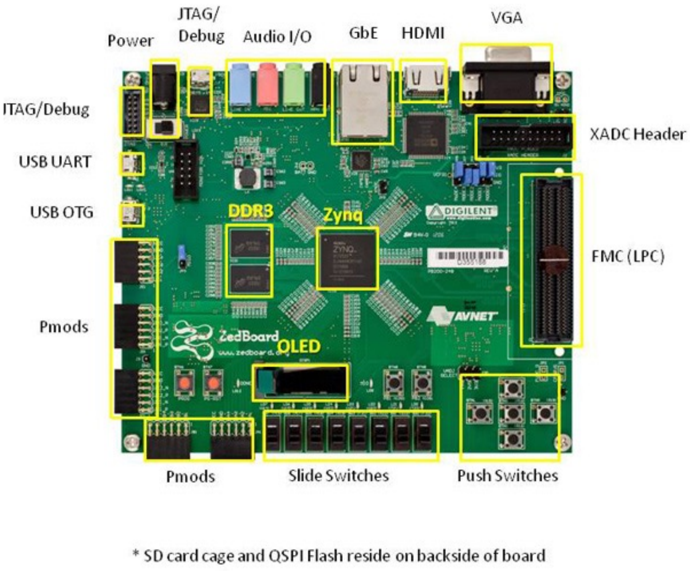


Figure 3: Complete scheme of the development board ZedBoard with highlighted the most important parts.

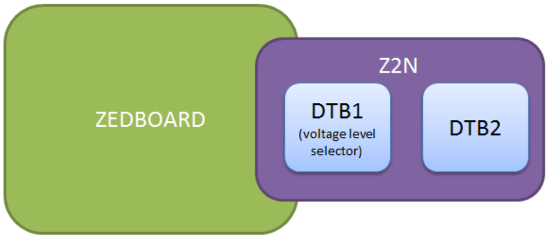


Figure 4: Block diagram of the system with ZedBoard, Zinq2Neuro and the two daughterboards.

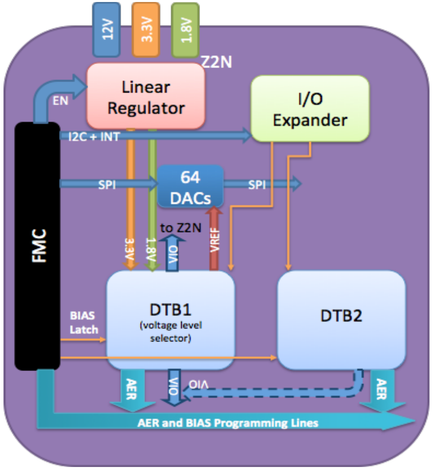


Figure 5: Detailed block diagram of the Zynq2Neuro module.

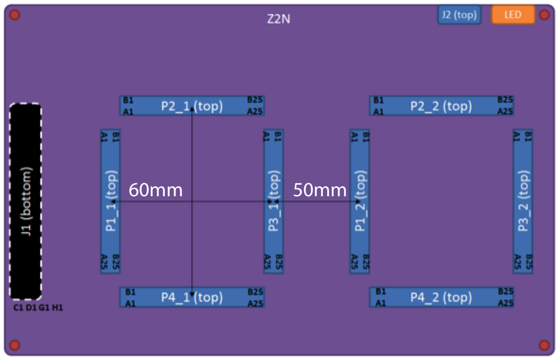


Figure 6: Detailed scheme of the motherboard indicating the real size and the position of the connectors.

2.2 Daughterboard DTB

The DTB host neuromorphic chips. Each has its own specifications, however each is connected to the Z2N mother board through 50 pin SAMTEC at fixed positions, as shown in Fig. 4.

