Teaching Human-Machine Teaming using Cyber-Physical Systems

Col. Christopher Lowrance, United States Military Academy

Colonel Christopher J. Lowrance is an Associate Professor and the Deputy Head of the Electrical Engineering and Computer Science Department at the United States Military Academy in West Point, NY.

Prof. Jean R.S. Blair, United States Military Academy
Devon Callahan, United States Military Academy

LTC Devon Callahan is an Assistant Professor in the Department of Electrical Engineering and Computer Science at the United States Military Academy, West Point. He teaches across the computer and cyber science curriculum, including Cyber Foundations (CY305), Programming Fundamentals (CY300), Network Engineering and Management (CY350), Introduction to Computing and Information Technology (IT105), Database Systems (CS393), and Advanced Networks (CS484), with a focus on software-defined networks.

He holds a Ph.D. in Computer Science from the University of Connecticut, a Master of Engineering in Computer Science from the University of South Carolina, and a Bachelor of Science in Computer Science from Methodist University. His research interests include artificial intelligence and machine learning, cybersecurity, network optimization, systems design, and integration and testing. His work emphasizes the application of network science to both computer and social networks to enhance system performance and resilience.

LTC Callahan began his military career as a Signal Soldier, enlisting with the 10th Mountain Division at Fort Drum, New York, and rising to the rank of Staff Sergeant. Commissioned through Officer Candidate School, he has served in multiple leadership positions across the Signal Corps. His previous assignments include Platoon Leader and Executive Officer in the 327th Signal Battalion at Fort Bragg, North Carolina; Division G6 staff officer and Company Commander, Bravo Company, 10th Sustainment Brigade, both with the 10th Mountain Division; and Signal Advisor to the Royal Saudi Air Defense Force under the U.S. Military Training Mission Task Force. He also commanded the Satellite Communications Station at Camp Roberts, California.

Lt. Col. Christa M Chewar, United States Military Academy
Michael Chiu, United States Military Academy
Brian Scott Petty, United States Military Academy
Edward Sobiesk, United States Military Academy

Teaching Human-Machine Teaming using Cyber-Physical Systems

Abstract

In the era of generative artificial intelligence (GenAI), students must develop foundational human–machine teaming (HMT) skills early in their academic careers in order to apply them effectively across subsequent coursework and professional contexts. This paper presents an instructional approach for developing such skills across all academic majors through project-based learning centered on microcontrollers and cyber-physical systems (CPSs). Prior to the advent of GenAI, the steep technical learning curve associated with microcontroller-based projects often deterred non-engineering students; however, GenAI has lowered this learning barrier, creating new opportunities for hands-on instruction in HMT principles. CPS-based projects proved effective for teaching HMT because they require deliberate problem solving and hardware–software integration, which in turn necessitate substantive collaboration with GenAI and reinforce key HMT tenets. Assessment results from the third iteration of a pilot course indicate increased confidence in experimenting with technology and practicing HMT skills; additionally, the results reveal comparable technical proficiency between STEM and non-STEM majors. Together, these results suggest that CPS-centered projects provide an effective and scalable pathway for introducing HMT concepts early in the curriculum.

1 Introduction

Generative artificial intelligence (GenAI) is rapidly transforming education and professional practice. As it becomes more widely adopted in the workplace, college graduates must be prepared to collaborate effectively with GenAI to enhance productivity, creativity, and problem-solving capabilities. Without such preparation, graduates risk being disadvantaged relative to peers who are more adept at human–machine collaboration. Recognizing this shift, many institutions of higher education have begun embracing the integration of AI into their curricula [1, 2].

A key challenge in this integration is ensuring that students develop foundational human-machine teaming (HMT) skills early in their academic careers. Establishing this foundation allows subsequent courses to reinforce and extend HMT concepts rather than introducing them for the first time. However, teaching HMT effectively requires problem contexts that are sufficiently complex to demand genuine and iterative collaboration between humans and GenAI, rather than tasks that can be easily solved through automated code or text generation with minimal human involvement. We have found that cyber-physical systems (CPSs), implemented through microcontroller-based projects, provide an effective context for developing these skills. This paper reports on our experiences teaching HMT principles and GenAI skills to students from nearly all academic majors using CPS-centered learning activities.

Prior to GenAI, microcontroller-based projects often presented formidable technical barriers for first-year students and non-STEM majors, as the simultaneous demands of low-level programming and circuit design created a steep learning curve. GenAI has substantially lowered these barriers by assisting with coding tasks and supporting interaction with Internet of Things (IoT) devices, sensors, and microcontrollers. As a result, students with little or no prior technical background can now participate meaningfully in the design and construction of functional CPSs through effective teaming with GenAI.

Although programming expertise is intentionally de-emphasized in this course, students still achieve a rich set of learning outcomes related to both problem solving and human–AI collaboration. The course employs a structured engineering problem-solving process as a guiding framework, requiring students to engage iteratively with GenAI at each stage, ranging from problem definition through prototype testing. Each stage of the problem-solving process presents distinct opportunities for collaboration with GenAI, helping students develop an understanding of not only how GenAI can be used, but also where its strengths and limitations lie.

Microcontroller-based CPS projects are particularly well suited to this instructional approach because they require the physical integration of hardware and software. This added complexity prevents students from relying solely on GenAI-generated outputs and instead requires them to assemble, test, and debug systems through hands-on experimentation and validation. These activities expose the limitations of GenAI firsthand: while models can generate code or design suggestions, they cannot perform wiring, debugging, or empirical testing on physical systems. This contrast reinforces a core principle of HMT, namely that effective teams leverage complementary strengths across human and machine contributors. In addition to this principle and the importance of validating automation, students develop key HMT competencies such as task allocation, multimodal interaction with GenAI (e.g., text, voice, images, and code), and the construction of effective prompts. Collectively, these skills position students for success in academic and professional environments where GenAI will be pervasive.

