Students' Self-Perception of C++ and Arduino Programming in Engineering and Agriculture: A Case Study

[1]Akinode John Lekan ,[2]Timothy Mathew Olufemi
Computer Science Department ,Federal Polytechnic Ilaro
Electrical Engineering Department,Federal Polytechnic Ilaro

ABSTRACT

This study examines students' self-perceptions of programming proficiency in C++ and Arduino within engineering and agriculture education. In the context of increasing demands for automation and smart farming solutions, the integration of hands-on, project-based learning, has become essential to bridge the gap between abstract programming theory and practical application . Utilizing a mixed-methods approach with pretest-posttest surveys and semi-structured interviews among 82 HND I students from Mechatronics and Agricultural Technology departments, the research assesses changes in attitudes, technical competence, and self-efficacy in both text-based C++ coding and microcontroller programming environments .

Quantitative findings indicate that students, particularly those from Mechatronics, reported higher initial self-assessed proficiency in tasks such as writing syntactically correct code, sensor interfacing, and circuit assembly. Nevertheless, even students in Agricultural Technology improved notably in confidence and overall technical skills following exposure to Water level monitoring sensor-based projects . Qualitative analyses reveal that the tangible outcomes of Arduino projects help demystify abstract concepts, enhance practical problem-solving, and stimulate interdisciplinary appreciation for programming's role in real-world applications.These insights underscore the importance of tailored, experiential instructional strategies to bolster programming self-efficacy and advance curricular reforms in both engineering and agricultural settings.

Keywords: C++ Programming,Arduino Microcontroller,,Mechatronics.Agricultural Technology, Self-Assessed Proficiency

INTRODUCTION

Globally, the recent surge in reliance on automation,smart farming techniques and Internet of Things (IoT), has aroused a critical need for advanced programming skills across technical disciplines.Furthermore ,in engineering and agriculture, the capacity to develop innovative solutions is directly Linked with proficiency in programming languages such as C++ and expertise in open-source microcontroller platforms such as Arduino. These programming skills are important not only for programming automation systems and sensor networks but also for enhancing real-time data acquisition, control algorithms, and easy integration of hardware and software. Engineering disciplines use these skills to improve embedded systems design, while agricultural applications, for example in precision agriculture and greenhouse monitoring, benefit from automated irrigation, data analytics, and improved resource management (Chondrogiannis et al., 2021; He et al., 2016).

Programming in these disciplines has gone past theories and has now evolved into a transformative ability supporting computational thinking and digital literacy. While engineering education has long been imparting programming as a part of its academic programs in control systems and hardware design, the dynamic shift in Industry 4.0 and Agriculture 4.0 requires a similarly solid incorporation into agricultural education. Conventional agricultural technical training has to now endorse a programmatic inclusion in automation, sensor interfaces, and IoT applications. Early exposure to the same has the ability to narrow the gap between theory and practice and equip the upcoming professionals to design and deploy smart farm solutions (Chondrogiannis et al., 2021; El-Abd, 2017).

This article discuss students' self-perception in terms of C++ and Arduino programming in the context of engineering and agricultural sciences, with emphasis on the relationship between developing technical skills and self-efficacy. Education on programming has become a core part of contemporary STEM educational programs as computational proficiency underpins the design, evaluation, and execution of engineering and agricultural systems. In engineering, C++ is among the most essential programming languages used to construct simulation models, embedded systems, and real-time control software used in applications from robotics to energy systems. In agriculture, the same skills are also used to a larger extent in precision farming, automated systems and sensor-based monitoring in which microcontroller systems like Arduino facilitate pratical learning and quick prototyping (Arigye et al., 2023; El-Abd, 2017).Despite the well-documented benefits of computational tools in these fields, a gap remains in the literature regarding how students' self-perception, specifically related to programming using C++ and Arduino, influences academic success and persistence. Previous studies have predominantly focused on biomedical and agricultural engineering students' programming self-beliefs in computational modelling contexts without a detailed comparative analysis of the self-perception differences in C++ and Arduino programming (Arigye et al., 2023; El-Abd, 2017).

The integration of Arduino in courses is viewed as an effort to concretize abstract programming concepts by linking them to physical computing applications, thereby addressing some of the challenges associated with traditional C++ programming instruction (Arslan & Tanel, 2020; Graven & Bjork, 2016). Despite the potential for such hands-on applications to enhance learning outcomes, multiple studies note that although student attitudes towards programming improve with active learning interventions, their self-efficacy in performing complex programming tasks often remains modest (Arslan & Tanel, 2020; Erol, 2020).

Inculcating microcntroller-based learning platforms like arduino within technical curricula has emerged as a dynamic approach to transform abstract programming knowledge into tangible, hands-on experiences. This progressive shift is evident in numerous studies that highlight improvements in student attitudes, self-efficacy, and technical competence when exposed to project-based learning using accessible platforms like Arduino alongside text-based languages such as C++ (Arslan & Tanel, 2020). In educational settings spanning engineering and agriculture, the challenge is to evaluate how these innovative educational interventions affect students' self-perceptions of their programming skills. With a sample of 83 students at Federal Polytechnic Ilaro, the current case study is designed to assess changes before and after curricular exposure while determining disciplinary differences and addressing the inherent challenges in integrating programming tools into technical education (Basha & Balaji, 2025; Arslan & Tanel, 2020).

This research study offers a close examination of students' self-perceptions, a comparison of the levels of skills across different programs, and a consideration of underlying causes drawn from empirical research and classroom intervention. The results shown in this research are essential to inform curricular changes to close the educational gap between the different educational environments of Agricultural Technology and Mechatronics. The article 's purpose is to present a close case study comparing self-perceptions between engineering and agricultural programs, exploring the determinants of such perceptions and providing recommendations to support pedagogical approaches in interdisciplines.