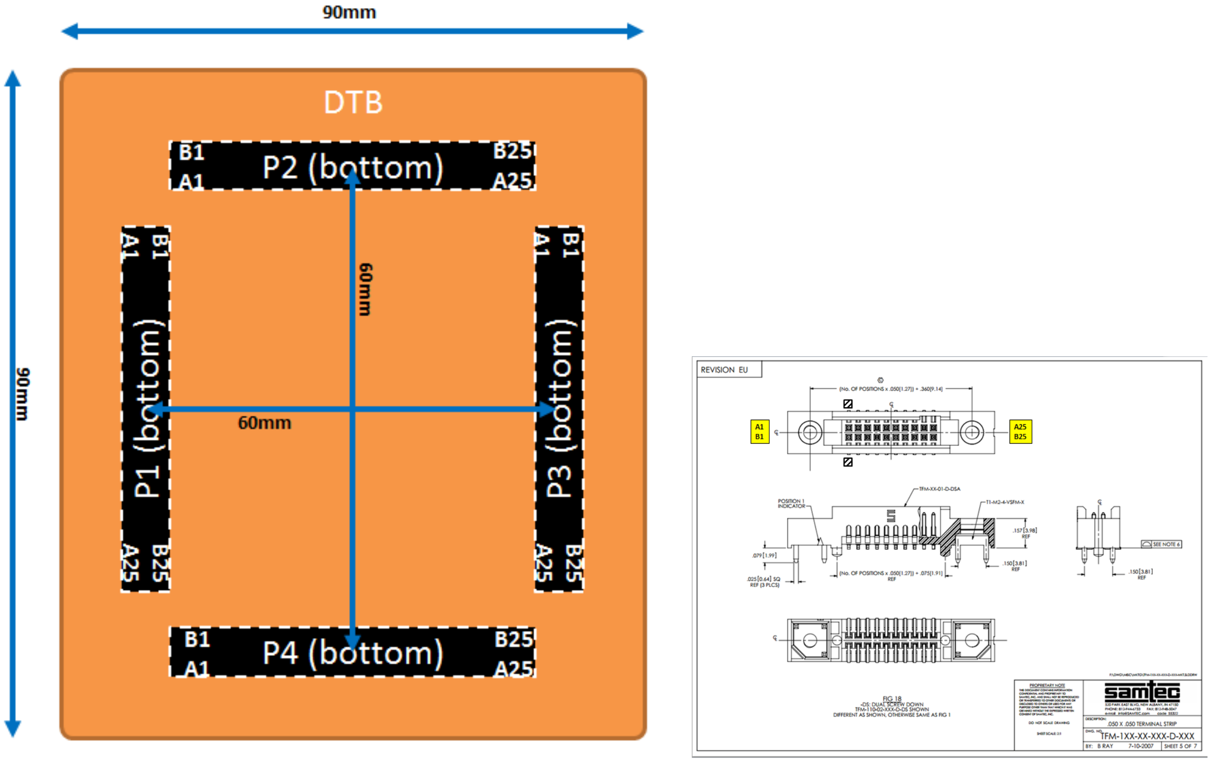


Figure 7: (left) Detailed scheme of the motherboard indicating the real size and the position of the TFM-125-01-L-D-A connectors (right).

Connectors P1, P3 are used for digital I/O and power supply, P2 and P4 are used for analog power supply and biases. Below is the pinout of each connector, implemented also in the template DTB schematics.

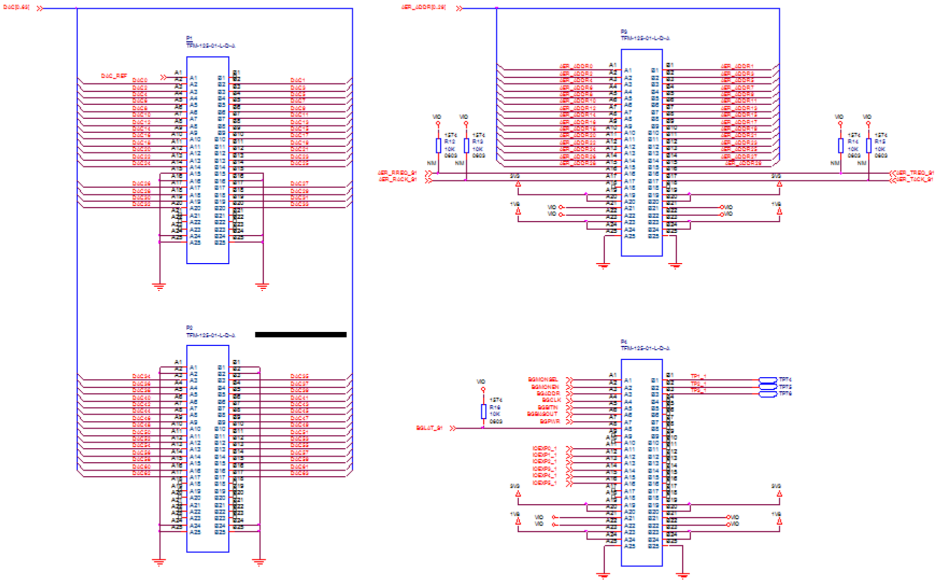


Figure 8: Pinout of the 4 connectors of the DTB. Zynq2Neuro

2.3 Connectors

The interface to the ZedBoard is the J1 FMC (LPC) connector. The Z2N will need external power supply, as explained below.

