TSUE
**TASHKENT STATE UNIVERSITY OF ECONOMICS**

International journal of topics

# INNOVATION SCIENCE AND TECHNOLOGY

**ISSUE 5**

Acceptance of papers **May, 2026**

**Acceptance of papers**

Published monthly

**Topics**

economics, technology, social sciences

**ISSN 3060-5229**

OpenAIRE Google Scholar Academic Resource Index ResearchBib BASE doi Digital Object Identifier OPEN ACCESS

Visit the website
**t.me/jurnal_IST2100**

**EDITOR-IN-CHIEF:**
Mirzaliyev Sanjar Makhamatjon ugli

**DEPUTY EDITOR-IN-CHIEF:**
Makhmudov Nosir Makhmudovich
DSc., Prof., Academican

**DEPUTY EDITOR-IN-CHIEF:**
Ochilov Bobur Bakhtiyor ugli – Senior lecturer at TSUI

THE SCIENTIFIC-POPULAR ELECTRONIC JOURNAL **"INNOVATION SCIENCE AND TECHNOLOGY"** HAS BEEN REGISTERED UNDER THE NUMBER **C-5669633** BY THE AGENCY FOR INFORMATION AND MASS COMMUNICATIONS (AOKA) OF THE REPUBLIC OF UZBEKISTAN, EFFECTIVE FROM OCTOBER 9, 2024.

**CONTACTS**

Phone: **+998 50 737 87 88**

Website: https://ist-journal.uz

Email: innovationist2025@gmail.com

The scientific electronic journal “Innovation Science and Technology” has been included in the list of scientific publications recommended for the publication of main scientific results of dissertations for the award of PhD and DSc degrees in economics and technical sciences, in accordance with the Resolution No. 370 of the Presidium of the Higher Attestation Commission of the Republic of Uzbekistan, dated May 8, 2025.

Electronic publication, Issue 5.
Approved for publication on May, 2026.

## Editorial board:

# CONTENTS

CONTENTS

# PROJECT-ORIENTED ESP32 IOT CURRICULUM WITH A SHARED-RESOURCE LABORATORY MODEL: DESIGN, IMPLEMENTATION, AND LESSONS LEARNED

**Ulugbek Tursunaliev**
School of Exact Sciences,
National Pedagogical University of Uzbekistan,
E-mail: u.tursunaliev@npuu.uz
ORCID: 0009-0009-2042-9916

**Lazizbek Yusupov**
Kimyo International University in Tashkent,
E-mail: l.yusupov@kiut.uz
ORCID: 0009-0002-9215-1000

**Abstract.** Traditional IoT education often follows a topic-based instructional model, progressing sequentially through sensors, actuators, communication protocols, and embedded programming concepts. This paper presents an alternative project-oriented IoT curriculum implemented over a 15-week semester at a private university in Uzbekistan for 44 undergraduate computer science students. The course was designed around complete functional projects rather than isolated theoretical topics, enabling students to develop practical engineering competencies through hands-on experience.

The laboratory environment employed a shared-resource model in which student pairs rotated through 12 workstations equipped with ESP32 microcontrollers and various sensor-actuator combinations, including DHT11 temperature and humidity sensors, ultrasonic sensors, RFID modules, and L298N motor drivers. To address concerns regarding excessive dependence on AI-assisted coding tools, the curriculum introduced a differentiated AI usage policy: students were encouraged to use artificial intelligence tools for conceptual understanding and research purposes, while all source code had to be written independently and verified through oral technical interviews.

The paper also provides a comparative analysis with a separate full-year embedded systems program in which more structured and standardized instructional approaches produced significantly stronger educational outcomes. Based on the analysis, the study proposes several practical recommendations for effective IoT curriculum design, including the implementation of signed technical requirements, rigorous early-stage assessments, mandatory mock technical interviews, and the avoidance of individual MVP projects in favor of standardized project-based exercises.

