

Ethernet and microcontrollers

A physical Ethernet connection serves as a dedicated hardware link that directly connects local microcontrollers to LANs and industrial fieldbuses. Unlike wireless solutions, a wired Ethernet port ensures reliable, low-latency, and secure communication, which is vital for real-time operations. When combined with internal or external *Media Access Control (MAC)* layers and a *Physical Layer (PHY)* transceiver, microcontrollers can efficiently process raw network packets, execute Modbus TCP functions, or stream data to central SCADA systems without experiencing signal loss.

The MAC (Media Access Control) - The Data Brain

The MAC operates at **Layer 2 (Data Link)** of the OSI model. It acts as the logic controller that packages data into standardized network frames.

- **Core Duties:** Appends or reads destination and source MAC addresses, calculates error-checking checksums (CRC), and enforces network access rules to prevent packet collisions.
- **Implementation:** It is entirely digital. In advanced microcontrollers (like an ESP32 or STM32), the MAC is a dedicated hardware block built right inside the main chip.

The PHY (Physical Layer Transceiver) - The Electrical Muscle

The PHY operates at **Layer 1 (Physical)** of the OSI model. It acts as the hardware translator that turns digital logic into raw physics.

- **Core Duties:** Translates the digital 1s and 0s from the MAC into differential voltage signals (or pulses) that travel down the physical copper twisted-pair Ethernet cable. It also handles auto-negotiation for line speed (10 vs. 100 Mbps).
- **Implementation:** It is an analog-digital hybrid circuit. Because it deals with physical line voltages, it is almost always housed in an external chip (like the LAN8720) or integrated into a combined chip layout adjacent to the isolation transformer (MagJack).

How They Talk: MII and RMII

The MAC and the PHY must communicate through a highly synchronised digital bus on your PCB. The most common standard is RMII (Reduced Media Independent Interface), which requires a shared, ultra-stable 50 MHz clock signal to precisely sync the transmission and reception lines between the two layers.

Comparison of some popular Ethernet solutions

Criteria	W5100	W5500	W6100	CH9120	ENC28J60	LAN8720
Price-Level	Medium (Becoming obsolete, making modules slightly pricier than W5500)	Medium-Low (Mass-produced, highly cost-effective for performance)	Medium (Slightly higher premium due to advanced IPv6 features)	Low (Extremely cheap integrated chip and minimal component count)	Very Low (The cheapest option on the market, but hidden costs in MCU RAM)	Low (Very inexpensive chip, but requires a premium MCU to operate)

