

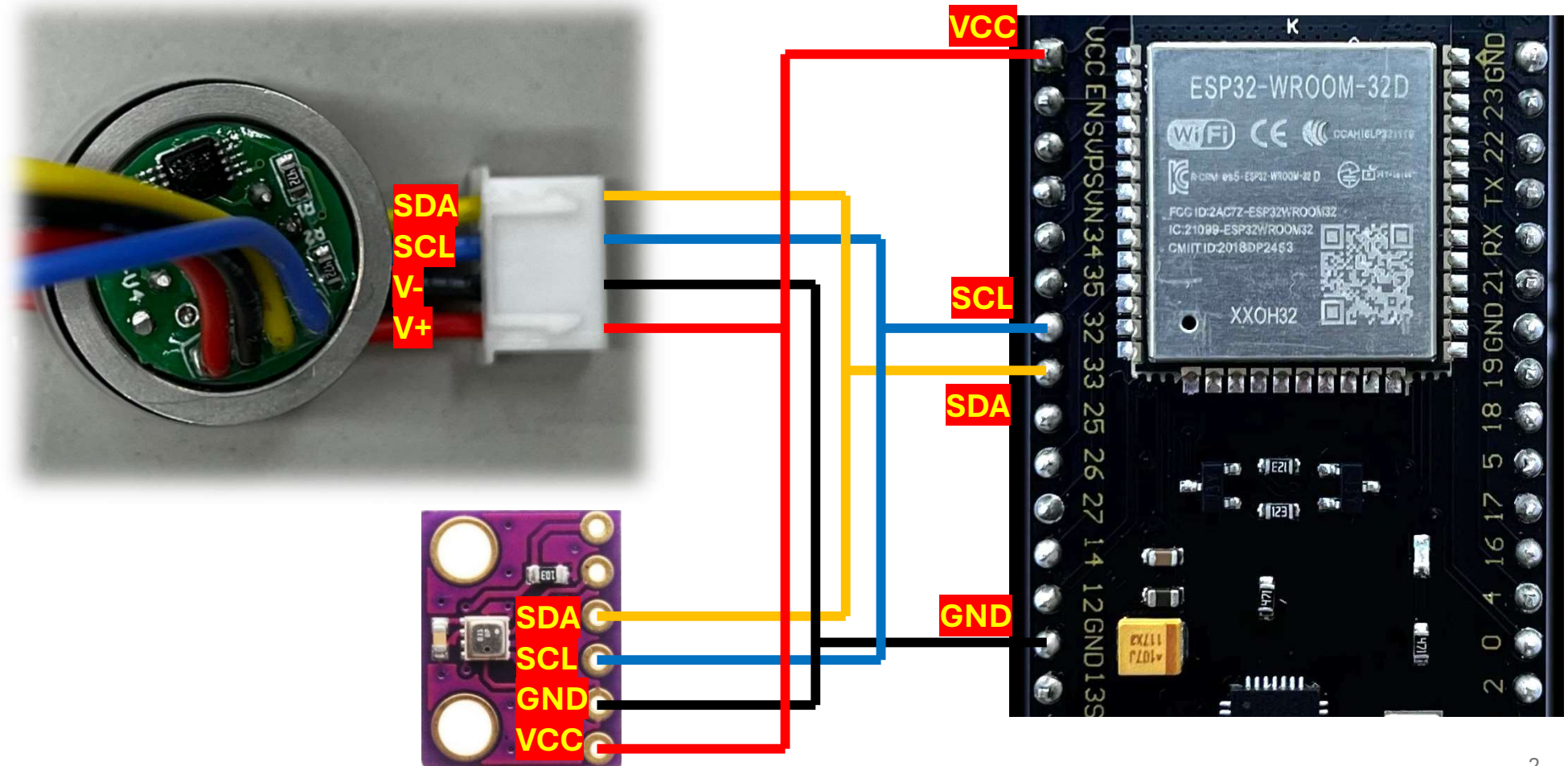
# Projetando dispositivos comerciais

**Handson**

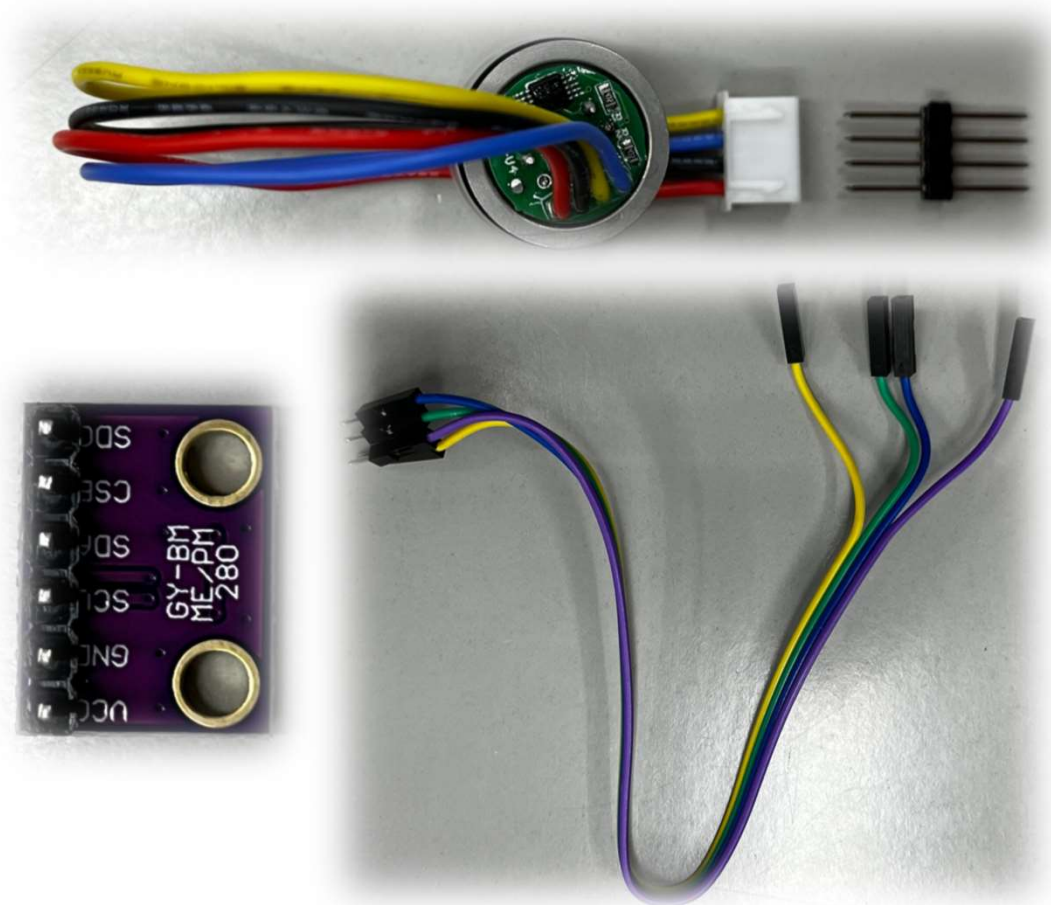
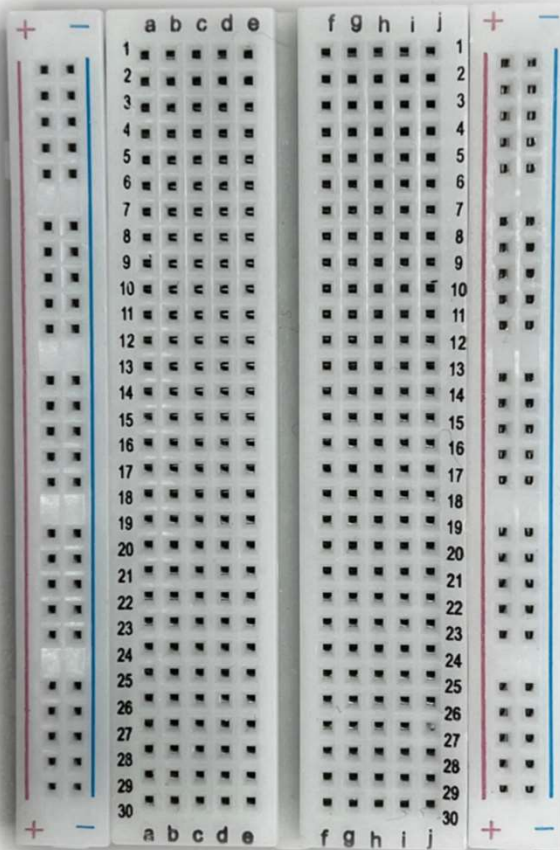


