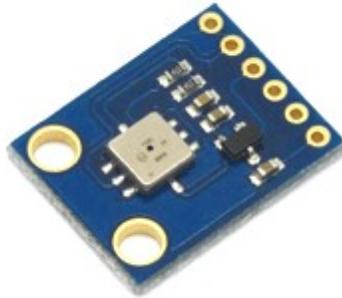


MEDIR PRESIÓN DEL AIRE Y ALTITUD CON ARDUINO Y BARÓMETRO BMP180 (Version 12-8-19)

Fuente: <https://www.luisllamas.es/medir-presion-del-aire-y-altitud-con-arduino-y-barometro-bmp180/>



ATENCIÓN; LAS PLACAS DEL SENSOR PUEDE TENER DISTINTAS PRESENTACIONES.

¿QUÉ ES UN BARÓMETRO BMP180?

De sus siglas BaroMetric Pressure

Un barómetro digital es un **dispositivo que mide la presión del aire y que puede usarse como altímetro**. Podemos conectar este sensor a un autómata o procesador como Arduino para registrar la medición de la presión del aire o estimar la altitud del sensor respecto al nivel del mar.

La presión barométrica está originada por el **peso de la columna del aire de la atmósfera**. La presión barométrica depende de diversos factores, especialmente de la temperatura, ya que esta influye en la densidad del aire y, por tanto, en el peso de la columna de aire. Otros factores que afectan a la medición son la humedad y el viento.

Dado que la presión barométrica depende de la altura de la columna de aire ubicada sobre el sensor también **puede ser empleada para estimar la altitud** a la que está ubicado en sensor respecto al nivel del mar.

Debemos tener en cuenta que la presión barométrica varía continuamente debido a las condiciones climatológicas por lo que **no proporciona una medición absoluta de la altitud** con precisión. Sin embargo, sí puede ser de utilidad en **mediciones diferenciales de altitud**, es decir, diferencias de cota en un desplazamiento vertical.

El BMP180 es un sensor barométrico digital, que incorpora un sensor de temperatura para compensar sus efectos en la medición de la presión barométrica. A su vez, **el BMP180 es una versión mejorada del BMP085**, siendo ambos modelos compatibles entre sí tanto en hardware como software.

El BMP180 es un sensor de **alta precisión y baja potencia**. El rango de medición es de 300hPa a 1110 hPa, equivalente a una altitud de -500m a 9000m sobre el nivel del mar. La precisión es configurable, desde 0.06hPa (0.5 metros) en el modo de bajo consumo, a 0.02hPa (0.17 metros) en el modo de alta precisión.

La comunicación se realiza a través del bus I2C, por lo que es sencillo obtener los datos medidos. La tensión de alimentación es de bajo voltaje entre 1.8 a 3.6V.

Frecuentemente se encuentran integrados en módulos como la GY-68 que **incorporan la electrónica necesaria para conectarla de forma sencilla a un Arduino**. En la mayoría de los módulos, este incluye un regulador de voltaje que permite alimentar directamente a 5V.

El consumo promedio es de 0.1µA en stand-by, y 650µA durante la medición, lo que supone un consumo promedio de 5µA tomando 1 muestra por segundo en precisión estándar. El tiempo de respuesta es de unos 5ms en resolución estándar, y 17ms en alta resolución

Los barómetros BMP180 y BMP085 **son ampliamente empleados en aplicaciones meteorológicas**, como estaciones registradoras, relojes que muestran el clima, o para predecir cuándo va a llover, entre otras. También pueden ser empleadas en aplicaciones de climatización o control de ventilación.

La capacidad para estimar alturas diferenciales también hace que sean usados en **vehículos aéreos como aviones o cuadricopteros**, o en aplicaciones que requieran medir la velocidad de desplazamiento vertical, por ejemplo, el ascenso de una máquina o el tiempo de caída de un dispositivo.

