Journal of Human Centered Technology
https://www.humentech.utm.my
E-ISSN: 2821-3467

Home Automation for Disabled Using Brain Computer Interface and Raspberry Pi

Mosa Muntadher Mohammed Mosa[1], Arief Ruhullah A. Harris[1,*]

[1]Department of Biomedical Engineering & Health Sciences, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

*Corresponding Author ruhullah@utm.my

Cite: https://doi.org/10.11113/humentech.v3n2.77

Research Article

Abstract:
Electroencephalography (EEG)-based smart home control systems are a key application of Brain-Computer Interfaces (BCIs). BCIs empower people with disabilities to achieve greater independence at home. These interfaces allow individuals with severe impairments to interact with their surroundings and communicate with others. Many people with special needs, particularly the elderly, face significant challenges in their daily lives that can severely impact their quality of life. This project aims to develop a non-invasive BCI for people with special needs to control household appliances and access an emergency system. A graphical user interface (GUI) will provide users with the ability to manage various smart home devices. This system will also benefit people with physical limitations by granting them greater control over their home's electrical and electronic appliances. This study successfully developed and implemented a BCI system for controlling home appliances. By leveraging the Steady-State Visually Evoked Potentials (SSVEPs) generated in response to flickering visual stimuli, the BCI system accurately interpreted user intentions through signal analysis and classification techniques.
Keywords: Brain-computer interface; Electroencephalography; Electronic gadgets; Non-invasive; Smart home.

1. INTRODUCTION

Patients who have physical impairments that limit their movement require a lot of help to improve their capacity to carry out everyday tasks. Those who are elderly or handicapped require assistance with everyday tasks. Several recent studies focus on improving Ambient Assisted Living (AAL). Nowadays, home automation may be achieved in a number of methods and the popular ones being1 voice control and touch- based reactions (1). Recently, a non-invasive commercial brain wave scanner has been used for home automation. An Arduino microcontroller uses data from a brain wave scanner to determine when to activate the gadget. Connectivity is established using Bluetooth modules (2). The brain computer interface (BCI) is a channel for brain-computer interface in device control, although systems can range from simple remote lighting control to complicated computer- or microcontroller- based networks with automation (3).

Most young, healthy adults are fully capable of performing all their Activities of Daily Living (ADLs) without any assistance. Failure to do routine activities might put one in danger and reduce the quality of life. To better identify patients who require assistance, healthcare providers should be aware of the value of ADL measurement (4). It is widely acknowledged that as a result of improved living conditions and healthcare, society as a whole is living longer. While the ability to survive into old age is something to be thankful for, the elderly confront a variety of difficulties that we should all be more aware of. We frequently ignore the concerns of our ageing population until we begin to age or see a loved one struggle. We can, however, do more as a society to improve their quality of life. One of the problems that the elderly may face, especially paralyzed ones, is difficulties in moving to perform their daily life activities. Additionally, they might have other problems like aphasia (talking loss) or seeing problems that can be associated with reducing their quality of life (5). Direct brain-to-computer interaction is made possible via BCIs. It is a communication technology that makes it easier to control external devices by using brain signals that are monitored.

Many people instinctively identify the word BCI with manipulating items with the power of the mind. Neuroscience in non-invasive EEG measuring is still in its early stages. Nonetheless, each step puts us closer to our objective and motivates a new generation of scientists and engineers to work in this sector. We have machine learning, which has just been around for a few years, and more than enough computational capacity to search for connections in EEG signals. The dataset's availability will be the sole shortcoming (6). The central nervous system's principal parts include the brain, spinal cord, and peripheral ganglia. The brain has more than 100 billion neurons, which make up the central nervous system. In truth, the notion of "Home Automation" has been around for a few years. Therefore, smart Home Control, which encompasses the notion of centralizing control of the state of the bulbs, home appliances, and so on, is becoming increasingly popular. As a result, the combination of BCI with smart home control is being advocated for the benefit of individuals with disabilities.

Received 31 March 2024; Accepted 13 May 2024
Available online 09 August 2024

Using a BCI could revolutionize home automation, allowing us to command the entirety of our dwelling with the power of our thoughts (7).

