Hardware and Software for Integrating Brain–Computer Interface with Internet of Things

Francisco Laport(✉), Francisco J. Vazquez-Araujo, Paula M. Castro, and Adriana Dapena

Department of Computer Engineering, Universidade da Coruña, Campus de Elviña s/n, 15071 A Coruña, Spain
francisco.laport@udc.es
http://www.gtec.udc.es

Abstract. This work shows a system that appropriately integrates a Brain–Computer Interface and an Internet of Things environment based on eye state identification. The Electroencephalography prototype for brain electrical signal acquisition has been designed by the authors. This prototype uses only one electrode and its size is very small, which facilitates its use for all type of applications. We also design a classifier based on the simple calculation of a threshold ratio between alpha and beta rhythm powers. As shown from some experiment results, this threshold-based classifier shows high accuracies for medium response times, and according to that state identification any smart home environment with those response requirements could correctly act, for example ON–OFF switching room lights.

Keywords: Brain–Computer Interface · EEG devices · Internet of Things

1 Introduction

A Brain–Computer Interface (BCI) is defined as a hardware and software communication system that records brain electrical activity, commonly obtained by means of Electroencephalography (EEG), and translates it into control commands for external devices [12]. These systems are especially interesting for people with severe motor disabilities since they allow them to interact with their environment without physical activity requirements.

Recent development of low-cost EEG devices together with emerging Internet of Things (IoT) have promoted the creation of new daily-used BCI applications

This work has been funded by the Xunta de Galicia (ED431C 2016-045, ED341D R2016/012), the Agencia Estatal de Investigacin of Spain (TEC2015-69648-REDC, TEC2016-75067-C4-1-R) and ERDF funds of the EU (AEI/FEDER, UE), and the predoctoral grant (ED481A-2018/156).

J. M. Ferrández Vicente et al. (Eds.): IWINAC 2019, LNCS 11486, pp. 22–31, 2019.
https://doi.org/10.1007/978-3-030-19591-5_3

in several domains [11]. As an extension of our proposal presented in [8], we will consider the utilization of BCI to determine the eye state and its integration with IoT.

Eye state identification or the eye-gaze analysis have become emerging topics of study in recent years due to its implication in human machine interfaces [10,14]. In particular, EEG eye state detection has been successfully applied in a wide variety of domains [19], such as infant sleep-waking state classification [6], driving drowsiness detection [20], stress features identification [18] and home automation control [7], among others.

Different approaches have been applied in the literature to classify and distinguish both eye states: closed eyes (cE) and open eyes (oE). Rösler and Suendermann [16] tested 42 different machine learning algorithms to predict the eye state from an EEG dataset of 117 s and 14 channels. The best performance was achieved by the K-star classifier with an error rate of 2.7 %.

Another study of Saghafi et al. based on that dataset employed Multivariate Empirical Mode Decomposition (MEMD) for feature extraction and Logistic Regression (LR), Artificial Neural Networks (NN), and Support Vector Machine (SVM) classifiers for detection of eye state changes [17]. Their proposed algorithm detected the eye state change with an accuracy of 88.2 % in less than 2 s. In this sense, Wang et al. [19] extracted the channel standard deviation and mean as features for an Incremental Attribute Learning (IAL) algorithm and achieved an error rate of 27.45 % from that dataset. In a recent study, Piatek et al. [13] tested 23 machine learning algorithms using four different datasets obtained from a 19-channels EEG device to classify three eye states: cE, oE and blinking. They showed that it is possible to predict eye states using EEG recordings with an accuracy range from about 96 % to 99 % in a real-time scenario.

Although some related work already achieve efficient and accurate detection of eye states, most of them collect brain activity using at least 14 electrodes and big–size EGG devices. Therefore, the main limitation of those devices is the user comfort and their difficulty to be used for long time periods or daily-life activities.

In contrast to these approaches, in this work we develop a BCI software tool integrated in an IoT system for non–critical real situations which only employs a single–channel EEG device to capture user's brain activity. This system monitors alpha (8–3 Hz) and beta (14–19 Hz) rhythms and extracts the mean power ratio between those bands as novel feature to determine user eye states. The extracted knowledge is then communicated to the rest of IoT devices as control commands using Message Queue Telemetry Transport (MQTT) [4].

