GRUPO SEMEAR

Aula 4: Integração

Eletrônica/Programação

23/10/2021

1

Datasheets

Definição

Datasheet ou folha de especificações é um documento com todas as **informações técnicas** sobre desempenho e **características** de determinado produto.

Normalmente, esse documento é criado pelo próprio **fabricante**, para permitir que os usuários entendam o **funcionamento e aplicabilidade** do produto.

SEMEAR

Onde encontrar

Fabricante

Lojas especializadas

Sites específicos

SEMEAR

Partes principais

- Introdução
- Características e propriedades
- Detalhamento elétrico
- Detalhamento mecânico
- Anexos e informações adicionais

Sensores

Em alguns casos, o datasheet fornecido será do componente e não do módulo completo

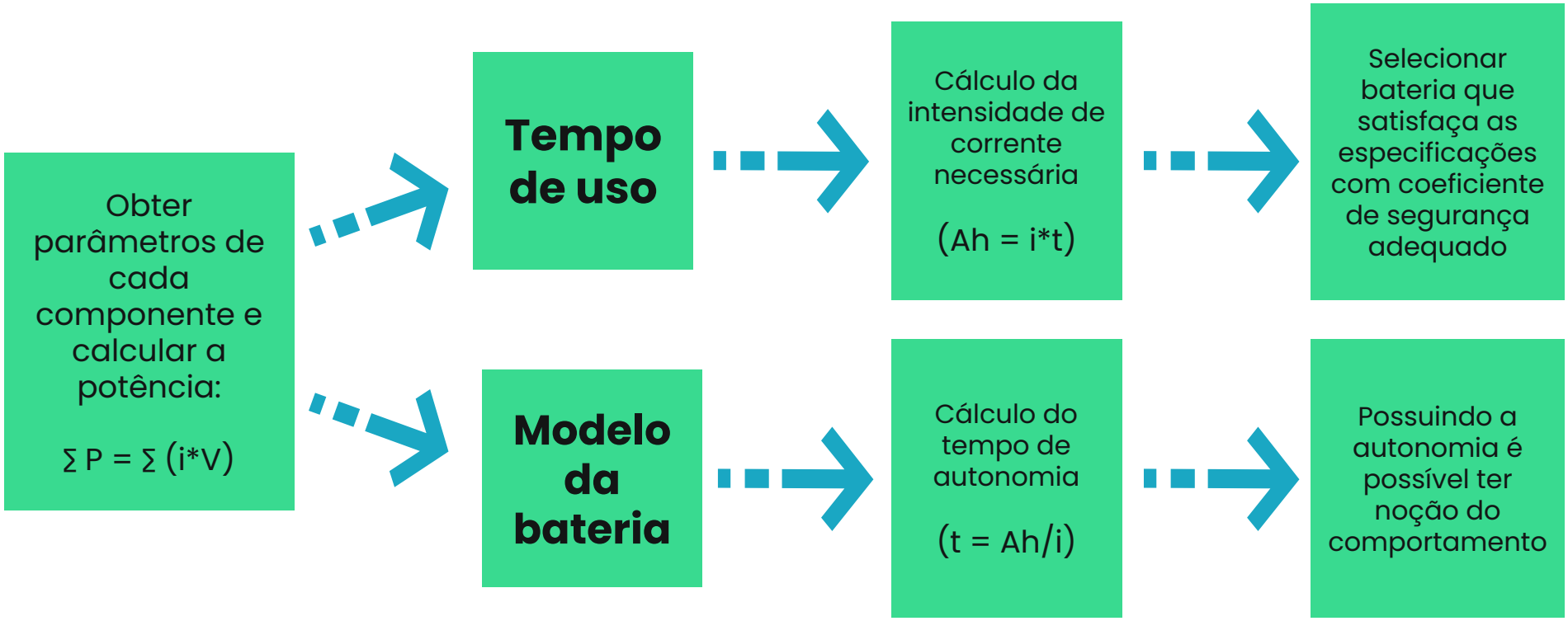
SEMEAR

Alguns exemplos

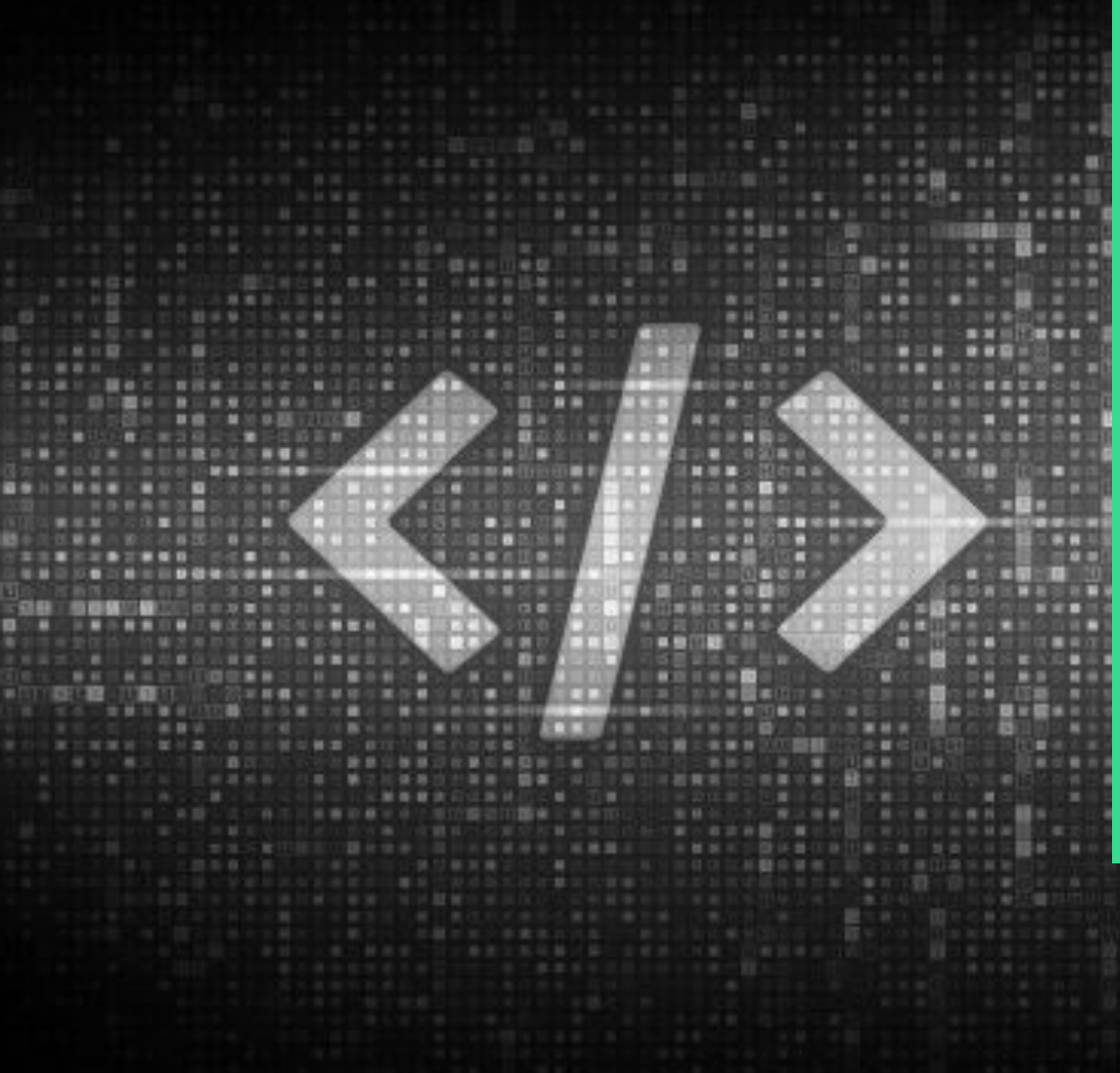
2

Dimensionamento de baterias

Dimensionamento



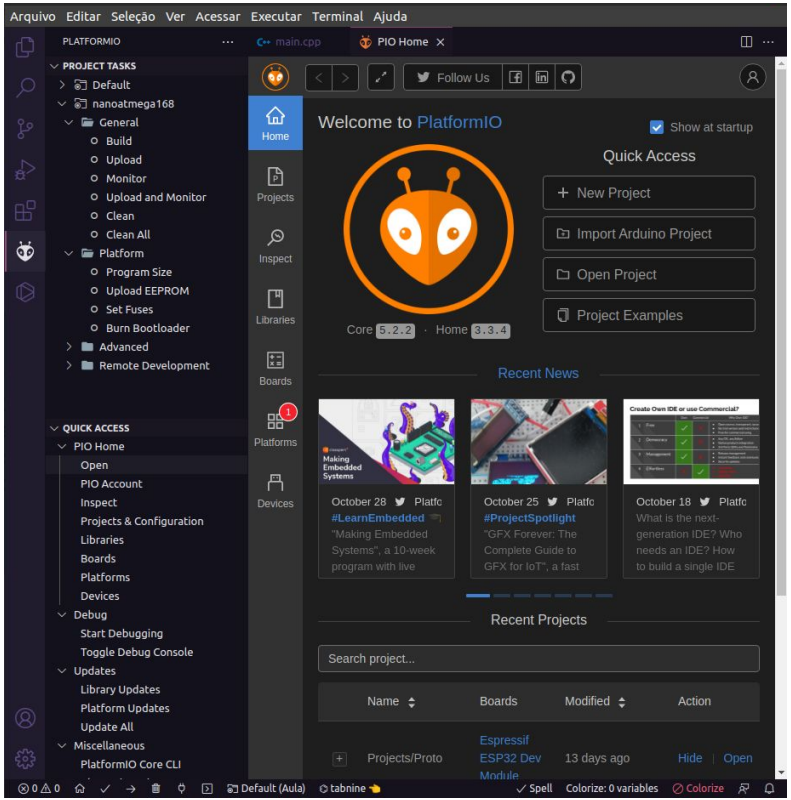
SEMEAR



3

PlatformIO

Definição



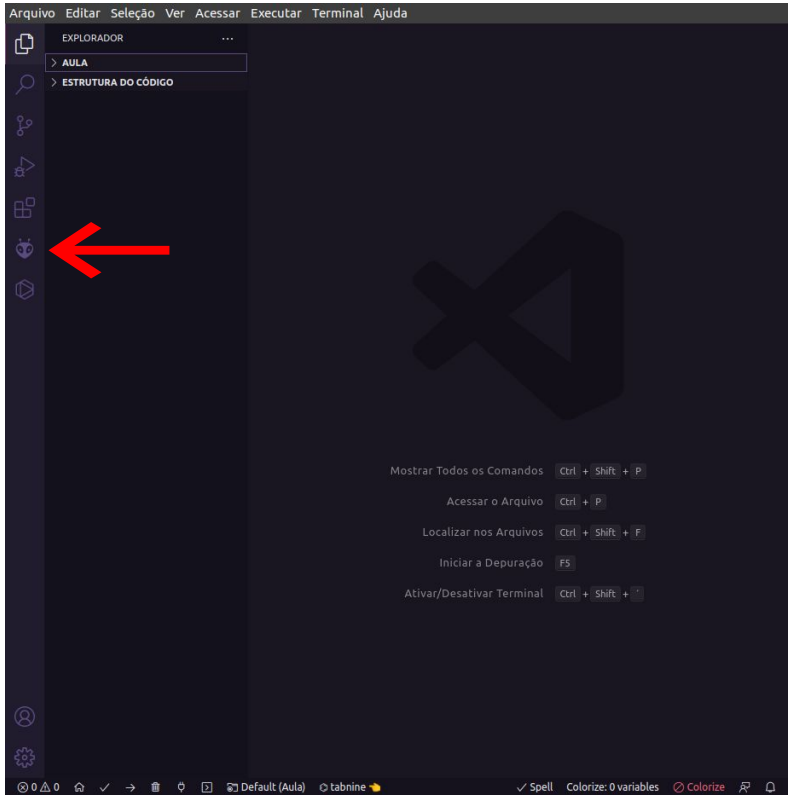
Extensão do Visual Studio Code, que permite carregar algoritmos em microcontroladores

Vantagens:

- Integração com editor de texto
- Interface simples
- Muitas funcionalidades

SEMEAR

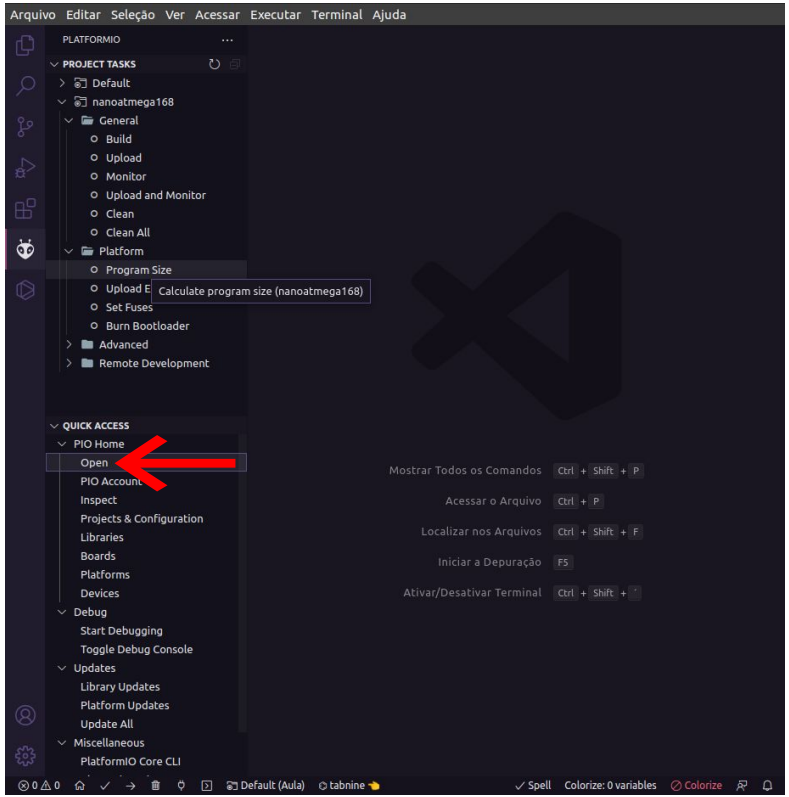
Como utilizar



Ao instalar a extensão ela pode ser acessada na barra lateral do VSCode

SEMEAR

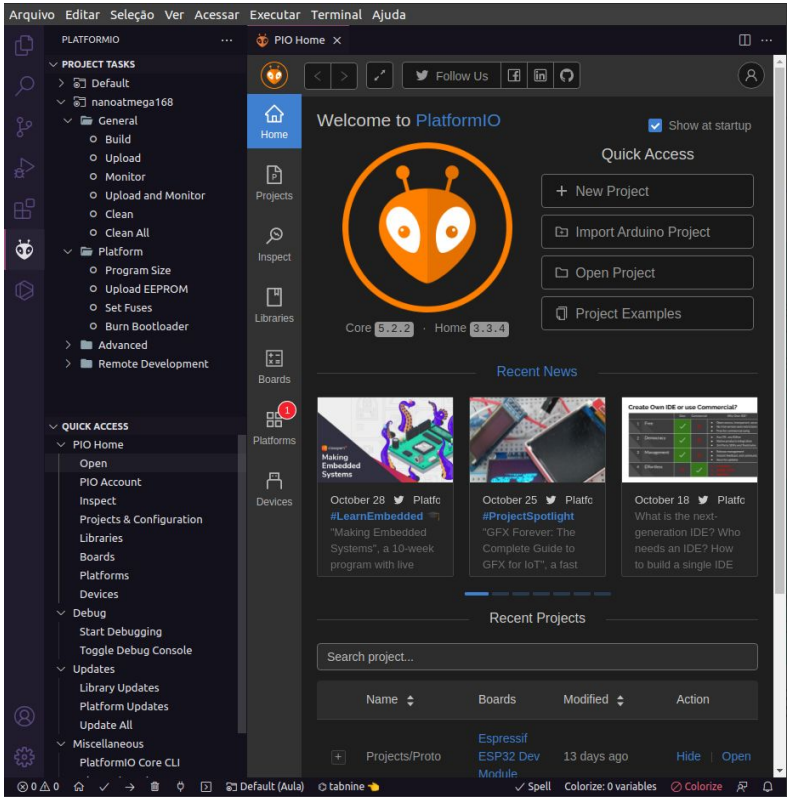
Como utilizar



Para abrir a aba principal é necessário clicar em “Open”

SEMEAR

Como utilizar

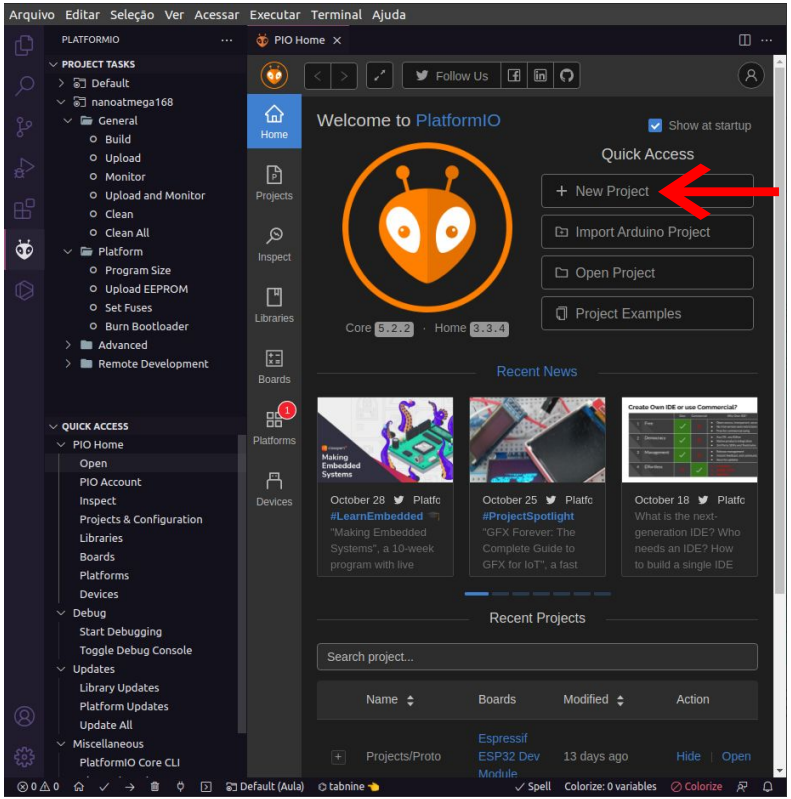


Divisões PIO Home:

- Home: Gerenciamento de projetos
- Project: Acesso de projetos existentes
- Inspect: Análise de projetos
- Libraries: Acesso as bibliotecas
- Boards: Microcontroladores compatíveis