Otro resumen:

El sensor de presión barométrica BMP180 está diseñado para leer la presión atmosférica y de esta forma estimar indirectamente la Altura sobre el nivel del mar. La presión atmosférica es la fuerza que ejerce el aire (atmósfera) sobre la superficie de la tierra. La presión atmosférica se debe al peso de la columna de aire sobre determinada área, es por esta razón que al medir la presión atmosférica en puntos con mayor altitud, el valor de la presión es menor por ser menor la cantidad de aire. La presión atmosférica también varía con el clima, principalmente con la temperatura, pues esta hace cambiar la densidad del aire, que se ve reflejado en un cambio en el peso y por consiguiente en un cambio de presión.

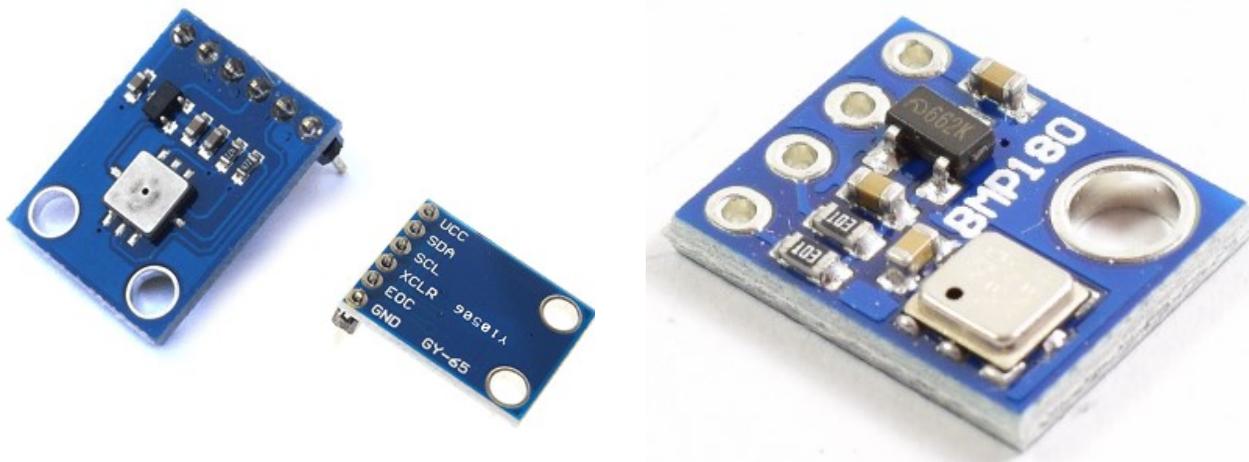
Entonces, la presión atmosférica varía con la temperatura y la altitud, estas dos variables son las más representativas para el cambio de presión. Factores como la humedad relativa y la velocidad del viento también influyen en la presión atmosférica en menor forma y pueden ser obviados.

Lo que mide el sensor BMP180 es la presión absoluta (Barométrica) y la temperatura, al sensar la temperatura podemos compensar su influencia en la presión y así determinar con mayor exactitud la altitud.

Nuestro módulo BMP180 incluye además del sensor BMP180, un regulador de voltaje (5V a 3.3V), resistencias pull-up y capacitores by-pass. El Módulo puede alimentarse directamente de la salida de 5V de Arduino. Posee un formato pequeño y de bajo consumo de corriente.

PRECIO

El precio de ambos sensores BMP085 y BMP180 es muy similar. Podemos encontrar cualquier de los dos modelos por 1.70€ en vendedores internacionales de eBay o AliExpress.



Dado que el sensor BMP180 es la versión mejorada de BMP085, que tienen el mismo precio, y que son totalmente compatibles en hardware y software, **lo normal es que siempre elegiremos el BMP180.**

