2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #52603

# Engaging Computer Science Students in ECE and Robotics through Fabrication and Micro-Credentials

**Dr. Stewart Thomas, Bucknell University**

Stewart J. Thomas received the B.S. and M.Eng. in Electrical Engineering from the University of Louisville in Louisville, Kentucky in 2006 and 2008, respectively, and the Ph.D. in Electrical and Computer Engineering from Duke University in Durham, North Carolina in 2013. He has served on the organizing committee for the IEEE International Conference on RFID series since 2014, serving as the Executive Chair in 2022, with research interests in areas of low-power backscatter communications systems and IoT devices. He is also interested in capabilities-based frameworks for supporting engineering education. He is currently an Assistant Professor at Bucknell University in the Electrical and Computer Engineering Department, Lewisburg, PA USA.

**Dr. Alan Marchiori, Bucknell University**
**Matthew Lamparter, Bucknell University**

# Engaging Computer Science Students in ECE and Robotics through Fabrication and Micro-Credentials

## Introduction

This action-based research paper discusses the development and implementation of a low-cost robotics system and integration of associated microcredentials into an introductory ECE course targeting undergraduate computer science students. Historically, participation in campus maker spaces by computer science has been low compared to ECE or mechanical engineering students. There could be many reasons for this, such as stigma of makerspaces being hardware-oriented, as opposed to more software-first spaces, biases about hands-on making abilities, or other social stigmas or pressures. No matter the reason, this can leave computer science students at a disadvantage. For engineers, rapid prototyping or other "maker" skills are important to facilitate experimentation and design. Lacking such exposure or skills can lead to further siloing of computer science students from other engineering disciplines, and limit cross-disciplinary opportunities.

A recent curricular change created an opportunity to create a structure for computer science student to engage with campus makerspaces. Before this change, all computer science & engineering students were required to take a traditional sequence of an introductory circuit analysis course that focused on dc analysis and basic electronics followed by a digital logic course that moved through logic gates and minimization to FPGA design of FSM-based controllers. The curricular change removed this two course sequence and replaced it with a new course. In this course, the focus shifted towards basic dc circuit analysis techniques and embedded systems. It is through this change, that an opportunity for makerspace engagement emerged. In the most recent offering, the course required computer science students to create a small, autonomous robot which they built from laser cut and 3d printed components which students fabricated in on-campus makerspaces. It is the course structure itself that created opportunities for computer science students to develop their own making skills, actively engage in the campus makerspace community, and work on hands-on projects, while covering basic ECE topics like circuit analysis, breadboard prototyping, and general ECE lab skills.

Details of the course structure and the robot are described later in this paper. The construction of the course robot required that students be trained and credentialed in 3d printing and laser cutting. By relying on commodity and constructed parts, the overall cost of the project is around $50 per robot and most of the components can be reused, making it affordable and accessible to a wide audience. These robots were then programmed using microPython to compete in a "Robot Olympics" that served as a final course project.

This structural change and requiring students to use makerspaces to create robot, leads to our research question: Does requiring makerspace microcredentials for a course project lead to sustained makerspace usage by CS students? This course ran for the first time in the fall 2024 semester, where students were not asked to make and build their own robot, and then again in fall 2025 in which students used the makerspace to build their own robot. In this paper, we analyze the available data (as of January 2025) regarding makerspace usage by computer science students, describe the overall course, and the specific robot intervention. Based on this data, we see an increase in makerspace usage by computer science students, and note strong self-reported interest in makerspaces.

**Background**

The integration of makerspaces into higher education engineering curricula has shifted from a novelty to a central pedagogical strategy for fostering innovation, retention, and skill development. With roots in Papert's constructionism, "learning-by-making" is seen as an important learning process that engages creation and manipulation of physical artifacts combined with social discussion and interaction [1]. This theory has been especially popular in engineering education where makerspaces can provide the physical and social infrastructure necessary to support a transition from students being a passive consumer of information to an active producer of technology [2]. As engineers are the creators of technology and can accelerate societal technological innovation [3], the opportunities which makerspaces provide are significant. It is important that computer science students be included in such communities, recognizing the role of software and technology in the modern, built world.