One prominent issue is the lack of portability and accessibility in current EEG-based home automation systems. Many available devices are large, complex, and expensive, making them impractical for widespread use by disabled individuals. This hampers the development of a portable and affordable system that can seamlessly integrate into users' daily lives. Furthermore, the high cost of high-tech assistive devices suitable for elderly or cognitively impaired individuals poses a significant barrier to accessibility. The development expenses associated with catering to a small market further limit their affordability. Consequently, there is a need for a more cost-effective and user-friendly solution that empowers disabled individuals to live more independently. Another challenge lies in the usability of the Graphical User Interface (GUI) employed in these systems. Existing GUI designs exhibit limitations, particularly in terms of user interface reliability. The placement of icons and buttons may cause difficulties for individuals with cognitive impairments or visual problems. It is essential to design a GUI that ensures ease of use, intuitive interaction, and efficient control over the home automation system. Moreover, current home automation systems for disabled individuals often have limited capabilities, restricting users' control over their home appliances. This limitation hinders the goal of enhancing independence and convenience. To overcome this, there is a need to develop a comprehensive and inclusive system prototype that enables users to control various appliances without physical movement.

A communication system called a brain-computer interface (BCI) solely interprets the user's instructions from their brainwaves and responds accordingly. Any BCI must depend on direct brain function measurements in the most widely accepted sense, provide user input, work online, and depend on deliberate control (that is, users must choose to execute a mental task to deliver a message or instruction any time they choose to use the BCI (8).BCI may be useful, particularly in safety-related applications or in situations where it is very difficult to move, and response speed is essential. Additionally, they may be utilized to improve the HCI systems' accuracy, resulting in a contribution from BCI in a variety of sectors, including business, education, advertising, entertainment, and smart transportation. Brain-computer interfaces must overcome technological barriers as well as obstacles posed by human acceptability to cope with such recently found technology, despite its predicted success (7).

Controlling a smart house has been the subject of several studies (9). The suggested system locates the user's position and orientation, which allows it to determine which devices the user is aiming at. The BCI module captures user attention, which is then analyzed to determine the action. The proposed approach's proof of concept was put into practice. The results of the studies show that the suggested system could precisely identify the target device and, as a result, control it (9). To locate the user in a 2-D plane using the image, a Kinect sensor was employed. The subject's EEG signal was recorded using the Neurosky MindWave BCI headset. With the aid of Neurosky MindWave SDK, the degree of meditation and attention values were estimated from, and values.

In order to make controlling any household appliances in the course of everyday life simpler, another study intends to give a salient manner of interaction between BCI and processors. A sensor placed above the head records the brain’s impulses. The sensor module itself carries out the fundamental pre-processing and data refining before transmitting it to the intermediary interface, an Android smartphone. The smartphone application receives the signal and attempts to decode it in accordance with the predetermined rules. If the brain reading agrees with the desired reaction, it also creates the control signal. This project has a lot of features, such as a special coding method for managing more devices and offering protection against unwanted control (10). The wireless brain-computer system is made up of three pieces, with the sensor module placed over the user's head. The CPU region of our Android smartphone is placed in the final segment, which is the controller section. Each will connect via a Bluetooth module, which necessitates certain installation and range-defining characteristics for the system. Java is used as the main programming language to create the Android application used in this study in the Android Studio software development environment. It has a file size of 16 Mb and works with any Android smartphone running an OS version greater than 4.0.

Lastly, this approach combined with a smart home system may be a successful treatment option for people with speech and motor impairments. In order to lessen speech and movement impairments, this research describes a home automation system based on Auditory Steady State Response (ASSR). A smart lamp and a fan connected to a smart socket were the two voice-controlled smart home appliances employed in the system implementation. Three trials for each of these two application states were conducted on four test participants. The system's response time to finish each job and the precision with which the tasks were completed were both being assessed (11). In this study, a non-invasive BCI and a Raspberry Pi are used to control a smart home remotely and without using physical movement. By implementing this project, patients with physical impairment can be more independent in their life by increasing their life quality. By using this device, the user can control several home appliances like light, fan, TV, associated with an emergency system that makes it unique compared to others.

2. MATERIALS AND METHODS

2.1 Proposed Method

The SSVEP-BCI is one of the most effective paradigms for putting into practice BCIs since it causes Steady State Visually Evoked Potentials (SSVEPs) in the brain by presenting a flickering visual stimulus at a certain frequency. Electroencephalography signal analysis may be used to measure and analyze the frequency specific SSVEP response that happens in response to such a stimulus. By associating various alternatives with various frequencies and asking the user to direct their attention openly or covertly towards the selected option, these reactions may be used for a user interface. The method is about stimuli the brain by the visual part of it using GUI that has images/colors that flicker with certain frequency which will be created in the brain. SSVEP-BCI has been successfully utilized for a variety of applications, including the control of a music player (12) and other user interfaces, human-robot interaction (13), text entry (14), game control (15), and many others. Due to its benefits of a greater information transmission rate (ITR), a better signal-to-noise ratio (SNR), and less training, SSVEP-BCI is more appropriate to drive this system.