ESQUEMA MONTAJE

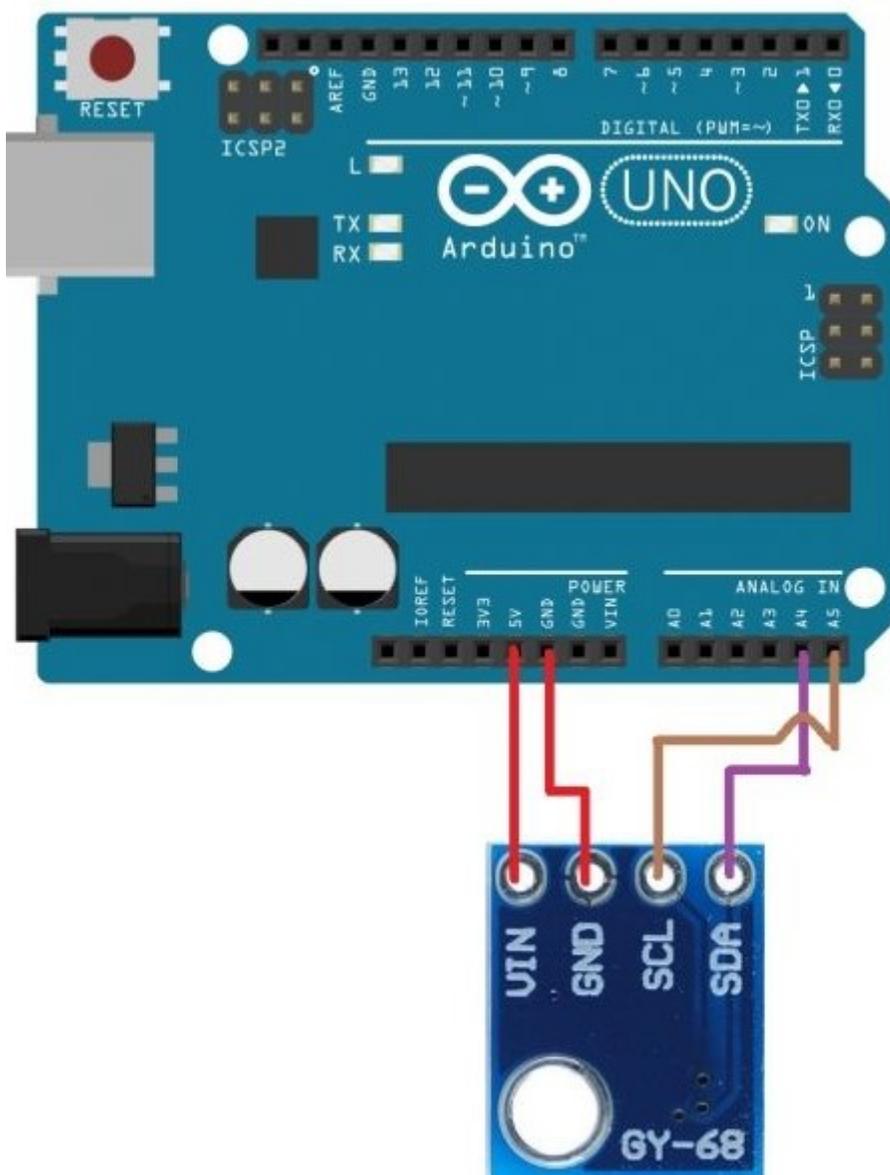
La conexión es sencilla, simplemente alimentamos el módulo desde Arduino mediante GND y 5V y conectamos el pin **SDA** y **SCL** de Arduino con los pines correspondientes del barómetro BMP180.

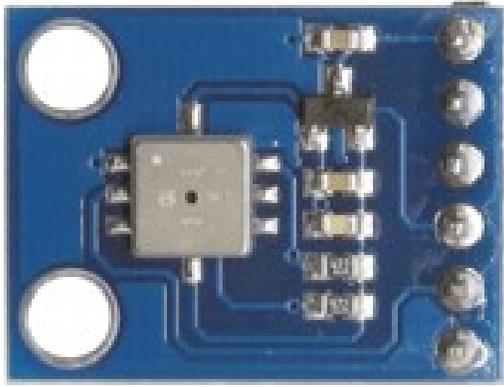
Conexiones entre Arduino y módulo BMP180

Las conexiones son como cualquier conexión I2C:

Conector	Arduino Uno, Nano, Mini.	Arduino Mega , DUE	Arduino Leonardo
VCC	5V	5V	5V
GND	GND	GND	GND
SCL	A5	21	3
SDA	A4	20	2

