

# Microcontroladores no contexto de IoT

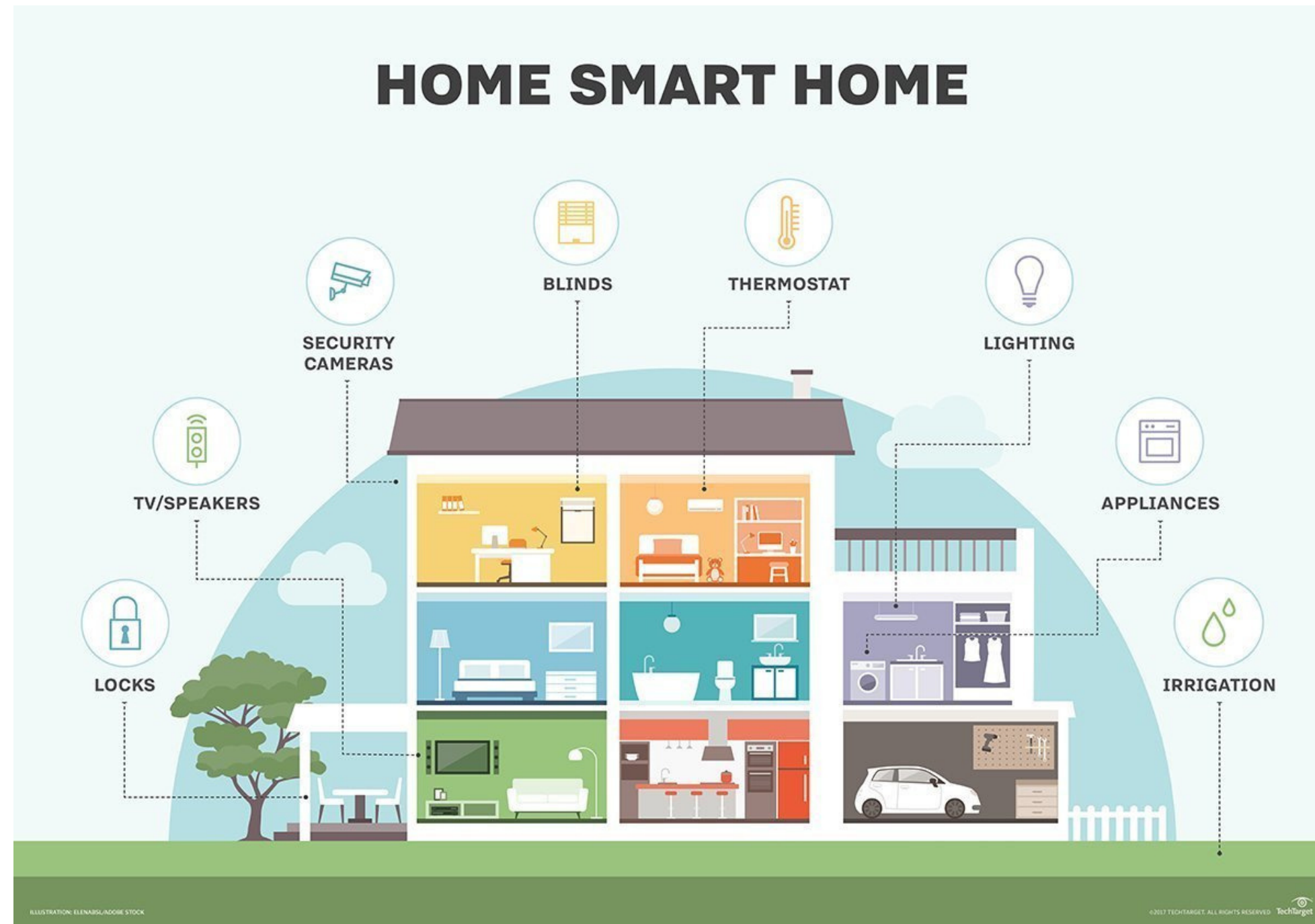
Empregando ESP32 e o sistema operacional FreeRTOS



