# A Multi-Purpose Brain-Computer Interface Output Device

**David E. Thompson and Jane E. Huggins**

Key Words
Amyotrophic Lateral Sclerosis
Assistive Technology
Brain-Computer Interface
Electroencephalography
Individuals with Disabilities
User-Computer Interface

**ABSTRACT**

While brain-computer interfaces (BCIs) are a promising alternative access pathway for individuals with severe motor impairments, many BCI systems are designed as stand-alone communication and control systems, rather than as interfaces to existing systems built for these purposes. An individual communication and control system may be powerful or flexible, but no single system can compete with the variety of options available in the commercial assistive technology (AT) market. BCIs could instead be used as an interface to these existing AT devices and products, which are designed for improving access and agency of people with disabilities and are highly configurable to individual user needs. However, interfacing with each AT device and program requires significant time and effort on the part of researchers and clinicians. This work presents the Multi-Purpose BCI Output Device (MBOD), a tool to help researchers and clinicians provide BCI control of many forms of AT in a plug-and-play fashion, i.e., without the installation of drivers or software on the AT device, and a proof-of-concept of the practicality of such an approach. The MBOD was designed to meet the goals of target device compatibility, BCI input device compatibility, convenience, and intuitive command structure. The MBOD was successfully used to interface a BCI with multiple AT devices (including two wheelchair seating systems), as well as computers running Windows (XP and 7), Mac and Ubuntu Linux operating systems.

**INTRODUCTION**

Brain-computer interfaces (BCIs) are intended as a promising alternative to existing movement-controlled interfaces for individuals with severe motor impairments. BCIs have the potential to provide these individuals with direct-brain control of technologies such as computers, augmentative communication devices, environmental control systems and neural prostheses. Using this technology to maintain agency and their relationships with others, prospective users would potentially experience an increased level of independence and a higher quality of life.[1]

Mason and Birch[2] proposed a framework for BCI design in which the user controls a device (e.g., a power wheelchair) through a series of functional components: (1) signal acquisition and amplification; (2) feature extraction; (3) feature translation; (4) control interface; and (5) device controller. To date, the majority of BCI research has focused on the design and optimization of the first four components of the framework - for a thorough review of these elements of BCI design, the interested reader is referred to Mason's later paper.[3]

As BCI design matures from a theoretical laboratory technology to a practical system that can be used in real-world situations by individuals with disabilities, there is growing interest in seamlessly interfacing the first four framework elements with existing device controllers. This approach has been taken by one group with a commercial robot and a commercial environmental control device,[4] as opposed to the more common strategy of recreating existing AT device controllers, e.g., a programmable infrared controller.[5] In light of the goal of seamlessly interfacing with existing device controllers, BCIs can be modeled under the principles of input device emulation. In this model, the BCI control (e.g., the electrodes, amplifier, feature extractor and feature translator) and the Control Interface are combined into a single input device. This BCI input device, via an interface technology, would emulate standard input devices and be able to interface with various Device Controllers, as depicted in Figure 1. The importance of input device emulation as a design criterion has been emphasized in assistive technology design, maximizing the number of people who can connect special input devices,[6] and providing a strategy that can accommodate the users' changing needs.[7] This latter issue is of particular importance to individuals with degenerative conditions such as amyotrophic lateral sclerosis (ALS) who experience frequent changes in their physical ability to interact with their environment; the ability to connect various input devices into the same Device Controller would minimize cost and adaptation for the user and his/her caregivers.

In order to realize this model, an effective interface technology must be developed that can connect the BCI Input Device to the Device Controller. Such an interface technology should be compatible with a variety of Device Controllers, able to interface with any BCI output, be convenient to use, and provide intuitive control of the Device Controller to the BCI Input Device so that the resulting system can easily be configured to user needs and preferences. The design specifications for such an interface technology are presented in Table 1.

This paper presents the design and implementation of one example of an interface technology that connects BCI Input Devices to Device Controllers: the Multi-Purpose BCI Output Device (MBOD). The MBOD is capable of translating plain-text User Datagram Protocol (UDP) outputs such as those produced by BCI2000[a] (an open-source BCI implementation) into physical switch closures, Universal Serial Bus (USB) keyboard key-presses, or USB mouse movements and clicks. The design of the MBOD according to the specifications presented in Table 1 will be described in the Methods section, while the Results section will present the results of testing the MBOD with various assistive technologies, operating systems and mobility devices. The

From the Departments of Biomedical Engineering (D. Thompson, J. Huggins) and Physical Medicine and Rehabilitation (J. Huggins), University of Michigan, Ann Arbor, Michigan.