Si usamos Arduino Uno



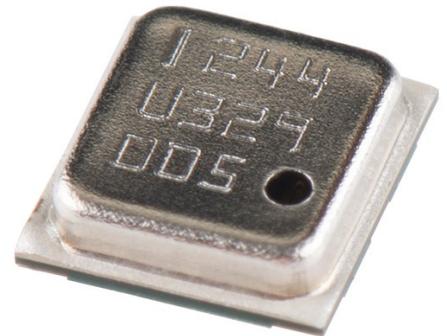


GND	—	GND
Vcc	—	5V
SDA	—	SDA (A4)
SCL	—	SCL (A5)

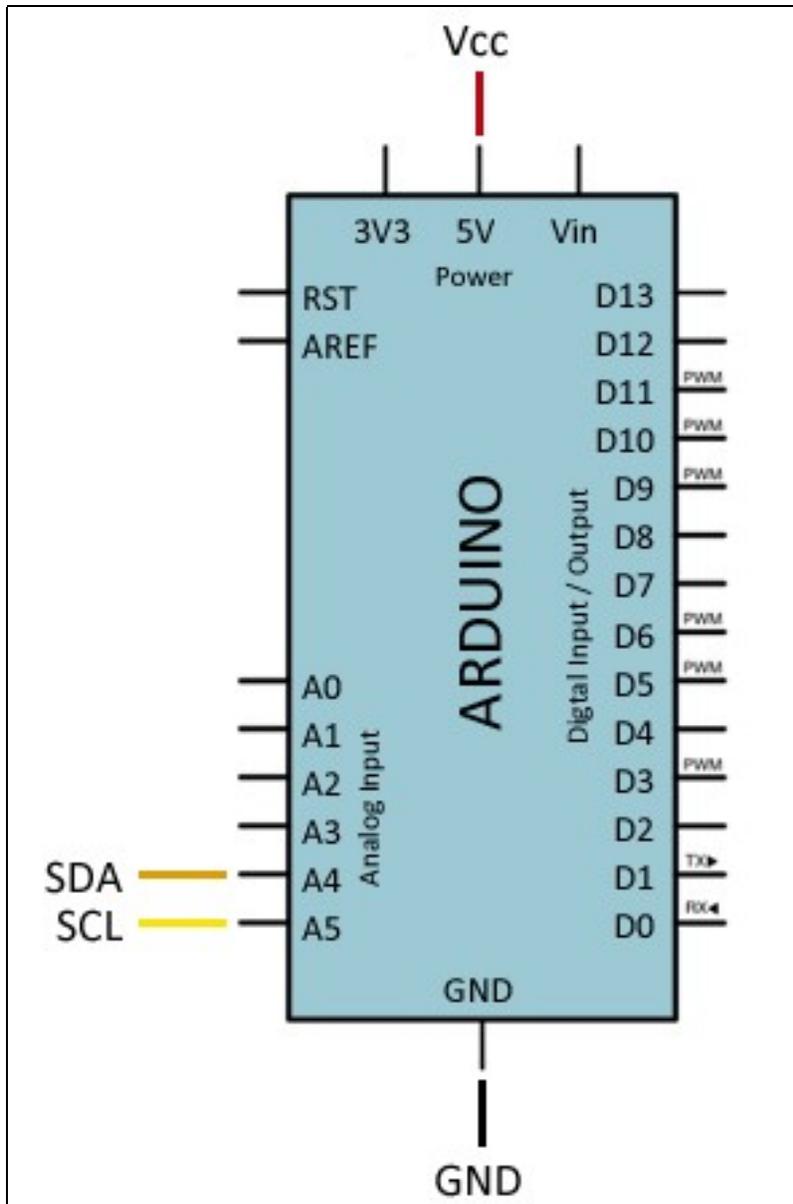
Verificar que la placa es compatible con 5V antes de conectarla a Arduino. Si no, tendremos que usar un adaptador de nivel lógico.

El sensor.