# Objetivos da Aula

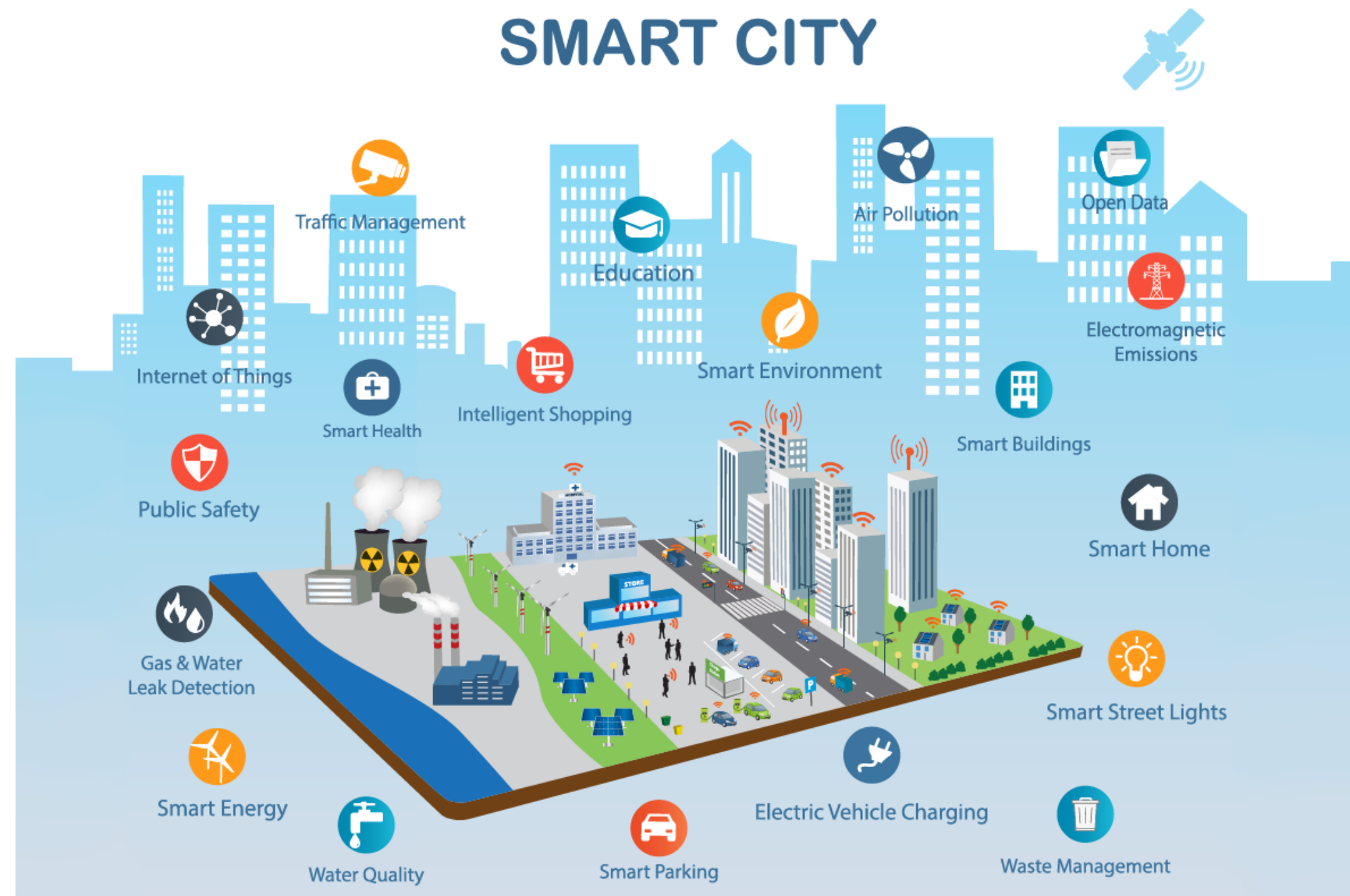
- Breve discussão de IoT e seus sistemas operacionais;
- Apresentar o microcontrolador ESP32;
- Empregar o microcontrolador na coleta de dados físicos do ambiente (como temperatura, umidade e pressão) através dos sensores DHT22 e BMP280;
- Introduzir a programação multitarefa no ESP32 através do sistema operacional FreeRTOS;

# Aplicações de IoT



## Smart Home

# Aplicações de IoT



## Smart City

# Aplicações de IoT



## Wearables

# Dispositivos IoT

## Microcontroller Unit (MCU)

Dispositivos de baixo poder computacional cujo principal propósito é o de gerenciar sensores e atuadores. Possuem capacidade limitada de processamento de dados.



