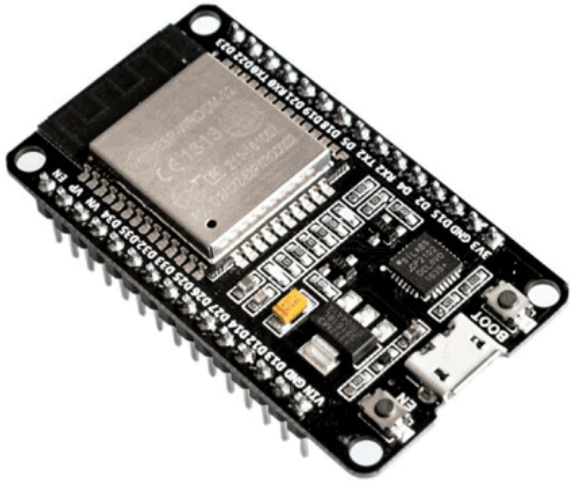


ESPRESSIF

Sensores - HC-SR04, TCRT5000 e KY-033

PROF. ME. HÉLIO ESPERIDIÃO

Sensor Ultrasonico HC-SR04

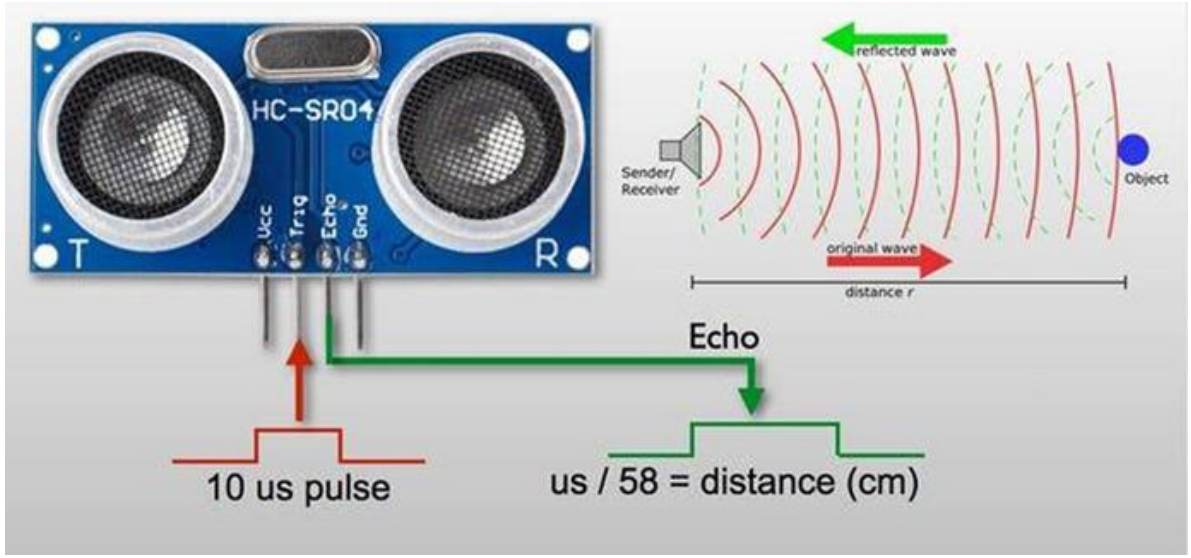
O **Sensor Ultrasonico HC-SR04** é aplicado com mais frequência em projetos de robótica, principalmente em chassis robóticos, robôs ou carrinhos. O sensor é capaz de medir com precisão (3mm de margem de erro) distâncias de 2cm até 4m.

– Especificações e características:

– Tensão de operação: 5VDC
– Corrente de operação: 15mA
– Faixa de detecção (ângulo): ±15º
– Alcance: 2cm a 4m
– Margem de erro: ±3mm

funcionamento

O funcionamento do HC-SR04 se baseia no envio de sinais ultrassônicos pelo sensor, que aguarda o retorno (echo) do sinal, e com base no tempo entre envio e retorno, calcula a distância entre o sensor e o objeto detectado.