Traditional engineering coursework often separates theory from practice with separate lecture and lab sections, imposing a structure where lab work exists in different spaces and modes of work than work associated with more theoretical lectures. This is particularly true in computer science curricula with a narrower focus on heavily abstracted (from the physical computing hardware) software environments. Makerspaces offer an alternative environment that mixes formal and informal learning separate from the normal rules that may govern formal learning environments [2]. Research indicates that access to fabrication tools (e.g., 3D printing, laser cutting, CNC routing) through makerspaces significantly improves students' "engineering self-efficacy" over time, which is otherwise found to decrease over time in an undergraduate career [4]. For computer science students specifically, who may be more apt to view hardware as a magic black box, the opportunity and act of fabricating their own robot chassis out of plastic and wood or soldering a custom circuit board bridges the gap between theoretical digital logic processes and physical reality.

Beyond technical skills, makerspaces also play a role in community formation and belonging which are important factors for student retention overall wellbeing [5], [6]. Makerspaces are known to be important in community forming [7], and who has access and is represented significant considerations [8]. Often in engineering, leaving a program may not be driven by a lack of academic ability, but instead by a lack of connection. Makerspaces function as community of practice, not functionally dissimilar to 'third spaces', where novices and experts can work and interact an informal setting. If such spaces are therefore important for university thriving, by integrating makerspace certification directly into curriculum, it may be possible to create structures that lower barriers of entry and expand access such that students feel both entitled to and welcome use these facilities as part of their growing professional identity.

With this importance in mind, the project used to bring computer science students into the makerspace was selected to be a robotics project. The specific robot is a custom design based on the rules of the Open Robotics Platform [9]. However, there are many other robots that could have been selected. This work shares some resemblance to Flower∞Bots [10] which use a similar chassis structure and also swappable or extensible parts. Other early options include the MBot [11]. There are many similar open source or inexpensive commercial robot options. However, a key goal of the project was for students to be involved in the fabrication and assembly of the project. For this reason, it was important to start with simple design plans that students could then fabricate on their own. The Open Robotics Platform was found to be a simple, but effective set of design rules that allows for expansion.

Having students create and assemble their own robot matched the active pedagogy of the course that incorporated components of project based learning (PBL) [12]. Early in the course, student simulated autonomous robots using WeBots software [13]. Later, students were asked to design a set of programs for their robot that would compete in six different 'events.' Students had some choice in events they wanted to do, and could determine how to implement their solutions. Such PBL components are known to affect student professional and identity development [14], [15].

**Robot Platform Description**

The robotic platform developed for this course is a two-wheel, direct-drive autonomous system built upon the Open Robotic Platform (ORP) [9] design principles. Unlike typical educational kits that arrive as a collection of prefabricated metal or plastic components, the ORP-based robot requires students to engage in the "making" process from the outset. The chassis consists of two 160 mm circular plates following a standardized 20 mm grid pattern with 3.5 mm holes that are laser-cut by the students. Figure 1 shows an example of an assembled robot with a complete hardware list detailed in Table 1.

The central theme of the platform is the movement away from specialized "black-box" hardware to open design standards. Where typical educational robotics platforms rely on a proprietary chassis and mounting brackets, this platform enabled student-fabricated 3D-printed motor holders and sensor brackets to standardize the interfaces between the ORP plates and non-standard parts like the TT gear motors and sensors. By requiring students to conform to standard design rules and fabricate the needed parts, they gain practical experience with additive manufacturing, tolerances, and assembly and reinforces the need and benefit of standards.

This fabrication-first approach addresses the "hardware stigma" by giving CS students agency over the physical form of their robot. Students are encouraged to modify the CAD files for the chassis and brackets, transforming the robot from a generic tool into a personalized artifact. The complete construction manual is available at `http://burl.live/build`.