**Keywords:** IoT education, ESP32, project-based learning, curriculum design, AI in education, engineering pedagogy

**Annotatsiya.** An’anaviy IoT ta’limi odatda mavzular asosidagi yondashuvga tayangan holda sensorlar, aktuatorlar va aloqa protokollarini ketma-ket oʻrganish orqali tashkil etiladi. Ushbu maqolada Oʻzbekistondagi xususiy universitetda 44 nafar kompyuter fanlari talabalari uchun 15 haftalik semestr davomida amalga oshirilgan loyiha yoʻnaltirilgan IoT oʻquv dasturi taqdim etiladi. Kurs alohida mavzular emas, balki toʻliq funksional loyihalar asosida tashkil etilib, talabalarning amaliy muhandislik kompetensiyalarini rivojlantirishga qaratildi.

Laboratoriya mashgʻulotlari umumiy resurslardan foydalanish modeli asosida tashkil qilinib, talabalar juftliklarda 12 ta ish stansiyasi boʻylab almashib ishladilar. Har bir stansiya ESP32 mikrokontrollerlari hamda DHT11 harorat va namlik sensori, ultratovush sensori, RFID modullari va L298N motor drayverlari kabi turli sensor va aktuator kombinatsiyalari bilan jihozlandi. Sun’iy intellekt yordamida kod yozish bilan bogʻliq muammolarni kamaytirish maqsadida differensial AI siyosati joriy qilindi: talabalar AI vositalaridan tushunchalarni oʻrganish va tadqiqot olib borishda foydalanishga ragʻbatlantirildi, biroq barcha dastur kodlari mustaqil yozilishi va ogʻzaki texnik suhbatlar orqali tasdiqlanishi talab etildi.

Maqolada shuningdek, tuzilmaviy va standartlashtirilgan yondashuvlar qo‘llanilgan boshqa bir yillik embedded systems dasturi bilan qiyosiy tahlil ham keltirilgan bo‘lib, unda ancha samarali natijalar kuzatilgani qayd etilgan. Tadqiqot asosida IoT o‘quv dasturlarini ishlab chiqish bo‘yicha quyidagi tavsiyalar ilgari suriladi: imzolangan texnik topshiriqlarni joriy qilish, dastlabki bosqichlarda qat’iy baholash tizimini qo‘llash, majburiy mock intervyular tashkil etish hamda individual MVP loyihalaridan voz kechib, standartlashtirilgan amaliy mashg‘ulotlarga ustuvorlik berish.

**Kalit so‘zlar:** IoT ta’limi, ESP32, loyiha asosida o‘qitish, o‘quv dasturi dizayni, ta’limda sun’iy intellekt, muhandislik pedagogikasi.

**Аннотация**Традиционное обучение IoT обычно строится на тематическом подходе, при котором последовательно изучаются датчики, исполнительные устройства и коммуникационные протоколы. В данной статье представлен альтернативный проектно-ориентированный учебный курс по IoT, реализованный в течение 15-недельного семестра в одном из частных университетов Узбекистана для 44 студентов направления Computer Science. Учебный процесс был организован вокруг полноценных функциональных проектов, а не отдельных теоретических тем, что позволило развивать практические инженерные компетенции студентов.

Лабораторные занятия были организованы по модели совместного использования ресурсов, при которой студенты работали в парах и поочередно использовали 12 рабочих станций, оснащённых микроконтроллерами ESP32 и различными комбинациями датчиков и исполнительных устройств, включая датчики температуры и влажности DHT11, ультразвуковые датчики, RFID-модули и драйверы двигателей L298N. Для решения проблемы чрезмерной зависимости от инструментов искусственного интеллекта была внедрена дифференцированная политика использования AI: студентам разрешалось использовать AI-инструменты для изучения концепций и проведения исследований, однако весь программный код должен был быть написан самостоятельно и подтверждён в ходе устных технических собеседований.

В статье также представлен сравнительный анализ с другой годичной программой по embedded systems, где использование более структурированных и стандартизированных подходов продемонстрировало значительно лучшие образовательные результаты. На основе проведённого анализа предлагаются практические рекомендации по разработке IoT-курсов, включая внедрение подписанных технических заданий, проведение строгих ранних оцениваний, обязательные mock-интервью и отказ от индивидуальных MVP-проектов в пользу стандартизированных практических заданий.

