Fostering Computational Thinking Through Neural Engineering Activities in High School Biology Classes

Ido Davidesco[1], Na'ama Y. Av-Shalom[1], John Settlage[2], Christopher Rhoads[2], Leslie G. Bondaryk[4], Aaron Kyle[3], & Bianca Montrosse-Moorhead[2]

[1] Boston College [2] University of Connecticut [3] Duke University [4] Concord Consortium

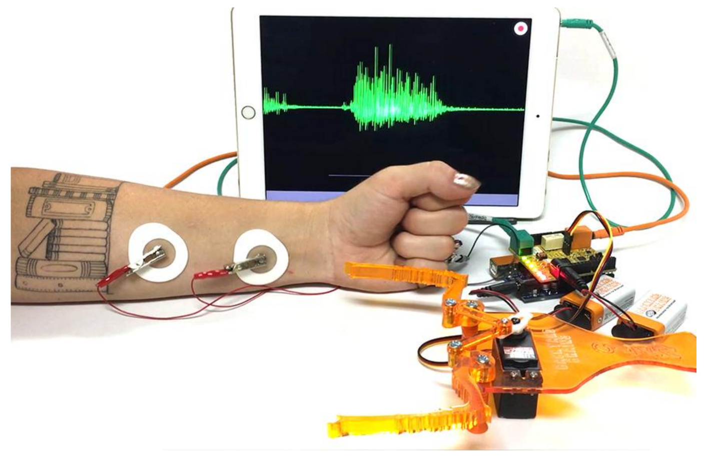
INTRODUCTION

Computational thinking (CT) practices, such as pattern recognition and problem decomposition, are embedded in virtually every STEM discipline, and CT is outlined as a central practice of science and engineering in the Framework for K-12 Science Education (National Research Council, 2012). However, most existing K–12 CT education efforts focus on programming or computer science courses (Hsu et al., 2018), which are only taken by a fraction of students (Computer Science Teachers Association, 2019). Therefore, there is a critical need to integrate CT into other STEM disciplines to broaden access, and so that students can engage with the inter-disciplinary nature of CT. **The current project aims to incorporate CT within two STEM disciplines: engineering and biology.**

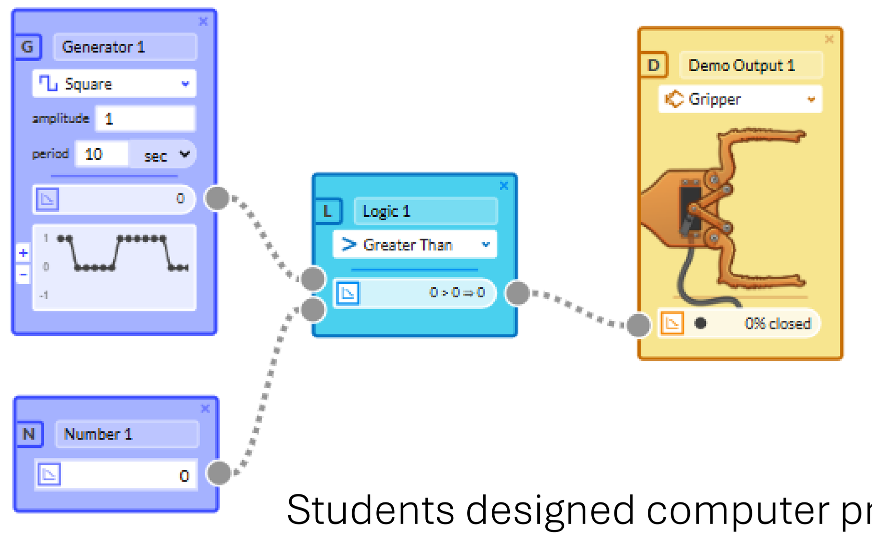
2023–24 IMPLEMENTATION

- 5 teachers
- 7 classes in 5 high schools in Northeastern USA
- ~325 students (172 consented)
- Grades 9–12
- Approximately one month of implementation per class

