

# Comunicação Serial

A comunicação serial UART (Universal Asynchronous Receiver/Transmitter) é um dos métodos mais antigos e amplamente utilizados para a transferência de dados entre dispositivos eletrônicos. Ela desempenha um papel crucial na interconexão de componentes em sistemas eletrônicos, permitindo a troca de informações confiável e eficiente. Neste guia, exploraremos o que é a comunicação serial UART, como ela funciona, sua importância e outros aspectos relevantes.

## O Que é Comunicação Serial UART?

A comunicação serial UART é uma técnica de comunicação de dados que envolve a transmissão e recepção de bits sequencialmente, um após o outro, através de um único par de fios. Ela é chamada de "assíncrona" porque os dispositivos envolvidos não precisam estar sincronizados no tempo, uma vez que os bits são acompanhados por informações de início e parada.

## Como Funciona a Comunicação Serial UART?

A comunicação serial UART é baseada em dois dispositivos, um transmissor (TX) e um receptor (RX). O transmissor converte os dados em uma sequência serial de bits e envia-os um a um através do fio de transmissão (TX). O receptor recebe os bits, decodifica-os e recria os dados originais.

Os elementos-chave da comunicação UART incluem:

1. **Bits de Dados:** Representam os próprios dados a serem transmitidos, geralmente 8 bits por caractere, mas podem variar.
2. **Bit de Start:** Indica o início de um caractere e sinaliza ao receptor que os dados estão chegando.
3. **Bits de Parada:** Indicam o final de um caractere e permitem ao receptor preparar-se para o próximo.
4. **Baud Rate:** A taxa de baud (baud rate) define a velocidade da transmissão, indicando quantos bits são transmitidos por segundo. É essencial que ambos os dispositivos estejam configurados com a mesma taxa de baud para uma comunicação bem-sucedida.

## Importância da Comunicação Serial UART:

A comunicação serial UART é uma pedra angular na eletrônica, sendo essencial em diversas aplicações, incluindo:

- **Conexão de Periféricos:** Dispositivos como teclados, mouses, sensores, displays e muitos outros são frequentemente conectados a microcontroladores ou computadores usando UART.
- **Comunicação entre Microcontroladores:** Muitos sistemas embarcados usam UART para permitir que microcontroladores se comuniquem uns com os outros.

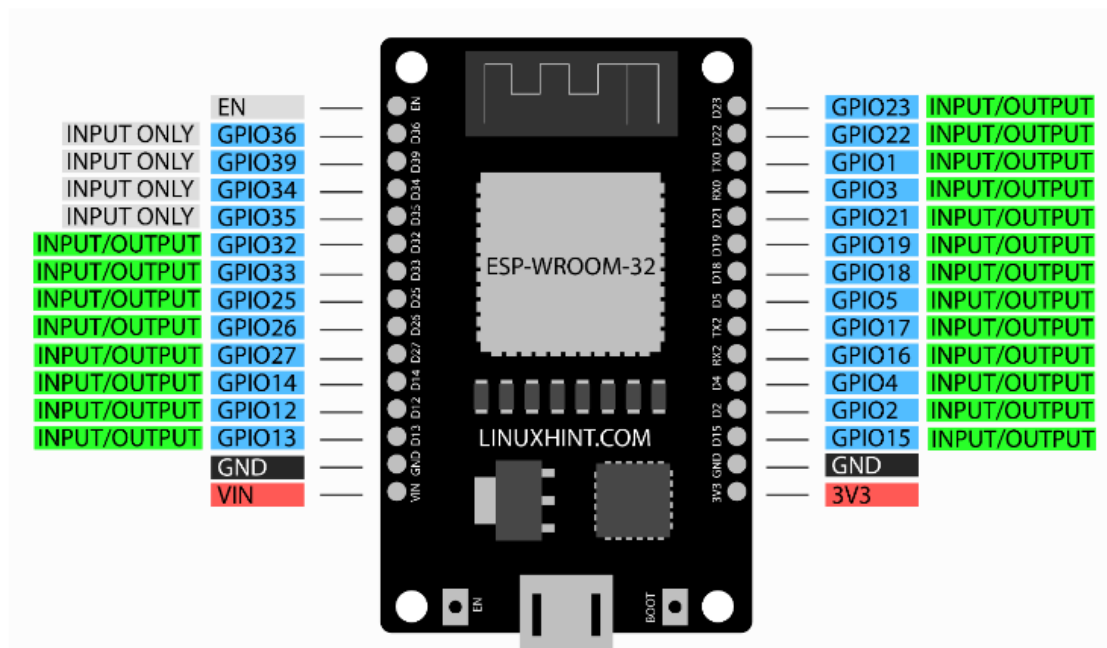
- **Programação e Depuração:** A UART é frequentemente usada para programação e depuração de microcontroladores e dispositivos eletrônicos.