SEMEAR

Como utilizar

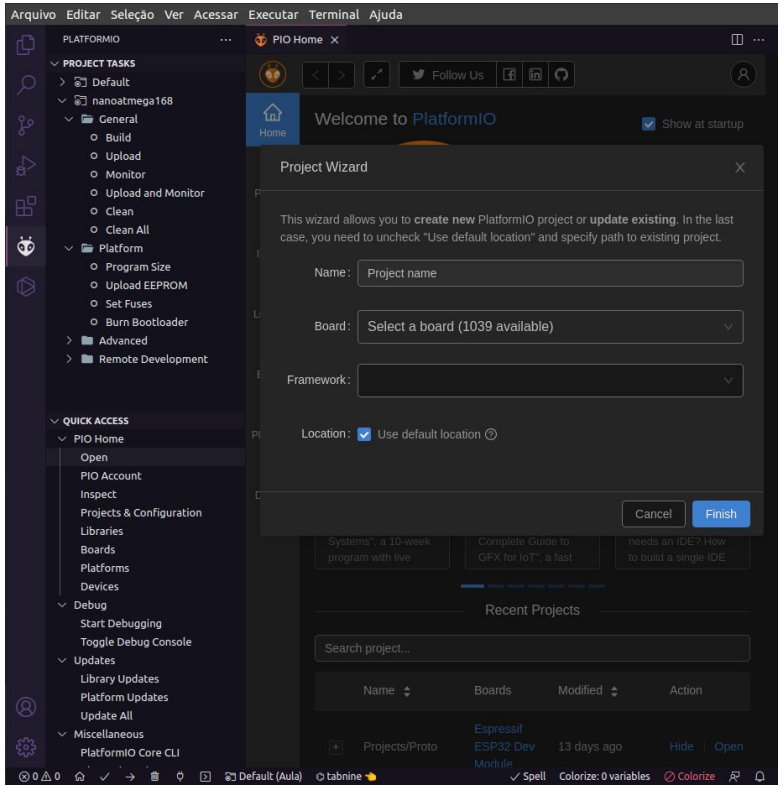


Na aba Home podemos:

- Criar novos projetos
- Importar projetos do Arduino IDE (outra plataforma)
- Abrir projetos salvos
- Abrir exemplos de projetos

SEMEAR

Como utilizar

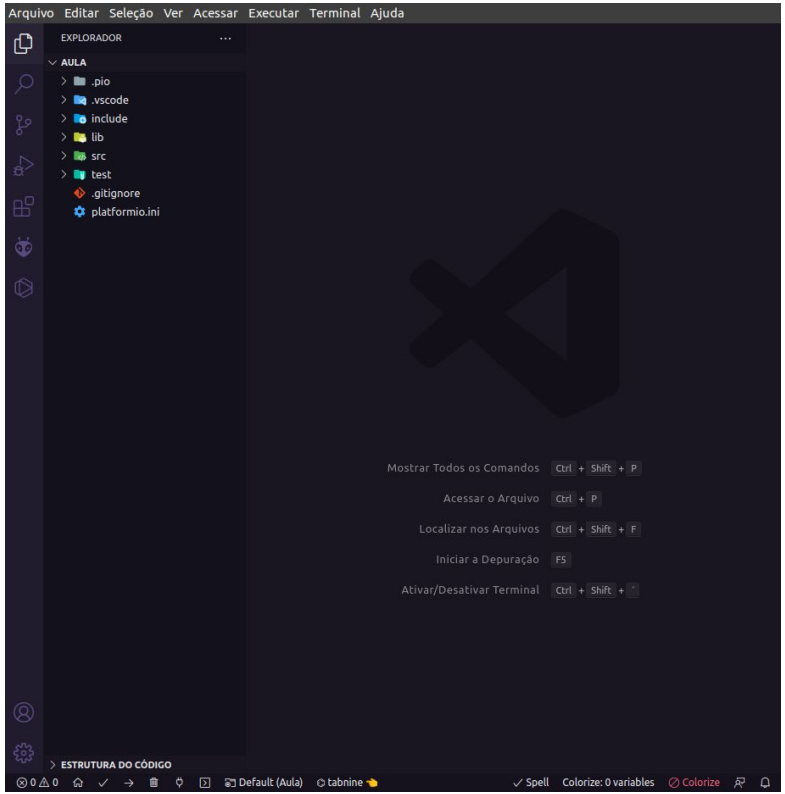


Para criar um novo projeto é necessário:

- Escolher um nome
- Selecionar o microcontrolador
- Selecionar o framework (Arduino)
- Escolher a localização do projeto

SEMEAR

Como utilizar



Árvore de arquivos:

- src: arquivos .c/.cpp (main)
- include: arquivos headers .h
- lib: bibliotecas privadas
- test: arquivos para unit test
- platformio.ini: configuração do projeto (micro, framework, dependências)

