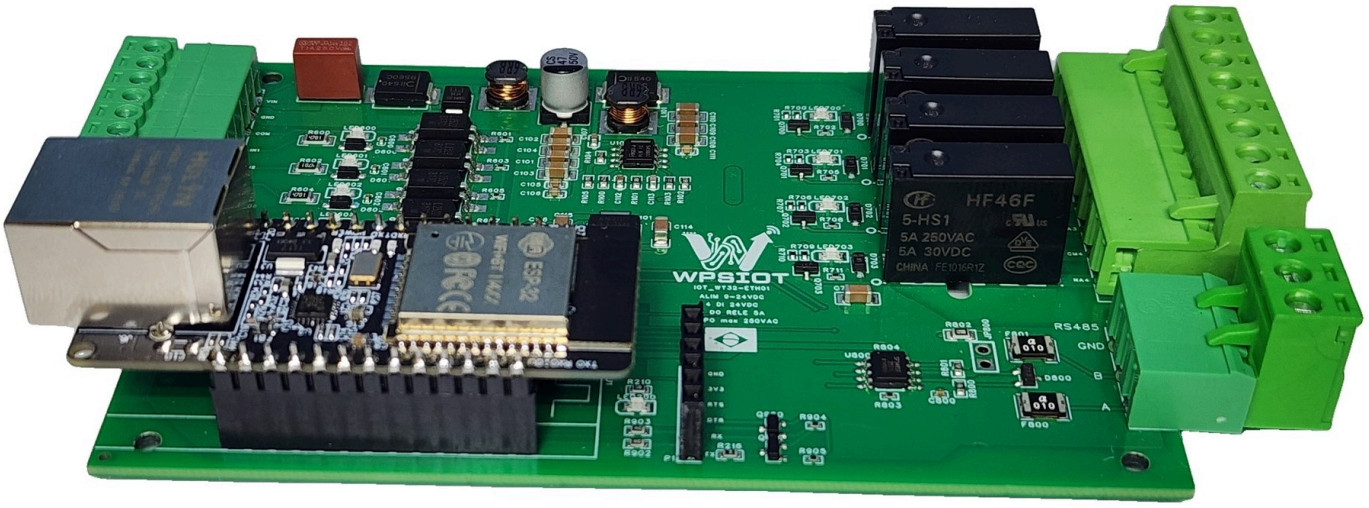
WPSIOT · TECNOLOGIA E INOVAÇÃO

Manual do Usuário

IOT_WT32-ETH01 4DI 4RELE 24V v1.0

Placa IoT com ESP32 e Ethernet (módulo WT32-ETH01) · 4 entradas digitais 24 Vcc optoacopladas · 4 saídas a relé · porta RS485 · alimentação 9 a 24 Vcc