This paper presents both our pedagogical approach and assessment results from the third iteration of a pilot course at the United States Military Academy (USMA). Students report increased confidence in using GenAI productively, improved prompting strategies, greater awareness of the technology's strengths and limitations, and an expanded sense of what they can accomplish with computing technologies. Objective performance data further demonstrates that non-STEM majors achieved technical proficiency comparable to their STEM peers, validating the accessibility of this approach. Assessment results also indicate increased student interest in CPS-related technologies and positive perceptions of the value of CPS-based, hands-on problem-solving experiences.

We propose CPS-centered projects as a scalable model for developing foundational HMT skills across all majors in the GenAI era. Whether implemented in a general education course, a first-year engineering experience, or a short workshop, such activities equip students with HMT competencies needed for the future workforce. The result is graduates who are not only proficient users of GenAI, but also adaptive, creative, and effective problem solvers.

2 Related Works

Recent work in engineering and computing education has explored the integration of generative artificial intelligence (GenAI) into coursework, particularly as a support tool for programming, tutoring, and problem solving [3]. Studies report that tools such as ChatGPT can lower barriers to entry, support novice learners, and accelerate learning outcomes [4, 5]. A growing number of classroom implementations demonstrate the use of GenAI in computer programming, emphasizing the importance of effective prompting techniques [6] and reporting gains in computational thinking, programming self-efficacy, and student motivation [7]. While these efforts demonstrate the potential of GenAI to enhance learning, they often focus on isolated integrations within specific courses rather than on developing foundational skills for effective human–AI collaboration early in a student's academic journey.

The concept of human–machine teaming (HMT) provides a useful theoretical lens for understanding how humans and intelligent systems can work together effectively. A core principle of HMT is the intentional allocation of roles that leverage the complementary strengths of humans and machines while mitigating their respective limitations [8]. Prior research also highlights the importance of trust and tolerance for both ambiguity and risk in effective teaming [9, 10]. In parallel, work in human–automation interaction underscores the continued role of humans in supervision, validation, and decision-making as autonomy increases [11]. For example, although GenAI can significantly accelerate software development tasks, machine-generated code may contain errors or vulnerabilities, reinforcing the need for human testing and validation [12]. Despite the maturity of these principles, their systematic application within undergraduate engineering curricula in the GenAI era remains limited.

Cyber-physical systems (CPSs) and microcontroller-based project learning have long been used to support hands-on, integrative learning in engineering education [13, 14]. Prior studies show that CPS projects can enhance student engagement, systems thinking, and understanding of hardware–software integration, including for first-year and non-electrical engineering students [13]. However, much of this work predates the widespread availability of GenAI. More recent studies demonstrate that tools such as ChatGPT can effectively support microcontroller programming for students with limited prior experience, increasing accessibility to embedded computing systems across disciplines [15].

Building on these principles of role allocation and human oversight, our approach deviates from traditional curricula by prioritizing intent-based design over rote syntax. Programming is not merely outsourced to the machine; rather, it is performed by the human through natural language, with GenAI serving as a teammate that interprets this intent and translates it into functional code. This approach abstracts low-level syntax to support a general education audience while maintaining the student's role as the primary creative and technical driver. In this environment, the human provides the conceptual logic and system architecture, while the machine facilitates the realization of that vision. Consequently, HMT principles are operationalized through a continuous loop of prompt engineering, code synthesis, and iterative testing. This reinforces foundational HMT skills, such as strategic task allocation, iterative refinement through human–machine interaction, and rigorous human oversight as students navigate the design, construction, and validation of CPS projects.

3 Curriculum Design for Teaching HMT Principles

3.1 Overall Course Construct

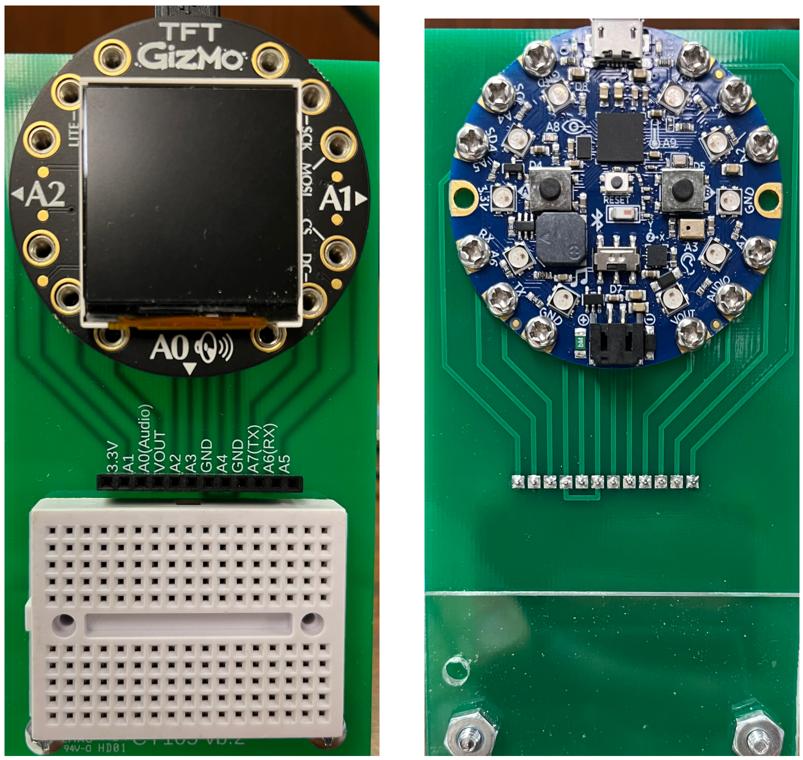
The course discussed in this paper is a required general education course taken by a majority of academic majors at USMA, with select engineering programs exempted. These exempted majors satisfy the general education requirement through alternative coursework within their degree programs. The course is designed to achieve multiple institutional learning outcomes beyond electronics and cyber-physical systems (CPSs). The broader structure and thematic goals of this predominantly technology-focused course, including its emphasis on project-based learning, building self-efficacy, and integrating emerging technologies, have previously been described in [16]. The present work focuses on a distinct instructional dimension of the course: the use of GenAI within a microcontroller-based CPS instructional sequence to develop foundational human–machine teaming (HMT) skills. Collaboration with GenAI is allowed and encouraged across a series of mini-projects, and this paper examines how the electronics and microcontroller-centric portions of the curriculum, in particular, provide a uniquely effective context for practicing HMT.

