

Do protótipo à fábrica: Usando Arduino e ESP32 em soluções de Indústria 4.0

Lucas Marim

Engenheiro Eletrônico | Hyperus
Technology



unesp 

 **HYPERUS** 

QUEM SOU EU

- Engenheiro eletrônico (UNESP)
- Mestre em Engenharia Elétrica (UNESP)
- Co-fundador da Hyperus
- Atua com:
 - Sistemas embarcados industriais
 - Monitoramento de máquinas
 - Indústria 4.0



HYPERUS

+600 linha de produção
+60 clientes
5 países



PROBLEMA INDUSTRIAL

- Máquinas sem dados confiáveis
 - Operação baseada em “achismo”
 - Falta de rastreabilidade
 - Baixa visibilidade de perdas
- 📌 “Sem dados, não existe melhoria”



O QUE É INDÚSTRIA 4.0

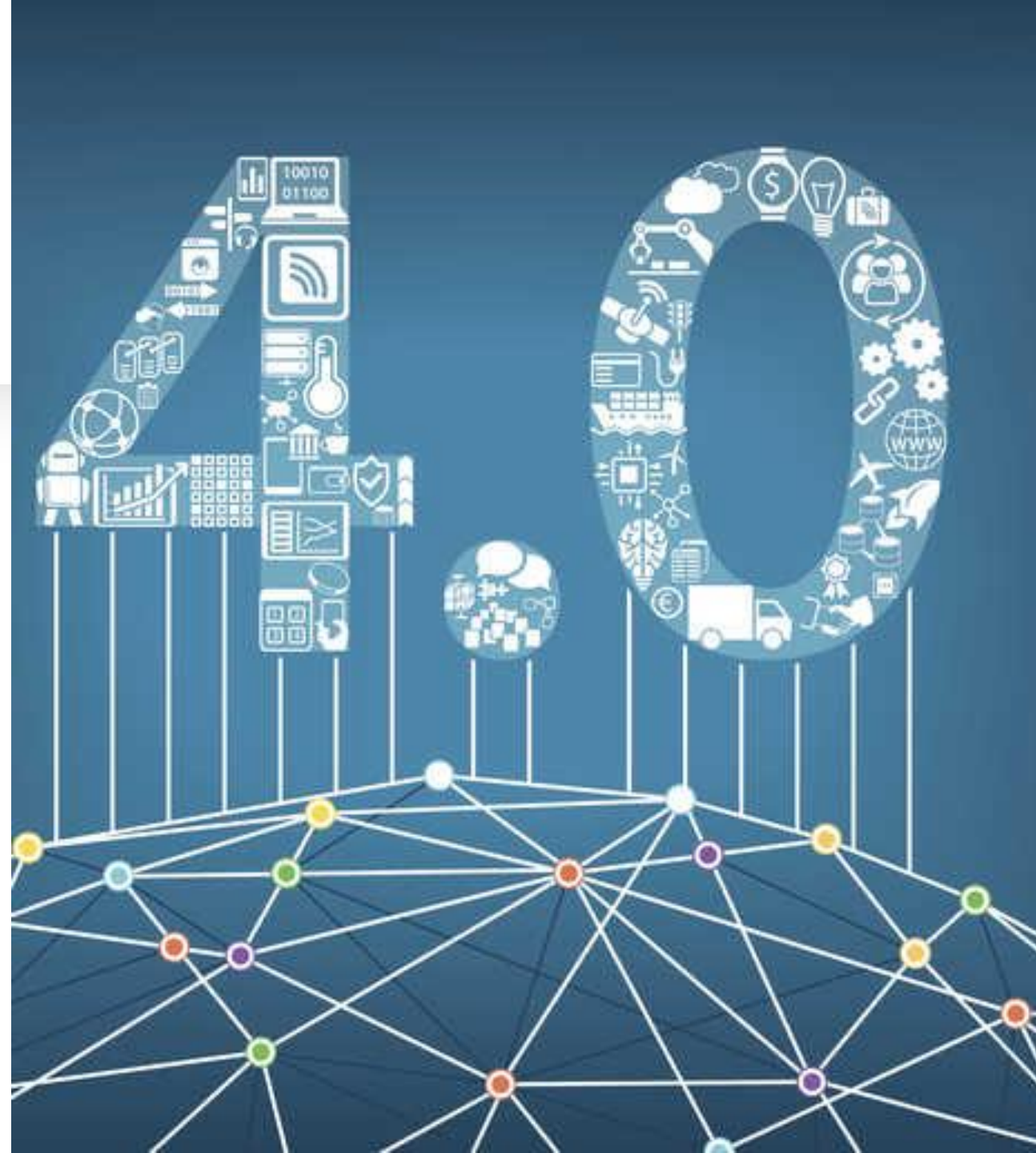
Definição:

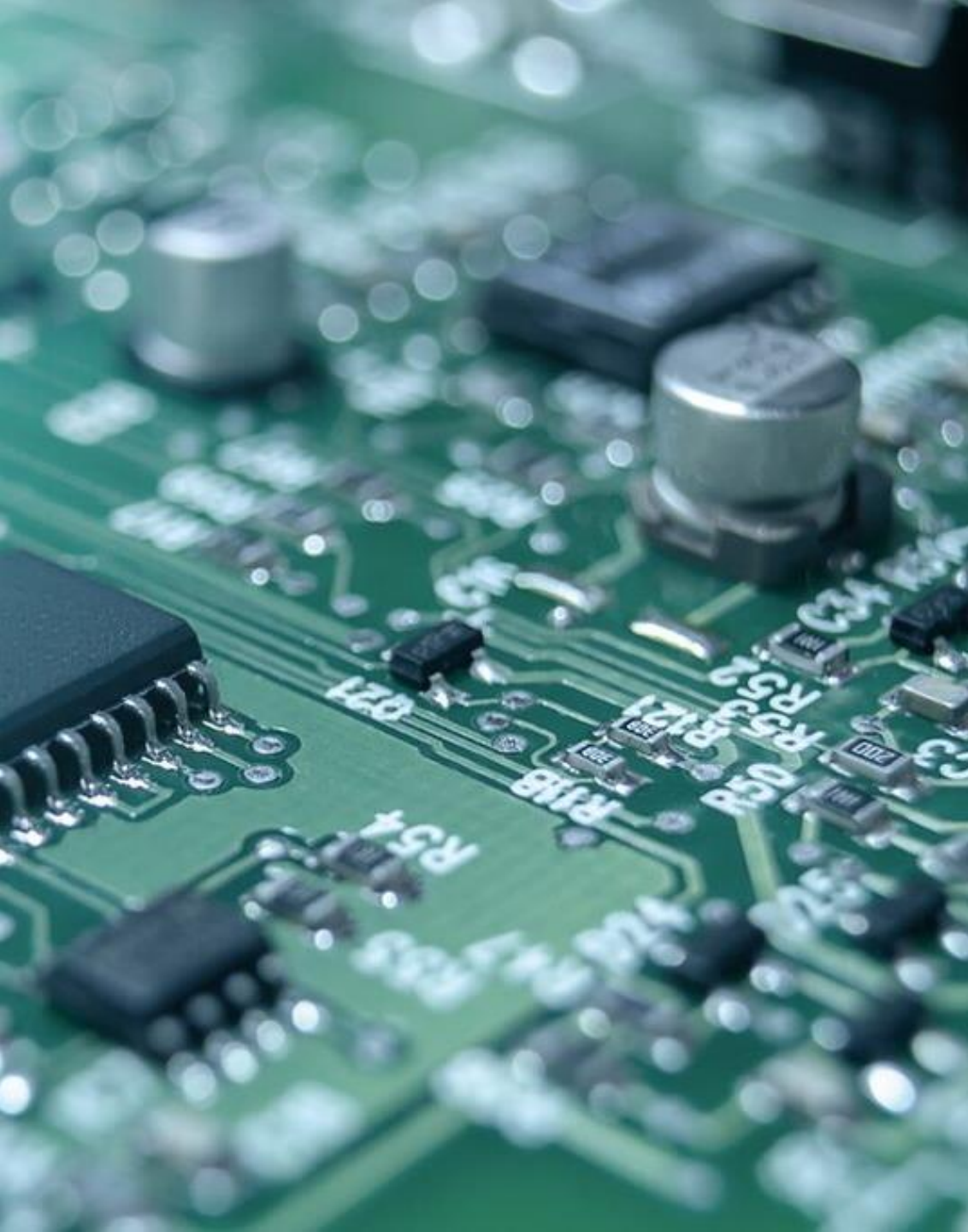
"A Indústria 4.0, sinônimo de manufatura inteligente, representa a concretização da transformação digital na área, oferecendo tomada de decisões em tempo real, maior produtividade, flexibilidade e agilidade para revolucionar a forma como as empresas fabricam, aprimoram e distribuem seus produtos." - IBM



