Перизат Шаймуратқызы Мәди – PhD докторы, «Электротехника және автоматтандыру» білім беру бағдарламалары тобының қауымдастырылған профессорының м.а., С. Сейфуллин атындағы Қазақ агротехникалық зерттеу университеті; e-mail: peri@mail.ru. ORCID: https://orcid.org/0000-0001-5930-8112.

Әділ Жандосұлы Қалиқан – 6B07108 «Процестер мен өндірістің автоматтандырылуы және энергия тиімділігі» мамандығы бойынша 1 курс студенті, С. Сейфуллин атындағы Қазақ агротехникалық зерттеу университеті; e-mail: adilespolov@gmail.com.

Ольга Валерьевна Гальцева – техникалық ғылымдар кандидаты, доцент, Бақылау және диагностика бөлімі: доцент, Томск политехникалық университеті; e-mail: tpuolga@tpu.ru. ORCID: https://orcid.org/0000-0001-6919-4833.

Information about the authors

Aiym Bakhtygalikyzy Abdykhamidova* – 2nd-year PhD student in «Control of Technical Systems», S. Seifullin Kazakh Agrotechnical Research University; e-mail: abdykhamidovaa@gmail.com. ORCID: https://orcid.org/0009-0006-3782-6668.

Yermek Zhaslanovich Sarsikeyev – PhD, Associate Professor, Associate Professor of the Educational Programs Group «Electrical Engineering and Automation», S. Seifullin Kazakh Agrotechnical Research University; e-mail: e.sarsikeev@kazatu.edu.kz. ORCID: https://orcid.org/0000-0002-7209-5024.

Perizat Shaimuratkyzy Madi – PhD, Acting Associate Professor of the Educational Programs Group «Electrical Engineering and Automation», S. Seifullin Kazakh Agrotechnical Research University; e-mail: peri@mail.ru. ORCID: https://orcid.org/0000-0001-5930-8112.

Adil Zhandosuly Kalikhan – 1st-year student on the 6B07108 course ‘Automation and Energy Efficiency of Processes and Production’, S. Seifullin Kazakh Agrotechnical Research University; e-mail: adilespolov@gmail.com.

Olga Valeryevna Galtseva – Candidate of Technical Sciences, Associate Professor, Department of Control and Diagnostics: Associate Professor, Tomsk Polytechnic University; e-mail: tpuolga@tpu.ru. ORCID: https://orcid.org/0000-0001-6919-4833.

Поступила в редакцию 06.04.2026
Поступила после доработки 20.05.2026
Принята к публикации 25.05.2026

https://doi.org/10.53360/2788-7995-2026-2(22)-24

IRSTI: 20.01.45

R.K. Kusainov[1], A.N. Aitkazin[2], D.B. Nurakhmetov[3], A.A. Aniyarov[3], A.K. Kenzhe[*]

[1]Shakarim University, 071404, Republic of Kazakhstan, Semey, st. Glinki, 20a

[2]College of Radio Engineering and Communications,
071401, Republic of Kazakhstan, Semey, Abay Street, 76

[3]Astana International University, 010000, Kazakhstan, Astana, Kabanbay Batyr Avenue, 8

[*]e-mail: yidgelek@gmail.com

DEVELOPMENT AND APPLICATION OF THE KEiT TRAINING STAND FOR PROGRAMMING ESP32 MICROCONTROLLERS IN PYTHON IN ENGINEERING EDUCATION

Abstract: *This paper presents a tutorial on programming the ESP32 microcontroller using MicroPython firmware and the Thonny development environment on the KEIT (Knowledge ESP32 Integration Trainer) training rig. The designed and manufactured rig, its operating principles, and a schematic diagram and printed circuit board (PCB) are provided. The MicroPython programming curriculum includes 12 lab sessions, ranging from basic GPIO control (LEDs, buttons, and buzzer) to integration with cloud services via the MQTT protocol. Sequentially completing the tasks allows students to master the fundamentals of working with digital and analog interfaces, learn to interact with sensors, displays, and actuators, and become familiar with the networking capabilities of the ESP32. Particular attention is paid to practical aspects, such as button debounce, working with interrupts, generating PWM signals, and measuring time intervals. The authors also present the specialized control software «KEIT Master Control», which complements the entire hardware and software*

