SRSTI 14.35.07

Shynar Azimkhan*[1], Celal Karaca[2]
[1]L.N. Gumilyov Eurasian National University, Astana, Kazakhstan
[2]Aksaray University, Aksaray, Turkey
(E-mail:[1]shynarazimkhan@gmail.com, [2]karaca42@gmail.com)

Using ThingsBoard as a tool in teaching IoT to Computer Science students

Abstract
Internet of Things (IoT) is expanding extremely fast in various areas, in particular in education sector. Computer science education nowadays requires practical learning environments. However, limited access to physical hardware can impede students from understanding and applying IoT concepts effectively. This study explores the use of ThingsBoard, an open-source IoT platform, as a virtual tool for teaching IoT to undergraduate computer science students. Students participated in a series of guided activities that involved simulating sensor data using Python, transmitting telemetry to ThingsBoard via HTTP, and visualizing the data through interactive dashboards. Pre- and post-tests were administered to assess students' theoretical understanding and practical skills. Results indicated a significant improvement in students' ability to understand IoT architecture, data flow, and cloud integration. Student feedback also highlighted the platform's advantages, engagement, and effectiveness in realization of real-world IoT scenarios. The study concludes that ThingsBoard provides a scalable and accessible solution for improving IoT education and recommends its integration into computer science curricula.

Keywords: IoT, ThingsBoard, platform, IoT scenarios, Computer Science.

Received from March 6, 2025; revised from March 16, 2025; accepted from March 26, 2025.

*Corresponding author: Shynar Azimkhan; E-mail: shynarazimkhan@gmail.com

Introduction

The Internet of Things (IoT) is a network that allow devices interact, communicate, and generate data in real-time. The demand for IoT solutions continues to grow among various industries, accordingly computer science students must obtain not only theoretical knowledge but also practical competencies to develop and manage IoT systems. In order to reach effective IoT education, access to physical sensors, microcontrollers, and communication networks is required, while the resources may be limited in many educational institutions.

Traditional IoT lab setups require significant investment such as microcontrollers, sensors, actuators, and networked environments causing difficulties for many institutions to provide stable practical experience to all students (Domínguez-Bolaño et al., 2024). This leads to a remarkable gap between what is taught and what is expected in the job market, where experimental knowledge and system integration skills are highly valued.

In order to bridge this gap, virtual platforms and cloud-based simulation tools have become known as valuable addition or alternatives to hardware-based labs (Viswanadh et al., 2024). One such tool is ThingsBoard, an open-source IoT platform that allows users to simulate data flow, process telemetry, and create real-time dashboards without the need for physical devices. It provides teachers with the comfort to build interactive assignments and projects that reflect real-world IoT use examples.

There are numerous studies related to the integration of platforms in education (Arduino, Raspberry Pi, and cloud services). However, many of these platforms require extensive setup or continuous service. They may also cause barriers for beginners due to programming complexity or infrastructure limitations, which can interfere initial engagement. Furthermore, commercial solutions often include hidden costs or have limited educational licenses, reducing their sustainability in long-term curriculum planning. ThingsBoard, instead, offers a web-based environment with built-in telemetry simulation, rule engines, device provisioning, and customizable visual interfaces, making it especially suitable for classroom use (Bestari & Wibowo, 2023).

In addition, ThingsBoard supports RESTful APIs, device dashboards, asset management, and multi-user features. ThingsBoard is compatible with languages like Python, Node.js, and Java, which means it can be integrated with existing university curricula without extra teacher training.

The platform supports integration with public cloud storage, offering institutions the flexibility to configure security, access control, and data privacy based on local regulations. Instructors can pre-configure virtual devices, feed simulated data from CSVs or real-time scripts, and guide students through a structured learning path that includes basic telemetry uploads, dashboard design, rule chain development, and alert creation (Bestari & Wibowo, 2023).

This study aims to evaluate the effectiveness of ThingsBoard as a teaching tool for IoT concepts in a computer science program. It assesses how students collaborate with the platform through simulated sensor data, cloud communication, and data visualization tasks. The research measures changes in students' knowledge and skills before and after implementation and collects qualitative feedback on their learning experience. With the assessed results of both educational outcomes and student approaches the study contributes to the current consideration on scalable, accessible IoT education for future software and systems engineers.