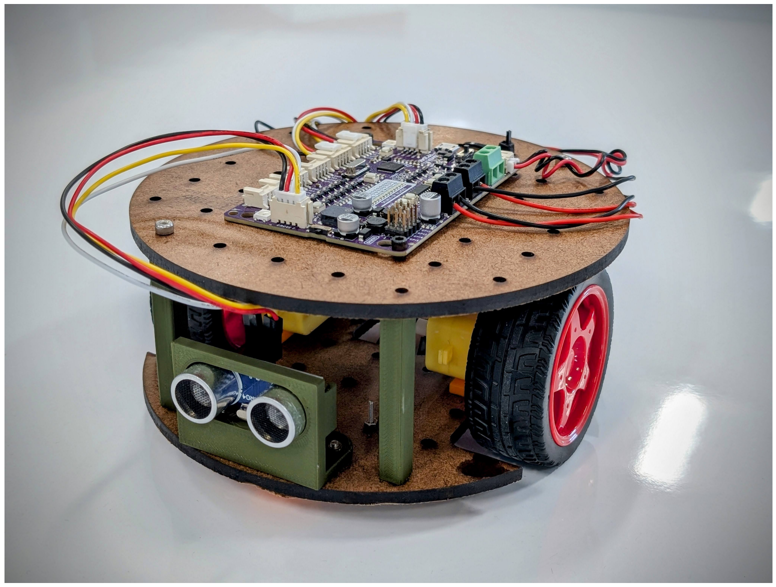


Figure 1: A completed robot.

**Software Architecture**

The software environment is designed to bridge the gap between high-level CS concepts and low-level ECE implementation. While many other platforms might use block-based programming or low-level C/C++, we employ microPython to leverage CS student's existing skills and experience with Python for rapid development. The architecture follows an object-oriented approach, requiring students to interact with three primary classes: Motor, LineSensor, and Robot.

The Motor class abstracts the Pulse Width Modulation (PWM) details for each wheel through a calibrated linear model (PWM $= m \cdot v + c$), where $v$ is the desired linear velocity at the wheel. This requires students to perform physical calibration to determine the unique slope and intercept for their specific robot, reinforcing the connection between code and physical reality.

A key requirement is that students must control the robot within a coordinate space defined by linear velocity ($v$, in mm/s) and angular velocity ($\omega$, in rad/s). Rather than being provided with a pre-written library for movement, students are tasked with deriving and implementing the kinematic control equations for a differential drive system. This requires them to translate the desired global robot velocities $(v, \omega)$ into individual wheel velocities $(v_L, v_R)$ using the robot's physical dimensions which can then be translated to PWM duty-cycle for each wheel to achieve the desired motion.

The LineSensor class manages the Pololu sensor array, reading the sensors and returning a physical offset in millimeters between the center of the robot to the center of the line. This allows students to implement a Proportional-Derivative (PD) control loop, where they tune $K_p$ and $K_d$ gains to achieve smooth navigation by minimizing this error.

Finally, the Robot class provides the high-level control by implementing various FSMs that read

Table 1: Robot Hardware Components, Fabrication Methods, and Procurement

| **Component** | **Qty** | **Fabrication Method / Source** | **Cost ($)** |
|---|---|---|---|
| *Chassis and Mechanical Structure* | | | |
| Circular Chassis Plate (160mm) | 2 | Laser-cut Hardboard (ORP Standard) | $\approx$ $3 |
| 3D Printed Motor Holders (L/R) | 2 | 3D Printed (PLA) | |
| 3D Printed Ultrasound Bracket | 1 | 3D Printed (PLA) | |
| Connection Rods (55mm Spacers) | 4 | 3D Printed (PLA) | |
| Caster | 1 | 3D Printed (PLA) | |
| TT motors | 2 | Commercial Off-The-Shelf (COTS) | $2.50 |
| Wheels | 2 | COTS | $2.25 |
| *Electronics and Sensing* | | | |
| Maker Pi RP2040 Controller Board | 1 | Cytron | $11.22 |
| QTRX-MD-06RC 6-Channel Line Sensor | 1 | Pololu | $11.78 |
| Ultrasonic Distance Sensor | 1 | HC-SR04 or clone | $3.95 |
| LiPo Battery 18650 (3.7V) | 1 | COTS | $8.99 |
| Grove Interconnect Cables | 1 set | Seeed Technology | $4.30 |
| *Fasteners and Assembly Hardware* | | | |
| M2/M3 Hardware (Screws and Nuts) | 1 set | COTS (6mm to 25mm lengths) | $\approx$ $2.50 |
| M3 Threaded Inserts | 8 | COTS (Heat-set) | $1.78 |
| **Total Estimated Cost** | | | **$52.27** |

the LineSensor, perform PD speed control, and set both motor PWM duty-cycles appropriately. Students implement high-level behaviors by transitioning between various states e.g., follow line, turn left, follow line, etc.