system and reinforces the theoretical foundations of microcontroller programming in MicroPython. The final lessons focus on creating a simple web server on the ESP32 and exchanging data via an MQTT broker, providing students with a basic understanding of the principles of the Internet of Things. The course concludes with mini-projects («smart room», «parking assistant», and «Wi-Fi weather station») that integrate the technologies covered and demonstrate the potential of the ESP32 for automation and smart home systems. The curriculum can be used in universities and colleges for students majoring in Information Technology, Automation, Mechatronics and Radioelectronics.
Key words: *ESP32, MicroPython, KEiT (Knowledge ESP32 Integration Trainer), Thonny IDE, Internet of Things (IoT).*

Introduction

The modern development of digital technologies and the Internet of Things (IoT) requires training specialists capable of working with microcontrollers and embedded systems. Traditionally, microcontroller programming was conducted in assembly languages or C/C++, which provided high efficiency and low-level control, but required significant knowledge of processor architecture, registers, and memory management. For beginners, this entry barrier is often too high. In recent years, many countries, including Kazakhstan, have increasingly prioritized Python in educational programs. This is due to its simplicity, readability, and widespread popularity among students and young professionals. Using MicroPython in microcontrollers, such as the ESP32, allows us to extend the familiar Python syntax to embedded systems while maintaining access to key hardware capabilities-digital pins, I²C, SPI, and UART interfaces, sensor support, and network connectivity [1, 2]. This approach has the following key advantages:

– lower entry barrier: students don't need in-depth knowledge of microcontroller architecture or low-level programming languages to begin controlling devices;

– a unified educational ecosystem: Python is used in data analysis, machine learning, and automation. The ability to control the system with MicroPython makes it possible to apply acquired knowledge to digital electronics.

Thus, using Python and MicroPython to teach microcontrollers in the classroom allows for a more accessible and modern educational process [3-5]. This helps prepare specialists prepared to work with smart home, IoT, and digital automation technologies, and increases the attractiveness of engineering and IT fields among students.

Materials and methods

Hardware architecture of the KEIT training system. The ESP32 microcontroller was selected as the computing core of the KEIT training system due to its comprehensive technical and functional characteristics. This microcontroller is equipped with integrated Wi-Fi and Bluetooth wireless modules, allowing it to be used for studying network technologies and IoT systems without the need for external communication modules. Furthermore, the ESP32 boasts a high clock rate and sufficient RAM and flash memory, enabling the implementation of both simple training examples and more complex applied tasks. A significant factor in this choice was official support for MicroPython firmware, enabling the use of the high-level Python programming language when working with the microcontroller's hardware [6]. The widespread use and availability of the ESP32 make it a suitable solution for widespread use in educational institutions.

The KEIT training system's hardware is complemented by a set of peripheral modules that support laboratory and practical exercises of varying complexity. The setup includes a 2004-format liquid crystal display connected via an I²C interface and designed to display text and service information. A DHT11 temperature and humidity sensor [7, 8] is used to study the principles of operating environmental sensors. The setup also incorporates actuators such as LEDs, a buzzer, and controlled loads, enabling the development of digital output control methods and pulse-width modulation (PWM) signal generation. Standard UART and I²C interfaces enable the study of data exchange protocols and the organization of microcontroller interaction with external devices, including a personal computer. Figure 1 shows the setup's printed circuit board.

When designing the setup, special attention was paid to the power supply system and protection of its components. The design and circuitry are aimed at reducing the risk of component damage due to improper connections or errors during laboratory work. This ensures safe operation of the setup in classrooms and allows it to be used for teaching students of various skill levels, including beginners.