2.3.1 FMC LPC

The Z2N board is connected to the ZedBoard via the FMC connector. Figure 8 shows the pinout of the connector on the Z2N.

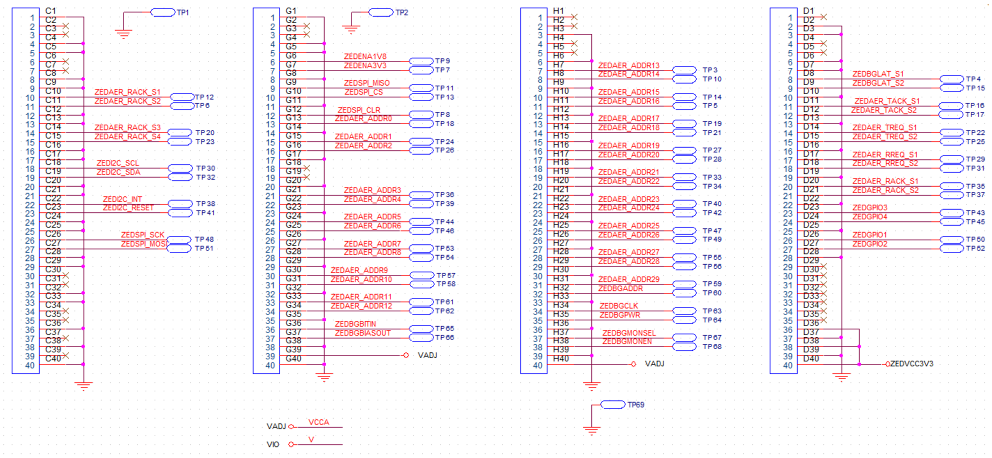


Figure 9: FMC LPC connector on the ZynQ2N board. It routes all of the signals to and from the ZedbBoard, being the unique gateway to access the neuromorphic chips mounted on the DTBs of the Z2N board.

2.3.2 Slot 1

The Z2N board hosts up to 2 DTBs. The connectors (Samtec SFM-125-T1-S-D-A) placement corresponds to the placement on the DTB.

The DTB in slot 1 is a master DTB, that sets the logical levels of the digital level shifters and the output range of the DAC. This is done by sending to the DTB both 3V3 and 1V8, then on the DTB the correct value has to be connected to pin A1 of P1 (dac_ref) and A22, A23, B22, B23 of P2 and P4 (vio) to set the Z2N signals voltage levels.

Next figure shows the pinout of DTB in slot 1.

The DTB in slot 1 is equipped with:

- 64 DACs (4 SPI lines for addressing the 8x8 DACs)
- BIAS Programming (7 lines)
- BIASLatch_1 (1 line)
- IOX_1 lines (2 lines I2C + 1 INT for the I/O expander)
- AER addresses (30 lines)
- REQX1 and ACKX1 (4 lines)

The DTB in slot 2 is equipped with:

- BIAS Programming (7 lines)
- BIASLatch_2 (1 line)
- IOX_2 lines (2 lines I2C + 1 INT for the I/O expander)
- AER addresses (30 lines)
- REQX2 and ACKX2 (4 lines)

As already explained some lines are common among the two DTBs, here they are shown in purple for clarity.