Our approach requires students to develop a deep understanding of the robot's hardware and system calibration. This ensures that students do not merely combine several calibrated functions but instead build a functional software stack that accounts for the imperfections of their fabricated hardware.

## Course Context

As mentioned, recent curricular changes in the computer science program removed a required two-course sequence that was replaced with a single course. The developed, single-semester course covers dc analysis, resistive networks and digital logic basics in the context of embedded systems. The course, titled *Electrical & Computer Engineering Systems*, is a 15-week course with three 50-minute lectures per week and a weekly two-hour lab component. It was originally offered in the Fall 2024 semester, and most recently in the Fall 2025 semester. In the Fall 2025 offering 24 students were enrolled who were all computer science majors of varying levels: sophomore (N=2), junior (N=16), senior (N=6). In the initial offering, a pre-built robot was used, whereas in Fall 2025 the robot project described in this paper was introduced.

The course is structured to guide students through the analog and digital domains with an emphasis on embedded systems and the co-design or intersection of hardware and software. The first third of the course focuses on analog systems fundamental electrical engineering concepts such as Kirchoff's Laws, dc analysis, and equivalent circuits. The second third moves into the digital domain with ideas such as data representation and Boolean algebra discussed in the context of programming a Raspberry PI RP2040 with microPython. This module also discussed serial data transfer and interfacing between sensors and digital systems, including ADC and DAC conversions. This sequences culminated the final module of the course where students built the robot described above and developed software for the robot to complete the events of a "Robot Olympics" competition.

While the Robot Olympics was fundamentally a competition between the robots, importantly course grades were not tied to place or rank order, but instead to simply completing and finishing the various events. The competition was therefore, a way to allow students to complete the work to an appropriate level to their ability, while still encouraging collaborative problem solving between groups. Example events included Curling where robots traveled toward a wall and stopped at a specified distance from a wall; 1-m dash in which robots raced heats of 1 m; and marathon where robots traveled laps around a course, avoiding obstacles. Most events were based on line-following algorithms and typically required integration of other sensors or actuators alongside the movement code.

Throughout the lab component of the course, students would vary between simulation of various robot control components using the WeBots platform, or practicing skills such as breadboarding or using bench-top test equipment to test and validate theories from the lecture section. During the final module of the course, students used the lab section to assemble the robot components, and then develop code such as line following algorithms that would be used in the Robot Olympics.

**Methods (Data Collection)**

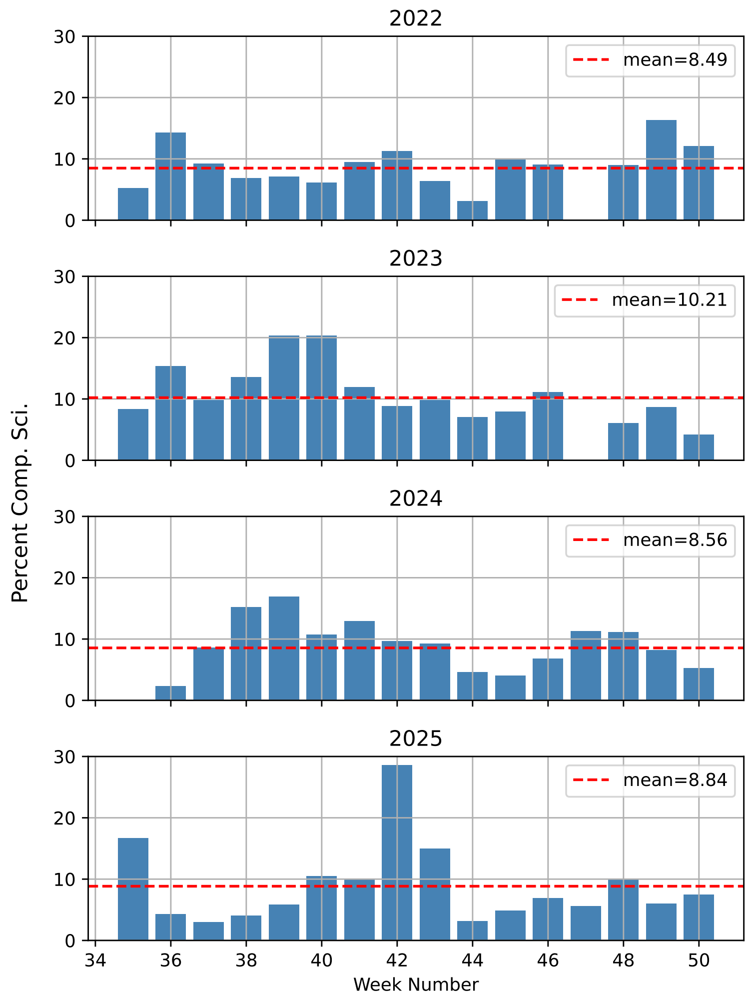
Data was collected and analyzed after IRB submission and approval[1]. Data was collected in two ways. First, equipment usage of a campus makerspace was analyzed. In this makerspace, each piece of equipment requires users to be trained and certified before use. Once certified by staff, users use a NFC card to verify credentials and operate the tool. We analyzed this data to look for trends in usage by comparing usage by major across weeks. Historic data prior to the new course being offered was compared to data from Fall 2025 when students used the makerspace to construct their robot. This allowed trends in percentage of computer science students using maker equipment, compared to all usage of equipment by week.