TECHNOLOGY TOOLS



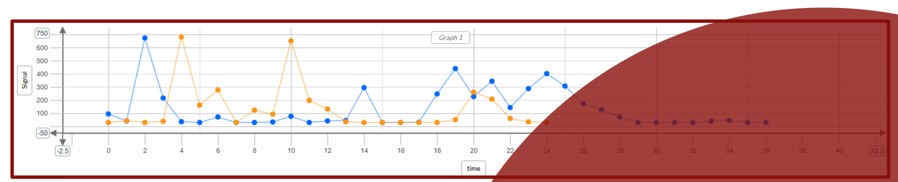
Students used a prefabricated EMG-controlled gripper throughout the unit.
(image credit: Backyard Brains)

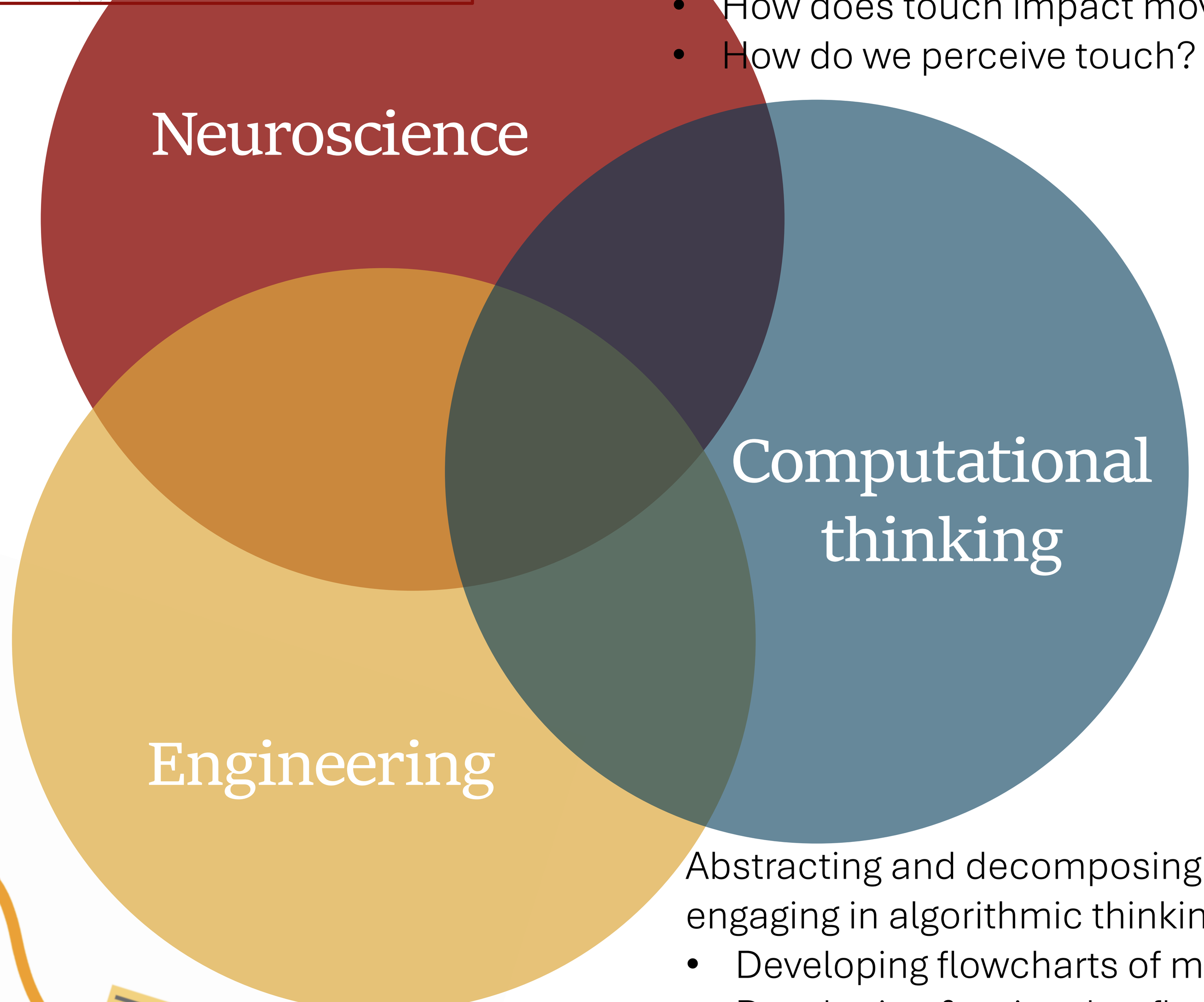


Students designed computer programs using "DataFlow," a node-based programming environment developed by the *Concord Consortium* (Bondaryk et al. 2021).