OBJECTIVES

i. To measure the change in students' attitudes and confidence levels towards programming as a direct result of hands-on Arduino activities.

ii. To explore contextual factors such as prior exposure to electronics, the relevance of programming in each field's curricula, and the perceived applicability of C++ in solving discipline-specific challenges

iii. To systematically identify the factors that either hinder or enhance the effective learning of programming skills using Arduino within the FPI context.

RESEARCH QUESTIONS

i. How does the integration of Arduino-based practical activities in C++ programming courses influence students' attitudes and self-efficacy perceptions relative to traditional teaching methods?

ii. Do engineering and agriculture students differ in their self-perception of programming competencies after experiencing an Arduino-based C++ curriculum, and if so, what are the underlying factors contributing to any observed differences?

iii. What are the specific challenges and facilitators that impact students' perceived competence in programming with C++ and Arduino, and how can instructional strategies be optimized to overcome these obstacles?

LITERATURE REVIEW

Programming in Engineering Education

In engineering disciplines, programming education is frequently intertwined with the development of embedded systems competencies. Numerous studies have demonstrated that programming, particularly using languages such as C++ and platforms such as Arduino, is essential for enabling hands-on learning and problem-solving in engineering curricula. Studies have clearly shown that Arduino, with its C++-inspired programming, is a valuable tool for quick prototyping. Addittionally,its capacity to support a wide range of projects has been linked to increased student motivation and better academic results (El-Abd, 2017; Hoffbeck et al., 2016). The focus on project-based learning in embedded systems courses has also led to significant gains in student performance. This includes higher success rates and a stronger ability to combine hardware and software expertise (El-Abd, 2017).

Research also highlights the benefits of adopting Arduino frameworks in education, such as enhancing system design abilities, improving programming skills, and lowering the perceived difficulty of low-level microcontroller programming. In educational settings where Arduino is a core component, students have displayed a heightened interest in engineering and a more solid comprehension of algorithmic problem-solving strategies (El-Abd, 2017). This indicates that well-conceived courses balancing text-based programming instruction with practical experimentation can effectively demystify intricate engineering concepts, thereby promoting more profound learning and the development of practical skills.

Programming in Agriculture Education

A fundamental shift in agricultural education is required to mirror the transformation occurring in farming due to digital technology. We are witnessing the dawn of Agriculture 4.0, where traditional techniques are gradually being supplemented or even overtaken by smart farming systems incorporating IoT and automation. The benefits of using Arduino sensing and control systems in agriculture have been demonstrated in various studies. A specific illustration is the use of Arduino microcontrollers for overseeing and automating greenhouse environments; by providing continuous data on factors such as temperature, humidity, soil moisture, and light, this technology strongly supports precision agriculture goals (Ardiansah et al., 2020

Arduino's simplicity, cost-effectiveness, and broad support community have made it an appealing tool not just for professional experts but also for instructors seeking to prepare agriculture students with working technological skills. The use of these tools as part of agriculture education supports project-based learning methods, which enable the linking of theoretical knowledge to real-world applications like sensor interfacing and automated irrigation systems. Further, through the integration of programing skills into agriculture curricula, instructors can prepare students to deal with intricate issues of precision agriculture and sustainable resource management, closing the knowledge gap between conventional agriculture and modern technological applications (Mohankumar & Gowtham, 2024).

Self-Perception and Learning Motivation

Students' self-perception of their programming skills is critical to both motivation and academic performance among students. Empirical evidence in engineering education suggests that self-efficacy, or the perception of one's competence in carrying out certain tasks, significantly affects how students approach programming languages like C++ and systems like Arduino (Steelman et al., 2020; Kovari & Katona, 2023). Those who see programming as functional and necessary for their future profession have greater confidence and are more disposed to spend time and effort acquiring the skills. However, students who see programming as esoteric and excessively difficult may experience demotivation and show lowered academic achievement.

Theories of cognitive load and self-efficacy further emphasize the need to design curricula that minimize initial programming barriers through the use of methods like problem-based learning and interactive multimedia instruction. These strategies can demystify difficult programming ideas and reduce the cognitive load involved in mastering abstract syntax and semantics, especially in languages like C++, which have a notorious reputation for possessing steep learning curves (Cheah, 2020; Steelman et al., 2020). The positive effects of these interventions indicate that an organized curriculum that situates the study of programming in real-world applications—like automation in engineering and agriculture—can dramatically improve the self-efficacy and motivation of students.

REVIEW OF RELATED WORKS

Chondrogiannis et al. (2021) have investigated the integration of STEM educationComputational Thinking(CT) into vocational education in Greek agriculture. Theauthors present an extended case study where the conventional training in agriculture iscompared to the new technological issues that have been caused withWith the advent of Industry 4.0, for instance, climate change, heightened growth in population,and the need for automating and intelligent farming systems in agriculture.

El-Abd (2017) offered an elaborate overview of embedded systems education in terms of application areas of using Arduino as an educational tool in engineering curricula. The study considered the issues with infusing embedded systems into curriculum and reviews how Arduino, with its easy-to-use C/C++ environment, can do an efficient mix of theoretical education and project building.

Marín-Marín et al. (2024), in their research, explore the manner in which early exposure to embedded system can enhance creative thinking and problem-solving abilitites. Their review outlines several studies that demonstrate the overall positive effect of incorporating Arduino-based project work into the classroom environment, with the emphasis being placed upon the advantage of applying a tangible, experiential style of learning to complement abstract concepts of programming routinely involved with standard curricula.