### Configuração do Baud Rate:

A escolha adequada da taxa de baud é fundamental para o desempenho da comunicação UART. Taxas de baud mais altas permitem uma transmissão de dados mais rápida, mas também exigem uma precisão maior na sincronização dos dispositivos. Taxas comuns incluem 9600, 19200, 38400, 57600 e 115200 baud.

A comunicação serial UART é uma tecnologia robusta e amplamente utilizada que desempenha um papel vital na interconexão de dispositivos eletrônicos. Com sua capacidade de transmitir dados de forma confiável e eficiente, ela é uma escolha frequente em uma variedade de aplicações, desde sistemas embarcados até dispositivos de consumo. A compreensão dos princípios da comunicação serial UART é fundamental para qualquer pessoa envolvida no desenvolvimento de hardware ou software relacionado a eletrônica.

## GPIO



Os microcontroladores são componentes essenciais em sistemas embarcados, capazes de controlar e interagir com diversos dispositivos. Sua pinagem desempenha um papel crucial, determinando como eles se conectam e se comunicam com o ambiente externo. Neste guia, abordaremos os principais aspectos dos pinos de microcontroladores, incluindo suas funções, limitações de corrente e as diferenças entre GPIO (General-Purpose Input/Output), entradas (input) e saídas (output).

### Funções dos Pinos de Microcontroladores:

1. **GPIO (General-Purpose Input/Output):** Os pinos GPIO são a espinha dorsal dos microcontroladores. Eles podem ser configurados como entrada ou saída e desempenham um papel fundamental na interação com dispositivos externos. Os pinos de saída podem fornecer níveis de tensão para controlar LEDs, motores, relés e outros componentes. Já os pinos de entrada podem ler informações de sensores, botões e outros dispositivos.
2. **Entradas Analógicas:** Alguns microcontroladores possuem pinos dedicados a entradas analógicas. Eles permitem a medição de sinais analógicos, como sensores de temperatura ou luz. Esses sinais são convertidos em valores digitais para serem processados pelo microcontrolador.
3. **Comunicação Serial (UART, SPI, I2C):** Muitos microcontroladores têm pinos dedicados para comunicação serial, como UART, SPI e I2C. Esses pinos são usados para conectar o microcontrolador a outros dispositivos, como sensores, displays e módulos de comunicação sem fio.
4. **Interrupções (Interrupts):** Alguns pinos podem ser configurados para gerar interrupções, permitindo que o microcontrolador seja notificado imediatamente quando ocorre uma mudança em um pino específico. Isso é útil para lidar com eventos em tempo real.

#### Limitações de Corrente:

Os pinos de microcontroladores têm limitações de corrente que devem ser respeitadas para evitar danos ao dispositivo. Geralmente, essas limitações incluem:

- **Corrente de Saída (Source Current):** É a quantidade máxima de corrente que um pino pode fornecer quando configurado como saída. Exceder esse valor pode danificar o microcontrolador.
- **Corrente de Dreno (Sink Current):** É a quantidade máxima de corrente que um pino pode absorver quando configurado como entrada ou como saída em estado baixo. Exceder essa corrente também pode causar danos.

