WIZNET MAGAZINE

WIZ Mag

Hybrid Network Edition



WHY CABLE?

Edge AI

ESP X W5500

Cable or Not Cable? - Why We Still Plug In

ESP at the Core, WIZnet on the Wire - Tutorial Roadmap

Frames Never Die - Espressif Camera & PoE Stories

24/7 Life - Audio · Robot · Gateway Case Studies

Boards, Bricks & Cables - Espressif x WIZnet Hardware Map

maker.wiznet.io

WIZnet



1 Cable or Not Cable?

Why We Still Plug In

1 Everyone starts with **wireless**

From IoT and smart buildings to factory automation and multi-room audio, most ESP32 projects today “start with Wi-Fi.” Developers drop the SSID and password into the firmware and run a quick test. Field engineers bring up a temporary AP and do the first commissioning. Even early-stage B2B PoC projects often report success as soon as “it’s up over wireless.”
For idea validation, demos, and temporary installs, nothing beats this.Wi-Fi has already become the standard “**tool that removes spatial constraints**” for experiments.
But at some point you have to ask yourself:
“**When this ships as a real service, a real product - are we really okay leaving it like this?**”

2 The moment when **“good enough”** isn’t

In some systems, even a brief dropout immediately turns into a fault or a loss.If a ROS2 robot misses a control packet, a 24/7 Bitcoin miner loses its pool connection, audio streaming in a hotel or commercial space keeps stuttering, or a remote monitoring camera drops frames at the exact moment of an event, users don’t say, “Well, that’s Wi-Fi for you.”
It shows up instantly as broken SLAs, lower uptime, and higher maintenance costs.
That’s usually when teams make the same decision:
“Keep wireless where it is, but pull a cable for the parts that absolutely must not fail.”
And when that happens, there’s a chip engineers all over the world tend to place right next to their Espressif device: **WIZnet Ethernet.**

3 The Hidden Standard: why engineers worldwide choose **WIZnet Ethernet**

Search GitHub for “ESP32” and “Ethernet” and you’ll notice a pattern.From DIY projects to commercial prototypes, an overwhelming share of SPI-based designs-where pin count really matters-use WIZnet Ethernet chips such as the W5500.
WIZnet is the brand first supported by Arduino’s standard Ethernet library,and its chips are also built into Espressif’s official framework (ESP-IDF) as built-in drivers.
Among all the available solutions, there’s a clear technical reason WIZnet Ethernet has become the **de facto standard: WIZnet’s hardware TCP/IP offload engine.**
The ESP32 is powerful, but running Wi-Fi and a full software network stack at the same time eats a lot of CPU cycles and RAM.WIZnet chips offload the complex networking work into dedicated hardware.
In other words, WIZnet takes responsibility for the network traffic,and the ESP32 is freed up to focus 100% on your application - robot control, AI inference, and everything else that actually differentiates your product.

4 ESP at the core, **WIZnet on the wire**

This clean partitioning of roles is the core idea running through this entire issue.
Robots / industrial systems:
The ESP32 drives sensors and motors, while WIZnet provides a wired backbone with virtually no packet drops.
Infrastructure equipment:
The ESP32-S3 handles logic and monitoring, and WIZnet keeps a 24/7 link alive with sub-millisecond latency.
Audio / streaming:
The ESP32 takes care of DSP and user controls, and WIZnet delivers audio data over the wire with no jitter.
Espressif is still the brain at the center.WIZnet is the rock-solid wire that lets that brain run flat out, with zero performance compromise.

5 Wi-Fi where you can, **wire where you must**

If this magazine had to leave you with just one line, it would be this: “**Wi-Fi where you can, wire where you must.**”
PoCs, prototypes, temporary installs, low-traffic sensors: Wi-Fi alone is more than fast and flexible enough.
24/7 services, robots, cameras, audio gateways: The cost of pulling a cable is nothing compared to the cost of a dropped connection.
The answer here is not “throw Wi-Fi away.”Wi-Fi is still your key tool for exploration, experimentation, and installation flexibility.
WIZnet Ethernet is the infrastructure-level partner that takes responsibility for link quality and availability.