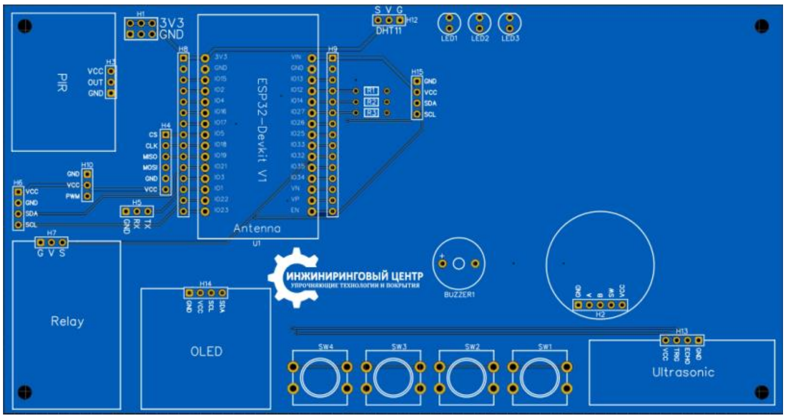

Figure 1 – Layout of the printed circuit board of the training stand

Microcontroller Software Implementation. The KEIT training rig's software implementation is based on MicroPython firmware, an interpreted runtime environment for microcontrollers. MicroPython was chosen due to its ability to quickly load and modify program code without the need for time-consuming compilation and reflashing of the device, significantly accelerating the training and debugging process. Sample programs for the rig provide programmatic access to all the main hardware resources of the ESP32 microcontroller, including digital and analog inputs and outputs, pulse-width modulation modules, timers, and hardware interrupts. The use of standard MicroPython libraries provides a unified and intuitive syntax for controlling peripheral devices, which is especially important when training students and novice developers [9]. During laboratory work, students gain practical skills in working with buttons, LEDs, actuators, and sensors, and also learn methods for contact debounce and measuring time intervals. A key feature of the setup's software implementation is support for the microcontroller's communication interfaces and network capabilities. The software supports UART and I²C interfaces, enabling data exchange with a personal computer and peripheral modules. For remote monitoring, the setup utilizes a built-in web server, displaying current device parameters and status in real time via a standard web browser. This mechanism clearly demonstrates the principles of building network applications and device interaction within the Internet of Things (IoT) concept [10, 11].

A UART interface is used to communicate between the microcontroller and the control software on the personal computer. The microcontroller's software and circuit architecture is modular, simplifying the expansion of the setup's functionality and enabling its use as a basic platform for custom and project-based tasks.

KEIT Master Control Control Software. To expand the functionality of the KEIT training system and ensure convenient user interaction with the hardware, specialized control software, KEIT Master Control (Figure 2), was developed for use on a personal computer. This software acts as a link between the ESP32 microcontroller and the user, providing data visualization, control of actuators, and monitoring of the system's operation during laboratory and practical exercises [12].

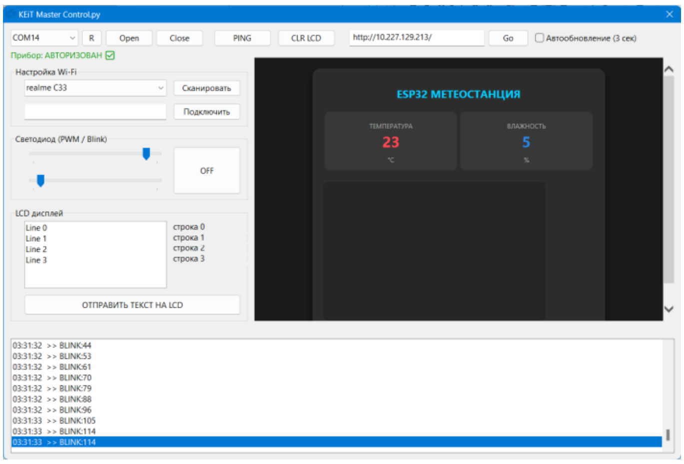

Figure 2 – The «KEIT Master Control» program window

PureBasic was chosen as the control software development environment due to its high performance, compact size, and the ability to create cross-platform desktop applications. Using PureBasic allows for the creation of executable files that are compact and require no additional libraries or dependencies, which is especially important when using software in classrooms and computer labs. Furthermore, this environment provides convenient tools for working with serial interfaces and a graphical user interface.