O QUE É INDÚSTRIA 4.0

- Conectividade
- IoT industrial
- Dados em tempo real
- Integração com sistemas





ONDE ENTRA O EMBARCADO

- Sensoriamento
- Aquisição de dados
- Comunicação
- Controle



A PORTA DE ENTRADA: ARDUINO

- Simples
- Barato
- Rápido
- Comunidade enorme

POR QUE ARDUINO É TÃO POPULAR

- Baixa curva de aprendizado
- IDE simples
- Bibliotecas prontas
- Excelente para ensino



EXEMPLO SIMPLES

- Leitura de sensor
- Envio serial
- Protótipo funcional em minutos



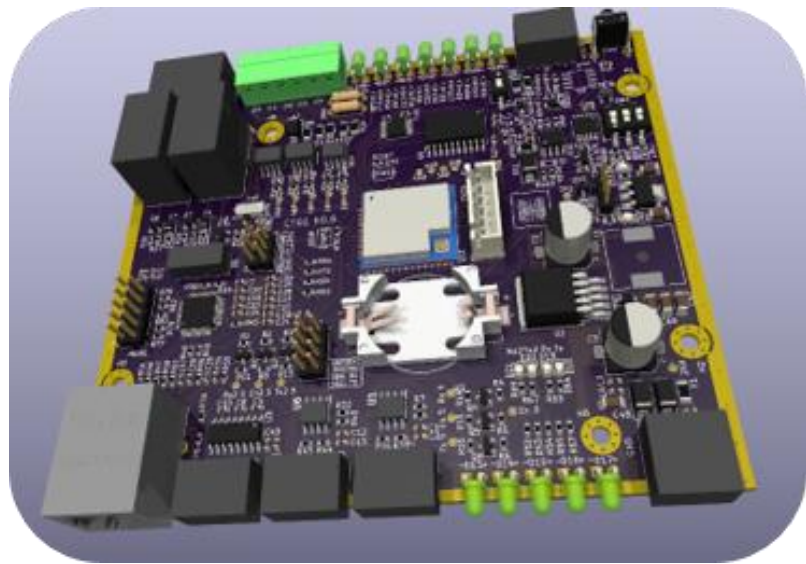
LIMITAÇÕES DO ARDUINO

Limitações do Arduino em aplicações industriais

- Baixo controle de hardware
- Uso intensivo de bibliotecas de alto nível
- Falta de tratamento robusto de erro
- Dificuldade de debug avançado
- Baixa previsibilidade em tempo real
- Escalabilidade limitada

Código cresce → complexidade explode

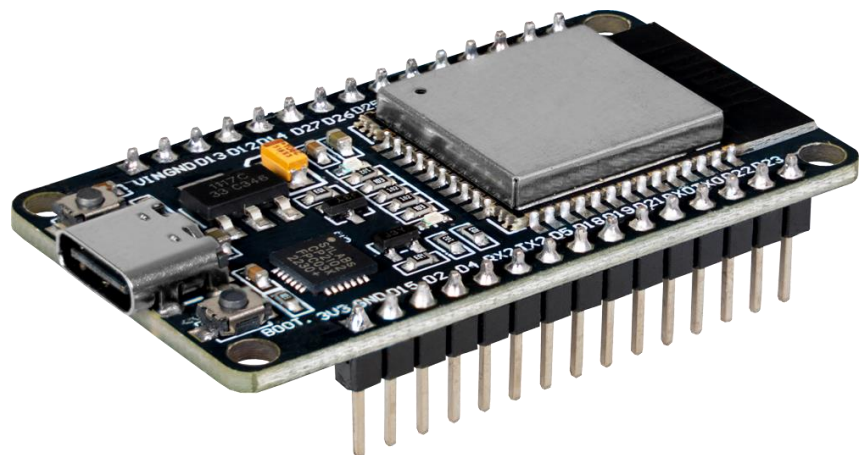




TRANSIÇÃO IMPORTANTE

“Protótipo funcional ≠ Produto industrial”

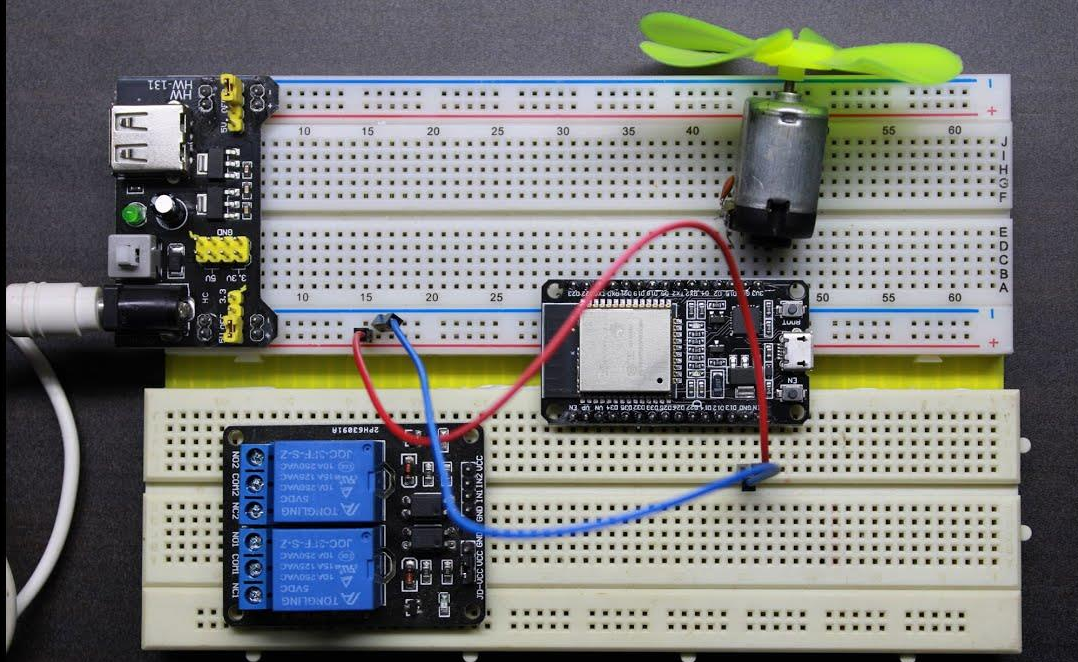
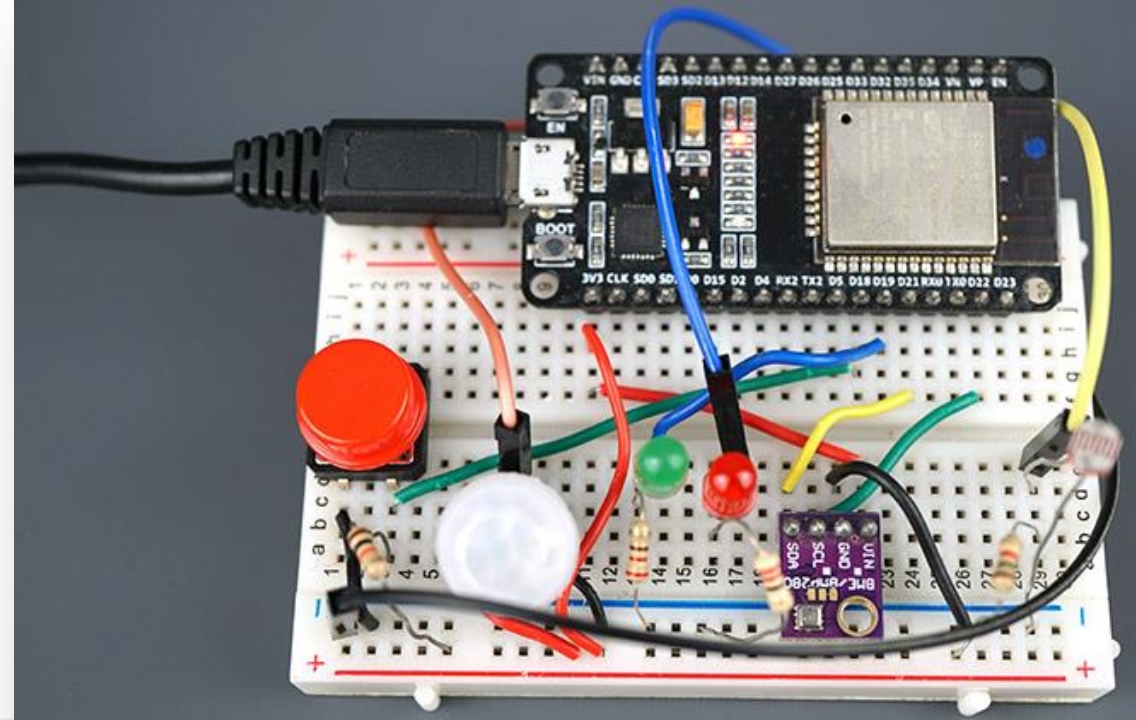
Como podemos integrar soluções?



