Characterization of a Raspberry Pi as the Core for a Low-cost Multimodal EEG-fNIRS Platform

Freddy del Angel Arrieta[1], Michelle Rojas Cisneros[1], Jesús Joel Rivas[1], Luis R. Castrejón[2], Luis Enrique Sucar[1], *SMIEEE*, Javier Andreu-Perez[3 4], *SMIEEE*, Felipe Orihuela-Espina[1]

***Abstract*— Poor understanding of brain recovery after injury, sparsity of evaluations and limited availability of healthcare services hinders the success of neurorehabilitation programs in rural communities. The availability of neuroimaging capacities in remote communities can alleviate this scenario supporting neurorehabilitation programs in remote settings. This research aims at building a multimodal EEG-fNIRS neuroimaging platform deployable to rural communities to support neurorehabilitation efforts. A Raspberry Pi 4 is chosen as the CPU for the platform responsible for presenting the neurorehabilitation stimuli, acquiring, processing and storing concurrent neuroimaging records as well as the proper synchronization between the neuroimaging streams. We present here two experiments to assess the feasibility and characterization of the Raspberry Pi as the core for a multimodal EEG-fNIRS neuroimaging platform; one over controlled conditions using a combination of synthetic and real data, and another from a full test during resting state. CPU usage, RAM usage and operation temperature were measured during the tests with mean operational records below 40% for CPU cores, 13.6% for memory and 58.85 ° C for temperatures. Package loss was inexistent on synthetic data and negligible on experimental data. Current consumption can be satisfied with a 1000 mAh 5V battery. The Raspberry Pi 4 was able to cope with the required workload in conditions of operation similar to those needed to support a neurorehabilitation evaluation.**

I. Introduction

Stroke represents a major cause of disability world-wide requiring continuum of care for survivors. Stroke imposes a handicap to those affected. Neurorehabilitation programs are aimed to reduce dependency of stroke survivors on third parties and afford them independent living. These programs foster after-injury plasticity and sheering the brain's functional reorganization. The success of these programs is constrained by (a) our limited understanding of brain reorganization mechanisms after the insult and (b) the sparsity of behavioural evaluations recorded during ward visits. In addition, neurorehabilitation programs are extremely costly [1] and require highly specialized carers that aren't always available in rural settings, in practice being almost unaffordable and unreachable to large portion of the population of developing countries.

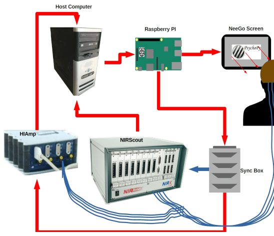
Corresponding author: f.orihuela-espina@inaoep.mx
[1]Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, Mexico 72840 [2]Hospital Universitario Benemérita Universidad Autónoma de Puebla, Puebla, Mexico 72410 [3]School of Computer Science and Electronic Engineering, University of Essex, Essex, UK CO4 3SQ[4]Department of Computer Sciences, University of Jaén, Jaén, Spain 23071

Electroencephalography (EEG) and functional near infrared spectroscopy (NIRS) are complementary neuroimaging modalities to interrogate brain activity and associated haemodynamics respectively. They are both portable, non-invasive and cheaper than MRI, MEG or PET. EEG and fNIRS are now viable in naturalistic settings and their cost is a fraction of what used to be [2], [3]. Combined multimodal operation is possible [4] and has been used before to provide insights of neurophysiological processes occurring during (gait) neurorehabilitation after stroke [5]. However, operation and interpretation of readings require skilled personnel. Studies have demonstrated the benefits of access to neuroimaging tools in rural settings in middle- and low-income countries for purposes other than neurorehabilitation [6]. This research intends to ameliorate the demands of neurorehabilitation assessment by affording a low-cost smart EEG-NIRS station that can be deployed to rural communities. Smart EEG-NIRS stations can yield observations and interpretations for neurorehabilitation beyond current proxy behavioural scores, and at a cost which can be made affordable from first level hospitals to rural clinics. Embedded artificial intelligence (AI) can improve our interpretability of concurrent EEG-NIRS recordings in the recovering brain and reduce instrumental costs, enhancing post-stroke monitoring, increasing affordability, and boosting accessibility to otherwise marginalized population.