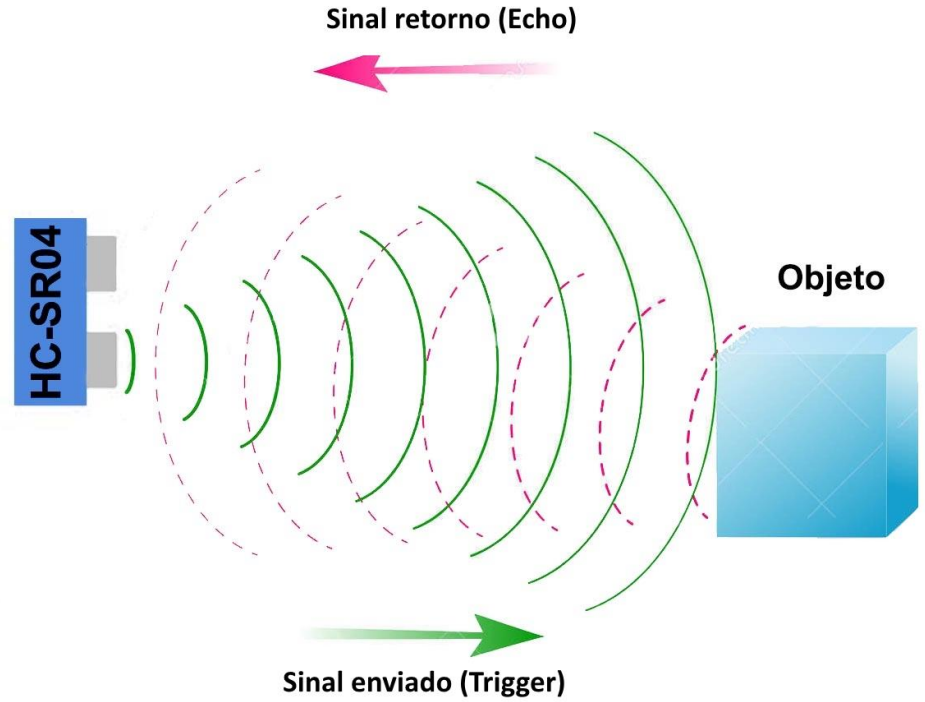


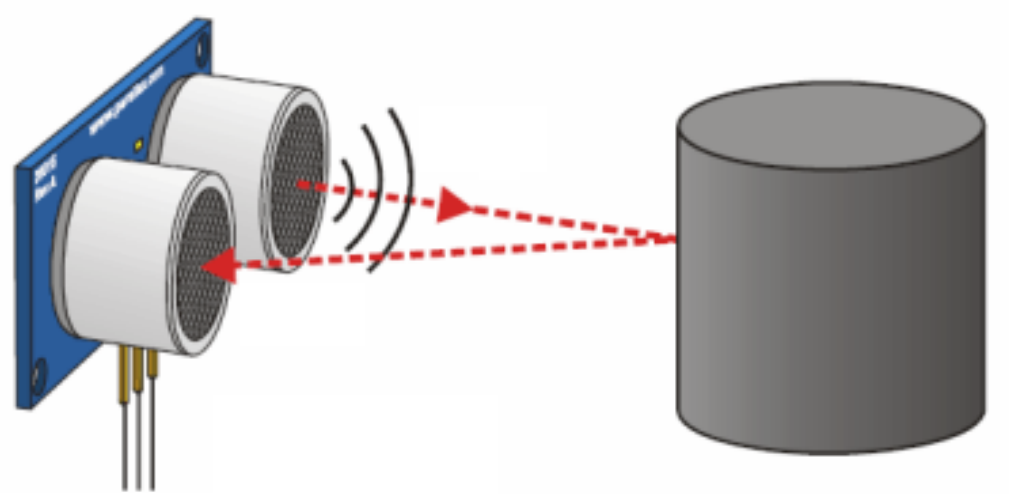
Distância

Primeiramente é enviado um pulso de 10µs, indicando o início da transmissão de dados. Depois disso, são enviado 8 pulsos de 40 KHz e o sensor então aguarda o retorno (em nível alto/high), para determinar a distância entre o sensor e o objeto, utilizando a equação