Address correspondence or requests for reprints to David Thompson, C640 Med Inn Bldg., 1500 E Medical Center Dr., Ann Arbor, MI 48109.

Email: dthomp@umich.edu

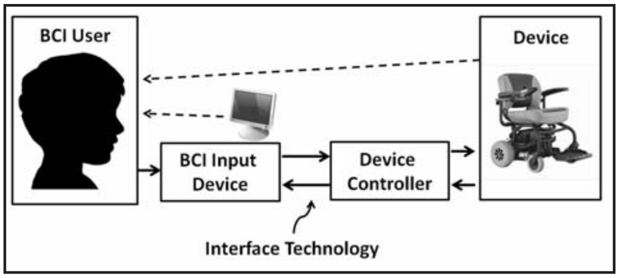


**Figure 1.**
The role of Interface Technology connecting a BCI Input Device to a Device Controller, within the framework of input device emulation.

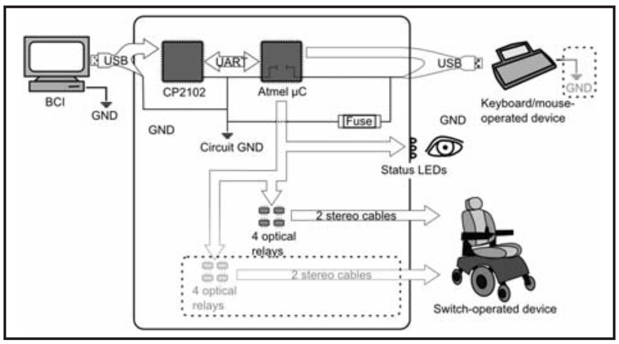


**Figure 2.**
Overview of the MBOD hardware.

| Table 1 | |
|---|---|
| Design specification for an Interface Technology to connect a BCI Input Device to a Device Controller | |
| **Specification** | **Explanation** |
| Device controller compatibility | Compatible with various assistive technologies, operating systems, etc. |
| BCI input device compatibility | Compatible with BCI2000 and custom BCI software |
| Convenience | Plug-and-play, no need for external power, configurable |
| Intuitive command structure | Complete and intuitive control of all output options |

**Table 2**
Supported inputs of common assistive technology devices

| Device | Assistive Technology Categories | Supported Input | | |
|---|---|---|---|---|
| | | Switch | Mouse | Keyboard |
| Vantage[1] | AAC | ✓ | ✓ | |
| DynaWrite[2] | AAC | ✓ | | ✓ |
| Imperium[3] | ECS | ✓ | | |
| Relax II[4] | ECS | ✓ | | |
| DASHER[5] | CA | | ✓ | |
| REACH Interface Author[6] | AAC, CA | ✓ | ✓ | ✓ |
| Cowriter[7] | CA | ✓ | | ✓ |
| PointSmart[8] | CA | | ✓ | |
| Assistive Mouse Adapter[9] | CA | | ✓ | |
| Smart Mustang Motorized Wheelchair[10] | M | ✓ | | |

[1]Prentke Romich Company, Wooster, OH, USA
[2]DynaVox Mayer-Johnson, Pittsburgh, PA, USA
[3]Tash, Inc, Richmond, VA, USA
[4]Tash, Inc, Richmond, VA, USA
[5]University of Cambridge, Cambridge, United Kingdom
[6]Applied Human Factors, Inc., Helotes, TX, USA
[7]Don Johnston Incorporated, Volo, IL, USA
[8]Infogrip, Inc, Ventura, CA, USA
[9]Montrose Secam Limited, Iver, Bucks, United Kingdom
[10]Smile Rehab Ltd, Newbury, Berkshire, United Kingdom

Discussion section will include the merits and limitations of the MBOD, and the paper will conclude with a note on the usefulness of the MBOD as a research tool to facilitate the study of clinical BCI use and the expectation of the eventual incorporation of this interface functionality into commercial BCIs.

## METHODS

### Design Overview

The MBOD is designed to provide various output options to a BCI, enabling the combination of the BCI with existing AT devices and software. In order to interface with AT devices or programs with device controllers that support a USB keyboard or mouse, the MBOD is capable of acting as an intervening layer of hardware between two USB hosts (e.g., computers). An intervening hardware layer is necessary because USB hosts are unable to directly emulate devices or be connected together. The MBOD also provides physical switch outputs for interfacing with AT device controllers that are normally switch-operated.

