BMP180, mesure de température et pression atmosphérique avec un Arduino ou ESP8266

### Cadre : Co-intervention MATH-ATELIER SN. Lieu : Salle de TP E24 et E25 SN et MULTIMEDIA

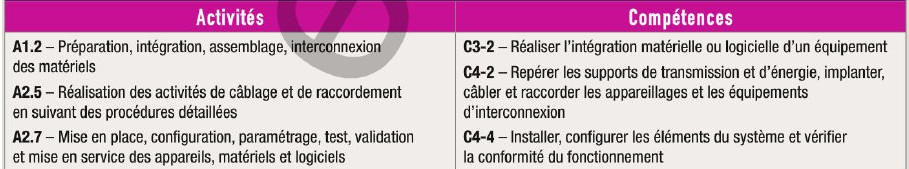
**Progression : suivant l’étape de la progression (sur ATRIUM) : Séance n°4 Ressources :**

### Logiciels (Fritzing, Arduino, documentation technique et langage C Arduino)



### Matériels : Un Arduino UNO avec ses câbles et ses divers composants associés (plaquette lab, résistances, leds, et les divers capteurs de la station météo)

**Compétences visées :**



### Sujet : Etude du capteur de pression Objectif

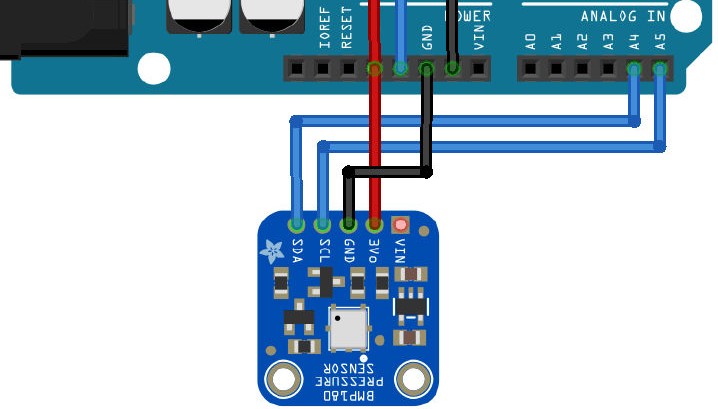
Etudier le capteur de pression BP180 avec l’ESP8266 ou l’Arduino uno

### Référentiel d’activité professionnelle (RAP) SN :

* Les capteurs en 1ere SN (taxonomie 3)
* Les logiciels (taxonomie 3)
* Langages de programmation
* Les micro-ordinateurs (taxonomie 3)
* Les organigrammes

### Math :

* Calcul de fréquence
* Colorimétrie et spectre de la lumière
* Loi d’Ohm
* Analyse de données Excel
* Schéma de câblage du capteur :



* BMP180

### Table des matières

* Branchement du BMP180 sur un Arduino
* Caractéristiques techniques du BMP180 :
* Recommandations importantes
* 3 méthodes pour lire la température et la pression atmosphérique sur le BMP180
* Pour aller plus loin

### Le BMP180 est un capteur qui permet de mesurer la pression atmosphérique et la température à l’aide de votre Arduino ou de votre Raspberry Pi très simplement. Le BMP180 est la nouvelle génération du capteur Bosch BMP085. Aucune modification n’a été apportée au niveau de 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 faudra faire une petite conversion d’unité pour présenter la valeur mesurée ou utiliser les fonctions de conversion disponibles dans certaines librairies. Vous pouvez acheter le BMP180 pour moins de 4€.