Distância = (Tempo echo em nível alto * velocidade do som) /2

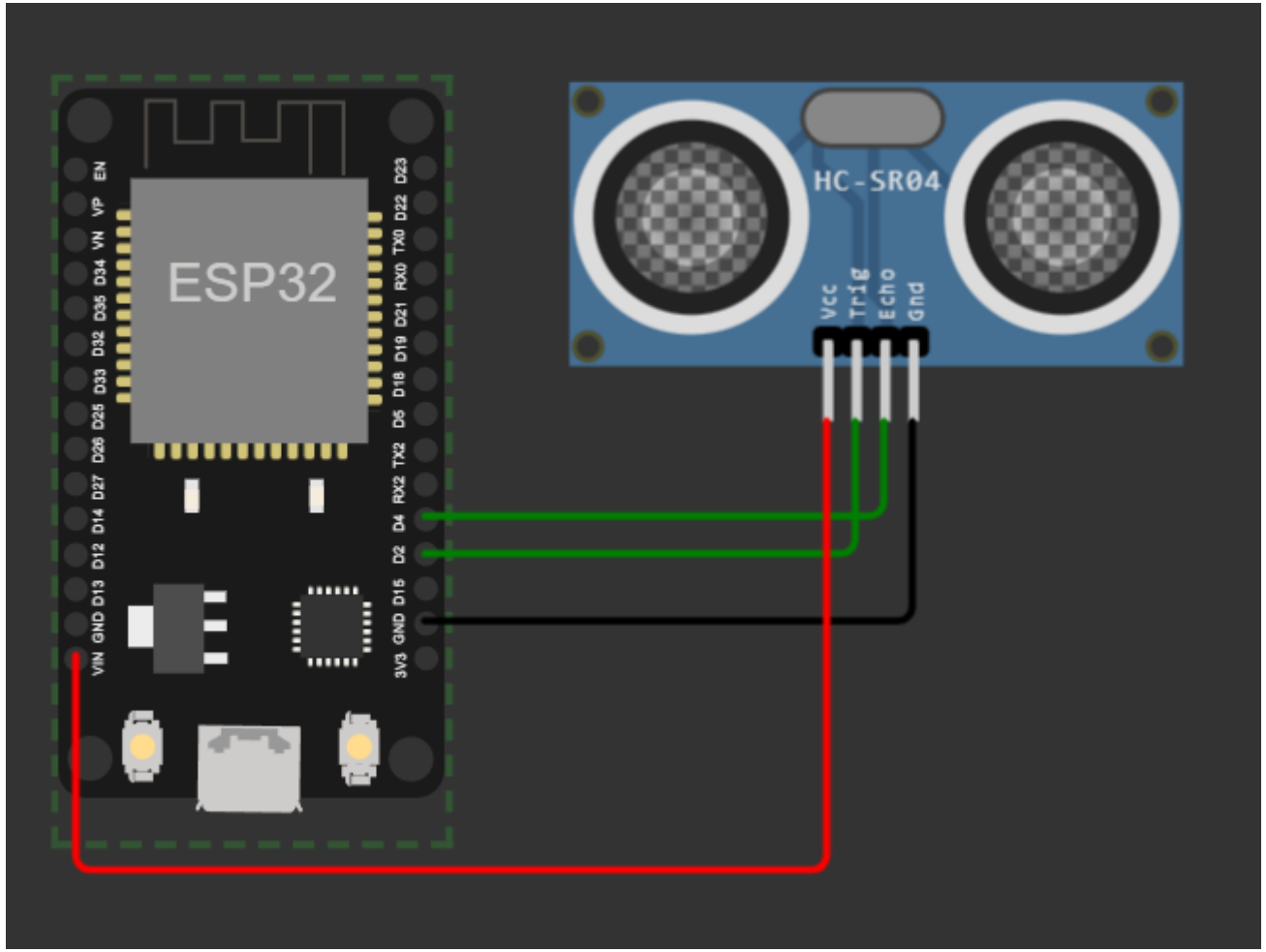
Funcionamento





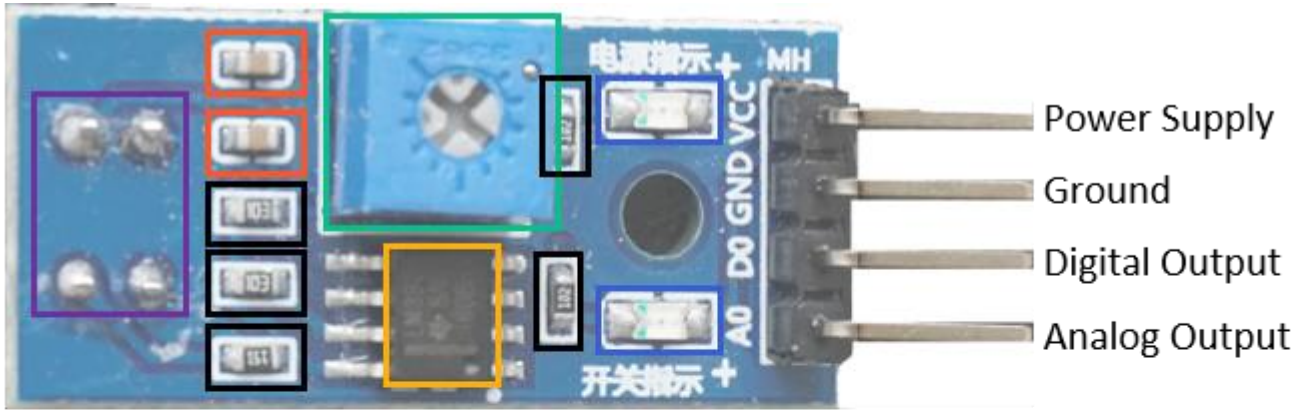
HC-SR04

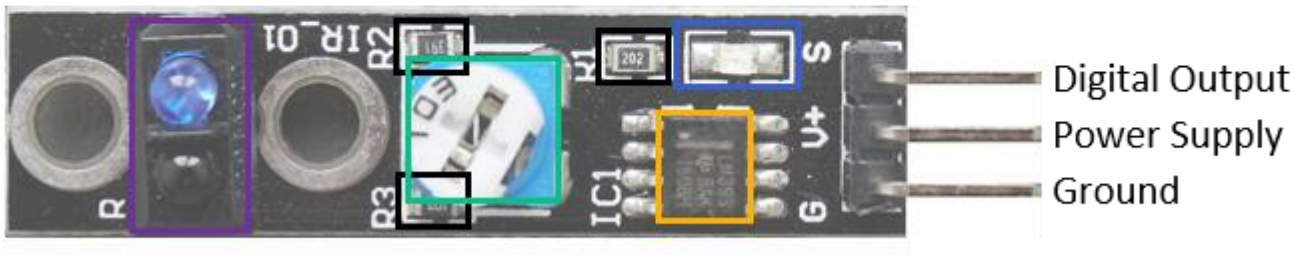
```
#include <Arduino.h>
void setup() {
  #define   TRIG 2
  #define   ECHO 4
  Serial.begin(9600);
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);
  Serial.println("Funcionando");
}

int calcularDistancia(){
  digitalWrite(TRIG, LOW);
  digitalWrite(TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);
  int tempo = pulseIn(ECHO,HIGH);
  int distanciaCm= 0.01723*tempo;
  return distanciaCm;
}
void loop() {
   Serial.printf("%icm\n",distancia());
   delay(1000);
}
```



TCRT5000 vs KY-033 (detecta linha, variação de luminosidade)

KY-033 não possui saída analógica





Potentiometer | LEDs | Resistors | LM393 dual comparator | Capacitors | TCRT5000

Código fonte leitura analógica/Digital

```
#include <Arduino.h>
#define PINO_DIGITAL   4
#define  PINO_ANALOGICO  2
void setup(){
  Serial.begin(9600);
  pinMode(PINO_DIGITAL  ,INPUT);
  pinMode(PINO_ANALOGICO,INPUT);
}
void loop(){
  int dadoAnalogico = analogRead(PINO_ANALOGICO);
  int dadoDigital   = digitalRead(PINO_DIGITAL);
  Serial.printf("Leitura Digital:%i \n",dadoDigital);
  Serial.printf("Leitura Analogica:%i \n",dadoAnalogico);
  delay(1000);
}
```