2.2 System Design Methodology

By pre-processing and classifying brain wave signals from a variety of neurons and translating them into activity using computer chips and programming, an individual who is unable to move independently can use BCI system to command a smart home, motorized wheelchair, prosthetic limb, or essentially any controllable electronic system or device. Initially, BCI was connected to Raspberry Pi to get the brain signal as shown in Figure 1. The signals were received by installing the libraries that were needed into the Raspberry Pi. Then, the electrodes were connected to the amplifiers on BCI. After the acquisition of brain signal and the performance of signal processing, several tests were carried out to read the signal according to specific thinking, to be translated later as an output. For example, the participant was required to think that the bulb was OFF, and the signals were recorded several times. A similar thing was done when the bulb was turned ON. Whenever the participant thought about a specific status of the light, an instruction was sent to turn the light ON or OFF. Figure 2 shows the general idea of the output system control.

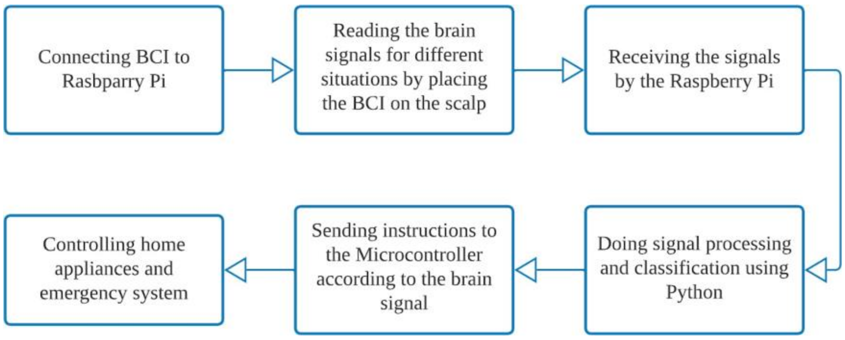


Figure 1. Project flow diagram.

Figure 2. Project overall system design and overall smart home control.

After the signals were read, serial data were delivered to the Raspberry Pi, allowing users to turn ON and OFF various home appliances through the Pi, such as lights, fans, and doors. A GUI was displayed graphically through the appearance of all menu options as shown in the following Figure 3. Any emergency lines were adopted in the GUI to be called by the users if necessary. From Figure 3(a), the user can control the system using only four outputs. The first two are the arrow up and down to switch between the home appliances that he/she wants to use. The second left and right signs will be used to turn that specific device on or off. Finally, by combining the previous processes and diagram, the overall of the complete system would look like Figure 3(b).

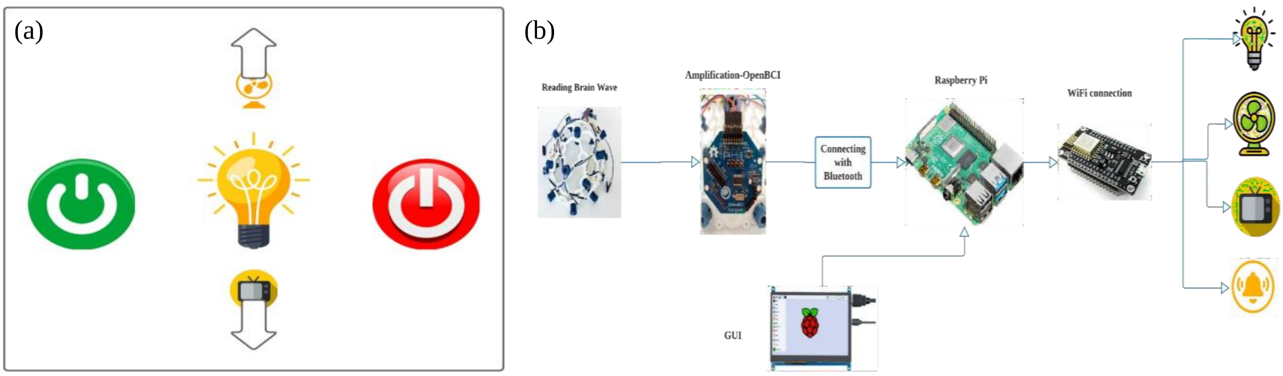


Figure 3. (a) GUI appearance for user and (b) complete system procedure.

https://doi.org/10.11113/humentech.v3n2.77

2.3 Hardware Connection

For the components that were used in this project, include the prototype components, are listed in Table 1 with the quantity of each component.

Table 1. List of components.