**Ключевые слова:** IoT-образование, ESP32, проектно-ориентированное обучение, разработка учебных программ, искусственный интеллект в образовании, инженерная педагогика.

## INTRODUCTION

Internet of Things (IoT) education represents a relatively new and rapidly developing component of computer science and engineering curricula worldwide [1-2]. Unlike traditional disciplines with decades of established pedagogical methodologies and standardized instructional frameworks, IoT education is still evolving, and educational institutions continue experimenting with different teaching models, laboratory structures, and assessment strategies [3]. The widespread availability of low-cost microcontroller platforms such as Arduino and ESP32 has significantly transformed embedded systems and hardware-oriented programming education by making practical experimentation more accessible to students [4]. Nevertheless, the most effective pedagogical approaches for teaching IoT systems remain insufficiently explored.

This paper contributes to the growing body of research on IoT engineering education by documenting both successful and unsuccessful experiences in curriculum implementation. The study was conducted at a private university in Uzbekistan with 44 third-year computer science students during a 15-week semester. The course utilized ESP32 microcontrollers programmed through the Arduino IDE environment and focused primarily on developing practical engineering skills rather than extensive theoretical coverage. Students worked in pairs using 12 laboratory workstations equipped with shared sensor and actuator components, including DHT11 temperature and humidity sensors, ultrasonic sensors, RFID modules, servo motors, relays, and L298N motor drivers. The total laboratory cost was approximately 2,000–3,000 USD, demonstrating the feasibility of implementing a resource-efficient IoT laboratory model in developing educational environments.

## LITERATURE REVIEW

Initially, the course was designed around the assumption that project-oriented learning — where students independently select, design, and implement individual IoT projects — would increase motivation, engagement, creativity, and conceptual understanding compared to traditional topic-based instruction [5-6]. However, practical implementation demonstrated that this hypothesis was largely incorrect in the context of introductory IoT education. The study documents the primary reasons why individual MVP-style projects produced weak educational outcomes and explains which alternative instructional approaches generated significantly better results in a comparative embedded systems program.

The rapid emergence of large language models (LLMs) and AI-assisted coding tools has also introduced new challenges into programming and engineering education [7-8]. Students are now capable of generating functional source code without fully understanding the underlying logic, algorithms, hardware interaction, or system architecture. As a result, many traditional assessment approaches based solely on project completion or code submission have become insufficient for accurately evaluating technical competence. This study therefore investigates how oral technical interviews, practical demonstrations, and structured questioning can be integrated into IoT education to verify authentic understanding and reduce overreliance on AI-generated solutions.

In addition to documenting curriculum implementation experiences, this paper aims to provide practical recommendations for educators designing introductory IoT and embedded systems courses under limited laboratory resources and increasing AI-assisted learning conditions. The findings may be particularly relevant for universities in developing countries where shared laboratory infrastructure and cost-effective teaching models are essential for sustainable engineering education.

## RESEARCH METHODOLOGY

This study employed a quasi-experimental comparative design to evaluate the effectiveness of different pedagogical approaches in IoT and embedded systems education at two private universities in Uzbekistan. The primary study site (University A) involved 44 third-year Computer Science students enrolled in a 15-week IoT course. The comparative study site (University B) included approximately 80 Information Systems Engineering students participating in a full-year instructional sequence consisting of an Embedded Systems course during the first semester followed by an IoT course in the second semester.

The comparative design enabled the analysis of differences in learning outcomes, student engagement, project quality, and knowledge retention between a short-term project-oriented curriculum and a longer, more structured instructional model. Both programs focused on practical embedded systems development using ESP32 microcontrollers and Arduino IDE environments.

The 15-week curriculum implemented at University A was divided into two major instructional phases. Weeks 1–7 focused on foundational hardware and embedded programming concepts, including:

digital input/output operations;
analog sensor integration;
serial communication protocols;
PWM control;
basic actuator management;
debugging and hardware troubleshooting.

Week 8 was allocated for the midterm examination and practical evaluation.

Weeks 9–15 concentrated on internet connectivity and IoT-oriented technologies, including:

WiFi communication;
HTTP requests and REST APIs;
MQTT communication protocols;
cloud-based IoT platforms;
remote monitoring systems;
multi-device integration.