Cuidado con tapar el orificio



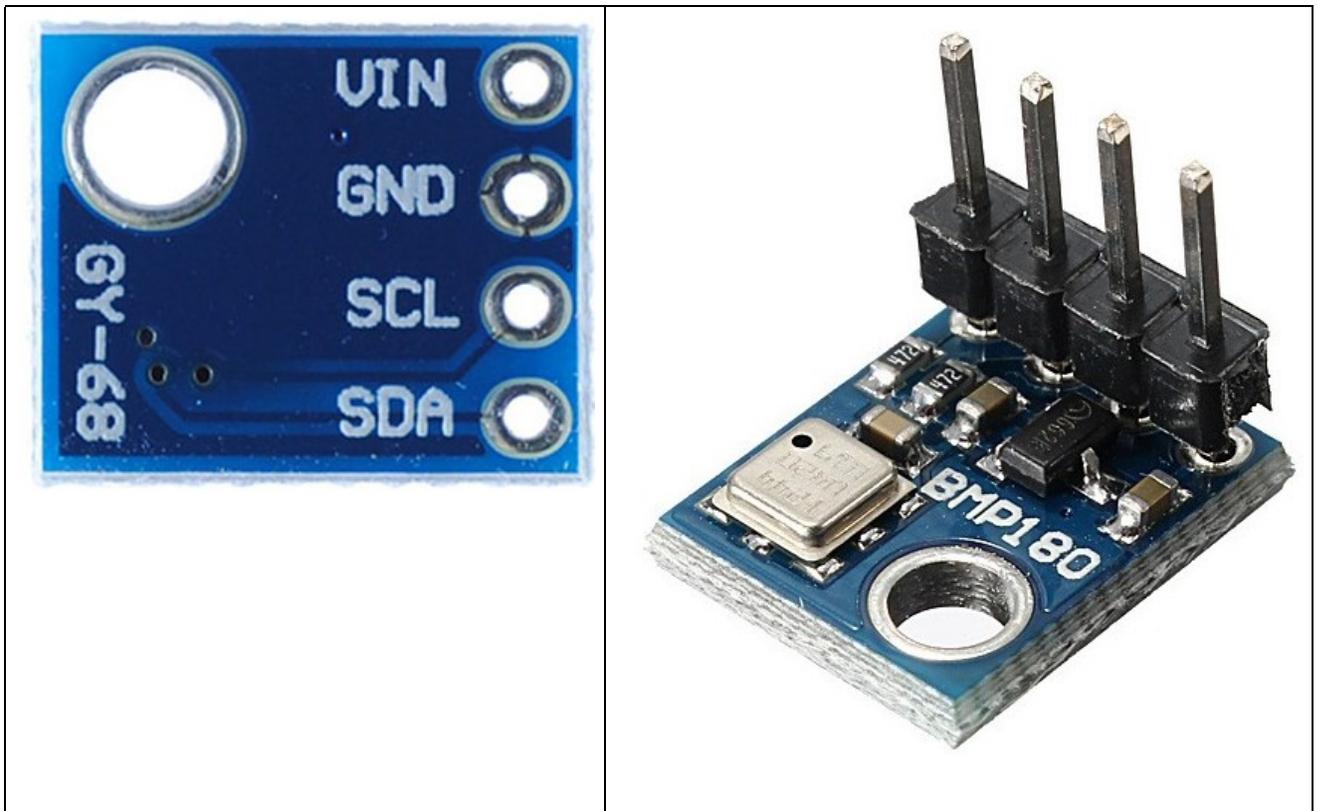
Mientras que la conexión vista desde el lado de Arduino quedaría así.



En Arduino Uno, Nano y Mini Pro, SDA es el pin A4 y el SCK el pin A5. Para otros modelos de Arduino consultar.

Verificar que vuestra placa es compatible con 5V antes de conectarla a Arduino. Si no, tendréis que usar un adaptador de nivel lógico.

El sensor puede tener otra distribución de pines, como el siguiente, donde VIN es VCC = 5v



EJEMPLOS DE CÓDIGO

Para realizar la lectura del BMP180 usaremos la librería:

SFE_BMP180-master.zip

La librería proporciona ejemplos de código, que resulta aconsejable revisar. Los siguientes ejemplos, por ejemplo, son modificaciones están basados a partir de los disponibles en la librería.