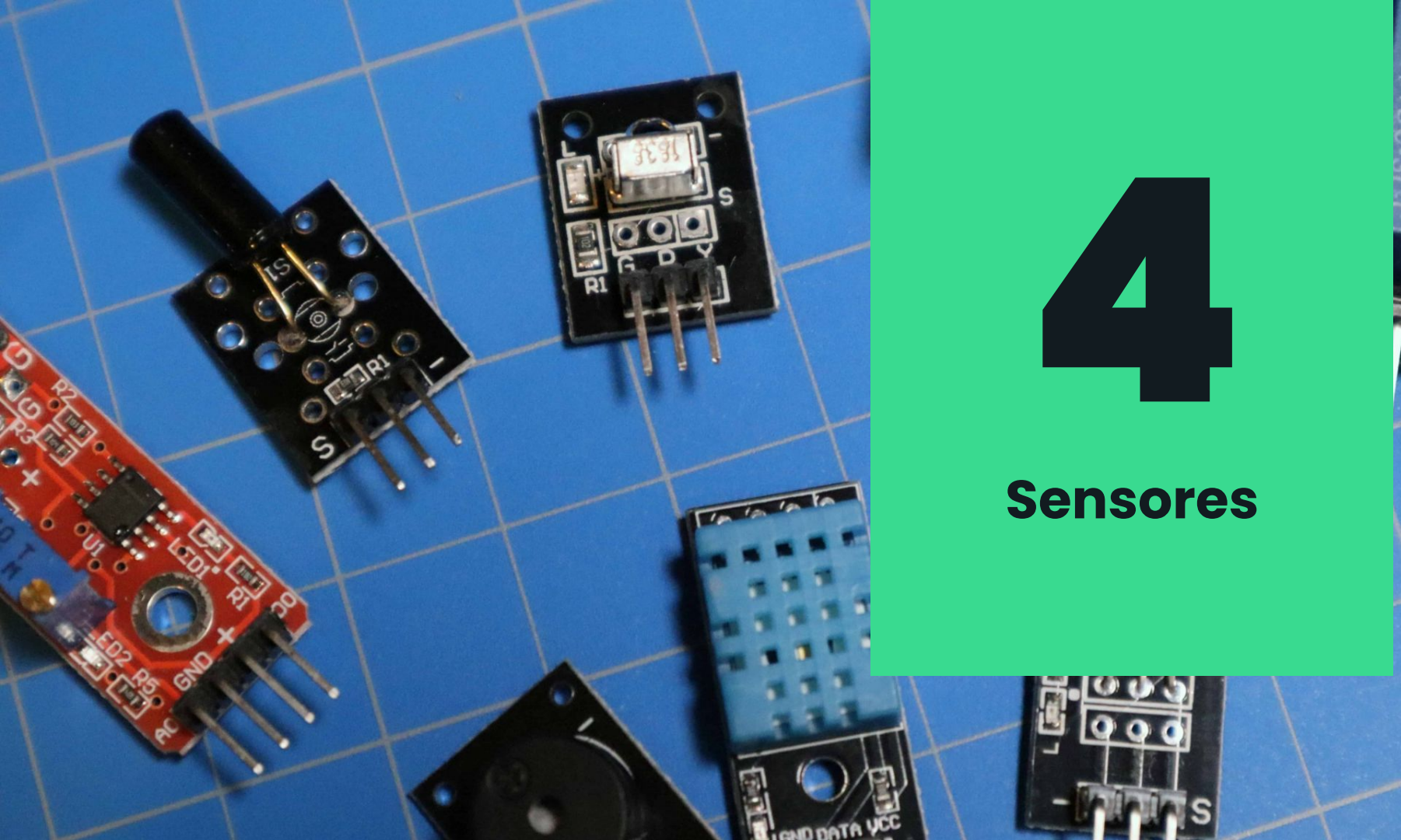
SEMEAR

Como utilizar

- Erros de compilação
- PIO Home
- Build/Compilação
- Upload/Carregar
- Limpar execução
- Monitoração Serial
- Novo terminal
- Troca de projetos

A plataforma ainda tem atalhos localizadas na barra inferior do editor de texto

SEMEAR



4

Sensores

HC SR04

Sensor ultrassônico

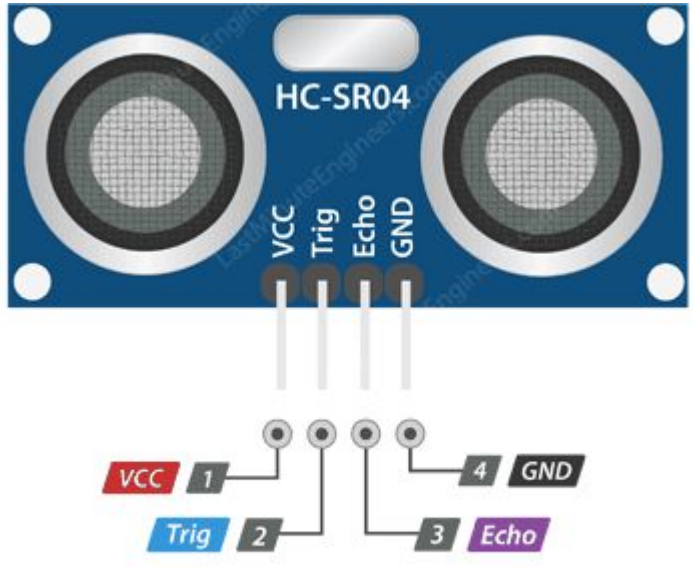
- Alimentação: 5V CC
- Corrente de operação: 2mA
- Ângulo de efeito: 15°
- Alcance: 2cm ~ 4m
- Precisão: 3mm

SEMEAR

HC SR04

Pinagem

- VCC: Tensão contínua
- GND: Aterramento
- Trigger: Emite pulsos de onda
- Echo: Emite pulsos elétricos ao receber sinais de onda