TEX.com.br

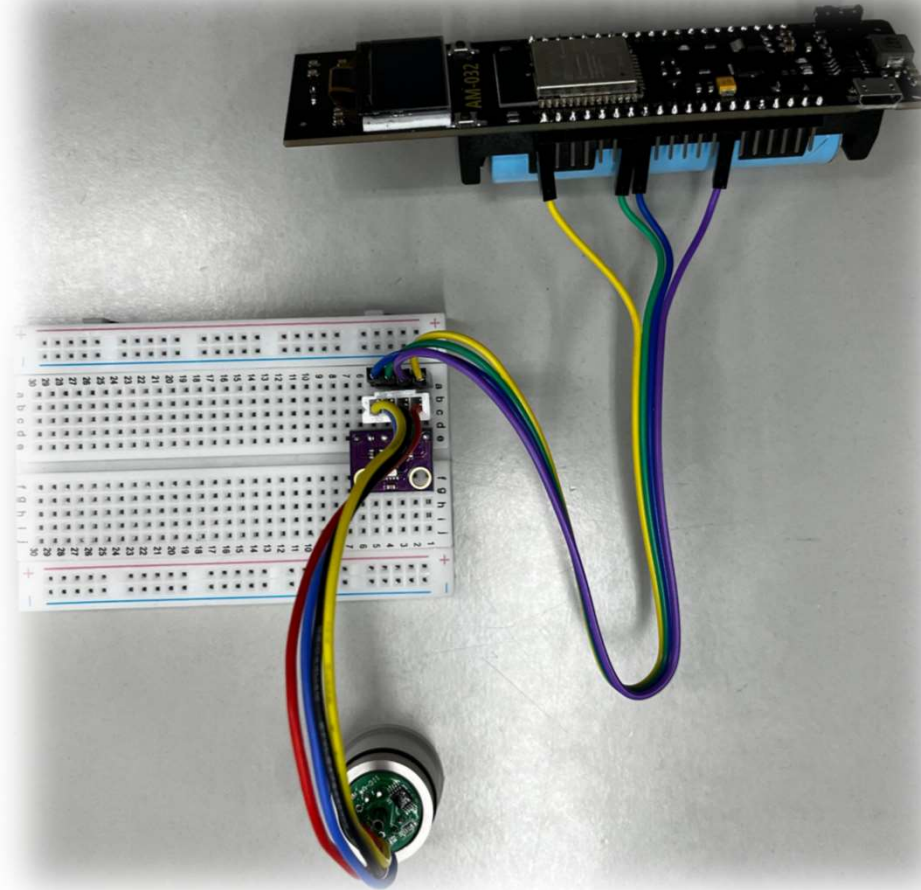
# Integrando o MVP



# Integrando o MVP

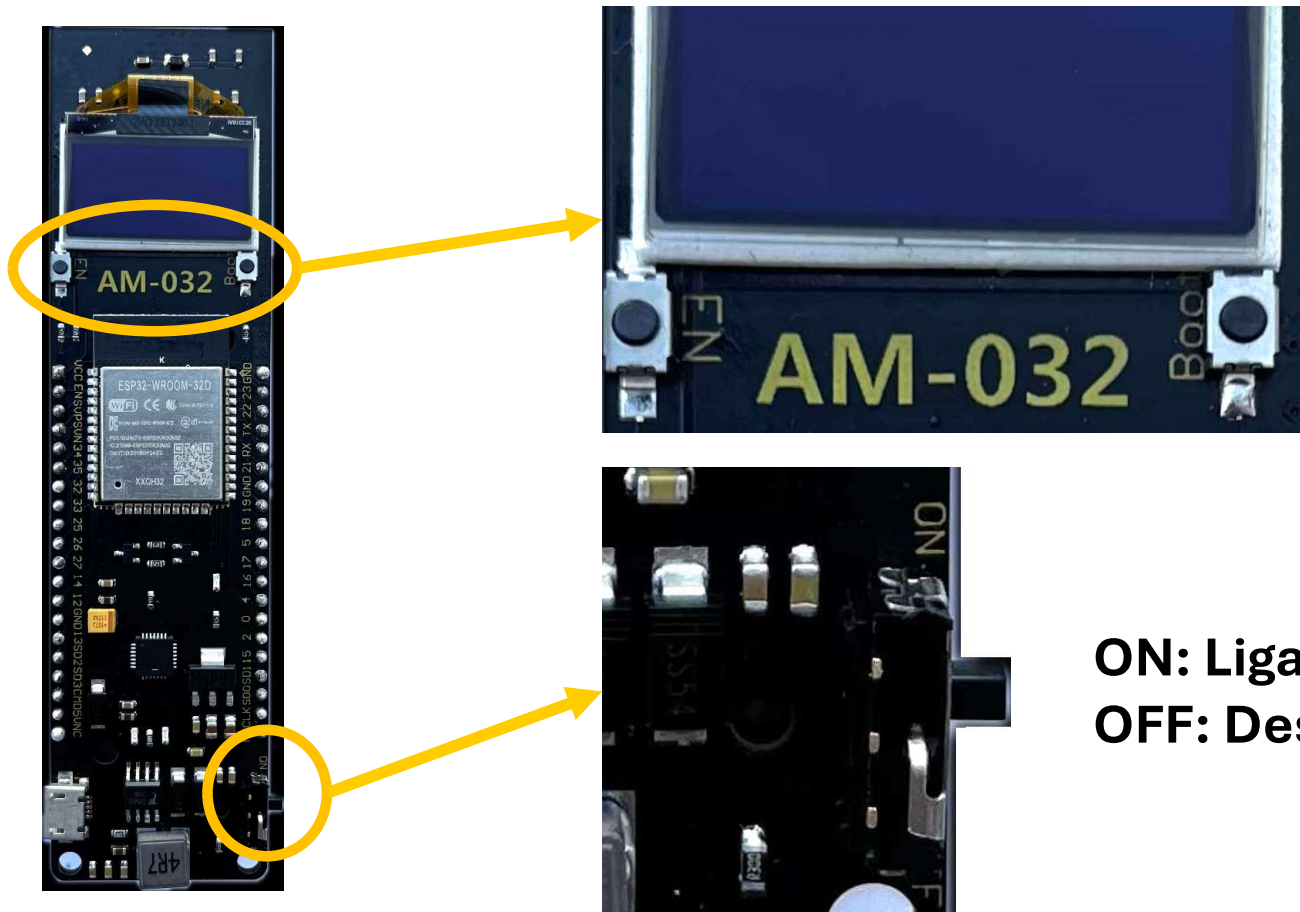


# Integrando o MVP





# Funções dos botões e chave



**EN: RESET**  
**Boot: Modo de gravação**

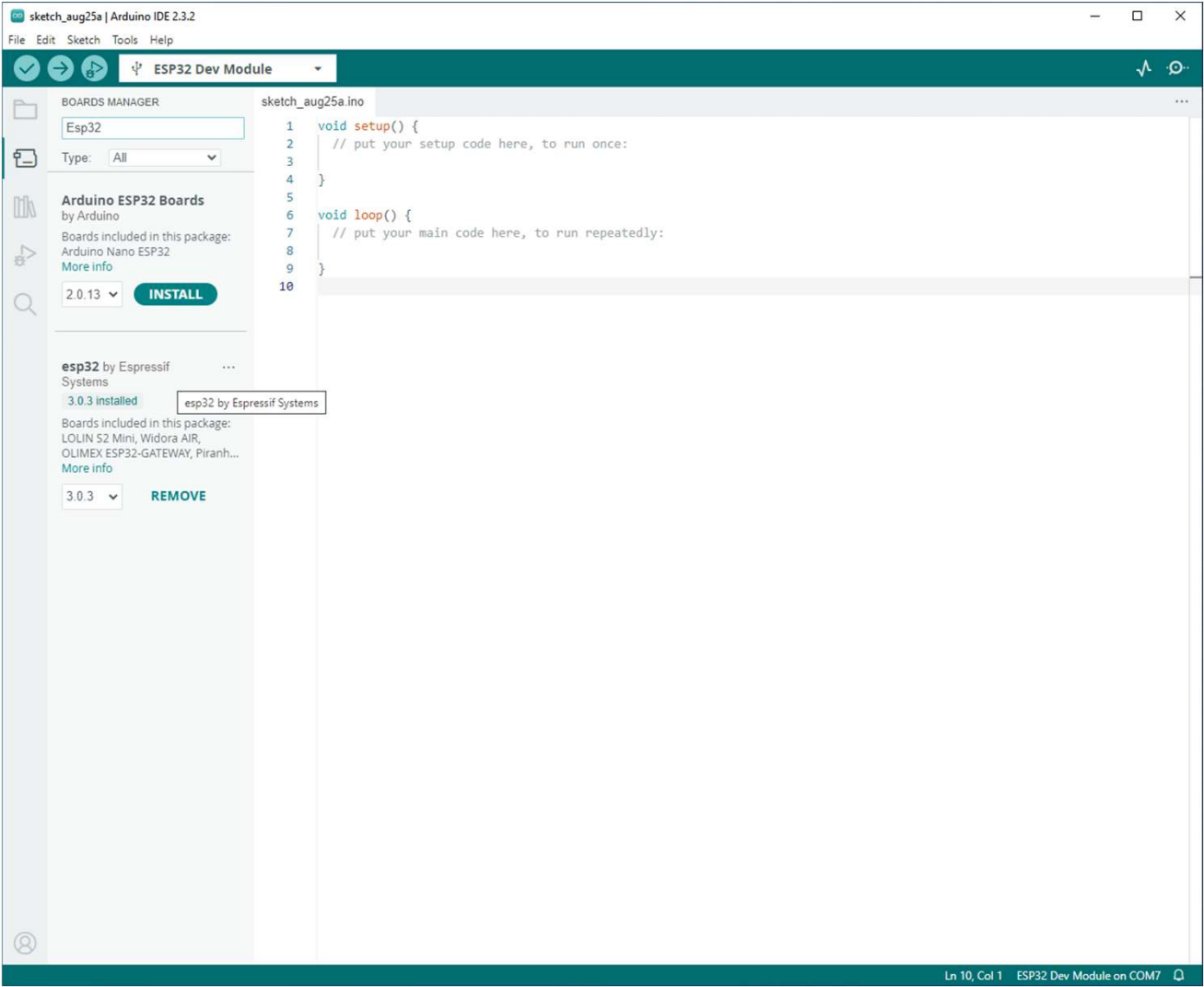
**ON: Liga**  
**OFF: Desliga**

# **Configurando ambiente Arduino**





# Instalar via Board Manager : esp32 by Espressif (3.0.3)





# Instalar via *Library Manager* : **Adafruit SSD1306 by Adafruit (2.5.11)**

sketch\_aug25a | Arduino IDE 2.3.2

File Edit Sketch Tools Help

ESP32 Dev Module

LIBRARY MANAGER

SSD1306

Type: All

Topic: All

More info

1.0.1

INSTALL

Adafruit SSD1306 by Adafruit

Adafruit

2.5.11

SSD1306 oled driver library for monochrome 128x64 and 128x32 displays SSD1306 oled...

More info

2.5.11

REMOVE

Adafruit SSD1306 EMULATOR by Adafruit...

SSD1306 emulator oled driver library for monochrome 128x64 and 128x32 displays SSD1306...

More info

0.1.0

INSTALL

Adafruit SSD1306 Wemos Mini OLED b...

SSD1306 oled driver library for Wemos D1 Mini OLED shield This is based on the Adafruit...

More info

1.1.2

INSTALL