# Branchement du BMP180 sur un Arduino

Le plus facile est de se procurer le BMP180 déjà soudé sur un PCB avec une interface I2C. Dans la version I2C, le câblage du BMP180 est très simple. Le tableau suivant récapitule le câblage en fonction de quelques cartes Arduino.

|  |  |  |
| --- | --- | --- |
| Broche du BMP180 | Fonction de la broche | Connection Arduino |
| DA (ou SDA) | Donnée I2C | N’importe quel Pin identifié SDA ou Uno, Redboard, Pro / Pro mini A4 Mega, Due 20  Leonardo, Pro Micro 2  ESP8266 D2 , GPIO-4 |
| CL (ou SCL) | Horloge I2C | N’importe quel Pin identifié SCL ou Uno, Redboard, Pro / Pro mini A5 mega, Due 21  Leonardo, Pro Micro 3  ESP8266 D1, GPIO-5 |
| – ou GND | Masse | GND |
| + ou VCC | 3,3 Volts | 3.3V |

# Caractéristiques techniques du BMP180 :

 

* Dimension du PCB : 10 x 12 x 2 mm (11mm de hauteur avec le connecteur droit

soudé)

* Alimentation : de 3 à 5 Volts
* Faible consommation : 5 µA pour 1 mesure par seconde
* Plage de mesure de la pression atmosphérique : de 300-1100 hPa (jusqu’à 9000m au dessus de niveau de la mer)
* Précision de mesure : 0,03hPa – 0,25m d’altitude
* Fonctionnement : de -40°C à +85°C
* La fiche technique complète est disponible ici

# Recommandations importantes

Le BMP180 doit être au contact de l’air ambiant pour réaliser les mesures. Si vous devez intégrer le capteur dans un boitier, n’oubliez pas de prévoir des trous pour réaliser une circulation d’air.Voici quelques recommandations importantes pour réaliser des mesures correctes et protéger le BMP180.

* N’exposez pas excessivement le BMP180 dans le flux d’air d’un ventilateur sous peine d’avoir des mesures erronées ou très fluctuantes.
* La mesure de la pression atmosphérique dépend de la température. Evitez de placer le BMP180 devant une source de chaleur, encore moins devant une source produisant des changements rapides (chauffage, fenêtre en plein soleil…).
* Le BMP180 est sensible à l’humidité et n’est pas prévu pour un contact direct avec de l’eau.
* De même il est sensible à la lumière. Il faudra le protéger au maximum de la lumière ambiante. Ne placez pas le capteur devant le trou de ventilation de votre boîtier par exemple.
* Le BMP180 accepte une tension d’alimentation comprise entre 1,8 et 3,6 Volts. Le mieux est d’utiliser la sortie 3,3V de votre Arduino sans jamais dépasser 3,6V (d’après Sparkfun).

# 3 méthodes pour lire la température et la pression atmosphérique sur le BMP180

## Sans utilisation de librairie externe

Créez un nouveau projet et collez le code ci-dessous initialement développé par Leo Nutz. Il n’utilise aucune librairie externe pour communiquer et réaliser les conversions de mesures. Moins pratique mais aussi un peu plus compact qu’une librairie externe ce qui peut s’avérer très utile dans un projet Arduino.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Capteur de pression atmosphérique et de température BMP180 /BMP085

Programme de test pour Arduino sans librairie externe [www.projetsdiy.fr](http://www.projetsdiy.fr/)

Version originale de Leo Nutz, [www.ALTDuino.de](http://www.ALTDuino.de/)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ #include <Wire.h>

#define ADDRESS\_SENSOR 0x77 // Addresse du capteur

int16\_t ac1, ac2, ac3, b1, b2, mb, mc, md; // Store sensor PROM values from BMP180

uint16\_t ac4, ac5, ac6; // Store sensor PROM values from BMP180

// Ultra Low Power OSS = 0, OSD = 5ms

// Standard OSS = 1, OSD = 8ms

// High OSS = 2, OSD = 14ms

// Ultra High Resolution OSS = 3, OSD = 26ms

const uint8\_t oss = 3; // Set oversampling setting

const uint8\_t osd = 26; // with corresponding oversampling delay

float T, P; // Variables globales pour la température et la pression

void setup()

{

Serial.begin(9600);

while(!Serial){;} // On attend que le port série soit disponible

delay(5000);

Wire.begin(); // Active le bus I2C init\_SENSOR(); // Initialise les variables

delay(100);

}

void loop()

{

int32\_t b5;

b5 = temperature(); // Lit et calcule la température (T)

Serial.print("Temperature: "); Serial.print(T, 2); Serial.print("\*C, ");

P = pressure(b5); // Lit et calcule la pressure (P)