ESP32

- Microcontrolador moderno
- Wi-Fi + Bluetooth
- Dual-core
- Alto desempenho

ESP32 NO MUNDO MAKER



- Programável pelo framework do Arduino
- Muito usado em IoT
- Baixo custo

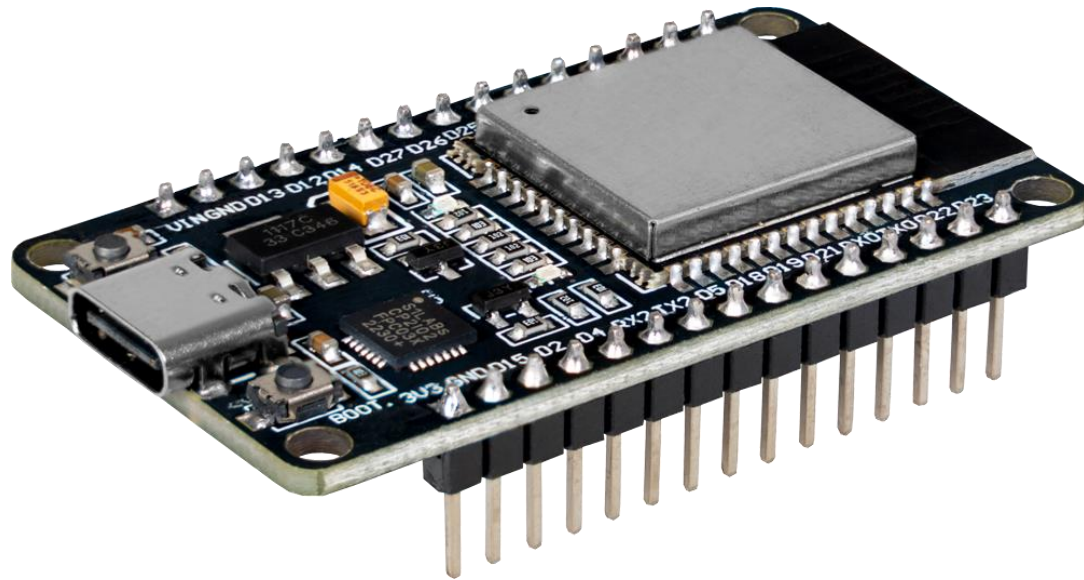
MAS EXISTE OUTRO
CAMINHO: ESP-IDF



ESP-IDF

O QUE É ESP-IDF

- Framework oficial da Espressif
- Controle total do hardware
- Mais robusto
- Padrão industrial



Critério	Arduino	ESP-IDF
Abstração	Alta	Baixa
Controle	Limitado	Total
Tempo de desenvolvimento	Rápido	Mais lento
Robustez	Média	Alta
Debug	Básico	Avançado
Uso típico	Protótipo	Produto

ESP32: ARDUINO vs ESP-IDF

- Análise comparativa

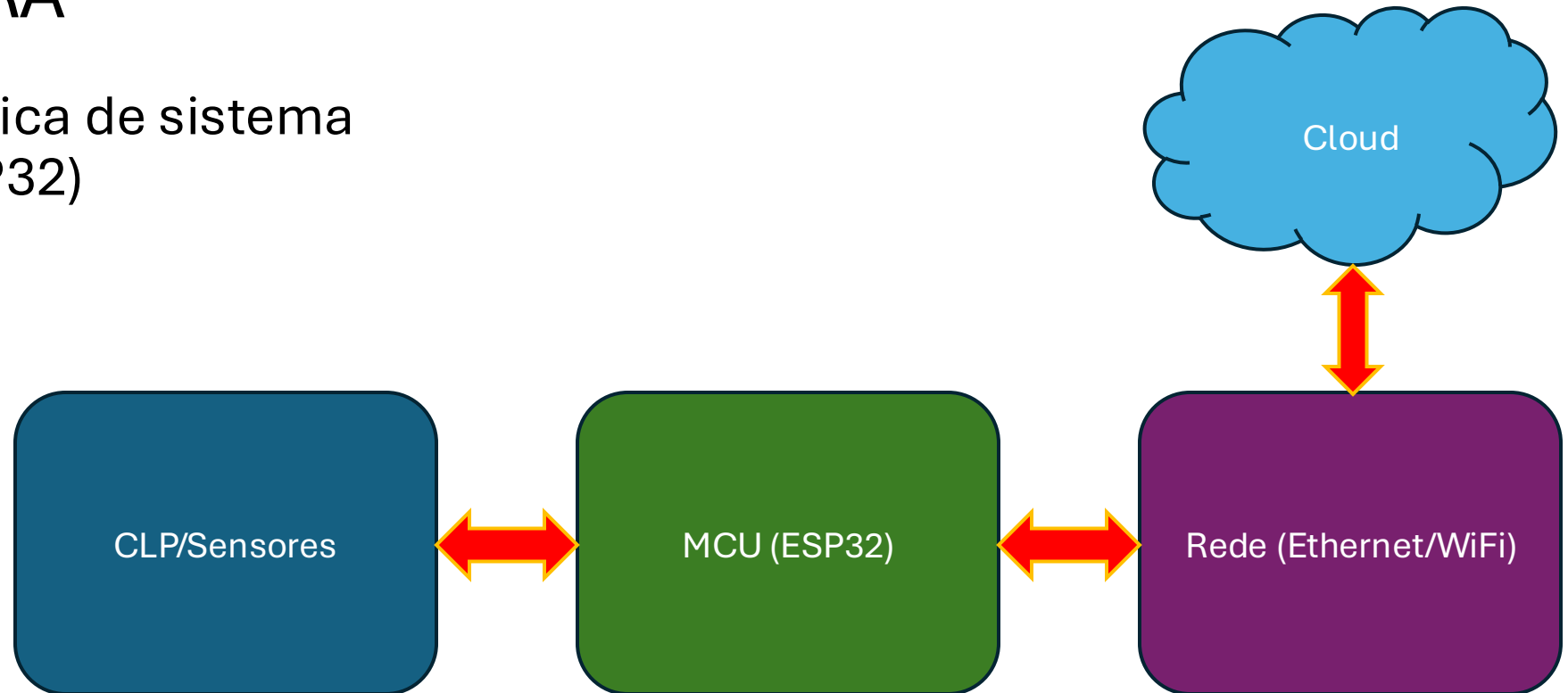


QUANDO USAR CADA UM

- Arduino:
 - Ensino
 - Prototipação rápida
- ESP-IDF:
 - Produto final
 - Sistemas críticos