**ATmega328** (Velocidade de clock de 16MHZ, 32 KB de memória flash, 2 KB de memória RAM)

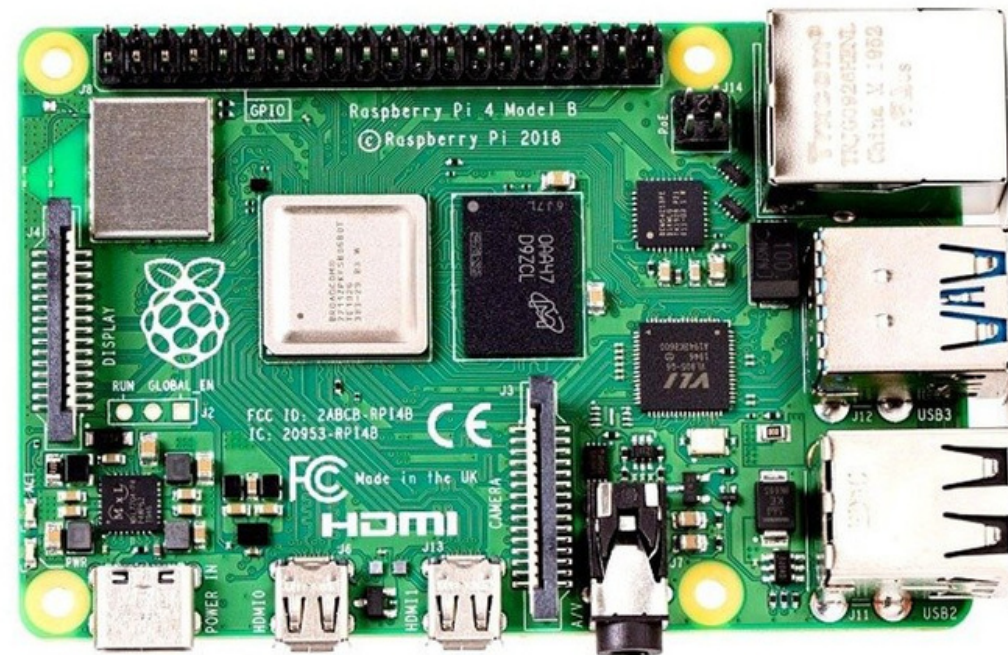
**ESP32** (CPU Xtensa® Dual-Core 32-bit, clock máximo de 240MHZ, 4 MB de memória flash, 520 KB de memória RAM, suporte microSD, conectividade WiFi 2.4 Ghz, Bluetooth, etc.)



# Dispositivos IoT

## Single Board Computer (SBC)

Dispositivos com maior poder computacional que, além de gerenciar sensores e atuadores, dispõem de maior capacidade para processamento local de dados, podendo aplicar muitas vezes algoritmos de machine learning e de deep learning.



**Raspberry Pi 4 B** (CPU Broadcom BCM2711 Quad-Core 64-bit 1.5 GHz, 1 GB - 8 GB de RAM, suporte a microSD, portas USB 2.0 e 3.0, portas HDMI com suporte a 4K, conector P2 para áudio, conectividade Gigabit Ethernet, WiFi 2.4/5.0 GHz, bluetooth 5, etc)

# Sistemas Operacionais IoT

## Single Board Computer (SBC)

Uma vez que possuem maior poder computacional, é possível instalar sistemas operacionais mais complexos, como os de propósito geral. Por exemplo, o Raspberry Pi OS, também conhecido como Raspbian, é baseado na distribuição Linux do Debian. Além dela, é possível instalar também outras distros Linux como o Ubuntu.





# Sistemas Operacionais IoT

## Microcontroller Unit (MCU)

Devido ao seu baixo poder computacional, os microcontroladores geralmente não possuem sistemas operacionais instalados. Geralmente, os programas são gravados com um conjunto de instruções de máquina suficientes para acessar os recursos do MCU. Ainda assim, é possível encontrar SOs para tais dispositivos, alguns baseados em Linux, como o Lite OS, outros não, como o Tiny OS, RIOT e Free RTOS.

No caso do ESP32, já existe uma build do sistema Free RTOS instalado nele, o que permite explorar a concorrência e paralelismo em programas multitarefas.