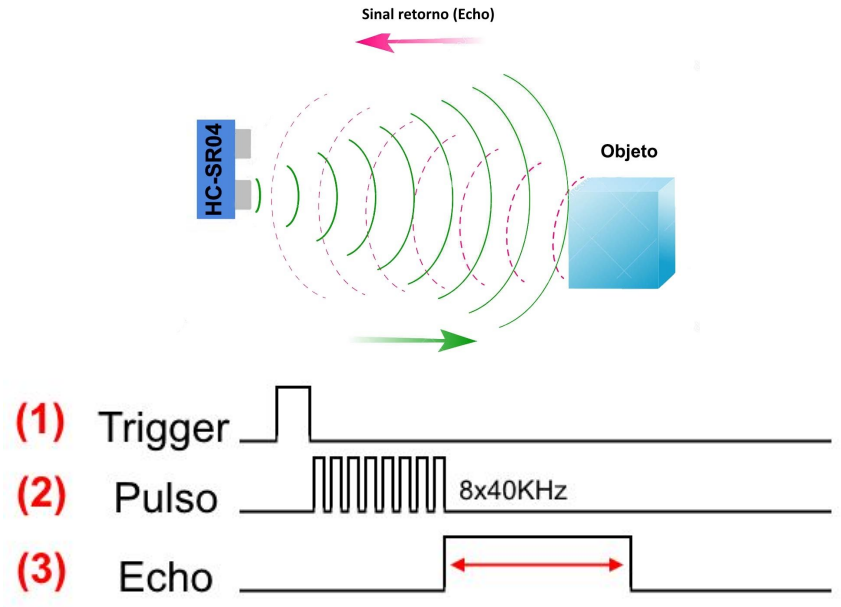


SEMEAR

Como funciona

Há o envio de pulsos de ondas ultrassônicas que irão retornar ao colidir com um objeto.

Ao retornar a onda irá ser percebida pelo echo que conseguirá obter a distância com base no tempo do pulso de onda.



SEMEAR

Algoritmo

```arduino
#include <Arduino.h>

#define TRIGPIN 1
#define ECHOPIN 2

void setup () {
    Serial.begin(9600);

    pinMode(TRIGPIN, OUTPUT);
    pinMode(ECHOPIN, INPUT);
}

void loop () {
    int distance;
    long duration;

    //Clear trigPin
    digitalWrite(TRIGPIN, LOW);
    delayMicroseconds(2);
    //Turn trigPin HIGH for 10 microseconds
    digitalWrite(TRIGPIN, HIGH);
    delayMicroseconds(10);
    //Turn trigPin LOW
    digitalWrite(TRIGPIN, LOW);

    //Input HIGH echoPin
    duration = pulseIn(ECHOPIN, HIGH);

    //Convert duration to distance
    distance = duration*0.034/2;

    //Serial prints
    Serial.print("Distance: ");
    Serial.println(distance);
}
```

Definição dos pinos

Utilização do sensor

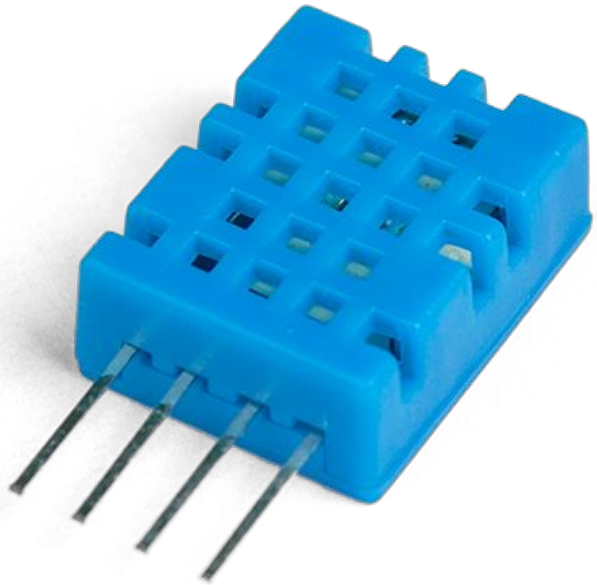
Formatação de dado

SEMEAR

DHT11

Sensor de temperatura e umidade

- Alimentação: 3~5V CC
- Corrente de operação: 0.5 ~ 2.5mA
- Faixa de temperatura: 0 ~ 50°
- Precisão temperatura: 2°
- Faixa de umidade: 20 ~ 90% UR
- Precisão umidade: 5% UR

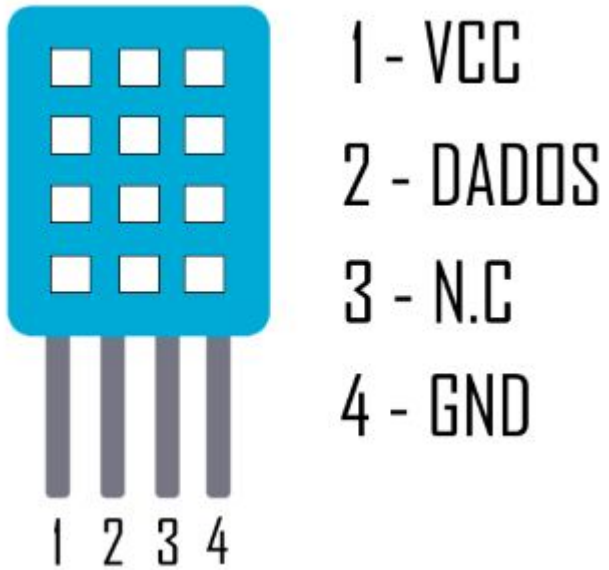


SEMEAR

DHT11

Pinagem

- VCC: Tensão contínua
- Dados: Passagem de dados
- N.C.: Nenhuma pinagem
- GND: Aterramento