Bonezegei\_SSD1306 by Bonezegei (Jofel Batutay)

OLED Library I2C SSD1306 OLED Library

More info

sketch\_aug25a.ino

1 void setup() {

2 // put your setup code here, to run once:

3 }

4 }

5 }

6 void loop() {

7 // put your main code here, to run repeatedly:

8 }

9 }

10 }

Ln 10, Col 1 ESP32 Dev Module on COM7





# Instalar via *Library Manager* : **Adafruit BMP280 Library by Adafruit (2.6.8)**

sketch\_aug25a | Arduino IDE 2.3.2

File Edit Sketch Tools Help

ESP32 Dev Module

LIBRARY MANAGER

BMP280

Type: All

Topic: All

BMP280 by dvarrel

DFRobot Standard library modified by dvarrel Used to read current temperature, air...

More info

1.0.3

INSTALL

Adafruit BMP280 Library by Adafruit

2.6.8 installed

Adafruit BMP280 Library by Adafruit

Arduino library for BMP280 sensors. Arduino library for BMP280 pressure and altitude...

More info

2.6.8

REMOVE

BMP280\_DEV by Martin Lindupp

An Arduino compatible, non-blocking, I2C/SPI library for the Bosch BMP280 barometer. Thi...

More info

1.0.21

INSTALL

BMx280MI by Gregor Christandl...

A library for the Bosch Sensortec BME280 and BMP280 Digital Pressure Sensors. The...