#### Diferenças entre GPIO, Entradas e Saídas:

- **GPIO (General-Purpose Input/Output):** Esses pinos podem ser configurados como entrada ou saída, permitindo uma ampla gama de aplicações. Como entrada, eles podem ler o estado de dispositivos externos, como sensores. Como saída, eles podem controlar dispositivos, como LEDs.
- **Entradas (Input):** Os pinos configurados como entrada são usados para receber informações do ambiente externo. Eles podem ler sinais, como níveis de tensão, para determinar se um botão foi pressionado ou para medir a temperatura de um sensor.
- **Saídas (Output):** Os pinos configurados como saída são usados para controlar dispositivos externos. Eles podem fornecer níveis de tensão ou corrente para acionar LEDs, relés, motores e outros componentes.

Em resumo, os pinos de microcontroladores são elementos fundamentais na interação entre o microcontrolador e seu ambiente. Compreender suas funções, limitações de

corrente e como configurá-los corretamente é essencial para o desenvolvimento de sistemas embarcados confiáveis e eficazes.

## Resistores de Pull-Up

Os resistores de pull-up são componentes eletrônicos comumente usados em circuitos eletrônicos para garantir que um sinal elétrico assuma um valor predefinido, geralmente alto (ou seja, um nível lógico "1"). Eles são frequentemente empregados em microcontroladores, microprocessadores e outros dispositivos eletrônicos digitais. Aqui estão algumas informações importantes sobre resistores de pull-up:

1. **Finalidade Principal:** A principal finalidade dos resistores de pull-up é garantir que um pino (ou linha) de sinal em um circuito digital assuma um valor lógico alto quando não está ativamente sendo conduzido para um valor lógico baixo. Isso é particularmente útil para evitar flutuações indesejadas nos níveis de sinal.
2. **Configuração:** Um resistor de pull-up é geralmente conectado entre o pino de sinal e a tensão de alimentação (VCC ou 5V, por exemplo). Quando não há uma fonte ativa para puxar o pino para o terra (GND), o resistor pull-up faz isso, mantendo o sinal em um estado alto. Quando o dispositivo conectado ao pino de sinal é ativado e puxa o pino para GND, ele anula o efeito do resistor pull-up e o sinal assume o valor lógico baixo.
3. **Valores de Resistência:** O valor da resistência do pull-up é selecionado de acordo com as especificações do circuito e a carga elétrica envolvida. Valores comuns são 4.7 kΩ e 10 kΩ, mas eles podem variar dependendo do aplicativo.
4. **Uso em Comunicação:** Em comunicação serial, como I2C ou SPI, os resistores pull-up são frequentemente usados nas linhas de dados (DAS e SCL) para garantir que os sinais sejam mantidos em níveis altos quando os dispositivos não estiverem transmitindo ativamente.
5. **Economia de Energia:** Os resistores pull-up ajudam a economizar energia, uma vez que a corrente flui através deles apenas quando o sinal é puxado para o nível baixo. Quando o sinal está alto, a corrente é mínima.
6. **Desvantagens:** Em circuitos de alta velocidade, resistores pull-up podem afetar o tempo de subida do sinal, o que pode ser crítico em algumas aplicações. Nesses casos, é importante escolher resistores com valores apropriados e considerar outras técnicas de projeto.

Em resumo, os resistores de pull-up são componentes eletrônicos fundamentais que desempenham um papel importante na garantia de níveis de sinal consistentes e na prevenção de flutuações indesejadas em circuitos digitais. Eles são especialmente úteis em sistemas com microcontroladores e microprocessadores para estabelecer estados lógicos claros em linhas de sinal.