Montironi et al. (2017) wrote about the design and utilization of the ChArduino tool, an educational platform that combines the use of a C/C++ interpreter, Ch, with Arduino boards for learning to program. The authors highlight the significance of the tool in easing the learning curve for programming, especially for robotics and electronics novices, through immediate, concrete feedback through an integrated graphical user interface.

Ardiansah et al. (2020), in their research, centered their focus on greenhouse automation and monitoring and offer an elaborate exposition of how precision farming can be revolutionized with the use of affordable sensors, wirelessly communicative technologies, and programmable microcontrollers like the Arduino UNO. The authors offer an elaborate technical exposition of how Arduino programming, which normally takes place in C/C++, can effectively be utilized to regulate environmental parameters like temperature, humidity, and soil moisture, hence real-time data acquisition and control required in automated farming systems.

Methodology

Reasearch design

The research adopted a mixed-methods technique that incorporates both quantitative and qualitative data to investigate students' self-perception and learning experiences with programming tools. Quantitative survey with Likert-scale questions assessed participants' confidence, perceived difficulty, and relevance of C++ and Arduino to industrial applications. Likewise, semi-structured interviews and open questions were administered to gain more insights into challenges, motivational issues, and real-world experience This data mix, along with pretest-posttest assessments and qualitative feedback, enabled an investigation of not just cognitive improvement and skills acquisition, but also affective aspects of self-efficacy and motivation in an inter-disciplinary scenario.

Population and Sampling

The study targeted HND I students from two departments—Mechatronics and Agricultural Technology—at the Federal Polytechnic Ilaro.The total sample size was 82 respondents, consisting of 60 Mechatronics students, who made up 73.2% of the total sample, and 22 Agricultural Technology students, who comprised 26.8% of the total.

Research Intervention:Water Level measurement Project

The aim of the hands-on project was to design and implement a system for measuring the water level in a tank using an Arduino Uno board and an Ultrasonic Sensor. The water depth is displayed on an LCD. Water level measurement and control are crucial in various applications such as water tanks, reservoirs, and other fluid containers. Traditional methods of measuring water levels can be cumbersome and inaccurate. The hands-on project leverages the Arduino Uno board and an Ultrasonic Sensor to create an efficient and automated water level measurement system.Table 1 below provides a brief description of the components required for the project and their features

Duration :4 weeks within a 12 weeks COM 311 course
Key Objective :Pratical Application of C++ and Arduino in a real-word objective
Specific focus: Water level measurement using Arduino UNO and Ultrasonic senso

Table 1: Summary of Water depth measurment/Features

Features	Features
16x2 LCD Display	Displays the water level readings.
Arduino UNO	Arduino UNO is a microcontroller board based on the ATmega328P...
Breadboard:	Temporary circuit prototyping.
Jumper Wires:	For interconnection.
Potentiometer:	Adjusts LCD brightness.
Resistors and Capacitors:	For signal conditioning and stability.
Power Supply:	Battery or USB source.

Data Analysis

Quantitative analysis was performed using descriptive statistics such as means, standard deviations, and medians for all Likert-scale items. Comparative analysis was carried out via t-tests to identify statistically significant differences between the Mechatronics and Agricultural Technology groups. In addition, pretest and posttest scores were compared to assess learning gains in key skill areas. For instance, differences in self-assessed proficiency in writing syntactically correct C++ code and the ability to assemble electronic circuits were statistically compared between groups .

Qualitative responses from the open-ended questions were analyzed using thematic analysis. The process involved coding the responses into emerging themes, such as challenges related to code debugging, sensor calibration, and hardware assembly; improvements in understanding the practical application of C++ in controlling real-world systems; and suggestions for increasing the efficacy of hands-on projects (). The combination of these methods provided a holistic picture of both technical and attitudinal outcomes in the context of interdisciplinary project-based programming education .

Results

Table 2: Demographic and Background Information

Academic Level	Mechatronics (n=60)	Agricultural Tech (n=22)	Total	Percentage (%)
HND I	60	22	82	100
Total	**60**	**22**	**82**	**100%**

As depicted in Table 2(demographic and Background Information), the student sample of 82 students participating in the survey were HND I students, with a department breakdown of 60 Mechatronics students (73.2%) and 22 Agricultural Technology students (26.8%). The total sample represents 100% of the partiticpants at a similar academic level.The departmental distribution visualized in Figure 2 illustrates the proportional representation of Mechatronics and Agricultural Technology students.