More info

1.2.3

INSTALL

CanSat Kit Library by

sketch\_aug25a.ino

1 void setup() {

2 // put your setup code here, to run once:

3 }

4 }

5 }

6 void loop() {

7 // put your main code here, to run repeatedly:

8 }

9 }

10 }

Ln 10, Col 1

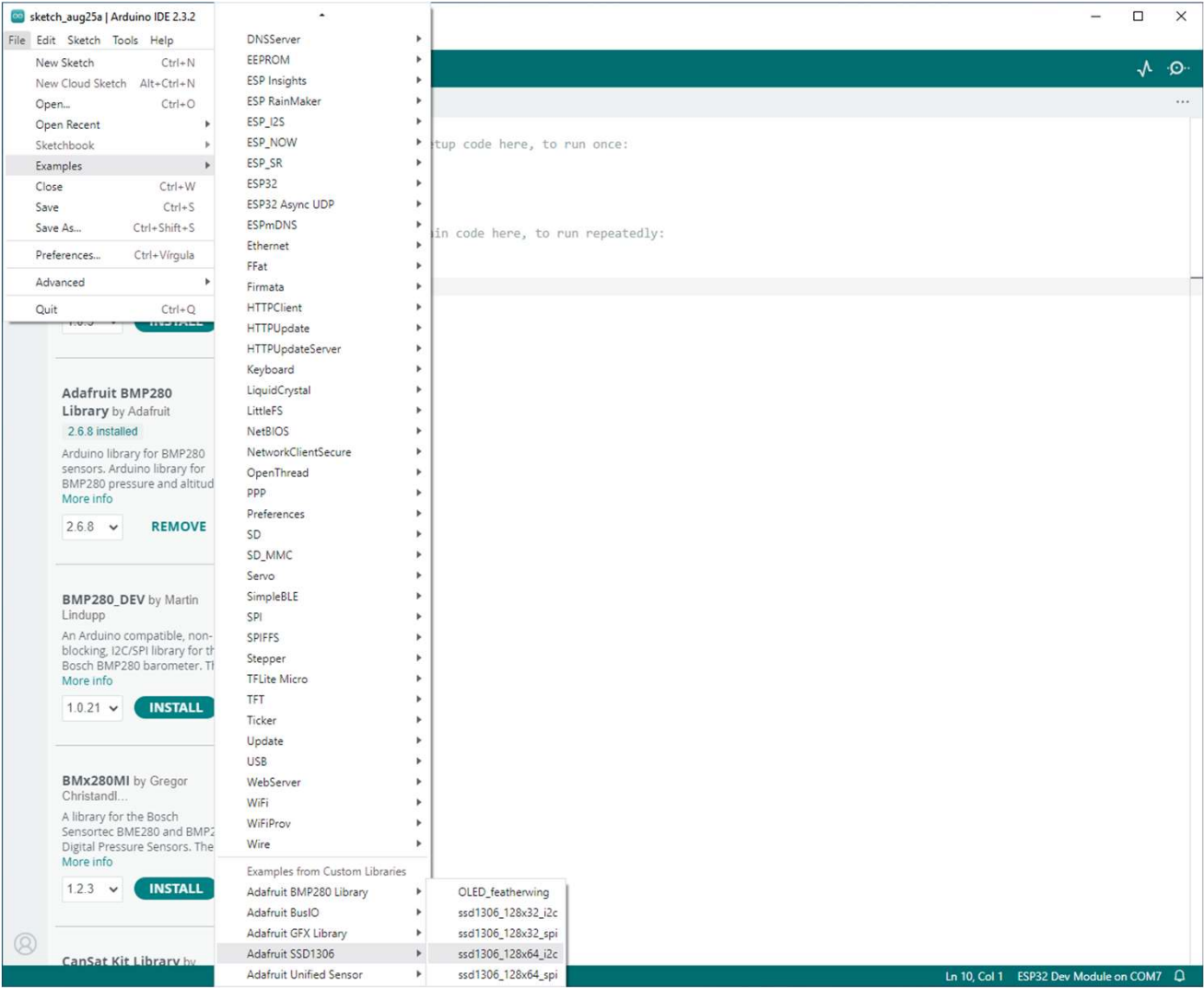
ESP32 Dev Module on COM7

# **Exemplo Arduino: Display OLED SSD1306**



[https://github.com/eng-software/HandsonINO/tree/main/OLED\\_SSD1306](https://github.com/eng-software/HandsonINO/tree/main/OLED_SSD1306)

Carregar o exemplo: **Adafruit SSD1306 -> ssd1306\_128x64\_i2c**





## Alterar o **SCREEN\_ADDRESS** para **0x3C**

Para o kit disponibilizado, o endereço I2C do display é 0x3C

```
ESP32 Dev Module

ssd1306_128x64_i2c.ino
20
21 #include <SPI.h>
22 #include <Wire.h>
23 #include <Adafruit_GFX.h>
24 #include <Adafruit_SSD1306.h>
25
26 #define SCREEN_WIDTH 128 // OLED display width, in pixels
27 #define SCREEN_HEIGHT 64 // OLED display height, in pixels
28
29 // Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
30 // The pins for I2C are defined by the Wire-library.
31 // On an arduino UNO:      A4(SDA), A5(SCL)
32 // On an arduino MEGA 2560: 20(SDA), 21(SCL)
33 // On an arduino LEONARDO:  2(SDA),  3(SCL)
34 #define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)
35 #define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32
36 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
37
38 #define NUMFLAKES 10 // Number of snowflakes in the animation example
39
40 #define LOGO_HEIGHT 16
41 #define LOGO_WIDTH 16
42 static const unsigned char PROGMEM logo_bmp[] =
43 { 0b00000000, 0b10000000,
44   0b00000001, 0b11000000,
45   0b00000001, 0b11000000,
46   0b00000011, 0b11000000,
47   0b11110011, 0b11000000,
48   0b11111110, 0b11110000,
49   0b01111110, 0b11111111,
50   0b00110011, 0b10011111,
51   0b00011111, 0b11111100,
52   0b00011011, 0b01110000,
53   0b00011011, 0b10100000,
54   0b00111111, 0b11100000,
55   0b00111111, 0b11100000,
56   0b01111100, 0b11100000,
57   0b01110000, 0b01110000,
58   0b00000000, 0b00110000 };
59
60 void setup() {
61   Serial.begin(9600);
62
63   // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
64   if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
65     Serial.println(F("SSD1306 allocation failed"));
66     for(;;); // Don't proceed, loop forever
  
```



Incluir na função **setup()** a linha em destaque:

Inicializamos o canal I2C 0 do ESP32 (Wire)

Pinos:

5 - SDA

4 - SCL

Frequência: 400KHz

```
ssid1306_128x64_i2c.ino
43 { 0b00000000, 0b11000000,
44     0b00000001, 0b11000000,
45     0b00000001, 0b11000000,
46     0b00000011, 0b11100000,
47     0b11110011, 0b11100000,
48     0b11111110, 0b11111000,
49     0b01111110, 0b11111111,
50     0b00110011, 0b10011111,
51     0b00011111, 0b11111100,
52     0b00001101, 0b01110000,
53     0b00011011, 0b10100000,
54     0b00111111, 0b11100000,
55     0b00111111, 0b11110000,
56     0b01111100, 0b11110000,
57     0b01110000, 0b01110000,
58     0b00000000, 0b00110000 };
59
60 void setup() {
61     Serial.begin(9600);
62
63     Wire.begin(5, 4, 400000);
64
65     // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
66     if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
67         Serial.println(F("SSD1306 allocation failed"));
68         for(;;); // Don't proceed, loop forever
69     }
70
71     // Show initial display buffer contents on the screen --
72     // the library initializes this with an Adafruit splash screen.
73     display.display();
74     delay(2000); // Pause for 2 seconds
75
76     // Clear the buffer
77     display.clearDisplay();
78
79     // Draw a single pixel in white
80     display.drawPixel(10, 10, SSD1306_WHITE);
81
82     // Show the display buffer on the screen. You MUST call display() after
83     // drawing commands to make them visible on screen!
84     display.display();
85     delay(2000);
86     // display.display() is NOT necessary after every single drawing command,
87     // unless that's what you want...rather, you can batch up a bunch of
88     // drawing operations and then update the screen all at once by calling
89     // display.display(). These examples demonstrate both approaches...
90 }
```

Agora compile e grave na placa.

