**AULA 1 – Raspberry**

**REDES e IoT**

**Raspberry Pi Pico**

Microcontrolador: Equipamento sem SO (Sistema Operacional) embarcado. Depende de uma máquina para ser configurado. Ex: **Raspberry Pi Pico**

Microprocessador: Equipamento com SO embarcado. Pode ser configurado nele mesmo. Ex: **Raspberry Pi**

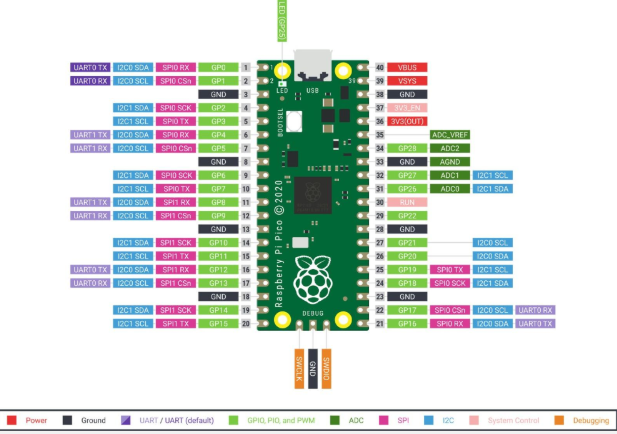
**Especificações do dispositivo (Raspberry Pi Pico 2020):**

* 2 núcleos ARM Cortex-M0+ a até 133 MHz
* 264 KB de RAM
* 2 MB de memória flash
* 26 pinos GPIO (entrada/saída digitais e analógicas)
* Comunicação via USB, UART, I2C, SPI
* Temporizadores, PWM, ADC, watchdog, PIO
* Pode ser programado em C/C++ ou MicroPython, que é uma versão enxuta do Python feita para microcontroladores (Contém comandos específicos para microcontroladores).

Vídeo Raspberry físico: <https://www.youtube.com/watch?v=qYqUJC9z2_E>

Link do simulador: <https://wokwi.com/pi-pico>

Cores e pinos do Raspberry:

Caixinhas: funções das portas (Todas fazem tanto INPUT quanto OUTPUT)

Pinos:

🔴 Power (Alimentação de Energia)

⚫ Ground (Terra / GND)

🟪 UART

🟩 GPIO, PWM e PIO

**🟩** ADC (Analog to Digital Converter)

🔵 SPI (Serial Peripheral Interface)

🟦 I2C (Inter-Integrated Circuit)

🟨 System Control

🟧 Debugging

**Power (Alimentação de Energia)**

Pinos responsáveis por fornecer ou receber energia elétrica.

* VBUS: Alimentação vinda da porta USB (geralmente 5V).
* VSYS: Sistema de alimentação geral da placa. Pode vir da USB ou de outra fonte.
* 3V3: Saída de 3,3 volts estabilizados (alimentação para sensores, por exemplo).
* 3V3\_EN: Pino que ativa/desativa a saída 3V3 — pode ser usado para desligar periféricos.

**Ground (Terra / GND)**

Pinos de referência elétrica. Tudo que funciona com energia precisa estar ligado a um GND (Ground = Terra).

* É o “pólo negativo” que fecha o circuito.

**UART**

Comunicação serial assíncrona, ou seja, sem necessidade de um sinal de controle externo para sincronizar a troca de dados.

* Os dispositivos só precisam concordar na mesma velocidade de comunicação (chamada de baud rate).
* UART (Universal Asynchronous Receiver-Transmitter) = Transmissor-Receptor Assíncrono Universal.
* Pinos:
  + TX (Transmit) = Envia dados
  + RX (Receive) = Recebe dados

Usado para comunicação com computadores, módulos Bluetooth, GPS, etc.

**GPIO, PWM e PIO**

* GPIO (General Purpose Input/Output) = Pinos de uso geral para entrada ou saída digital.
* PWM (Pulse Width Modulation) = Modulação por largura de pulso, usada para simular variação de energia em dispositivos como LEDs e motores (ex.: fazer um LED piscar com intensidade).
* PIO (Programmable I/O) = Entrada/Saída programável — permite criar novos tipos de comunicação controlados pelo usuário, útil para protocolos personalizados.

Esses pinos podem mudar de função dependendo do código que você carregar.

**ADC (Analog to Digital Converter)**

Conversor **Analógico-Digital**

* Permite ler **sinais variáveis (como tensão de 0 a 3,3V)** e convertê-los em valores numéricos que o programa entende.
* Exemplo: leitura de sensores de temperatura, luminosidade ou potenciômetros.
* No Pico, os pinos **ADC0, ADC1 e ADC2** são exclusivos para isso.

**SPI (Serial Peripheral Interface)**

Protocolo de comunicação **serial síncrona**, ou seja, **usa um sinal de relógio (clock) para controlar o momento exato em que os dados são enviados e recebidos**.

* Essa sincronia permite **maior velocidade** e **transmissão confiável** entre dispositivos.
* Pinos:  
    
  + **MOSI (Master Out Slave In)** = Pico envia dados
  + **MISO (Master In Slave Out)** = Pico recebe dados
  + **SCK (Serial Clock)** = Pulso de relógio que sincroniza os dados
  + **CS (Chip Select)** = Escolhe o dispositivo que vai se comunicar

Usado com displays, sensores, cartões SD etc.

**2C (Inter-Integrated Circuit)**

Também é um protocolo de comunicação **serial síncrona**, ou seja, **usa um pulso de relógio para coordenar o envio de dados**, mas com uma vantagem: **apenas 2 fios para se comunicar com vários dispositivos**.

* **SDA (Serial Data Line)** = Fio por onde os dados circulam
* **SCL (Serial Clock Line)** = Fio que envia o sinal de tempo para organizar a comunicação

Muito usado com sensores e módulos simples, como displays OLED ou sensores de temperatura.

**System Control**

Pinos usados para controle do sistema:

* **RUN**: Reinicia o microcontrolador (reset)
* **BOOTSEL**: Usado para colocar a placa em **modo de gravação** (por exemplo, ao segurar o botão e conectar ao USB para carregar firmware).

**Debugging**

Pinos usados para **depuração (debug)** do código, geralmente por programadores avançados.

* Permitem **analisar e controlar o que está acontecendo dentro da placa**, passo a passo.
* Interface padrão: **SWD (Serial Wire Debug)**