Component	Quantity
Open BCI Cyton Board	1
BLED112 Dongle	1
AA Battery Pack	1
AA Battery	4
EEG Dry Electrodes	5
Raspberry Pi	1
Monitor	2
NodeMCU8266	1
Relay 5V	3
Buzzer	1
Bulb	1
Fan	1

The hardware connection for this study involved the NodeMCU, a buzzer, and a relay that were connected to a fan, bulb, and monitor. The NodeMCU serves as the controller that received instructions from the Raspberry Pi. The output of the NodeMCU was connected to both the buzzer and the relay. The buzzer was responsible for generating audible feedback, while the relay controlled the power supply to the fan, bulb, and monitor. By toggling the relay ON or OFF, the system can turn these devices ON or OFF accordingly. This hardware setup enabled the BCI-powered smart home system to effectively control and interact with the connected household devices.

2.4 Data Visualization and Signal Processing

The study incorporated data visualization and signal processing techniques to enhance the analysis and interpretation of brain signals. The Butterworth filter was employed to perform signal filtering, specifically for removing unwanted frequencies and noise from the raw EEG data. This filtering technique helped to enhance the accuracy and reliability of the subsequent signal processing steps. The signal is then classified into four frequencies of interest using the Fast Fourier Transform (FFT). By extracting and analyzing these specific frequencies, the system can accurately identify the user's intended commands. The classified signals were visualized using the plt function, which allows for the plotting of the FFT results, providing a clear representation of the signal's frequency components. This data visualization and signal processing approach enables effective signal analysis and facilitates the reliable control of household devices based on the user's brain activity.

3. RESULTS AND DISCUSSION

The BLED112 Bluetooth dongle was used in the initial step of this research to establish a Bluetooth connection between the OpenBCI Cyton and Raspberry Pi microcontroller. After the Bluetooth connection was successful, the pyOpenBCI library was used in the second stage to collect the data, and the collected data was then classified in real-time in the final stage.

3.1 OpenBCI Board Connection to Raspberry Pi

The integration of the OpenBCI Cyton board with the Raspberry Pi for data acquisition is a critical aspect of the study. The provided code demonstrated the establishment of a connection and the continuous streaming of data from the Cyton board to the Raspberry Pi. To facilitate this integration, the study utilized the pyOpenBCI library, specifically the OpenBCICyton class, which enables communication with the Cyton board. The OpenBCICyton class offers essential functionalities for data retrieval and processing. The coding showed that the handle sample function acted as the callback function responsible for processing the received data. Within this function, the channel data from the sample was accessed and subsequently processed. In the given implementation, the channel data for the first four channels (indices 0 to 3) was printed to the console. Adjustments can be made to the channel indices based on the specific channel configuration of the OpenBCI Cyton board. This flexibility allows for adaptability to various electrode placements and configurations. Furthermore, the code incorporated the capture of a timestamp using the time.time() function, providing an accurate recording of the time at which each sample is received. The sample data, accompanied by the corresponding timestamp, were then appended to a CSV file named 'rowdata.csv'.

This facilitated the storage and enables further analysis of the acquired EEG data. It is worth noting that the structure and organization of the CSV file can be modified to suit specific requirements. Upon initializing an instance of the “OpenBCICyton” class, the code established a connection between the OpenBCI Cyton board and the Raspberry Pi. The appropriate port for the board was specified in the port parameter, considering the operating system-specific requirements. In the given example, 'COM6' was designated as the port for Windows, while '/dev/ttyUSB*' was used for macOS and Linux. The correct port number must be provided to ensure a successful connection. To enable the handling of incoming samples, the “set_sample_callback” method was employed to set the “handle_sample” function as the callback function. This mechanism ensures that each sample is processed promptly and efficiently.

To initiate the data streaming from the OpenBCI Cyton board to the Raspberry Pi, the “start_streaming” method was called on the board instance. This commenced the continuous transmission of data, allowing real-time monitoring and analysis. To ensure uninterrupted data storage, a while loop was implemented, which keeps the main thread active. Within this loop, a “time.sleep” function introduced a 1-second delay, facilitating any necessary data processing or analysis.Importantly, the code included a mechanism to gracefully terminate the program. This was achieved by capturing the “KeyboardInterrupt” exception (Ctrl+C). Upon raising this exception, the streaming was stopped using the “stop_streaming” method, and the loop was exited, leading to the proper termination of the program. This ensures the integrity of the acquired data and the smooth execution of the application.

3.2 Data Processing and Classification

For data processing, first, the code loaded the data from the CSV file named "dataframe.csv" into the 'data' list. The data were processed to ensure it contains only numeric values and is stored in a numpy array for further analysis. The timestamps and channel data were extracted from the loaded data. Next, the code defined the parameters for a Butterworth bandpass filter. The filter was then applied to each channel of the data using the signal. Filter function from the scipy library. This resulted in a filtered data numpy array containing the filtered channel data. To normalize the filtered data, each channel was divided by the maximum absolute value of that channel. This ensures that the channel data were scaled within the range of -1 to 1.