As part of the final assessment, students were required to develop projects integrating at least three different hardware or software technologies into a single functional IoT system.

The laboratory infrastructure was organized according to a shared-resource rotational model in which student pairs rotated through available hardware components and workstations [9]. The laboratory equipment included:

13 ESP32 development boards;
13 robotic car kits with L298N motor drivers;
DHT11 temperature and humidity sensors;
MQ-series gas sensors;
ultrasonic distance sensors;
RFID modules;
Bluetooth communication modules;
LCD displays with I2C interfaces;
servo motors;
joysticks;
PIR motion sensors;
capacitive touch sensors;
additional breadboards, jumper wires, and power supply components.

This model allowed efficient utilization of limited laboratory resources while ensuring that all students obtained hands-on experience with diverse hardware configurations.

A flexible 130-point assessment rubric was developed to evaluate student performance while accommodating varying levels of project complexity and completion. Although the maximum officially recorded course grade remained 100 points, the extended rubric provided opportunities for bonus achievements and differentiated evaluation of advanced technical implementations. Similar approaches are frequently utilized in interdisciplinary engineering education where project scope and technical depth may differ substantially among student groups [10].

The assessment framework included the following components:
laboratory participation and attendance;
weekly practical exercises;
midterm practical examination;
final IoT project implementation;
technical documentation;
oral technical interviews;
hardware integration quality;
code readability and functionality;
presentation and demonstration performance.

During the semester, additional assessment criteria related to hardware housing design, power supply reliability, PCB integration, and system stability were introduced after observing inconsistencies in project quality and project management practices among student teams.

Special emphasis was placed on oral technical interviews as a mechanism for verifying authentic student understanding in the context of widespread AI-assisted code generation. Students were required to explain system architecture, wiring decisions, programming logic, communication protocols, and debugging processes during project evaluations. This approach helped distinguish independently developed understanding from externally generated solutions (Table 1).

Table 1
Assessment rubric (130 points available, 100 maximum recorded)[1]

| Criterion | **Points** | **Description** |
|---|---|---|
| Core functionality | 45 | Sensors read correctly, actuators respond, logic executes |
| Code quality | 20 | Modular architecture, meaningful names, inline comments |
| Housing/enclosure | 20 | 3D printed (full) or cardboard/fomex (reduced) |
| Standalone power | 20 | TP4056 + converters (full) or alkaline (reduced) |
| PCB implementation | 15 | Soldered connections, no breadboard |
| Online publication | 10 | GitHub, Hackster.io, Thingiverse documentation |

---

[1] Author's development

The curriculum implemented a differentiated AI usage policy designed to balance the educational benefits of modern AI tools with the necessity of developing authentic programming competence. Students were encouraged to use artificial intelligence tools, online documentation, and YouTube tutorials for conceptual exploration, component familiarization, troubleshooting assistance, and general technical research [7]. However, all source code submitted for assignments and projects was required to be written independently by students.

To ensure authentic understanding and reduce overreliance on AI-generated solutions, verification was conducted through structured oral technical interviews [11], [12]. During these interviews, students were required to explain their implementations line by line, including hardware connections, software logic, communication protocols, and debugging decisions. The interview methodology intentionally combined hardware-oriented and software-oriented questioning in rapid succession to evaluate practical understanding rather than memorized explanations.

A commonly used interview strategy involved abrupt transitions between physical hardware and source code analysis. For example, instructors asked questions such as: *"This wire is connected to pin 10 — show exactly where pin 10 is configured and used in your program code."*

University B implemented a fundamentally different pedagogical model emphasizing structured foundational learning before hardware interaction. Students initially spent approximately half of the first semester studying pure C programming concepts prior to working with physical embedded systems hardware. Seminar activities required students to write code manually on paper, while the use of mobile phones and internet-connected devices was strictly prohibited during practical sessions.

The program utilized the ARM STM32 platform together with STM32CubeMX, an industry-standard embedded systems development environment widely used in professional engineering practice. Unlike the ESP32-based project-oriented curriculum, the comparative program focused primarily on mastering underlying technologies and communication principles rather than specific sensors or peripherals.

