

Vulcano
v.0.3 2024
Em atualização

PROBOT

19 de março de 2024



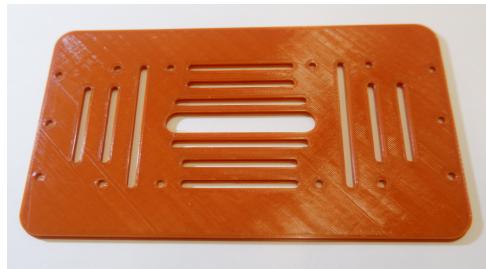
Conteúdo

1 Material	3
1.1 Material para o chassis	3
1.2 Electrónica	5
1.3 Montagem Componentes eletrónicos	7
1.4 Ligações electrónicas	7
1.4.1 Ligações dos motores aos controladores.	7
1.4.2 Ligações dos sensores de chama	7
1.4.3 Ligações dos sensores I2C - Temperatura e cor	8
1.5 Programação	9
1.5.1 Teste de sensores I2C (Inter-Integrated Circuit)	9
1.5.2 Teste dos motores	10
1.5.3 Sensores de chama	10
1.5.4 MQTT	10

1 Material

Para montar o nosso Vulcano vamos precisar do material que a seguir apresentamos. Nota que neste desafio estás a receber o robô Vulcano parcialmente montado.

1.1 Material para o chassis



Duas Bases



Dois suportes de bases



Quatro suportes de motor



Quatro whegs



Suporte sensores chama



Suporte controladores



Parafusos Motores 3x30 (8)



Parafuso suporte controladores 3x16 (6)

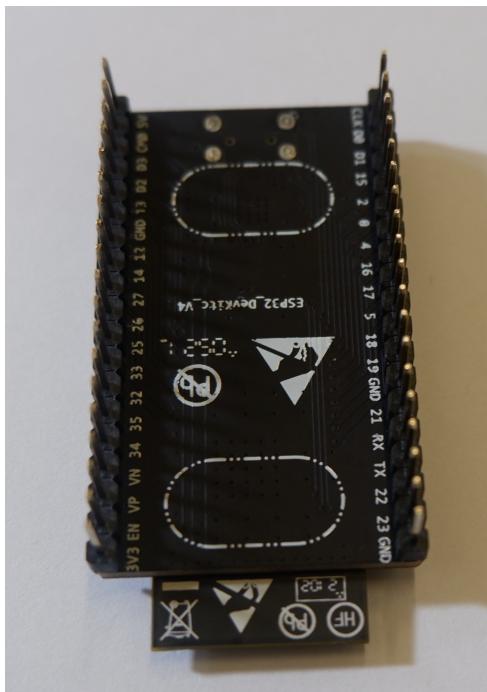


Parafusos chassis 3x10 (30)

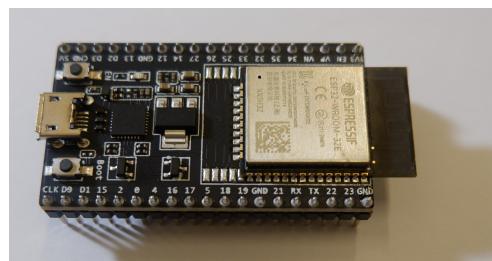


Porcas parafusos chassis (30)