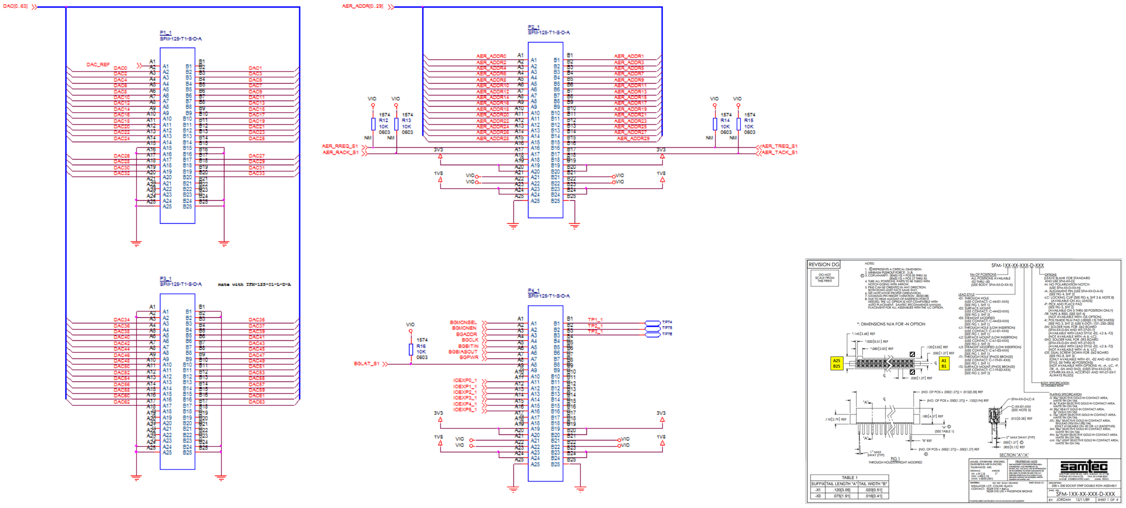


Figure 10: (left) DTB in slot 1, Pinout of 4 connectors of the DTB in slot 1. (right) connectors SFM-125-T1-S-D-A on the Z2N to connect the DTB.

2.3.3 Power supply

The Z2N board generates internally two power supplies, 3V3 (2A) and 1V8(2A), by means of linear voltage regulators with ENABLE signals. The power (5V) is provided externally (Molex 43650-0313 connector to an external power supply) and the ENABLE is driven by the ZedBoard trough the NET **ZEDENA1V8** and **ZEDENA3V3** (pins **G6** and **G7** of the connector **J1**).

A 1.5A fuse is mounted on the 5V line.

2.3.4 DAC

P2 and P4 are connected to 64 analog voltages generated by 8 DAC (LTC2600), programmed through SPI and connected in daisy chain. The output level of the DAC can be set to 3.3V or to 1.8V by setting in the DTB the **DAC_REF** signal (connector **P1_1**, pin **A1**). The SPI signals in **J1** are:

Pin Number	Net Name	Description
C26	ZEDSPI_SCK	SPI SCK for the DACs
C27	ZEDSPI_MOSI	SPI MOSI for the DACs
G9	ZEDSPI_MISO	SPI MISO for the DACs
G10	ZEDSPI_CS	SPI chip select for the DACs
G12	ZEDSPI_CLR	SPI Clear for the DACs

2.3.5 Digital I/O

All of the digital signals from and to the ZB have a bidirectional level shifter 2V5 to VIO. VIO is set by the DTB in slot 1 and can be 3.3V, 2.5V or 1.8V. VIO is set by the DTB in the SLOT1 trough the connector **P2_1** or **P4_1** pins **A21-A22**, **B21-B22**.

2.3.6 Digital I/O - AER

The AER signals for handshake (transmit/receive req/ack) are dedicated for each DTB, while the address lines are shared, this means that if the chip is based on SCX protocol, then the address lines can be connected directly to the Z2N, otherwise the DTB has to mount an additional component to put the address lines in tri-state.

2.3.7 Digital I/O - BIAS generator

There are 7 digital lines to program the on-chip bias generator. These are shared by both DTB slots. One of them (the LATCH signal) has to be connected separately to each DTB to decide which shall be programmed. In this way we support different bias settings for each chip and different chips for each DTB, as long as they share the same power supply and relative digital and analog signal levels. The pins of the connector **J1** are:

Pin Number	Net Name	Description
C26	ZEDSPI_SCK	SPI SCK for the DACs
C27	ZEDSPI_MOSI	SPI MOSI for the DACs
G9	ZEDSPI_MISO	SPI MISO for the DACs
G10	ZEDSPI_CS	SPI chip select for the DACs
G12	ZEDSPI_CLR	SPI Clear for the DACs

2.3.8 Digital I/O - Configuration Bits

6 I/O configuration bits for each SLOT (used for additional digital signals, if required). I/O expander based on I2C with 24 I/O. The pins in the connector J1 to control the I/O expander are:

Pin Number	Net Name	Description
C18	ZEDI2C_SCL	I2C clock for the IO Expander
C19	ZEDI2C_SDA	I2C data in/out for the IO Expander
C22	ZEDI2C_INT	I2C interrupt for the IO Expander
C23	ZEDI2C_RESET	I2C reset for the IO Expander

