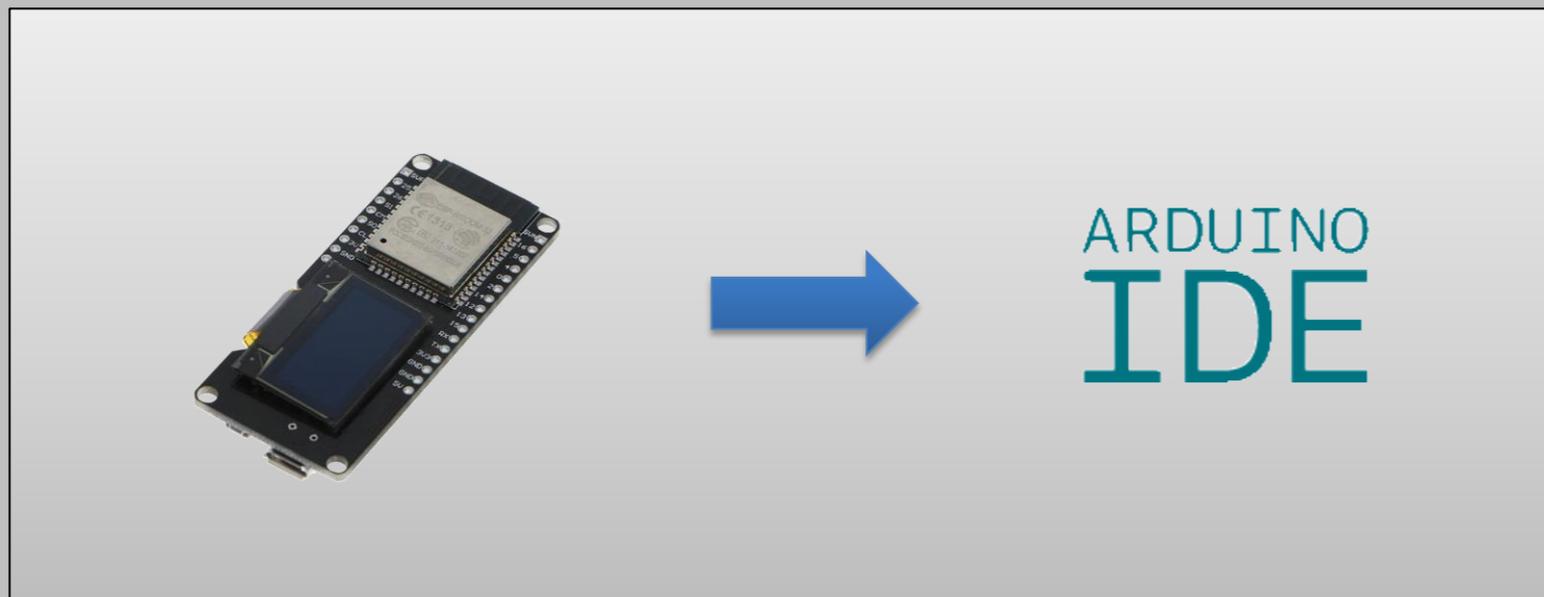


# ESP32 OLED - PROGRESS BAR



Por Fernando Koyanagi

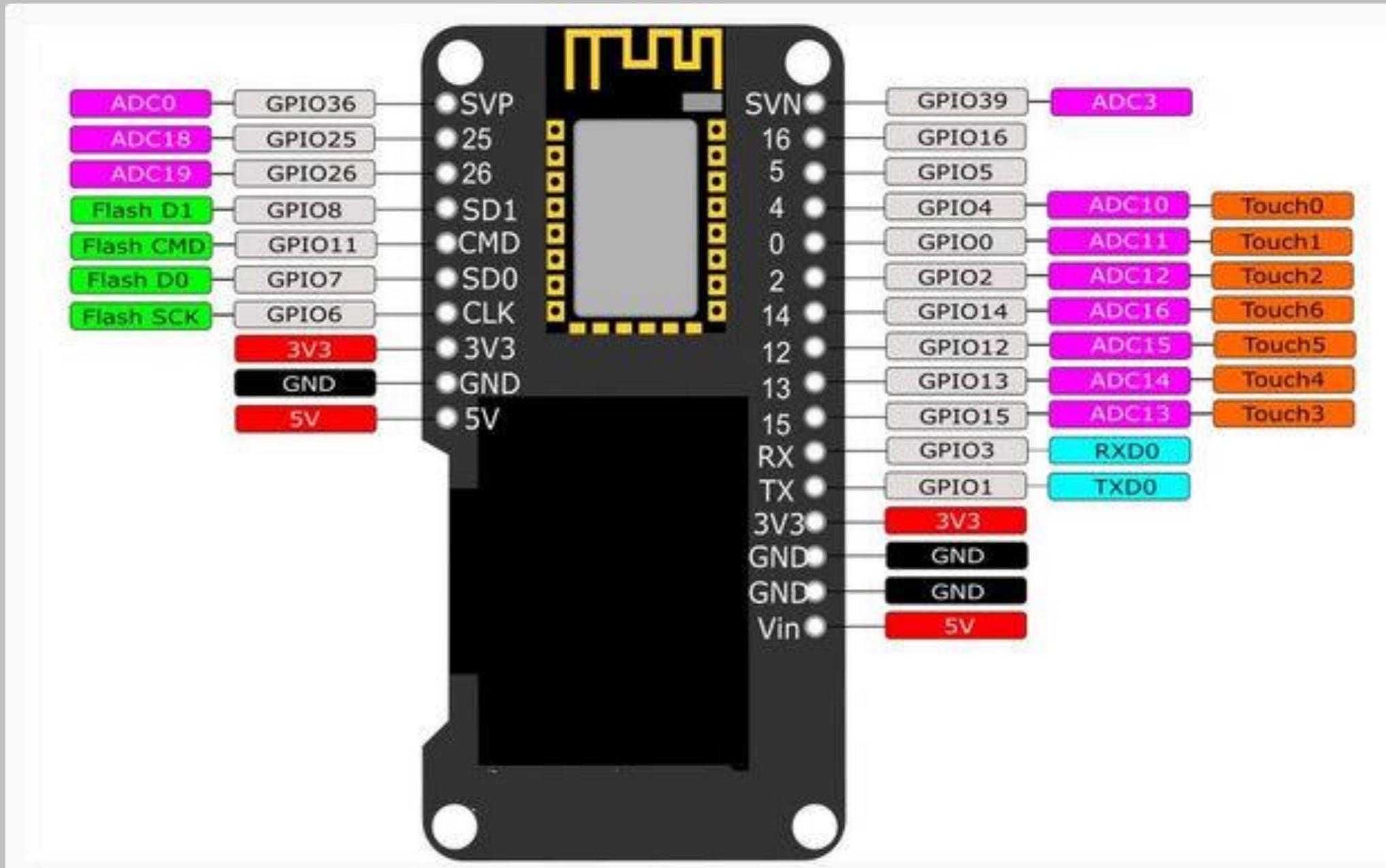
# Biblioteca

Para utilizar o display oled precisamos configurar a biblioteca na IDE do arduino. Para isso, faça o download da biblioteca através do [link](#). Descompacte o arquivo e cole na pasta de bibliotecas da IDE do arduino.