The KEIT Master Control software provides visual monitoring of parameters received from the microcontroller, including data from temperature and humidity sensors, as well as the status of the test bench's actuators. Through the graphical interface, the user can interactively control the device's operating modes, change pulse-width modulation parameters, turn loads on and off, and send control commands via the UART interface. This approach allows for a clear demonstration of the principles of data exchange between the system's software and hardware components.

To enhance information security and prevent the use of unauthorized equipment, the software implements a hardware identification system. Access to the test bench's control functions is based on verification of unique device data, which creates a list of authorized hardware identifiers (WhiteList). This mechanism allows the test bench to be used in training and examination modes and reduces the risk of incorrect connection of third-party devices.

An additional feature of the software is integration with the microcontroller's web interfaces. The KEIT Master Control software includes a built-in browser that displays graphs and data generated by the ESP32's built-in web server. This allows for the unification of local control and network monitoring within a single software environment, helping students develop a holistic understanding of the interaction between desktop and embedded applications.

These features demonstrate that the KEIT Master Control software complements the KEIT hardware and software, enhances the clarity of the learning process, and expands the range of educational tasks, including the study of data exchange protocols, human-machine interface principles, and the fundamentals of information security in automation systems.

Methodological Value and Application Scenarios. The KEIT training rig offers high methodological value and focuses on a practice-oriented approach to teaching microcontroller and embedded systems programming. The rig's architecture and the MicroPython software platform allow for a step-by-step learning process, starting with basic digital input and output control operations and gradually moving on to more complex tasks related to data processing, networking, and integration into IoT systems.

The KEIT training course includes a series of lab exercises covering key aspects of microcontroller systems. Initially, students learn to control LEDs, buttons, and a buzzer, as well as the principles of timers and pulse-width modulation. Subsequent stages focus on working with sensors, signal processing, and displaying information on an LCD display, which develops students' understanding of the interaction between software code and physical processes [13-15].

The networking capabilities of the ESP32 microcontroller play a key role in the learning process. Using built-in wireless modules, students can explore issues of scanning Wi-Fi networks, establishing network connections, and transmitting data via various protocols. Implementing an embedded web server and messaging via an MQTT broker clearly demonstrates the principles of building distributed systems and provides students with a basic understanding of the Internet of Things [16-18]. Using scanning Wi-Fi networks as an example, we will explain the process as follows.

```
import network
import binascii
# Инициализация Wi-Fi интерфейса в режиме станции (клиента)
wlan = network.WLAN(network.STA_IF)
wlan.active(True)
print("Сканирование доступных сетей...")
# Выполнение сканирования
# Возвращает список кортежей: (ssid, bssid, channel, RSSI, authmode, hidden)
networks = wlan.scan()
print(f"Найдено сетей: {len(networks)}\n")
print(f"{'SSID':<25} {'BSSID':<20} {'Канал':<10} {'RSSI':<10}")
print("-" * 65)
for net in networks:
```

```
    ssid = net[0].decode('utf-8') if net[0] else "Hidden SSID"
    bssid = binascii.hexlify(net[1]).decode()
    channel = net[2]
    rssi = net[3]
    print(f"{ssid:<25} {bssid:<20} {channel:<10} {rssi:<10}")
```

MicroPython uses the basic network module to work with wireless interfaces. The process of obtaining a list of available access points can be divided into three key stages: initialization, activation, and data processing. When preparing a network interface, the ESP32 microcontroller can operate in several modes. In this case, Station mode (STA_IF) is used for scanning, in which the device acts as a client.

```
wlan = network.WLAN(network.STA_IF) # Создание объекта интерфейса
wlan.active(True)                   # Программное включение радиомодуля
```

Activating the module (active(True)) supplies power to the Wi-Fi chip. It's important to remember that an enabled radio module significantly increases the device's power consumption, which is critical when developing standalone IoT sensors.