Criteria	W5100	W5500	W6100	CH9120	ENC28J60	LAN8720
Advantage	* Hardware TCP/IP offloads MCU * Massive community libraries * Industry-proven stability	* Fast SPI clock (80MHz) * Low power consumption * 8 independent sockets	* Native IPv4 and IPv6 support * Pin-compatible upgrade for W5500 * 8 sockets	* Extremely easy to use * Requires no TCP/IP stack library * Pure serial-to-ethernet bridge	* Very low cost * Small footprint * Widely available in cheap modules	* True high-speed throughput * Low latency * Uses MCU's native MAC hardware
Disadvantage	* Old technology * High power consumption / runs warm * Low socket count (4)	* Limited to IPv4 only * Requires external library (Wiznet)	* Slightly higher cost * Libraries are newer and less mature than W5500	* Not flexible for complex protocols * Limited to only 2 concurrent sockets	* No hardware TCP/IP stack * Consumes massive MCU RAM and Flash * Slow SPI limits speed	* No MAC layer inside * Requires complex RMII wiring (10+ pins) * Only works with advanced MCUs
Supported MCUs	* All 8-bit to 32-bit MCUs * Arduino Uno/Mega * STM32, ESP32, RP2040	* All 8-bit to 32-bit MCUs * Arduino, ESP32, ESP8266 * STM32, RP2040/RP2350	* Modern 8-bit to 32-bit MCUs * STM32, ESP32, RP2040 * Arduino (with updated library)	* Any MCU with a basic UART port * Arduino, PIC, MSP430 * Attiny, STM32, ESP32	* Primarily 8-bit/32-bit MCUs with large RAM * Arduino, STM32, AVR	* Premium MCUs with built-in MAC * ESP32, STM32F4/F7/H7 * NXP i.MX, TI Tiva C
Technical Parameters	* Speed: 10/100 Mbps * Sockets: 4 * Buffer: 16 KB Tx/Rx * Voltage: 3.3V (5V tolerant IO)	* Speed: 10/100 Mbps * Sockets: 8 * Buffer: 32 KB Tx/Rx * Voltage: 3.3V (5V tolerant IO)	* Speed: 10/100 Mbps * Sockets: 8 * Buffer: 32 KB Tx/Rx * Voltage: 3.3V	* Speed: 10/100 Mbps * Sockets: 2 * Buffer: Internal (Fixed configuration) * Voltage: 3.3V - 5V	* Speed: 10 Mbps only (No Fast Eth) * Sockets: Managed by software * Buffer: 8 KB Tx/Rx * Voltage: 3.3V	* Speed: 10/100 Mbps * Sockets: Managed by software/OS * Buffer: Managed by MCU DMA * Voltage: 1.6V to 3.6V
Communication Bus	* SPI * Parallel Bus	* High-Speed SPI (Up to 80 MHz)	* High-Speed SPI	* UART (Serial) via AT Commands / Config Tool	* SPI (Up to 20 MHz)	* RMII (Reduced Media Independent Interface)
Simatic Modbus TCP	* Yes (Via Arduino/MCU Modbus client/server software library)	* Yes (Excellent choice; responsive and stable via MCU Modbus libraries)	* Yes (Supports Modbus TCP over both IPv4 and IPv6 networks)	* Limited (Only via transparent serial bridge; requires specialized Simatic gateway settings)	* Poor (Possible, but software TCP/IP stack often drops Modbus packets under load)	* Excellent (Native speed, perfect for industrial PLC interfacing via lwIP stack)

What is the difference between on-board and external solutions?

Criteria	On-board Ethernet (Native MAC + External PHY)	External Ethernet Solutions (SPI/UART Controllers)
Description	The microcontroller has a built-in Ethernet MAC core. It connects directly to a physical layer transceiver chip (PHY like LAN8720) via RMII/MII interfaces.	The Ethernet MAC, PHY, and often the TCP/IP stack are integrated into a single external chip (like W5500). It connects to any standard MCU via SPI or UART.
Advantage	* Maximum Speed: Offers full 10/100 Mbps or Gigabit line rates without bus bottlenecks. * Low Latency: Direct memory access (DMA) bypasses serial buses. * Highly Flexible: Complete software control over raw packets, custom protocols, and advanced routing.	* Saves MCU Resources: Hardware offloads the complete TCP/IP stack; uses near-zero MCU RAM and Flash. * Universal Compatibility: Works with any basic MCU possessing a standard SPI or UART port. * Simpler Hardware Design: Requires significantly fewer PCB traces and pins (4-6 pins vs 10+ pins).
Disadvantage	* High MCU Resource Consumption: The software network stack (e.g., lwIP) consumes substantial MCU RAM and Flash. * Complex PCB Routing: Requires high-frequency 50MHz trace matching and strict noise isolation guidelines. * High Pin Count: Consumes 10 to 12 GPIO pins for the RMII interface.	* Bandwidth Bottleneck: Maximum throughput is limited by the SPI or UART bus clock speed. * Limited Concurrent Connections: Restricted to a fixed number of hardware sockets (e.g., 8 sockets for W5500, 2 for CH9120). * No Raw Packet Control: Harder to implement specialized or non-standard network protocols.
Typical Components	* ESP32, STM32F4/F7/H7, NXP i.MX * Paired with PHY chips: LAN8720, RTL8201, DP83848	* Any MCU (Arduino Uno, Attiny, RP2040, PIC) * Paired with: W5500, W6100, W5100, CH9120, ENC28J60
Ideal Use Cases	* Industrial gateways with heavy traffic. * Video streaming or high-frequency data logging. * Systems running a Real-Time OS (RTOS) or Linux with advanced security (TLS/SSL).	* Simple IoT sensors, MQTT clients, and Modbus TCP nodes. * Projects using small, low-cost microcontrollers with limited RAM. * Rapid prototyping and simple PCB layouts.

