

Simulation of Wheelchair Control by Integration of Computers and Electronics Platform in BCI Controlled Systems

Manjot Kaur
Computer Science
BBSBEC
FGS, India
Manjot793@gmail.com

Birinder Singh
Computer Science
BBSBEC
FGS, India
Birinder.singh@bbsbec.ac.in

Abstract—Device control using brain signals is a technique which could dramatically change the life of disabled individuals. An attempt has been made to control such devices using brain signals. This work shows the hardware implementation of controlling a wheelchair using brain signals which is developed using electronics open source platforms to simulate real wheelchair control. This system uses Arduino UNO microcontroller board, L293D driver circuit, two DC motors, wheelchair hardware and wires for connections to produce simulation of wheelchair control.

Keywords- BCI, SSVEP, EEG, Serial communication, Signal Movement

I. INTRODUCTION

Brain computer interface is a technology which allows individuals to interact with their environment with the help of their brain signals without using any of the body parts. This technology is extremely helpful for those which are locked in useless bodies. This is possible with the help of BCI technology. In order to capture signals Electroencephalography (EEG) technique is used which is a non-invasive way of acquiring the signals by placing electrodes over the scalp. Steady state visual evoked potential technique (SSVEP) is used to originate brain signals with the help of visual stimuli which flickers at a particular frequency. Such type of systems leads to development of various assistive devices.

In this work has been done to move to move electronically developed wheelchair using Brain signals. This paper presents hardware implementation of wheelchair control in which wheelchair would not actually move instead a simulation of wheelchair movement has been developed using electronics platform and executed with the help of algorithm developed in Matlab.

Arduino UNO microcontroller board is used to control the signal movements in the wheelchair. It communicates with the computing platform to receive the signals from the computer and sends it to the hardware platform to perform actual motion.

L293D motor driver circuit is responsible for receiving the signals from the Arduino microcontroller and sending it to the DC motors. It serves as switch between microcontroller board and DC motors which actually transmits the signals but do not manipulate them.

DC (Direct Current) motors has been used to bring the wheels in motion and to power the wheelchair. DC motor is preferable for small scale applications due to their advantage of consuming less power and supports higher speed applications.

II. MATERIALS REQUIRED

A. Arduino UNO Microcontroller

Arduino is an open source electronics platform used to control various actuators such as motors, lights etc. It is a

board with microcontroller embedded on it. This microcontroller contains 28 pins in total. ATmega328 has digital and analog pins available on it. Digital pins works with digital values of pins. It needs power supply of 5volts for its operation. It receives serial values from the laptop with the help of USB supply. Its microcontroller is programmed on the basis of application used. Download libraries for particular application and upload into the microcontroller using Arduino IDE. Arduino board is shown in Fig. 2.1.



Figure 2.1. Arduino UNO microcontroller.

Arduino IDE is used to program the arduino microcontroller according to the requirement. Arduino IDE is windows based software written in java. Its programs are divided into three parts. First one includes declaration of Pins, second is about specifying the mode as 'input' or 'output', third one is about writing the values on it. Fig. 2.2 shows arduino windows based software to program the microcontroller.

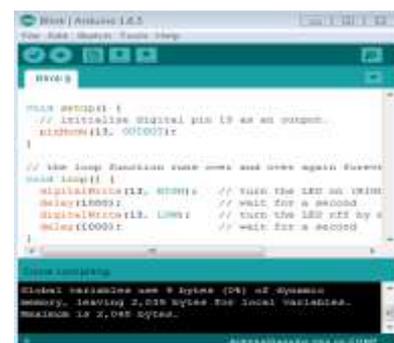


Figure 2.2. Arduino IDE.

B. L293D Motor Driver Circuit

It is a driver circuit which allows the rotation of both the motors simultaneously either clockwise or anticlockwise direction. This contains 16 pins but when soldered on printed circuit board then only 6 pins are used for its functionality. It takes the signals from the microcontroller, amplifies it to enhance its power and then send it to motors. It needs power supply of 5volts supplied by microcontroller. Since microcontroller supplies only 5 v but motor operation needs power supply of 12v hence motors cannot be directly controlled by the microcontroller. It provides power supply of 12v to the motors with the help of external source. Fig. 2.3 shows driver circuit.



Figure 2.3.L293D Motor Driver.

C. DC Motor

Two DC motors are attached to the rear wheels of the wheelchair simulator. These motors rotate with the speed of 100 revolutions per minute. For the movement of wheelchair in forward or backward direction both the motors will rotate but in opposite directions. For left or right movement only one motor remain in motion and another remains in stop position. Figure 2.4 shows DC motor used.