# Sistemas Operacionais IoT

## Free RTOS

O Free RTOS é um sistema operacional de tempo-real. Tais sistemas priorizam o tempo de resposta a determinados eventos, internos ou externos, garantindo que as tarefas sejam executadas em um intervalo de tempo pré-definido. Isso difere dos sistemas operacionais de propósito geral no qual a execução das tarefas não possui um intervalo de tempo determinado, sendo priorizada a experiência do usuário em vez do tempo de resposta.

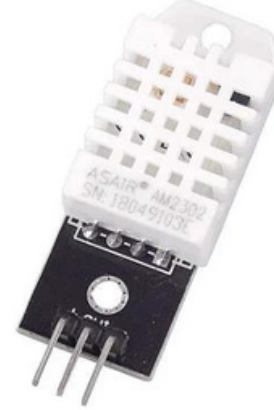
Outro conceito importante sobre os sistemas operacionais de tempo real é o escalonamento prioritário, no qual determinadas tarefas possuem prioridade na utilização do tempo de uso da CPU. Assim, o Free RTOS pode ser muito útil em situações que a leitura de um sensor é crítica. Além disso, é possível ainda explorar a concorrência e paralelismo entre múltiplas tarefas por meio dele.



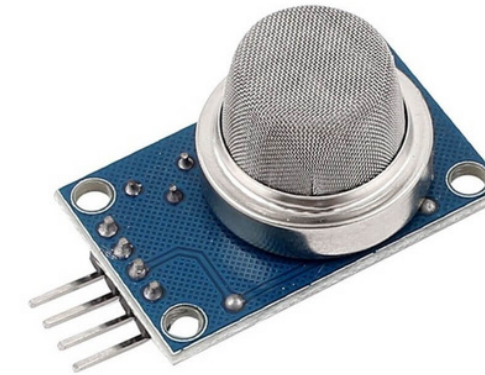
# Sistemas Operacionais IoT



**Sensor LDR**  
(Luminosidade)



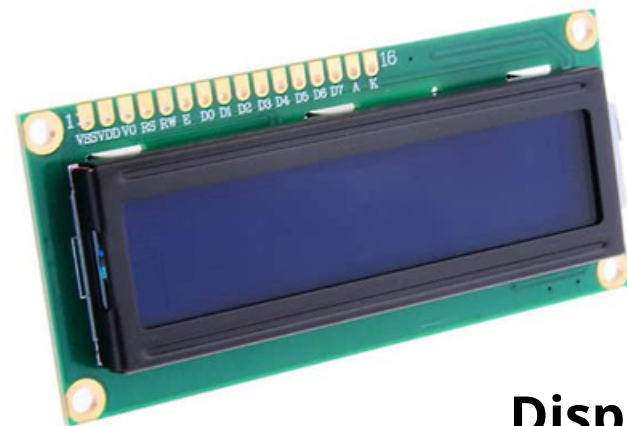
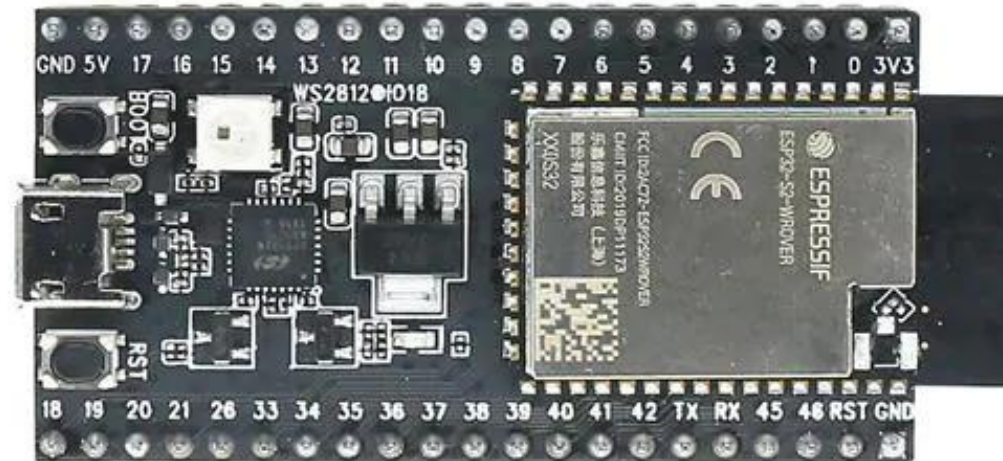
**Sensor DHT22**  
(Temperatura e Umidade)