This paper is organized as follows. Section 2 is devoted to show the system design and its architecture. Section 3 shows the main results achieved with the proposed system and some concluding remarks are made in Sect. 4.

2 System Design and Architecture

For the integration of both BCI application and IoT environment we propose the architecture shown in Fig. 1. The aim of this system is to capture the user's brain activity during its daily-life home activities and detect his/her eye states to control different environment devices. The main details about this architecture are described in this section.

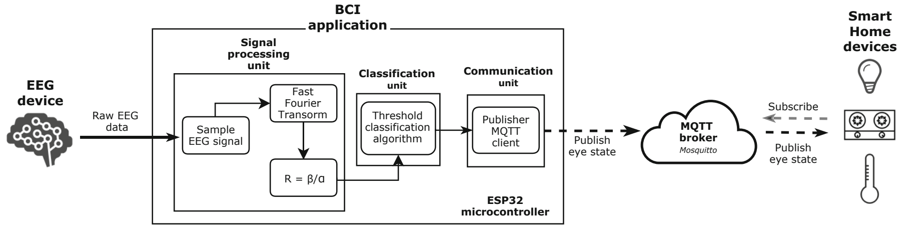


Fig. 1. Proposed system architecture.

2.1 EEG Device

The developed wireless EEG prototype is shown in Fig. 2. It employs three electrodes to capture EEG signals: input, reference and ground electrodes. The prototype uses the AD8221 instrumentation amplifier followed by a 50 Hz notch filter, a second order low pass filter with a cutoff frequency of 29.20 Hz, a second order high pass filter with a cutoff frequency of 4.74 Hz and a final bandpass filter with a frequency range from 4.7 Hz to 22 Hz with adjustable gain. The resulting EEG signal is sampled by the ESP32 microcontroller module [3] at a rate of 128 Hz.

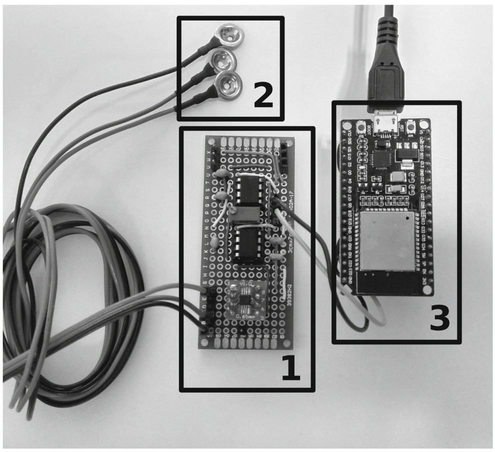


Fig. 2. Proposed EGG device prototype. (1) Amplifier; (2) Electrodes; (3) ESP32 microcontroller.

2.2 Signal Processing and Classification

The ESP32 microcontroller captures the brain signal received from the EEG device and carries out its processing and classification. Due to its dual core nature, complex processing tasks, such as Fast Fourier Transform (FFT), can be performed while the signal is sampled and the extracted knowledge is sent to the IoT environment. For the FFT implementation we will use an Arduino library [1].

The proposed eye state classifier makes use of the mean power value of the alpha (α) and beta (β) brain rhythms. Several studies have proved that the α power increases during closed eyes state while significant reductions are produced when subjects open their eyes. On the other hand, beta power does not show relevant differences between both eye states [5,9].

According to these studies, the proposed classifier obtains both powers considering a fixed time window and their ratio, defined as $R = \beta/\alpha$, is then calculated. This will be the extracted feature to be fed back to the threshold–based system responsible for deciding the user's eyes state. Thus, low ratios are associated to cE state due to the higher alpha, while higher ratios are connected to oE states due to lower alpha powers. Consequently, those ratio values smaller than a predetermined threshold will be classified as closed eyes. By contrast, the values higher than that threshold will represent the open eyes state. The classifier criteria is then defined by the following decision rule,

$$\begin{array}{ll} \text{cE}, & R \leq T_{\mathbf{h}}, \\ \text{oE}, & R > T_{\mathbf{h}}, \end{array} \tag{1}$$

where $T_{\mathbf{h}}$ and R are the threshold and ratio values, respectively.