Throughout the course, full collaboration with GenAI is authorized and encouraged on all homework assignments and projects. However, to isolate and reinforce the development of HMT skills, collaboration between students was restricted. This constraint was intentionally imposed to force students to rely on GenAI as their primary teammate for technical assistance and debugging, rather than defaulting to peer-to-peer support. Students are required to include a brief documentation statement with each submission that identifies the GenAI platform(s) used and provides links to any associated chat sessions. This practice allows instructors to examine how students are engaging with GenAI and to better understand their teaming strategies during the design and implementation of CPSs.

The CPS instructional block spans seven 75-minute in-class lessons. The first lesson introduces microcontrollers through a case study highlighting real-world innovations, emphasizing bottom-up design approaches to motivate student interest. Subsequent lessons cover the fundamentals of direct-current (DC) circuits and rapid prototyping using breadboards. Instruction on basic circuit elements is limited to resistors, light-emitting diodes, transistors used as electrical switches, and common sensors, which are analyzed using foundational models such as Ohm's Law and Kirchhoff's Voltage Law. In addition, students are introduced to finite state machine (FSM) diagrams as a formal method for specifying system behavior prior to implementation.

Outside of class, students complete three mini-projects as part of the CPS instructional block. All projects are deliberately designed to require physical wiring of external circuitry connected to a microcontroller development board. This constraint ensures that the projects are not purely software-based, preventing students from relying solely on GenAI-generated solutions and instead requiring meaningful hands-on engagement and validation.

Students are issued an Adafruit Bluefruit microcontroller with a TFT Gizmo display, as shown in Figure 1. The microcontroller and display are mounted on a custom printed circuit board (PCB) that breaks out selected input and output pins and includes space for a miniature breadboard. This configuration enables students to easily connect external circuitry to the microcontroller, facilitating the direct physical interfacing required to add components and extend the functionality of their CPSs. Although the development board includes several onboard sensors and light-emitting diodes, the added breadboard and breakout pins provided by the custom PCB play a critical role in supporting hands-on construction and rapid prototyping. For software development, students program the device using CircuitPython and the Mu integrated development environment.



(a) Front View (b) Back View

Figure 1: Front and back views of the custom circuit board with integrated microcontroller, display, and breadboard used for rapid CPS prototyping.

3.2 Design and Execution of the Mini-Projects

To facilitate process-based learning [17] and reinforce an engineering problem-solving approach, students are guided to use a structured problem-solving process (PSP). Because each stage of the PSP involves distinct cognitive and technical activities, the process naturally supports different modes of collaboration with GenAI, enabling instruction in foundational HMT principles. While multiple engineering problem-solving frameworks exist, this course adopts a four-stage model consisting of *Define, Design, Create, and Validate*. Table 1 summarizes some representative student activities at each stage, the corresponding roles assumed by GenAI, and the HMT skills emphasized.

In initial iterations of the course, the *Define* stage was largely given by written instructions. Observations revealed that providing written assignment instructions inadvertently encouraged

Table 1: Alignment of the Problem Solving Process (PSP) with GenAI Roles and HMT Skills

PSP Stage	Typical Student Activity	Role of GenAI	HMT Skill Emphasized
Define	Reframe and analyze problem statement	Review problem statement, stakeholder requirements – offer feedback	Collaborative sensemaking
	Identify requirements, constraints, inputs & outputs	Suggest feedback, potential gaps, and challenges	Critical evaluation of GenAI feedback
Design	Brainstorm design approaches	List strengths and weaknesses of each approach	Creative ideation with GenAI support
	Decompose problem into subsystems; sketch state machine diagram and design circuits	Review and offer feedback on diagram	Task decomposition and abstraction
Create	Wire sensors and electrical components	Review wiring and respond to targeted questions	Embodied execution informed by GenAI
	Prompt GenAI for software code	Provide software code	Effective prompting and iterative refinement
Validate	Develop test cases	Review test plan and offer feedback	Verification planning and completeness checking
	Test each case and verify functionality	Offer troubleshooting advice if problems encountered	Complementary troubleshooting and diagnosis

superficial engagement during this critical phase. Students frequently pasted assignment text directly into GenAI tools without supplying necessary context that was implied but not explicitly stated in the prompt, such as the specific microcontroller model (Adafruit Bluefruit), the display (TFT Gizmo), the required programming language (CircuitPython), or the integration of specific discrete components. Lacking these constraints, GenAI models often generated code using incompatible libraries or incorrect languages (e.g., C instead of Python), leading students into 'rabbit holes' of debugging code that was fundamentally unsuited for the hardware at hand.