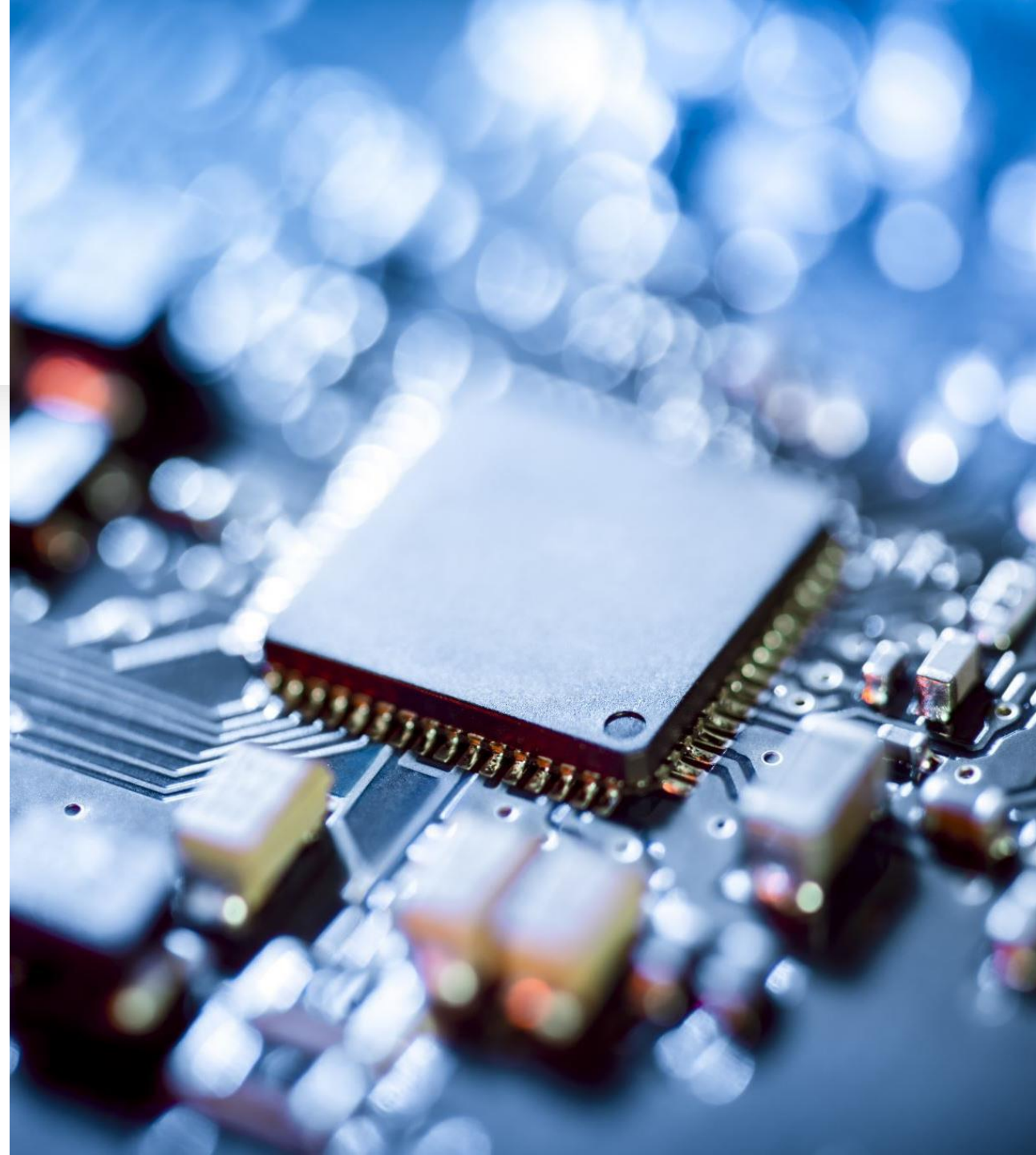
ARQUITETURA

- Arquitetura típica de sistema industrial (ESP32)



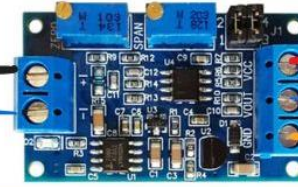
COMPONENTES DO SISTEMA

- Sensores
- Microcontrolador
- Comunicação
- Backend



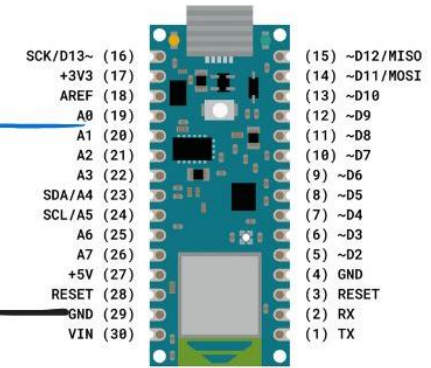


4-20mA to 0-5v



pressure sensor

9v battery



Arduino nano

PADRÕES DE ENTRADA

- Analógica:
 - 4-20mA
 - 0-10V
- Digital
 - Comandos (24Vdc ou 220Vac)

PROTOCOLOS

- MODBUS TCP
 - Padrão industrial
 - Comunicação com máquinas
- HTTP
 - Integração com servidores
- MQTT
 - Comunicação leve (IoT)
- Ethernet vs Wi-Fi
 - Alta confiabilidade

EtherCAT®

NopeN®

CC-LinkIE



EtherNet/IP™

DeviceNet™

PROFI®
NET

PROFI®
BUS



DESAFIOS REAIS

- **Ambiente industrial (realidade)**

- Ruído eletromagnético
- Quedas de energia
- Vibração constante
- Temperaturas elevadas
- Redes instáveis

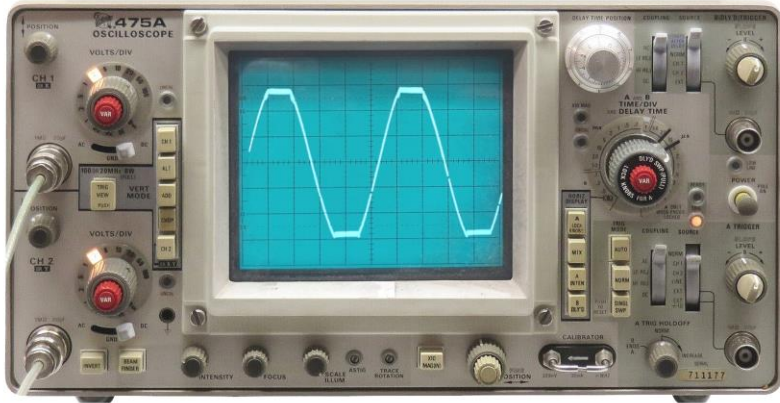
Protótipo funciona → campo falha



CONFIABILIDADE

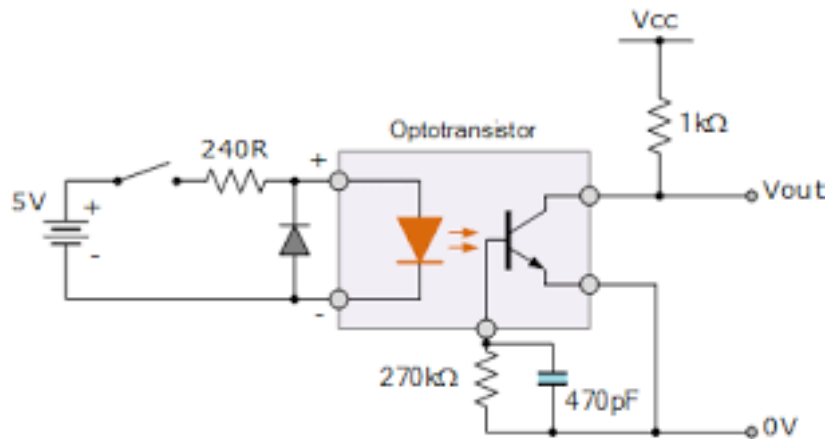
- **Estratégias de confiabilidade**
 - Watchdog Timer
 - Reconexão automática de rede
 - Buffer de dados offline
 - Logs persistentes
 - Fail-safe states