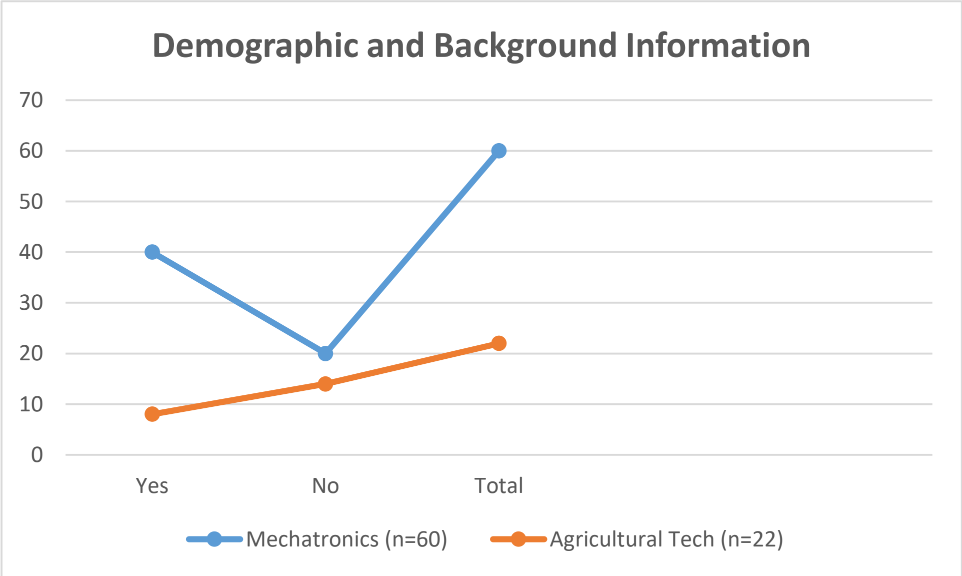

Figure 1

Table 2 : Prior Experience and Self-Assessment of Technical Skills

2. What is your prior experience with programming in C/C++?

Experience Level	**Mechatronics (n=60)**	**Agricultural Tech (n=22)**	**Total**	**Percentage (%)**
None	8	10	18	22.0
Limited	20	7	27	32.9
Moderate	25	5	30	36.6
Extensive	7	0	7	8.5
Total	**60**	**22**	**82**	**100%**

The survey data, as presented in Table 3 revealed that when asked about their prior programming experience in C/C++, most students reported having either limited (32.9%) or moderate (36.6%) exposure. However, a notable difference emerged between departments: close to 45.5% of Agricultural Tech students indicated no prior experience, while only 13.3% of Mechatronics students reported the same

.

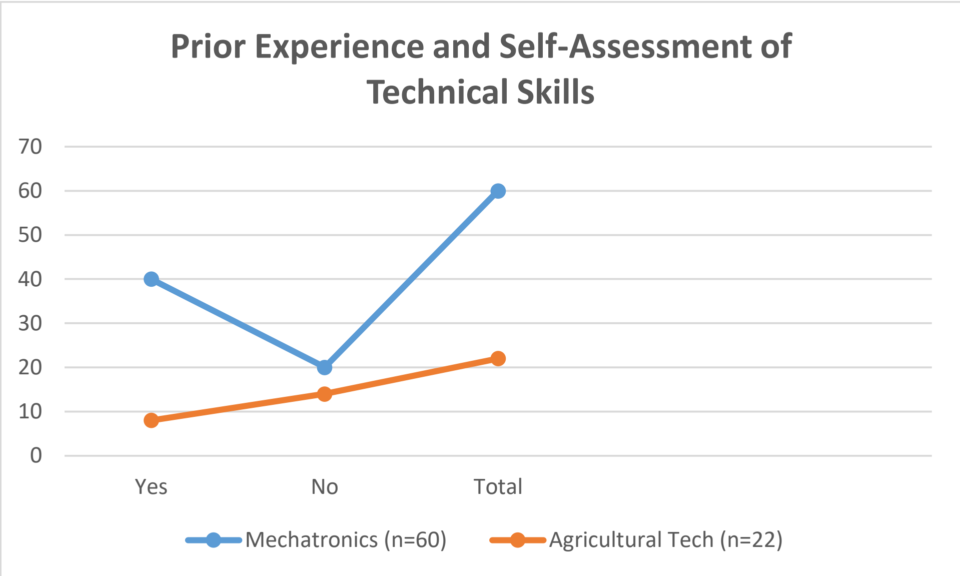

Figure 2

Table 3: Previously knwoledge about Arduino or any microcontroller platforms?

Response	**Mechatronics (n=60)**	**Agricultural Tech (n=22)**	**Total**	**Percentage (%)**
Yes	40	8	48	58.5
No	20	14	34	41.5
Total	**60**	**22**	**82**	**100%**

Table 3 provides an insight into the prior use of Arduino or similar microcontroller platforms, 58.5% of the entire cohort reported having used these tools, but departmental differences were pronounced—with 66.7% of Mechatronics students indicating familiarity versus only 36.4% among Agricultural Technology students

.

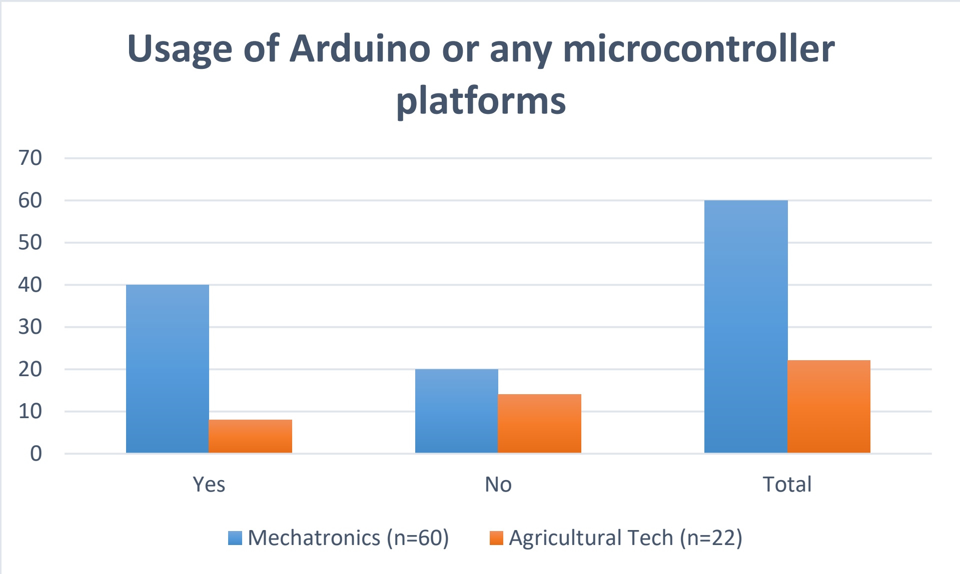

Figure 3

Table 4:

B. Prior Experience and Self-Assessment of Technical Skills :Summary Statistics Table

S/N	Questionnaire Item	Group	Mean	Standard Deviation	Median
1	I can write syntactically correct C++ code to solve basic problems.	Mechatronics	4.8	1.2	5
		Agricultural Tech	3.2	1.0	3
		Overall	4.4	1.3	4
2	I feel comfortable in reading and understanding technical documentation for C++ libraries.	Mechatronics	4.5	1.1	4
		Agricultural Tech	3.1	1.2	3
		Overall	4.1	1.2	4
3	I have sufficient experience in interfacing sensors (e.g., water level sensors) with Arduino boards.	Mechatronics	4.2	1.3	4
		Agricultural Tech	2.7	1.0	2
		Overall	3.9	1.3	4
4	I possess adequate skills to assemble and	Mechatronics	5.1	1.0	5

	troubleshoot basic electronic circuits.	Agricultural Tech	3.0	1.1	3
		Overall	4.6	1.3	5
5	I understand the fundamentals of physical computing and how to integrate hardware with software.	Mechatronics	4.7	1.1	5
		Agricultural Tech	3.4	1.0	3
		Overall	4.3	1.2	4

Self-assessment responses in the technical skills survey highlighted five key items:
1. In the domain of writing syntactically correct C++ code, Mechatronics students reported a mean of 4.8 (median 5) compared to Agricultural Tech students' mean score of 3.2 (median 3).
2. For comfort in reading and understanding technical documentation, the means were 4.5 (Mechatronics) versus 3.1 (Agricultural Tech).
3. Regarding sensor interfacing with Arduino boards, the reported means were 4.2 for Mechatronics students and 2.7 for Agricultural Tech students.
4. In terms of assembling and troubleshooting basic electronic circuits, there was a marked difference: a mean of 5.1 for Mechatronics and 3.0 for Agricultural Tech.
5. Concerning understanding the fundamentals of physical computing, Mechatronics respondents averaged 4.7 compared to 3.4 among Agricultural Tech .

The overall trend indicates that Mechatronics students rated themselves significantly higher across all measured areas, suggesting a stronger technical background and better preparedness for hands-on projects .

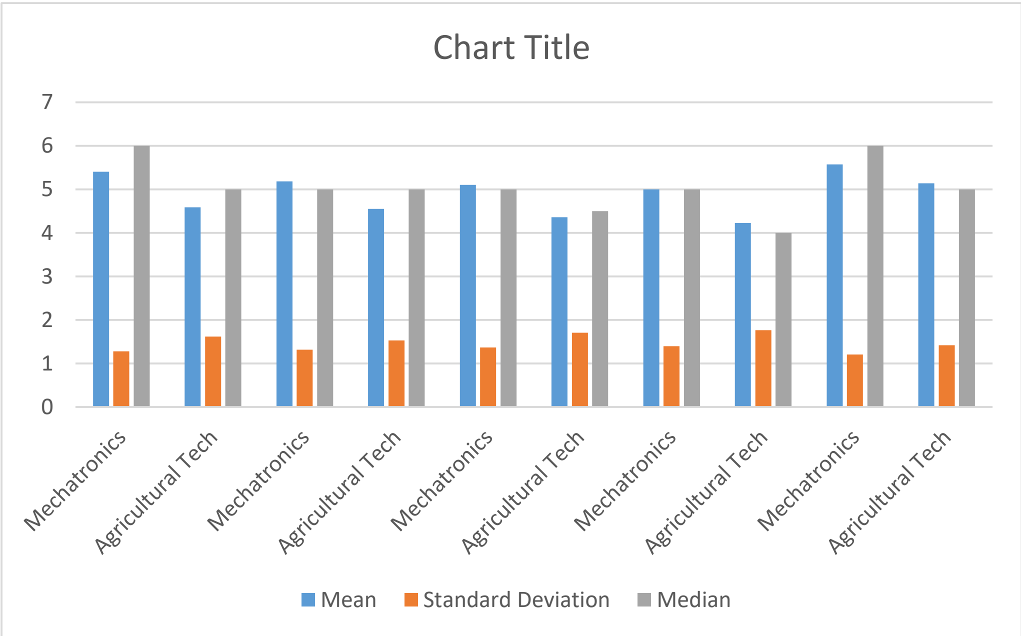

Figure 4

Table 5:

C. Attitudinal Assessment: Summary Statistics Table

S/N	Questionnaire Item	Group	Mean	Std. Deviation	Median	Interpretation
1	I enjoy learning programming and consider it an important skill in my future career.	Mechatronics	4.37	0.82	5.00	Strong positive attitude and recognition of programming's relevance.
		Agricultural Tech	3.91	1.02	4.00	Generally positive attitude, though slightly less enthusiastic.
2	The use of Arduino for practical projects makes the learning process more engaging.	Mechatronics	4.20	0.93	4.00	Majority agree Arduino enhances engagement.
		Agricultural Tech	3.91	1.02	4.00	Positive perception, though slightly more varied in opinion.
3	I believe that programming in C++ offers a significant advantage when integrating hardware projects.	Mechatronics	4.00	1.00	4.00	General agreement on C++'s utility, though not unanimous.
		Agricultural Tech	3.59	1.17	4.00	Moderately positive, but with wider divergence in understanding.
4	Project-based learning improves my interest in pursuing advanced technical courses.	Mechatronics	4.30	0.87	4.00	High level of motivation sparked by practical learning.
		Agricultural Tech	3.91	1.02	4.00	Practical learning positively influences interest, though less strongly.
5	I am motivated to further develop my skills after participating in hands-on programming projects.	Mechatronics	4.53	0.72	5.00	Very high motivation for continued learning.
		Agricultural Tech	3.95	1.12	4.00	Motivated group, though slightly more uncertain compared to Mechatronics.