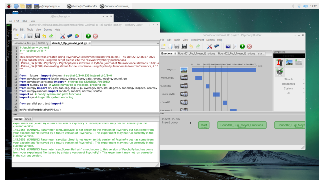
Here, we test the feasibility of a Raspberry Pi 4 to act as the core of a low-cost multimodal EEG-NIRS neuroimaging station. We show that the Raspberry Pi 4 is capable of dealing with presenting the neurorehabilitation evaluation stimuli, and the concurrent acquisition of data from streaming EEG and fNIRS devices as well as being responsible for synchronization and annotation of data all at once, whilst still leaving sufficient computational power to handle AI at a later stage.

II. Methods

The multimodal EEG-fNIRS neuroimaging platform is based on a Raspberry Pi 4 with 4GB of RAM and a Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) processor running Raspberry Pi OS 10.0 (previously Raspian). A desktop version of the operating system is necessary for supporting PsychoPy. For the experiments presented here, for imaging we have used a HIAmp EEG 128 channel system (g.tec, Austria) -although not all channels were used- and a NIRScout (NIRx, USA), a modular, and robust lab-based fNIRS device with a total of 16 laser sources with 2 wavelengths (750mm, and 860mm) and 8 detectors set up

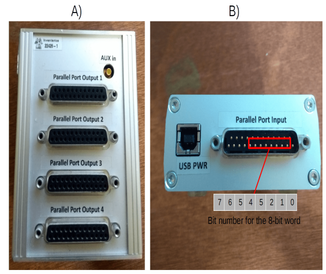


(a) Platform

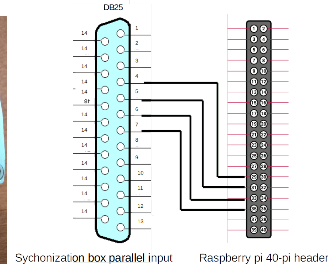


(b) PsychoPy on Raspberry

Fig. 1: Multimodal EEG-fNIRS Platform. (a) Raspberry based setup. (b) PsychoPy running on the Raspberry pi.



(a)



(b)

Fig. 2: (a) Synchronization box: A) Top view B) Back view and pins for 8-bit word. (b) Connection to the Raspberry Pi.

as 24 channels for all the experiments on this work. These middle range imaging devices will be substituted by low cost alternatives (e.g. OpenBCI and NinjaNIRS) later. The setup diagram is depicted on Fig. 1a.

A. *PsychoPy for stimuli presentation in a Raspberry Pi*

Experimental stimuli are presented using PsychoPy [7] to create and project the stimuli to the subject. The Raspberry Pi is not designed to manage heavy load graphics. This represents a problem for the most recent version of PsychoPy which relies on the capacity of heavy graphics handling from the hardware. Hence, an older PsychoPy (version 1.83.04) was compiled for the Raspberry pi (Fig. 1b). The OpenGL driver was manually activated before operating it.

Experimental video stimuli are divided in 2 groups; stimuli oriented to neurorehabilitation assessment of the hand based on the corresponding Fugl-Meyer sub-scale (FMA) [8]; and those oriented to maintain attention and elicit motivation and therapy adherence – Emotion Elicitation (EE). The videos accompanying the FMA stimuli were recorded in house. The EE set of stimuli are part of the affective computing dataset MAHNOB-HCI [9]. All videos were made to last exactly 20 seconds.

B. *Data Synchronization*

For trigger synchronization, a parallel port replicator (PPR) was used (Fig. 2a). The box takes an 8-bit word through the pins marked on Fig. 2a, and replicates it to the four parallel port outputs. The PPR synchronization box accepts Transistor–Transistor Logic (TTL) level signals. A voltage level ranging from 1.8 to 5 volts can be used to send a logic 1 to the box. The Raspberry Pi 4 model B has a 40 pin header from which only 24 pins corresponds to the General purpose Input/Output (GPIO) available for its use through code. The schematic diagram of the connection between the Raspberry Pi and the synchronization box is shown on Fig. 2b. GPIO pins from the Raspberry pi are capable of 3.3v outputs but with very low currents limited to 50mA. However, the resistors on the input stage of the PPR and its external power supply is enough to limit the current out from the Raspberry Pi.