O exemplo da Adafruit no display deverá aparecer

# **Exemplo Arduino: Sensor BMP280**



<https://github.com/eng-software/HandsonINO/tree/main/BMP280>



Agora, carregue o exemplo : **Adafruit BMP280 Library -> bmp280test**



```
File Edit Sketch Tools Help
New Sketch Ctrl+N
New Cloud Sketch Alt+Ctrl+N
Open... Ctrl+O
Open Recent
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Preferences... Ctrl+Virgula
Advanced
Quit Ctrl+Q

12 // Hardware by p
13
14 Written by Li
15 with contribu
16 BSD license,
17 All text abov
18 included in a
19 *****
20
21 #include <SPI>
22 #include <Wire>
23 #include <Adafruit_SSD1306>
24 #include <Adafruit_BMP280>
25
26 #define SCREEN_WIDTH 128 // OLED display width, in pixels
27 #define SCREEN_HEIGHT 64 // OLED display height, in pixels
28
29 // Declaration
30 // The pins for the display
31 // On an Arduino Uno or Mega
32 // On an Arduino Pro Mini
33 // On an Arduino Pro Micro
34 #define OLED_RESET -1 // Reset pin (if sharing Arduino reset pin)
35 #define SCREEN_WIDTH 128 // OLED display width, in pixels
36 #define SCREEN_HEIGHT 64 // OLED display height, in pixels
37
38 #define NUMFLANES 1 // The number of planes to display
39
40 #define LOGO_HEIGHT 10
41 #define LOGO_WIDTH 10
42 static const unsigned char PROGMEM logo_bmp[] = {
43 0b00000000,
44 0b00000001,
45 0b00000001,
46 0b00000011,
47 0b11110011,
48 0b11111110
49
50 // ...
51 // -1 if sharing Arduino reset pin)
52 // set for Address; 0x3D for 128x64, 0x3C for 128x32
53 // N_HEIGHT, &Wire, OLED_RESET);
54
55 // Takes in the animation example
56
57 // =
58 }
```



Altere a instancia de **bmp** como mostrado em destaque

Inicializamos o  
BMP280 no canal  
I2C 1 do ESP32  
(Wire1)

```
bmp280test | Arduino IDE 2.3.2
File Edit Sketch Tools Help
ESP32 Dev Module

bmp280test.ino
4  Designed specifically to work with the Adafruit BMP280 Breakout
5  ----> http://www.adafruit.com/products/2651
6
7  These sensors use I2C or SPI to communicate, 2 or 4 pins are required
8  to interface.
9
10 Adafruit invests time and resources providing this open source code,
11 please support Adafruit and open-source hardware by purchasing products
12 from Adafruit!
13
14 Written by Limor Fried & Kevin Townsend for Adafruit Industries.
15 BSD license, all text above must be included in any redistribution
16 *****/
17
18 #include <Wire.h>
19 #include <SPI.h>
20 #include <Adafruit_BMP280.h>
21
22 #define BMP_SCK  (13)
23 #define BMP_MISO (12)
24 #define BMP_MOSI (11)
25 #define BMP_CS   (10)
26
27 Adafruit_BMP280 bmp(&Wire1); // I2C
28 //Adafruit_BMP280 bmp(BMP_CS); // hardware SPI
29 //Adafruit_BMP280 bmp(BMP_CS, BMP_MOSI, BMP_MISO, BMP_SCK);
30
31 void setup() {
32   Serial.begin(9600);
33   while ( !Serial ) delay(100); // wait for native usb
34   Serial.println(F("BMP280 test"));
35   unsigned status;
36   //status = bmp.begin(BMP280_ADDRESS_ALT, BMP280_CHIPID);
37   status = bmp.begin();
38   if (!status) {
39     Serial.println(F("Could not find a valid BMP280 sensor, check wiring or "
40     "try a different address!"));
41     Serial.print("SensorID was: 0x"); Serial.println(bmp.sensorID(),16);
42     Serial.print("    ID of 0xFF probably means a bad address, a BMP 180 or BMP 085\n");
43     Serial.print("    ID of 0x56-0x58 represents a BMP 280.\n");
44     Serial.print("    ID of 0x60 represents a BME 280.\n");
45     Serial.print("    ID of 0x61 represents a BME 680.\n");
46     while (1) delay(10);
47   }
48
49   /* Default settings from datasheet. */
50   bmp.setSampling(Adafruit_BMP280::MODE_NORMAL, /* Operating Mode. */
51   Ln 27, Col 36  ESP32 Dev Module on COM7
```



Na função **setup()** adicione a linha em destaque:

Inicializamos o I2C  
1 do ESP32 (Wire1)  
Pinos:  
33 - SDA  
32 - SCL  
Frequência: 100KHz

```
ESP32 Dev Module
bmp280test.ino
9
10 Adafruit invests time and resources providing this open source code,
11 please support Adafruit and open-source hardware by purchasing products
12 from Adafruit!
13
14 Written by Limor Fried & Kevin Townsend for Adafruit Industries.
15 BSD license, all text above must be included in any redistribution
16 *****/
17
18 #include <Wire.h>
19 #include <SPI.h>
20 #include <Adafruit_BMP280.h>
21
22 #define BMP_SCK (13)
23 #define BMP_MISO (12)
24 #define BMP_MOSI (11)
25 #define BMP_CS (10)
26
27 Adafruit_BMP280 bmp(&Wire1); // I2C
28 //Adafruit_BMP280 bmp(BMP_CS); // hardware SPI
29 //Adafruit_BMP280 bmp(BMP_CS, BMP_MOSI, BMP_MISO, BMP_SCK);
30
31 void setup() {
32   Serial.begin(9600);
33   while ( !Serial ) delay(100); // wait for native usb
34
35   Wire1.begin(33, 32, 100000);
36
37   Serial.println(F("BMP280 test"));
38   unsigned status;
39   //status = bmp.begin(BMP280_ADDRESS_ALT, BMP280_CHIPID);
40   status = bmp.begin();
41   if (!status) {
42     Serial.println(F("Could not find a valid BMP280 sensor, check wiring or "
43     | "try a different address!"));
44     Serial.print("SensorID was: 0x"); Serial.println(bmp.sensorID(),16);
45     Serial.print(" ID of 0xFF probably means a bad address, a BMP 180 or BMP 085\n");
46     Serial.print(" ID of 0x56-0x58 represents a BMP 280,\n");
47     Serial.print(" ID of 0x60 represents a BME 280.\n");
48     Serial.print(" ID of 0x61 represents a BME 680.\n");
49     while (1) delay(10);
50   }
51
52   /* Default settings from datasheet. */
53   bmp.setSampling(Adafruit_BMP280::MODE_NORMAL, /* Operating Mode. */
54   | Adafruit_BMP280::SAMPLING_X2, /* Temp. oversampling */
55   | Adafruit_BMP280::SAMPLING_X16, /* Pressure oversampling */
56   | Adafruit_BMP280::FILTER_X16, /* Filtering */
57   | Adafruit_BMP280::MODE_SLEEP);
58 }
```