Você não precisa se preocupar em conectar um pull-up externo na ESP32 pois ela possui pinos com essa configuração internamente: Os GPIO's que possuem pull-up internos são:

Pin Number	Pin Name	Built-in Pull-Up Resistor
0	GPIO0	Yes
2	GPIO2	Yes
4	GPIO4	Yes
5	GPIO5	Yes
12	GPIO12	Yes
13	GPIO13	Yes
14	GPIO14	Yes
15	GPIO15	Yes
25	GPIO25	Yes
26	GPIO26	Yes
27	GPIO27	Yes
32	GPIO32	Yes
33	GPIO33	Yes
34	GPIO34	No
35	GPIO35	No
36	GPIO36	No
39	GPIO39	No

Para acionar um resistor de pull-up interno, você vai utilizar o seguinte comando:

```
pinMode(pinoBotao, INPUT_PULLUP);
```

## Relés

Relés são dispositivos eletromecânicos utilizados para controlar circuitos elétricos por meio de um interruptor operado eletricamente. Eles desempenham um papel importante em várias aplicações, permitindo que sinais elétricos de baixa tensão controlem cargas de alta tensão ou corrente. Aqui estão algumas informações essenciais sobre relés:

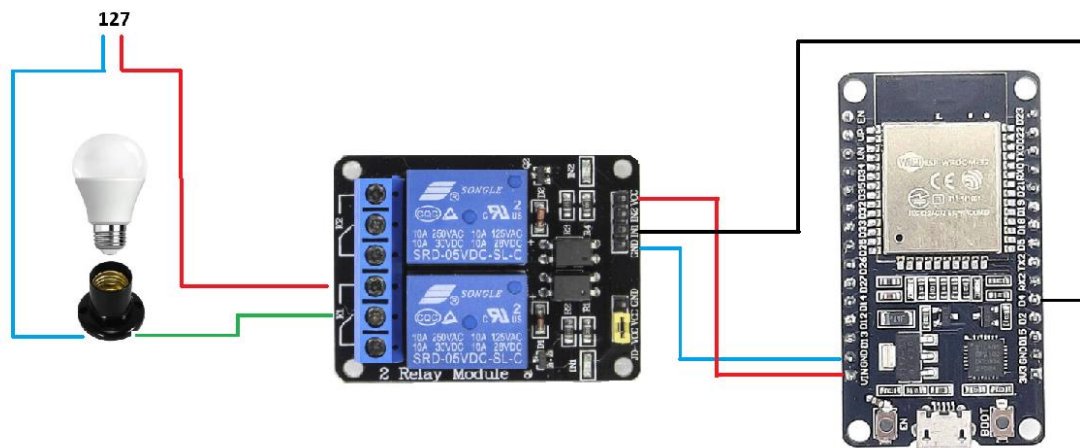
1. **Funcionamento Básico:** Um relé consiste em um interruptor que pode ser ativado ou desativado por uma corrente elétrica em um circuito de baixa tensão. Ele possui dois circuitos elétricos separados: o circuito de controle e o circuito de carga.

2. **Circuito de Controle:** O circuito de controle é composto por uma bobina de fio enrolado ao redor de um núcleo ferromagnético. Quando uma corrente é aplicada a essa bobina, ela cria um campo magnético que atrai uma alavanca ou uma série de contatos metálicos móveis no relé.
3. **Circuito de Carga:** O circuito de carga é onde a carga de alta tensão ou alta corrente é conectada. Quando os contatos do relé são movidos pela ativação da bobina de controle, eles fecham ou abrem o circuito de carga, permitindo ou interrompendo o fluxo de corrente para a carga.
4. **Tipos de Relés:** Existem vários tipos de relés, incluindo relés de propósito geral, relés de tempo, relés de segurança, relés de estado sólido e relés de proteção. Cada tipo é projetado para atender a requisitos específicos em diferentes aplicações.
5. **Aplicações Comuns:** Relés são usados em uma ampla variedade de aplicações, incluindo automação industrial, sistemas de controle de iluminação, sistemas de alarme, sistemas de proteção contra sobrecarga em motores elétricos, controle de temperatura, telecomunicações, sistemas automotivos e muito mais.
6. **Vantagens dos Relés:** Relés têm várias vantagens, incluindo a capacidade de isolar circuitos de controle de circuitos de carga, permitindo o controle de cargas de alta potência usando um sinal de baixa potência, e a capacidade de fornecer proteção contra sobrecarga e curto-circuito em circuitos.
7. **Desvantagens dos Relés:** Embora sejam amplamente usados, os relés também têm desvantagens, como desgaste mecânico ao longo do tempo devido à operação mecânica, tempo de resposta limitado em comparação com dispositivos de estado sólido, e a possibilidade de faíscas ou arcos elétricos durante a comutação de cargas de alta potência.
8. **Relés de Estado Sólido:** Para superar algumas das limitações dos relés eletromecânicos, os relés de estado sólido são usados em muitas aplicações. Eles usam componentes semicondutores, como transistores, para realizar a comutação, eliminando assim os contatos mecânicos e oferecendo maior confiabilidade e velocidade de comutação.