The MBOD consists of (1) hardware and firmware that receives USB input from a BCI Input Device and outputs the desired keyboard, mouse or switch state to a Device Controller and; (2) a “companion” program, running on the BCI Input Device that performs ASCII-to-USB translation and interfaces with the MBOD. An overview of the MBOD hardware is provided in Figure 2; briefly, a SiLabs CP2102 USB-to-UART bridge receives input from the BCI Input device, and an Atmel AT90USB1287 microcontroller receives the translated commands, generates the corresponding USB or switch output, and handles all associated protocols and timing concerns. Switch output is accomplished through the use of optorelays. The hardware includes three status light-emitting diodes (LEDs) that provide feedback of the device’s state to aid in troubleshooting. Descriptions of the design decisions that shaped the details of the implementation are provided in the following sections.

### Design Goal 1 - Device Controller compatibility

The first design goal for the MBOD was to ensure its compatibility with the maximum number of Device Controllers that prospective BCI users might employ for the purposes of augmentative and alternative communication (AAC), environmental control systems (ECS), computer access (CA) and movement (M). Table 2 lists a number of common assistive technology devices used by individuals with disabilities and the type (e.g., switch, mouse, keyboard) of input required by their device controllers.

As illustrated in Table 2, seven of the ten listed devices have controllers that can be operated from switch input and seven of the ten can be operated with either keyboard or mouse input. To maximize Device Controller compatibility, the MBOD was designed to be able to emulate all three devices (USB keyboard, USB mouse, and physical switch).

The MBOD was programmed to identify itself as a keyboard and/or mouse using the USB Human Interface Device (HID) standards. Using these standards allows the MBOD to be recognized as a valid input device without modifications to the device controller (i.e., without the installation of drivers or software). USB ports are ubiquitous on personal computers regardless of operating system, and are frequently present on assistive technology devices. USB output is additionally attractive as USB specifications have a high degree of backward compatibility (e.g., both low- and high-speed USB 1.0 devices are supported by USB 2.0 and 3.0 hosts). By employing the USB HID standards, the MBOD is automatically compatible with all USB hosts that accept external keyboard or mice, without the need for installing drivers or software on the target system. This is particularly important as many assistive technology devices do not allow the installation of custom software, including drivers.

To emulate switch output, the MBOD also includes optorelays controlled by the microcontroller. To date, both Toshiba's TLP222A and Clare's PAA132 have been used successfully to generate switch output to assistive devices such as wheelchair seating systems, including an Invacare TDX SP and Pride Quantum 600. As illustrated in Figure 2, the MBOD board layout can accommodate up to eight optorelays, though only units with four optorelays have been built and tested to date. The MBOD provides configurable timing for the duration of both keyboard and switch outputs, as some programs and devices cannot recognize short key-presses and precision timing of switch outputs is useful for interfacing with wheelchair seating systems. The combination of USB and optorelay output enables the MBOD to interface with every assistive device identified in Table 2.

**Design Goal 2 - BCI Input Device compatibility**

While there have been a number of attempts to develop unifying frameworks and platforms for BCI research, a significant proportion of the field consists of BCI systems that are designed idiosyncratically.[8] With the introduction of BCI2000, an open-source BCI research and development platform,[9] many research laboratories presented investigations and developments based on this platform. While the MBOD was designed to accommodate input from BCI2000, care was taken to ensure compatibility with custom BCIs as well. This was done by creating a companion program to handle USB communication with the MBOD and thus abstract the hardware considerations away from the BCI. While the companion program could have been integrated into BCI2000, separating the functionality ensures that a custom BCI could easily work with the MBOD by implementing the simple BCI2000-style outputs described below.