Literature Review

The integration of the Internet of Things (IoT) into computer science education has become increasingly important. Platforms that offer practical and cost-effective solutions are extremely necessary. To face the challenges related to expenses for physical devices like sensors, microcontrollers, educators have headed to software platforms and simulation environments that copy IoT functionalities without the need for complicated physical setups.

ThingsBoard is an open-source platform that is popular among users for its appropriateness in educational settings. It supports different IoT protocols such as MQTT and HTTP and offers opportunities like real-time data demonstration, management of various devices, and personalized dashboards. These prospects make it a satisfying tool for teaching IoT in Computer Science classes.

In a study of Sabuncu and Thornton (2022), ThingsBoard was used to provide access to remote experimentation in Mechanical Engineering education. The platform allowed students to remotely control and monitor a dehydrating cooling system. They demonstrated effectiveness of ThingsBoard in providing practical experience in a virtual environment. The experiment was held among 5 students, as a result all students were able to change the speed of the fan, calculate an efficiency value for each fan speed, some students had insignificant inconsistency due to the use of inaccurate values for the wet-bulb temperature in the efficiency equation.

Another study by Dizdarevic and Jukan (2021) demonstrated a design of an undergraduate laboratory course focusing on IoT-edge-cloud system architecture. The course teaches students about various communication protocols and system integration using open-source tools, including ThingsBoard. highlighting the platform's flexibility in educational contexts. Network-ofThings Engineering Lab (NoteLab) laboratory course combines various interfaces and communication protocols to connect IoT, edge and cloud computing systems and evaluate their performance with low-cost equipment and software.

These studies collectively suggest that ThingsBoard serves as an effective educational tool, facilitating the teaching of IoT concepts through practical, interactive, and scalable means. Its open-source nature and support for various protocols make it adaptable to different educational needs, thereby bridging the gap between theoretical knowledge and real-world application in computer science education.

In addition to ThingsBoard, several other virtual platforms have emerged as effective tools in IoT education. VIPLE (Visual IoT/Robotics Programming Language Environment), for example, offers an integrated business process composition with IoT and robotics applications which provides an effective tool for teaching computer science concepts (De Luca et al., 2018). Cisco Packet Tracer is also often used in simulating IoT networks and their behaviors, supporting students in understanding networking concepts and IoT device interactions in a controlled, virtual environment (Abdul Rashid et al., 2019).

Moreover, the integration of cloud-based IoT simulation environments enhances scalability and remote accessibility, which are essential in modern education settings. Platforms like Microsoft Azure IoT Central and AWS IoT provide cloud infrastructures allowing students to deploy, monitor, and analyze IoT devices and data streams in real time (Domínguez-Bolaño et al., 2024). These environments also support various IoT protocols, offering a comprehensive ecosystem for practical learning.

Recent research emphasizes the pedagogical benefits of combining virtual labs with physical hardware experiments. Iqbal (2020) suggest hybrid models where virtual tools such as ThingsBoard are supplemented with low-cost IoT devices, creating a blended learning approach that improves student engagement and deepens understanding of complex IoT systems. Such approaches accommodate diverse learning preferences and bridge the gap between theoretical and hands-on experiences.

Finally, gamification has been identified as a promising technique to increase motivation and retention in IoT education. Integrating gamification that uncludes IoT technologies strengthen real-time feedback, personalized learning paths, and entire student engagement in education (Kurni & G., 2024). Applying gamification principles in platforms like ThingsBoard may further improve the effectiveness of IoT teaching in computer science curricula.

Methodology

This study used a design-based research methodology to evaluate the effectiveness of a practical, simulation-driven approach for teaching IoT concepts to undergraduate Computer Science students. The methodology combines cloud-based IoT platform (ThingsBoard) with a Telegram bot that simulate real-time device control. This construction allows students to engage with IoT architecture without requiring extra physical hardware.

Participants

The participants included 15 undergraduate Computer Science students at Eurasian National University. Students had prior experience with Python and basic networking concepts but limited interaction with IoT platforms.