GPIO pins can be controlled with Python scripts. There are already a few libraries developed for this purpose. We opted for the RPi.GPIO which provides a class to control the GPIO on a Raspberry Pi. A bespoken Python script using this library controls these pins through PsychoPy. An example of this script can be founded on the GitHub page Each stimuli video (6 for FMA, 5 for EE) has a unique identifier represented as a hexadecimal number (0x01, . . . , 0x0B). Two additional identifiers were added to mark the beginning (0x0C) and the end (0x0D) of the experiment.

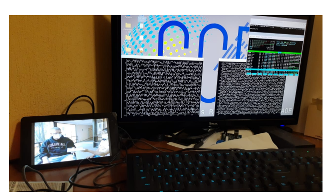
C. *Lab Streaming Layer for data acquisition.*

We chose Lab Streaming Layer (LSL) for data acquisition. Python scripts also available at GitHub were developed for the data acquisition from our fNIRS and EEG set up and visualization communicating with PsychoPy. The data coming from the devices through a local network is saved on a text file containing the timestamps, the imaging data itself and the synchronization triggers. An exemplary test running LSL with the PsychoPy experiment is shown in Fig. 3a.

III. Experiments and Results

A. *Performance during FMA assessment load*

A test was performed to measure the use of resources during acquisition. The following performance variables were observed: CPU usage from each core, amount of RAM memory used and CPU temperature throughout the entire experiment. Synthetic EEG (random) data and real experimental data (NIRScout data stream) was collected during a 20 minute period. Performance variables were sampled at 0.2 Hz. This sample frequency is compatible with the buffer



(a)

(b)

Fig. 3: (a) Raspberry running the PsychoPy based experiment (NeeGo screen) along with data acquisition of data streams simulating EEG and fNIRS signals (2nd screen only used for demonstration purposes). (b) Resting state test.

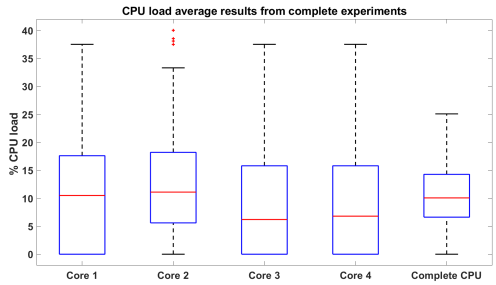


Fig. 4: CPU usage test results from each Raspberry core and its average value on psychopy projection experiment

time from the Raspberry Pi. The test results are depicted on Figures 4, 5a and 5b.

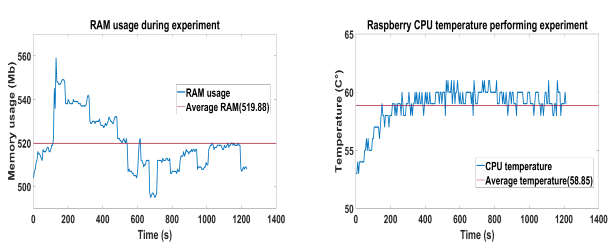
In terms of memory, Fig. 5a shows that the RAM usage never rises over 560 Mb and has an average usage of 520 Mb representing 13.6% of the 4 Gb of RAM available.

The Raspberry Pi 4 is designed to work up to 85^{o}C as a critical point. Operating close or above this point will indicate the need for an additional cooling system to the platform. As shown in Fig. 5b, the CPU temperature never raised over 61^{o}C with an average temperature of 58.85^{o}C well below the critical point. However, once additional analysis operations are added this value is expected to increase.

B. Performance during resting state load

Three resting state experimental tests were also carried out. Testing on resting state allow to discern what is strictly attributable to the FMA assessment stimuli separately. During these sessions, the subject remained in a resting stage with eyes closed for a 5 minutes period, without any stimuli projected (Fig. 3b). The Raspberry pi only task was capturing the two data streams; the fNIRS at 10Hz and the ECG at 512Hz.

Figure 6 shows the average results of CPU load for each core during the resting state. All single cores stayed below 20% load and the average usage was below 6%. This represents about half of CPU load compared to the FMA test where the stimuli is presented. RAM memory usage (Fig. 7a) and CPU operation temperature (Fig. 7b) also decreased accordingly. That is, the stimuli presentation



(a) (b)

Fig. 5: Performance during test on synthetic data. (a) RAM usage (blue) and its average value (red). (b) CPU temperature (blue) and its average value (red). (Best seen in color.)