6 Why this magazine exists

This “**WHY CABLE? – Espressif × WIZnet Maker**” special isn’t just for makers.It’s also for engineers who ship products and for teams designing B2B services.
Through real-world examples in robots, cameras, audio, gateways, and infrastructure, we’ll show how ideas that start life on Wi-Fi evolve-using the ESP × WIZnet combo-into “**24/7 services**” and “**field-proven products.**”
To close, let’s write out the slogan that runs through this entire issue one more time:
“**Go wireless to explore. Go wired to deliver.**”
In your next ESP project, this magazine will help you find exactly where that “**one extra cable**” needs to be plugged in.

ESP at the Core, **WIZnet on the Wire** Tutorial Roadmap

2

A Technical Roadmap: Adding Wired Reliability to Wireless Freedom

We're all fans of Espressif. The ESP32's "democratization of wireless" completely changed the maker scene. But the moment a project steps out of the prototype phase and into the real world, we often hit a wall.

In a trade show packed with RF noise, or in a robot-control environment where even 1 ms of latency is unacceptable, the freedom of Wi-Fi sometimes hands you a bill labeled "**instability.**"

That's where we propose **Hybrid Connectivity**: a strategy that uses the **ESP32 as the brain (core)** and the **WIZnet W5500 as the nervous system (wire).**

Use Wi-Fi for setup and OTA updates. Route the data that must **never** drop through WIZnet's wired engine. That's the way professionals build IoT systems that ship.

In this article, we've handpicked **three of the most solid, field-proven tutorials** from the WIZnet Maker site.

Follow this roadmap and take your project from "fun toy" to **industrial-grade.**

The Roadmap: From "Hello World" to "Industrial Robot"

The roadmap we've prepared is simple and clear.It walks you step by step-from basic connectivity, to a robust gateway, and finally into advanced control with ROS2.

STEP 1. The Awakening (Beginner)

"First, connect. Then, start talking."

Every journey starts with a ping.

In this step, you physically connect the ESP32 and W5500 over SPI, then exchange data using a basic MQTT setup.

Bringing up that first stable link-hardware plus protocol-is **the first step toward hybrid IoT.**

STEP 2. The Stabilizer (Intermediate)

"Connecting is easy. Staying connected is hard."

What matters more than the initial connection is keeping it alive. Out in the field, power isn't always clean. Switches get rebooted. Cables get unplugged and plugged back in.

In Step 2, we focus on the essential skeleton of a **gateway** that can automatically detect failures, recover the network, and reinitialize itself-even under these real-world fault conditions.

STEP 3. The Professional (Advanced)

"Into the realm of real-time control."

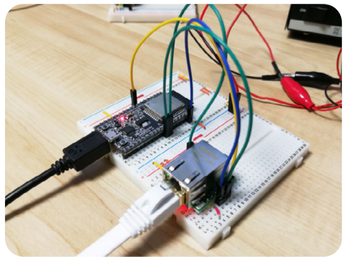
Now we move beyond simple sensors into **robot control**. In a ROS2 robotics environment, Wi-Fi latency and jitter can be fatal. Here, we combine the ESP32's compute power with the W5500's rock-solid wired link to show how to implement an **mROS2 node over Ethernet.**

This is where your ESP-based project crosses the line from "connected gadget" into the territory of **professional, real-time robotic systems.**

Featured Tutorials (WIZnet Maker's Pick)

Here are three battle-tested tutorials from the WIZnet Maker community that you can follow along with right now.

1 [Basic] Communicating via Ethernet-MQTT with ESP32 & W5500



Repo:https://maker.wiznet.io/scott/projects/w5500%E3%82%92%E4%BD%BF%E3%81%A3%E3%81%A6esp32%E3%81%A7ethernet%2Dmqtt%E9%80%9A%E4%B-F%A1%E3%81%99%E3%82%8B/

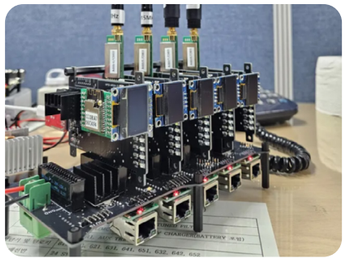
Keywords
ESP32, W5500, MQTT, Visual Guide

Difficulty
★★☆☆☆

Summary
"Seeing is believing."This tutorial explains how to combine the ESP32 and W5500 with clear images and wiring diagrams. It walks through jumper wiring, library setup, and the first successful MQTT communication with clean, minimal visuals.

Why this?
Beginners usually hit the wall of **unclear documentation**.
Here, both the hardware connections and code flow are carefully visualized. It doesn't just prove "it connects" - it shows **how to connect it cleanly**, making it a solid reference for a first, hassle-free ESP32 + W5500 setup.