Em resumo, relés são componentes fundamentais em eletrônica e automação, permitindo o controle seguro de cargas de alta tensão ou corrente por meio de sinais elétricos de baixa potência. Eles desempenham um papel essencial em uma variedade de aplicações industriais, comerciais e domésticas.

Vale lembrar que a ESP32 não tem os requisitos de corrente para acionar o relé, por isso, é interessante utilizar um módulo com um Optoacoplador ou um transistor para realizar

o acionamento. Em nossa aula, realizamos o acionamento de uma lâmpada com o relé:



## PWM

A Modulação por Largura de Pulso (PWM), em inglês Pulse Width Modulation, é uma técnica amplamente utilizada em eletrônica para controlar a intensidade média de um sinal elétrico, gerando uma saída analógica eficaz a partir de uma saída digital. Neste guia, exploraremos o que é PWM, como funciona, suas aplicações comuns e como implementá-lo.

### O Que é PWM?

PWM é uma técnica que varia a largura dos pulsos de um sinal digital, mantendo a frequência constante. Os pulsos mais largos representam um valor mais alto e os pulsos mais estreitos representam um valor mais baixo. Ao ajustar a largura dos pulsos em um ciclo, é possível controlar a média do sinal resultante.

### Como Funciona o PWM?

O PWM é frequentemente usado para controlar dispositivos que exigem ajustes de intensidade, como motores, lâmpadas, ventiladores e servomotores. Aqui está como funciona:

1. **Frequência de PWM:** Primeiro, você define uma frequência de PWM, que determina a taxa de repetição dos pulsos. A frequência é constante durante todo o ciclo.
2. **Duty Cycle (Ciclo de Trabalho):** O Duty Cycle, também chamado de razão de marcação, é a porcentagem do tempo em que o sinal está em nível alto (pulso largo) durante um ciclo completo. Por exemplo, um Duty Cycle de 50% significa que o sinal está ligado metade do tempo e desligado metade do tempo.
3. **Valor Médio:** O valor médio do sinal PWM é calculado multiplicando-se o Duty Cycle pela amplitude do sinal. Quanto maior o Duty Cycle, maior será o valor médio.

4. **Controle da Intensidade:** O dispositivo conectado ao sinal PWM interpreta o valor médio como uma intensidade. Por exemplo, um motor pode girar mais rápido ou mais devagar, dependendo do valor médio do sinal PWM aplicado.