BCI2000 and potentially other BCIs output user datagram protocol (UDP) packets that are human-readable strings in ASCII format containing (1) an identifier, (2) a whitespace, and (3) an output string. The companion program catches these UDP packets, translates them into a sequence of USB and switch commands, and forwards these commands to the MBOD hardware via a USB port on the BCI Input Device. Consider, for example, a BCI user who has just input the letter "A" using a BCI system running BCI2000 and one of its modules, the P300 speller. The output of this BCI Input Device is a UDP packet containing the string "P3Speller_Output A". The companion program would parse this UDP packet sent by the BCI, translate the content into the keystroke "A" in USB codes, and subsequently send this information onto the MBOD, which forwards it to a device controller, e.g., a communication system. The companion program is designed to accommodate key-presses (including special characters and strings of arbitrary length), mouse movements and switch commands in a similar manner. The system for handling special characters is described under design goal 3. For mouse and switch commands, the configurable identifier in the UDP packet is different, and the output string is numeric for mouse movements. If used with BCI2000's ConnectorModule, the companion program will listen for the Signal values for X and Y coordinate movement while ignoring all other state variable output, though this default behavior may be modified.

**Design Goal 3 - Convenience**

The MBOD was designed to enable BCIs to emulate a USB plug-and-play device, wherein the device facilitates its own discovery and installation into the USB host system, without the need for device configuration or user intervention. In other words, the MBOD was designed to maximize convenience for the user and to offer immediate functionality once it was connected between the BCI Input Device and the Device Controller. To facilitate this ability, three features were considered: drivers, power, and flexible output capabilities. Ideally, the MBOD would be able to interface with Device Controllers without requiring driver installation; this is addressed in detail under design goal 1 via compliance with the USB HID standard. To maximize convenience, the MBOD should not necessitate the use of an external power source or batteries. Consequently, the MBOD is designed to operate from power taken from the USB port of the BCI Input Device. The MBOD's maximum current draw, with all three status LEDs and eight optorelays simultaneously active, is less than 100 mA; this is easily supplied by any standard-compliant USB port. This specification allows the MBOD to work from a battery-operated laptop; if the BCI amplifier is battery-powered or powered from the laptop as well, the entire BCI system can be mounted to a mobile platform, e.g., a powered wheelchair. In addition, the low current draw of the MBOD should not unduly affect the battery life of laptop computers. Finally, the MBOD is designed to accommodate a wide range of output configurations, rendering hardware configuration unnecessary for the end-user. The MBOD firmware is capable of handling simultaneous keyboard, mouse, and switch commands, as a simple packet structure supports sending different types of data to the microcontroller.

**Design Goal 4 – Intuitive Command Structure**

The MBOD was designed such that all three output modalities mapped intuitively from the BCI Input Device to the connected Device Controller. In the case of switch output, this simply involved mapping a single switch command from the BCI Input Device to a single switch selection through the Device Controller, a function that was easily implemented in the MBOD companion program as described above. To interface with the widest variety of devices, the MBOD is capable of translating the BCI Input Device's switch commands into variable-length switch presses. Similarly, intuitive mapping of mouse movements from the BCI Input Device, which might be used with continuous-output BCI modalities (e.g., sensorimotor rhythms), to the Device Controller involved mapping changes in the X and Y coordinate commands to corresponding changes in mouse position on the output device. Once again, this functionality was implemented easily in the MBOD companion program.

By contrast, generating a full intuitive mapping of the keyboard required special consideration; while the UDP packet output of the BCI Input Device easily represented letters of the keyboard in ASCII symbols, keys without ASCII equivalents (e.g., F1 or arrow keys) were not directly

**Table 1**
Testing results for the MBOD with different output devices, using BCI control

| MBOD Functionality | Assistive Technologies | | Operating Systems | | | Power Wheelchairs | |
|---|---|---|---|---|---|---|---|
| | DynaWrite (AAC) | Scanning Director II ECS | Windows (XP, 7) | Mac OS | Ubuntu Linux | Invacare TDX | Pride Quantum |
| Keyboard output | Y | N/A | Y | Y* | Y | N/A | N/A |
| Mouse output | N/A | N/A | Y | Y | Y | N/A | N/A |
| Keyboard/mouse combo | N/A* | N/A | Y | Y* | Y | N/A | N/A |
| Switch output | N/A | Y | N/A | N/A | N/A | Y | Y |

'Y' indicates successful operation, 'N' indicates unsuccessful operation, and 'N/A' indicates that the input mode is not supported by the device. Asterisks indicate notes in the following subsections. Non-Windows operating systems were tested with simulated brain activity.

represented. These special keys were therefore encoded in human-readable and user-configurable format using a command language in the companion program to handle modified key-presses and combinations (e.g., the Windows "copy" command "CTRL-C"). In this command language, configurable start and end characters (default "[" and "]") enclose tokens indicating the special characters. Tokens are defined for every key on a full keyboard (including a full numeric keypad). The default tokens are verbose to avoid conflicts, but can be redefined with a single line in a .ini file on the computer running the BCI. For example, the default token "KEYBOARD_F1" can be redefined to just "F1"by adding the line "KEYBOARD_F1=F1". To minimize interference with phrases, if a substring does not parse to a command, it is passed onto the Device Controller unchanged. There is also an "ECHO" keyword that instructs the command parser to ignore substrings in a single packet that would normally be considered commands, leaving a character-by-character ASCII-to-USB translation in place.