The threshold value is calibrated from different EEG recordings and eye states. Thus, $T_{\mathbf{h}}$ is defined as follows

$$T_{\mathbf{h}} = \frac{\max(R_{\text{cE}}) + \min(R_{\text{oE}})}{2}, \tag{2}$$

where R_{cE} and R_{oE} respectively represent the ratio value for closed and open eyes.

Once the user's eye state is classified, that state is communicated to the IoT environment using the MQTT protocol.

2.3 IoT Environment

The IoT ecosystem is composed firstly by the EEG device and its BCI application and secondly, by the rest of household devices which consult the received information to determine its behavior.

The communication between different IoT agents is based on the MQTT protocol. It is a publish/subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth networks. The publish/subscribe model is built around a central broker and a number of clients connected to the broker. The broker acts like an intermediary agent, responsible for relating that information provided by the publishers with the subscribers clients [15].

These publishers send messages to the broker about an specific topic and the subscribers register their interest in some of them with the broker. The broker acts as a matchmaker, dealing with authentication and controlling who is allowed to publish or subscribe to which topics. These topics can be easily combined and created, so the system could be expanded by the inclusion of new devices or applications into the new topics.

The BCI application, running on the ESP32, is the first publisher client of the IoT ecosystem. It detects the user's eye state and, making use of the Wi-Fi module incorporated in the microcontroller, publishes the extracted information to the broker.

The MQTT broker deals with the messages received from the BCI application and forwards it to interested subscribers. The sent data correspond to 1–byte data, which represents the user's eye state. The broker is deployed in a Raspberry Pi 2 model B and implemented using Eclipse Mosquitto [2], an open source and lightweight MQTT broker.

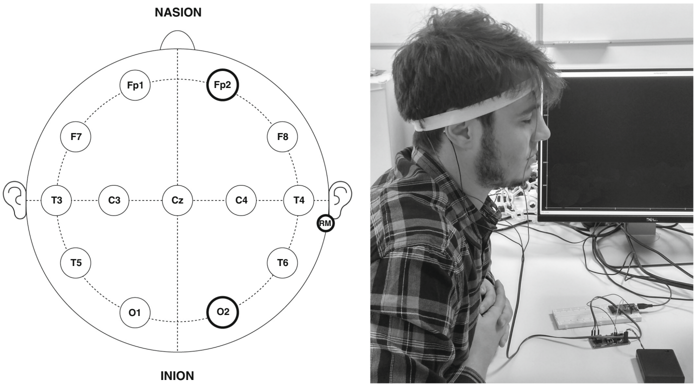
A wide variety of household devices could be incorporated to the system as subscriber clients (e.g light bulbs, kitchen burners, heating system, and so on). These devices receive information from the broker and react accordingly to it i.e., if the kitchen burner client receives that the user had the eyes closed for a long time, which likely means that he/she has fallen asleep, then the subscriber client should turn off burners in order to avoid risks.

3 Experimental Results

The experiments conducted in this study will aim to prove the accuracy in classification of the proposed system and its possible implementation in a real-life scenario. For this purpose, two different experiments have been developed: firstly, an off-line experiment, which tests classifier performances and secondly, an on-line experiment, which demonstrates the integration of both BCI and IoT environments. The details of these experiments are described in this section.

3.1 Off-line Experiments

The proposed classifier uses 42 EEG recordings captured from four healthy male volunteers, i.e. a total of 168 recordings is considered. Each one is composed by 20 s of each eye state. Therefore, we have 84 of them corresponding to cE and also 84 to oE. The subjects were asked not to move or speak during the experiment. Brain signals were captured at 128 Hz and, according to the 10–20 International System, the input electrode was located at the FP2 position, while reference and ground electrodes were placed in O2 and right mastoid positions, respectively. Figure 3 shows this electrode position (left) and a picture of a subject during the closed eyes recording using the proposed EEG device.



(a) (b)

Fig. 3. Electrode position (left) and a subject's photo in a closed eyes task (right).