2 [Structure] SOHO MQTT Multi-Stack Gateway (ESP32 + LoRa + W5500)



Repo: https://maker.wiznet.io/Lihan__/projects/soho-mqtt-multi-stack-gateway-with-wiznet-w5500/

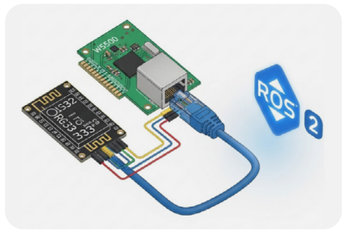
Keywords
ESP32, LoRa, Multi-Protocol

Difficulty
★★★★☆

Summary
This project captures the essence of **Hybrid Connectivity.**
Short-range sensors use fast, simple ESP-NOW; long-range sensors use LoRa; all data is sent to an MQTT broker via W5500 Ethernet in a fully working **multi-stack** gateway design.

Why this?
It answers not just "does it connect?" but "**how should we structure it?**"
By clearly separating the wireless sensor network and the wired backbone, you learn how to manage queues under load without data loss and how to design **hot-swappable, modular** hardware and software. It's a practical textbook for integrated architecture.

3 [Advanced] SPI-Ethernet for mROS2-esp32



Repo: https://maker.wiznet.io/lawrence/projects/spi-ethernet-module-w5500-us-age-manual-for-mros2-esp32/

Keywords
Robotics, ROS2, Real-time, Low Latency

Difficulty
★★★★★

Summary
This tutorial integrates the W5500 into **mROS2** on ESP32, bringing ROS2-core to embedded robot control-onto a wired Ethernet link. It shows how to move from Wi-Fi to W5500 Ethernet to prevent data loss and minimize communication latency between ROS2 nodes.

Why this?
Imagine your robot entering a Wi-Fi dead zone while driving autonomously - terrifying, right?For engineers building AGVs or robot arms, this guide is a technical proof of **why robots need a wired (or industrial wireless) backbone** for critical control traffic, not just hobby-grade Wi-Fi.



Frames Never Die

Espressif Camera & PoE Stories

3

Always-on eyes, never-dropping power

When do you need a security camera the most?In that one second when something actually happens.

The ESP32-CAM gave makers all over the world affordable "**vision**"-for under ten dollars.But it also came with a serious challenge: **the unreliability of wireless links.**

Every wall a Wi-Fi signal passes through cuts its strength. Whenever the bandwidth wobbles, your video turns to mush. And an even bigger problem sits behind all of this: **power.**

Even a "wireless" camera still needs a power cable.So you end up with wireless for data, wired for power-and this dual burden is exactly why so many projects never make it past the lab bench.

Our answer is **Ethernet & PoE (Power over Ethernet):** combining rock-solid data and convenient power in a single LAN cable.This is the true "**single-cable solution.**"

However, adding Ethernet (W5500) to an ESP32-CAM is harder than it looks.The main obstacle is a technical one: **SPI pin conflicts.**

In this chapter, we'll walk through step-by-step solutions:

1. **Precise wiring guidance** to avoid conflicts
2. **Driver tuning** for high-quality video streaming
3. **Ready-made PoE hardware** that can go straight into the field with no complex rework

The Technical Barrier: The Battle for the SPI Bus

The fundamental reason it's hard to give the ESP32-CAM a wired interface is simple:there just aren't enough free GPIO pins.

To push high-bandwidth video, the camera **monopolizes the ESP32's primary SPI bus**. If you hook up an Ethernet module in the usual way, the camera and Ethernet end up fighting for the same SPI lines-often resulting in both peripherals failing.

To fix this, the W5500 Ethernet module must be forced onto a **secondary SPI bus** that the camera doesn't use.But on the ESP32-CAM board, almost no pins are left over.

So the key is to discover a "**golden pin combination**": a set of pins that:

* does not conflict with the SD card slot, and
* still provides reliable SPI communication for the W5500.

The projects we're about to introduce either:

* share the "answer sheet" for this pin map, found through extensive trial and error, or
* eliminate the problem entirely with **hardware designs** that solve the SPI conflict for you.