## ANALYSIS AND RESULTS

All 44 students at University A chose to develop individual projects rather than participate in standardized laboratory exercises or predefined system implementations. The selected projects covered a broad range of IoT application domains, including:

smart door lock systems;
automated garbage sorting devices;
pet feeding systems;
baby monitoring systems;
soil moisture monitoring and irrigation systems;
LED-based hourglass simulations;
smart vacuum cleaner prototypes;
heater–cooler automation systems;
camera-based monitoring and security applications.

At the initial stage, the high degree of project freedom significantly increased student enthusiasm and engagement. Students expressed strong motivation toward building personalized systems that reflected their own interests and creativity. However, as the semester progressed, substantial pedagogical and organizational challenges emerged.

The final examination results indicated that 41 out of 44 students successfully passed the course, corresponding to a 93% pass rate. Three students initially failed the course assessment but later completed successful retake examinations. In particular, two students working on heater–cooler automation projects demonstrated notable improvement during the retake process after receiving additional guidance and participating in repeated technical interviews (Figure 1).

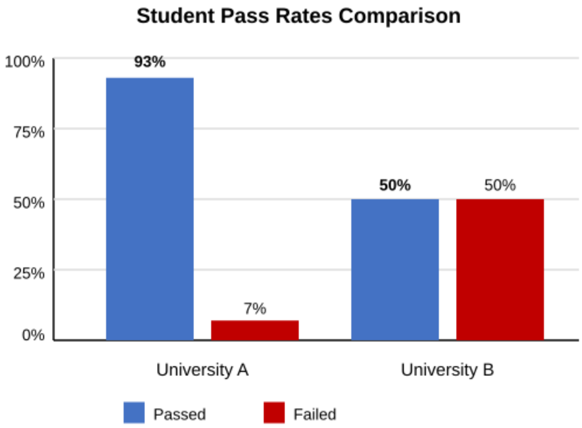

Figure. 1. Student pass rates comparison between University A (project-oriented, one semester) and University B (structured, full year with harsh midterm assessment)[2]

The study revealed that 100% of students used large language models (LLMs) for code generation despite the official policy requiring independently written source code. Evidence of extensive AI-assisted development became immediately apparent during oral technical interviews.

The findings demonstrated a strong correlation between heavy dependence on AI-generated code and weak conceptual understanding. In general:

the strongest students demonstrated understanding of approximately 70–80% of their codebase;

average-performing students understood roughly 40–50%;

struggling students frequently understood almost none of the implementation beyond superficial functionality.

The physical implementation quality varied substantially across the student cohort, as illustrated in Figure 2. While several projects demonstrated relatively organized wiring, stable hardware integration, and acceptable enclosure design, many others exhibited inconsistent construction quality, unreliable connections, inadequate cable management, and unstable power configurations (Figure 2).

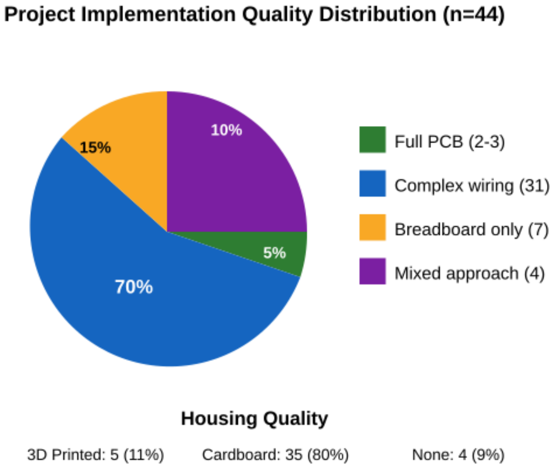

Fig. 2. Distribution of implementation quality across 44 student projects showing PCB approach and housing materials[3]

Only 2–3 students successfully achieved full PCB-based implementations in their final projects. The overwhelming majority produced complex wired assemblies that were functionally operational but poorly organized and difficult to maintain. This occurred after students had been informed that projects relying exclusively on breadboard-based prototyping would receive reduced evaluation scores. Among all technical components, power supply design and implementation emerged as the weakest aspect across nearly all projects. Many systems suffered from unstable voltage delivery, insufficient power regulation, or unreliable

---

[2] Author's development

[3] Author's development

wiring practices, negatively affecting system reliability and demonstration quality.