The classifier is trained by selecting 8 random recordings, 4 for each eye state. Then, considering these training recordings, the threshold value is obtained by applying Eq. (2). According to this threshold value, the test recordings will be classified applying the criterion defined in Eq. (1), i.e. instances with a ratio value smaller than this threshold are classified as closed eyes, while those with higher values are classified as open eyes.

Depending on system applications, $T_{\mathbf{h}}$ and R parameters could be calculated considering different sizes for the time windows. Figure 4 shows the classifier accuracy as a function of time window size. It can be observed that as this size increases, the obtained accuracy improves, and vice versa. Thus, the accuracy is smaller than 70 % for all the subjects with windows of 1 s and greater than 90 % for a size of 13 s. Therefore, there is an important trade-off between system response time and classifier accuracy.

As can be observed from the figure, the proposed algorithm is appropriate for non–critical applications where short response times are not required. Consequently, optimal window sizes will be those with higher classifier accuracy for medium response times. For that reason, the window sizes range from 10 s to 19 s are selected. Figure 5 shows the corresponding threshold values obtained for these window sizes.

On the other hand, it is also important to highlight that those thresholds are highly user-dependent and, as a consequence, the classifier accuracy also depends on the brain characteristics of each subject.

3.2 On-line Experiments

The integration of both BCI and IoT environments will be tested now using a more realistic scenario. For this purpose, the BCI application will perform an on-line detection of user's eye state and according to this information the IoT ecosystem will control different elements of its surroundings.

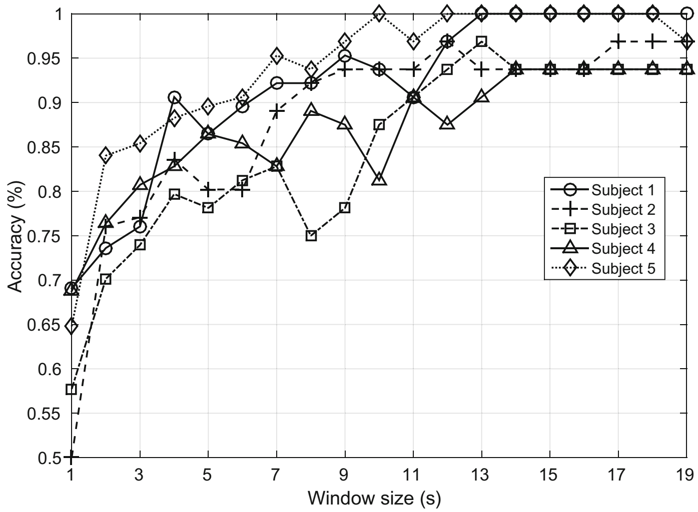


Fig. 4. Accuracy of the proposed classifier for different time window sizes.

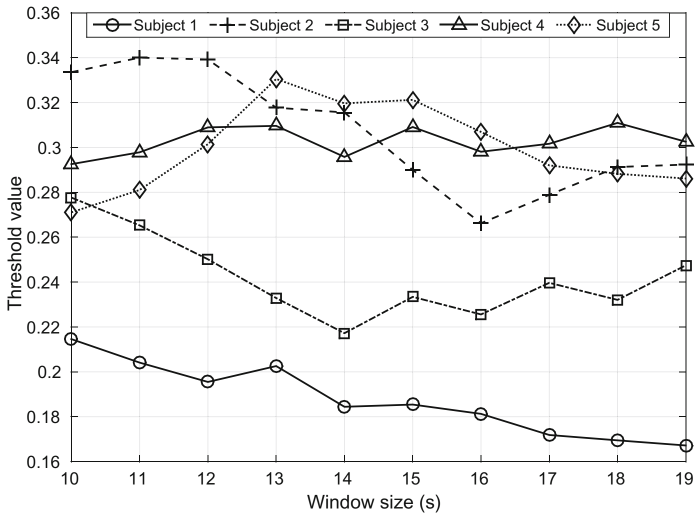


Fig. 5. Threshold values of each subject for different time window sizes.