Figure 2.4. DC motor [20]

III. METHODS EMPLOYED

In order to run the wheelchair simulator software results has been integrated with hardware. Hardware interfacing is done to accomplish this. The generated command must be sent to the wheelchair. Four stimuli presented on the screen represent four commands for wheelchair in forward, backward, left and right direction.

In order to send the command from laptop to wheelchair simulator, a custom middleware layer is created to interface the wheelchair and the BCI software. The first step is to transmit the signal from the computer application to the Arduino board. The last step is to send the signal from Arduino Uno to the motor driver in order to operate the motor and correspondingly the wheelchair. Generated sequence of bits must be fed to Arduino to change the state of its output pins. Detailed description is provided in below. Fig. 3.1 shows block diagram of flow of signals in the system. Above mentioned process is accomplished in three steps.

1) *Interfacing of Matlab with Arduino Board:* This is done to send the commands from the Matlab application to Arduino board in order to set or reset its pins. To accomplish this goal firstly Arduino board is prepared to communicate with computer application. Arduino windows based software is used to program the microcontroller on the board. With the help of Arduino IDE the COM port used for serial communication and board type is selected. Then adioes.pde sketch, available in arduino libraries, is uploaded into microcontroller which enables it to receive data from Matlab. There are digital and analog pins available out of which digital pins are required for our work. The bits lead to high and low states of digital pins where high indicates supply of 5v whereas low indicates 0v supply.

For serial communication between software and hardware two programs need to be executed. The first one is in the Arduino and the second one is in Matlab. Matlab program contains arduino coding in which firstly pins are initiated as output pins using pinMode() function where output mode indicates arduino is going to receive output from Matlab. Pins have been set to high or low using digitalWrite()function. The

digitaldata is sent from Matlab to Arduino using serial communication via COM port. Matlab sends digital commands to Arduino. On the basis of received values Arduino pins has been set to high or low values. Digital pins 3, 5, 7 and 8 used along with their digital values are shown in table 3.1.

pinMode(): Configures the specified pin to behave as an input or output pin.

Syntax:pinMode(pin, mode) where pin represents any pin and mode='input' or 'output'.

digitalWrite(): write a high or low value to a digital pin.

Syntax:digitalWrite(pin, value) where pin= any pin and value='0' or '1'.

TABLE3.1 DIGITAL LOGIC OF ARDUINO PINS FOR VARIOUS MOVEMENTS

MOVEMENT TYPE	PINS USED	VALUES
Forward	3, 5, 7, 8	0, 1, 0, 1
Backward	3, 5, 7, 8	1, 0, 1, 0
Left	3, 5, 7, 8	0, 0, 0, 1
Right	3, 5, 7, 8	0, 1, 0, 0
Stop	3, 5, 7, 8	0, 0, 0, 0

2) *Interfacing of Arduino Board and Motor Driver:* Microcontrollers cannot drive the motors directly. In order tocontrol speed or direction of motor some type of drivers are needed. They work as interface between microcontroller and motors. Motor needs power supply higher than 5v which cannot be directly provided by microcontroller. Driver's converts low current signal to high current signal which is used to drive the motors. It can provide higher power to the

motor for its operation. It can control two motors simultaneously. After soldering on PCB 6 pins are sufficient to control its operation which are A1, A2, B1, B2, 5v and Gnd.

Arduino programs the pins of motor driver where pin 3 sets the value of A1, pin 5 sets the value of A2, pin 7 sets the value of B1, pin 8 sets the value of B2, 5v is connected to power supply pin in arduino which supplies power to the driver and Gnd is connected to ground pin of arduino. Pin 3, 5, 7 and 8 of Arduino board control pins A1, A2, B1 and B2 of the Motor driver as shown in table 3.2. These pins on motor driver control another 4 input pins on the L293D motor driver on the left and right sides. Values of these pins vary in synchronization with A1, A2, B1 and B2.

TABLE 3.2 RELATIONSHIP BETWEEN ARDUINO AND MOTOR DRIVER PINS.

Arduino Pin No.	Operation	Driver Pin Name
3	Controls	A1
5	Controls	A2
7	Controls	B1
8	Controls	B2
5v	Controls	5v
Gnd	Controls	Gnd

3) *Interfacing of Motor Driver and DC Motors:* Motor driver pins are attached to DC motors with the help of wires. Power supply is given to DC motors with the help of adapter which is connected to power pins available on motor driver. On the basis of values provided by Motor drivers motors rotates correspondingly.