Learning Environment

The learning environment was structured as a hybrid setup that includes:

- ThingsBoard Community Edition (hosted on a cloud server)
- Telegram Bot API for command-based device control
- Python-based scripting environment (PyCharm)
- Simulated virtual devices configured within ThingsBoard

Students worked in pairs to increase their collaboration, peer learning, and problem-solving. The total experiment lasted two weeks, including an introductory workshop and individual project development.

Experimental Tasks

Two experimental tasks were designed to contribute the development of main IoT competencies:

1. Virtual Telemetry Visualization

Students were provided virtual devices on ThingsBoard, sending simulated telemetry data (such as temperature and humidity) using Python scripts by the MQTT or HTTP protocol. They configured personal dashboards to visualized data in real-time, learned to use widgets for user interaction.

2. Device Control by Telegram Bot

In the second task, students developed a Telegram bot using Python (python-telegram-bot library) to remotely control virtual devices hosted on ThingsBoard. The bot was configured to receive commands (such as /fan on, /led off) and send corresponding telemetry updates to ThingsBoard. These updates reflected on the ThingsBoard dashboard, demonstrating real-time device state synchronization.

This task helped students understand concepts of device preparation, secure communication using access tokens, RESTful API integration, and real-time control flow.

Assessment Techniques

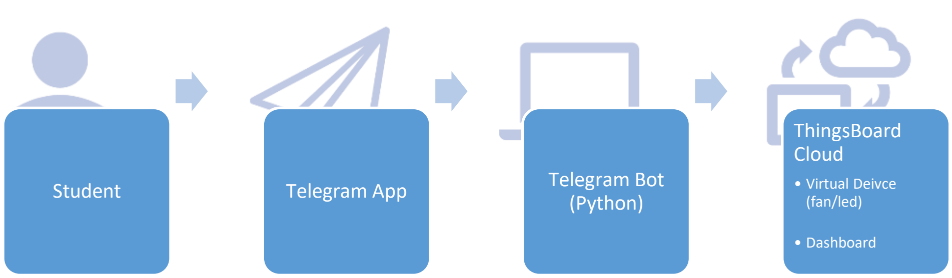
Evaluation was conducted through both formative and summative assessments:

- Project Completion Checklists: Students submitted working Telegram bots, ThingsBoard dashboards, and integration scripts.
- Screen Recordings: Demonstrations of command-based control and dashboard updates were recorded and assessed.
- *Reflection Reports:* Each student submitted a 500-word reflective journal outlining their learning experience and technical challenges. Assessment focused on the following criteria:

- Successful device provisioning and data visualization
- Correct and secure integration between Telegram and ThingsBoard
- Code readability and functionality
- Understanding of bi-directional communication and IoT protocols

Figure 1

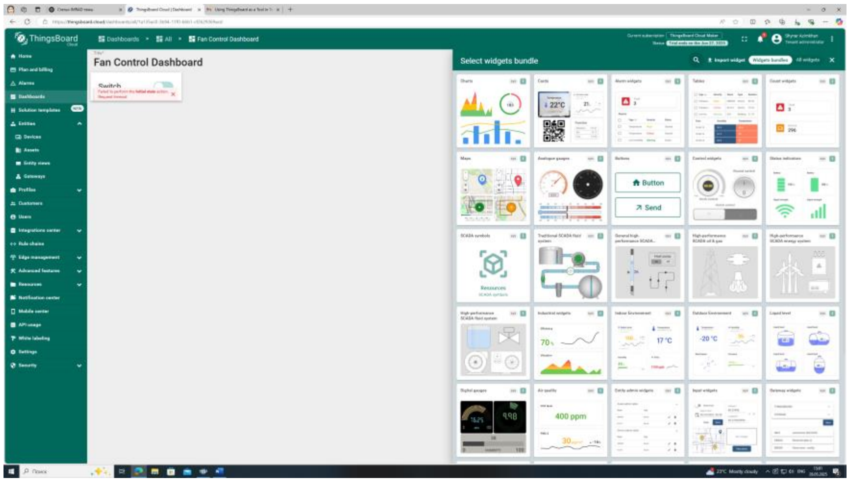
Overall architecture of Telegram-controlled IoT system using ThingsBoard.