Figure 6 shows the user's flowchart for a recording starting from an open eyes state. Forty EEG recordings were captured from subject 1 with the electrode position used for the off-line experiment described in the previous subsection. Each recording is composed of 277 **s** constituted by a short training period, for threshold calibration, and a longer test period, for system performance evaluation. As shown from off-line experiments, the window size should be of 10 s as minimum, and according to that, we choose a size of 13 s. Consequently, the

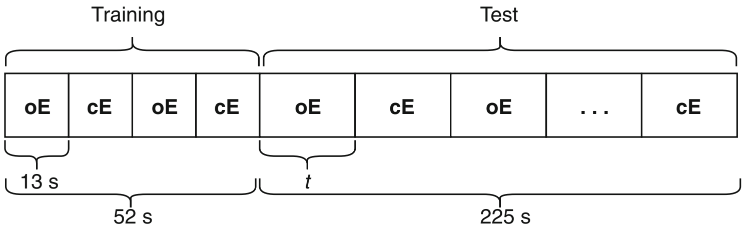


Fig. 6. User's experiment flowchart, with $t \in \{25\,\text{s}, 45\,\text{s}\}$ (cE and oE for closed and open eyes state, respectively).

training period is constituted by four windows of that size, two for cE state and two for oE state. On the other hand, for the test period, EEG recordings were captured using different time intervals, denoted as t, for the eye tasks, specifically 25 **s** and 45 s, having captured 20 recordings for each time interval. Moreover, in order to avoid any data correlation, half of the test recordings started with the oE task and the other half with cE.

EEG data captured from training period is processed and then used to calculate the threshold value according to Eq. (2). Applying this $T_{\mathbf{h}}$ value and following the criterion defined in Eq. (1), 13 s–windows are classified during the test period. Note that there are three types of windows: oE, cE and overlapped. It is important to say that, since eye state changes occur every 45 or 25 s, some windows could contain information from both eye states. In the transition windows, the window state is considered as that with a greater number of seconds during that time slot. As a consequence, the response time of the detection system will vary according to the window type to be classified i.e.,

- *Non–overlapped windows*, which only contain information related to a single state (cE or oE). In such a case, the response time is equal to the window size i.e., 13 **s**.
- *Overlapped windows*, which contain information related to both eye states. Since the window state corresponds to the dominant state, two possibilities can be considered. In the first one, *cE state is dominant*, and therefore the response time may be less than the window size. For example, this occurs when a window starts with oE and the state changes only 2 **s** later. Thus, the window state will be cE, since it contains 2 **s** for oE against 11 **s** for cE, and the system detects only 11 **s** later than the eye state switching. In the second one, *oE state is dominant* and the response time is equal to the window size, although it produces a delay in the following detection. For example, think about a window with 7 **s** of oE and 6 **s** of cE. After 13 **s** the window state is detected and classified as oE, but the following detection of cE will suffer from that delay of 6 **s**.

According to this criterion, the detection delay of the system will range between 7 **s** and 19 **s**. Table 1 shows the accuracy and the mean delay obtained by our classifier considering two time intervals for the eye task duration. It should be noted that the accuracy achieved for non–overlapped windows (i.e. only oE or cE) is above 93 % for all cases, while for overlapped windows (i.e. those with information of both eye states) it drops until 69.55 %. This reveals that our system better performs detecting eye states than changes between those states. On the other hand, it is important to highlight that although the detection delay can vary from 7 **s** to 19 **s**, its mean remains close to the window size.

Table 1. Accuracy and mean delay obtained by our classifier considering two time intervals for the eye task duration.

t	oE accuracy	cE accuracy	Overlapped accuracy	Mean delay
25 **s**	100 %	96.93 %	69.55 %	11.93 s
45 **s**	93.47 %	94.17 %	87.50 %	13.12 s

Remember that a 13 s–window is processed and on-line classified using the ESP32 microcontroller while the EEG signal is being sampled. Once the eye state has been determined, the system employs the MQTT publisher client to communicate that decision to the IoT ecosystem. The broker receives this information and forwards it to interested subscriber clients. In this experiment, an Arduino UNO connected to a light system has been implemented as a Smart Home (SH) subscriber client. This SH device monitors the user's eye state during long time periods and according to that information that light is regulated. All these MQTT messages were received by the subscriber with a latency lower than 40 ms.