The code then defined a function named "classify" to calculate the real-valued Fast Fourier Transform (rFFT) for each channel. The rFFT data were stored in the "rfft_channels_data" numpy array. Within the "classify" function, the frequencies to compare against were specified. The mean of each channel was calculated using np.mean, and then the absolute values of the rFFT at the specified frequencies were compared to the channel mean.

Based on the comparison results, the code determined the action to be taken. If a certain frequency exceeded the channel mean for at least three consecutive samples and has the highest count among all frequencies, a specific action was triggered. The actions included turning a device off ("OFF"), turning a device on ("ON"), moving to the next component ("UP"), moving to the previous component ("DOWN"), or taking no action ("NO Order"). It also included print statements to display the corresponding action on the console. The codes included commented outlines for sending real-time results to a specified URL using the requests library, as well as time delays between actions.

3.3 Raspberry Pi and NodeMCU Connection

The Raspberry Pi and NodeMCU connection allows for communication between the two devices, enabling control and interaction based on received data through WiFi. The provided code established an HTTP server on the NodeMCU, allowing the Raspberry Pi to send data to the NodeMCU via HTTP requests. The NodeMCU received the data and processed it accordingly.in the setup function (Figure 4), the NodeMCU connected to the specified Wi- Fi network.

recieve

```
const char* ssid = "MUN";
const char* password = "123456789";
int state;
int y;
String x1;
ESP8266WebServer server(80);
String webpage = "";

void handleReceive() {
state = server.arg("number1").toInt();
y = server.arg("number2").toInt();
x1 = server.arg("number3").toInt();

  // Display the received number on the Serial Monitor
  Serial.print("state: ");
  Serial.println(state);
  // Display the received number on the Serial Monitor
  Serial.print("y: ");
  Serial.println(y);

  // Process the received number as needed
  if (state == 0  ||  y == 4){
    digitalWrite(1, HIGH);}
    else {
          digitalWrite(1, LOW);
```

Figure 4. NodeMCU Code to receive commands from Raspberry Pi.

Once connected, it started the HTTP server on port 80. The server was set to handle the "/receive" endpoint, and the received data were processed in the handleReceive function. The received values were stored in the "state," "y," and "x1" variables. Based on the received values, certain GPIO pins on the NodeMCU were controlled. For example, if the "state" value is 0 or the "y" value is 4, the GPIO pin 1 is set to HIGH; otherwise, it is set to LOW. The function then sent a response containing the "x1" value back to the Raspberry Pi. In the loop function, the server's client was handled, and the switch statement checked the value of "y". Depending on the value of "y" and "state", specific GPIO pins were controlled accordingly. For each case, different GPIO pins were set to HIGH or LOW, or a delay was applied. This connection between the Raspberry Pi and NodeMCU enabled the Raspberry Pi to send data to the NodeMCU, triggering specific actions on the NodeMCU based on the received values. The NodeMCU, in turn, controlled the GPIO pins to perform the desired operations.

3.4 Graphical User Interface

Figure 5 shows the images displayed on the canvas including icons representing different devices such as a light bulb, a TV, a fan, and a help icon. There are also red arrow icons pointing up and down, and red icons representing "ON" and "OFF" states. These images were positioned at specific coordinates on the canvas and flickering at specific frequencies (13, 14, 15, 16 Hz) to stimuli the brain. Some of the images were initially hidden using the canvas Item configure function, while others were initially visible. The visibility of these images was then toggled periodically using the toggle image function. Each image has a specific index, and the toggle image function toggles the visibility of the image at the given index after a certain delay according to the four frequencies. There was also a GUI switch function that takes an input parameter a to witch the images when the user focus on the arrow up/down.

Figure 5. Icons representing different devices such as light bulb, TV, fan, and help icon on GUI.

3.5 Prototype

After the system setup was done, one subject was chosen to be trained to use the device. The training process may take more than one month to get a good result. After the user focused on one image, the frequency of that image flickering was generated in the brain that was measured by the BCI and sent to Raspberry Pi. The Raspberry Pi will send commands to NodeMCU through WiFi after the signal processing and classification to perform the wanted task (To turn ON/OFF that specific device).