For example, one student project initially proposed as a “smart vacuum cleaner” never progressed beyond basic Bluetooth-controlled robotic car functionality. Technically, the system could no longer be classified as an IoT application because no internet connectivity or remote communication infrastructure was implemented. In another case, a vehicle license plate recognition project was quietly transformed into a simple motion-detection barrier system after one member of the student team left the project group without notifying the instructor.

The structured instructional model implemented at University B produced substantially different educational outcomes. Approximately 40 students failed the program; however, these failures were primarily associated with students who invested minimal or no meaningful effort into the course. Importantly, students who successfully passed the program demonstrated significantly stronger conceptual understanding, technical confidence, and problem-solving ability compared to the project-oriented cohort at University A.

One major factor contributing to this difference was the implementation of rigorous early-stage assessment procedures. Harsh initial evaluations effectively filtered out students who were not actively engaging with the material, thereby establishing clear academic expectations from the beginning of the program (Table 2).

Table 2
Comparative analysis of two IoT/embedded programs[4]

| Aspect | University A | University B |
|---|---|---|
| Duration | One semester | Full year |
| Students | CS, 44 students | ISE, ~80 students |
| Platform | ESP32 + Arduino IDE | STM32 + CubeMX |
| Projects | Individual MVP | Standardized exercises |
| Housing/PCB/Power | Required | Eliminated |
| Midterm severity | Lenient | Harsh (~50% fail) |
| Mock interviews | None | Mandatory, 2-3 attempts |
| Code writing | Computer with LLM access | Paper, no phones |
| Final pass rate | 93% | ~50% |
| Knowledge retention | Poor | Satisfactory |

One of the unexpected findings concerned patterns of student engagement and classroom behavior. Students who were relatively quiet during lessons, rarely participated in discussions, and attracted little attention often produced significantly better project outcomes when they consistently focused on implementation quality and independent practice. In many cases, these students exceeded instructor expectations through disciplined and systematic work habits.

Conversely, highly talkative students who frequently sought instructor attention and actively participated in classroom discussions often demonstrated weaker practical results and lower technical competence during final evaluations. This observation suggests that visible classroom engagement does not necessarily correlate with meaningful learning outcomes or engineering competence in project-oriented courses [13].

The central finding of this study is highly counterintuitive: providing students with extensive freedom to select and independently develop individual IoT projects produced significantly weaker educational outcomes than structured and standardized practical exercises. This result challenges the widely promoted narrative surrounding unrestricted student-driven project-based learning [5, 14] and suggests that such pedagogical models require extensive scaffolding, supervision, and constraint mechanisms to function effectively in introductory engineering education contexts.

The findings indicate that novice IoT learners often lack sufficient technical foundations, project management discipline, and debugging experience to independently manage open-ended engineering projects. Without strict structural boundaries, students frequently prioritize visible functionality over conceptual understanding, resulting in shallow learning and weak long-term retention.

Another major observation concerns the universal adoption of large language models despite explicit institutional policies restricting AI-generated code usage. The study demonstrates that prohibition-based approaches are largely ineffective in modern programming education environments. Students will continue using AI tools whenever accessible, especially when under time pressure or facing technical difficulty.

Consequently, the educational focus must shift away from attempting to prohibit AI usage toward developing assessment methods capable of verifying genuine understanding regardless of how code was initially produced.

---

[4] Author’s development

Structured oral interviews proved highly effective for this purpose; however, they require substantial instructor time, preparation, and technical expertise [11-12].

The comparative findings further suggest that stronger foundational instruction, repetitive low-level programming practice, and technology-focused exercises may produce more robust engineering competence than highly individualized creative projects during early stages of embedded systems education.

Based on the findings of this study, the following recommendations are proposed for the design and implementation of IoT and embedded systems curricula:

Standardized exercises with clearly defined inputs, outputs, constraints, and expected behaviors produce more reliable educational outcomes than unrestricted open-ended projects.