The wlan.scan() method causes the controller to sequentially switch between radio channels and listen for beacon frames from access points. The function returns a list of tuples. Each tuple contains technical information about the detected network. Particular attention should be paid to the RSSI (Received Signal Strength Indicator) parameter during training-this value is measured in decibel-milliwatts (dBm). Data received from the module is received in raw form (bytes), which requires conversion for display to the user. The data is listed in Table 1.

Table 1 – Data received from the module

Parameter	Description	Data Type in MicroPython
SSID	Access point name	bytes (requires decode())
Channel	Radio channel	($1-13$)int
RSSI	Signal strength	int (negative number)
Authmode	Security protocol	int (constant 0-5)
BSSID	Physical address (MAC) of the access point	Converted to hexadecimal for readability

The final stage of training consists of mini-projects focused on integrating previously learned technologies and programming methods. By completing projects such as a «smart room», «parking assistant», or «Wi-Fi weather station», students consolidate their skills with hardware interfaces, network protocols, and user interfaces, and gain experience designing complete application systems. Compared to traditional Arduino-compatible platforms, the KEIT platform facilitates faster learning through the use of the Python language and a unified educational ecosystem. This lowers the entry barrier to the subject, increases student motivation, and fosters sustainable practical competencies in demand in the modern fields of automation, digital electronics, and the IoT.

Conclusion. The KEIT training system's methodological value lies in its practice-oriented approach, allowing students to progress from basic I/O port control to the creation of complex distributed systems. Using MicroPython in combination with the ESP32 architecture significantly lowers the learning curve, allowing students to focus on algorithmic logic and interface principles rather than low-level compilation complexities.

Key educational outcomes of the program:

– Mastering hardware interfaces: Students gain solid skills in working with digital and analog signals, PWM, interrupts, and communication protocols (I2C, UART).

– Learning networking technologies and IoT: Using the integrated Wi-Fi module, students practice working with a wireless network:

1. Initialization: Activating the radio module in client mode (STA_IF).
2. Scanning: Analyzing the airwaves and obtaining access point parameters, including signal strength (RSSI, measured in dBmd).
3. Interaction: Implementing web servers and exchanging data via the MQTT protocol.

– System Integration: Final mini-projects, such as «Smart Room», «Parking Assistant», and «Wi-Fi Weather Station», allow students to combine disparate knowledge into a single application system.

The KEIT booth is a ready-made educational ecosystem that provides a faster learning curve than traditional Arduino platforms. This helps students develop relevant competencies in automation, digital electronics, and the Internet of Things, which are in demand in today's labor market.

