

 <h2 style="text-align: center;">BMP180 : Capteur d'altitude et de pression atmosphérique</h2>	
Sommaire :	
I - Présentation du BMP180.....	1
II - Mise en œuvre du BMP180 avec un ESP32.....	1
III - Mise en œuvre du BME680 avec un Raspberry Pi.....	2

I - Présentation du BMP180

Le **BMP180** est un capteur **I2C** qui permet de mesurer la **pression atmosphérique** et la **température**. Le **BMP180** est la nouvelle génération du capteur Bosch **BMP085**. Aucune modification n'a été apportée au niveau du firmware ce qui permet de continuer à utiliser les librairies et les exemples déjà existants. Le **BMP180** renvoie la pression atmosphérique absolue en **Pascal (Pa)**.

Il faut savoir que si la pression atmosphérique moyenne au niveau de la mer vaut **1013.25 hPa** alors on obtient la variation suivante en fonction de l'altitude (4 % tous les 300 m) :



II - Mise en œuvre du BMP180 avec un ESP32

Connexion du **BMP180** à l'**ESP32** :

BMP180	ESP32 Pin
3.3V	3.3V
GND	GND
SDA	GPIO21
SCL	GPIO22

Exemple de programme :

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP085.h>
#define seaLevelPressure_hPa 1013.25

Adafruit_BMP085 bmp;

void setup() {
```

```
Serial.begin(9600); // Initializes the Serial communication
delay(2000);

if (!bmp.begin()) {
  Serial.println("BMP180 Not Found. CHECK CIRCUIT!");
  while (1) {}
}

void loop() {
  Serial.print("Temperature = ");
  Serial.print(bmp.readTemperature());
  Serial.println(" *C");

  Serial.print("Pressure = ");
  Serial.print(bmp.readPressure());
  Serial.println(" Pa");

  Serial.print("Pressure at sealevel (calculated) = ");
  Serial.print(bmp.readSealevelPressure());
  Serial.println(" Pa");

  Serial.print("Altitude = ");
  Serial.print(bmp.readAltitude());
  Serial.println(" meters");

  Serial.print("Real altitude = ");
  Serial.print(bmp.readAltitude(seaLevelPressure_hPa * 100));
  Serial.println(" meters");

  Serial.println();
  delay(1000);
}
```

III - Mise en œuvre du BME680 avec un Raspberry Pi

Voir ressource sur le **bus I2C**.