Creation of a custom ASCII extension to include special keys was rejected because the above scheme provides a more intuitive and efficient setup (e.g., typing "[CTRL-C]" instead of looking up an arbitrary hex value). It is also useful to the user, as target text is often displayed as the result on the BCI screen and these tokens are more interpretable than, e.g., the meaningless square that is currently produced on the BCI2000 screen when a non-standard ASCII value is output. Finally, such a custom extension would likely conflict with existing extensions used in, e.g., some non-English languages.

## RESULTS

The functionality of the MBOD was tested by using it to interface an EEG-based BCI (BCI2000 running on Windows XP with a g.USBamp amplifier) with various assistive technologies, operating systems and power wheelchairs. The interface was considered successful if users were able to use the BCI to control the functionality of the output device. The results of the test are presented in Table 3. The non-Windows operating systems were tested with simulated brain activity.

### Assistive Technologies

The impact of plug-and-play use enabled by the MBOD on users' abilities to operate a BCI underwent rigorous testing with 24 individuals using it to interface BCI2000 with an assistive technology device (the DynaWrite AAC device) and a computer running Windows XP (unpublished data). Results from three of the individuals are available in a preliminary report.[10]

One limitation of an early version of the MBOD came to light during this testing. The DynaWrite AAC device did not recognize the keyboard if the MBOD was placed in keyboard/mouse-combo mode, though it retained full functionality when the MBOD was placed in keyboard-only mode. The MBOD's USB operating mode (i.e., keyboard, mouse, or keyboard/mouse combo) can now be changed at runtime, which has proved to be a useful feature for troubleshooting compatibility issues.

Testing of the MBOD switch function was performed using a mu-rhythm BCI to generate switch closures, as well as with a P300 interface in the power wheelchair tests described below. This mode was successfully tested with the Scanning Director II environmental control system, although timing of outputs of the mu-rhythm BCI for operation of the scanning interfaces was challenging (unpublished data).

### Operating Systems

The MBOD was tested on its ability to interface with three different Operating System families: Windows (XP and 7); Mac and Ubuntu Linux. Testing was considered successful if the MBOD functioned on the Operating System without the need for device configuration or user intervention. The MBOD functioned successfully on all three Operating Systems, though the Mac OS required a special key sequence upon the first installation.

### Power Wheelchairs

The MBOD can produce switch outputs, and many wheelchair systems are designed for switch operation. However, feed-forward control is unstable and unsafe unless the wheelchair includes obstacle avoidance (see, e.g., Rebsamen[11]). Using a 3-D accelerometer to measure the current tilt position, we developed a system to allow control of the tilt-in-space system of powered wheelchairs.[12] Using this system, the MBOD enabled BCI control of two different powered wheelchair tilt-in-space systems - a Pride Quantum 600 and an Invacare TDX SP. The Invacare chair was used in a laboratory experiment to measure the effect of rotational movement on the P300 response.[13] Since the interaction was in a controlled environment, the MBOD switch was plugged directly into the chair through the standard device controller. The Pride chair was used in a subject's home, so ensuring that the switch only controlled tilt was more important than in the laboratory environment. The MBOD was thus used to interface with a dedicated tilt control board produced by the seating system manufacturer.

## DISCUSSION

This paper presents the design of the MBOD - an intervening layer of hardware which enables a BCI Input Device to emulate a keyboard, mouse, or a switch output and therefore interface with many Device Controllers. The MBOD is compatible with a wide variety of assistive technology, can receive commands from BCI Input Devices, and provides intuitive mapping of keyboard, mouse or switch functions. Additionally, the commands are configurable, allowing researchers to use terms intuitive to their lab and thus reduce errors when interfacing with the BCI.