Comparison of some microcontrollers with on-board Ethernet

Product Setup	Solution Type	Controller / PHY Used	Price-Level (Approx.)	Main Use Cases	Effectivity & Network Performance
Waveshare RP2040-ETH (Raspberry Nano Ethernet style)	External (Integrated Chip)	WCH CH9120 (UART-to-Ethernet Bridge)	Low (~€11 - €15)	* Simple headless serial data streaming * Remote Modbus-RTU to Modbus-TCP bridges * Compact IoT telemetry devices	Medium-Low: Extremely lightweight on MCU resources, but bandwidth is capped by internal UART speeds. Not suitable for complex web hosting or heavy network loads.

Product Setup	Solution Type	Controller / PHY Used	Price-Level (Approx.)	Main Use Cases	Effectivity & Network Performance
Arduino Uno R3 / R4 + Arduino Ethernet Shield 2	External (Stackable Shield)	WIZnet W5500	Medium (~€35 - €45 total)	* Learning & academic prototypes * Home automation switches * Small local web servers for status toggles	Medium: Excellent hardware offloading (TCP/IP is handled by the shield). However, an 8-bit Uno R3 can struggle with large data buffers despite the chip's speed.
Arduino Nano + Nano Ethernet Shield v1.0	External (Mini Shield Base)	Microchip ENC28J60	Very Low (~€8 - €12 total)	* Lowest-cost custom sensors * Simple ping utilities * Tiny local network triggers	Low: The shield lacks a hardware TCP/IP stack, meaning the software network stack must run directly on the Nano's tiny 2KB RAM. Highly prone to crashes under moderate network loads.
Arduino MKR Series + Arduino MKR ETH Shield	External (Industrial Form Factor)	WIZnet W5500	High (~€55 - €70 total)	* Industrial IoT edge gateways * Smart building sensors * Professional-grade PLC Modbus TCP nodes	High: Blends a high-performance, 32-bit Cortex-M0+ processor with the W5500's stable hardware TCP/IP stack. Ideal for zero-packet-drop industrial applications.
ESP32 DevKitC + LAN8720 Module	On-board (Native MAC + Ext. PHY)	Native ESP32 EMAC + LAN8720 PHY	Low (~€8 - €12 total)	* High-speed Wi-Fi to Ethernet bridging * Real-time data logging / WebSocket servers * TLS/SSL secured internet appliances	Very High: Native DMA access completely eliminates serial bus bottlenecks, pushing real line speeds past 30+ Mbps. Requires software stack memory management (lwIP).

Product Setup	Solution Type	Controller / PHY Used	Price-Level (Approx.)	Main Use Cases	Effectivity & Network Performance
STM32F407VET6 Black Board (with native RJ45 port onboard)	On-board (Native MAC + Integrated PHY)	Native STM32 MAC + LAN8720 (or DP83848)	Medium-Low (~€15 - €25)	* Industrial control machinery * Heavy-traffic Modbus gateways to SCADA systems * Multithreaded RTOS network routing	Maximum: Enterprise-grade performance. High processing power and dedicated memory buses allow raw, sub-millisecond packet control without saturating the processor.