#### Aplicações Comuns do PWM:

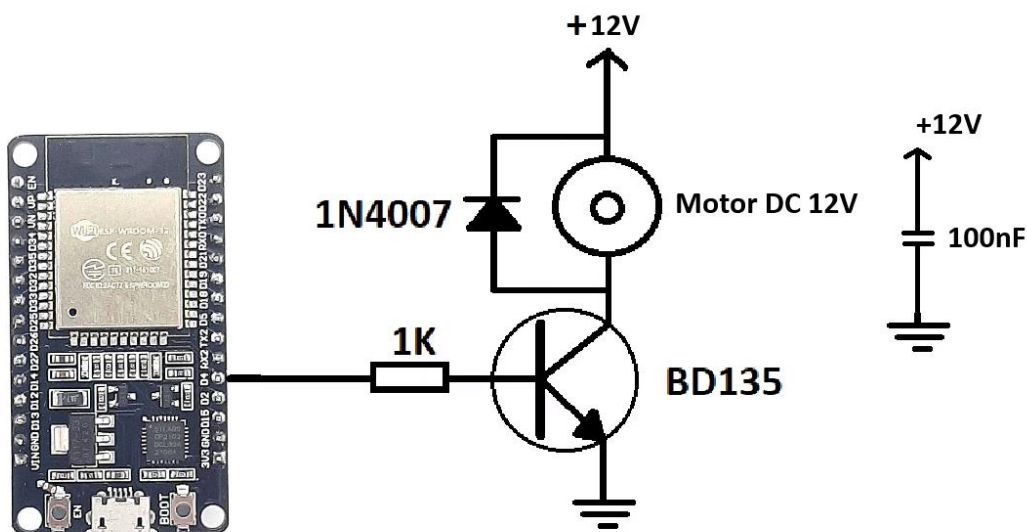
1. **Controle de Motores:** PWM é amplamente usado para controlar a velocidade e a direção de motores elétricos, tornando-o essencial em robótica e automação.
2. **Controle de Brilho:** É utilizado para ajustar o brilho de LEDs, lâmpadas e displays.
3. **Controle de Temperatura:** PWM é usado em sistemas de controle de temperatura para ligar e desligar dispositivos de aquecimento ou resfriamento com precisão.
4. **Controle de Servomotores:** Servomotores utilizam PWM para definir a posição do eixo com alta precisão.
5. **Controle de Áudio:** Em amplificadores de áudio digitais, PWM é usado para amplificar e modular o sinal de áudio.

#### Implementação do PWM:

A implementação do PWM geralmente envolve hardware dedicado, como timers e comparadores em microcontroladores ou circuitos integrados específicos. Além disso, muitos microcontroladores modernos têm módulos PWM embutidos, simplificando sua utilização.

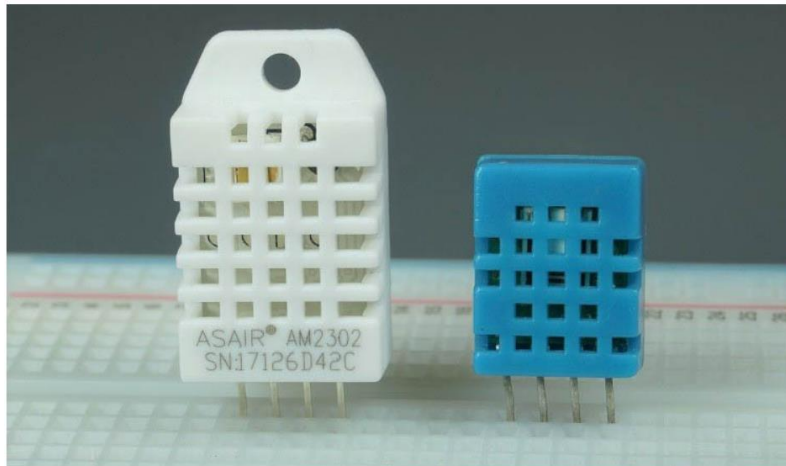
A frequência de PWM e o Duty Cycle podem ser ajustados conforme necessário para atender às especificações da aplicação.