THE CURRICULUM: NEURAL ENGINEERING

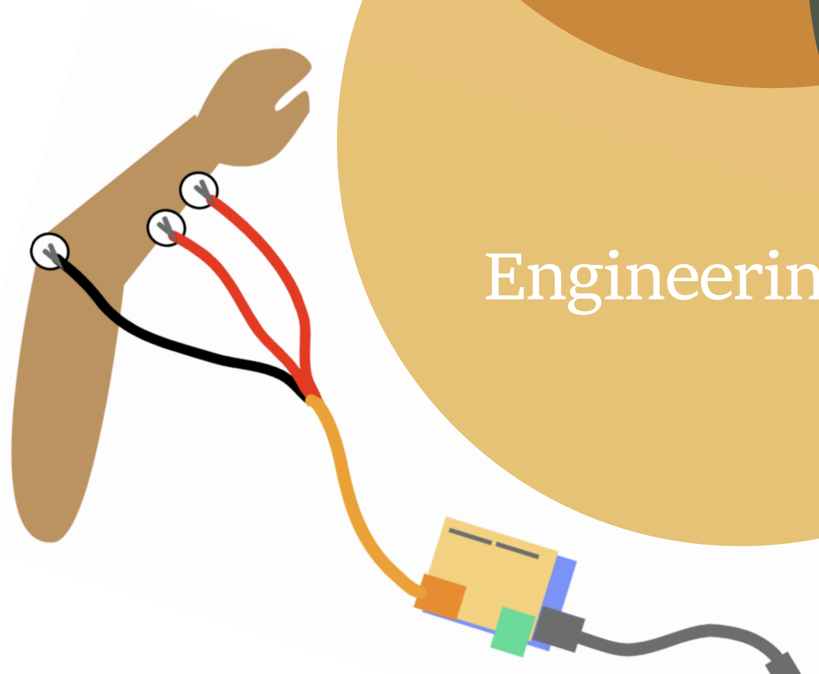
Students were introduced to Tilly, a teenager who has received bionic arms after an amputation. Throughout the unit, students collected their own nervous system data (e.g., reaction time) and were supported towards developing a basic prototype of a bionic arm.



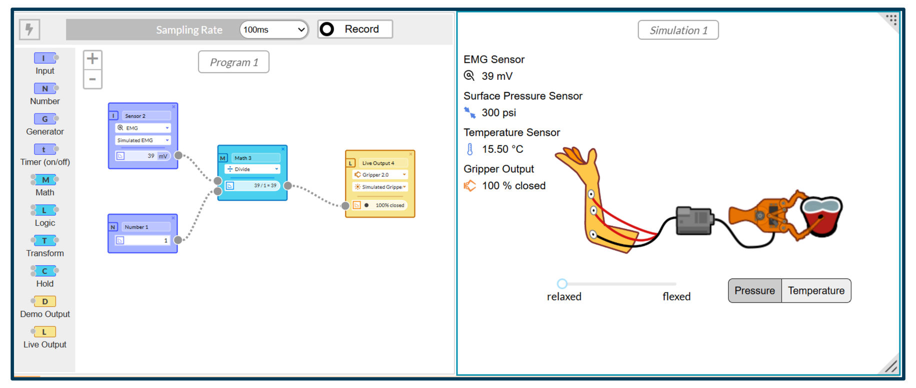


Nervous system:
- How do nerves and muscles work?
- How does touch impact movement?
- How do we perceive touch?