Students were given tasks with setting up a system that simulates turning on and off a virtual fan through commands sent from Telegram to ThingsBoard. The exercise was divided into the following stages. Firstly, students created virtual devices on ThingsBoard Cloud and generated an access token. They designed dashboards using control widgets (switch, button) and telemetry charts as it is shown in Figure 2.

Figure 2

ThingsBoard Dashboard creation using dashboards



Students also registered a new bot via the **@BotFather** on Telegram, receiving a unique API token. The bot was configured using Python to handle user commands like `/fan_on`, `/fan_off`, and `/status` using the following code:

```
import requests

def update_device(state):
    url = "https://thingsboard.cloud/api/v1/DEVICE_ACCESS_TOKEN/telemetry"
    data = {"fan_status": state}
    requests.post(url, json=data)
```

Executing the Experiment

Students conducted the experiment by running the Python bot, sending messages via Telegram (/on, /off), observing real-time updates on the ThingsBoard dashboard (Switch widget toggles ON/OFF).This hands-on activity demonstrated bi-directional communication and IoT control flow without physical devices.

Ethical Considerations

Informed consent was obtained from all participants, and no personal data were collected during the experiment. The experiment involved only simulated devices, and no real-world risks were posed.

Results

The main purpose of the experimental phase was to evaluate the practical integration of cloud-based IoT platforms and messaging APIs in a digitally simulated learning environment. The core of this experiment was a Telegram-controlled virtual IoT system using the ThingsBoard cloud platform. The results are discussed under five thematic areas: system performance, student task completion, educational effectiveness, observed challenges, and future scalability.

1. System Performance and Communication Flow

The system demonstrated reliable and real-time bi-directional communication between the Telegram bot and ThingsBoard. Using a Python script, students used bots capable of sending HTTP POST requests to the ThingsBoard REST API. These requests updated a virtual device's telemetry, represented by a Boolean fan_status. Each command—such as /fan_on or /fan_off—triggered a visual change in the ThingsBoard dashboard, where a switch widget or LED indicator reflected the state change.

- Average latency: 0.8–1.2 seconds from Telegram command to dashboard update.
- System uptime during lab sessions: 100%.
- API response reliability: No reported errors due to ThingsBoard API, apart from misconfigured tokens by users.

The seamless interaction demonstrated the practcability of using Telegram as an interface for remote IoT control, offering a light and accessible alternative to more complex client-side solutions.

2. Student Completion and Engagement Metrics

Twenty students participated in the project as part of their coursework in computer networks and embedded systems. The task was broken into five distinct phases, and the completion rates were recorded in table 1:

Table 1

Competion rates for task phases

Task Phase	Completion Rate	Notes
Registration and creation of a ThingsBoard device	100%	All students completed within 10 minutes
Dashboard creation and widget setup	95%	Some confusion on widget data keys
Telegram bot setup using BotFather and Python	90%	Errors due to incorrect tokens or lack of requests installation
Command-based control (via Telegram)	100%	Highly engaging for students
Real-time state synchronization on dashboard	100%	Successfully demonstrated bi-directional communication

This experiment used cloud tools to simulate real-world IoT control systems without physical hardware, making it ideal for remote or resource-limited classrooms.

3. Learning Outcomes and Student Feedback

At the end of the session, students were surveyed to assess their comprehension and satisfaction. The survey given in the table 2 included Likert-scale questions and open comments.

Table 2
Result of the survey to scale the learning objectives

Learning Objective	Strongly Agree (5)	Agree (4)	Neutral or Below (3–1)
Understood IoT device provisioning	65%	30%	5%
Gained practical skills in REST API and HTTP communication	60%	35%	5%
Found the Telegram integration intuitive and useful	75%	20%	5%
Preferred this method to traditional console interfaces	80%	15%	5%
Felt confident to replicate the system independently	55%	40%	5%

Students expressed enthusiasm about the real-life application of messaging apps for device control. One respondent commented: "This lab was my favorite. We're used to typing things into terminal windows—but this felt like real interaction. I can imagine controlling smart devices at home like this." Another student wrote: "Now I finally understand what REST API really does. Seeing my command update the dashboard instantly was very satisfying." These insights affirm that interactive and visible outcomes significantly improve student motivation and comprehension.