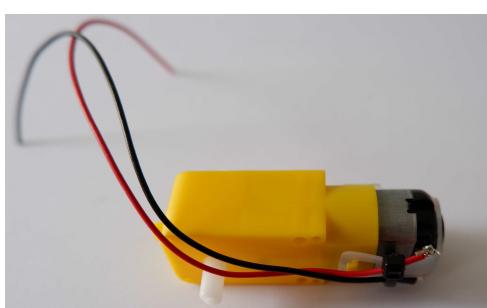
1.2 Electrónica



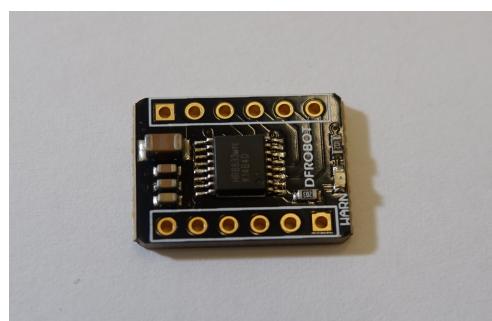
ESP32 - Vista frontal



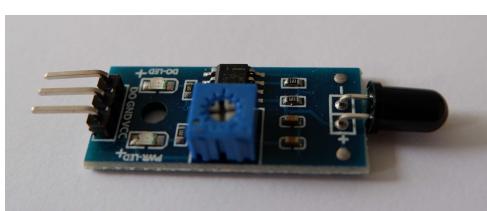
ESP32 - Vista lateral



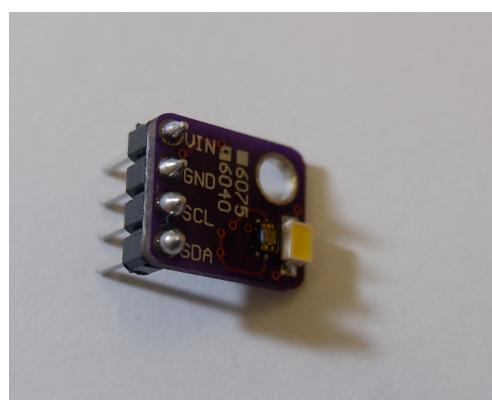
Motores (4)



Controlador de motor (1)



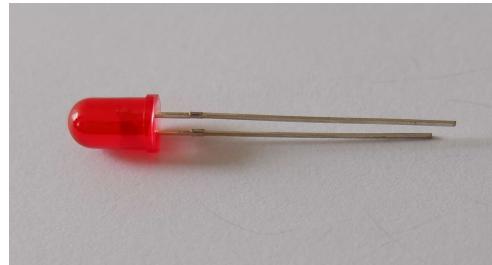
Sensor de chama (3)



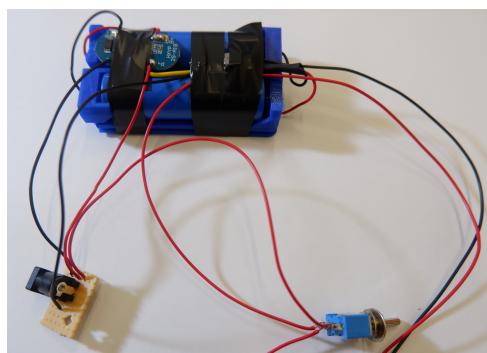
Sensor de cor (1)



Sensor de Temperatura (1)



Led (1)



Bateria (1)



Carregador (1)

1.3 Montagem Componentes eletrónicos

1.4 Ligações electrónicas

1.4.1 Ligações dos motores aos controladores.

Pinos do ESP32:

Motores lado esquerdo: Pinos 17 e 18.

Motores lado direito: Pinos 26 e 27.