To address this 'context gap' and promote more deliberate engagement during the *Define* stage, future iterations of the course will modify how assignment instructions are delivered. Rather than providing written project specifications, students will receive the intent of the system design verbally. This approach requires students to actively synthesize requirements and formulate their own prompts, rather than relying on verbatim transcription of assignment text into GenAI tools. To further support this process, a requirements checklist will be provided to prompt students to explicitly define system constraints and assumptions, including the hardware platform, relevant libraries, and known inputs and outputs, prior to engaging the AI. These instructional scaffolds are intentionally limited in scope, supporting problem framing without prescribing solution strategies. Together, they ensure that the student, as the human teammate, practices their critical role in defining the problem effectively, while reducing frustration associated with under-specified prompts and misaligned machine-generated solutions.

The *Design* stage includes two required submissions that students must complete before proceeding to implementation: a finite state machine (FSM) diagram and a completed circuit wiring diagram. While the FSM requires students to specify the system's behavior, the wiring diagram ensures that they understand the physical connections between the microcontroller, sensors, and actuators. Because of the limited coverage of electronics in the course, students are provided with a partially completed schematic and are expected to identify and sketch the additional connections required to make the system functional. These checkpoints ensure that students have developed a sound technical blueprint that addresses stakeholder requirements and prepares them for the physical wiring and software creation phase. When these artifacts are introduced, an in-class activity is used to demonstrate the multiple modalities through which students and instructors can collaborate with GenAI in real time. These interactions include sharing photographs of hand-drawn FSM sketches, screenshots of completed wiring diagrams, and engaging in live, conversational exchanges using GenAI-enabled mobile applications that access device microphones and cameras. Through this activity, students experience how GenAI can interpret visual artifacts, identify potential wiring errors or logic gaps, and simultaneously respond to spoken questions. This enables immediate feedback on design decisions and provides GenAI with additional context to support subsequent code development.

While current GenAI tools are effective at interpreting FSMs and wiring diagrams to provide feedback, they often struggle to generate these visual artifacts from scratch in a way that meets all required specifications. In particular, GenAI-produced diagrams frequently omit important details such as labeled transition arrows, clearly defined state actions, or correct pin-to-component mappings. As a result, this stage requires substantial human involvement, attention to detail, and critical evaluation of GenAI outputs. In doing so, the *Design* stage of the problem-solving process reinforces key HMT principles, including leveraging complementary strengths and validating automated outputs against a human-defined technical requirement.

The *Create* stage of the problem-solving process primarily involves wiring the necessary circuitry and collaborating with GenAI to develop software code. This stage provides multiple opportunities to reinforce HMT principles and to practice effective GenAI teaming techniques, including structured prompting strategies. In particular, students learn the importance of providing sufficient contextual information to GenAI and develop an understanding of context windows within individual chat sessions. Through this experience, students quickly recognize the need to specify details such as the microcontroller model, display type, programming language, and compatible libraries in order to obtain useful results. Without this context, GenAI frequently generates code that relies on incompatible libraries or unsupported features.

To address this challenge, students are introduced to specific prompting practices, including the use of 'warm-up' prompts to prime the model with relevant context at the beginning of a new chat session. Students are also encouraged to specify known-compatible libraries, such as those provided by Adafruit for the Bluefruit platform, to obtain more reliable results. Together, these practices reinforce the importance of high-quality prompting for effective collaboration with GenAI.

Students also learn that effective collaboration with GenAI typically requires iterative interaction rather than one-shot prompting. While a well-crafted prompt may occasionally yield functional code in a single attempt, this outcome is not typical. Instead, students are encouraged to implement and test individual subsystems before incrementally integrating them into a complete solution. This approach requires students to break down complex requirements into manageable tasks and to engage GenAI on narrowly scoped subtasks. By constraining each interaction in this way, students obtain more focused and precise responses that are easier to test and verify. In turn, this reinforces a core human–machine teaming principle: the human's responsibility for incremental development, continuous testing, and the validation of automated outputs throughout the integration process.

Collaboration with GenAI during the *Create* stage extends beyond software development. Students must also physically assemble and debug their circuits. Instructors demonstrate how GenAI mobile applications can be granted access to device cameras and microphones, enabling GenAI to inspect physical wiring and respond to spoken questions in real time. This multimodal interaction allows GenAI to provide targeted guidance when students encounter hardware-related issues and reinforces the idea that effective HMT can involve multiple modes of interaction.

The final stage of the problem-solving process requires students to *Validate* their CPSs. Students revisit their finite state machine diagrams to confirm the expected behavior for each system state and associated actions. They are required to submit a concise video demonstrating system functionality, including correct state transitions and appropriate responses to valid inputs. In addition, students must demonstrate system robustness by showing that invalid or unexpected inputs do not result in unintended behaviors, thereby reinforcing the importance of systematic testing and validation of GenAI-generated code. This phase is especially important in the GenAI era, as verification and validation remain essential human oversight responsibilities in effective human–AI teaming.

Notably, this validation process serves as a mechanism for calibrating student trust in the technology. As students encounter discrepancies, such as unexpected state actions or missing transitions, they learn that GenAI outputs are not infallible. These errors, whether resulting from insufficient prompting or the model's inherent limitations, underscore the necessity of human oversight. Through methodical testing, students internalize a core HMT principle: that while GenAI can be a powerful partner, it cannot be blindly trusted, and its contributions must be rigorously verified against physical reality.

3.3 Examples of Mini-Projects

A wide range of CPS projects can be used to support instruction in HMT with GenAI. A key design criterion for the projects used in this course is the requirement that students wire external circuit components rather than rely solely on sensors and outputs integrated on the microcontroller development board. This constraint ensures that students must engage in hands-on circuit construction and testing, rather than simply uploading code and verifying system behavior through software alone. By requiring the integration of external sensors, light-emitting diodes, resistors, and transistors, the projects increase system complexity in ways that necessitate greater human involvement through physical circuit wiring while reinforcing foundational electronics concepts.