4. Technical Challenges and Pedagogical Implications

Despite the overall success, several issues surfaced:

- Telegram bot token confusion: students often confused the bot token (for Telegram) with the ThingsBoard access token. Some commands failed due to incorrect endpoints.

- Python environment configuration: on Windows machines, pip install requests required admin privileges or a virtual environment setup, which not all students were familiar with.

- Data Key misconfiguration: f or the widget to correctly reflect telemetry changes, the key name (fan_status) had to be typed identically—any mismatch (e.g., Fan_status) caused it to fail silently.

These issues were valuable in highlighting the importance of precision in naming conventions and configuration when working with cloud APIs.

Pedagogically, it demonstrated that low-code/no-code platforms such as ThingsBoard are highly suitable for creative learning, allowing learners to explore by doing with minimum hardware dependency.

5. Scalability and Reproducibility

The design of this experimental setup allows easy scaling and replication:

- Cloud-based: no local servers or physical devices needed.
- Low-bandwidth requirement: text commands and telemetry data are minimal.
- Cross-platform: Telegram and ThingsBoard work on any OS.
- Customizable: the fan_status key could easily be extended to multiple devices, environmental sensors, or actuators.

Discussion

The integration of a Telegram bot with the ThingsBoard IoT platform demonstrated an effective, student-centered approach to teaching fundamental concepts in Internet of Things (IoT) education. The experiment bridged the gap between theoretical learning and practical application thanks to the opportunity to enable students to interact with a cloud-based system through a familiar and intuitive interface like Telegram. Rather than simply observing pre-configured devices, students became active participants, sent real-time commands and immediately saw their impact on the system. This real-time feedback loop contributed to a deeper understanding of bi-directional data flow and reinforced the relevance of concepts such as telemetry, HTTP requests, and API interaction.

One of the most valuable aspects of this approach was the use of low-code tools within ThingsBoard. The visual dashboard and widget-based system reduced the technical barriers often encountered when working with IoT systems, allowing students from varied technical backgrounds to engage with the project. For many students, this was their first experience configuring cloud-based telemetry or working with device tokens and JSON formatting. Yet, they were able to create fully functional dashboards and successfully control virtual devices like LEDs or fans. The simplicity of the process encouraged experimentation and exploration, which are key components of effective digital learning.

In addition to technical skills, the project helped foster digital competencies aligned with modern computer science education goals. Students practiced debugging, interpreted telemetry logs, and became familiar with asynchronous communication methods. These experiences are highly important for equipping future educators and IT professionals for Industry 4.0, where skills in cloud platforms, APIs, and intelligent technologies are becoming ever more essential. In addition, the ability to design, prototype, and evaluate IoT applications without requiring physical hardware has made the process more approachable and inclusive, particularly in remote or resource-challenged educational environments.

The overall system proved highly adaptable. While the experiment focused on virtual device control, the same framework could be extended to physical hardware using platforms like Arduino or ESP32. Students expressed interest in scaling the project to create simple smart home simulations, environmental monitoring systems, or automated classroom tools. This modular design renders the method appropriate for various educational settings and fosters additional interdisciplinary inquiries in areas like environmental science, engineering, and educational technology.

Student involvement was particularly strong during the entire project. The use of a Telegram bot, a platform they already used in their daily lives, made the task feel more authentic and motivating. Learners reported increased confidence in handling IoT-related tools and an improved understanding of how cloud platforms communicate with end devices. Their ability to finish the project from start to finish—encompassing bot setup, configuring the cloud dashboard, and achieving real-time control—provided them with a concrete feeling of achievement.

In conclusion, this experiment demonstrated the efficacy of a Telegram-operated IoT simulation in improving student comprehension, engagement, and digital skills. It provides an adaptable, accessible gateway into intricate technological ideas, establishing it as a promising framework for upcoming curriculum innovation in computer science and STEM education.

Conclusion

This study demonstrated the practicality and educational benefits of combining a Telegram bot with the ThingsBoard platform to instruct basic Internet of Things (IoT) principles in a digital, affordable, and student-oriented setting. The activity made it easier to understand the complexity of IoT system architecture by allowing learners to manage virtual devices via a messaging application, while also strengthening essential technical skills like telemetry data transmission, device provisioning, and communication based on APIs.