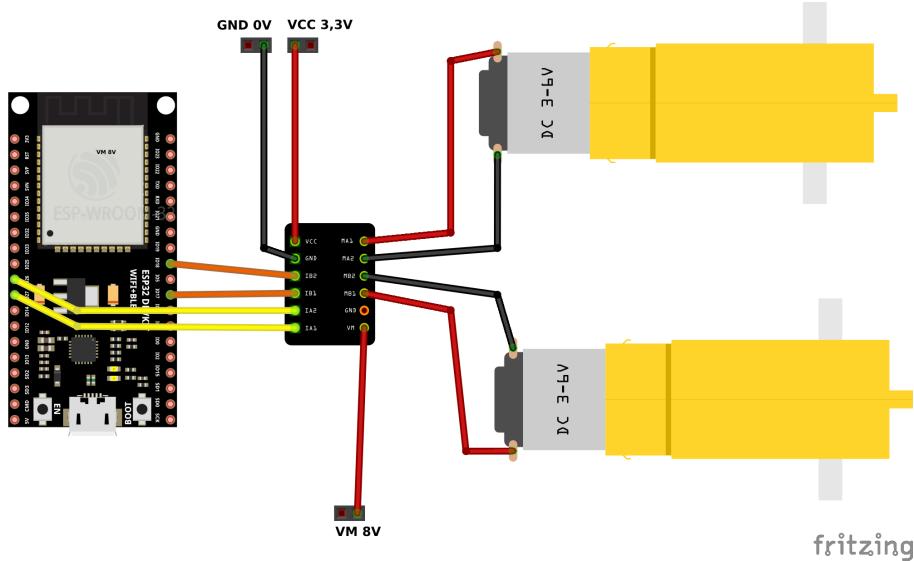


Figura 1: Ligação dos motores

1.4.2 Ligações dos sensores de chama

A figura 2 mostra um esquema das ligações dos sensores de chama ao controlador ESP32. Os sensores têm saída digital e é necessário declarar na função *setup* o modo dos mesmos - ex.

```
pinMode (34, INPUT);
```

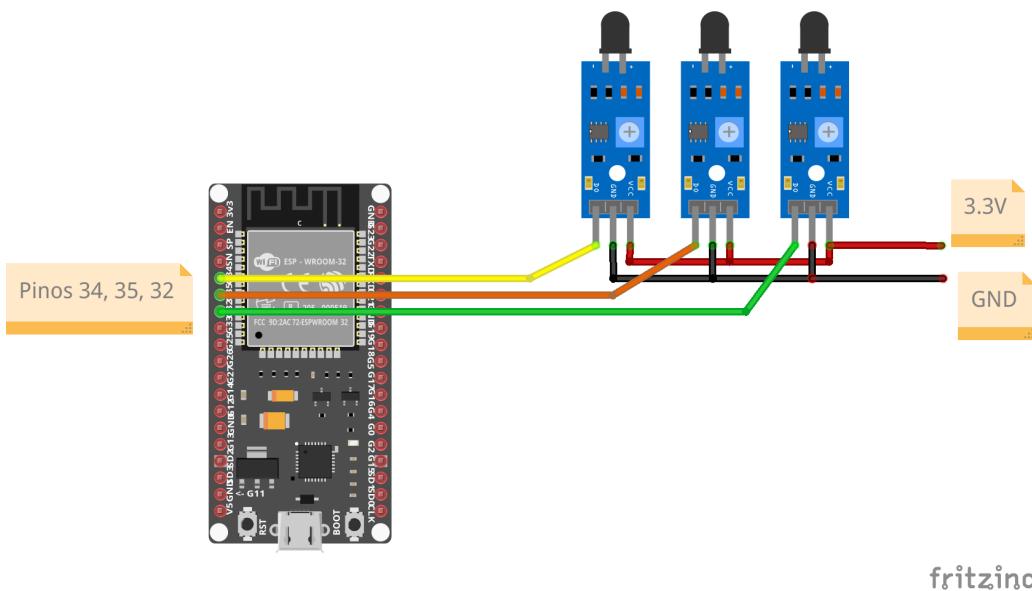


Figura 2: Ligações dos sensores

1.4.3 Ligações dos sensores I2C - Temperatura e cor

A figura 3 mostra o esquema das ligações dos sensores de temperatura e cor ao controlador ESP32. Os pinos do ESP32 devem ser sempre o pino 21, correspondente a SDA (Serial Data) e o pino 22 (Serial Clock).

