

Community + Culture features practitioner perspectives on designing technologies for and with communities. We highlight compelling projects and provocative points of view that speak to both community technology practice and the interaction design field as a whole. — **Sheena Erete, Editor**

Physiological Computing for All: Exploring Neural Interface Education

Chris S. Crawford, University of Alabama

In 2019, Meta purchased CTRL-Labs, a neural interface start-up, for more than \$500 million. A recent report by Morgan Stanley analysts valued the total addressable market of brain-computer interfaces (BCIs) at around \$400 billion in the U.S. alone. Headlines discussing the exploration of novel neural interface technologies by companies such as Blackrock Neurotech, Synchron, and Neuralink have become increasingly common. Although the media often piques our curiosity about this technology, few people truly comprehend its underlying mechanics. Over the past decade, I have dedicated my career to addressing this gap. This article shares experiences and insights obtained from introducing students to physiological computing through the Neuroblock software [1].

PHYSIOLOGICAL COMPUTING

Physiological computing [2] involves using physiological data from our bodies as inputs to a system or application. Major innovations in this space are typically driven by researchers seeking to address challenges in the medical field (e.g., neuroprosthetics). The advent of more cost-effective physiological sensors has led to the rise of nonmedical physiological computing applications. In human-computer interaction research, various types of physiological data, particularly muscle (EMG) and brain (EEG) signals, have been investigated for their potential as control modalities.

When introducing physiological computing to educators or students, their reactions often combine skepticism, curiosity, and a hint of

science fiction. This technology has roots dating back to 1786, when Italian physician Luigi Galvani identified links between frog muscle activity and electricity. Recently, HCI researchers have explored the use of physiological signals for a wide range of applications. These innovations have advanced physiological computing for the past 30 years. Research on hands-on approaches to educating the general population about physiological computing, however, is almost nonexistent, likely because most research featuring neural interfaces prioritizes system accuracy and speed.

NEUROBLOCK

My path toward realizing gaps in physiological computing education began in 2016, when my collaborators Marvin Andujar and France Jackson and I held the world's first public brain-drone race at the University of Florida. During this event, students used a consumer-grade EEG device to move a drone toward the finish line. The first person to get their drone across the finish line won and moved on to the next round.

The event attracted members of the local community and University of Florida students and faculty. Many of the attendees only recognized the

concept from movies. Students and educators showed enthusiasm for physiological computing, but there was a shortage of suitable resources and software for middle and high school classrooms. In response, I started developing Neuroblock [3], a platform designed to give students practical experience creating physiological computing applications.

Software Design. To my knowledge, Neuroblock is the first software that natively combines physiological sensor data, physical robots, and a block-based programming environment. The core interface shown in Figure 1 includes two interactive graphs on the left half of the user interface (UI). The top left graph features a visualization of brain activity data in real time. The bottom left graph provides feedback on extracted features. Each feature represents an EEG frequency band commonly mapped to different affective states. For example, high relative delta power levels are frequently linked to deep sleep states, while beta power is often used to interpret attention or mental workload levels. The toolbox shown at the center of the UI features multiple block categories. The data category features a set of predefined blocks that are mapped to each EEG frequency band power. The toolbox also features logic, loops, math, and drone block categories. Students use the scripting panes and action buttons to edit, test, and debug their programs. For example, the script shown in Figure 1 features code that will make the drone take off and move upward any time beta power passes a threshold of 30. Combined with scaffolded preliminary exercises,

Insights

- Physiological sensors can be leveraged to improve K-12 students' interest in STEM.
- Additional work is needed to improve the quality and inclusiveness of physiological sensors.



we have observed that Neuroblock may improve students' self-efficacy and increase curiosity about STEM.

MUSCLE-COMPUTER INTERFACE

While brain-computer interfaces tend to capture much attention, muscle-computer interfaces (MCIs) can be more suitable in certain situations. For example, MCI-based physiological computing tools can be used by students as early as fifth grade. During our exploration of these interfaces, students created programs that responded to the electrical activity in their muscles. The script shown in Figure 2 caused a video game character to jump when the EMG activity passed a specific threshold. A significant benefit of MCIs over BCIs is the straightforward connection between specific motor actions, such as squeezing a fist, and the corresponding data spikes displayed on the real-time graph. This clear mapping enhances understanding of the relationship between physiological events and data interpretation.

PHYSIOLOGICAL COMPUTING AND COMPUTATIONAL THINKING

One key goal of my previous work was to observe how computational thinking concepts and perspectives emerged during educational physiological computing exercises. Computational thinking (CT) has been acknowledged as a problem-solving process to perform computational tasks.

CT has often been utilized to assess students working with Scratch [4], a widely used visual tool for computer science education. One popular article describes computational thinking as a combination of concepts like sequences, loops, and conditionals, along with practices such as testing, debugging, and abstraction [5]. It also encompasses perspectives that focus on expression and connection. During our preliminary studies, unique patterns emerged as high school students learned to apply CT concepts with real-time physiological data.