Project documentation should explicitly define:

hardware components;
software requirements;
implementation milestones;
individual team member responsibilities;
assessment criteria.

Midterm evaluations should establish strict academic expectations early in the semester. High initial failure rates are acceptable if they effectively identify disengagement and motivate serious participation.

Students should practice oral technical defense procedures before formal evaluations. Multiple interview attempts may be required to develop confidence and authentic understanding.

Tasks such as enclosure construction, advanced power supply packaging, and PCB manufacturing may distract students from core learning objectives in introductory courses [6].

Equipment removal should be prohibited, and systematic laboratory cleanup procedures should be mandatory. Instructors should allocate 5–10 minutes before class completion for disassembly and equipment return.

Requiring students to purchase certain inexpensive components may improve responsibility and reduce logistical complications associated with shared institutional equipment.

Courses emphasizing embedded systems fundamentals before introducing internet connectivity and IoT applications produce stronger long-term retention and technical maturity [4].

Carefully designed tutorials and guided instructional videos often provide more coherent and pedagogically effective learning experiences than iterative AI prompting [7-8].

Instructors should demonstrate hardware concepts in real time using projectors and cameras focused on physical devices, while reserving seminar sessions primarily for supervised hands-on implementation.

Future research should investigate the optimal balance between student project freedom and structural instructional constraints through controlled experimental designs. Additional studies are needed to explore effective methods for identifying and addressing LLM-dependent learning behaviors before superficial understanding becomes deeply established [8, 15].

The scalability of oral technical interviews in large engineering courses also requires further investigation, particularly regarding instructor workload, standardization procedures, and assessment reliability. Moreover, future work should examine whether the findings of this study are transferable to different cultural, institutional, and economic educational environments beyond the context of private universities in Uzbekistan.

Further research may also explore:

AI-assisted pedagogical frameworks for embedded systems education;
automated methods for detecting shallow code comprehension;
hybrid assessment models combining oral, practical, and written evaluation;
optimal laboratory resource-sharing strategies for low-budget engineering programs;
long-term retention differences between structured and project-oriented curricula.

Overall, the findings suggest that successful IoT education in the era of AI-assisted programming requires substantial reconsideration of traditional project-based teaching assumptions and assessment methodologies.

## CONCLUSION AND RECOMMENDATIONS

This paper documented the limitations and challenges of project-oriented IoT education when implemented without sufficient structural guidance, accountability mechanisms, and pedagogical control. As IoT education remains a relatively new academic discipline without fully established instructional traditions or standardized teaching methodologies [1-2], it is important for educational institutions to document unsuccessful experiences as carefully as successful implementations. Such analysis contributes to the development of more effective engineering education practices.

The research also highlights the transformative impact of large language models on programming education. Traditional assessment approaches based primarily on code submission are no longer sufficient for verifying authentic student understanding. Functional code alone can no longer be considered reliable evidence of programming competence or engineering knowledge.