The digital I/O goes to the SLOT 1 and 2 of the Z2N respectively trough the connector **P4_1** and **P4_2**:

P4_1

Pin Number	Net Name	Description
A11	IOEXP0_1	General purpose IO
A12	IOEXP1_1	General purpose IO
A13	IOEXP2_1	General purpose IO
A14	IOEXP3_1	General purpose IO
A15	IOEXP4_1	General purpose IO
A16	IOEXP5_1	General purpose IO

P4_2

Pin Number	Net Name	Description
A11	IOEXP0_2	General purpose IO
A12	IOEXP1_2	General purpose IO
A13	IOEXP2_2	General purpose IO
A14	IOEXP3_2	General purpose IO
A15	IOEXP4_2	General purpose IO
A16	IOEXP5_2	General purpose IO

2.4 Glossary

ZB	**Z**ed**B**oard
DTB	**D**aughter**B**oard
Z2N	**Z**ynQ**2N**euro

3 Experimental Setup

In this section we present several pictures showing the experimental set-up used to run the VLSI-Based bidirectional brain-machine interface.

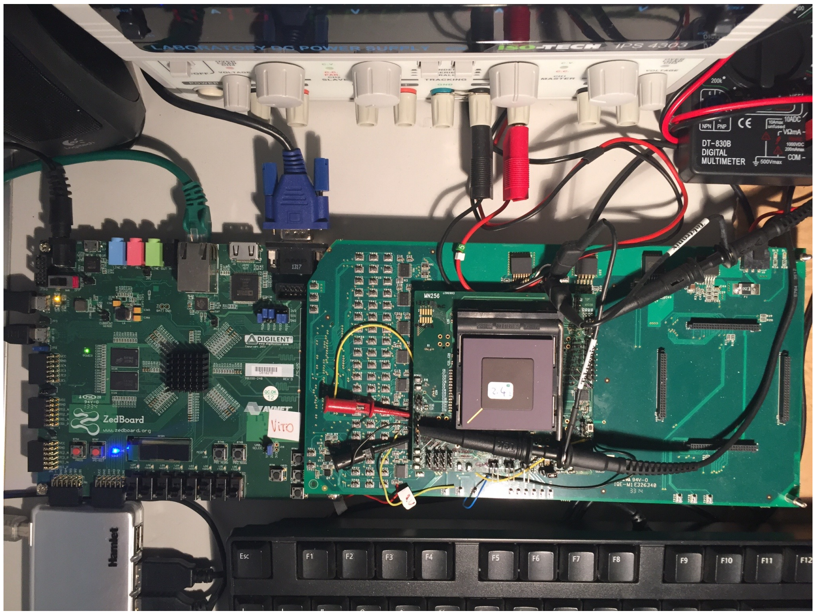


Figure 11 : A global view of the experimental set up with the Zedboard (left) and the neuromorphic chip (right) connected with the custom made Zynq2Neuro PCB board.