Serial.print("Pression: "); Serial.print(P, 2); Serial.print(" mbar, ");

Serial.print(P \* 0.75006375541921, 2); Serial.println(" mmHg"); Serial.println("");

delay(1000); // Delai entre chaque mesure

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Initialise les variables du capteur

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ void init\_SENSOR()

{

ac1 = read\_2\_bytes(0xAA); ac2 = read\_2\_bytes(0xAC); ac3 = read\_2\_bytes(0xAE); ac4 = read\_2\_bytes(0xB0); ac5 = read\_2\_bytes(0xB2); ac6 = read\_2\_bytes(0xB4); b1 = read\_2\_bytes(0xB6);

b2 = read\_2\_bytes(0xB8); mb = read\_2\_bytes(0xBA); mc = read\_2\_bytes(0xBC); md = read\_2\_bytes(0xBE);

Serial.println("");

Serial.println("Données de calibration du capteur :"); Serial.print(F("AC1 = ")); Serial.println(ac1); Serial.print(F("AC2 = ")); Serial.println(ac2); Serial.print(F("AC3 = ")); Serial.println(ac3); Serial.print(F("AC4 = ")); Serial.println(ac4); Serial.print(F("AC5 = ")); Serial.println(ac5); Serial.print(F("AC6 = ")); Serial.println(ac6); Serial.print(F("B1 = ")); Serial.println(b1); Serial.print(F("B2 = ")); Serial.println(b2); Serial.print(F("MB = ")); Serial.println(mb); Serial.print(F("MC = ")); Serial.println(mc); Serial.print(F("MD = ")); Serial.println(md); Serial.println("");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Calcul de la pressure

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ float pressure(int32\_t b5)

{

int32\_t x1, x2, x3, b3, b6, p, UP; uint32\_t b4, b7;

UP = read\_pressure(); // Lecture de la pression renvoyée par le capteur

b6 = b5 - 4000;

x1 = (b2 \* (b6 \* b6 >> 12)) >> 11;

x2 = ac2 \* b6 >> 11; x3 = x1 + x2;

b3 = (((ac1 \* 4 + x3) << oss) + 2) >> 2;

x1 = ac3 \* b6 >> 13;

x2 = (b1 \* (b6 \* b6 >> 12)) >> 16; x3 = ((x1 + x2) + 2) >> 2;

b4 = (ac4 \* (uint32\_t)(x3 + 32768)) >> 15; b7 = ((uint32\_t)UP - b3) \* (50000 >> oss);

if(b7 < 0x80000000) { p = (b7 << 1) / b4; } else { p = (b7 / b4) << 1; }

// ou p = b7 < 0x80000000 ? (b7 \* 2) / b4 : (b7 / b4) \* 2; x1 = (p >> 8) \* (p >> 8);

x1 = (x1 \* 3038) >> 16; x2 = (-7357 \* p) >> 16;

return (p + ((x1 + x2 + 3791) >> 4)) / 100.0f; // Retourne la pression en mbar

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Lecture de la température (non compensée)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ int32\_t temperature()

{

int32\_t x1, x2, b5, UT;

Wire.beginTransmission(ADDRESS\_SENSOR); // Début de transmission avec l'Arduino

Wire.write(0xf4); // Envoi l'adresse de registre

Wire.write(0x2e); // Ecrit la donnée

Wire.endTransmission(); // Fin de transmission delay(5);

UT = read\_2\_bytes(0xf6); // Lecture de la valeur de la TEMPERATURE

// Calcule la vrai température

x1 = (UT - (int32\_t)ac6) \* (int32\_t)ac5 >> 15; x2 = ((int32\_t)mc << 11) / (x1 + (int32\_t)md); b5 = x1 + x2;

T = (b5 + 8) >> 4;

T = T / 10.0; // Retourne la température in celsius

return b5;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Lecture de la pression

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ int32\_t read\_pressure()

{

int32\_t value;

Wire.beginTransmission(ADDRESS\_SENSOR); // Début de transmission avec l'Arduino

Wire.write(0xf4); // Envoi l'adresse de registre Wire.write(0x34 + (oss << 6)); // Ecrit la donnée Wire.endTransmission(); // Fin de transmission delay(osd);