Students completed three CPS-focused mini-projects during this instructional block; representative examples are shown in Figures 2–4. In the first project (Figure 2), students designed, implemented, and demonstrated a notional temperature-monitoring system, beginning with the development of a finite state machine (FSM) diagram to guide system behavior. An external potentiometer was used to vary the simulated temperature input, which was read by the microcontroller through analog-to-digital conversion. The system consisted of three states: safe, warning, and danger. Each state triggered specific actions, such as illuminating LEDs of different colors and activating the onboard buzzer. Students implemented this design on the microcontroller and demonstrated correct operation across all states and transitions. This project provided practice in basic breadboarding techniques and introduced key concepts related to potentiometer operation and analog sensing. As with all projects in the sequence, students collaborated with GenAI to generate and refine software code and were required to validate system behavior using a structured test plan that exercised all possible states and transitions.

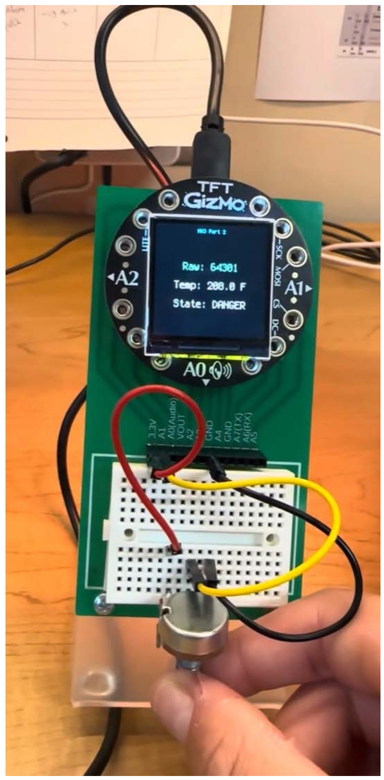


Figure 2: Student-built CPS intended to model as a temperature-monitoring system, where an external potentiometer was used to vary the system's input state.

The second mini-project (Figure 3) focused on signal detection and transistor-controlled actuation. In this assignment, students configured the microcontroller to detect a trigger-event signal from a push-to-talk handheld radio. Upon detection, the microcontroller drove an output pin high, biasing an external bipolar junction transistor and allowing current to flow through an LED with a current-limiting resistor. To increase design complexity, the FSM included an *inactive* state in which detected signals were ignored when an onboard slide switch was set to the disabled position. This project deliberately required students to make multiple external breadboard connections, reinforcing basic electronics concepts and emphasizing hands-on circuit construction. Students interfaced the radio's headphone output with the microcontroller to sense voltage changes and correctly wired and biased the transistor for reliable operation. Despite introducing new circuit elements, the assignment remained accessible to students from all academic majors and was typically completed within one week as an out-of-class activity when paired with supporting in-class instruction.

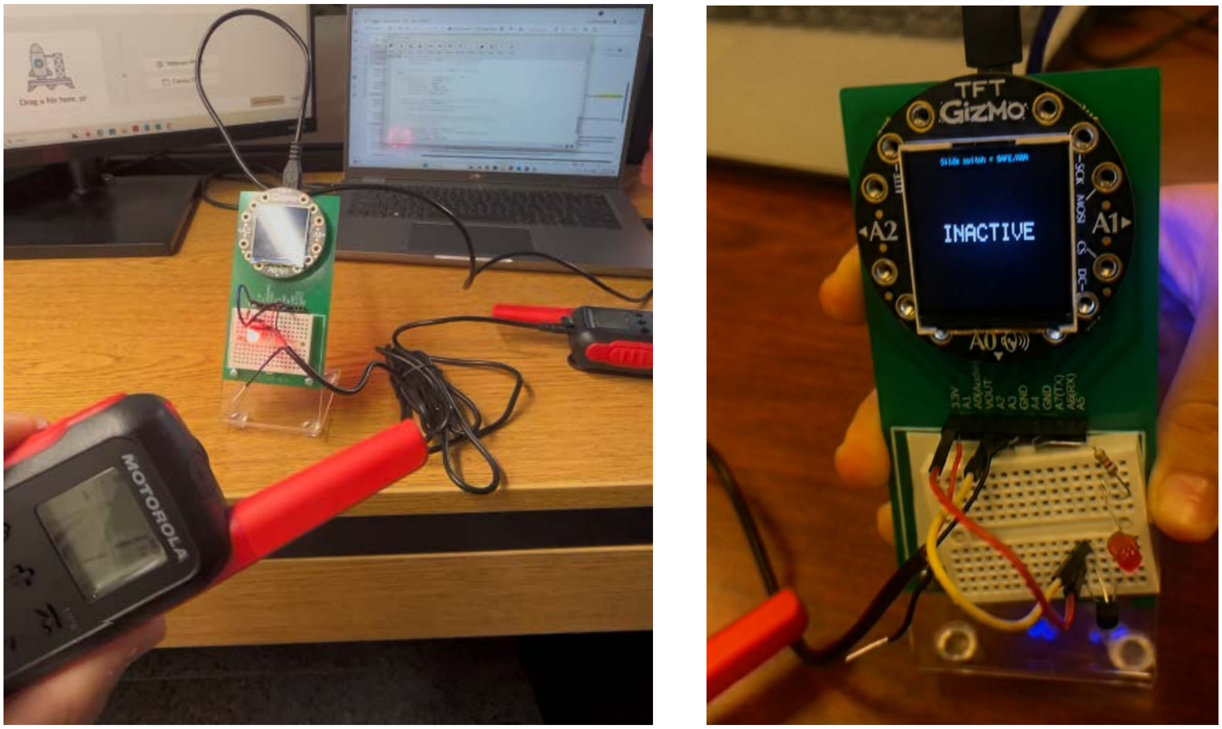


Figure 3: Student-built CPS demonstrating the wireless activation of an LED from a push-to-talk radio.