Ethernet topics on lamaPLC

Page	Date	Tags
• Ethernet and microcontrollers	2026/07/27 21:46	ethernet , communication , microcontroller , esp32 , arduino , stm32f4 , w5100 , w5500 , w6100 , ch9120 , enc28j60 , lan8720 , mac , phy , tcp ip , uart
• lamaPLC Communication: ControlNet	2026/04/23 21:51	bus , communication , devicenet , cip , can , ethernet ip , ethernet , controlnet , mac , peer-to-peer , allen-bradley , rockwell automation , fieldbus , rg-6 , coaxial cable
• lamaPLC Communication: DeviceNet	2024/11/16 20:43	bus , communication , devicenet , 62026-3 , cip , can , ethernet ip , ethernet , controlnet , mac , peer-to-peer , canopen , allen-bradley , rockwell automation , bosch
• lamaPLC Communication: EtherNet/IP	2026/04/23 21:51	communication , ethernet , bus , ip , cip , ethernet ip , udp , industrial ethernet , ie , tcp , unicast , devicenet , controlnet , ieee 802.3
• lamaPLC Communication: IEC 61850 basic	2026/04/23 21:51	1-wire , communication , bus , xml , iec 61850 , iec , ethernet , scl , goose , ied
• lamaPLC Communication: IoT	2026/04/23 21:51	communication , iot , internet , iomt , 6lowpan , ipv4 , ipv6 , bluetooth , ble , li-fi , nfc , rfid , wi-fi , zigbee , z-wave , lte-advanced , 5g , lora , dash7 , lpwan , lorawan , sigfox , nb-iot , weightless , rpma , mioty , vsat , ethernet , thread , matter
• lamaPLC Communication: KNX / EIB	2026/04/23 21:51	communication , bus , knx , knx tp , knx pl , knx rf , knx ip , eib , ehs , batibus , ethernet , ets
• lamaPLC communication: Matter	2026/04/23 21:51	communication , iot , matter , ethernet , tcp , udp , ipv6 , thread , ble , cellular , wifi , pki , ip
• lamaPLC Communication: PowerLink	2026/04/23 21:51	communication , bus , ethernet , tcp , udp , http , powerlink , fast ethernet , opensafety
• lamaPLC Communication: TCP / UDP Basic	2025/11/20 22:49	communication , bus , tcp , udp , tcp ip , darpa , ietf , smtp , http , https , ftp , ftps , ssh , pop3 , imap , mysql , cpanel , whm , ssl , webmail , mqtt , ethernet , ip
• lamaPLC communication: Thread	2024/11/15 23:07	bus , communication , matter , thread , matter over thread , zigbee , aes , ipv6 , iot , 6lowpan , ethernet , wifi
• lamaPLC Communication: Z-Wave	2026/04/23 21:51	communication , ethernet , bus , ip , z-wave , ism

• lamaPLC Communication: Zigbee	2026/04/23 21:51	communication , ethernet , bus , ip , zigbee , wpan , bluetooth , wi-fi , ism , ieee 802.15.4 , zdo , zigbee pro , zc , zr , zed , smart energy , homegrid , homeplug , powerline , ipso , sunspec , 6lowpan , ipv6 , rf4ce
• LamaPLC: Arduino MKR Family	2025/08/24 12:19	arduino , board , mkr fox 1200 , mkr zero , ethernet , shield , m-bus , rs-485 , can , env , sd card , rgb , gps , iot , wifi , sigfox , lora , gsm , lte-m
• lamaPLC: ESP32 / ESP8266	2025/11/22 00:07	esp8266 , esp32 , esp32-c2 , esp32-c3 , esp32-c5 , esp32-c6 , esp32-c61 , esp32-h2 , esp32-s2 , esp32-s3 , esp32-p4 , espressif systems , communication , ethernet , ip , wi-fi , thread , zigbee , matter , homekit , bluetooth , mqtt , adc , spi , uart , i2c , i2s , rmt , pwm , usb , usb otg , twai
• lamaPlc: Industrial Ethernet (IE) Basic	2026/04/23 21:51	communication , bus , ethernet , industrial ethernet , ie , ethercat , ethernet ip , profinet , powerlink , sercos , iii , cc-link , modbus tcp , ieee , rstp , stp
• lamaPlc: Profinet	2026/04/23 21:51	communication , bus , profinet , mrp , rstp , gsdml , xml , profibus , ethernet , tcp

[Ethernet](#), [communication](#), [microcontroller](#), [ESP32](#), [Arduino](#), [STM32F4](#), [W5100](#), [W5500](#), [W6100](#), [CH9120](#), [ENC28J60](#), [LAN8720](#), [MAC](#), [PHY](#), [TCP/IP](#), [UART](#)

This page has been accessed for: Today: 4, Until now: 5

From:
<https://lamaplc.com/> - **lamaPLC**

Permanent link:
https://lamaplc.com/doku.php?id=com:eth_and_micro

Last update: **2026/07/27 23:01**