1 [The Foundation] Solving the SPI Conflict & Optimization



Role

[Build & Code] Hardware wiring and software setup guide for resolving pin conflicts

Summary

A clear technical answer to the question:
"Why does my ESP32-CAM Ethernet connection keep failing?"
This project provides an exact schematic that assigns the W5500 to the **HSPI port (IO14, 12, 13, 15)**, avoiding interference with the camera on VSPI. It also includes code examples showing how to declare and configure these pins in software, turning it into a "textbook" reference for makers building DIY cameras.

Why this?

It cleanly fixes issues most online tutorials miss, such as **floating CS pins** and **boot mode failures.** Before uploading any code, check your hardware wiring against this guide. It drives your failure rate down to effectively **0%**.

Repo: https://maker.wiznet.io/lawrence/projects/esp32-cam-using-hspi-interface-and-w5500-module/

2 [Industrial Standard] The Enclosed Solution: M5Stack PoE Camera



Role

[Ready-made / Industrial] Fully enclosed industrial PoE camera

Summary

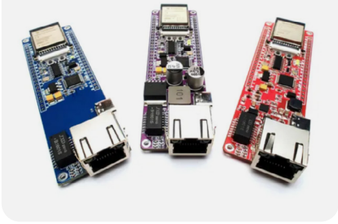
"A camera with a bare PCB does not belong in the field."
This feature introduces the **M5Stack PoE Cam**, an industrial-grade product with ESP32 + W5500 + PoE fully integrated. It uses an **OV3660 (3 MP)** sensor for sharper images than the common OV2640 (2 MP), and comes in a dedicated plastic enclosure so you can mount it directly on ceilings or walls without any extra housing.

Why this?

A true **plug & play** solution: buy it, mount it, power it.It's the perfect choice for readers who want a stable product without soldering or custom wiring, and for engineers who need to deploy many cameras on-site quickly and reliably.

Repo: https://maker.wiznet.io/Benjamin/resellers/m5stack-poe-camera-with-wi-fi-ov3660/

3 [Next Gen] The S3 & Active PoE: ESP32-Stick-PoE-A-Cam



Role

[Future / High-Spec] Fusion of the latest ESP32-S3 and Active PoE

Summary

"Smaller, and more powerful."
This article presents the **ESP32-Stick-PoE-A-Cam**, one of the most advanced ESP32 camera boards currently available.
It goes beyond a standard ESP32 with an **ESP32-S3** capable of AI workloads, and integrates **W5500 Ethernet** plus **Active PoE (802.3af)** into a stick roughly the size of a finger.

Why this?

Many makers struggle with power standards, but this board simply plugs into any standard PoE switch in an office or server room. Its compact design makes it ideal for early adopters who want to install **pro-level CCTV** inside a 3D printer or in tight, hard-to-reach spaces.

Repo: https://maker.wiznet.io/Alan/resellers/esp32%2Dstick%2Dpoe%2Da%2Dcam%2D%2Dan%2Desp32%2Ds3%2Dcamera%2Dboard%2Dwith%2Dactive%2Dpoe/

4

24/7 Life

Audio · Robot Gateway Case Studies



The wired backbone behind wireless freedom

Wi-Fi has given us the gift of **spatial freedom.**
We can reach devices without cables and tweak settings from a smartphone on the couch-this is now baseline convenience.
But when a system has to run **24/7/365**, there are plenty of moments when asking wireless to carry all the load is a bit too much.

- **Robot**:
  The robot's movement should be free, but control signals can't tolerate even 0.1 seconds of error.
- **Energy**:
  Power plants can be monitored remotely, but data collection must not drop a single frame, whatever the weather.
- **Audio**:
  You pick tracks from the sofa on your phone, but the audio stream itself should never waver.

In the professional world, **wired** doesn't replace wireless-it becomes the solid **backbone** that covers wireless blind spots. In this section, we introduce four hybrid projects that combine the ESP32's smart wireless features with WIZnet's rock-solid wired connectivity, capturing the best of both: **convenience and reliability.**

1 [Audio] Non-stop Music: WebRadio & DLNA Player



The Case Study

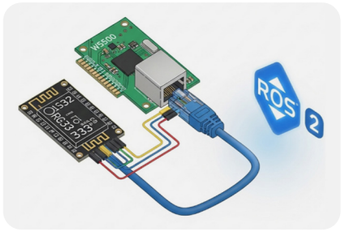
The biggest annoyance with internet radio and DLNA streaming is buffering.This project uses the ESP32 and W5500 to build a high-quality audio player that doesn't stutter. The key is **hybrid usage:**
The user lies in bed and picks music using a smartphone via DLNA (**wireless control**)
The actual audio data is delivered over a stable wired network to the speaker (**wired streaming**)

Why Hybrid?