*C:/Program Files (x86)/Arduino/libraries*

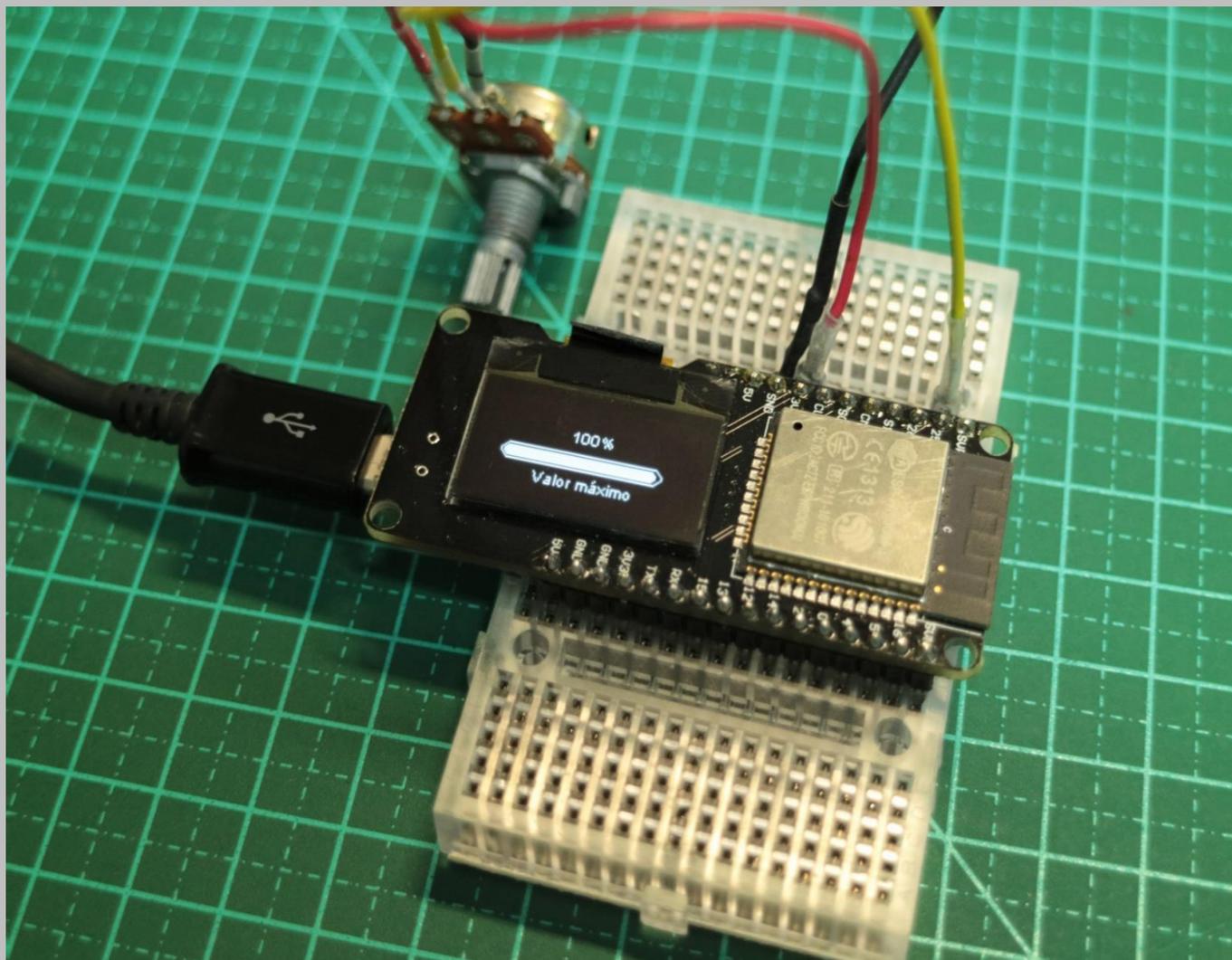


# Wemos Lolin ESP32 OLED



# Exemplo

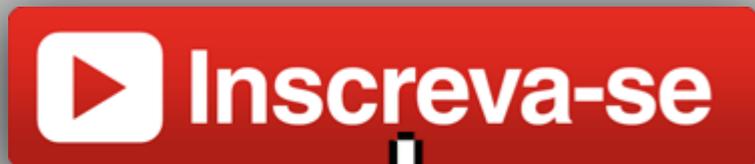
**Veremos agora como utilizar o display oled para exibir uma barra de progresso controlada por um potenciômetro.**