A survey instrument was also administered in the final week of the course to record responses from student to understand their interest and motivation in the project. This survey data included both likert-scale questions and free response questions. The participants in this survey were 16 computer science students in their second, third and fourth year of undergraduate study who participated in the Fall 2025 course.

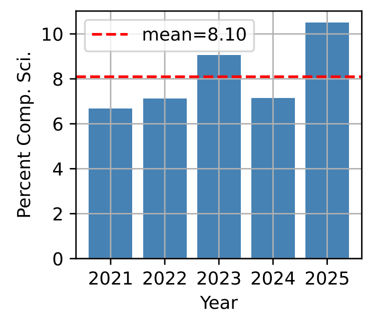
**Results & Discussion**

In total, there were 10,829 machine activations from 832 unique users across the four years examined. Data was analyzed to look at the number of unique users per week by major. Figure 2a shows a significant increase makerspace use by computer science students in week 42 when the

[1]IRB #: 2526-058




(a) Weekly trends.



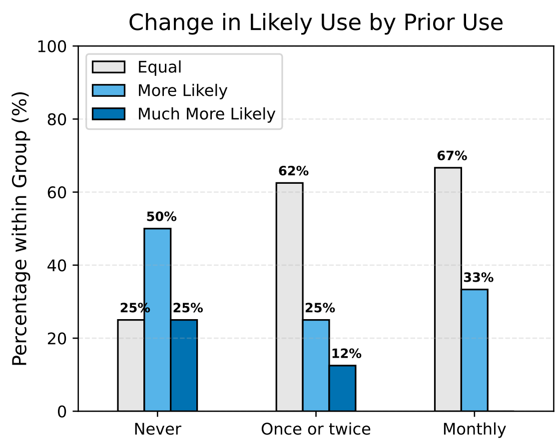

(b) Total users.

Figure 2: Percent of unique makerspace users in the fall semester with a declared computer science major over a 4-year period, (a) per week and (b) per fall semester.

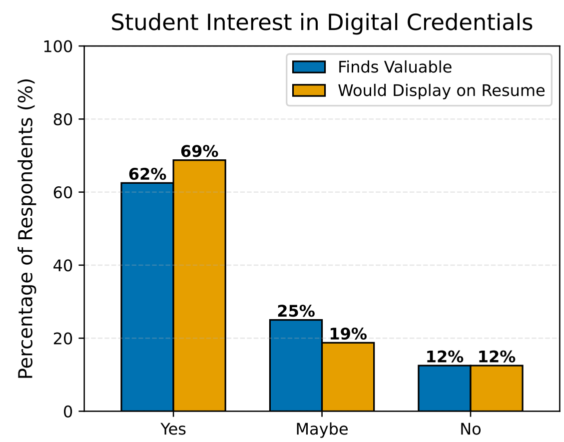
project was assigned, where nearly 30% of all unique users to the makerspace for that week were computer science students. Over the entire semester (Figure 2b), the fraction of makerspace users from computer science achieved a 5-year peak, exceeding 10%.

Responses to the survey suggest that the structured integration of makerspace activities can increase makerspace use by CS students. This is supported by data for likert-scale questions illustrated in Figure 3a.