For forward motion and backward motion both motors rotate simultaneously but in opposite directions. For turn to left or right one motor remains in stop position while other possesses motion in forward direction. For instance, for right turn motor on the right side is in rest whereas motor at left side moves in forward direction results in right movement occurs. Same scenario is applicable to left movement of wheelchair.

To communicate with wheelchair, the computer send the classified command as digital data through the serial port to an Arduino UNO which then transmits the signal to the motor driver which is further connected to the DC motors. These DC motors are connected to the rear wheels which brings the wheelchair in motion. Fig. 3.2 shows flow of signals among various hardware devices.

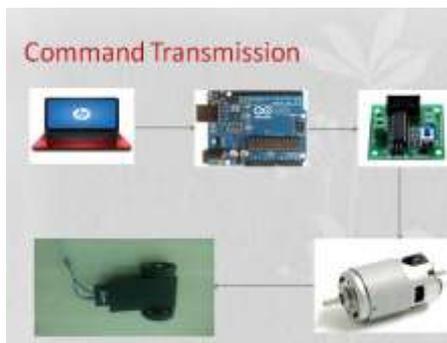


Figure 3.2 Flow of digital output

IV. RESULTS

With the implementation of above hardware architecture a wheelchair could be controlled successfully. This wheelchair uses two motors for its movement. As it receives a particular bit sequence on the basis of attended stimuli it moves in particular direction depending on the bit sequence. This model has been tested number of times by different individuals and no error has been encountered in the hardware part of the system. Table 4.1 shows the digital bits and movement of wheelchair corresponding to each frequency.

TABLE 4.1 OPERATIONS PERFORMED BY WHEELCHAIR

Serial No.	Frequency(Hz)	Bit Sequence	Operation Executed
1.	8	1,0,1,0	Backward Movement
2.	10	0,1,0,1	Forward Movement
3.	13	0,0,0,1	Left Movement
4.	15	0,1,0,0	Right Movement

D. CONCLUSION

This shows the demonstration of BCI technology for real world applications. Here this simulator moves in all the four directions depending on the user's intent. Hardware implementation is done successfully and has shown 100% accuracy. Bit pattern is successfully sent to arduino. Arduino efficiently controls drivers and drivers are able to communicate with motors. With more modifications this system could be converted to real system which can support the lives of amputees.

ACKNOWLEDGMENT

I would like to acknowledge all those people who helped me during this work. Especially I would like to thank my guide Prof. Birinder Singh for his valuable contribution in this work. I would also like to thanks Mindmedia for the development of excellent technology to capture brain signals.

REFERENCES

- [1] Bakardjiana, H., Tanakaa, T. and Cichockia, A., 2010. "Optimization of SSVEP brain responses with application to eight-command Brain-Computer Interface", Neuroscience Letters, ELSEVIER, Tokyo, Japan, pp. 34-38.
- [2] Bi, L., Fan, X., Jie, K., Teng, T., Ding, H., and Liu, Y., 2014. "Using a Head-up Display-Based Steady-State Visually Evoked Potential Brain-Computer Interface to Control a Simulated Vehicle", IEEE Transactions on Intelligent Transportation Systems, Vol. 15, No. 3, pp. 959-966.
- [3] Buksh, R., Routh, S., Mitra, P., Banik, S., Mallik, A., and Gupta, S.D., 2014. "Implementation of MATLAB based object detection technique on Arduino Board and iROBOT CREATE", International Journal of Scientific and Research Publications, Vol. 4, No. 1, pp. 1-5.
- [4] Chen, Z., Jao, T., Qin, M., and Zhao, X., 2010. "BMW: Brainwave Manipulated Wagon", IEEE Transactions on Biomedical Engineering, Vol. 59, No. 11, pp. 1-7.
- [5] Fan, X., Bi, L., Teng, T., Ding, H., and Liu, Y., 2015. "A Brain-Computer Interface-Based Vehicle Destination Selection System Using P300 and SSVEP Signals", IEEE