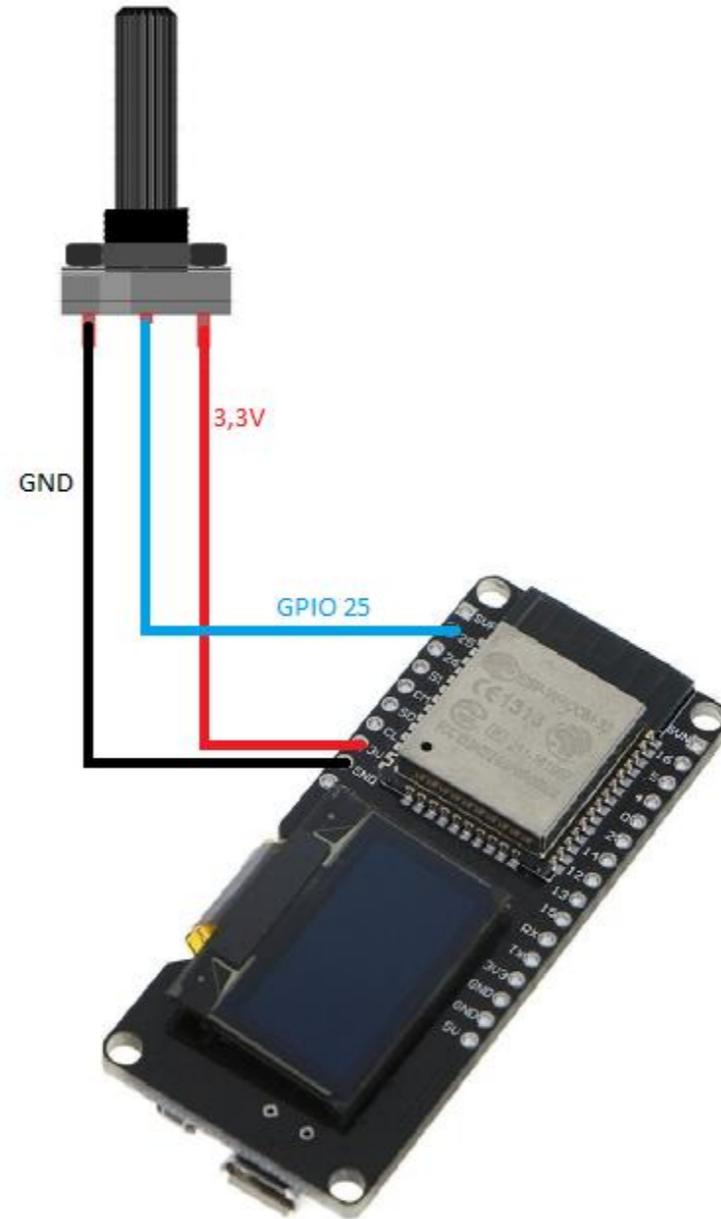
## EXEMPLO FUNCIONANDO

Em [www.fernandok.com](http://www.fernandok.com)

Download arquivo **PDF** dos diagramas  
Download arquivo **INO** do código fonte



# Montagem



# Código

Primeiramente adicionamos a biblioteca “SSD1306.h”, com ela que acessaremos o display oled. Feito isso criamos um objeto “display” do tipo SSD1306 que será o responsável por controlar o conteúdo que será exibido no display oled.

```
#include "SSD1306.h"

//objeto controlador do display de led
/*
0x3c : é um identificador único para comunicação do display
pino 5 e 4 são os de comunicação (SDA, SDC) i2c
*/
SSD1306 screen(0x3c, 5, 4);

//pino que ligamos o potenciometro
#define PINO_POTENCIOMETRO 25

//utilizado para fazer o contador de porcentagem
int percProgress;
```