Attitudinal survey items revealed that both groups generally held positive views toward programming; however, Mechatronics students consistently scored higher across the factors measured. For instance, when asked whether they enjoy learning programming and see it as essential to their future career, Mechatronics students had a mean score of 4.37 (median 5) while Agricultural Tech students scored 3.91 (median 4). Similar trends were observed for items that indicated the engagement-promoting effect of using Arduino for practical projects, and for recognition of C++'s advantages in integrating hardware projects. Moreover, project-based learning was highly valued by both groups, but with slightly higher means among Mechatronics respondents .

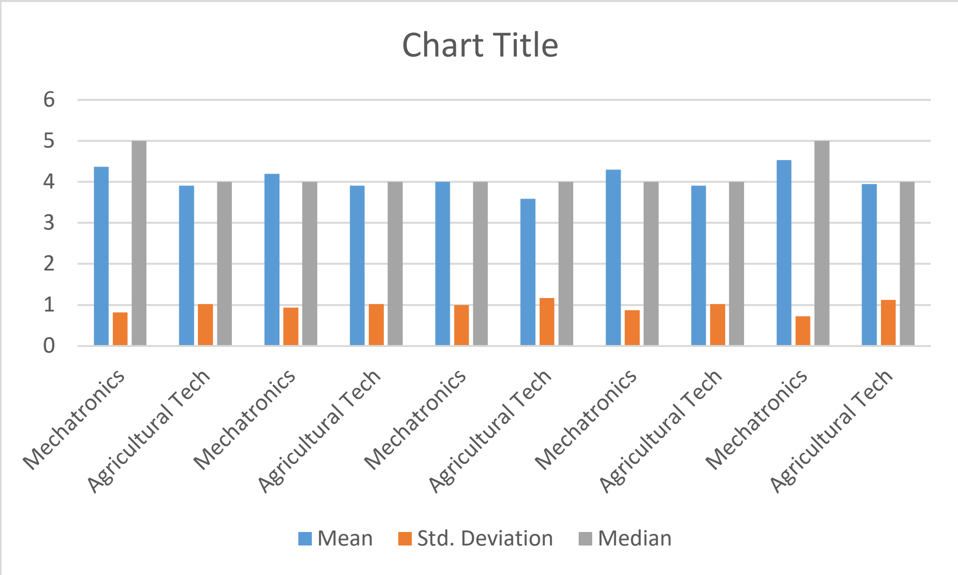

Figure 5

Table 6: Self-Efficacy Measurement

S/N	Questionnaire Item	Group	Mean	Standard Deviation	Median
1	I am confident in my ability to write a C++ program that accurately controls sensor data acquisition.	Mechatronics	5.40	1.28	6.0
		Agricultural Tech	4.59	1.62	5.0
2	I feel assured in my ability to troubleshoot and debug code on the Arduino platform.	Mechatronics	5.18	1.32	5.0
		Agricultural Tech	4.55	1.53	5.0
3	I can effectively integrate sensor data from a water level sensor into a functional Arduino program.	Mechatronics	5.10	1.37	5.0
		Agricultural Tech	4.36	1.71	4.5
4	I am confident in designing and assembling the hardware components required for the water level measurement project.	Mechatronics	5.00	1.40	5.0
		Agricultural Tech	4.23	1.77	4.0
5	Overall, I believe my programming and hardware interfacing skills will improve as a result of this project.	Mechatronics	5.57	1.21	6.0
		Agricultural Tech	5.14	1.42	5.0

The self-efficacy component focused on confidence in critical tasks such as writing C++ programs for sensor data acquisition, troubleshooting, integrating sensor data, and designing or assembling hardware. Mechatronics students reported high confidence levels in these areas, typically scoring means around 5.0 or higher (e.g., overall confidence improvement mean of 5.57). Agricultural Tech students also reported moderate to high confidence but consistently lower compared to their Mechatronics counterparts (e.g., overall confidence mean of 5.14, with lower scores on hardware assembly tasks averaging around 4.23). These quantitative results indicate that while both groups see value and possibility for growth via the project experience, Agricultural Tech students may require further support to reach similar levels of confidence as measured by self-efficacy scales

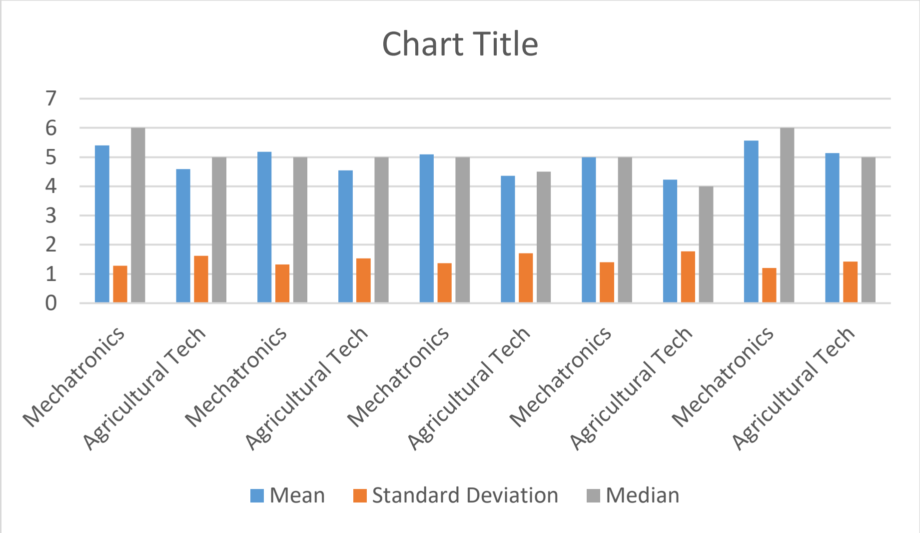

Figure 6

Table 7: Independent Samples T-Test Results Comparing Mechatronics and Agricultural Technology Students