E altere a inicialização do **bmp** como em destaque:

```
ESP32 Dev Module

bmp280test.ino

9
10 Adafruit invests time and resources providing this open source code,
11 please support Adafruit and open-source hardware by purchasing products
12 from Adafruit!
13
14 Written by Limor Fried & Kevin Townsend for Adafruit Industries.
15 BSD license, all text above must be included in any redistribution
16 *****/
17
18 #include <Wire.h>
19 #include <SPI.h>
20 #include <Adafruit_BMP280.h>
21
22 #define BMP_SCK (13)
23 #define BMP_MISO (12)
24 #define BMP_MOSI (11)
25 #define BMP_CS (10)
26
27 Adafruit_BMP280 bmp(&Wire1); // I2C
28 //Adafruit_BMP280 bmp(BMP_CS); // hardware SPI
29 //Adafruit_BMP280 bmp(BMP_CS, BMP_MOSI, BMP_MISO, BMP_SCK);
30
31 void setup() {
32   Serial.begin(9600);
33   while ( !Serial ) delay(100); // wait for native usb
34
35   Wire1.begin(33, 32, 100000);
36
37   Serial.println(F("BMP280 test"));
38   unsigned status;
39   //status = bmp.begin(BMP280_ADDRESS_ALT, BMP280_CHIPID);
40   status = bmp.begin(0x76);
41   if (!status) {
42     Serial.println(F("Could not find a valid BMP280 sensor, check wiring or "
43       "try a different address!"));
44     Serial.print("SensorID was: 0x"); Serial.println(bmp.sensorID(),16);
45     Serial.print("      ID of 0xFF probably means a bad address, a BMP 180 or BMP 085\n");
46     Serial.print("      ID of 0x56-0x58 represents a BMP 280,\n");
47     Serial.print("      ID of 0x60 represents a BME 280.\n");
48     Serial.print("      ID of 0x61 represents a BME 680.\n");
49     while (1) delay(10);
50   }
51
52   /* Default settings from datasheet. */
53   bmp.setSampling(Adafruit_BMP280::MODE_NORMAL, /* Operating Mode. */
54     Adafruit_BMP280::SAMPLING_X2, /* Temp. oversampling */
55     Adafruit_BMP280::SAMPLING_X16, /* Pressure oversampling */
56     Adafruit_BMP280::FILTER_X16, /* Filtering */
57     0, /* Averaging */
58     0, /* Mode */
59     0);
60 }
```

Configuramos o  
endereço do  
BMP280 que é 0x76

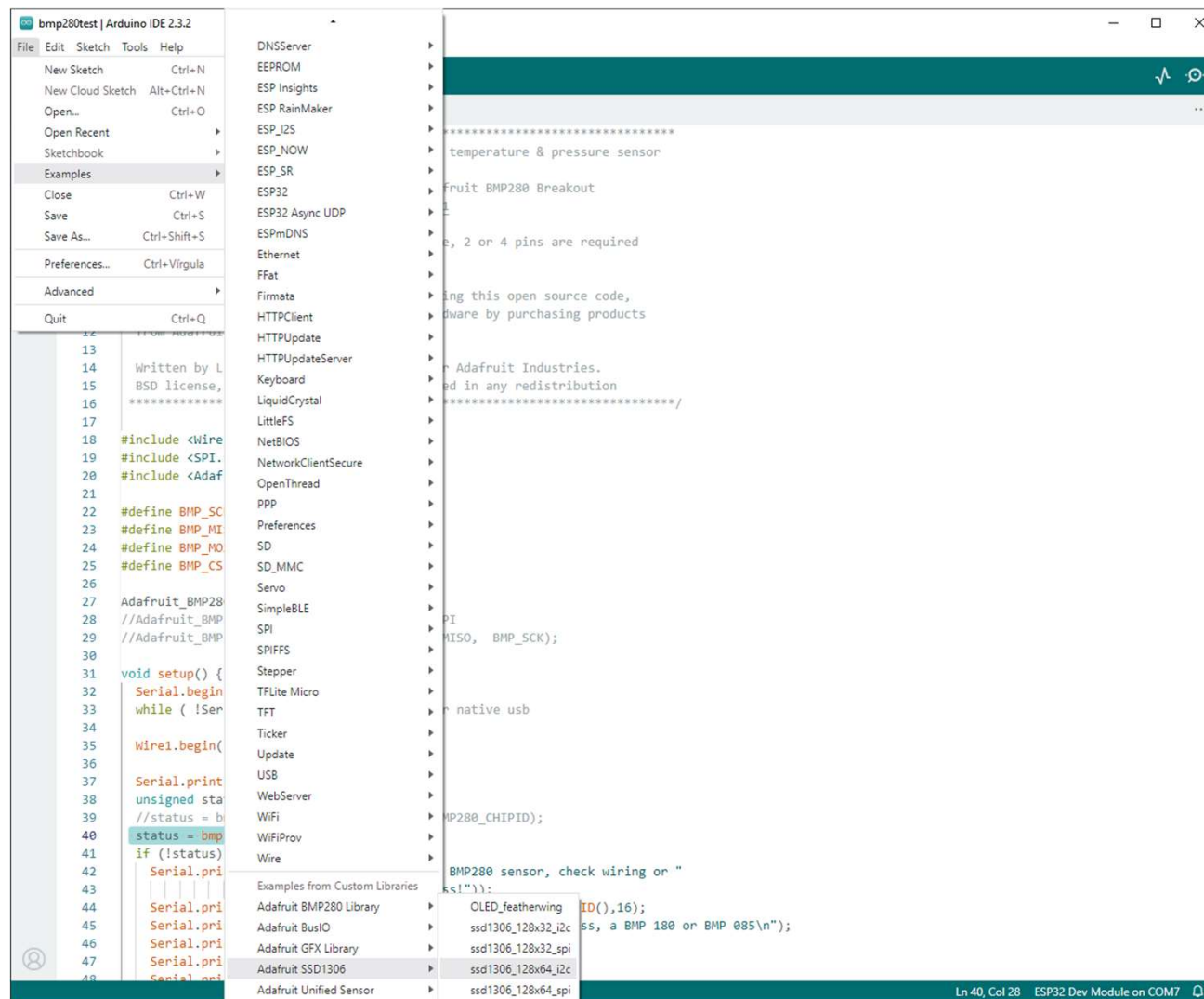
Agora compile e grave na placa.