# setup

**Na função setup(), vamos inicializar nosso objeto display para podermos controlar o que será exibido. Através desse objeto também vamos configurar a fonte de escrita para os textos que serão exibidos. E por fim, configuramos o pino que ligamos o potenciômetro como INPUT para fazermos leitura do valor.**

```
void setup()
{
  // Inicializa o objeto que controlará o que será exibido na tela
  screen.init();

  //setFont recebe como parâmetro o tamanho da fonte de escrita
  /*pode-se utilizar as seguintes fontes:
   * ArialMT_Plain_10
   * ArialMT_Plain_16
   * ArialMT_Plain_24
  */
  screen.setFont(ArialMT_Plain_10);

  //configura o pino para fazer a leitura do potenciômetro.
  pinMode(PINO_POTENCIOMETRO, INPUT);
}
```



# loop

**Na função loop(), faremos a leitura do valor atual do potenciômetro, repare que estamos usando da função “map” logo após a leitura do valor, isso porque o valor lido é muito alto para colocar em uma barra de progresso, então mapearemos o valor para estar dentro do intervalo de 0 a 100.**

```
void loop()
{
    //leitura do valor do potenciômetro
    int valor = analogRead(PINO_POTENCIOMETRO);

    //mapeando o valor do potenciômetro para o valor da barra de progresso
    //potenciômetro faz a leitura do valor no intervalo de 0 a 4095
    //a barra de progresso espera um valor entre 0 e 100
    percProgress = map(valor, 0, 4095, 0, 100);

    //limpa todo o display, apaga o conteúdo da tela
    screen.clear();

    //desenha a barra de progresso
    desenhaBarraProgresso();

    //exibe na tela o que foi configurado até então.
    screen.display();
    delay(10);
}
```



**Na função “desenhaBarraProgresso()”, utilizaremos o valor lido do potenciômetro que está salvo na variável “percProgress” para setar na barra de progresso. Também vamos colocar um texto logo acima da barra de progresso, indicando a porcentagem atual.**

```
//função para desenhar a progress bar no display
void desenhaBarraProgresso() {

    // desenha a progress bar
    /* drawProgressBar(x, y, width, height, value);*/
    screen.drawProgressBar(0, 32, 100, 20, contador);

    // configura o alinhamento do texto que será escrito
    //nesse caso alinharemos o texto ao centro
    screen.setTextAlignment(TEXT_ALIGN_CENTER);
    //escreve o texto de porcentagem
    /* drawString(x,y,text); */
    screen.drawString(64, 15, String(contador) + "%");

    //se o percProgress está em zero, escreve a string "valor mínimo" abaixo da barra
    de progresso
    if(percProgress == 0){
        screen.drawString(64, 45, "Valor mínimo");
    }
    //se o percProgress está em 100, escreve a string "valor máximo" abaixo da barra
    de progresso
    else if(percProgress == 100){
        screen.drawString(64, 45, "Valor máximo");
    }
}
```



# Algumas outras funções interessantes

## Display

*//coloca o display de ponta cabeça*

**void flipScreenVertically();**

## Desenho

*//desenha um único pixel da tela*

**void setPixel(int16\_t x, int16\_t y);**

*//desenha uma linha*

**void drawLine(int16\_t x0, int16\_t y0, int16\_t x1, int16\_t y1);**

*//desenha um retângulo*

**void drawRect(int16\_t x, int16\_t y, int16\_t width, int16\_t height);**

*//desenha um círculo*

**void drawCircle(int16\_t x, int16\_t y, int16\_t radius);**

*// preenche um círculo*

**void fillCircle(int16\_t x, int16\_t y, int16\_t radius);**

*// desenha uma linha horizontal*

**void drawHorizontalLine(int16\_t x, int16\_t y, int16\_t length);**

*// desenha uma linha vertical*

**void drawVerticalLine(int16\_t x, int16\_t y, int16\_t length);**

## Texto

*//configura o alinhamento do texto que será escrito*

*// TEXT\_ALIGN\_LEFT, TEXT\_ALIGN\_CENTER, TEXT\_ALIGN\_RIGHT, TEXT\_ALIGN\_CENTER\_BOTH*

**void setTextAlignment(OLEDDISPLAY\_TEXT\_ALIGNMENT textAlignment);**



Em [www.fernandok.com](http://www.fernandok.com)

Download arquivo **PDF** dos diagramas  
Download arquivo **INO** do código fonte