Among students who had never used the makerspace before the course, 75% reported they were now more likely (50%) or much more likely (25%) to use the space in the future. For students who were already regular users, the impact was not as strong, suggesting that they may already see value in maker-related activities. Importantly, no student reported a decrease in interest in the




(a) Likely makerspace use




(b) Interest in digital credentials

Figure 3: Post-course student survey results for reported change in (a) likely makerspace use and (b) interest in digital credentials for making.

makerspace after the course.

The survey also gauged student interest in the micro-credentials required for equipment operation within the makerspace. As shown in Figure 3b students saw value in formalizing their makerspace skills; 62% of respondents reported that they found the digital credentials valuable and 69% of students indicated they would display these making credentials on their professional resumes. Tying projects and credentialzing with exportable items that can be displayed as professional certifications is something that students value.

By integrating these credentials directly into the curriculum, the course successfully created a structure where students felt both entitled to and welcome in the makerspace. This cross-disciplinary exposure is critical for computer science students to avoid further siloing and to better prepare them for design and experimentation in engineering fields.

In the survey instrument, students had an option to answer free-response questions. These questions are shown in Table 2. The responses to Q1 highlighted the communal aspect of the makerspace areas. Students used the space because it was required of the course and project, but over 60% of responses reported they were likely to return to the makerspace areas for personal projects or social interactions. One student wrote that:

> *I never used to go into the Makerspaces to build things. Now I love to work in there and sometimes build mini-projects.*

Another student noted that:

> *Using the makerspace for the robot helped me learn a new skill and get certified in laser cutting. I thought it was a great experience, as well as the student who helped me get certified was very helpful and kind. Having multiple skills like 3d printing and laser cutting allows me to do projects on my own, as well as encouraging me to get*

Table 2: Free Response Questions in Survey

| ID | Question Text |
|---|---|
| Q1 | Describe the impact of using the makerspace for robot construction had on your confidence and subsequent engagement with the makerspace. (For example, were you more likely to use the makerspace after this project or for other purposes once you were certified?)<br>If you feel the makerspace access created significant obstacles, or did not benefit your learning, please provide suggestions on how the certification process, training, or facilities could be improved. |
| Q2 | What is one thing you would change about the robot build process to make it a better learning experience? |
| Q3 | Describe how you feel this project has helped you better understand the hardware side of computing (e.g., how the software interacts with physical components, the fabrication process, or the limitations of embedded systems).<br>If you feel the project did not significantly improve your understanding of the hardware side, please provide suggestions on how the project or curriculum could be improved to better bridge the gap between software and physical computing. |

> *certified for more.*

The social barrier of entry was lowered as the project required makerspace interactions, and the majority of students found a welcoming space, learned new skills and envision further projects to work on. The main data source analyzed was tool usage, but these comments suggest that analyzing door may reveal usage trends for students who use the space for social interactions but not 'making.'

Responding to Q2, students generally viewed the robot build process positively with approximately one third of respondents satisfied and not suggesting changes. Changes that were suggested primarily focused on increased autonomy and 'student-designed' elements (i.e., customizations). This desire to individualize the robot or for more open-ended work was paired with requests for more preparatory work (e.g., breadboarding and debugging skills; better understanding of key robot drive equations). As might be expected, this indicates a perennial Goldilocks problem of balancing scaffolding and open-endedness.

In response to Q3, student responses indicate the project helped provide exposure for more hands-on work with physical computing and also helped to increase students' confidence in working with such systems. Importantly for computer science students, over half of the responses mention some aspect of the project helping students see connections between software (which can often be abstracted away from physical processes) and the physical world. For example, one respondent enjoyed the robot project because:

> *…the robot project [was] my first exposure to using my software and coding skill [with] a hardware tool that is actually moving. It taught me a lot about how hardware and software is connected and how I can use both to make something cool.*

Using code to move a physical object which the students had built was an impactful experience.

**Conclusion**
The implementation of a low-cost robotics project within an introductory ECE course successfully fostered direct engagement between computer science students and campus makerspaces. By moving away from "black-box" educational kits toward a fabrication-first approach, the curriculum provided students with the agency to transform generic components into personalized, functional artifacts. At the end of the course, a survey was administered to assess the course's impacts, revealing that this structured exposure not only increased students' technical confidence but also encouraged them to view makerspace skills as a valuable part of their professional identity.