4 Conclusions

In this work we demonstrate the appropriate integration of both Brain–Computer Interface and Internet of Things when Electroencephalography signals are acquired, the accurate identification of closed and open eyes states using a threshold–based classifier and how that extracted information can be correctly transmitted to a simple smart home environment consisting on on–off light switching. The experiments show high classifier accuracies and a correct working of the whole system. Experiment results have shown that classification accuracies, mean delays for detection or system working are sound enough for non-critical and monitoring applications. As future work, we have in mind to incorporate more electrodes to our prototype which will allow us to detect more complex mental states.

References

1. ArduinoFFT library. https://github.com/kosme/arduinoFFT
2. Eclipse Mosquitto. https://mosquitto.org/

3. ESP32 Series. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf/
4. MQTT. http://mqtt.org/
5. Barry, R.J., Clarke, A.R., Johnstone, S.J., Magee, C.A., Rushby, J.A.: EEG differences between eyes-closed and eyes-open resting conditions. Clin. Neurophysiol. **118**(12), 2765–2773 (2007)
6. Estévez, P., et al.: Polysomnographic pattern recognition for automated classification of sleep-waking states in infants. Med. Biol. Eng. Comput. **40**(1), 105–113 (2002)
7. Kirkup, L., Searle, A., Craig, A., McIsaac, P., Moses, P.: EEG-based system for rapid on-off switching without prior learning. Med. Biol. Eng. Comput. **35**(5), 504–509 (1997)
8. Laport, F., Vazquez-Araujo, F.J., Castro, P.M., Dapena, A.: Brain-computer interfaces for Internet of Things. Multi. Digit. Publishing Inst. Proc. **2**(18), 1179 (2018)
9. Li, L., Xiao, L., Chen, L.: Differences of EEG between eyes-open and eyes-closed states based on autoregressive method. J. Electron. Sci. Technol. **7**(2), 175–179 (2009)
10. Ma, J., Zhang, Y., Cichocki, A., Matsuno, F.: A novel EOG/EEG hybrid human-machine interface adopting eye movements and ERPs: application to robot control. IEEE Trans. Biomed. Eng. **62**(3), 876–889 (2015)
11. Narayana, S., Prasad, V., Warmerdam, K.: Mind your thoughts: BCI using single EEG electrode. IET Cyber-Phys. Syst.: Theor. Appl. (2018). https://doi.org/10.1049/iet-cps.2018.5059
12. Nicolas-Alonso, L.F., Gomez-Gil, J.: Brain computer interfaces, a review. Sensors **12**(2), 1211–1279 (2012)
13. Piatek, Ł., Fiedler, P., Haueisen, J., et al.: Eye state classification from electroencephalography recordings using machine learning algorithms. Digital Med. **4**(2), 84 (2018)
14. Reddy, T.K., Behera, L.: Online eye state recognition from EEG data using deep architectures. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 000712–000717. IEEE (2016)
15. Robinson, J.M., Frey, J.G., Stanford-Clark, A.J., Reynolds, A.D., Bedi, B.V.: Sensor networks and grid middleware for laboratory monitoring. In: 2005 First International Conference on E-Science and Grid Computing, pp. 8-pp. IEEE (2005)
16. Rösler, O., Suendermann, D.: A first step towards eye state prediction using EEG. In: Proceedings of the AIHLS (2013)
17. Saghafi, A., Tsokos, C.P., Goudarzi, M., Farhidzadeh, H.: Random eye state change detection in real-time using EEG signals. Expert Syst. Appl. **72**, 42–48 (2017)
18. Sulaiman, N., Taib, M.N., Lias, S., Murat, Z.H., Aris, S.A., Hamid, N.H.A.: Novel methods for stress features identification using EEG signals. Int. J. Simul.: Syst. Sci. Technol. **12**(1), 27–33 (2011)
19. Wang, T., Guan, S.U., Man, K.L., Ting, T.: Time series classification for EEG eye state identification based on incremental attribute learning. In: 2014 International Symposium on Computer, Consumer and Control (IS3C), pp. 158–161. IEEE (2014)
20. Yeo, M.V., Li, X., Shen, K., Wilder-Smith, E.P.: Can SVM be used for automatic EEG detection of drowsiness during car driving? Saf. Sci. **47**(1), 115–124 (2009)