The third and final mini-project (Figure 4) was designed to expose students to the broader possibilities of Internet of Things (IoT) systems and wireless communication. In this project, students connected their smartphones to the microcontroller using the Adafruit Connect mobile application over Bluetooth. When the system was placed in an appropriate state using an onboard slide switch, the microcontroller responded to commands received via Bluetooth by actuating an external servo motor. The primary external hardware requirement for this project was the correct wiring of a servo motor. This assignment emphasized system integration, wireless communication, and state-dependent actuation while reinforcing the principles of effective HMT through GenAI-supported development and validation. Collectively, these mini-projects provided repeated opportunities for students to learn fundamental principles of HMT and practice GenAI techniques through each stage of the PSP.

While the microcontroller-centric portions of the curriculum provide a tangible context for HMT, they represent only one phase of a broader instructional sequence integrated throughout the course. Other thematic blocks operationalize HMT principles within diverse technological domains to reinforce the broad applicability of effective human–AI collaboration. For instance, during the study of network protocols, students utilize tools like Wireshark to perform packet captures and subsequently team with GenAI to analyze the data, generating protocol distribution charts and identifying network behavior patterns. Similarly, a separate module challenges

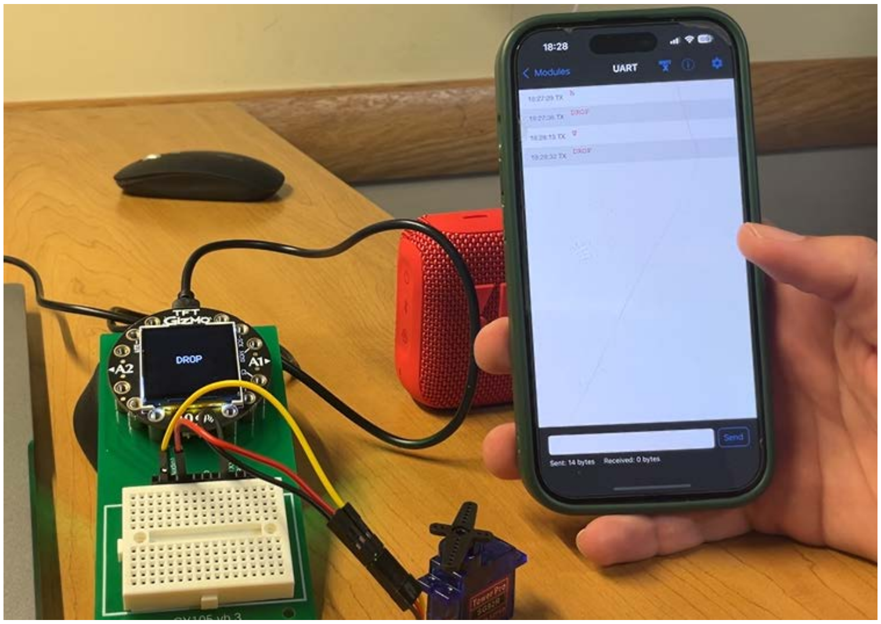


Figure 4: Student-built CPS demonstrating Bluetooth activation of servo motor.

students to leverage GenAI for creative brainstorming and the production of influential short-form video messages. By situating HMT within these varied contexts, the course ensures that students develop a robust, generalizable fluency in directing AI as a teammate.

4 Student Experience

To examine student perceptions of the CPS instructional block, an optional and anonymous post-instructional survey was administered. During the Fall 2025 semester, the course enrolled a total of 329 students, of whom 197 elected to complete the survey. The respondents represented a diverse range of academic backgrounds, reflecting the course's role as a general education requirement; approximately 42.6% of enrolled students were pursuing degrees in STEM disciplines, while 57.4% were majoring in the humanities or social sciences. Survey items used a five-point Likert scale ranging from 1 = *Strongly Disagree* to 5 = *Strongly Agree*. The survey was designed to capture student perceptions related to key learning outcomes associated with HMT and GenAI fluency. Figure 5 presents a 100% stacked bar chart summarizing student responses to these items. The full text of the survey statements corresponding to the abbreviated labels shown in the figure is provided in Table 2.

To further examine student responses, descriptive summary statistics were computed and are reported in Table 3. Across all survey items, mean scores were near or above 4.0 on a five-point Likert scale, indicating generally positive student perceptions. Particularly strong agreement was observed for items related to GenAI awareness, effective prompting strategies, and confidence in using GenAI productively. Standard deviations were relatively low across items, suggesting a high degree of consistency in student responses. While interest in furthering their study of CPSs showed slightly lower mean scores than GenAI-related items, responses remained above neutral, indicating broad acceptance of CPS-based projects across academic majors.

Table 2: Mapping of abbreviated labels used in Figure 5 to the full Likert-scale survey item statements.

Abbreviated Label	Full Statement
Learning Value	This instructional block was a valuable learning experience
Tech Confidence	This instructional block increased my confidence in exploring and experimenting with technology
CPS Interest	This instructional block inspired me to want to learn more about microcontrollers
GenAI Partnering	I used generative AI during this instructional block, and it inspired me to want to partner more with generative AI
GenAI Awareness	This instructional block helped me discover the strengths and weaknesses of generative AI
Prompting Skills	This instructional block taught me the importance of constructing effective prompts to get useful, accurate results from generative AI
Barrier to Entry	After this instructional block, I feel the barrier to entry to adapt and innovate with technology is lower than I anticipated
GenAI Coding Confidence	I feel more confident that I could use generative AI to create effective computer code and accomplish more complex tasks

Table 3: Descriptive statistics for Likert-scale survey items assessing student perceptions of the CPS instructional block and GenAI-supported learning.