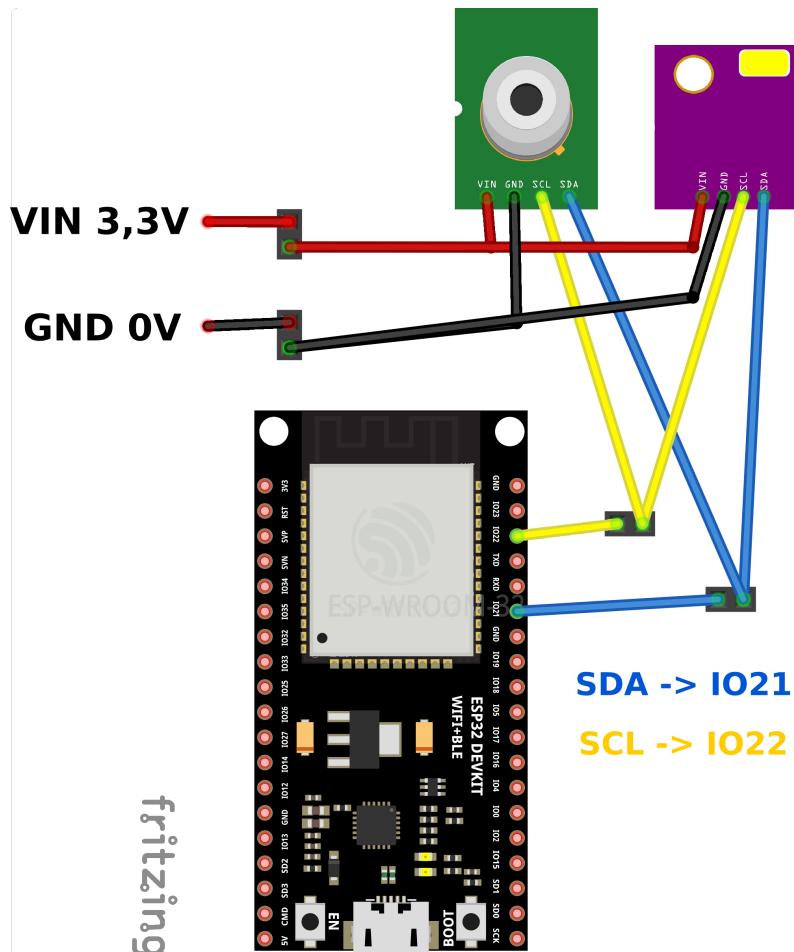


Figura 3: Ligações dos sensores de temperatura e cor

1.5 Programação

Para programares o Vulcano podes usar o Visual Studio Code adicionando a extensão PlatformIO, ou em opção usar o Arduino IDE. Em qualquer dos casos deves começar por criar um novo projeto (VScode+Platformio) ou um novo Sketch (Arduino IDE). As bibliotecas deverão ficar na mesma pasta do ficheiro principal. A placa a escolher será ESP32 DEV MODULE. No caso de usares o Arduino IDE não esqueças de adicionar as placas ESP32 ao IDE. Em Arquivo/Preferências (File/Preferences) adicionar o seguinte link https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json na opção URL do gerenciador de placas adicionais (Additional boards manager URL's)

No repositório Github do Vulcano poderás encontrar pequenos exemplos de código para testares todos os componentes electrónicos do robô. Estes estão incluídos nas seguintes pastas:

- i2c_scan
- motores
- sensor_chama
- sensor_rgb
- sensor_temp
- temperatura_led
- motores_temperatura
- mqtt_temperatura

Em quase todas as pastas existem diversos ficheiros. O ficheiro *main.cpp* é aquele onde desenvolvemos o nosso código. Os outros correspondem a bibliotecas necessárias à programação de sensores e motores e deverão constar dos ficheiros *main.cpp* através da diretiva *include* - no exemplo *motores* necessitamos da biblioteca *ESP32MotorControl*:

```
#include "ESP32MotorControl.h"
```

Se usares o Arduino IDE a extensão do ficheiro principal será *.ino* em vez de *.cpp*.

1.5.1 Teste de sensores I2C (Inter-Integrated Circuit)

I2C (Inter-Integrated Circuit) é um bus de comunicação em série que utiliza uma arquitectura multi-master-slave.

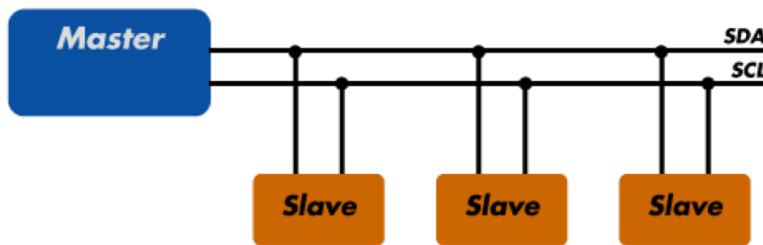


Figura 4: I2C

Devido à sua simplicidade, o I2C é amplamente utilizado na comunicação entre microcontroladores e conjuntos de sensores, ecrãs, dispositivos IoT, EEPROMs, etc.