Wire.beginTransmission(ADDRESS\_SENSOR); Wire.write(0xf6); Wire.endTransmission(); Wire.requestFrom(ADDRESS\_SENSOR, 3);

if(Wire.available() >= 3)

{

value = (((int32\_t)Wire.read() << 16) | ((int32\_t)Wire.read() << 8) | ((int32\_t)Wire.read())) >> (8 - oss);

}

return value; // Renvoie la valeur

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Lecture d'un byte sur la capteur BMP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ uint8\_t read\_1\_byte(uint8\_t code)

{

uint8\_t value; Wire.beginTransmission(ADDRESS\_SENSOR); Wire.write(code); Wire.endTransmission(); Wire.requestFrom(ADDRESS\_SENSOR, 1);

if(Wire.available() >= 1)

{

value = Wire.read();

}

return value;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Lecture de 2 bytes sur la capteur BMP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

uint16\_t read\_2\_bytes(uint8\_t code)

{

uint16\_t value; Wire.beginTransmission(ADDRESS\_SENSOR); Wire.write(code); Wire.endTransmission(); Wire.requestFrom(ADDRESS\_SENSOR, 2);

if(Wire.available() >= 2)

{

value = (Wire.read() << 8) | Wire.read(); // Récupère 2 bytes de données

}

return value; // Renvoie la valeur

}

## Avec la librairie Adafruit\_BMP085

La librairie d’Adafruit (Adafruit\_BMP085) n’a pas été mise à jour. Elle est toujours basée sur le BMP085. Contrairement à la librairie de Sparkfun, l’estimation de l’altitude est (presque) correcte.

#include <Wire.h>

#include <Adafruit\_BMP085.h>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This is an example for the BMP085 Barometric Pressure & Temp Sensor

Designed specifically to work with the Adafruit BMP085 Breakout

----> https://[www.adafruit.com/products/391](http://www.adafruit.com/products/391)

These displays use I2C to communicate, 2 pins are required to interface

Adafruit invests time and resources providing this open source code, please support Adafruit and open-source hardware by purchasing products from Adafruit!

Written by Limor Fried/Ladyada for Adafruit Industries.

BSD license, all text above must be included in any redistribution

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Connect VCC of the BMP085 sensor to 3.3V (NOT 5.0V!)

// Connect GND to Ground

// Connect SCL to i2c clock - on '168/'328 Arduino Uno/Duemilanove/etc thats Analog 5

// Connect SDA to i2c data - on '168/'328 Arduino Uno/Duemilanove/etc thats Analog 4

// EOC is not used, it signifies an end of conversion

// XCLR is a reset pin, also not used here Adafruit\_BMP085 bmp;

void setup() { Serial.begin(9600); if (!bmp.begin()) {

Serial.println("Could not find a valid BMP085 sensor, check wiring!"); while (1) {}

}

}

void loop() {

Serial.print("Temperature = "); Serial.print(bmp.readTemperature()); Serial.println(" \*C");

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

// Calculate altitude assuming 'standard' barometric

// pressure of 1013.25 millibar = 101325 Pascal Serial.print("Altitude = "); Serial.print(bmp.readAltitude()); Serial.println(" meters");

// you can get a more precise measurement of altitude

// if you know the current sea level pressure which will

// vary with weather and such. If it is 1015 millibars

// that is equal to 101500 Pascals. Serial.print("Real altitude = "); Serial.print(bmp.readAltitude(101500)); Serial.println(" meters");

Serial.println(); delay(500);

}

## Avec la librairie Sparfun

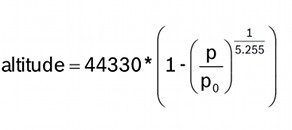
La librairie Sparkfun est disponible ici. Deux exemples sont livrés dont l’un intègre une estimation de l’altitude d’après la formule décrite plus bas.

