

## **SETTING UP A LOCAL WIRELESS NETWORK FOR IoT DEVICES USING THE ESP–NOW PROTOCOL**

In today's world, the rapid development of the Internet of Things (IoT) concept requires the creation of reliable, energy-efficient, and high-speed communication systems between various smart devices. Traditional wireless networks based on the Wi-Fi standard, although they provide high bandwidth, typically require a central router or access point. In cases where network infrastructure is absent, or when deploying a full-fledged Wi-Fi network is impractical, there is a need for alternative data transmission methods. In addition, the classic Wi-Fi protocol requires significant energy consumption and time to establish a connection (handshake), which is a critical drawback for battery-powered devices.

One of the most promising solutions for local communication between microcontrollers is the use of the ESP–NOW[1] wireless protocol, developed by Espressif Systems. This is a connectionless protocol that allows devices to exchange short messages directly with one another. It is supported by the ESP8266, ESP32, ESP32-S, and ESP32-C series of microcontrollers and can operate in parallel with Wi-Fi and Bluetooth LE.

The aim of this work is to develop, theoretically justify, and experimentally verify a decentralized wireless sensor network based on ESP32 microcontrollers using the ESP-NOW protocol to ensure instantaneous data transmission with minimal power consumption.

A key feature of the ESP–NOW[2] architecture is that this protocol operates at the data-link layer, reducing the five layers of the classic OSI network model to a single layer. As a result, data does not need to pass through the network, transport, session, presentation, and application layers. The absence of the need to create and process complex packet headers at each of these stages ensures ultra-fast system response. Transmission latency is measured in milliseconds, which completely eliminates packet loss issues in congested networks.

The protocol allows for the transmission of a payload of up to 250 bytes in a single packet. Data transmission security is supported by the integrated ECDH and AES cryptographic algorithms. ESP–NOW allows for the combination of both encrypted and unencrypted nodes within a single network. However, there is a hardware limitation: in standard mode (Station), the microcontroller supports up to 10 encrypted peer nodes, and the total number of unencrypted nodes must not exceed 20.

An ESP-NOW-based network is highly flexible and allows for the implementation of various interaction topologies:

1. Point-to-Point connection – the simplest method of direct data exchange between two microcontrollers.
2. “One-to-Many” topology – a single master device sends commands to several slave devices. This architecture is ideal for creating remote controls.

3. Many-to-One topology – a configuration in which several independent sensor nodes collect information (such as temperature, humidity, or data from PIR sensors) and synchronously send it to a single central gateway receiver.
4. Many-to-Many communication – each network node functions simultaneously as both a receiver and a transmitter, forming a complex structure for a smart home.

The hardware implementation of the system is based on the use of microcontrollers from the ESP32 family. The software is developed in the Arduino IDE (using the `<esp_now.h>` library) or in the MicroPython language (using the `network` and `espnow` modules). To establish a direct connection between the boards, their unique MAC addresses must be determined in advance.

The operating algorithm of the transmitting node involves several stages. First, the module is initialized in Wi-Fi station mode (`WIFI_STA`), after which the base function `esp_now_init()` is called. Next, using `esp_now_add_peer()`, information about the receiving device – including its MAC address and wireless channel – is added to the system.

A key advantage of ESP-NOW is its callback mechanism. On the sender's side, the `esp_now_register_send_cb()` function is registered, which automatically triggers immediately after the packet is sent and returns the delivery status (success or failure). On the receiver side, `esp_now_register_recv_cb()` is used, which instantly responds to the arrival of a signal, decrypting the data into a pre-prepared structure (`struct`) using the `memcpy()` function. The use of structures allows combining different data types (`char`, `int`, `float`, `bool`) into a single compact packet.

Practical testing of the “many-to-one” network topology demonstrated high stability. The sensor nodes are configured to operate in Deep Sleep mode, which reduces power consumption to microamperes. Waking up on a timer, the microcontroller instantly sends data via ESP-NOW[3] and goes back to sleep. The absence of a lengthy connection process to the router allows the module to complete the entire cycle in a matter of milliseconds. As a result, power from a single 18650 lithium-ion battery or even a CR2032 battery can keep the device running for several years. In open terrain, the stable signal transmission range exceeds 200 meters.

Thus, the research results confirm that the ESP-NOW protocol is an extremely efficient, reliable, and cost-effective tool for developing local IoT networks. Reducing the layers of the OSI network model ensures minimal latency and excellent energy efficiency. The developed hardware-software architecture can serve as a basis for designing security systems, industrial automation, and smart home solutions.

## REFERENCES

1. ESP-NOW Wireless Communication Protocol. *Espressif Systems*. URL: <https://www.espressif.com/en/solutions/low-power-solutions/esp-now>.
  2. Getting Started with ESP-NOW (ESP32 with Arduino IDE) | Random Nerd Tutorials. *Random Nerd Tutorials*. URL: <https://randomnerdtutorials.com/esp-now-esp32-arduino-ide/>.
- MicroPython: ESP-NOW with ESP32 - Receive Data (many-to-one) | Random Nerd Tutorials. URL: <https://randomnerdtutorials.com/micropython-esp-now-esp32-many-to-one/>.