4 entradas
digitais 24 Vcc optoacopladas

4 relés
contato seco 5 A / 250 Vca

Ethernet + Wi-Fi
ESP32 no módulo WT32-ETH01

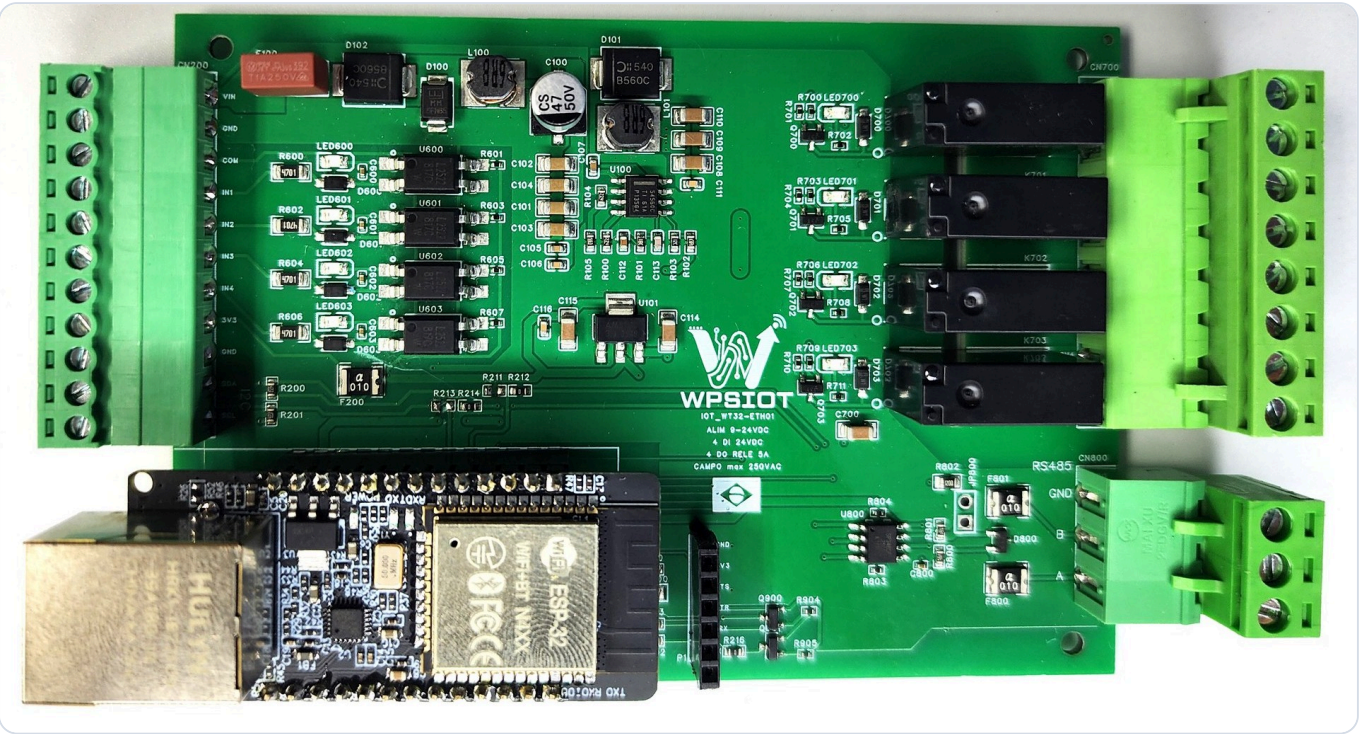
RS485
Modbus RTU e outros protocolos

Suporte e dúvidas pelo **WhatsApp**
11 99761-7226

2026
Rev. 1.0 · 12/11/2026

Apresentação da Placa

A **IOT_WT32-ETH01 4DI 4RELE 24V v1.0** é uma placa de automação e IoT da WPSIOT baseada no módulo **WT32-ETH01** (ESP32 com Ethernet, Wi-Fi e Bluetooth). Ela reúne, numa placa só, a fonte, as entradas isoladas, os relés e a porta RS485 necessários para ligar o ESP32 direto a sensores e cargas de campo.



Características:

- Módulo **WT32-ETH01 incluso**: ESP32 com Ethernet (LAN8720), Wi-Fi e Bluetooth
- Alimentação: **9 a 24 Vcc**, com fusível e proteção contra inversão de polaridade
- **4 entradas digitais** optoacopladas para 24 Vcc, com LED indicador
- **4 saídas a relé** (HF46F), contato seco NA de 5 A / 250 Vca, com LED indicador
- **Ethernet** (RJ45) e **Wi-Fi** pelo módulo
- Porta **RS485** (configuração de fábrica), com terminação de 120 Ω selecionável por jumper
- **I2C** (SDA/SCL) no borne, como alternativa à RS485
- Saída de **3,3 V** no borne
- LED de status programável e conector de gravação serial de 6 vias
- Bornes plugáveis · placa de 115 × 80 mm com 4 furos de fixação