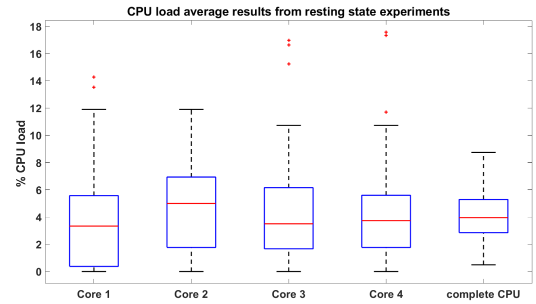
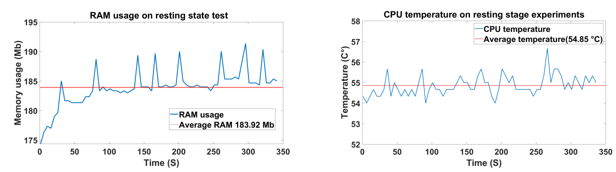


Fig. 6: CPU usage test results from each Raspberry core and its average value during resting stage test.

and synchronization operations take almost half of the operational demands.



(a) (b)

Fig. 7: Performance during resting state tests. (a) RAM usage (blue) and its average value (red). (b) CPU temperature (blue) and its average value (red). (Best seen in color.)

C. System load drop

An additional test was performed with the Raspberry only projecting the Psychopy stimuli without LSL communication corresponding to the system load drop from the FMA test (Stimuli projection and acquisition) to the resting stage test. The result of this test was compared to the FMA test. CPU load, RAM usage and temperature were measured in the same way as in the resting state test. A 50% decrease on CPU load, a 63% decrease on RAM usage, and the temperature dropped of one degree with respect to the FMA experiment.

D. Package loss assessment

In any communication systems there is always risk of package loss. A test was carried out to establish whether the Raspberry Pi can handle the communication with minimal or no package loss.

For the fNIRS, two data sets (SET001 and SET002) were simultaneously acquired by the Raspberry Pi through the communication system, and two additional computers directly hosting the neuroimaging devices following the full experimental design (i.e. projection of stimuli, synchronization and 2 data stream acquisition all at once). We compared the data recorded by the Raspberry Pi to those data acquired directly by the neuroimaging host computers to evaluate possible data lost on the streaming. These data sets were acquired with a 6.25 sampling rate with 44 signals (22 channels with a pair of signals -two wavelengths- for each channel). SET001 does not include the trigger marks while SET002 does. Exhaustive sample by sample comparison

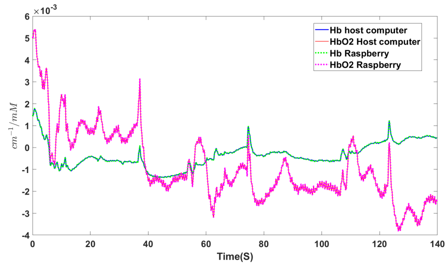


Fig. 8: Hemoglobin response from one fNIRS channel from SET001. The perfect overlapping between both records, host and raspberry Pi, is obvious demonstrating no loss of data.

from every sample for each channel was performed to detect any data loss. The Raspberry Pi successfully recorded all data without a single lost due to streaming in any of the data sets. For exemplary purposes, Fig 8 shows a comparison of the first channel from SET001 acquired through the host computer to the ones acquired by the Raspberry via LSL. Both datasets from the Raspberry as well as the host computer can be found in the OSF repository.

For EEG, 32 random signals simulated EEG-like records. These samples were sent through LSL to the Raspberry Pi. The sampling rate of the stream was raised on power of 2 steps from 128 Hz until 4096 Hz. Ten data files for each sample rate were sent and received after this process to account for variability in performance due to potential varying operating conditions. The values sent on every file were of type double of 64 bits. The data files acquired by the Raspberry were then compared to the original streamed ones on a sample by sample to quantify potential loss of data as above. Only two losses of 3 and 6 samples were detected on two files due to connection errors, a condition that can be detected on the fly.