CONCEPTS

Physiological variables. In Neuroblock, physiological variables are displayed as data blocks. These

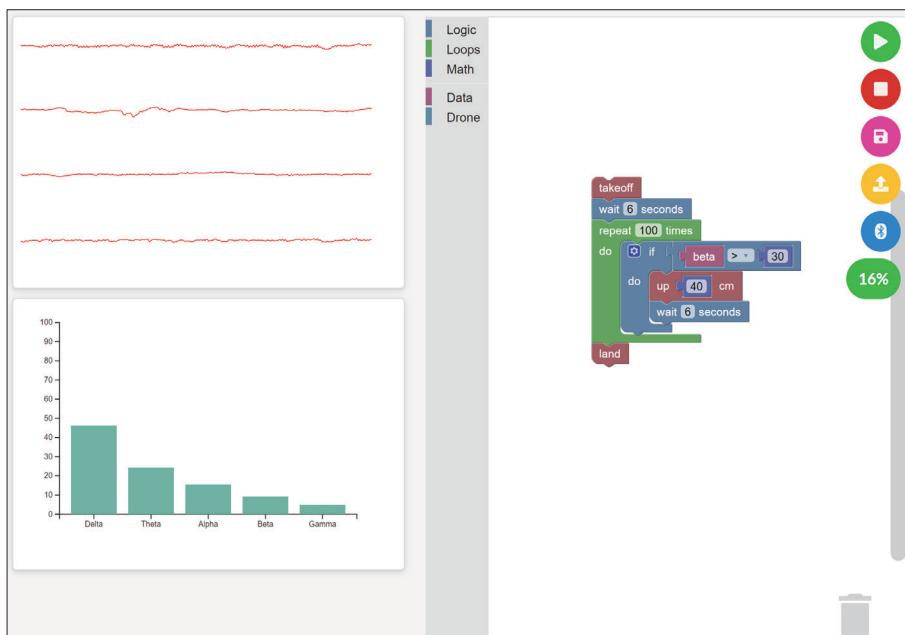


Figure 1. Neuroblock interface for brain-computer interface.

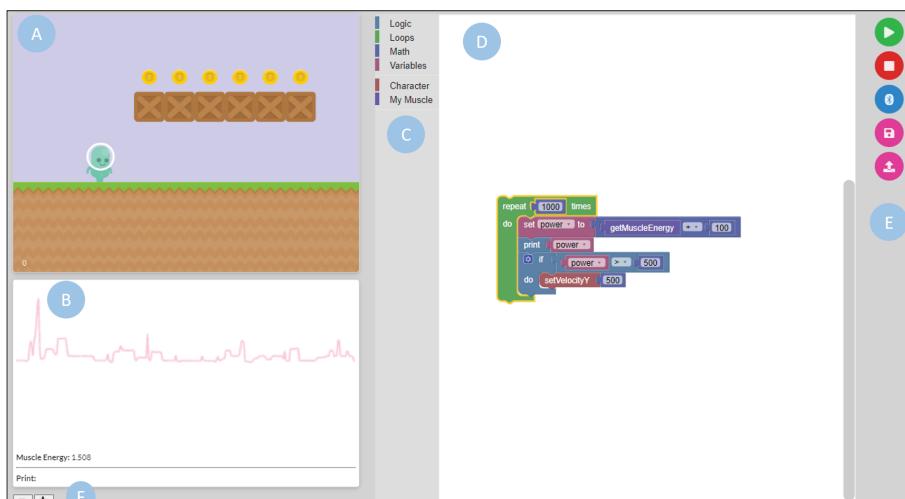


Figure 2. Example program created by a student that makes a character jump when the arm muscle is activated: game feedback (A), visual signal feedback (B), block toolbox (C), scripting pane (D), action buttons (E), and signal zoom controls (F).

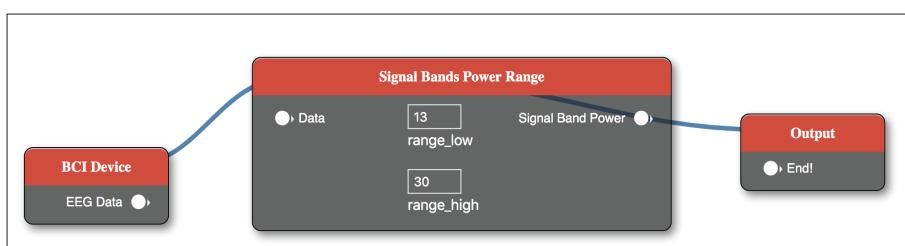


Figure 3. Flow-based interface for physiological variable construction.

blocks are continuously refreshed with the latest processed information sent from a physiological sensor. Students developed physiological variables by utilizing various input-output connections, represented in a flow-based interface (Figure 3). Each input-output

connection features a function that captures the physiological sensor data stream and returns a processed output. Students used this processed output to learn how to map simple physiological events (e.g., brain activity changes, squeezing a fist, mental arithmetic) to



Competitor participating in the inaugural 2016 brain-drone race.