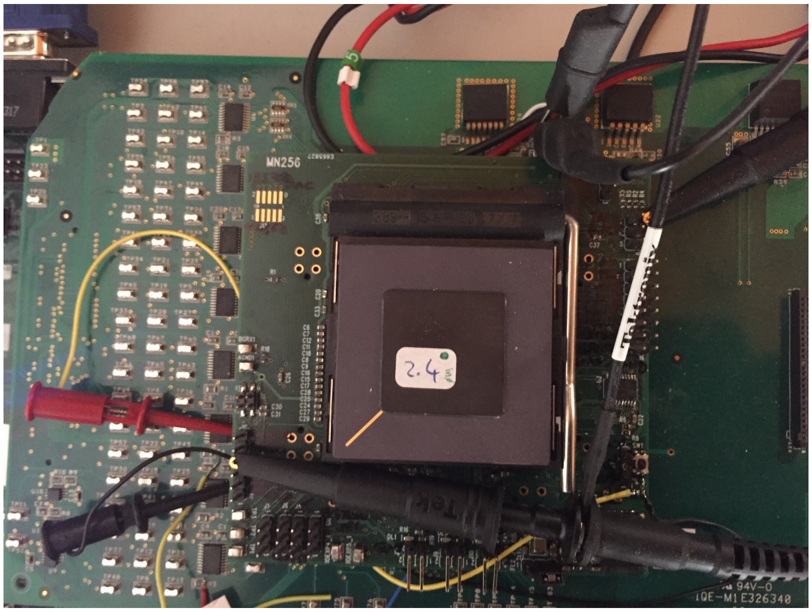
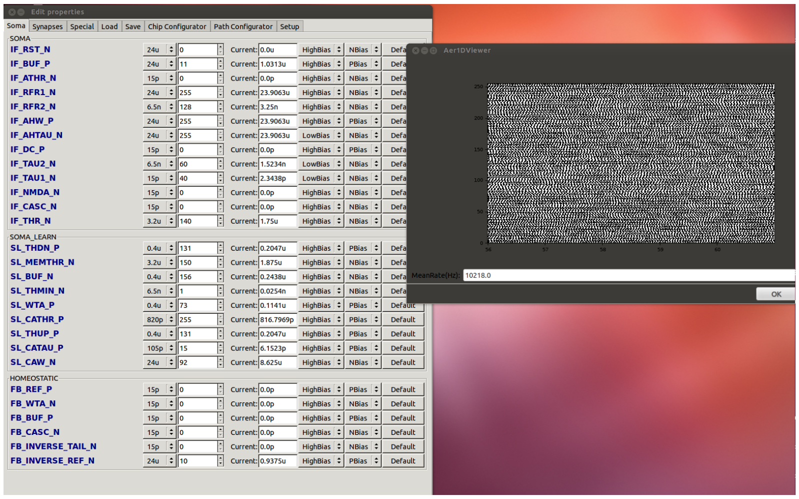


Figure 12: A closer view of the Daughterboard DTB with the neuromorphic chip mounted.



SOMA

IF_RST_N	24u	0	Current:	0.0u	HighBias	NBias	Defa
IF_BUF_P	24u	11	Current:	1.0313u	HighBias	PBias	Defa
IF_ATHR_N	15p	0	Current:	0.0p	HighBias	NBias	Defa
IF_RFR1_N	24u	255	Current:	23.9063u	HighBias	NBias	Defa
IF_RFR2_N	6.5n	128	Current:	3.25n	HighBias	NBias	Defa
IF_AHW_P	24u	255	Current:	23.9063u	HighBias	PBias	Defa
IF_AHTAU_N	24u	255	Current:	23.9063u	LowBias	NBias	Defa
IF_DC_P	15p	0	Current:	0.0p	HighBias	PBias	Defa
IF_TAU2_N	6.5n	60	Current:	1.5234n	LowBias	NBias	Defa
IF_TAU1_N	15p	40	Current:	2.3438p	LowBias	NBias	Defa
IF_NMDA_N	15p	0	Current:	0.0p	HighBias	NBias	Defa
IF_CASC_N	15p	0	Current:	0.0p	HighBias	NBias	Defa
IF_THR_N	3.2u	140	Current:	1.75u	HighBias	NBias	Defa

SOMA_LEARN

SL_THDN_P	0.4u	131	Current:	0.2047u	HighBias	PBias	Defa
SL_MEMTHR_N	3.2u	150	Current:	1.875u	HighBias	NBias	Defa
SL_BUF_N	0.4u	156	Current:	0.2438u	HighBias	NBias	Defa
SL_THMIN_N	6.5n	1	Current:	0.0254n	HighBias	NBias	Defa
SL_WTA_P	0.4u	73	Current:	0.1141u	HighBias	PBias	Default
SL_CATHR_P	820p	255	Current:	816.7969p	HighBias	PBias	Default
SL_THUP_P	0.4u	131	Current:	0.2047u	HighBias	PBias	Default
SL_CATAU_P	105p	15	Current:	6.1523p	HighBias	PBias	Default
SL_CAW_N	24u	92	Current:	8.625u	HighBias	NBias	Default

HOMEOSTATIC

FB_REF_P	15p	0	Current:	0.0p	HighBias	PBias	Default
FB_WTA_N	15p	0	Current:	0.0p	HighBias	NBias	Default
FB_BUF_P	15p	0	Current:	0.0p	HighBias	PBias	Default
FB_CASC_N	15p	0	Current:	0.0p	HighBias	NBias	Default
FB_INVERSE_TAIL_N	15p	0	Current:	0.0p	HighBias	NBias	Default
FB_INVERSE_REF_N	24u	10	Current:	0.9375u	HighBias	PBias	Default