E. Current consumption

For deployment to rural environments, current consumption is critical. A current consumption test was performed following the same methodology that in subsection III-A, as shown in Fig. 9 and in subsection III-B. Fig. 10 shows occasional current spikes. These spikes occurred during EE stimuli and were due to audio reproduction. Despite the spikes, current consumption never raised over 950 mA. This current demand can be satisfied with a 1000 mAh 5V battery for a FMA assessment measurement (20 minutes). As expected, Fig. 10 depicts a consumption drop of almost half percent with respect to the complete experiment, showing that most of Raspberry power is used for stimuli projection and not data acquisition

IV. Conclusions

The proposed multimodal EEG-fNIRS platform based on Raspberry Pi can successfully render the experimental stimulus using PsychoPy whilst concurrently communicating with the fNIRS and EEG devices using the LSL network

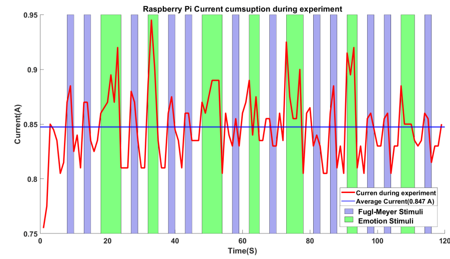


Fig. 9: Raspberry current consumption (red line). Vertical coloured bands indicate the different stimuli.

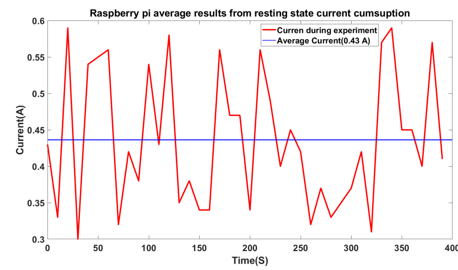


Fig. 10: Raspberry current consumption (red line). Average current value (blue line).

protocol to acquire the data. The Raspberry core demonstrated sufficient capacity to handle the operational demands that would occur during a Fugl-Meyer neurorehabilitation assessment.

While we develop a direct communication with the fNIRS device, for these tests, the NIRStar software was run in a Windows XP virtual machine inside the Raspberry Pi. The next steps are substituting the neuroimaging devices for their low cost counterparts and adding the AI module for signal analysis.

References

[1] B. Norrving and B. Kissela, "The global burden of stroke and need for a continuum of care," *Neurology*, vol. 80, no. 3 Supp. 2, pp. S5–S12, 2013.

[2] M. Kiani, J. Andreu-Perez, H. Hagras, E. I. Papageorgiou, M. Prasad, and C.-T. Lin, "Effective brain connectivity for fnirs with fuzzy cognitive maps in neuroergonomics," *IEEE Trans. Cogn. Devel. Syst.*, 2019.

[3] J. Andreu-Perez, D. Leff *et al.*, "Disparity in frontal lobe connectivity on a complex bimanual motor task aids in classification of operator skill level," *Brain Connect.*, vol. 6, no. 5, pp. 375–388, 2016.

[4] A. M. Chiarelli, F. Zappasodi, F. Di Pompeo, and A. Merla, "Simultaneous functional near-infrared spectroscopy and electroencephalography for monitoring of human brain activity and oxygenation: a review," *Neurophotonics*, vol. 4, no. 4, p. 041411, 2017.

[5] A. Berger, F. Horst *et al.*, "Current state and future prospects of eeg and fnirs in robot-assisted gait rehabilitation: a brief review," *Front. Hum. Neurosci.*, vol. 13, p. 172, 2019.

[6] S. Lloyd-Fox, M. Papademetriou *et al.*, "Functional near infrared spectroscopy (fnirs) to assess cognitive function in infants in rural africa." *Scientific reports*, vol. 4, no. 1, pp. 1–8, 2014.

[7] J. Peirce, J. R. Gray, S. Simpson *et al.*, "Psychopy2: Experiments in behavior made easy," *Behav. Res. Methods*, vol. 51, no. 1, pp. 195–203, 2019.

[8] A. R. Fugl-Meyer *et al.*, "The post-stroke hemiplegic patient. 1. a method for evaluation of physical performance." *Scand. J. Rehabil. Med.*, vol. 7, no. 1, pp. 13–31, 1975.

[9] M. Soleymani, J. Lichtenauer, T. Pun, and M. Pantic, "A multimodal database for affect recognition and implicit tagging," *IEEE Trans. Affect. Comput.*, vol. 3, no. 1, pp. 42–55, 2011.