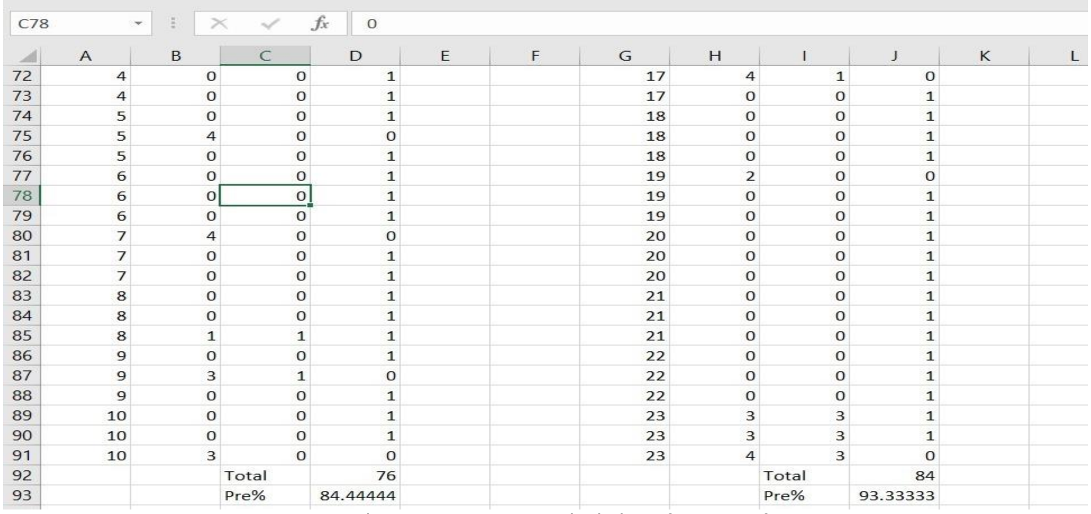
3.6 Data Interpretation

This study focused on the implementation of BCI system for controlling home appliances within a smart home environment. The system allowed users to control a fan, TV, bulb, and alarm using their brain signals, which were processed and translated into actionable commands. The BCI system was integrated with a GUI, a Raspberry Pi, and a NodeMCU controller to enable seamless communication and control between the user and the devices. The operation of the BCI system involved several key components and steps. The system relied on the SSVEPs paradigm to generate brain responses by presenting flickering visual stimuli associated with each device. These SSVEPs were then measured and analyzed using EEG signal analysis techniques.

The system's operation revolved around the accurate analysis of brain signals and the efficient transmission of data from the Raspberry Pi to the NodeMCU controller. The accuracy of a system (Figure 6) is an important aspect to evaluate its performance and effectiveness. In the context of the provided data, the system's accuracy can be assessed by comparing the predicted results with the actual results (16,17). The data consists of two sets of results, labeled as "Result" and "Actual," along with an additional column indicating whether the prediction was correct.

Analyzing the first set of data in Figure 7, a total of 76 instances was observed, the system accurately predicted 64 cases, resulting in an accuracy rate of approximately 84.44%. Similarly, in the second set of data, the system achieved an accuracy rate of around 93.33% by correctly predicting 78 out of 84 instances. These accuracy rates indicate that the system has performed reasonably well in predicting the outcomes based on the given data. However, it is important to note that additional information about the system, such as its purpose and the significance of the predictions, would provide a more comprehensive understanding of its effectiveness.

It is also worth mentioning that accuracy alone may not provide a complete picture of the system's performance. Other metrics, such as precision, recall, or F1 score, could be considered to assess the system's predictive capability more comprehensively. Furthermore, to evaluate the accuracy of the system more rigorously, it is essential to conduct statistical analysis, such as hypothesis testing or confidence interval estimation (18), to determine the significance and reliability of the observed accuracy rates. These analyses would provide a more solid foundation for drawing conclusions about the system's accuracy.



C78 | fx 0

	A	B	C	D	E	F	G	H	I	J	K	L
72	4	0	0	1			17	4	1	0		
73	4	0	0	1			17	0	0	1		
74	5	0	0	1			18	0	0	1		
75	5	4	0	0			18	0	0	1		
76	5	0	0	1			18	0	0	1		
77	6	0	0	1			19	2	0	0		
78	6	0	0	1			19	0	0	1		
79	6	0	0	1			19	0	0	1		
80	7	4	0	0			20	0	0	1		
81	7	0	0	1			20	0	0	1		
82	7	0	0	1			20	0	0	1		
83	8	0	0	1			21	0	0	1		
84	8	0	0	1			21	0	0	1		
85	8	1	1	1			21	0	0	1		
86	9	0	0	1			22	0	0	1		
87	9	3	1	0			22	0	0	1		
88	9	0	0	1			22	0	0	1		
89	10	0	0	1			23	3	3	1		
90	10	0	0	1			23	3	3	1		
91	10	3	0	0			23	4	3	0		
92			Total	76					Total	84		
93			Pre%	84.44444					Pre%	93.33333		

Figure 6. Accuracy Calculation of two sets for 30s.