References

1. Bell C. MicroPython for the Internet of Things. Apress: New York, NY, USA. – 2017.
2. Gololobov V.N. Where do robots begin? About the Arduino project for schoolchildren (and more) [S chego nachinayutsya roboty? O proekte Arduino dlya shkol'nikov (i ne tol'ko)]. BHV-Petersburg. – 2013.
3. An IoT-Based Solution for Monitoring a Fleet of Educational Buildings Focusing on Energy Efficiency / D. Amaxilatis et al // Sensors. – 2017. – № 17(10). – P. 2296. https://doi.org/10.3390/s17102296.
4. Programming of ATmega8 microcontrollers in C language: Methodological manual [C tilinde ATmega8 mikrokontrollerlerin bagdarlamalaý: Ádistemelik qural] / R.K. Kusainov et al // Shakarim University. – 2021.
5. Sánchez-Gómez J.S., & Cajiao M.C.R. Design of a Python programming course by big ideas with a STEM approach. https://doi.org/10.18687/LACCEI2024.1.1.361.
6. Espressif Systems. 2024. ESP32 technical reference manual (Version 5.2). www.espressif.com.
7. Oner V.O. 2021. Developing IoT Projects with ESP32: Automate your home or business with inexpensive Wi-Fi devices. Packt Publishing Ltd.
8. IEEE 802 LAN/MAN Standards Committee. 2009. IEEE Standard for Information technology-Telecommunication and information exchange between systems-Local and metropolitan area networks-Specific requirements Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment1: Radio Resource Measurement of Wireless LANs. http://standards. ieee. org/getieee802/download/802.11 n-2009. pdf .
9. MicroPython Group. 2024. MicroPython documentation: Quick reference for the ESP32. docs.micropython.org
10. Ionescu V.M. Investigating the performance of MicroPython and C on ESP32 and STM32 microcontrollers / V.M. Ionescu, F.M. Enescu // In 2020 IEEE 26th International Symposium for Design and Technology in Electronic Packaging (SIITME). – 2020. – P. 234-237. https://doi.org/10.1109/SIITME50350.2020.9292199.
11. Thonny IDE Development Team. 2024. Thonny: Python IDE for beginners (Version 4.1). thonny.org
12. PureBasic Support. 2023. PureBasic language reference. www.purebasic.com
13. Somantri Y. Low Cost, Advanced, Integrated Microcontroller Training Kit. / Y. Somantri, I. Fushshilat // In IOP Conference Series: Materials Science and Engineering. – 2020. – Vol. 180, № 1. – P. 012107. IOP Publishing. https://doi.org/10.1088/1757-899X/180/1/012107.
14. Habibi M.W. Impact of Training Kit-Based Internet of Things to Learn Microcontroller Viewed in Cognitive Domain / M.W. Habibi, I.G.P.A. Buditjahjanto // TEM Journal. – 2024. – № 13(2). – P. 1157.
15. Development of a Microcontroller Training Kit to Increase Student Learning Motivation / F. Eliza et al // Engineering, Technology & Applied Science Research. – 2025. – № 15(3). – P. 22319-22326. https://doi.org/10.48084/etasr.10142.
16. MicroPython PyBoard for IoT / J.S. Parab et al // In Python Programming Recipes for IoT Applications. – 2023. – P. 89-122. Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-19-9466-1_4.
17. Standard O.A.S.I.S. 2019. MQTT Version 5.0. Retrieved June, 22(2020), 1435.
18. Ni Z. A sensing system based on public cloud to monitor indoor environment of historic buildings / Z. Ni, Y. Liu, M. Karlsson, S. Gong // Sensors. – 2021. – № 21(16). – P. 5266. https://doi.org/10.3390/s21165266.

Р.К. Кусаинов[1], А.Н. Айтқазин[2], Д.Б.Нұрахметов[3], А.А. Анияров[3], А.К. Кенже[1]*
[1]Шәкәрім Университеті,
071404, Қазақстан Республикасы, Семей қаласы, Глинка көшесі, 20а
[2]Радиотехника және байланыс колледжі,
071401, Қазақстан Республикасы, Семей қаласы, Абай көшесі, 76
[3]Халықаралық Астана университеті,
010000, Қазақстан, Астана қ., Қабанбай батыр даңғылы, 8
*e-mail: yidgelek@gmail.com

ИНЖЕНЕРЛІК БІЛІМ БЕРУДЕ PYTHON ТІЛІНДЕ ESP32 МИКРОКОНТРОЛЛЕРЛЕРІН БАҒДАРЛАМАЛАУҒА АРНАЛҒАН KEIT ОҚУ-ЖАТТЫҒУ СТЕНДІН ӘЗІРЛЕУ ЖӘНЕ ҚОЛДАНУ