O exemplo da Adafruit envia no monitor serial a leitura do sensor a cada 2 segundos

# **Exemplo Arduino: Display OLED SSD1306 e BMP280**

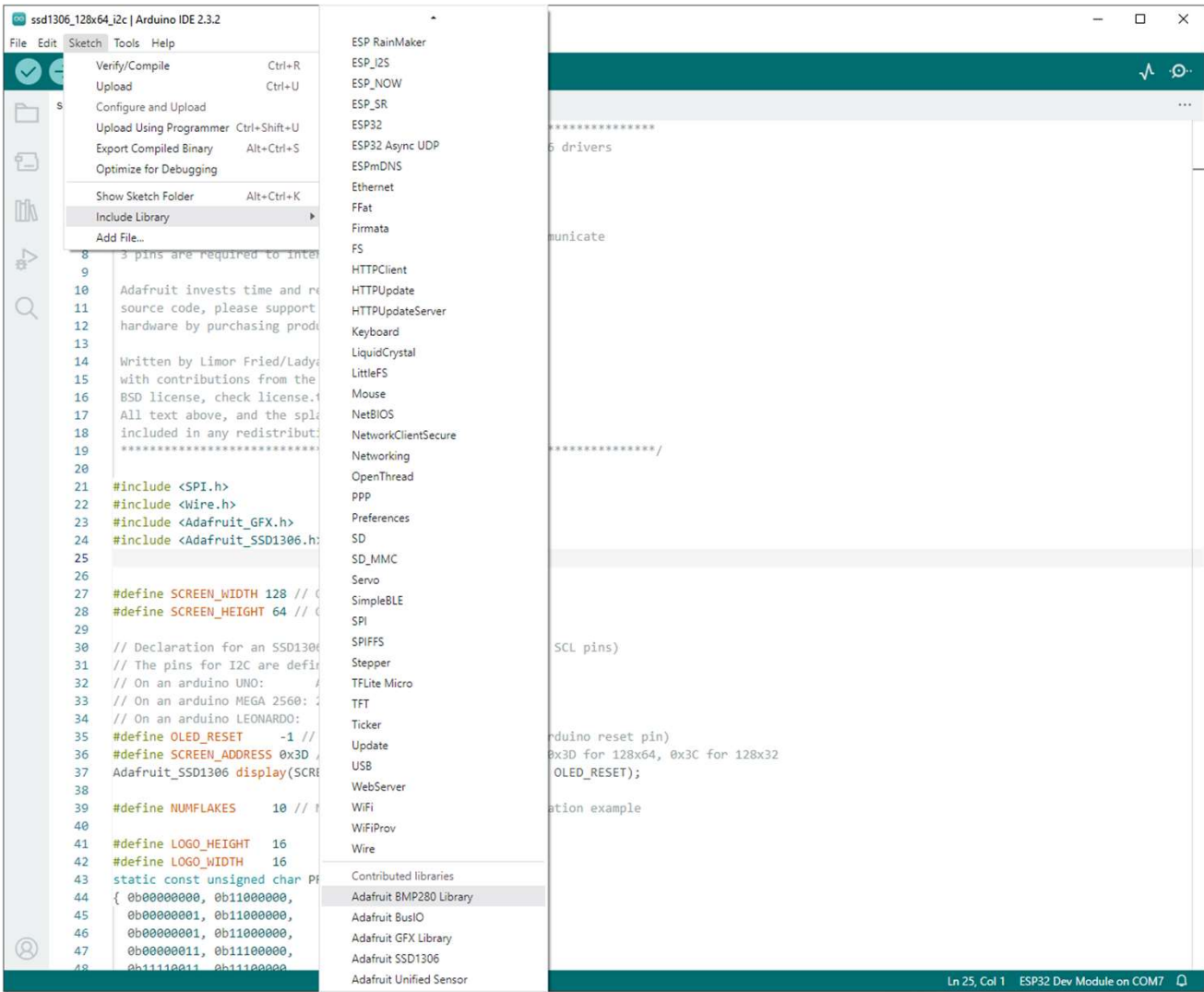


[https://github.com/eng-software/HandsonINO/tree/main/OLED\\_SSD1306\\_BMP280](https://github.com/eng-software/HandsonINO/tree/main/OLED_SSD1306_BMP280)





Inclua a biblioteca: **Adafruit BMP280 Library**





Altere o **SCREEN\_ADDRESS** e instancie o **BMP280** como em destaque

O display SSD1306  
usará o canal I2C 0  
do ESP32 (Wire)

O sensor BMP1306  
usará o canal I2C 1  
do ESP32 (Wire1)

```
ESP32 Dev Module

ssid1306_128x64_i2c.ino
6  Pick one up today in the adafruit shop!
7  -----> http://www.adafruit.com/category/63_98
8
9  This example is for a 128x64 pixel display using I2C to communicate
10 3 pins are required to interface (two I2C and one reset).
11
12  Adafruit invests time and resources providing this open
13  source code, please support Adafruit and open-source
14  hardware by purchasing products from Adafruit!
15
16  Written by Limor Fried/Ladyada for Adafruit Industries,
17  with contributions from the open source community.
18  BSD license, check license.txt for more information
19  All text above, and the splash screen below must be
20  included in any redistribution.
21  *****/
22
23  #include <SPI.h>
24  #include <Wire.h>
25  #include <Adafruit_GFX.h>
26  #include <Adafruit_SSD1306.h>
27  #include <Adafruit_BMP280.h>
28
29  #define SCREEN_WIDTH 128 // OLED display width, in pixels
30  #define SCREEN_HEIGHT 64 // OLED display height, in pixels
31
32  // Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
33  // The pins for I2C are defined by the Wire-library.
34  // On an arduino UNO:   A4(SDA), A5(SCL)
35  // On an arduino MEGA 2560: 20(SDA), 21(SCL)
36  // On an arduino LEONARDO:  2(SDA),  3(SCL), ...
37  #define OLED_RESET    -1 // Reset pin # (or -1 if sharing Arduino reset pin)
38  #define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32
39  Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
40  Adafruit_BMP280 bmp(&Wire1);
41
42  #define NUMFLAKES 10 // Number of snowflakes in the animation example
43
44  #define LOGO_HEIGHT 16
45  #define LOGO_WIDTH 16
46  static const unsigned char PROGMEM logo_bmp[] =
47  { 0b00000000, 0b11000000,
48    0b00000001, 0b11000000,
49    0b00000001, 0b11000000,
50    0b00000011, 0b11000000,
51    0b11110011, 0b11000000,
52    0b11111110, 0b11110000,
53    0b01111110, 0b11111111,
```



Adiciona na função **setup()** as linhas em destaque

Inicializamos os  
dois canais I2C

Inicializamos o  
BMP280 no  
endereço 0x76 e  
verificamos se o  
sensor está  
respondendo

```
ESP32 Dev Module

ssd1306_128x64_i2c.ino

46 static const unsigned char PROGMEM logo_bmp[] =
47 { 0b00000000, 0b11000000,
48 0b00000001, 0b11000000,
49 0b00000001, 0b11000000,
50 0b00000011, 0b11000000,
51 0b11110011, 0b11000000,
52 0b11111110, 0b11111000,
53 0b01111110, 0b11111111,
54 0b00110011, 0b10011111,
55 0b00011111, 0b11111100,
56 0b00011011, 0b01110000,
57 0b00011011, 0b10100000,
58 0b00111111, 0b11100000,
59 0b00111111, 0b11110000,
60 0b01111100, 0b11110000,
61 0b01110000, 0b01110000,
62 0b00000000, 0b00110000 };

63
64 void setup() {
65   Serial.begin(9600);
66
67   Wire.begin(5, 4, 400000);
68   Wire1.begin(33, 32, 100000);
69
70   if( !bmp.begin(0x76) ){
71     Serial.println(F("SBMP280 not found"));
72     for(;;); // Don't proceed, loop forever
73   }
74
75   // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
76   if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
77     Serial.println(F("SSD1306 allocation failed"));
78     for(;;); // Don't proceed, loop forever
79   }
80
81   // Show initial display buffer contents on the screen --
82   // the library initializes this with an Adafruit splash screen.
83   display.display();
84   delay(2000); // Pause for 2 seconds
85
86   // Clear the buffer
87   display.clearDisplay();
88
89   // Draw a single pixel in white
90   display.drawPixel(10, 10, SSD1306_WHITE);
91
92   // Show the display buffer on the screen. You MUST call display() after
```