Measure	Mechatronics M (SD)	Agricultural M (SD)	t-value	df	p-value	Cohen's d
C++ Code Writing Proficiency	4.8 (1.2)	3.2 (1.0)	5.71	80	<.001 *	1.43
Technical Documentation Understanding	4.5 (1.1)	3.1 (1.2)	5.09	80	<.001 *	1.23
Sensor Interfacing Experience	4.2 (1.3)	2.7 (1.0)	5.02	80	<.001 *	1.29
Electronic Circuit Skills	5.1 (1.0)	3.0 (1.1)	8.32	80	<.001 *	2.01
Physical Computing Understanding	4.7 (1.1)	3.4 (1.0)	4.93	80	<.001 *	1.23
Programming Enjoyment/Relevance	4.37 (0.82)	3.91 (1.02)	2.09	80	.039*	0.49

*Significant at p < .05 Note: Cohen's d values: 0.2 = small effect, 0.5 = medium effect, 0.8+ = large effect

Independent samples t-tests were conducted to examine differences in self-perception between Mechatronics (n=60) and Agricultural

Technology (n=22) students across multiple dimensions of programming competency and attitude. The analysis revealed statistically significant differences across all measured variables.

Mechatronics students demonstrated significantly higher self-perceived proficiency in C++ code writing ($t(80) = 5.71$, $p < .001$) with a large effect size ($d = 1.43$). Similarly, substantial differences were observed in technical documentation understanding ($t(80) = 5.09$, $p < .001$, $d = 1.23$) and sensor interfacing experience ($t(80) = 5.02$, $p < .001$, $d = 1.29$). The largest disparity was found in electronic circuit skills ($t(80) = 8.32$, $p < .001$, $d = 2.01$), indicating that Mechatronics students rated their hardware proficiency substantially higher than Agricultural Technology students.

Attitudinal measures showed a smaller but still significant difference in programming enjoyment and perceived career relevance ($t(80) = 2.09$, $p = .039$, $d = 0.49$), suggesting that while both groups recognized the importance of programming, Mechatronics students demonstrated more positive attitudes toward the subject.

Qualitative Feedback and Thematic Analysis

Analysis of open-ended responses uncovered several recurring themes. In response to questions about challenges faced in the water level measurement project, the most frequently mentioned issues were code debugging, syntax errors, sensor calibration, and circuit assembly. It was evident from the data that Mechatronics students predominantly faced challenges related to programming details – such as debugging C++ code – whereas Agricultural Tech students reported greater difficulty with hardware interfacing and sensor accuracy. This divergence is indicative of the differential exposure both groups have to embedded systems and low-level programming tasks .

Other open-ended responses pointed to a strengthened appreciation for the practical applications of C++ in controlling hardware and automating measurement tasks. Many students described enhanced understanding of technical documentation and a clearer insight into how embedded systems bridge theoretical constructs and real-world applications. In terms of cross-disciplinary relevance, respondents highlighted that sensor-based projects have notable implications in both automation and smart agriculture. They emphasized the significance of real-time monitoring and data-driven decision-making in improving resource management, particularly in irrigation and environmental monitoring contexts.

Discussion

The results of this study reveal a nuanced grasp of the differential experiences of Mechatronics and Agricultural Technology students in acquiring programming and embedded systems skills.

The t-test results provide statistical confirmation of significant differences in self-perception between the two student groups, with large effect sizes across all technical competency measures. The particularly large effect size ($d = 2.01$) for electronic circuit skills indicates this may be the area of greatest discrepancy between the disciplines. These findings suggest that prior curricular exposure plays a critical role in shaping students' technical self-efficacy.

Interestingly, the smallest effect size was observed in programming enjoyment and perceived relevance ($d = 0.49$), suggesting that despite technical proficiency differences, Agricultural Technology students recognize the importance of programming skills nearly as much as their Mechatronics counterparts. This represents an important foundation upon which educators can build when introducing programming concepts to agricultural students.

These statistical findings complement the qualitative data by quantifying the magnitude of the interdisciplinary differences and highlight specific areas where targeted instructional interventions may be most beneficial. The significant differences across all measures underscore the importance of discipline-specific approaches when integrating programming education across diverse technical fields.

The quantitative data consistently demonstrate that Mechatronics students report higher initial technical proficiency and self-efficacy. Their greater exposure to programming and embedded system projects, as evidenced by a substantially higher percentage of prior use of Arduino and more positive self-assessment ratings on technical items, suggests that the Mechatronics curriculum is more closely aligned with the requirements for automating sensor-based systems .

Agricultural Technology students, by contrast, began the project with significantly lower levels of prior experience, especially in programming and hardware interfacing.

Close to 50% of the Agricultural Technology respondents reported no prior experience with C/C++ programming, while only a small fraction of Mechatronics students fell into this category. This imbalance is further reflected in lower self-assessed scores in areas such as electronic circuit assembly and understanding of technical documentation. In spite these difficulties, Agricultural Technology students stated the project improved their knowledge of sensor integration and real-world applications of C++ in automation. This suggests that while the baseline differences are evident, the hands-on project experience triggered an important positive shift in their perceptions and technical self-efficacy

Thematic findings from the qualitative analysis indicate that experiential learning through projects not only helps bridge the technical skills gap but also increases interdisciplinary awareness. For instance, Mechatronics students often gravitate towards debugging techniques and improving code structure, while Agricultural Technology students prioritized hardware integration—particularly in setting sensors for efficient water depth measurement. However,both groups, demonstrated a stronger appreciation for the importance of C++ and Arduino in designing practical applications that extend beyond conventional classroom activities to real-world contexts such as precision agriculture and smart farming. The open-ended responses confirmed that project-based learning promotes increased understanding of the relationship between software and hardware,consequently motivating students to pursue further training in technology-driven courses