The conclusions of this work are limited, however. Data is only available through January 2025, so it is not yet possible to see if this makerspace engagement is sustained or if it is a small 'spike' only associated with the course requirements. Further data collection and analysis is needed to look for longer term trends. Additionally, the small number of students means the data has a small sample size. This could be further confounded by external factors affecting students, such as work load in other courses that may affect work availability, or their own interest in working on personal projects. Lastly, the spike of engagement seen may only be caused by the course requirement itself, and not internal motivation. Likewise, survey respondents may be responding to what they perceive the faculty desire to hear (even though, students were told results would not be looked at until after grades were turned in so that responses would be anonymous and not influence grades).

Even with these limitations, the opportunity for computer science students to engage in hands on projects involving makerspaces was impactful for these students. It lowered the social barrier for engagement with makerspace communities on campus and also helped shift a perspective as to the interaction between hardware and software. Future work will look for continued usage of makerspaces by groups of students who engage in similar built-your-own-robot activities.

**Acknowledgements**
This material is based upon work supported by the National Science Foundation under Grant No. EEC-2022271. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## References

[1] I. Harel and S. Papert, *Constructionism: Research Reports and Essays, 1985-1990*. Ablex Publishing Corporation, 1991.
[2] E. R. Halverson and K. Sheridan, "The Maker Movement in Education," *Harvard Educational Review*, vol. 84, no. 4, pp. 495–504, 2014.
[3] W. W. Hanlon, "The Rise of the Engineer: Inventing the Professional Inventor During the Industrial Revolution," National Bureau of Economic Research, Cambridge, MA, Tech. Rep., Feb. 2022.
[4] M. E. Andrews, M. Borrego, and A. Boklage, "Self-efficacy and belonging: The impact of a university makerspace," *International Journal of STEM Education*, vol. 8, no. 1, p. 24, Mar. 2021.

[5] K. F. Osterman, “Students’ Need for Belonging in the School Community,” *Review of Educational Research*, vol. 70, no. 3, pp. 323–367, Sep. 2000, Publisher: American Educational Research Association.
[6] J. Davishahl and A. Boklage, “Board 331: Making Space in a Makerspace: Increasing Belonging through Social Engagement,” presented at the 2024 ASEE Annual Conference & Exposition, Jun. 23, 2024.
[7] M. Galaleldin, F. Bouchard, H. Anis, and C. Lague, “The Impact of Makerspaces on Engineering Education,” *Proceedings of the Canadian Engineering Education Association (CEEA)*, 2016.
[8] S. Vossoughi, P. K. Hooper, and M. Escudé, “Making Through the Lens of Culture and Power: Toward Transformative Visions for Educational Equity,” *Harvard Educational Review*, vol. 86, no. 2, pp. 206–232, 2016.
[9] N. Bartnik. “ORP - Design rules,” Accessed: Jan. 17, 2026. [Online]. Available: https://openroboticplatform.com/designrules.
[10] C. A. Berry, A. M. Larregola, K. Collins, and J. McGee, “Robotics for the streets: Open-source robotics for academics,” in *2023 IEEE Frontiers in Education Conference (FIE)*, 2023, pp. 1–5.
[11] P. Gaskell, J. Pavlasek, T. Gao, A. Narula, S. Lewis, and O. C. Jenkins, “Mbot: A modular ecosystem for scalable robotics education,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 18 294–18 300.
[12] J. Chen, A. Kolmos, and X. Du, “Forms of implementation and challenges of PBL in engineering education: A review of literature,” *European Journal of Engineering Education*, vol. 46, no. 1, pp. 90–115, Jan. 2021, Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/03043797.2020.1718615.
[13] O. Michel, “Webots: Professional mobile robot simulation,” *Journal of Advanced Robotics Systems*, vol. 1, no. 1, pp. 39–42, 2004.
[14] J. Chen, M. A. Hasan, X. Du, and A. Kolmos, “Exploring Students’ Perception of the Influence of PBL Elements on the Development of Engineering Identity,” *IEEE Transactions on Education*, vol. 66, no. 4, pp. 393–403, Aug. 2023.
[15] J. Chen et al., “An Exploration of Sources Fostering First-Year Engineering Students’ Academic Well-Being in a PBL Environment,” *IEEE Transactions on Education*, vol. 66, no. 5, pp. 421–430, Oct. 2023.