Survey Item (Abbreviated)	N	Mean	SD	% Agree / Strongly Agree
Learning Value	197	3.97	0.81	78.68%
Technology Confidence	197	4.00	0.79	79.19%
CPS Interest	197	3.68	1.05	62.44%
GenAI Partnering	197	4.13	0.86	81.73%
GenAI Awareness	197	4.25	0.80	85.28%
Prompting Skills	197	4.26	0.81	85.28%
Barrier to Entry	197	3.94	0.89	76.65%
GenAI Coding Confidence	197	4.20	0.83	85.28%

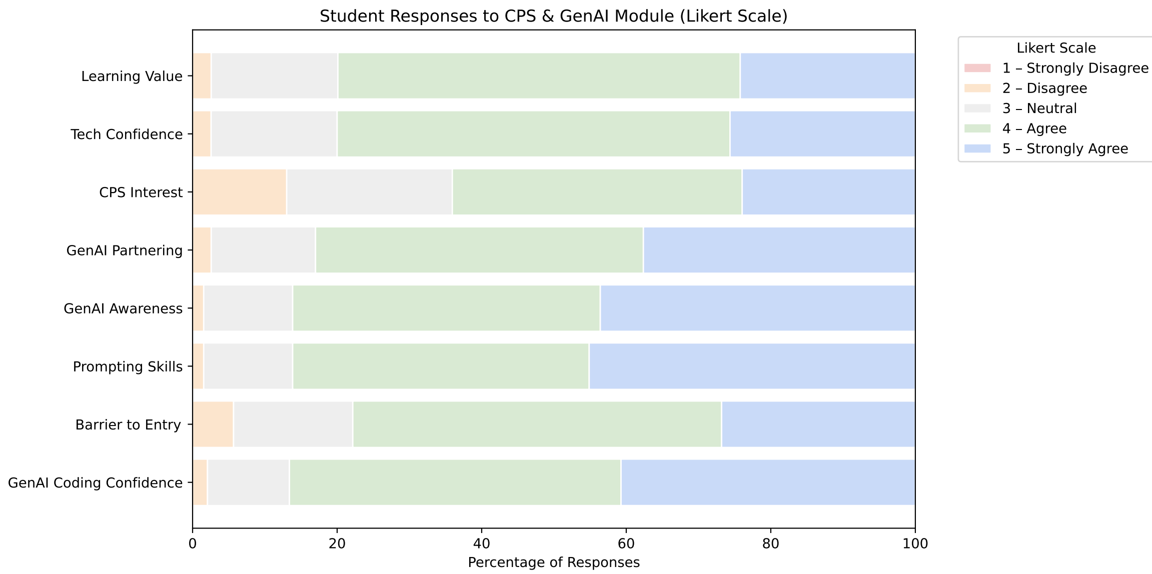


Figure 5: 100% stacked horizontal bar chart showing student responses to Likert-scale survey items assessing perceptions of the CPS instructional block and GenAI-supported learning.

To complement these self-reported perceptions, we also analyzed objective performance data to examine the accessibility of the technical content for non-STEM majors. A comparison of CPS mini-project grades revealed comparable achievement across academic backgrounds, with STEM majors averaging 90.7% and non-STEM majors averaging 91.3% across the instructional block. Performance variability was also similar between groups, with overlapping standard deviations observed across all assignments. Detailed summary statistics for each mini-project, including assignment-level means and standard deviations by academic background, are provided in Table 4. This parity in outcomes suggests that the GenAI-supported workflow effectively mitigated technical barriers often associated with microcontroller-based coursework, enabling non-STEM students to achieve proficiency levels comparable to their STEM peers.

Table 4: Mean CPS Mini-Project Scores (Percent) by Academic Background

Assignment (Points)	**STEM**		**Non-STEM**	
	Mean (%)	**SD (%)**	**Mean (%)**	**SD (%)**
Temperature Warning System (35)	90.41	19.75	90.91	13.84
Push-to-Talk Radio Signal Detection (60)	91.02	18.57	90.95	15.05
Bluetooth-Controlled Servo (60)	90.73	14.50	91.94	8.82
Total (155)	**90.69**	**11.70**	**91.32**	**8.87**

5 Conclusion and Future Work

This paper presented a project-based approach for developing foundational human–machine teaming (HMT) skills using cyber-physical systems (CPSs), with the aim of preparing graduates for the GenAI era. By structuring the curriculum around a systematic engineering problem-solving process, the course offers a critical broadening experience for students of all academic majors, introducing them to real-world engineering methodologies. The curriculum teaches HMT by emphasizing the incremental development and testing of solutions, positioning GenAI as a collaborative teammate throughout the entire problem-solving process. The physical nature of microcontroller-based projects required students to integrate hardware and software while preventing overreliance on GenAI. This constraint reinforced core HMT principles, specifically the need to leverage complementary strengths and validate machine-generated outputs against physical reality.

Assessment results from the third iteration of the course provide strong evidence for the accessibility of this approach. In addition to positive self-reported perceptions, objective performance data revealed comparable, and in some cases, slightly higher achievement by non-STEM majors relative to their STEM peers on CPS mini-projects. This outcome suggests that GenAI-supported human–machine teaming can effectively lower traditional barriers to entry associated with microcontroller-based coursework. By leveraging this approach, students without prior technical backgrounds achieved proficiency levels on par with, or exceeding, those of their STEM peers. Survey responses further indicate increased GenAI awareness, confidence in effective prompting strategies, and positive perceptions of the value of CPS-based, hands-on learning across academic backgrounds.