Abstracting and decomposing ideas, engaging in algorithmic thinking
- Developing flowcharts of mechanisms
- Developing & using dataflow programs
- Working with simulated data at varying levels of abstraction



Understanding engineering problems, developing and testing solutions:
- How might a bionic arm work?
- Can robots sense objects?
- How can you control a robotic gripper?
- Managing the complexity of real data



STUDENTS: QUANTITATIVE RESULTS

Instrument	Subscale	Pre: mean (SD)	Post: mean (SD)
Computational Thinking Scale (CTS) (Tsai, et al., 2021)	Overall	4.98 (0.78)	5.05 (0.91)
	Abstraction	4.95 (0.89)	5.04 (1.03)
	Decomposition	4.59 (1.04)	4.77* (1.07)
	Algorithmic thinking	5.15 (0.89)	5.14 (1.01)
	Evaluation	5.02 (0.97)	5.11 (1.02)
	Generalization	5.04 (0.87)	5.11 (0.96)
Additional item: "I can use flowcharts to solve a problem"		3.90 (1.40)	4.50** (1.33)
Engineering Design Survey (EDS) (Carberry, et al., 2010)		4.53 (0.98)	4.82** (1.15)
S-STEM (Unfried et al., 2015)		3.66 (0.82)	3.74 (0.89)

Note. *p<0.05, **p<0.01; of the areas in the CTS, abstraction, decomposition, and algorithmic thinking were foregrounded in the curriculum, whereas evaluation and generalization were not.

STUDENTS: NEXT STEPS

- What led to the quantitative findings?
  - Curriculum analysis: was decomposition foregrounded more than expected, or other areas backgrounded?
  - Qualitative analysis of student artifacts: how did students interpret or engage with the activities?

TEACHERS: PRELIMINARY QUALITATIVE RESULTS

- Explored teachers' feedback from pre-implementation PD and post-intervention interviews
- Teachers were excited by and interested in the unit, but felt that they needed more support

It was really awesome. I was glad that I could present something that is new … that my students were able to do a true STEM activity of the arm that was created for people to use.

The conversations and the ideas that they were sharing … I thought that was really good. And I really liked that engagement piece and that thinking piece of it.

I was excited to do something that I didn't know how to do … it's like this is an opportunity for me to like stretch myself and like learn new aspects of it.

I, as a teacher, know that I need to take a course in coding … I needed my hand held a little bit more so that I could make sure I understood everything, so that I could impart it to the students.

I feel like I was doing a lot of work to like understand every lesson before I gave it so that I could fully explain it.

I wasn't too confident about in how to do [some of the Dataflow activities] and what to put for a different thing. So, when the students were struggling, I feel like I wasn't confident enough in some of that and how that worked with the, okay, 'which block do I put in here?'

TEACHERS: NEXT STEPS

- What supports do biology or general science teachers need to effectively implement the neural engineering unit?
  - Teacher follow-up interviews focusing on content and pedagogical knowledge around CT and engineering practices

Ido.Davidesco@bc.edu

Naama.Av-Shalom@bc.edu

Additional results from an earlier implementation:
Aldemir, T., Davidesco, I., Kelly, S. M., Glaser, N., Kyle, A. M., Montrosse-Moorhead, B., & Lane, K. (2022). Investigating Students' Learning Experiences in a Neural Engineering Integrated STEM High School Curriculum. *Education Sciences*, 12(10), 705.

REFERENCES

Bondaryk, L. G., Hsi, S., & Van Doren, S. (2021). Probeware for the modern era: IoT dataflow system design for secondary classrooms. *IEEE Transactions on Learning Technologies*, *14*(2), 226–237.
Carberry, A. R., Lee, H., & Ohland, M. W. (2010). Measuring Engineering Design Self-Efficacy. Journal of Engineering Education, 99(1), 71–79. https://doi.org/10.1002/j.2168-9830.2010.tb01043.x
Computer Science Teachers Association, Code.org Advocacy Coalition, & Expanding Computing Education Pathways Alliance. (2019). 2019 State of Computer Science Education.
Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. Computers & Education, 126, 296–310. https://doi.org/10.1016/j.compedu.2018.07.004
National Research Council. (2012). A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas. National Academies Press.
Tsai, M.-J., Liang, J.-C., & Hsu, C.-Y. (2021). The Computational Thinking Scale for Computer Literacy Education. Journal of Educational Computing Research, 59(4), 579–602. https://doi.org/10.1177/0735633120972356
Unfried, A., Faber, M., Stanhope, D. S., & Wiebe, E. (2015). The Development and Validation of a Measure of Student Attitudes Toward Science, Technology, Engineering, and Math (S-STEM). Journal of Psychoeducational Assessment, 33(7), 622–639. https://doi.org/10.1177/0734282915571160
Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127–147.

This material is based upon work supported by the National Science Foundation under grant number 2101615. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.