### Limitations and Future Work

The choice of USB communication did bring up an additional consideration: grounding. USB ground is typically connected directly to case ground on personal computers. Connecting two case grounds together forms a ground loop. Although unlikely to be a problem in a user's home, ground loops can be a problem in large office buildings or hospitals. The user is protected from this unlikely event by the electrical isolation in BCI signal acquisition amplifiers, but such a ground loop could possibly damage equipment. In the MBOD, this issue has been addressed by separating the grounds with a fuse. The fuse has non-negligible resistance, alleviating the ground loop issue. Additionally, the fuse will protect the equipment on both sides of the MBOD from high current flow.

However, this solution is not ideal. In the event of failure, the fuse must be replaced before the MBOD will function again. Additionally, though the resistance is non-negligible, it is still small. Since the design and construction of existing MBOD units, a better solution has become commercially available: a USB opto-isolator chip. This chip, if placed between the Atmel microcontroller and the Device Controller, would completely prevent the ground loop, operate even if the grounds were at different potential, and not need to be replaced as is the case with a fuse. While it is not possible to add the chip to our existing MBOD units, future designs will incorporate this feature. Alternatively, wireless communication from the MBOD to a USB dongle could be implemented. The USB dongle to convert the wireless signal to USB would be necessary since current AT devices do not directly accept wireless protocols such as Bluetooth.

While the MBOD allows interfacing a BCI to any AT that accepts a USB keyboard, USB mouse, or physical switch, it does introduce a second computer screen when used with a visual BCI and computer-based AT devices. Careful design of a commercial BCI with plug-and-play capabilities could reduce this concern, however. Minimizing BCI display size is an obvious step, since a small display requires less eye movement, provides more positioning options, and provides similar accuracy to a large display.[14] In the future, plug-and-play BCIs could use a transparent screen or retinal projection so the display appears at the same location as the target device.[15] Alternatively, a plug-and-play BCI could use auditory presentation of options[16,17] or a visual BCI could operate an AT device with auditory feedback. For now, if the AT to be controlled is a piece of software that can be installed on the BCI computer, the separate display can be avoided by connecting the plug-and-play BCI back to the BCI host computer. While this configuration has been successfully tested with the MBOD, the special case of controlling AT software on the BCI computer can also be addressed in software rather than hardware.

A final limitation of the MBOD is that it only provides information flow from the BCI to the AT; only extremely limited information about USB configuration is returned to the BCI. In many cases, particularly when the BCI is used for communication through an AT device, this limitation is not important; the AT device is designed to present feedback to the user. In situations where the BCI needs information from the device or device controller, additional sensors or communication channels may be added, such as the 3-D accelerometer used in the wheelchair tilt control application described earlier. In the case of switch-scanning AT devices, the lack of feedback may prove particularly problematic for system-paced BCIs.

### Advantages for the BCI Community

As BCI research moves towards practical implementation in real-world settings with individuals with severe motor impairments, there is a growing need for theoretical frameworks, as well as hardware and software solutions to facilitate this transition. BCIs are merely one aspect of a complex technological and social network that may enable individuals with disabilities to maintain autonomy and the ability to live in relationship with others. The integration of this technology into the existing networks of prospective BCI users is crucial for its acceptance and use.[18] The MBOD presents an intermediate solution to allow researchers to interface laboratory BCIs with the users' preferred assistive technologies and output devices, enabling the systematic study of the effect of BCIs in these settings. The time and resources necessary to integrate with a device such as the MBOD are considerably less than would be needed to replicate all the functionality of existing assistive technology and output devices for each new BCI user. The advantages offered by the MBOD interface would ideally be built into any commercial BCI system; however, in lieu of such technology, this device is presented to the BCI community as a tool for the development of BCIs that are fully integrated into the technological and social lives of individuals who rely on non-motor channels to act and communicate. Source code and hardware design files (e.g., GERBER files) are available to any member of the BCI community upon request to the corresponding author.

## CONCLUSIONS

As BCIs transition from laboratory technologies to systems used by individuals with disabilities in real-world situations, it is important to begin systematically investigating the technical and social effects of integrating BCIs into the practical contexts of users' day-to-day lives. The MBOD is designed to facilitate this transition and has been successfully used as an intervening layer of hardware to enable BCIs to emulate keyboard, mouse or switch inputs to assistive technologies, computer operating systems and mobility devices. While its functionality should eventually be built into commercial BCI systems, this interface technology is currently available as a tool to the BCI research community to facilitate research toward the ultimate goal of enabling individuals with severe motor impairments to maintain their autonomy and their relationships with others in a real-world, day-to-day context.