Several opportunities for future work remain. Planned iterations of the course will more explicitly scaffold the *Define* stage of the problem-solving process to promote deeper engagement with problem framing and requirement generation, particularly through open-ended and verbally delivered project prompts. Additional work will refine assessment instruments to more directly measure specific HMT competencies, such as incremental development and validation practices, beyond self-reported perceptions. Longitudinal studies examining how students transfer these skills to subsequent courses or professional experiences would further strengthen understanding of the lasting impact of this approach.

More broadly, this work demonstrates the potential of CPS-based, GenAI-supported projects as a scalable and inclusive model for developing HMT skills and GenAI fluency across the curriculum. By emphasizing effective human–machine collaboration rather than narrow technical specialization, this approach offers a pathway for preparing students from all majors for future academic and professional contexts in which GenAI will play an increasingly central role.

6 Acknowledgments

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the U.S. Military Academy, the Department of the Army, the Department of War, or the U.S. Government.

References

[1] Office of Academic Affairs. *AI Fluency*. The Ohio State University, Accessed: Jan. 4, 2026. 2026. URL: https://oaa.osu.edu/ai-fluency.

[2] U.S. Military Academy West Point. *Advancing Military Leadership in the Age of Artificial Intelligence*. Accessed: Jan. 4, 2026. URL: https://www.westpoint.edu/about/modernization-plan/artificial-intelligence-military-leadership.

[3] E. Kasneci et al. "ChatGPT for Good? On Opportunities and Challenges of Large Language Models for Education". In: *Learning and Individual Differences* 103 (Apr. 2023), p. 102274. DOI: 10.1016/j.lindif.2023.102274.

[4] I. McCulloh et al. "Generative AI in Computer Science Education: Accelerating Python Learning with ChatGPT". In: *arXiv preprint arXiv:2505.20329* (May 2025). DOI: 10.48550/arXiv.2505.20329.

[5] P. Manorat et al. "Artificial Intelligence in Computer Programming Education: A Systematic Literature Review". In: *Computers and Education: Artificial Intelligence* 8 (June 2025), p. 100403. DOI: 10.1016/j.caeai.2025.100403.

[6] P. Denny et al. "Prompt Problems: A New Programming Exercise for the Generative AI Era". In: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education*. Vol. 1. Portland, OR, USA, 2024, pp. 296–302. DOI: 10.1145/3626252.3630909.

[7] R. Yilmaz and F. G. K. Yilmaz. "The Effect of Generative Artificial Intelligence (AI)-Based Tool Use on Students' Computational Thinking Skills, Programming Self-Efficacy and Motivation". In: *Computers and Education: Artificial Intelligence* 4 (Jan. 2023), p. 100147. DOI: 10.1016/j.caeai.2023.100147.

[8] F. Bocklisch and N. Huchler. "Humans and Cyber-Physical Systems as Teammates? Characteristics and Applicability of the Human-Machine-Teaming Concept in Intelligent Manufacturing". In: *Frontiers in Artificial Intelligence* 6 (Nov. 2023), p. 1247755. DOI: 10.3389/frai.2023.1247755.

[9] I. Rauf et al. "Human-Machine Teaming and Team Effectiveness in AI Tools for Software Engineering". In: *2025 IEEE/ACM 18th International Conference on Cooperative and Human Aspects of Software Engineering (CHASE)*. 2025, pp. 75–80. DOI: 10.1109/CHASE66643.2025.00017.

[10] C. Pfaff et al. *Trusting AI: Integrating Artificial Intelligence into the Army's Professional Expert Knowledge*. Monograph. U.S. Army War College Press, Feb. 2023. URL: https://press.armywarcollege.edu/monographs/959.

[11] M. R. Endsley. "From Here to Autonomy: Lessons Learned From Human–Automation Research". In: *Human Factors* 59.1 (Feb. 2017), pp. 5–27. DOI: 10.1177/0018720816681350.

[12] J. Vaidya and H. Asif. "A Critical Look at AI-Generated Software: Coding with the New AI Tools Is Both Irresistible and Dangerous". In: *IEEE Spectrum* 60.7 (July 2023), pp. 34–39. DOI: 10.1109/MSPEC.2023.10177044.

[13] U. Idem, S. Purzer, and J. Morphew. "How Microelectronics and Microcontrollers Are Integrated into First-Year and Sophomores Engineering Programs". In: *2025 ASEE Annual Conference & Exposition*. Accessed: Dec. 27, 2025. Montreal, Quebec, Canada, 2025.

URL: https://nemo.asee.org/public/conferences/365/papers/46517/view.
[14] C. J. Lowrance and J. R. Rogers. "Stimulating Higher Order Thinking in Mechatronics by Comparing PID and Fuzzy Control". In: *Computers in Education Journal* 10.3 (Sept. 2019). arXiv preprint arXiv:2601.08865, pp. 1–9. arXiv: 2601.08865 [cs.CY].
[15] D. Johnson et al. "Using Generative AI for Microcontroller Programming: Effects on Experienced and Inexperienced Programmers". In: *Journal of Agricultural Systems, Technology, and Management* 36.1 (Dec. 2025), pp. 27–41. URL: https://jastm.org/index.php/jastm/article/view/16155.
[16] C. Lowrance et al. "Harnessing Case Studies and Hands-on Learning to Empower Non-Engineers to Excel in a Tech-Driven World". In: *2025 ASEE Annual Conference & Exposition*. Montreal, Quebec, Canada, 2025, p. 56682. DOI: 10.18260/1-2--56682.
[17] Center for Advancing Teaching and Learning Through Research. *Process-focused Learning and Assessment for AI and Learning Skills*. Accessed: 2026-01-10. Northeastern University, 2025. URL: https://learning.northeastern.edu/aitips-process-focused-learning/.