Em nossa aula, fizemos o controle de velocidade de um motor DC, onde foi necessário ampliar a tensão do sinal para 12V. Para fazer isso, utilizamos um transistor que receberá o sinal de 3,3V na base e vai ampliá-lo para a tensão da fonte externa (12V). Se os requerimentos de corrente e tensão fossem compatíveis aos GPIO's da placa, você poderia fazer a conexão diretamente também, sem o auxílio de um transistor:





## Sensor de Temperatura DHT11

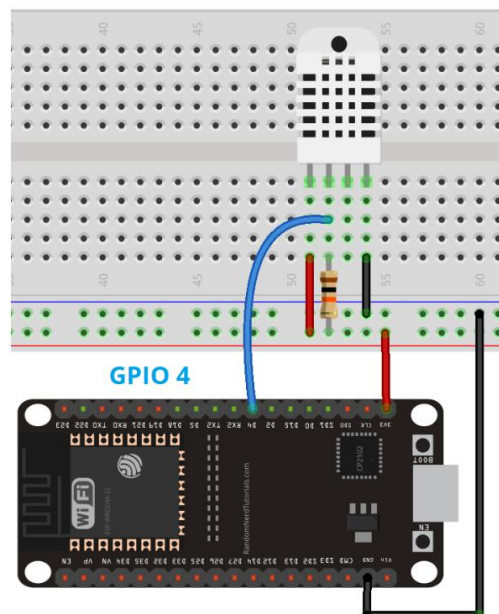


Fonte: <https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-sensor-arduino-ide/>

O sensor DHT11 é um dispositivo amplamente utilizado para medir temperatura e umidade relativa em projetos de Internet das Coisas (IoT). Ele é um sensor relativamente simples de usar e oferece informações valiosas para muitas aplicações de IoT, como monitoramento de ambientes, controle de climatização, gerenciamento de estufas e muito mais.

**O sensor deve ser alimentado com 3,3V para que a tensão de comunicação dele seja de 3,3V, afim de proteger o GPIO da ESP32.**

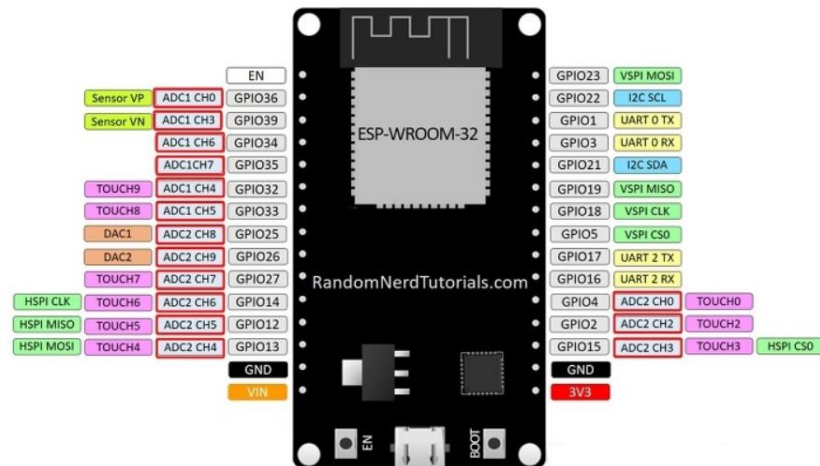
A cada leitura, o DHT11 formata os dados em um pacote de 40 bits e disponibiliza em seu canal de comunicação. A ESP32 possui várias bibliotecas que decifram este pacote de dados, como a: <https://registry.platformio.org/libraries/hwspeedy/DHT-Sensor>



Fonte: <https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-sensor-arduino-ide/>

## Pinos ADC

Um conversor analógico-digital (CAD), também conhecido como ADC (do inglês "Analog-to-Digital Converter"), é um dispositivo eletrônico que converte um sinal analógico em uma representação digital equivalente. Isso é fundamental em eletrônica, pois permite que sistemas digitais processem informações analógicas, como sinais de sensores, áudio ou vídeo. Os canais são representados por ADC1 e ADC2. A ESP32 suporta medidas em 18 canais diferentes, apenas 15 estão disponíveis no DevKit v1.



**Lembre-se que a tensão máxima deve ser de 3,3V.** A ESP32 transforma este valor analógico em um digital de 12 bits (0 a 4095) para que você possa utilizá-lo para o que quiser no código. Não é possível utilizar os canais ADC2 junto com o WiFi da placa, dê preferência para os canais ADC1.