Students testing Neuroblock code that maps EEG frequency band power to drone commands.

commands that controlled the behavior of a video game character (e.g., changing speed, jumping). By being introduced to this simple input-output feedback loop, students gained a clearer insight into the construction of physiological computing systems.

Physiological logic. We defined physiological logic as the combination of a physiological variable, a conditional statement, and a set of desired instructions. During group exercises, students learned how to implement physiological logic to create programs that make decisions based on specific physiological events. For example, students would start by identifying a physiological variable holding information commonly related to attention levels (e.g., beta frequency power). Next, the student learned to use a conditional if statement to evaluate

the value of the physiological variable (e.g., if beta greater than 0.75). Students would then nest code within the if block to trigger whenever increases in attention levels were observed (e.g., make character jump, increase character speed). When students learn to integrate physiological variables with conditional statements and desired outcomes, they gain insight into the fundamentals of physiological computing applications.

PERSPECTIVES

While analyzing recordings of sessions featuring Neuroblock, we also observed themes related to computational thinking perspectives. Three predominant patterns emerged across most sessions: physiological expressions, physiological design, and wearable experiences.

Physiological expressions and physiological design. Students demonstrated physiological expressions by naturally incorporating technical terms they learned in class when communicating with one another. For example, while testing her program, one student stated, “Look, girl, you a beta thinker.” In this example, the student used the term *beta* in reference to the physiological data block featured in her program. Furthermore, she verbalized this when she noticed changes in the beta-band data visualizer and the video game character’s position. In another instance, a student asked, “Can you go delta? I think that is the hardest one to get. You have to be asleep.” This comment refers to the delta frequency band, which is often used in research assessing deep sleep. These types of exchanges typically occur while students are in the planning stages of their projects. We use the term *physiological design* to describe the process of students creating code while considering constraints associated with physiological data.

Wearable experiences. During the Neuroblock activities, many students experience attaching themselves and their peers to a physiological sensor for the first time. This typically marks a significant moment of engagement for the participants. We leverage the term *wearable experience* to describe instances of students expressing experiences or attitudes related to the wearable device. For example, one high school student stated, “Thank God I didn’t wear makeup today.” Another was concerned that her wig might interfere with the EEG device’s performance. A similar concern

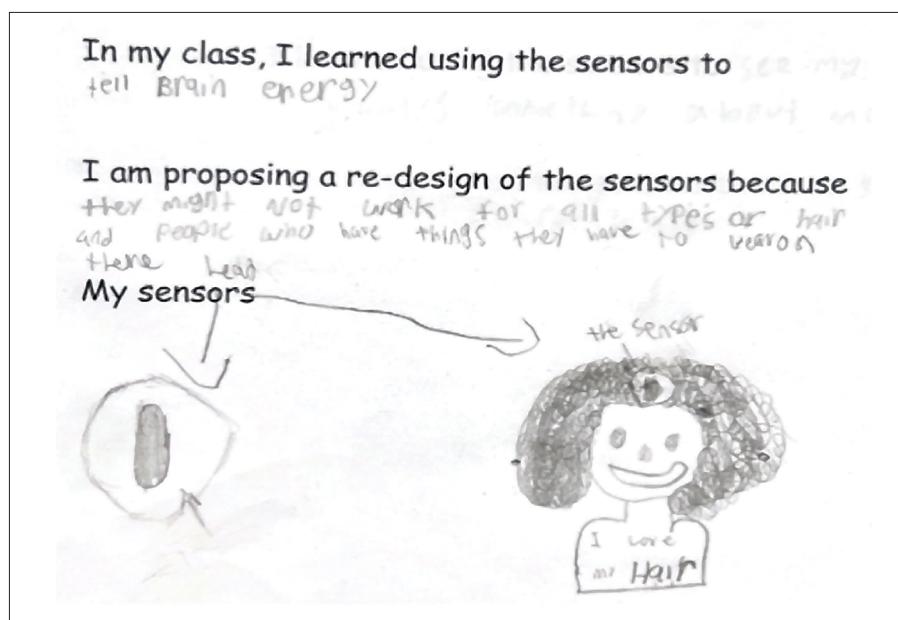
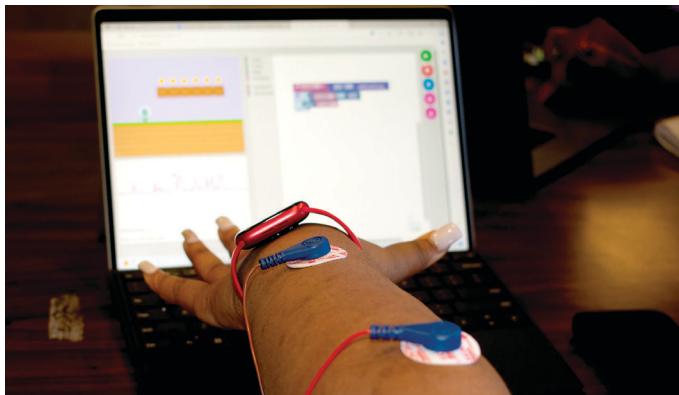