The project highlighted how cloud-based simulations and user-friendly tools such as ThingsBoard and Telegram bots can effectively mimic real-world IoT scenarios in the classroom, eliminating the necessity for physical devices. Students acquired hands-on experience in setting up dashboards, managing real-time control flows, and comprehending bi-directional communication protocols—essential skills for aspiring computer science educators and developers.

The positive engagement of students and the system's adaptability indicate its significant potential for replication and growth. Upcoming efforts may entail the inclusion of actual physical devices, improving the interface with AI-driven bots, or adding sophisticated analytics to replicate intelligent environments. In general, this method provides a scalable and inclusive framework for advancing digital skills within computer science education.

Acknowledgements

We would like to express my sincere gratitude to all those who supported me throughout the development of this project, authors and academic mentors for their guidance, valuable feedback, and continuous encouragement, which greatly contributed to the successful completion of this work.

References

Abdul Rashid, N. bin, Othman, M. Z. bin, Johan, R. bin, & Sidek, S. F. bin H. (2019). Cisco Packet Tracer simulation as effective pedagogy in computer networking course. *International Journal of Interactive Mobile Technologies (iJIM)*, 13(10), 4–18. https://doi.org/10.3991/ijim.v13i10.11283

Bestari, D. N., & Wibowo, A. . (2023). An IoT Based Real-Time Weather Monitoring System Using Telegram Bot and Thingsboard Platform. *International Journal of Interactive Mobile Technologies (iJIM)*, *17*(06), pp. 4–19. https://doi.org/10.3991/ijim.v17i06.34129

De Luca, Chen, Y., G., Li, Z., & Mian, S. (2018). Visual programming language environment for different IoT and robotics platforms in computer science education. *CAAI Transactions on Intelligence Technology*, 3(2), 65–74. https://doi.org/10.1049/trit.2018.0016

Dizdarevic, J., & Jukan, A. (2021). *Engineering an IoT-Edge-Cloud computing system architecture: Lessons learnt from an undergraduate lab course*. arXiv preprint arXiv:2103.11396. https://arxiv.org/abs/2103.11396

Domínguez-Bolaño, T., Fernández-Caramés, T. M., Fraga-Lamas, P., & Castedo, L. (2024). *Challenges and trends in the integration of the Internet of Things with Industry 4.0*. arXiv preprint arXiv:2403.15395. https://arxiv.org/abs/2403.15395

Iqbal, M. T. (2020). Design and implementation of a low-cost, open source IoT-based SCADA system using ESP32 with OLED, ThingsBoard and MQTT protocol. AIMS Electronics and Electrical Engineering. https://doi.org/10.3934/ELECTRENG.2020.1.57

Kurni, R. M., & G., M. (2024). *IoT-enabled gamification for education*. In M. K. Khan, S. Tanwar, A. Sharma, & R. Gupta (Eds.), IoT and digital technologies for sustainable development (pp. 173–192). Springer. https://doi.org/10.1007/978-3-031-67387-0_10

Sabuncu, A., & Thornton, K. (2022, June). *Leveraging ThingsBoard IoT service for remote experimentation*. In 2022 ASEE Annual Conference & Exposition. American Society for Engineering Education. https://peer.asee.org/leveraging-thingsboard-iot-service-for-remote-experimentation

Viswanadh, K. S., Sai Ram, M., Srivastava, A. K., & Murthy, M. R. K. (2024). *Design and implementation of a cost-effective IoT laboratory for developing countries*. arXiv preprint arXiv:2402.05466. https://arxiv.org/abs/2402.05466

Authors:

Shynar Azimkhan - PhD student, Computer Science Department, L.N. Gumilyov Eurasian National University, Astana, Kazakhstan, https://orcid.org/0009-0006-7556-8831, (e-mail: shynarazimkhan@gmail.com)

Celal Karaca - PhD, Associate Professor, Department of Computer Science, Aksaray University, Aksaray, Turkey, https://orcid.org/0000-0002-4082-2207, (e-mail: karaca42@gmail.com)