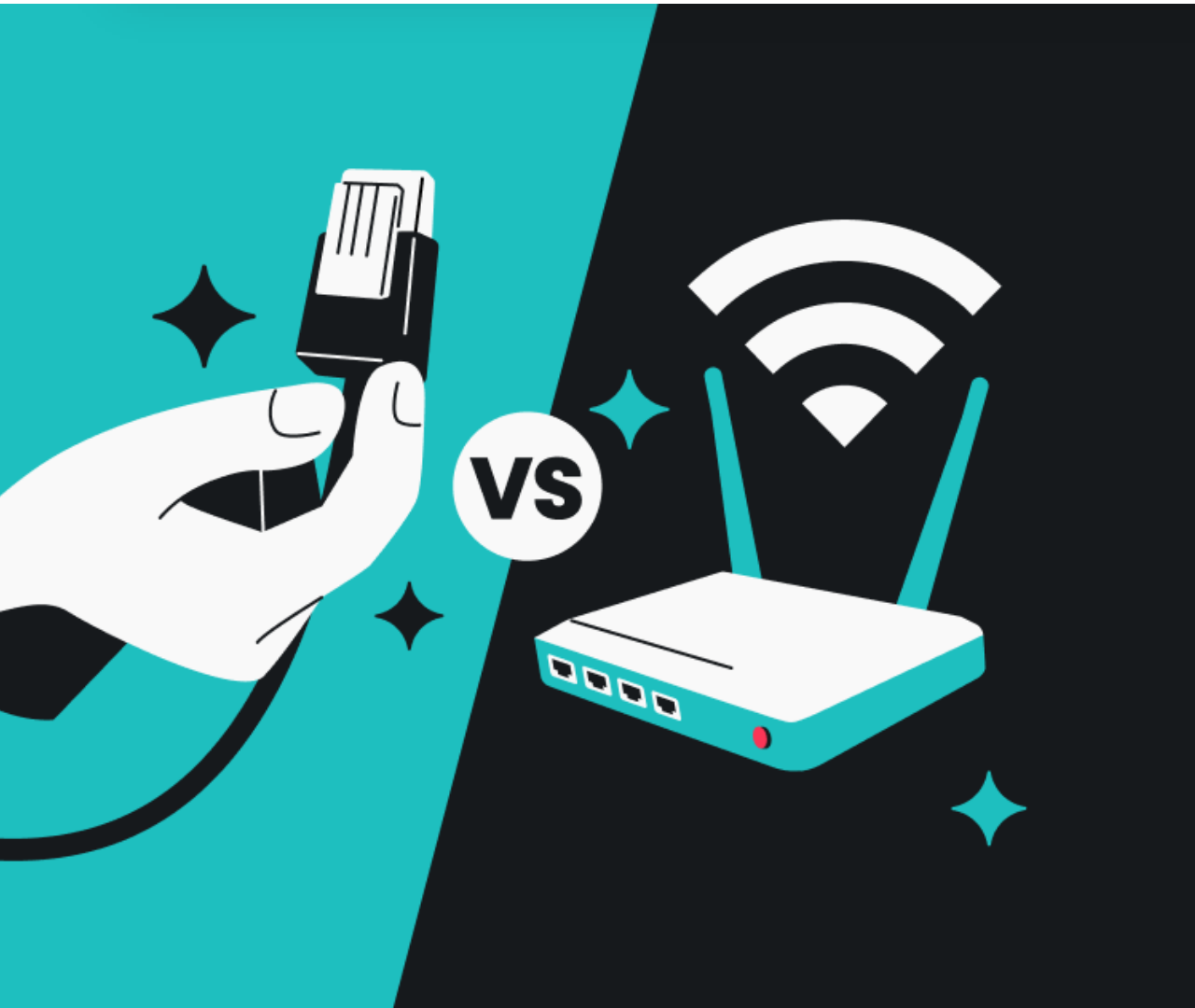


HARDWARE INDUSTRIAL

- Isolação
- Proteção elétrica
- Fonte estável



Wi-Fi vs Ethernet

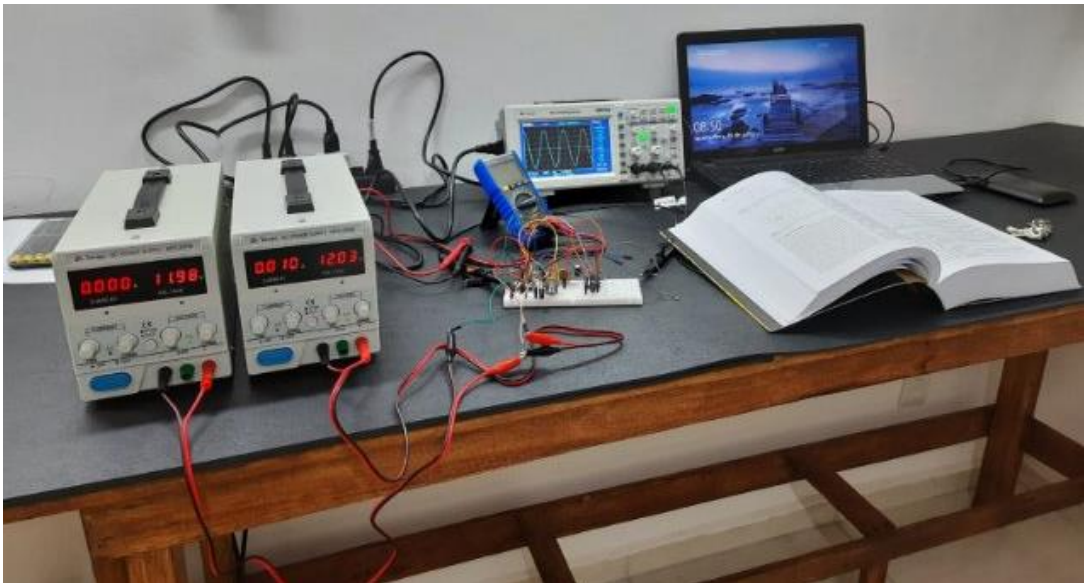
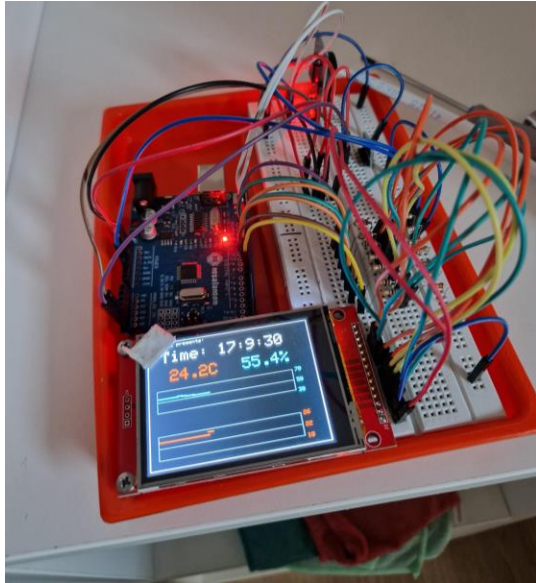
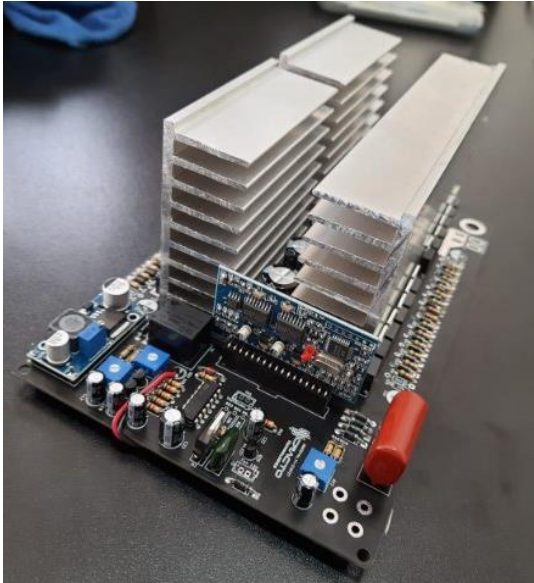