Figure 4. A fifth-grade student’s proposed redesign of an EEG-based physiological sensor.



Student equipped with a device that measures muscle activity.



Students engaging with Neuroblock software.

emerged when engaging a small group of fifth-grade girls with Neuroblock. Figure 4 shows an illustration produced by a student tasked with enhancing an existing EEG-based physiological sensor design [6]. The student mentioned that mounting the sensors could pose difficulties for individuals with coarse hair texture, which might affect many of her peers, particularly those who are African American. These responses were mostly related to the “questioning” of computational perspective. Gaining hands-on experience mounting and using the physiological devices may have empowered the students to ask questions regarding the limitations of current physiological wearable devices.

CHALLENGES AND FUTURE DIRECTIONS

Noise. A key component of all physiological computing systems is data processing. In particular, most physiological sensors pick up a combination of the desired signal (e.g., brain activity related to a specific task) and noise or artifacts (e.g., blinking of eyes, clenching of jaw, electricity from a laptop charger). Neuroblock includes a processing pipeline that allows for real-time signal filtering. The ever-changing environment of typical classrooms, however, makes it difficult to obtain clean and accurate data from consumer-quality physiological sensors. The primary objective of our work is not to achieve the most precise signal but rather to inspire students through innovative technologies. This exposure is intended to enhance their interest and curiosity in STEM fields. As a

result, we often build on noise-related challenges by intentionally showing students how certain actions influence the physiological sensor data. Students usually doubt that EEG sensors are capturing data from their bodies. This skepticism changes when they witness the line graph react to their eye blinks. The noise generated by blinking is often disregarded and considered undesirable in conventional physiological computing studies. Embracing noise, however, can be valuable to learning in physiological computing education.

Ethical concerns. As personalized physiological data becomes increasingly common, it is essential to address potential ethical issues. One key concern is users’ control of the physiological data storage and post-capture use. Neuroblock seeks to mitigate this issue by ensuring data is not stored on a remote server [7]. All operations required to process the physiological data are managed directly within the user’s local browser. This approach eliminates the need for remote processing. Restricting physiological applications to local processing can enhance privacy, but utilizing cloud-based solutions could improve the accuracy of neural interfaces over time. Engaging in discussions about these trade-offs is essential for responsible advancement of the field of physiological computing.

ACKNOWLEDGMENTS

This material is based on work supported by the National Science Foundation (awards #2045561 and #1838815). The Neuroblock project

has also received support from the Alabama Power Foundation and Intel Corporation.

ENDNOTES

1. What is NeuroBlock? Human Technology Interaction Lab at the University of Alabama; <https://htilua.org/neuroblock>
2. Fairclough, S.H. Fundamentals of physiological computing. *Interacting with Computers* 21, 1-2 (2009), 133–45.
3. Hernandez-Cuevas, B., Egbert, W., Denham, A., Mehul, A., and Crawford, C.S. Changing minds: Exploring brain-computer interface experiences with high school students. *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, New York, 1–10.
4. Resnick, M. et al. Scratch: Programming for all. *Communications of the ACM* 52, 11 (2009), 60–67.
5. Brennan, K. and Resnick, M. New frameworks for studying and assessing the development of computational thinking. *Proc. of the 2012 Annual Meeting of the American Educational Research Association* 1. AERA, 2012, 25.
6. Luo, F., Liu, R., Awoyemi, I.D., Crawford, C.S., and Nasrin, F. Novel insights into elementary girls’ experiences in physiological computing. *Proc. of the 55th ACM Technical Symposium on Computer Science Education* 1. ACM, New York, 2024, 764–70.
7. Stegman, P., Crawford, C.S., Andujar, M., Nijholt, A. and Gilbert, J.E. Brain-computer interface software: A review and discussion. *IEEE Transactions on Human-Machine Systems* 50, 2 (2020), 101–15.

Chris S. Crawford is an associate professor in the University of Alabama’s Department of Computer Science. His research focuses on human-robot interaction and brain-computer interfaces. He has investigated systems that provide computer applications and robots with information about a user’s cognitive state. crawford@cs.ua.edu