O I2C é constituído por duas linhas, SCL (Serial Clock) e SDA (Serial Data). O I2C suporta múltiplos slaves, ou seja, podem ser ligados múltiplos dispositivos I2C slave a um controlador I2C. Estes diferentes dispositivos I2C slave têm diferentes endereços de dispositivo, pelo que o controlador principal I2C pode aceder a um determinado dispositivo especificado através do seu endereço. No nosso caso o ESP32 será o controlador e os sensores de temperatura e cor os slaves.

O sensor de temperatura MLX90614 tem o endereço 0x5A (hexadecimal) e o sensor de cor tem o endereço 0x10 (hexadecimal). Para saberes se os sensores do teu robô estão ligados corretamente podes usar o ficheiro da pasta *i2c_scan* e utilizando o monitor serial terás a informação dos sensores ligados e o respectivo endereço.

1.5.2 Teste dos motores

Após a montagem e teste dos sensores deves proceder aos testes dos motores. Na pasta *motores* tens um exemplo que permite ver o funcionamento dos motores. Se tudo estiver corretamente ligado estes funcionarão da seguinte forma:

- Em frente durante 5 segundos (velocidade 50);
- Vira à esquerda durante 5 segundos;
- Vira à direita durante 5 segundos;
- Para durante 2 segundos.

Se na fase inicial em que os motores funcionam **em frente**, os motores de um lado rodam para trás, deves trocar os fios que ligam o ESP32 ao controlador dos motores - se for do lado esquerdo, trocar os pinos 17 e 18, se for do lado direito, trocar os pinos 26 e 27.

1.5.3 Sensores de chama

Os sensores de chama usados no Vulcano têm uma saída digital que envia um sinal 0 quando em presença de radiação infravermelha e 1 quando não deteta essa radiação. Poderás usá-los para orientar o teu Vulcano na subida.

1.5.4 MQTT

MQTT, sigla de Message Queuing Telemetry Transport, é um protocolo de mensagens leve para sensores e pequenos dispositivos móveis otimizado para redes TCP/IP.

Neste protocolo um conjunto de equipamentos - sensores, microcontroladores, computadores, ... - comunicam enviando mensagens (publishers) e recebendo mensagens (subscribers). Qualquer um dos equipamentos pode ser ao mesmo tempo publisher e subscriber. As mensagens são separadas por tópicos.

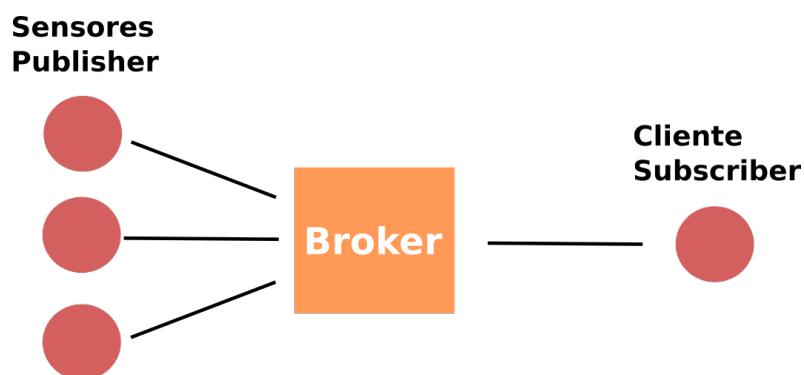


Figura 5: MQTT

Durante os dias do festival terás um *broker* MQTT a funcionar - o IP do computador em que o *broker* funciona será disponibilizado no primeiro dia. Para testares o envio de mensagens para o broker podes usar o seguinte software - <http://mqtt-explorer.com/> e publicar para os tópicos que entenderes. No desafio propriamente dito terás de publicar para os seguintes tópicos:

- **temperatureOBJ** -> o valor da temperatura do solo
- **temperatureAMB** -> o valor da temperatura ambiente
- **hot** -> uma string *HOT* indicando um ponto quente
- **top** -> uma string *TOP* indicando a zona vermelha da rampa

Esta é a fase mais complexa do desafio mas não te preocipes – durante o workshop daremos exemplos e lá estaremos para te apoiar!

Bom trabalho e acima de tudo diverte-te!