**Pour aller plus loin**

Voici un petit aide mémoire extrait de wikipedia pour les conversions d’unités

* 1 hPa = 100 Pa = 1 mbar = 0.001 bar
* 1 hPa = 0.75006168 Torr
* 1 hPa = 0.01450377 psi (mesure impériale, pounds per square inch)
* 1 hPa =0.75006375541921 mmHg (millimètres de mercure)
* 1 hpa = 0.00098692 atm (atmosphère)

Connaissant la pression atmosphérique, il est possible d’en déterminer approximativement l’altitude grâce à cette relation (certaines librairies intègrent cette fonction).



**Quoi faire de plus avec la pression atmosphérique ?**

Maintenant que nous pouvons mesurer la pression atmosphérique, nous pouvons fabriquer notre propre baromètre numérique et ainsi prévoir notre météo locale. Pour cela, nous pouvons nous appuyer sur les tables de prévision de l’abbé Moreux. Ecrites dans les années 30, elles sont plus précisent qu’un simple baromètre car elles tiennent compte de la saison et de la direction du vent.

**Déterminer la météo en mesurant la pression atmosphérique au printemps**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Direction du vent / Baromètre** | **Nord** | **Est / Nord-Est** | **Sud / Sud-Est** | **Sud-Ouest** | **Ouest / Nord-Ouest** |
| > 1020 hPa | Beau ou assez beau. Journées chaudes, nuits fraîches; gelées matinales. | Beau ou assez beau. Journées douces ou assez chaudes, nuits fraîches; gelées possibles. | Beau ou assez beau. Journées chaudes, nuits fraîches. | Beau ou assez beau. | Beau.  Assez chaud dans la journée, frais la nuit; gelées possibles au petit matin. |
| De 1013 à  1020 hPa | Ondées ou giboulées. Températures fraîches | Giboulées. Journées fraîches, nuits froides; gelées à craindre. | Assez beau ou ondées orageuses.  Journées chaudes, nuits un peu fraîches. | Ondées ou averses.  Températures douces. | Nuageux avec ondées possibles; giboulées en montagne.  Temps frais. |
| De 1006 à  1013 hPa | Ondées ou giboulées. Températures fraîches | Ondées ou giboulées avec vent. Temps frais. | Pluie ou averses avec un peu de vent.  Temps doux. | Pluie et vent assez fort.  Temps doux. | Ondées en plaine, giboulées en montagne.  Temps frais. |
| < 1006 hPa | Pluie ou neige avec vent.  Températures basses. | Ondées, giboulées ou averses orageuses; neige en montagne.  Vent faible ou modéré.  Températures basses. | Pluie et vent assez fort.  Températures douces. | Pluie et vent assez fort.  Températures douces. | Pluie en plaine, neige en montagne.  Temps frais. |

**Déterminer la météo en mesurant la pression atmosphérique en été**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Direction du vent / Baromètre** | **Nord** | **Est / Nord-Est** | **Sud / Sud- Est** | **Sud-Ouest** | **Ouest / Nord- Ouest** |
| > 1020 hPa | Beau ou assez beau. Journées chaudes, nuits fraîches. | Beau. Journées chaudes, nuits fraîches. | Beau, mais orages possibles. Très chaud le jour, chaud la nuit. | Beau ou très beau. Chaud ou très chaud. | Beau. Chaud dans la journée, assez frais la nuit. |
| De 1013 à  1020 hPa | Assez beau, ondées possibles.  Assez chaud. | Beau ou assez beau avec parfois des averses orageuses.  Chaud ou assez chaud le jour, nuits fraîches | Beau, avec possibilité d’averses orageuses. Assez chaud. | Beau, ondées orageuses possibles. Assez chaud. | Assez beau, mais ondées possibles en montagne.  Températures douces. |
| De 1006 à  1013 hPa | Ondées ou averses orageuses. Températures douces. | Pluies orageuses. Températures douces. | Temps lourd et orages avec averses. Chaud. | Temps lourd et orages possibles.  Assez chaud. | Ondées et averses avec un peu de vent.  Températures douces. |
| < 1006 hPa | Pluie et vent. Températures douces. | Pluies orageuses avec un peu de vent. Temps lourd et humide. | Averses orageuses et vents violents.  Chaud et humide. | Orages et averses orageuses. Assez chaud. | Pluie à tendance orageuse avec vent. Doux et humide. |