SEMEAR

Algoritmo

```ino
#include <Arduino.h>

#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

#define DHTPIN 1
#define DHTTYPE DHT11
#define TIME 1000

DHT_Unified dht(DHTPIN, DHTTYPE);

void setup () {
    Serial.begin(9600);
    dht.begin();
}

void loop () {
    delay(TIME);
    sensors_event_t event;

    dht.temperature().getEvent(&event);
    Serial.print(F("Temperature: "));
    Serial.print(event.temperature);
    Serial.println(F("°C"));

    dht.humidity().getEvent(&event);
    Serial.print(F("Humidity: "));
    Serial.print(event.relative_humidity);
    Serial.println(F("%"));
}
```

Definição dos pinos

Utilização do sensor

Formatação de dado

SEMEAR

Algoritmo

```ino
#include <Arduino.h>

#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

#define DHTPIN 1
#define DHTTYPE DHT11
#define TIME 1000

DHT_Unified dht(DHTPIN, DHTTYPE);

void setup () {
    Serial.begin(9600);
    dht.begin();
}

void loop () {
    delay(TIME);
    sensors_event_t event;

    dht.temperature().getEvent(&event);
    Serial.print(F("Temperature: "));
    Serial.print(event.temperature);
    Serial.println(F("°C"));

    dht.humidity().getEvent(&event);
    Serial.print(F("Humidity: "));
    Serial.print(event.relative_humidity);
    Serial.println(F("%"));
}
```

```ini
[env:nanoatmega168]
platform = atmelavr
board = nanoatmega168
framework = arduino

lib_deps =
    adafruit/DHT sensor library @ 1.4.3
```

SEMEAR

5

Motores

PWM

Problemática: conseguir controlar a velocidade de rotação de motores elétricos rotativos através de algoritmos

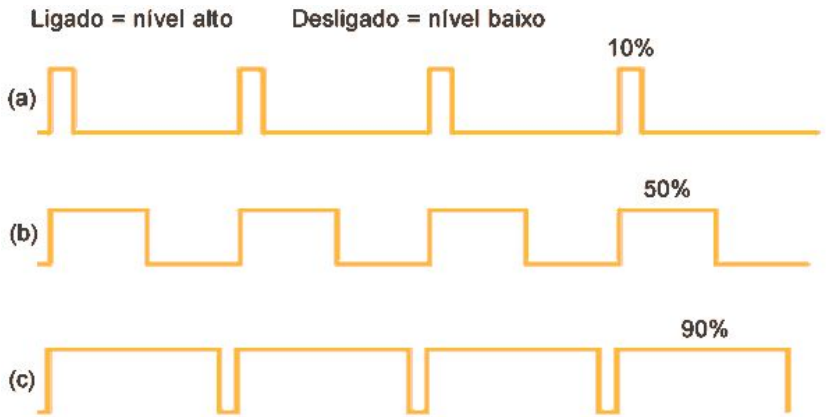
Solução: utilização de PWM, através da junção de sinais digitais e “análogicos” para o controle dos motores

SEMEAR

PWM

A modulação por largura de pulso (PWM), permite um sinal digital composto por apenas 0 e 1 enviar um sinal com diferentes valores inteiros

Para isso, ele emite pequenos ciclos, nos quais a porcentagem de nível alto emitido corresponde ao valor inteiro enviado



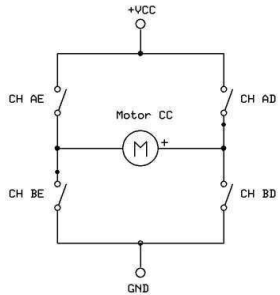
SEMEAR

Algoritmo/Implementação

```ino
#include <Arduino.h>

#define PWM 1
#define MOTOR_L 2
#define MOTOR_R 3

#define TIME 1000
#define SPEED 200 // 0 - 255

void setup () {
    Serial.begin(9600);

    pinMode(PWM, OUTPUT);
    pinMode(MOTOR_L, OUTPUT);
    pinMode(MOTOR_R, OUTPUT);
}

void loop () {
    /* Type Moves*/
    //Stopped
    digitalWrite(MOTOR_L, LOW);
    digitalWrite(MOTOR_R, LOW);

    //Forward
    analogWrite(PWM, SPEED);
    digitalWrite(MOTOR_L, HIGH);
    digitalWrite(MOTOR_R, HIGH);

    //Back
    analogWrite(PWM, -SPEED);
    digitalWrite(MOTOR_L, HIGH);
    digitalWrite(MOTOR_R, HIGH);

    //Curves (left)
    analogWrite(PWM, SPEED);
    digitalWrite(MOTOR_L, LOW);     // HIGH to turn right
    digitalWrite(MOTOR_R, HIGH);    // LOW to turn right
}
```

Ponte H

Circuito eletrônico que tem seu funcionamento se dá pela utilização do PWM



SEMEAR

Obrigado!

Saiba mais em:

SEMEAR - EESC/USP

@semear.usp

Grupo SEMEAR - EESC USP

Equipe Atena EESC-USP

www.semear.eesc.usp.br