Жұмыста KEIT (Knowledge ESP32 Integration Trainer) оқу-жаттығу стендінде MICROPYTHON микробағдарламасын және Thonny әзірлеу ортасын пайдалана отырып, ESP32 микроконтроллерін бағдарламалауды оқыту процесі ұсынылған. Әзірленген және өндірілген стенд пен жұмыс принципі толықтай сипатталған және электр схемасы мен платасы берілген. MicroPython бағдарламалауды оқыту жоспары негізгі GPIO басқарудан (жарықдиодты шамдар, түймелер, буззер) бастап MQTT протоколы арқылы бұлттық қызметтерге біріктіруге дейінгі 12 зертханалық сабақты қамтиды. Тапсырмаларды дәйекті орындау студенттерге сандық және аналогтық интерфейстермен жұмыс істеу негіздерін меңгеруге, сенсорлармен, дисплейлермен және атқарушы құрылғылармен өзара әрекеттесуді үйренуге және ESP32 желілік мүмкіндіктерімен танысуға мүмкіндік береді. Практикалық аспектілерге ерекше назар аударылады: түймелерді бұруға қарсы, үзілістермен жұмыс, PWM сигналдарын генерациялау, уақыт аралықтарын өлшеу. Сондай-ақ, авторлар әзірлеген «KEIT master Control» арнайы басқару бағдарламалық жасақтамасы ұсынылған, ол бүкіл аппараттық-бағдарламалық кешеннің қосымша буыны болып табылады және MicroPython-да микроконтроллерді бағдарламалау негіздерінің теориялық дамуын бекітеді. Аяқталатын сабақтар ESP32-де қарапайым веб-сервер құруға және MQTT брокері арқылы мәліметтер алмасуға арналған, бұл студенттерде заттар интернетінің принциптері туралы негізгі түсінікті қалыптастырады. Ұсынылған курс үйренген технологияларды біріктіретін және автоматтандыру жүйелері мен смарт үй үшін ESP32 қолдану мүмкіндіктерін көрсететін шағын жобалармен («ақылды бөлме», «тұрақ көмекшісі», «Wi-Fi метеостанциясы») аяқталады. Оқу жоспарын жоғары оқу орындары мен колледждерде «Ақпараттық технологиялар», «Автоматтандыру», «Мехатроника» және «Радиоэлектроника» бағыттарындағы студенттерді оқыту кезінде пайдалануға болады.

Түйін сөздер: *ESP32, MicroPython, KEIT (knowledge ESP32 integration Trainer), Thonny ide, Заттар интернеті (IoT).*

Р.К. Кусаинов[1], А.Н. Айтқазин[2], Д.Б.Нұрахметов[3], А.А. Анияров[3], А.К. Кенже[1]*
[1]Шәкәрім университет,
071404, Республика Казахстан, г. Семей, ул. Глинки, 20а
[2]Колледж радиотехники и связи,
071401, Республика Казахстан,г. Семей, ул. Абая, 76
[3]Международный университет Астаны,
010000, Казахстан, г. Астана, проспект Кабанбай Батыра, 8
*e-mail: yidgelek@gmail.com

РАЗРАБОТКА И ПРИМЕНЕНИЕ УЧЕБНО-ТРЕНИРОВОЧНОГО СТЕНДА КЕіТ ДЛЯ ПРОГРАММИРОВАНИЯ МИКРОКОНТРОЛЛЕРОВ ESP32 НА ЯЗЫКЕ PYTHON В ИНЖЕНЕРНОМ ОБРАЗОВАНИИ

В работе представлен процесс обучения программированию микроконтроллера ESP32 с использованием прошивки MicroPython и среды разработки Thonny на учебно-тренировочном стенде KEIT (Knowledge ESP32 Integration Trainer). Подробно описан разработанный и изготовленный стенд и принцип работы и приведены принципиальная электрическая схема и печатная плата. План обучения по программированию в MicroPython включает 12 лабораторных занятий, начиная с базового управления GPIO (светодиоды, кнопки, буззер) и заканчивая интеграцией в облачные сервисы по протоколу MQTT. Последовательное выполнение заданий позволяет студентам освоить основы работы с цифровыми и аналоговыми интерфейсами, научиться взаимодействовать с датчиками, дисплеями и исполнительными устройствами, а также познакомиться с сетевыми возможностями ESP32. Особое внимание уделено практическим аспектам: антидребезг кнопок, работа с прерываниями, генерация ШИМ-сигналов, измерение временных интервалов. Также представлена разработанная авторами специальное управляющее *программное обеспечение «KEIT Master Control», которое является дополняющим звеном всего аппаратно-программного комплекса и закрепляющий теоретическое освоение основ программирования микроконтроллера в MicroPython. Завершающие занятия посвящены созданию простого веб-сервера на ESP32 и обмену данными через MQTT-брокер, что формирует у студентов базовое понимание принципов Интернета вещей. Предложенный курс завершается мини-проектами («умная комната», «парковочный помощник», «Wi-Fi метеостанция»), которые интегрируют изученные технологии и демонстрируют возможности применения ESP32 для систем автоматизации и умного дома. Учебный план может использоваться в вузах и колледжах при обучении студентов направлений «Информационные технологии», «Автоматизация», «Мехатроника» и «Радиоэлектроника».*