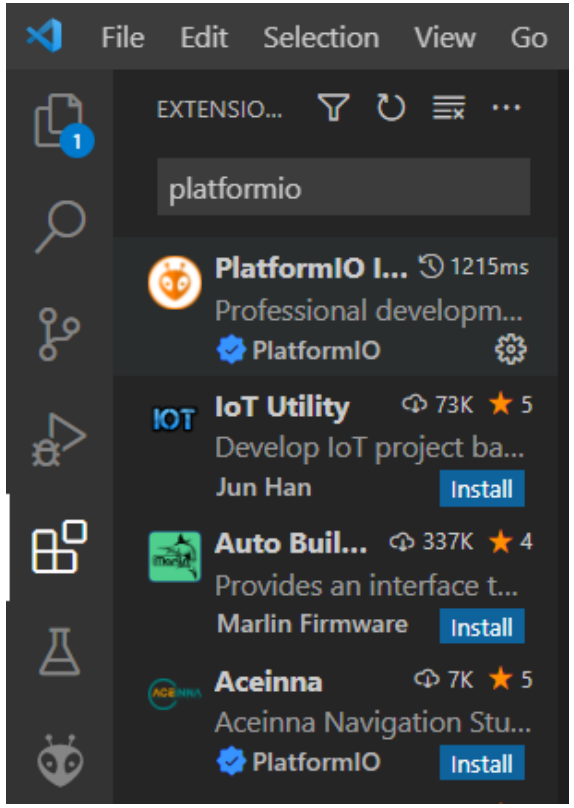
IDE de desenvolvimento

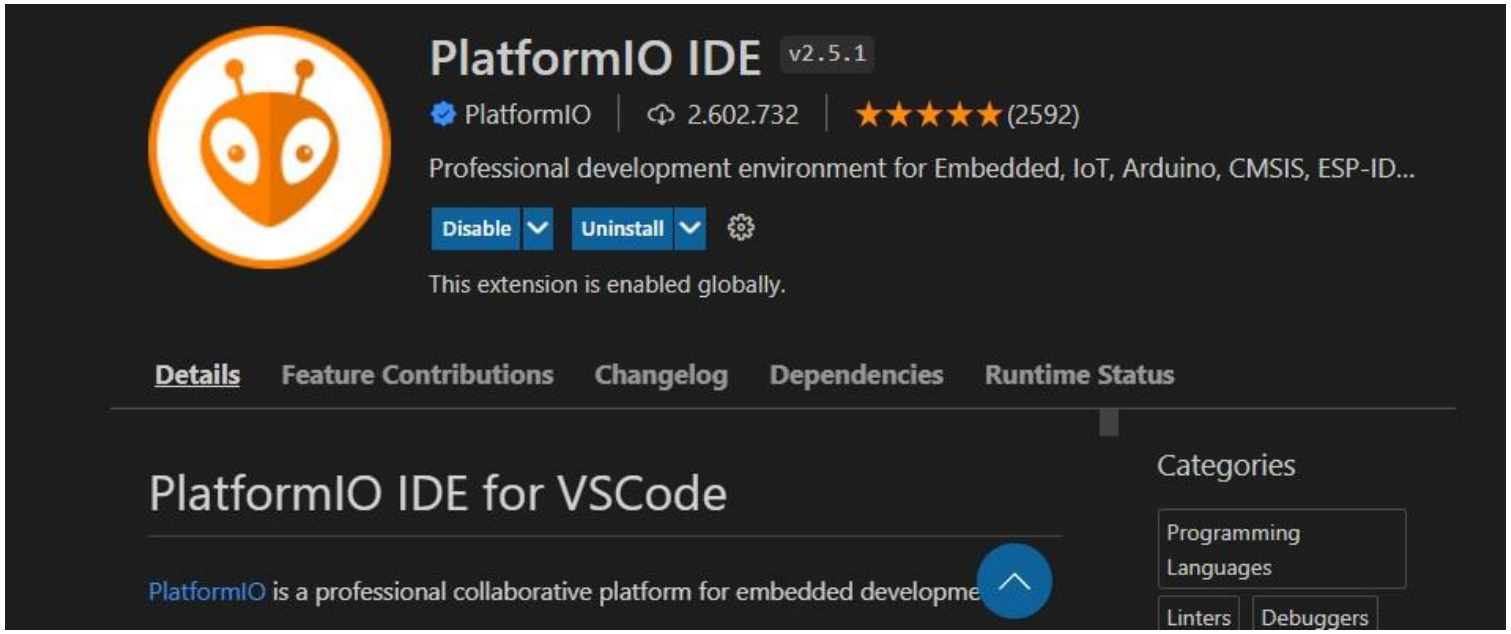
Visual studio code

Extensão: platformio

Instalar extensão

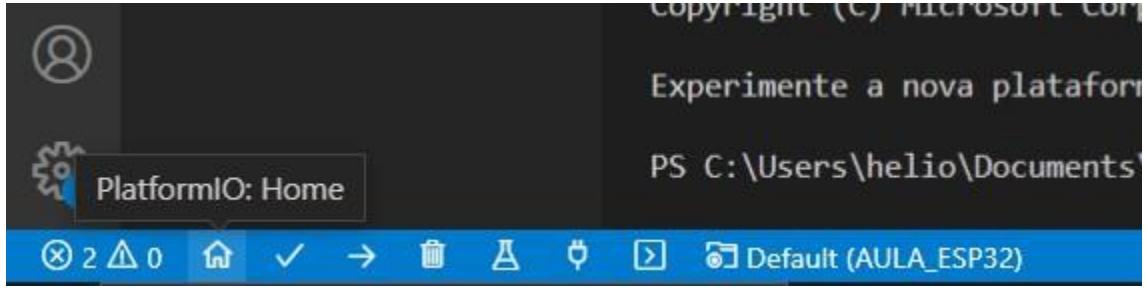
Procure a extensão do PlatformIO IDE



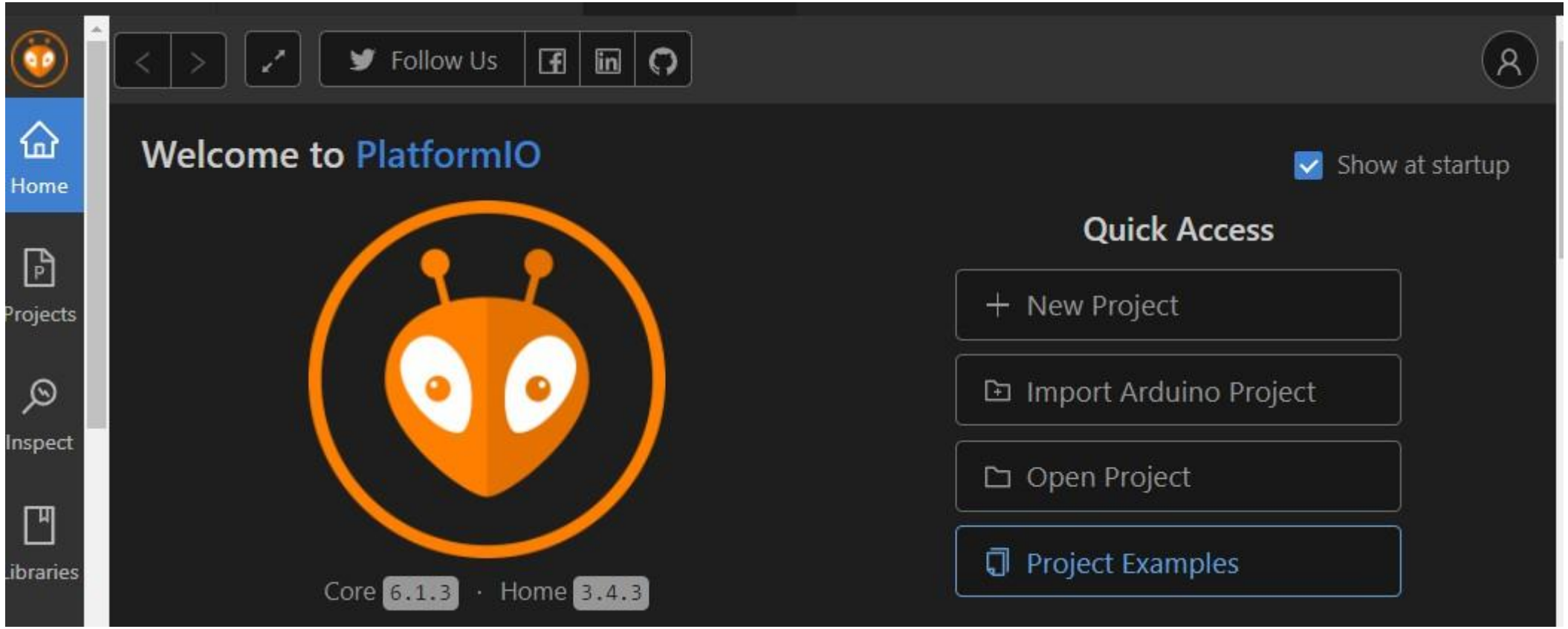