## REFERENCES

1. M. Kusrini, A. Ashari, and S. Hartati, “Development of Educational Low-Cost ESP32-Based Platform for IoT Applications,” *Engineering Proceedings*, vol. 56, no. 1, 2023.
2. D. Hercog, T. Lerher, M. Truntič, and O. Težak, “Design and Implementation of ESP32-Based IoT Devices,” *Sensors*, vol. 23, no. 15, Art. no. 6739, Jul. 2023.
3. J. Froyd, P. Wankat, and K. Smith, “Five Major Shifts in 100 Years of Engineering Education,” *Proceedings of the IEEE*, vol. 100, pp. 1344–1360, 2012.
4. M. El-Abd, “A Review of Embedded Systems Education in the Arduino Age: Lessons Learned and Future Directions,” *International Journal of Engineering Pedagogy*, vol. 7, no. 2, pp. 79–93, 2017.
5. P. Guo, N. Saab, L. S. Post, and W. Admiraal, “A Review of Project-Based Learning in Higher Education: Student Outcomes and Measures,” *International Journal of Educational Research*, vol. 102, Art. no. 101586, 2020.
6. D. Kolb, *Experiential Learning: Experience as the Source of Learning and Development*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
7. E. L. Ouh, B. K. S. Gan, K. J. Shim, and S. Wlodkowski, “ChatGPT, Can You Generate Solutions for My Coding Exercises?” in *Proceedings of ITiCSE 2023*, Turku, Finland, pp. 54–60, 2023.
8. M. Wermelinger, “Using GitHub Copilot to Solve Simple Programming Problems,” in *Proceedings of SIGCSE 2023*, Toronto, Canada, 2023.
9. J. García-Zubía et al., “Empirical Analysis of the Use of the VISIR Remote Lab in Teaching Analog Electronics,” *IEEE Transactions on Education*, vol. 60, no. 2, pp. 149–156, 2017.
10. C. Picard, C. Hardebolle, R. Tormey, and J. Schiffmann, “Which Professional Skills Do Students Learn in Engineering Team-Based Projects?” *European Journal of Engineering Education*, vol. 47, no. 2, pp. 314–332, 2022.
11. M. Huxham, F. Campbell, and J. Westwood, “Oral versus Written Assessments: A Test of Student Performance and Attitudes,” *Assessment & Evaluation in Higher Education*, vol. 37, no. 1, pp. 125–136, 2012.
12. H. Qi et al., “Can Oral Exams Increase Student Performance and Motivation?” in *Proceedings of the ASEE Annual Conference*, Minneapolis, MN, 2022.
13. M. Hasan, J. M. Lodge, A. Karim, and M. S. H. Khan, “Exploring Students’ Conceptions of Project-Based Learning,” *IEEE Transactions on Education*, vol. 67, no. 2, pp. 234–244, 2024.
14. J. Zambrano-Mieles, M. Tupac-Yupanqui, S. Mari-Loardo, and C. Vidal-Silva, “Integrating ESP32-Based IoT Architectures and Cloud Visualization,” *Computers*, vol. 15, no. 1, Art. no. 51, 2026.
15. Gilchrist, *Industry 4.0: The Industrial Internet of Things*. New York: Apress, 2016.
16. R. Sanchez-Iborra and A. F. Skarmeta, “TinyML-Enabled Frugal Smart Objects,” *IEEE Access*, vol. 8, pp. 155279–155287, 2020.

**Proofreader:** Zokir ALIBEKOV

**Layout and Designer:** Oloviddin Sobir ugli

## 2026. № 5

To publish articles in this journal, you may submit articles, advertisements, stories, and other creative materials through the following links. Materials and advertisements are published on a paid basis.

You may subscribe to the journal at any time using the following details. Once subscribed, please send a screenshot or photo of your payment confirmation to our Telegram page @iqtisodiyot_77. Based on this, we will send the latest issue of the journal to your address each month.

“The journal “INNOVATION SCIENCE AND TECHNOLOGY” has been registered by the Agency for Information and Mass Communications under the Administration of the President of the Republic of Uzbekistan from 09.10.2024 under the registration number №390637. License number: C-5669633. PNFL: 30407832680027

**Our address:** Tashkent city, Yunusobod district, 19th block,
House 17.

TASHKENT STATE
UNIVERSITY OF ECONOMICS

Google Scholar
OPEN ACCESS
ULRICHSWEB™ GLOBAL SERIALS DIRECTORY
Academic Resource Index ResearchBib
ISSN INTERNATIONAL STANDARD SERIAL NUMBER INTERNATIONAL CENTRE
CYBERLENINKA
OpenAIRE
ROAD
НАУЧНАЯ ЭЛЕКТРОННАЯ БИБЛИОТЕКА eLIBRARY.RU
BASE
INDEX COPERNICUS INTERNATIONAL
Crossref
doi Digital Object Identifier
CiteFactor Academic Scientific Journals
zenodo

**Acceptance of articles**

Published every monthly

## Directions

Social, economic, political, technological, scientific

Scopus || **Scientific electronic journal specializing in Scopus**

**CERTIFICATE NUMBER: №390637**

**ORDER NUMBER ACCORDING TO THE LICENSE REGISTER: C-5669633**

**CONTACT:**

Contact us
**+998 50 737 87 88**

Telegram channel
**t.me/scopus_IST2100**

Journal official website
**https://ist-journal.uz/index.php/IST**