3.7 Data Visualization and Signal Processing

The signal frequency FFT graph (Figure 7) generated from the previous code provides valuable insights into the spectral composition of the signal. By applying the FFT algorithm to the signal data, the graph depicts the amplitude of different frequency components present in the signal. The graph showcases peaks at specific frequencies, indicating the dominant frequencies within the signal. The height or magnitude of these peaks represents the strength or amplitude of the corresponding frequency components. Therefore, by analysing the peaks on the FFT graph, significant frequencies can be identified and their corresponding amplitudes, allowing the determination of spectral characteristics of the signal more effectively.

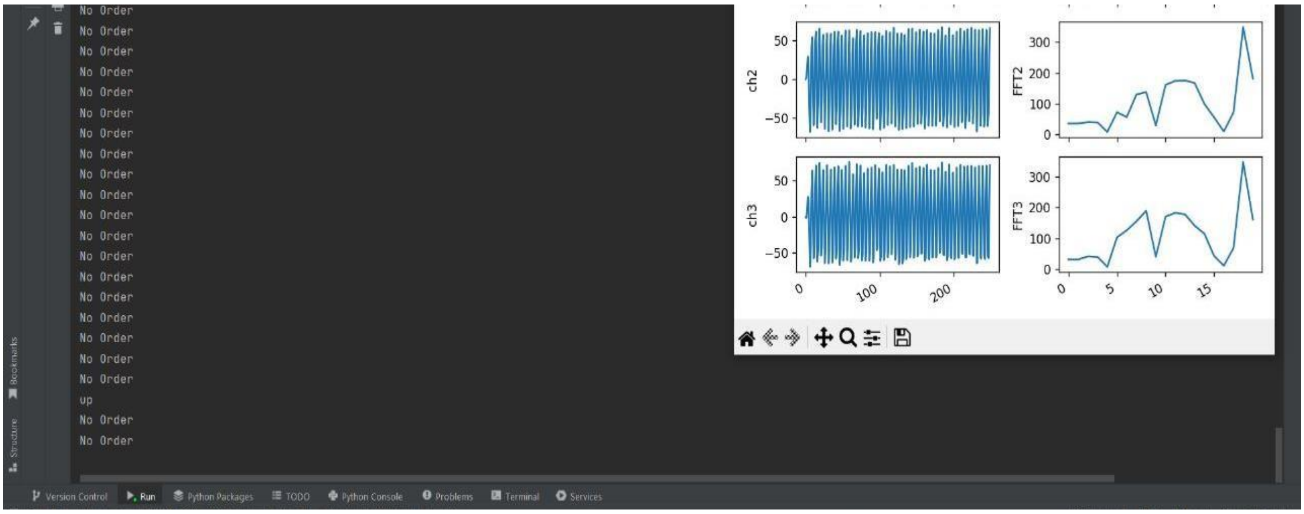


Figure 7. Signal FFT peak graph.

The GUI was created using Tkinter and displayed multiple images with the ability to toggle their visibility. The system accuracy was evaluated based on a set of data, where the predicted and actual results were compared. The analysis indicated a total of 76 correct predictions out of 84, resulting in an overall prediction accuracy of 84.44%.

4. CONCLUSION

In conclusion, this study successfully developed and implemented BCI system for controlling home appliances. By leveraging the SSVEPs generated in response to flickering visual stimuli, the BCI system accurately interpreted user intentions through signal analysis and classification techniques. The system utilized the Raspberry Pi for signal processing and the NodeMCU controller for seamless communication with home appliances. The integration of a user-friendly graphical interface allowed users to effortlessly navigate through devices and control their states. The study demonstrates the potential of BCI technology in enabling individuals with limited mobility to interact with their environment in a convenient and efficient manner.

ACKNOWLEDGMENT

The authors would like to thank Department of Biomedical Engineering & Health Sciences, Universiti Teknologi Malaysia for the provided facilities and area.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