Barra de atalhos do platformio

Click em platformIO: home



Crie um novo projeto

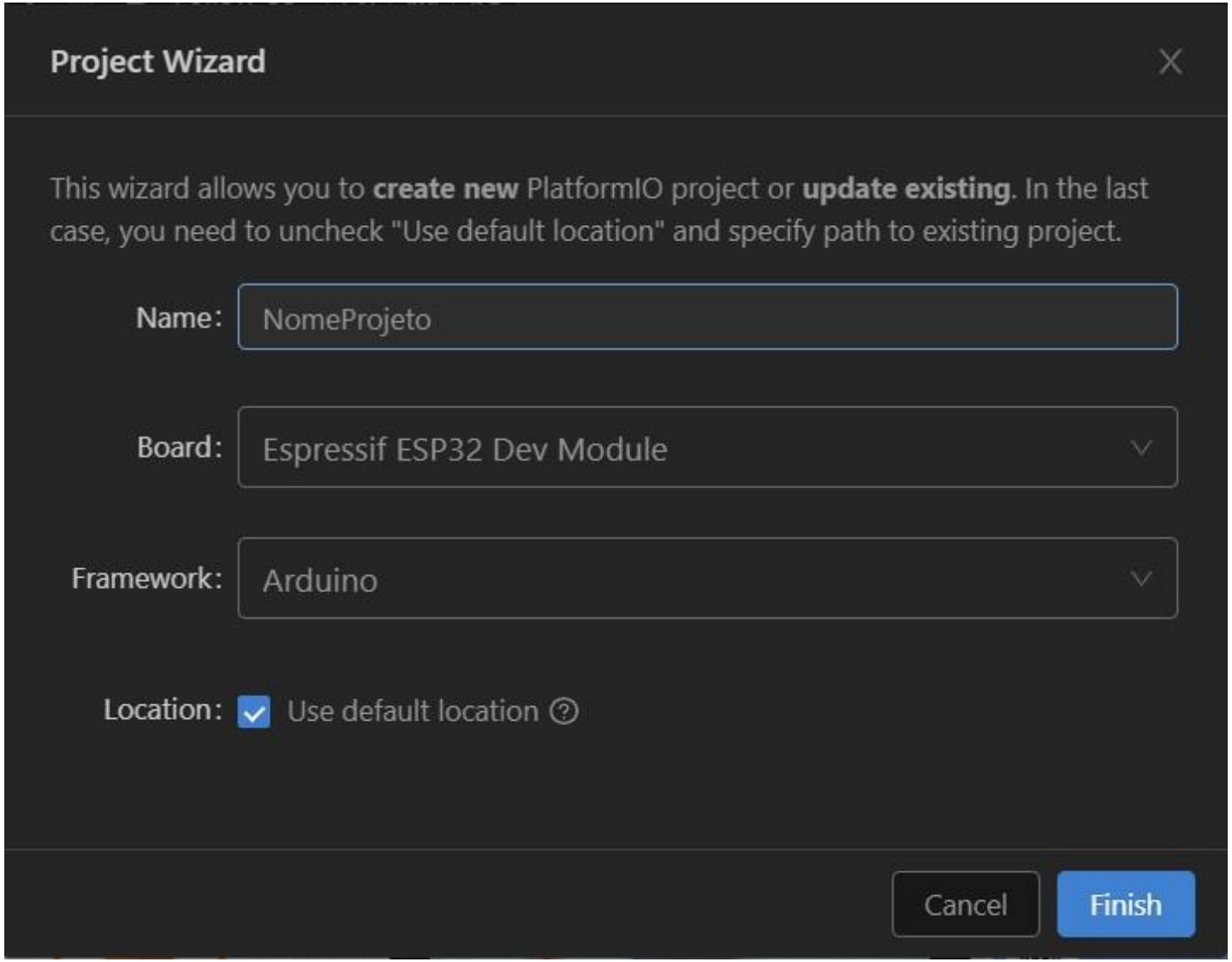


Definição de projetos

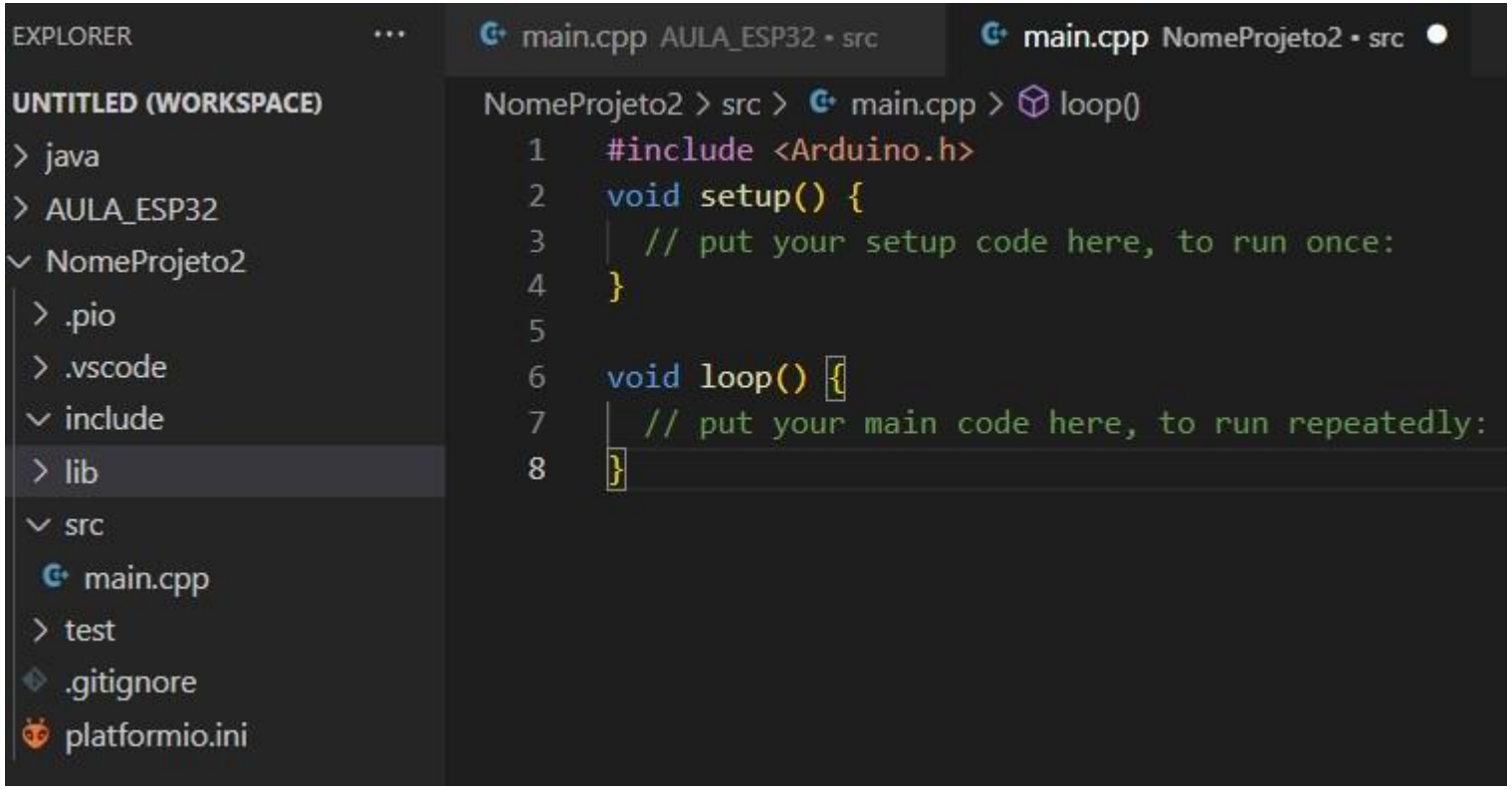
Defina um nome

Escolha a placa Espressif ESP32 Dev Module

Escolha o framework do arduino



main;.cpp


EXPLORER
UNTITLED (WORKSPACE)
java
AULA_ESP32
NomeProjeto2
.pio
.vscode
include
lib
src
main.cpp
test
.gitignore
platformio.ini
main.cpp AULA_ESP32 • src
main.cpp NomeProjeto2 • src
NomeProjeto2 > src > main.cpp > loop()
#include <Arduino.h>
void setup() {
// put your setup code here, to run once:
}
void loop() {
// put your main code here, to run repeatedly:
}

Compilar e gravar o programa

Build: Compila o projeto

Upload: compila e faz upload do programa para o esp32

Obs. Algumas placas necessitam que você pressione o botão de boot para fazer o upload