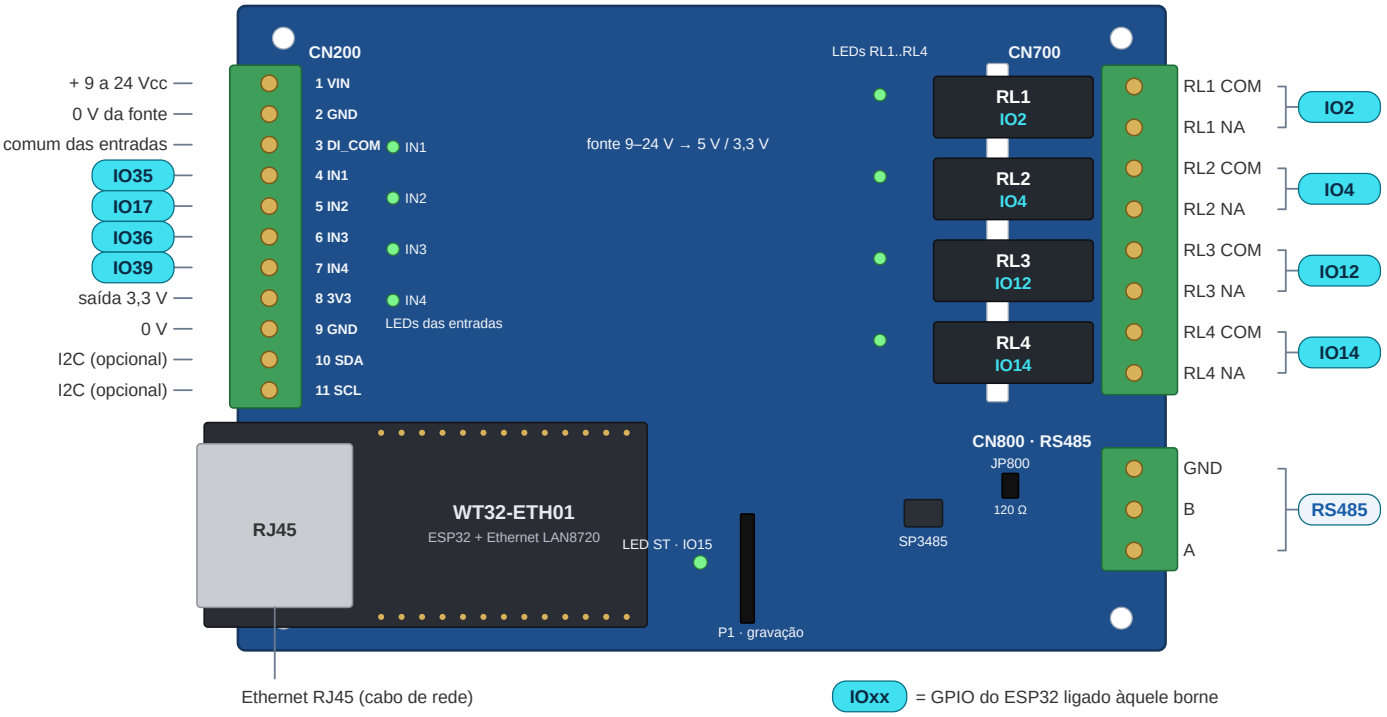
Aplicações

Por usar um ESP32 padrão, a placa pode ser programada pela **Arduino IDE** e usada com firmwares como **ESPHome** e **Tasmota**, ou trabalhar como CLP com o **OpenPLC**. Serve para automação residencial e predial, acionamento de cargas pela rede, leitura de sensores e botoeiras, e integração de equipamentos por Modbus RTU (RS485) ou pela rede Ethernet.

> **Acompanha:** 1 placa IOT_WT32-ETH01 4DI 4RELE 24V v1.0 + 1 módulo WT32-ETH01. A gravação do programa requer um adaptador USB-serial de 3,3 V (não incluso) — veja a página 7.

Pinagem

Vista de cima da placa, em escala. Os bornes são plugáveis: **CN200** (alimentação, entradas, 3,3 V e I2C), **CN700** (contatos dos relés) e **CN800** (RS485). O número de cada via está na serigrafia da placa.



CN200 — alimentação e entradas		
1	**VIN**	positivo da fonte, 9 a 24 Vcc
2	**GND**	0 V da fonte
3	**DI_COM**	comum das entradas (ligar no 0 V)
4	**IN1**	entrada 1 · **IO35**
5	**IN2**	entrada 2 · **IO17**
6	**IN3**	entrada 3 · **IO36**
7	**IN4**	entrada 4 · **IO39**
8	**3V3**	saída de 3,3 V (até 0,1 A)
9	**GND**	0 V
10	**SDA**	I2C, opcional · **IO32**
11	**SCL**	I2C, opcional · **IO33**