### Déterminer la météo en mesurant la pression atmosphérique en automne

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Direction du vent / Baromètre** | **Nord** | **Est / Nord- Est** | **Sud / Sud-Est** | **Sud-Ouest** | **Ouest / Nord- Ouest** |
| > 1020 hPa | Beau ou assez beau. Chaud le jour. Gelées à craindre la nuit. | Beau ou assez beau. Chaleur modérée. Gelées à craindre. | Beau ou assez beau. Assez chaud le jour, frais la nuit; gelées possibles. | Beau.  Chaud ou assez chaud le jour, nuits fraîches. | Beau ou modérément nuageux.  Chaleur modérée le jour, nuits fraîches; gelées à craindre. |
| De 1013 à  1020 hPa | Ondées locales. Températures fraîches. | Assez beau, avec possibilité | Nuages modérés; ondées et | Ondées locales et vents faibles. | Assez beau, avec ondées locales et giboulées en |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Direction du vent / Baromètre** | **Nord** | **Est / Nord- Est** | **Sud / Sud-Est** | **Sud-Ouest** | **Ouest / Nord- Ouest** |
|  |  | d’ondées. Frais le jour et gelées locales. | éclaircies. Assez chaud. | Températures douces. | montagne. Températures douces, parfois un peu fraîches. |
| De 1006 à  1013 hPa | Averses. Temps frais. | Ondées en plaine, giboulées en montagne.  Temps frais. | Nuageux, avec pluies éparses et vent modéré. Temps doux. | Pluie possible, vent modéré. Températures douces. | Ondées en plaine, giboulées en montagne; vent faible.  Temps frais. |
| < 1006 hPa | Averses de pluie ou de neige. Temps froid et températures en baisses. | Averses orageuses et vent modéré.  Temps frais. | Pluie et parfois averses orageuses avec vents forts.  Températures douces. | Pluie et vent fort. Temps doux et humide. | Fortes pluies et vent fort, bourrasques de neige en montagne.  Temps frais. |

**Déterminer la météo en mesurant la pression atmosphérique en hiver**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Direction du vent / Baromètre** | **Nord** | **Est / Nord- Est** | **Sud / Sud-Est** | **Sud-Ouest** | **Ouest / Nord- Ouest** |
| > 1020 hPa | Beau, avec tendance à la brume et au brouillard.  Températures fraîches. | Beau ou assez beau, un peu brumeux. Journées froides et gelées nocturnes. | Beau ou assez beau, parfois brumeux.  Températures modérées le jour, nuits froides; gelées possibles. | Beau ou assez beau, brumes. Températures douces ou assez douces le jour; gelées nocturnes. | Beau et un peu brumeux.  Températures modérées le jour; fortes gelées la nuit. |
| De 1013 à  1020 hPa | Assez beau. Giboulées en montagne.  Froid. | Temps brumeux; giboulées ou neige. Froid. | Assez beau à modérément nuageux, avec ondées possibles.  Températures douces. | Quelques nuages, ondées. Températures fraîches. | Modérément nuageux, giboulées en montagne.  Froid. |
| De 1006 à  1013 hPa | Neige ou giboulées. Froid. | Neige ou giboulées. Froid. | Pluie possible, avec vent fort. Températures douces. | Pluie ou neige avec vent.  Températures douces. | Giboulées ou neige.  Froid. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Direction du vent / Baromètre** | **Nord** | **Est / Nord- Est** | **Sud / Sud-Est** | **Sud-Ouest** | **Ouest / Nord- Ouest** |
| < 1006 hPa | Neige et vent parfois violent. Froid. | Giboulées, neige possible; vent modéré à assez fort. Froid. | Pluie ou neige fondue, avec vents forts.  Temps assez froid, parfois doux. | Pluie ou neige fondue, avec vent violent. Températures douces. | Pluie et bourrasques, ou neige.  Froid. |

## D’après l’article :

**https://projetsdiy.fr/bmp180-capteur-pression-temperature-barometre/ Sources de l’article :**

* learn.sparkfun.com
* Alertes Météo
* Adafruit