Critério	Wi-Fi	Ethernet
Estabilidade	Média	Alta
Latência	Variável	Baixa
Interferência	Alta	Baixa
Uso industrial	Limitado	Preferido

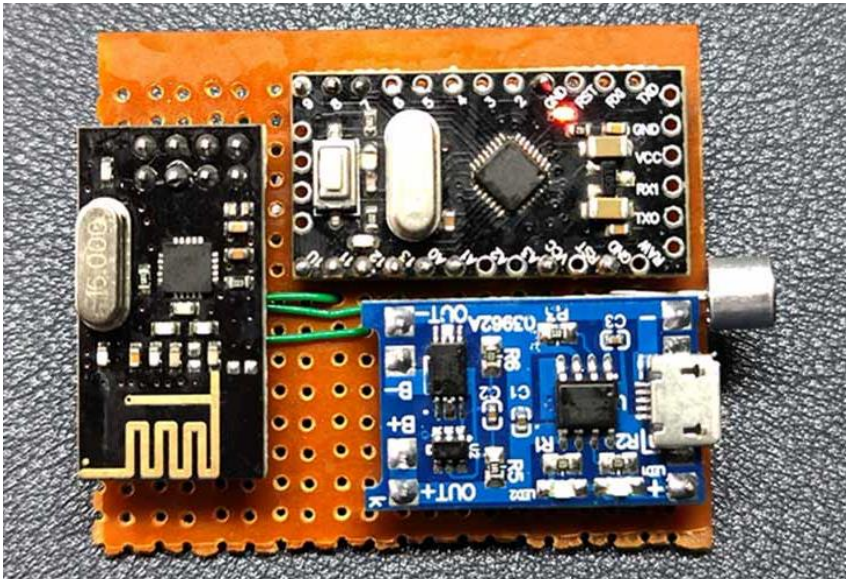
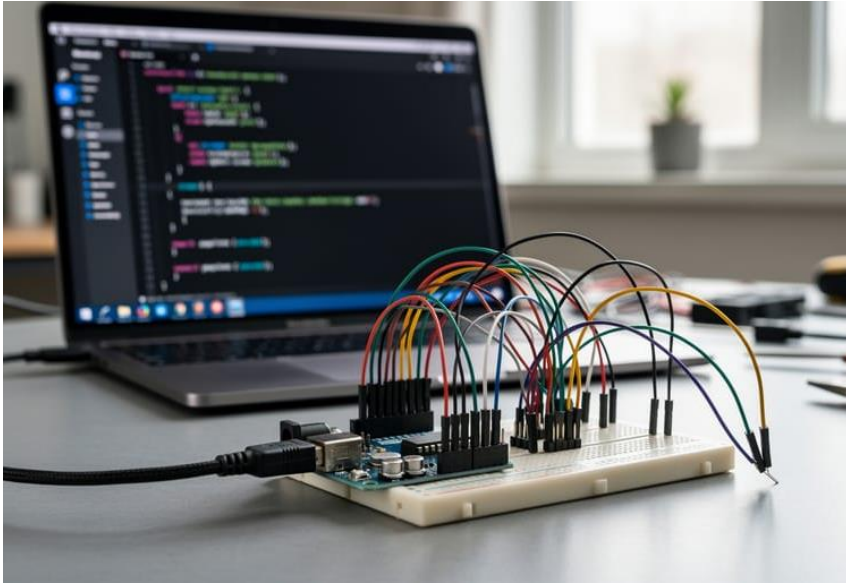
DO PROTÓTIPO AO PRODUTO

- Etapas:
 - Ideia
 - Protótipo (Arduino)
 - Validação
 - Reprojetado
 - Produto





Protótipos

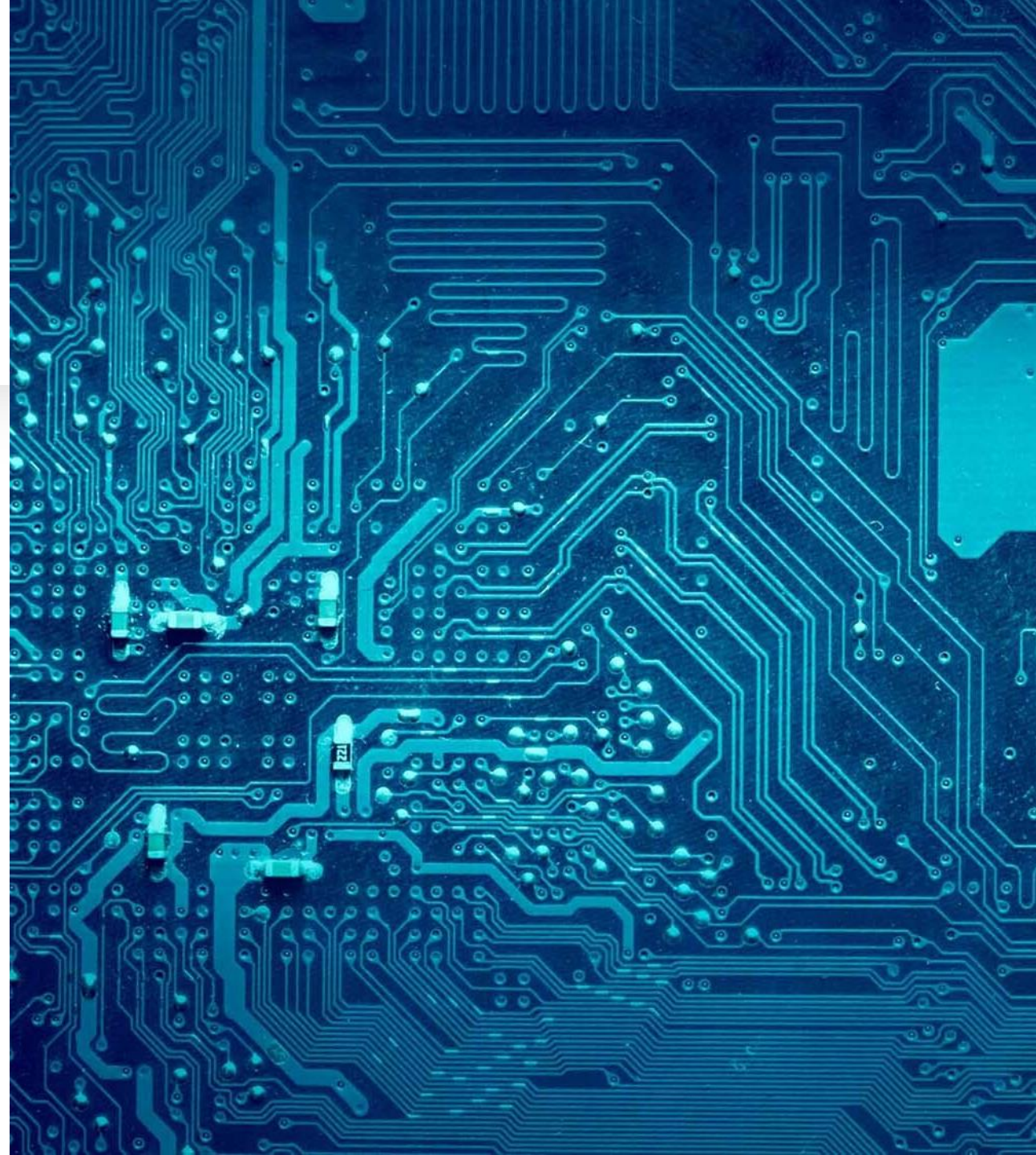


ERRO CLÁSSICO

Tentar escalar produto com
código Arduino direto

BOAS PRÁTICAS

- **Boas práticas de firmware**
 - Separação de camadas (lógica, comunicação)
 - Código modular
 - Tratamento de erro explícito
 - Logs estruturados