- Transactions on Intelligent Transportation Systems, Vol. 16, No. 1, pp. 274-283.
- [6] Graimann, B., Allison, B. and Pfurtscheller, G., 2010. "BRAIN-COMPUTER INTERFACES Revolutionizing Human-Computer Interaction", The Frontiers Collection, DOI 10.1007/978-3-642-02091-9, copyright by Springer-Verlag Berlin Heidelberg.
- [7] Jia, C., Gao, X, Hong, B., and Gao, S., Fellow, 2011. "Frequency and Phase Mixed Coding in SSVEP-Based Brain-Computer Interface", IEEE Transactions on Biomedical Engineering, Vol. 58, No. 1, pp. 200-206.
- [8] Johnson, G.D. and Krusienski, J., 2011. "Asynchronous Real-Time Control of a Robotic arm using a Brain-Computer Interface", IEEE Transactions on Biomedical Engineering, Vol. 57, No. 17, pp. 11-15.
- [9] Khare, V., Santhosh, J., Anand, S. and Bhatia, M., 2011. "Brain Computer Interface Based Real Time Control of Wheelchair Using Electroencephalogram", International Journal of Soft Computing and Engineering, Vol. 1, No. 5, pp. 41-45.
- [10] Layan, O.C., 2014. "A Portable and Embedded SSVEP BCI System", MS Thesis, Institute of Science and Engineering, Galatasaray University.
- [11] Li, Y., Pan, J., Wang, F., and Yu, Z., 2013. "A Hybrid BCI System Combining P300 and SSVEP and Its Application to Wheelchair Control", IEEE Transactions on Biomedical Engineering, Vol. 60, No. 11, pp. 3156-3166.
- [12] Lin, J.S. and Shieh, C.H., 2014. "An Ssvep-Based Bci System and its Applications", International Journal of Advanced Computer Science and Applications, Vol. 5, No. 10, pp. 54-60.
- [13] Rak, R.J., Kołodziej, M. and Majkowski, A., 2012. "Brain-Computer Interface as Measurement and Control System", Metrol. Meas. Syst., Warsaw, Poland, Vol. 19, No. 3, pp. 427-444.
- [14] Shruthi, B. N., Shivraj and Sumathi, S., 2014. "Handgesture Based Direction Control of Robocar using Arduino Microcontroller", International Journal of Recent Technology and Engineering, Vol. 3, No. 3, pp. 32-35.
- [15] Shyu, K.K., Lee, P.L., Lee, M.H., Lin, M.H., Lai, R.J. and Chiu, Y., 2010. "Development of a Low-Cost FPGA-Based SSVEP BCI Multimedia Control System", IEEE Transactions on Biomedical Circuits and Systems, Vol. 4, No. 2, pp. 125-132.
- [16] Tello, R.G., Ferreira, A. and Bastos, T., 2015. "Development of a Human Machine Interface for Control of Robotic Wheelchair and Smart Environment", IFAC Symposium on Robot Control, Salvador, Brazil, pp. 26-28.
- [17] Thorp, E.B., Abdollahi, F., Chen, D., Farshchiansadegh, A., and Ferdinando, A., 2011. "Using Upper-Body Motions to Control Power Wheelchairs for Individuals with Tetraplegia", IEEE Transactions on Biomedical Engineering, Vol. 66, No. 18, pp. 122-127.
- [18] Wu, Z., Lai, Y., Xia, Y., Wu, D., and Yao, D., 2008. "Stimulator selection in SSVEP based BCI", Medical Engineering & Physics, ELSEVIER, ChengDu, China, pp. 1079-1088.
- [19] Yin, E., Zhou, Z., Jiang, J., Yu, Y., and Hu, D., 2013. "A dynamically optimized SSVEP brain-computer interface (BCI) speller", IEEE Transactions on Biomedical Engineering, Vol. 67, No. 15, pp. 1-10.
- [20] *DC motor Permanent Magnet*. (n.d.). Retrieved April 29, 2016, from www.directindustry.com/prod/chiaphua-components/product-61070-575718.html

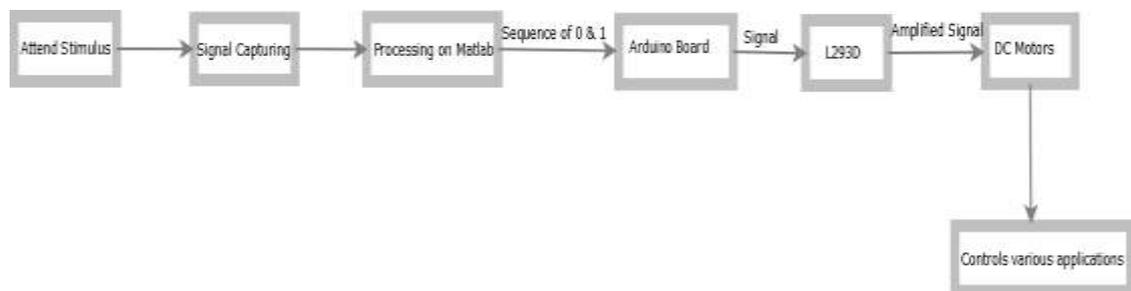


Figure 3.1 Block Diagram for flow of signals in the system.