CONCLUSION

These findings provide compelling evidence that a hands-on, project-based techniques using C++ and Arduino can greatly improve both the proficiency and attitude of students that participated in cross-disciplinary programs. Mechatronics students demonstrated higher baseline proficiency and self-efficacy across technical metrics, whereas Agricultural Technology students, despite starting at a lower level, showed marked improvements in their understanding of embedded systems and sensor integration through project participation. In conclusion, both groups reported increased confidence in their ability to program, troubleshoot, and assemble hardware components after completing the water level measurement project. The increase in mean scores from pretest to post-test on parameters such as C++ proficiency and self-efficacy validates the approach and emphasizes the importance of active, experiential learning in addressing real-world technological challenges .

RECOMMENDATION

In light of these findings, the following recommendations are proposed for educators and curriculum developers:

i. Develop an integrated curriculum that begins with foundational, interactive programming exercises in Arduino, progressing naturally to more complex C++ assignments as student confidence and competence increase.
ii. Emphasize real-world applications that demonstrate the practical benefits of programming in discipline-specific contexts, such as greenhouse automation for agricultural students and robotics for engineering students
iii. Introduce peer mentoring, collaborative learning groups, and regular formative assessments to reinforce self-efficacy and diminish the intimidation associated with complex programming syntax
iv. Incorporate modern digital tools, including augmented reality and simulation-based experiments, into laboratory sessions to provide immediate feedback on code execution and hardware interfacing challenges

v. Tailor instructional strategies to address discipline-specific needs, ensuring that Agriculture Technology students, who may face distinct challenges due to traditionally fewer exposures to technical programming, receive additional support through bridging modules and contextualized exercises
vi. Plan longitudinal studies and expand research across multiple institutions to validate these findings and refine approaches in interdisciplinary programming education

REFERENCES

Ardiansah, I., Bafdal, N., Suryadi, E., & Bono, A. (2020). Greenhouse monitoring and automation using Arduino: A review on precision farming and Internet of Things (IoT). International Journal on Advanced Science, Engineering and Information Technology, 10(2), 703-709.

Arigye, J., Magana, A., Lyon, J., & Pienaar, E. (2023). Biomedical and agricultural engineering undergraduate students programming self-beliefs and changes resulting from computational pedagogy. 2023 ASEE Annual Conference & Exposition Proceedings.

Arslan, K., & Tanel, Z. (2020). Analyzing the effects of Arduino applications on students' opinions, attitude and self-efficacy in programming class. Education and Information Technologies, 25, 1-20.

Basha, A. S., Balaji, M. S., & Rajbunisa. (2025). Reinforcing skill development in students of middle school level using Arduino with respect to the reformation of Indian education system. East Journal of Human Science.

Cheah, C.-S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. Contemporary Educational Technology, 12(2), ep280. https://doi.org/10.30935/cedtech/8261

Chondrogiannis, E., Symeonaki, E., Papachristos, D., Loukatos, D., & Arvanitis, K. G. (2021). Computational thinking and STEM in agriculture vocational training: A case study in a Greek vocational education institution. European Journal of Investigation in Health, Psychology and Education, 11(3), 920-936.

El-Abd, M. (2017). A review of embedded systems education in the Arduino age: Lessons learned and future directions. International Journal of Engineering Pedagogy (iJEP), 7(2), 79-93. https://doi.org/10.3991/ijep.v7i2.6845

Erol, O. (2020). How do students' attitudes towards programming and self-efficacy in programming change in the robotic programming process? International Journal of Progressive Education, 16(5), 407-420.

Graven, O. H., & Bjork, J. (2016). The use of an Arduino pocket lab to increase motivation in electrical engineering students for programming. In 2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE) (pp. 101-106). IEEE.

He, J., Lo, D., Xie, Y., & Lartigue, J. W. (2016, October). Integrating Internet of Things (IoT) into STEM undergraduate education: Case study of a modern technology infused courseware for embedded system course. 2016 IEEE Frontiers in Education Conference (FIE), 1-5.

Hoffbeck, J. P., Dillon, H. E., Albright, R. J., Lu, W., & Doughty, T. A. (2016, October). Teaching programming in the context of solving engineering problems. 2016 IEEE Frontiers in Education Conference (FIE), 1-2.

Kovari, A., & Katona, J. (2023). Effect of software development course on programming self-efficacy. Education and Information Technologies, 28, 3349-3367.

Marín-Marín, J.-A., García-Tudela, P. A., & Duo-Terrón, P. (2024). Computational thinking and programming with Arduino in education: A systematic review for secondary education. Heliyon, 10(8), Article e24026.

Mohankumar, A., & Gowtham, R. (2024). Revolutionizing automated agriculture: A brief overview of an advanced smart agriculture system using Arduino UNO. Irish Interdisciplinary Journal of Science & Research, 8(1), 1-7.

Montironi, M. A., Qian, B., & Cheng, H. H. (2017). Development and application of the ChArduino toolkit for teaching how to program Arduino boards through the C/C++

interpreter Ch. Computer Applications in Engineering Education, 25(6), 1053-1065.

Steelman, K. S., Jarvie-Eggart, M., Tislar, K. L., Wallace, C., Manser, N., Bettin, B., & nference Content Access Proceedings.

Ureel, L. C. (2020). Work in progress: Student perception of computer programming within engineering education: An investigation of attitudes, beliefs, and behaviors. 2020 ASEE Virtual Annual Co