**Sensor MQ-2**  
(Gás inflamável e Fumaça)



**Sensor PIR**  
(Aproximação)



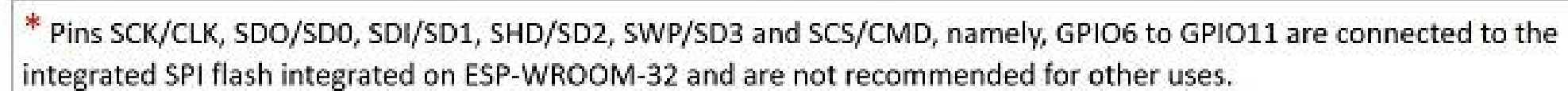
**Display LCD**



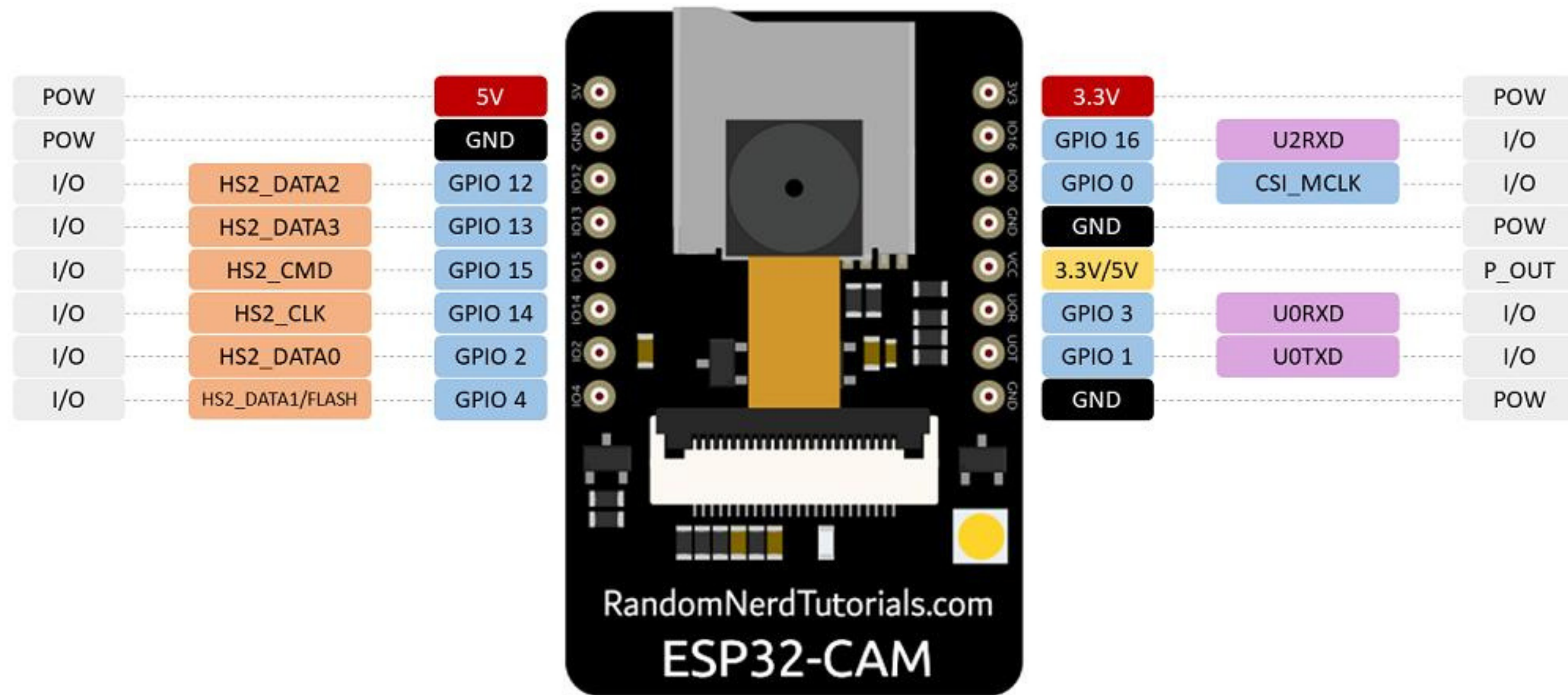
**LED**



### version with 36 GPIOs









## INTERFACING BMP280 AND ESP32