## ACKNOWLEDGMENTS

The authors thank Stefanie Blain for her assistance and advice with manuscript preparation. Also Dean Camera, the creator of LUFA, for the availability of his library and his advice during the design phase of the project; finally, the creators and maintainers of BCI2000 for the availability of the platform and their assistance in getting up to speed quickly.

This work was supported in part by the National Institute of Child Health and Human Development (NICHD), the National Institutes of Health (NIH) under Grant No. R21HD054697, and the National Science Foundation (NSF) Graduate Research Fellowship under Grant No. DGE 0718128.

## DISCLOSURE AND CONFLICT OF INTEREST

D.E. Thompson and J.E. Huggins have no conflicts of interest in relation to this article.

## REFERENCES


1. Craig A, Tran Y, McIsaac P, Boord P. The efficacy and benefits of environmental control systems for the severely disabled. Med Sci Monitor 2005; 11(1): RA32-RA39.
2. Mason S, Birch G. A general framework for brain-computer interface design. IEEE Trans Neural Syst Rehab Eng 2003; 11(1): 70-85.
3. Mason S, Bashashati A, Fatourechi M, Navarro K, Birch G. A comprehensive survey of brain interface technology designs. Ann Biomed Eng 2007; 35(2): 137-169.
4. Cincotti F, Mattia D, Aloise F, Bufalari S, Schalk G, Oriolo G, et al. Non-invasive brain-computer interface system: towards its application as assistive technology. Brain Res Bull 2008; 75(6): 796-803.
5. Xiaorong G, Dingfeng, Ming C, Shangkai G. A BCI-based environmental controller for the motion-disabled. IEEE Trans Neural Syst Rehab Eng 2003; 11(2): 137-140.
6. Bergman E, Johnson E. Towards accessible human-computer interaction. In: Nielson J, (ed). Advances in Human-Computer Interaction. New Jersey: Ablex Publishing Corporation; 1995.
7. Ross, D. Implementing assistive technology on wearable computers. IEEE Intell Syst Appl 2001; 16(3): 47-53.
8. Wolpaw J, Birbaumer N, McFarland D, Pfurtscheller G, Vaughan T. Brain-computer interfaces for communication and control. Clin Neurophysiol 2002; 113(6): 767-791.
9. Schalk G, McFarland D, Hinterberger T, Birbaumer N, Wolpaw J. BCI2000: a general-purpose brain-computer interface (BCI) system. IEEE Trans Biomed Eng 2004; 51(6): 1034-1043.
10. Thompson D, Baker J, Sarnacki W, Huggins J. Plug-and-play brain-computer interface keyboard performance. 4th International IEEE/EMBS Conference on Neural Engineering, Antalya, Turkey 2009: 433-435.
11. Rebsamen B, Guan C, Zhang H, Wang C, Teo C, Ang M, Burdet E. A brain controlled wheelchair to navigate in familiar environments. IEEE Trans Neural Syst Rehab Eng 2010; 18(6): 590-598.
12. Thompson D, McCann M, Huggins J. Controlling wheelchair tilt with a brain-computer interface. 4th International BCI Meeting, Asilomar, California; 2010. http://bcimeeting.org/2010/poster_abstracts.shtml#R7.
13. Thompson D, McCann M, Lee C, Huggins J. Impact of wheelchair tilt on P300 brain-computer interface accuracy. Proc Neurosci 2010.
14. Salvaris M, Sepulveda F. Visual modifications on the P300 speller BCI paradigm. J Neural Eng 2009; 6(4): 046011.
15. Takahashi H, Hirooka S. Stereoscopic see-through retinal projection head-mounted display. In proceedings of SPIE 68031N-68031N-8. San Jose, CA, USA; 2008. http://spie.org/x648.html?product_id=765786.
16. Furdea A., Halder S., Krusienski D, Bross D, Nijboer F, Birbaumer N, Kübler A. An auditory oddball (P300) spelling system for brain-computer interfaces. Psychophysiology 2009; 46(3): 617–625.
17. Kübler A, Furdea A, Halder S, Hammer E, Nijboer F, Kotchoubey B. A brain-computer interface controlled auditory event-related potential (p300) spelling system for locked-in patients. Ann NY Acad Sci 2009; 1157(Disorders of Consciousness): 90-100.
18. Scherer M. Living in the State of Stuck: How technology impacts the lives of people with disabilities. Cambridge, MA: Brookline Books; 1993.