Ключевые слова: *ESP32, MicroPython, KEIT (Knowledge ESP32 Integration Trainer), Thonny IDE, Интернет вещей (IoT).*

Information about authors

Rinat Kenzheevich Kusainov – Master of Technical Sciences, Head of the Engineering Center, Shakarim University, Semey, Kazakhstan; e-mail: rinat.kus@mail.ru. ORCID: https://orcid.org/0000-0001-5166-4761.

Ayan Nurtaevich Aitkazin – student, College of Radio Engineering and Communications, Semey, Kazakhstan; e-mail: ayan_aitkazin@mail.ru.

Daulet Bagdatovich Nurakhmetov – PhD, Associate Professor, Astana International University, Astana, Kazakhstan; e-mail: nurakhmetov@math.kz. ORCID: https://orcid.org/0000-0003-2358-9092.

Almir Askarovich Aniyarov – candidate of physical and mathematical sciences, associate professor, Astana International University, Astana, Kazakhstan; e-mail: almir.aniyarov@aiu.edu.kz. ORCID: https://orcid.org/0000-0003-4786-1556.

Aruzhan Kenzhekyzy Kenzhe* – student of group DYa-2502, Shakarim University, Semey, Kazakhstan; e-mail: yidgelek@gmail.com.

Авторлар туралы мәлімет

Ринат Кенжеевич Кусаинов – физика магистрі, инжинирингтік нентра басшысы, Шәкәрім университеті, Қазақстан; e-mail: rinat.kus@mail.ru. ORCID: https://orcid.org/0000-0001-5166-4761.

Аян Нұртайұлы Айтқазин – студент, Радиотехника және байланыс колледжі, Қазақстан; e-mail: ayan_aitkazin@mail.ru.

Дәулет Бағдатұлы Нұрахметов – философия докторы (PhD), қауымдастырылған профессор, Астана халықаралық университеті, Астана қ., Қазақстан; e-mail: nurakhmetov@math.kz. ORCID: https://orcid.org/0000-0003-2358-9092.

Альмир Аскарович Анияров – физика-математика ғылымдарының кандидаты, қауымдастырылған профессор, Астана халықаралық университеті, Қазақстан; e-mail: almir.aniyarov@aiu.edu.kz. ORCID: https://orcid.org/0000-0003-4786-1556.

Аружан Кенжекызы Кенже* – ДЯ-2502 студенттік тобы, Шәкәрім университеті, Қазақстан; e-mail: yidgelek@gmail.com.

Информация об авторах

Ринат Кенжеевич Кусаинов – магистр физики, руководитель Инжинирингового Центра, Шәкәрім университет, г. Семей, Казахстан; e-mail: rinat.kus@mail.ru. ORCID: https://orcid.org/0000-0001-5166-4761.

Аян Нуртаевич Айтқазин – студент, Колледж радиотехники и связи, г. Семей, Казахстан; e-mail: ayan_aitkazin@mail.ru.

Даулет Багдатович Нұрахметов – доктор философии (PhD), ассоциированный профессор, Международный университет Астана, Казахстан; e-mail: nurakhmetov@math.kz. ORCID: https://orcid.org/0000-0003-2358-9092.

Альмир Аскарович Анияров – кандидат физико-математических наук, ассоциированный профессор, Международный университет Астана, Казахстан; e-mail: almir.aniyarov@aiu.edu.kz. ORCID: https://orcid.org/0000-0003-4786-1556.

Аружан Кенжекызы Кенже* – студент группы ДЯ-2502, Шәкәрім университет, г. Семей, Казахстан; e-mail: yidgelek@gmail.com.

Received 12.04.2026
Revised 25.05.2026
Accepted 29.05.2026