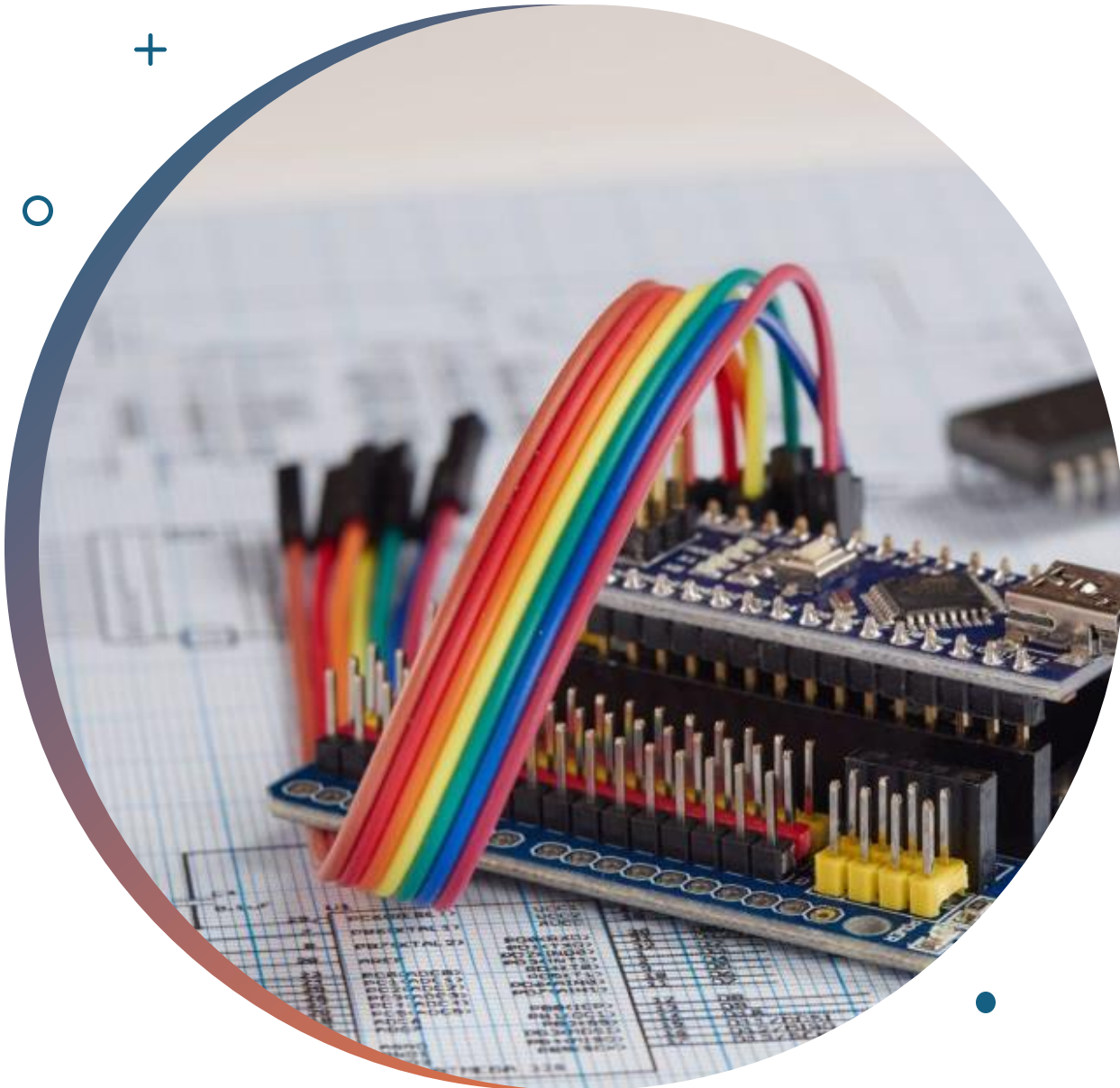
TEMPO REAL

- **FreeRTOS no ESP32**
 - Multitarefa real
 - Prioridades de execução
 - Tasks independentes:
 - Leitura de sensores
 - Comunicação
 - Processamento



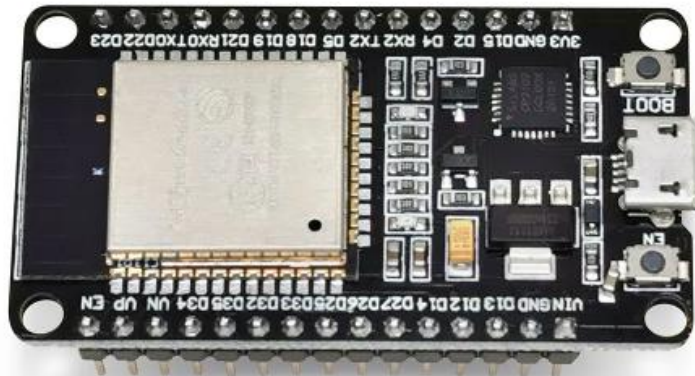
SEGURANÇA

- **Segurança em sistemas IoT industriais**
 - Autenticação (tokens/keys)
 - Criptografia (TLS)
 - Atualização remota (OTA)
 - Controle de acesso