"Free to choose, never interrupted while playing."
If Wi-Fi interference cuts your music mid-song, the mood dies with it.Wireless delivers the best UX to the listener; wired delivers maximum stability to the hardware. Together, they create a background music system that can run **24/7 without dropping a beat.**

Repo: https://maker.wiznet.io/Hannah/projects/esp32%2Dwebradio%2Dplus-dlna/

2 [Robot] The Safety Net: mROS2 on Wire



The Case Study

In a ROS2 environment, robots broadcast tons of sensor data over wireless.But if RF noise in a factory cuts the link, the robot can stop-or worse, misbehave.
This project adds a W5500 SPI Ethernet link to an mROS2 node to implement **communication redundancy**.
Under normal conditions, data can flow over wireless.
But emergency stop (E-Stop) and other critical control signals are designed to travel over a **wired backbone** that you can truly trust.

Why Hybrid?

"A hidden safety net behind free movement."
Wireless gives robots freedom to move; wired gives them **safety**. The W5500 provides a second, highly reliable "nervous system" for ESP32-based robots, offsetting the risks of unpredictable wireless environments.

Repo: https://maker.wiznet.io/lawrence/projects/spi-ethernet-module-w5500-usage-manual-for-mros2-esp32/

3 [Gateway] Extending the Reach: SolarIT Energy Logger



The Case Study

Solar inverters are often installed on rooftops or in sheds, far from any Wi-Fi router-true RF dead zones.
SolarIT tries wireless first, but wherever the signal is weak or unstable, it extends the network using the W5500. In other words, **wired** takes over where wireless can't reach, acting as a **relay** to guarantee continuous data collection end-to-end.

Why Hybrid?

"Connecting the places wireless can't reach."
Sometimes it's far more reliable to pull a single Ethernet cable than to stack Wi-Fi repeaters.This project keeps the convenience of wireless access while using wired links to casually leap over environmental constraints like distance and walls.

Repo: https://maker.wiznet.io/viktor/projects/solarit/

4 [Compact Node] The Tiny Titan: XIAO W5500 Adapter



The Case Study

As IoT devices get smaller, wireless antennas hit physical limits.The thumb-sized Seeed XIAO (ESP32-C3/S3) series is powerful, but once it's embedded in tight spaces, Wi-Fi can become unstable.
This content introduces an ultra-compact W5500 adapter that stacks on top of the XIAO like a sandwich. With virtually no increase in volume, it turns a shaky wireless node into **the smallest, most reliable wired micro-server.**

Why Hybrid?

"Even where you can't see it, the link must stay strong."
Inside wall switches, small sensor boxes, and other RF-hostile enclosures, this tiny adapter gives you a **24/7 uninterrupted** wired connection. It's a model answer for smart home sensor nodes that must stay **compact on the outside and rock-solid on the network.**

Repo: https://maker.wiznet.io/viktor/resellers/xiao%2Dw5500%2Dethernet%2Dadapter/



A hardware map to turn your ideas into physical reality

You can patch software. You can't roll back hardware.

That's why we have to be the most careful at the final stage of a project. If there's a bug in the code, you rewrite it. But pick the wrong board, and you ruin the entire **form factor**, scrap your enclosure design, and usually end up burning money.

Before you turn the last page of this magazine, there's a brutally practical question we all have to face:

"So… what should I actually buy?"

The cameras and miners we introduced earlier are almost "finished products" with a specific purpose.

But makers' ideas are far more diverse:

- Someone wants a controller with a flashy display.
- Someone else needs a fingertip-sized sensor hidden behind a wall.
- Another is hunting for the cheapest possible gateway, ready for mass production.

At this point, we all fall into the classic **choice dilemma:**

- Do I build it myself with solder and jumper wires, or buy something ready-made?
- Is cost more important, or development speed?
- Do I optimize for expansion, or for compactness?

In this chapter, from the many ways to combine ESP32 and W5500, we've selected **our standard hardware options** that give clear answers to those questions:

① **LILYGO** for those who want **powerful expandability**

② **M5Stack** for those who care about **compact, beautiful use of space**

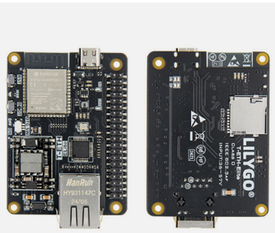
③ **Wireless Tag** for those focused on **aggressive cost efficiency**

④ **WIZnet original modules** for those who want maximum **custom freedom**

This is more than a shopping list.