(1) Lotte F, Faller J, Guger C, Renard Y, Pfurtscheller G, Lécuyer A, Leeb R. Combining BCI with virtual reality: Towards new applications and improved BCI. In: Allison BZ, Dunne S, Leeb R, Millán JDR, Nijholt A, editors. Towards practical brain-computer interfaces. Springer: Berlin, Heidelberg; 2012. p. 197–220. https://doi.org/10.1007/978-3-642-29746-5_10
(2) Bhemjibhaih DP, Sanjay GD, Sreejith V, Prakash B. Brain-computer interface-based home automation system for paralysed people. IEEE Recent Adv Intell Comput Syst. 2018; 230–233. https://doi.org/10.1109/RAICS.2018.8635060
(3) Qin LY, Nasir N M, Huq MS, Ibrahim B, Narudin SK, Alias NA, Ab Ghani M A. Smart home control for disabled using brain computer interface. Int J Integr Eng. 2020;12(4):74–82.
(4) Edemekong PF, Bomgaars DL, Sukumaran S, Levy SB. Activities of daily living. In StatPearls [Internet]. StatPearls Publishing, 2021 [cited 2024 March 31]. Available from: https://www.ncbi.nlm.nih.gov/books/NBK470404
(5) Bowling ANN, Grundy, E. Activities of daily living: changes in functional ability in three samples of elderly and very elderly people. Age and Ageing. 1997; 26(2):107–114. https://doi.org/10.1093/ageing/26.2.107
(6) Rakhmatulin I, Volkl S. Brain-computer-interface controlled robot via RaspberryPi and PiEEG. ArXiv Preprint 2022.
(7) Abdulkader SN, Atia A, Mostafa MSM. Brain computer interfacing: Applications and challenges. Egypt Inform J. 2015; 16(2):213–230. https://doi.org/https://doi.org/10.1016/j.eij.2015.06.002
(8) Brunner C, Birbaumer N, Blankertz B, Guger C, Kübler A, Mattia D, Millán J del R, Miralles F, Nijholt A, Opisso E. BNCI Horizon 2020: Towards a roadmap for the BCI community. Brain-Comput Interfaces. 2015; 2(1):1–10. https://doi.org/10.1080/2326263X.2015.1008956
(9) Bhemjibhaih DP, Sanjay GD, Sreejith V, Prakash, B. Brain-computer interface based home automation system for paralysed people. 2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS). 2018; 230–233. https://doi.org/10.1109/RAICS.2018.8635060
(10) Raj U, Mukul MK. Home automation using brain computer interface (BCI). In: Nath V, Mandal JK. Microelectronics, communication systems, machine learning and internet of things. Springer: Berlin, Heidelberg; 2020. p. 639–649. https://doi.org/10.1007/978-981-19-1906-0_52
(11) Shivappa VKK, Luu B, Solis M, George K. Home automation system using brain computer interface paradigm based on auditory selection attention. 2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC). 2018; 1–6. https://doi.org/10.1109/I2MTC.2018.8409863
(12) Zerafa R, Camilleri T, Falzon O, Camilleri KP. A real-time SSVEP-based brain-computer interface music player application. XIV Mediterranean Conference on Medical and Biological Engineering and Computing 2016. 2016; 173– 178. https://doi.org/10.1007/978-3-319-32703-7_36
(13) Güneysu A, Akin HL. An SSVEP based BCI to control a humanoid robot by using portable EEG device. 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). 2013; 6905–6908. https://doi.org/10.1109/EMBC.2013.6611145
(14) Chen X, Wang Y, Gao S, Jung TP, Gao X. Filter bank canonical correlation analysis for implementing a high-speed SSVEP-based brain- computer interface. J Neural Eng. 2015; 12(4):046008. https://doi.org/10.1088/1741-2560/12/4/046008
(15) Lalor EC, Kelly SP, Finucane C, Burke R, Smith R, Reilly RB, Mcdarby G. Steady-state VEP-based brain-computer interface control in an immersive 3D gaming environment. EURASIP J Adv Signal Process. 2005; 19:1–9. https://doi.org/10.1155/ASP.2005.3156
(16) Thompson DE, Quitadamo LR, Mainardi L, Gao S, Kindermans PJ, Simeral JD, Fazel-Rezai R, Matteucci M, Falk T H, Bianchi L. Performance measurement for brain–computer or brain–machine interfaces: A tutorial. J Neural Eng. 2014,11(3), 035001. https://doi.org/10.1088/1741-2560/11/3/035001
(17) Wang X, Hersche M, Tömekce B, Kaya B, Magno M, Benini L. An accurate eegnet-based motor-imagery brain–computer interface for low-power edge computing. 2020 IEEE International Symposium on Medical Measurements and Applications (MeMeA). 2020; 1–6. https://doi.org/10.1109/MeMeA49120.2020.9137134
(18) Zhang R, Xu Z, Zhang L, Cao L, Hu Y, Lu B, Shi L, Yao D, Zhao X. The effect of stimulus number on the recognition accuracy and information transfer rate of SSVEP–BCI in augmented reality. J Neural Eng. 2022; 19(3), 036010. https://doi.org/10.1088/1741-2552/ac6ae5