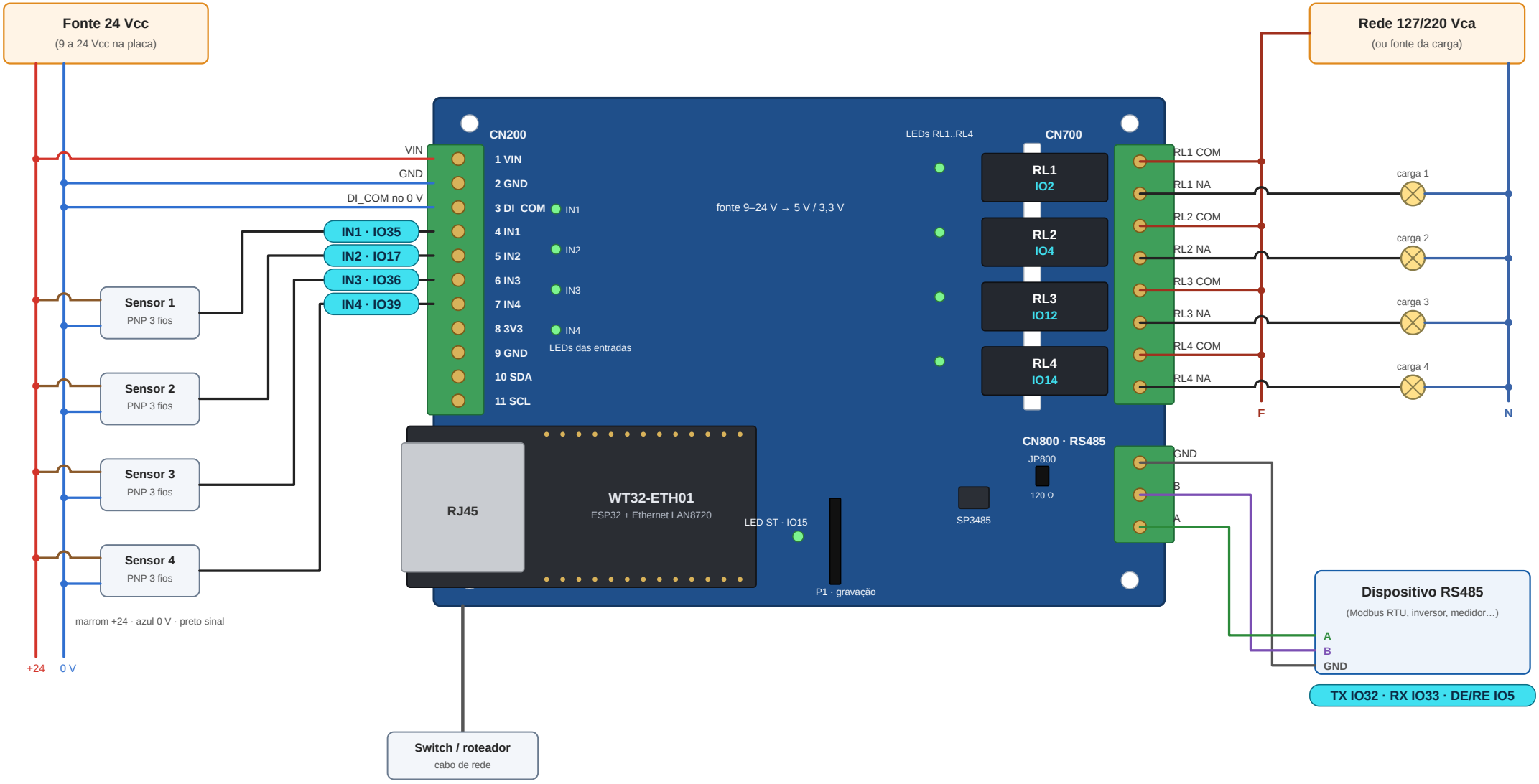
CN700 — saídas a relé	
RL1 COM · RL1 NA	relé 1 · **IO2**
RL2 COM · RL2 NA	relé 2 · **IO4**
RL3 COM · RL3 NA	relé 3 · **IO12**
RL4 COM · RL4 NA	relé 4 · **IO14**

CN800 — RS485	
A	linha A do barramento
B	linha B do barramento
GND	referência (0 V)

Outros conectores: RJ45 do módulo (Ethernet) e **P1**, soquete de 6 vias para gravação (página 7).

Esquema de Ligação

Vista de cima, em escala (gerada a partir do PCB). Em cada borne: a função e o GPIO do ESP32 que a controla.



Entradas: sensor PNP com DI_COM no 0 V. Com 24 V na entrada o optoacoplador conduz e o GPIO lê 0 (LOW); sem tensão lê 1 (HIGH).