It's the **physical foundation** you'll use to turn your ideas into real-world hardware.

1 [The New Trend] LILYGO T-ETH-Elite (ESP32-S3)



The Hardware

"Not just a board. An expandable gateway."

LILYGO T-ETH-Elite pairs the ESP32-S3 with a W5500 on the main board and lets you stack LTE or LoRa modules on top like shields. For makers dreaming of a "super gateway" that unifies multiple wireless networks, this is a top-tier option.

Why This?

You get the ESP32-S3's stronger compute with W5500's stable Ethernet. LTE/LoRa modules plug in like LEGO bricks, so you can build a wired backbone plus wireless sensor network without messy wiring. If you're planning a complex, feature-rich project, this board is an ideal canvas.

Type	Target	Specs
Dev Board (Expandable Gateway)	High-End Maker, LoRa/LTE ProjectsKey	ESP32-S3, W5500, PoE slot, expansion shield

Repo: https://maker.wiznet.io/bruno/resellers/lilygo%2Dt%2Deth%2Delite%2Desp32%2Ds3%2Dethernet%2Ddevelopment%2Dboard/

2 [The Compact Core] M5Stack ATOM PoE (Base Kit)

The Hardware

"No camera, no screen. Just connection."

If the PoE camera from Article 3 was the "eye," ATOM PoE is the "nerve."Inside this 24 × 24 mm cube are an ESP32, W5500, and full PoE circuitry. With everything trimmed down to focus on network connectivity, it's perfect for tight breaker panels, behind furniture, or other cramped spaces.



Why This?

DIY boards look unfinished; industrial gear is often huge. ATOM PoE sits in the sweet spot. Plug a sensor into the GROVE port, connect PoE, and you have an ultra-compact IoT node in about a minute. It proves, very technically, that "small is beautiful."

Type	Target	Specs
Module (Enclosed)	Industrial Node, Sensor HubKey	ESP32-Pico, W5500, PoE (802.3af), 24 × 24 mm

Repo: https://docs.m5stack.com/ja/atom/atom_poe

3 [The Visual Controller] Makerfabs MaTouch ESP32-S3 (4.0" TFT)

The Hardware

"When you need a window for your data."

The endpoint of many IoT projects is a display.Makerfabs MaTouch integrates an ESP32-S3, a 4" high-resolution touchscreen, and a W5500 on a single board. Pushing rich graphics over Wi-Fi can make the UI stutter; this board pulls data over wired Ethernet for smooth, tear-free interfaces.



Why This?

Building a smart home wall pad, 3D printer controller, or factory panel?Skip buying and wiring a separate screen and Ethernet module. This board solves **display + wired communication** together in one clean package.

Type	Target	Specs
HMI Display (Integrated)	Dashboard, Control Panel, KioskKey	ESP32-S3, W5500, 4.0" IPS Touch, RGB interface

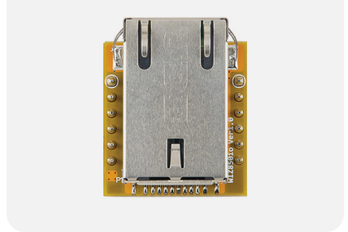
Repo: https://maker.wiznet.io/Benjamin/projects/esp32-s3-smart-display-portal-with-w5500-ethernet-and-ws2812-led-control/

4 [The Evergreen Classic] WIZ850io Module

The Hardware

"The end of tuning is stock; the start of creation is a module."

Despite all the flashy products out there, WIZ850io is still a bestseller. It's a minimal combination of a W5500 and RJ45 jack. Whatever MCU you use-ESP32, Arduino, Pico, etc.-this module adds wired Ethernet with just four SPI wires.



Why This?

It frees you from any single vendor's board ecosystem and gives you real independence. From first breadboard prototypes to custom PCBs for mass production, WIZ850io is a core part you can rely on. For many IoT developers, the beginning and end of Ethernet development comes back to this little red module.

Type	Target	Specs
Component (Module)	DIYer, Custom PCB, PrototypingKey	W5500, integrated MagJack, SPI interface

Repo: https://wiznet.io/products/ethernet-modules/wiz850io

Go wireless to explore.
Go wired to deliver



maker.wiznet.io