MeanRate(Hz): 10218.0

Figure 13: Bias GUI and spiking activity monitor of the ROLLS chip

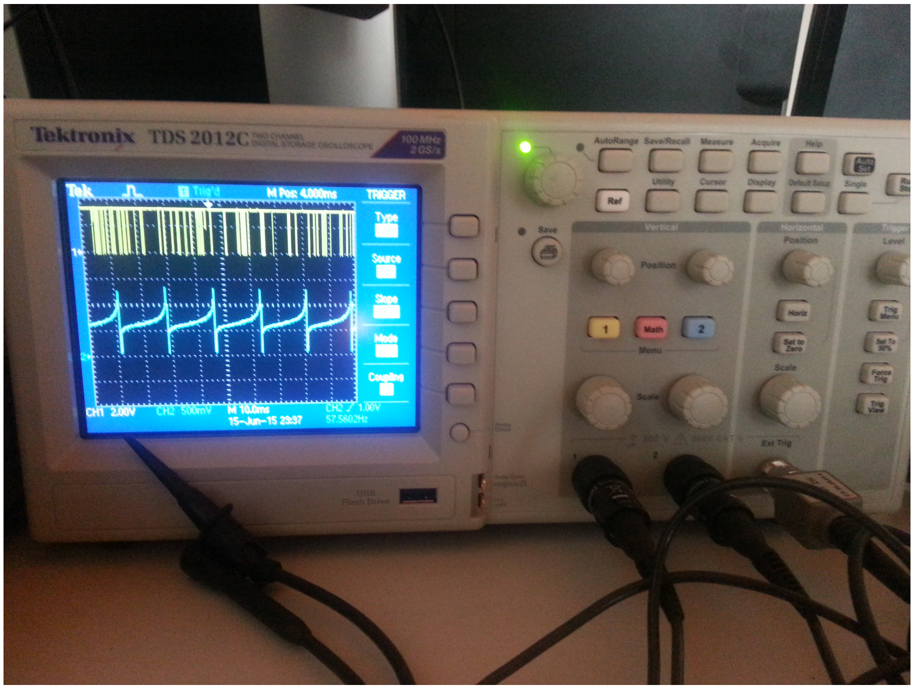


Figure 14 : Output Spiking Activity of the ROLLS chip and membrane potential of one artificial neuron.

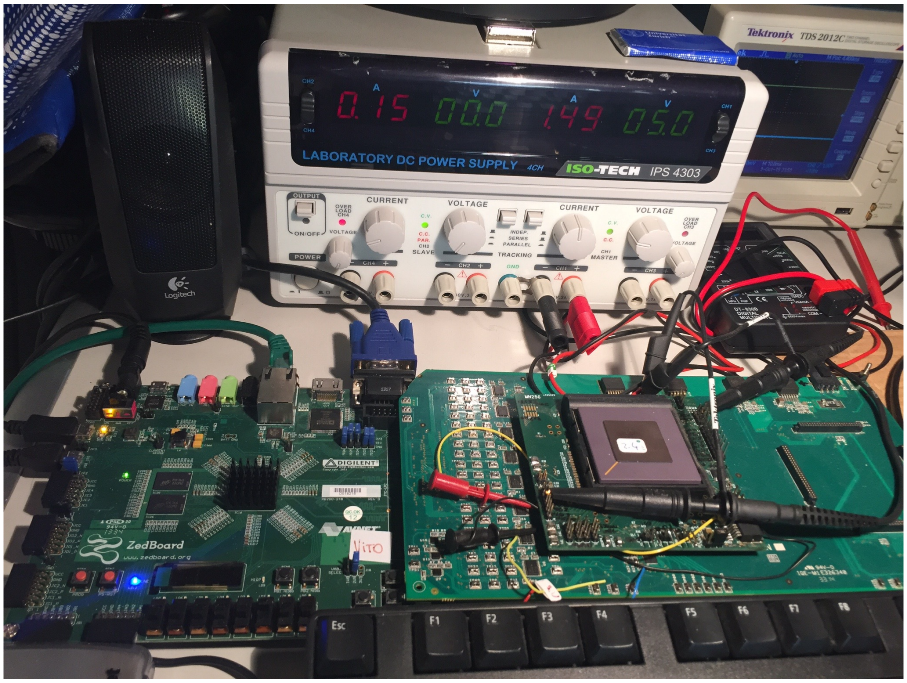


Figure 15: A global view of the experimental set-up .