Saídas: GPIO em 1 (HIGH) liga o relé e fecha COM–NA. Contato seco HF46F: 5 A / 250 Vca. O LED do relé acende junto.

RS485: A e B do barramento + GND de referência. JP800 fechado = terminação de 120 Ω (use só nos dois extremos do barramento).

I2C: SDA/SCL do CN200 usam os mesmos IO32/IO33 da RS485. De fábrica a placa sai em RS485; para usar I2C, veja a página 6.

Entradas e Saídas

GPIO do ESP32 ligado a cada entrada e a cada relé. “Pino” é o número do pino no módulo WT32-ETH01; alguns têm no módulo uma serigrafia de outra função (entre aspas), mas o que vale é o GPIO.

Entradas digitais — borne CN200

Borne	Entrada	GPIO	Pino	Observação
4	IN1	**IO35**	4	só entrada
5	IN2	**IO17**	21 “TXD”	entrada/saída comum
6	IN3	**IO36**	8	só entrada
7	IN4	**IO39**	9	só entrada

Lógica invertida: com tensão entre INx e DI_COM o optoacoplador conduz e o GPIO lê **0 (LOW)**; sem tensão, o pull-up de 10 kΩ da placa deixa o GPIO em **1 (HIGH)**. Cada entrada tem um LED que acende junto. No programa, use `pinMode(pino, INPUT)`.

- Entradas para **24 Vcc**, isoladas por optoacoplador. O positivo do sinal entra em INx e o **DI_COM vai no 0 V** (sensor PNP, botão ou contato ligado ao +24 V).
- A entrada é reconhecida a partir de cerca de 7 V, então também trabalha com sinais de 12 Vcc. Não passe de 30 Vcc.
- O filtro da entrada limita a leitura a cerca de 200 Hz: serve para sensores, botões e contatos, não para contagem rápida.

Saídas a relé — borne CN700

Relé	Contatos no CN700	GPIO	Pino	Observação
RL1	RL1 COM · RL1 NA	**IO2**	2	pino de boot
RL2	RL2 COM · RL2 NA	**IO4**	3	
RL3	RL3 COM · RL3 NA	**IO12**	5	pino de boot
RL4	RL4 COM · RL4 NA	**IO14**	6	pulsa na partida

Lógica direta: GPIO em **1 (HIGH)** liga o relé e fecha COM–NA; o LED do relé acende junto. Contato seco (relé HF46F): **5 A / 250 Vca** ou 5 A / 30 Vcc, normalmente aberto.

Na partida: IO2 e IO12 são pinos de configuração do boot do ESP32 — o programa só deve ligá-los depois de iniciado. O IO14 solta pulsos durante o boot, então o RL4 pode dar um clique ao energizar a placa: evite usá-lo em carga que não possa receber um pulso curto.

RS485, I2C e Demais Pinos

Porta RS485 — borne CN800 (A · B · GND)

Sinal do SP3485	Função	GPIO	Pino	Caminho
DI	TX — dado que sai	IO32	18 “CFG”	pelo 0 Ω R212
RO	RX — dado que chega	IO33	19 “485_EN”	pelo 0 Ω R214
DE + /RE	direção (1 = transmite, 0 = recebe)	IO5	20 “RXD”	pull-down 10 kΩ: em repouso recebe

Arduino: `Serial2.begin(9600, SERIAL_8N1, 33, 32);` (RX = IO33, TX = IO32) e `Serial2.setPins(33, 32, -1, 5); Serial2.setMode(UART_MODE_RS485_HALF_DUPLEX);` para o DE/RE no IO5 ser controlado pela própria UART.

- Ligue A com A e B com B em todos os equipamentos do barramento, mais o GND de referência.
- Terminação de 120 Ω pelo jumper **JP800**: fechado = terminado. Use só nos dois extremos do barramento.
- As linhas A e B têm PPTC de 0,1 A e proteção contra surto.

RS485 ou I2C

O IO32 e o IO33 também chegam aos bornes **SDA** e **SCL** do CN200 (vias 10 e 11). Quem escolhe o destino dos dois pinos são quatro resistores de 0 Ω (jumpers) ao lado do módulo:

Uso	R212 e R214	R211 (SDA) e R213 (SCL)	
RS485	montados	abertos	**configuração de fábrica**
I2C	retirar	montar	desabilita a RS485

Um ou outro: a placa sai de fábrica em RS485. Para usar o I2C é preciso desabilitar a RS485, passando os dois resistores de 0 Ω de R212/R214 para R211/R213. Nunca deixe os quatro montados ao mesmo tempo.