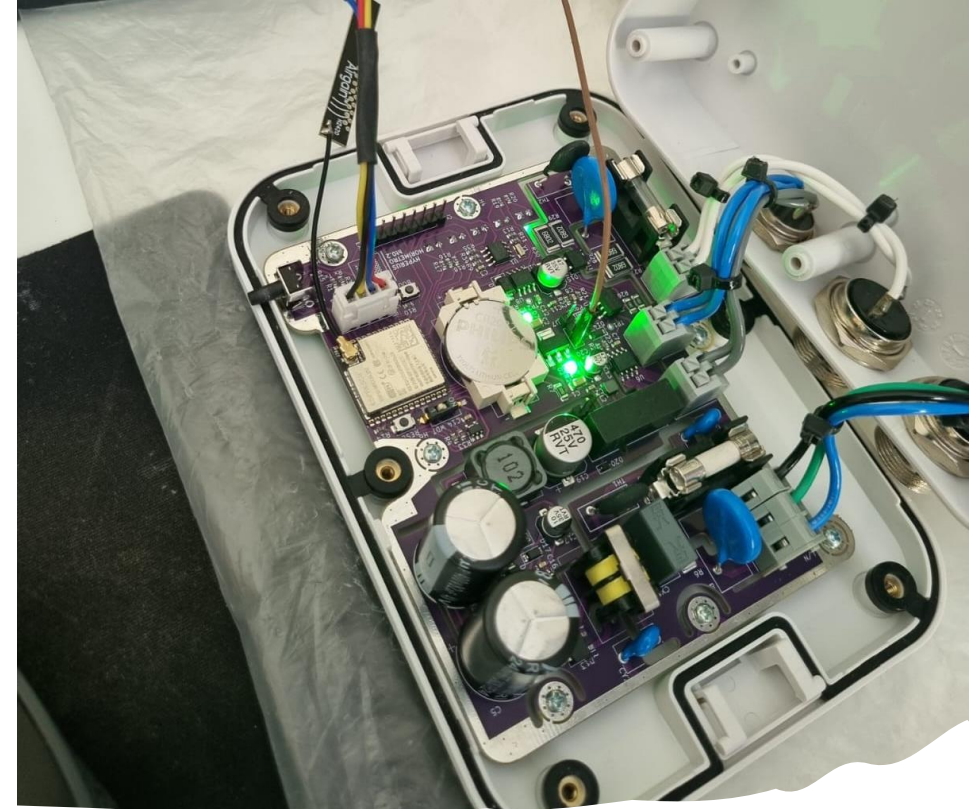
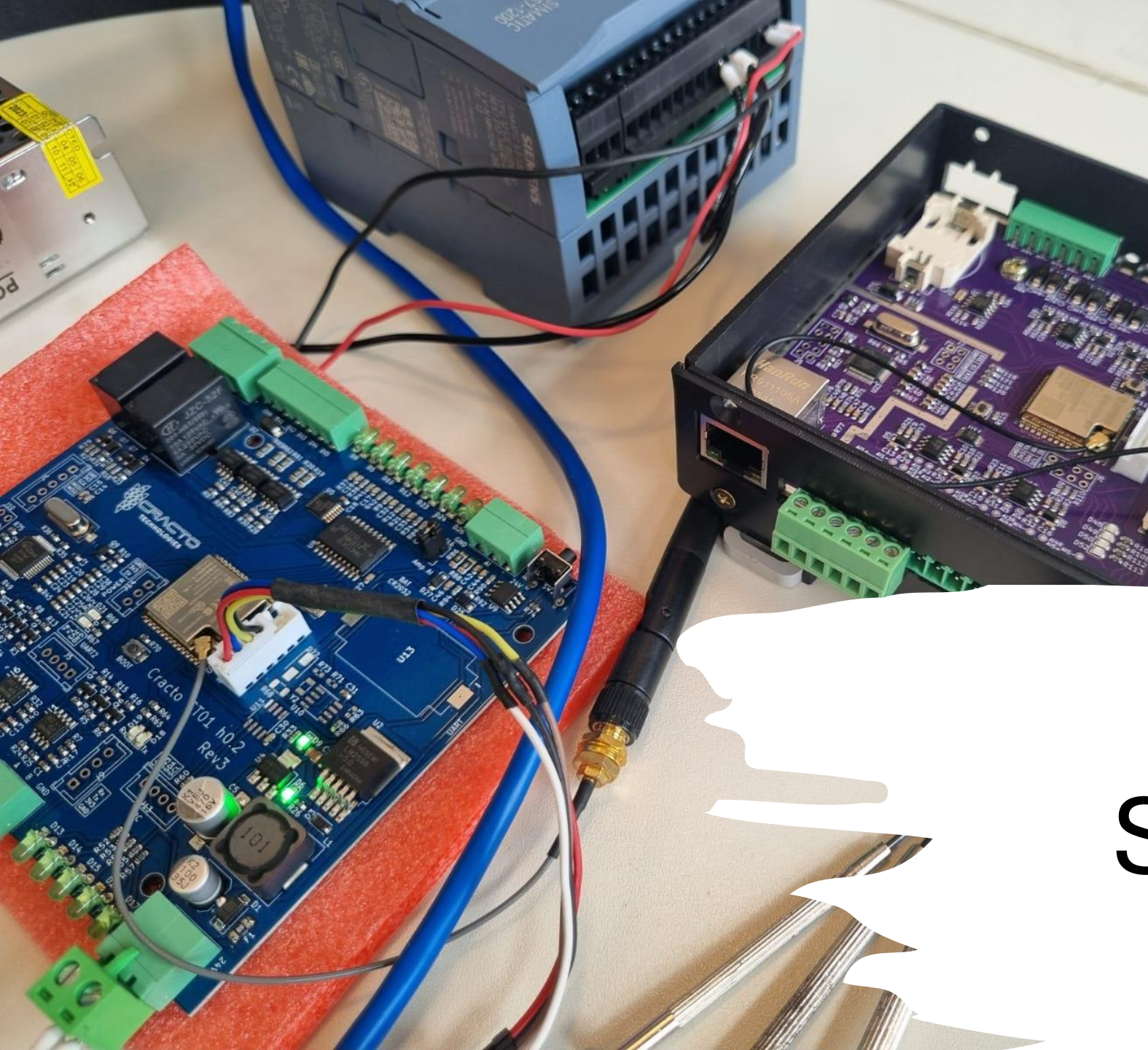


DICAS

- **Para quem está começando**
 - Comece com Arduino
 - Entenda comunicação (rede)
 - Aprenda arquitetura de software
 - Evolua para ESP-IDF
 - Pense em produto desde o início




É possível
desenvolver soluções
comerciais com
ESP32?



SIM

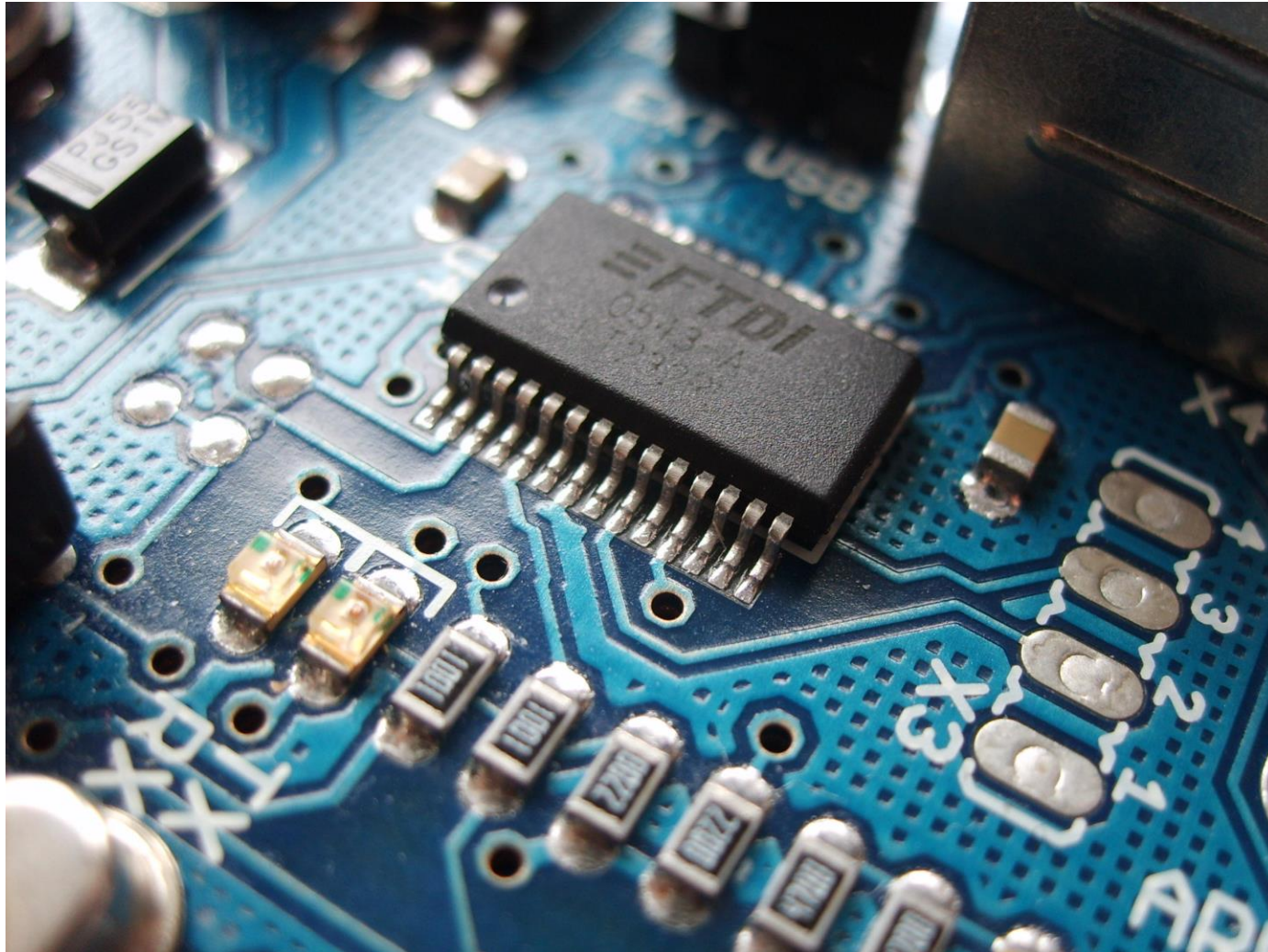
O que fazemos





Protótipos validam
ideias. Produtos geram
valor real.

Arduino é um excelente ponto de partida
para o engenheiro de produto.



Obrigado!