Usando **#if 0** e **#endif**, remova a execução do exemplo como abaixo

```
ESP32 Dev Module

ssd1306_128x64_i2c.ino
72   for(;;); // Don't proceed, loop forever
73   }
74
75   // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
76   if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
77     Serial.println(F("SSD1306 allocation failed"));
78     for(;;); // Don't proceed, loop forever
79   }
80
81   // Show initial display buffer contents on the screen --
82   // the library initializes this with an Adafruit splash screen.
83   display.display();
84
85   #if 0
86   delay(2000); // Pause for 2 seconds
87
88   // Clear the buffer
89   display.clearDisplay();
90
91   // Draw a single pixel in white
92   display.drawPixel(10, 10, SSD1306_WHITE);
93
94   // Show the display buffer on the screen. You MUST call display() after
95   // drawing commands to make them visible on screen!
96   display.display();
97   delay(2000);
98   // display.display() is NOT necessary after every single drawing command,
99   // unless that's what you want...rather, you can batch up a bunch of
100  // drawing operations and then update the screen all at once by calling
101  // display.display(). These examples demonstrate both approaches...
102
103  testdrawline(); // Draw many lines
104
105  testdrawrect(); // Draw rectangles (outlines)
106
107  testfillrect(); // Draw rectangles (filled)
108
109  testdrawcircle(); // Draw circles (outlines)
110
111  testfillcircle(); // Draw circles (filled)
112
113  testdrawroundrect(); // Draw rounded rectangles (outlines)
114
115  testfillroundrect(); // Draw rounded rectangles (filled)
116
117  testdrawtriangle(); // Draw triangles (outlines)
118
119  testfilltriangle(); // Draw triangles (filled)
120
121  testdrawchar(); // Draw characters of the default font
122
123  testdrawstyles(); // Draw 'stylized' characters
124
125  testscrolltext(); // Draw scrolling text
126
127  testdrawbitmap(); // Draw a small bitmap image
128
129  // Invert and restore display, pausing in-between
130  display.invertDisplay(true);
131  delay(1000);
132  display.invertDisplay(false);
133  delay(1000);
134
135  testanimate(logo_bmp, LOGO_WIDTH, LOGO_HEIGHT); // Animate bitmaps
136  #endif
137  }
138
139  void loop() {
140  }
141
142  void testdrawline() {
143    int16_t i;
144
145    display.clearDisplay(); // Clear display buffer
146
147    for(i=0; i<display.width(); i+=4) {
148      display.drawLine(0, 0, i, display.height()-1, SSD1306_WHITE);
149      display.display(); // Update screen with each newly-drawn line
150      delay(1);
151    }
152  }
```



No final do arquivo crie a função ***displayBMP280()*** como abaixo

Esta função formata  
e desenha no  
display as  
informações de  
temperatura e  
pressão

```
ssd1306_128x64_i2c.ino
393
394 // Initialize 'snowflake' positions
395 for(f=0; f< NUMFLAKES; f++) {
396   icons[f][XPOS] = random(1 - LOGO_WIDTH, display.width());
397   icons[f][YPOS] = -LOGO_HEIGHT;
398   icons[f][DELTAY] = random(1, 6);
399   Serial.print(F("x: "));
400   Serial.print(icons[f][XPOS], DEC);
401   Serial.print(F(" y: "));
402   Serial.print(icons[f][YPOS], DEC);
403   Serial.print(F(" dy: "));
404   Serial.println(icons[f][DELTAY], DEC);
405 }
406
407 for(;;) { // Loop forever...
408   display.clearDisplay(); // Clear the display buffer
409
410   // Draw each snowflake:
411   for(f=0; f< NUMFLAKES; f++) {
412     display.drawBitmap(icons[f][XPOS], icons[f][YPOS], bitmap, w, h, SSD1306_WHITE);
413   }
414
415   display.display(); // Show the display buffer on the screen
416   delay(200); // Pause for 1/10 second
417
418   // Then update coordinates of each flake...
419   for(f=0; f< NUMFLAKES; f++) {
420     icons[f][YPOS] += icons[f][DELTAY];
421     // If snowflake is off the bottom of the screen...
422     if (icons[f][YPOS] >= display.height()) {
423       // Reinitialize to a random position, just off the top
424       icons[f][XPOS] = random(1 - LOGO_WIDTH, display.width());
425       icons[f][YPOS] = -LOGO_HEIGHT;
426       icons[f][DELTAY] = random(1, 6);
427     }
428   }
429 }
430
431
432 void displayBMP280(){
433   display.clearDisplay();
434   display.setTextSize(2); // Normal 1:1 pixel scale
435   display.setTextColor(SSD1306_WHITE); // Draw white text
436   display.setCursor(0,0); // Start at top-left corner
437   display.printf("%6.2fC\n%6.0fPa", bmp.readTemperature(), (float)bmp.readPressure());
438   display.display();
439 }
440
```

Building sketch

Ln 431, Col 1 ESP32 Dev Module on COM7



E na função ***loop()*** faça a chamada da função ***displayBMP280()***

Em ***loop()***  
chamamos a função  
criada para manter  
as medições  
atualizadas no  
display

```
ESP32 Dev Module
ssd1306_128x64_i2c.ino
115 testfillroundrect(); // Draw rounded rectangles (filled)
116
117 testdrawtriangle(); // Draw triangles (outlines)
118
119 testfilltriangle(); // Draw triangles (filled)
120
121 testdrawchar(); // Draw characters of the default font
122
123 testdrawstyles(); // Draw 'stylized' characters
124
125 testscrolltext(); // Draw scrolling text
126
127 testdrawbitmap(); // Draw a small bitmap image
128
129 // Invert and restore display, pausing in-between
130 display.invertDisplay(true);
131 delay(1000);
132 display.invertDisplay(false);
133 delay(1000);
134
135 testanimate(logo_bmp, LOGO_WIDTH, LOGO_HEIGHT); // Animate bitmaps
136 #endif
137 }
138
139 void loop() {
140   displayBMP280();
141 }
142
143 void testdrawline() {
144   int16_t i;
145
146   display.clearDisplay(); // Clear display buffer
147
148   for(i=0; i<display.width(); i+=4) {
149     display.drawLine(0, 0, i, display.height()-1, SSD1306_WHITE);
150     display.display(); // Update screen with each newly-drawn line
151     delay(1);
152   }
153   for(i=0; i<display.height(); i+=4) {
154     display.drawLine(0, 0, display.width()-1, i, SSD1306_WHITE);
155     display.display();
156     delay(1);
157   }
158   delay(250);
159
160   display.clearDisplay();
161 }
```

Este exemplo irá mostrar no display a leitura da pressão e temperatura lida do ***BMP280***