Demais pinos

Item	Onde	Detalhe
Alimentação	CN200 · 1 VIN, 2 GND	9 a 24 Vcc
Saída 3,3 V	CN200 · 8 3V3, 9 GND	protegida por PPTC de 0,1 A
LED de status	IO15	acende com o GPIO em 0 (LOW)
Ethernet	RJ45 do módulo	IO0 e IO16 são usados pelo Ethernet do módulo: não usar como I/O
Gravação	soquete P1	serial TX0/RX0 + reset automático (página 7)

Gravação do Programa

O WT32-ETH01 não tem USB. O programa é gravado pelo soquete **P1** (6 vias) com um **conversor USB-serial em 3,3 V** (FT232RL, CP2102, CH340 ou similar), com a placa alimentada pela própria fonte de 9 a 24 Vcc.

P1	Serigrafia	Função na placa	Ligar no conversor
1	(sem rótulo)	TX da placa	**RX**
2	RX	RX da placa	**TX**
3	DTR	reset automático	**RTS**
4	RTS	reset automático	**DTR**
5	3V3	saída de 3,3 V do módulo	**não ligar**
6	GND	referência	**GND**

Atenção: coloque o jumper do conversor em **3,3 V** e não ligue o VCC dele na placa. O **DTR e o RTS vão trocados** em relação à serigrafia (RTS do conversor na via 3, DTR na via 4). Se a gravação terminar em "Failed to connect" ou "Wrong boot mode", inverta entre si os fios das vias 3 e 4.

Na Arduino IDE

- Instale o pacote **esp32** (Espressif) e escolha a placa **WT32-ETH01 Ethernet Module**.
- Selecione a porta do conversor e clique em gravar: a placa entra e sai do modo de gravação sozinha.
- Com essa placa selecionada, `ETH.begin()` já usa os pinos do Ethernet do módulo.

Conversor sem RTS

- Ligue só TX, RX e GND.
- Com a placa desligada, una o pino **IO0** do módulo ao **GND** e ligue a alimentação.
- Grave o programa, retire o fio do IO0 e religue a placa.

Exemplo: cada entrada liga o relé de mesmo número

```
const uint8_t ENTRADA[4] = {35, 17, 36, 39};   // IN1..IN4  (LOW = entrada acionada)
const uint8_t RELE[4]    = {2, 4, 12, 14};     // RL1..RL4  (HIGH = relé ligado)
const uint8_t LED_ST     = 15;                 // LED de status (LOW = aceso)

void setup() {
  for (int i = 0; i < 4; i++) {
    pinMode(ENTRADA[i], INPUT);
    digitalWrite(RELE[i], LOW);               // garante o relé desligado antes de virar saída
    pinMode(RELE[i], OUTPUT);
  }
  pinMode(LED_ST, OUTPUT);
}

void loop() {
  for (int i = 0; i < 4; i++)
    digitalWrite(RELE[i], digitalRead(ENTRADA[i]) == LOW ? HIGH : LOW);
  digitalWrite(LED_ST, (millis() / 500) % 2);  // pisca para mostrar que o programa está rodando
}
```

Soluções sob Medida

Precisa de uma placa com outra combinação de entradas e saídas, de um firmware pronto para a sua aplicação ou de um projeto de automação completo? A **WPSIOT** desenvolve soluções sob medida em eletrônica, automação e IoT.

Placas eletrônicas
Projeto de placas dedicadas com ESP32, Ethernet, RS485, entradas isoladas e saídas a relé, no padrão das placas WPSIOT.

Firmware
Programas para ESP32 conforme a aplicação: lógica de acionamento, comunicação em rede e integração com outros equipamentos.

Automação industrial
Projetos com CLP, IHM e supervisório, da lógica à partida em campo.

Integração
Leitura e comando de equipamentos por Modbus RTU (RS485), Ethernet e Wi-Fi.

Fale com a gente pelo **WhatsApp**

11 99761-7226

Suporte da placa, dúvidas de ligação e orçamento de projetos

WPSIOT

TECNOLOGIA E INOVAÇÃO