Es necesario descargar e importar la librería a nuestro IDE de Arduino, antes de empezar con los ejemplos explicaremos las funciones que utilizaremos:

begin()

Inicializa el sensor BMP180, nos retorna 1 si la inicialización es correcta o 0 si ha fallado

startTemperature()

Función para iniciar una medición de temperatura y nos retorna el tiempo en milisegundos que necesitamos esperar antes de obtener la lectura. Si nos retorna un 0, es porque ha fallado el inicio de la medición de temperatura

getTemperature(T)

Obtener la temperatura en la variable T, antes de usar esta función es necesario llamar a la función startTemperature() y que haya transcurrido el tiempo adecuado para la lectura; retorna 1 o 0 si la lectura se ha realizado con éxito o no respectivamente

startPressure(Sobremuestreo);

Función para iniciar una medición de presión, hay que indicar la cantidad de muestras adicionales (de 0 a 3) que el sensor debe tomar para la lectura de la presión y nos

retorna el tiempo en milisegundos que necesitamos esperar antes de obtener la lectura. Si nos retorna un 0, es porque ha fallado el inicio de la medición de presión

getPressure(P, T);

Obtener el valor de la medición iniciado previamente con startPressure(); es necesario darle como parámetro la temperatura T el cual servirá para compensar la influencia de la temperatura en el cálculo de la presión, el valor de la presión absoluta se guarda en la variable P. Retorna 1 o 0 si la lectura se ha realizado con éxito o no respectivamente

altitude(P, Po);

Calcula la altitud entre el punto donde se ha tomado la lectura de presión P (en mbar) con respecto a un punto de referencia con presión Po (en mbar). Nos retorna el valor de la altitud en metros

sealevel(P, A);

Esta función realiza el cálculo inverso a altitude(P, Po) , Dado una presión P (en mbar) y una altitud A (en metros) calcula la presión al nivel del mar o punto desde donde se mide la altura. Retorna el valor de la presión en mbar

OBTENER LOS VALORES DE PRESIÓN Y TEMPERATURA *(Enyado por Prof: DJB)*

El primer ejemplo, obtiene la medición de los valores en bruto (RAW) de presión y temperatura, y los muestra en la pantalla. Estos valores son de utilidad, por ejemplo, para hacer una estación meteorológica.

```
#include <SFE_BMP180.h>
#include <Wire.h>

SFE_BMP180 bmp180;

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

  if (bmp180.begin())
    Serial.println("BMP180 iniciado");
  else
  {
    Serial.println("Error al iniciar BMP180");
    while(1); // bucle infinito
  }
}

void loop()
{
  char status;
  double T,P;

  status = bmp180.startTemperature(); //Inicio de lectura de temperatura
  if (status != 0)
  {
    delay(status); //Pausa para que finalice la lectura
    status = bmp180.getTemperature(T); //Obtener la temperatura
    if (status != 0)
    {
```



```

if (bmp180.begin())
  Serial.println("BMP180 iniciado");
else
{
  Serial.println("Error al iniciar el BMP180");
  while(1);
}
}

void loop()
{
  char status;
  double T,P,A;

  status = bmp180.startTemperature(); //Inicio de lectura de temperatura
  if (status != 0)
  {
    delay(status); //Pausa para que finalice la lectura
    status = bmp180.getTemperature(T); //Obtener la temperatura
    if (status != 0)
    {
      status = bmp180.startPressure(3); //Inicio lectura de presión
      if (status != 0)
      {
        delay(status); //Pausa para que finalice la lectura
        status = bmp180.getPressure(P,T); //Obtener la presión
        if (status != 0)
        {
          Serial.print("Temperatura: ");
          Serial.print(T);
          Serial.print(" *C , ");
          Serial.print("Presion: ");
          Serial.print(P);
          Serial.print(" mb , ");

          A= bmp180.altitude(P,PresionNivelMar); //Calcular altura
          Serial.print("Altitud: ");
          Serial.print(A);
          Serial.println(" m");
        }
      }
    }
  }
  delay(1000);
}

```

ESTIMAR LA DIFERENCIA DE ALTURA ENTRE DOS PUNTOS

En el último ejemplo, realizamos la medición de la diferencia de altura entre dos puntos. Simplemente medimos la diferencia de cotas entre dos puntos de presión, pero en lugar de tomar la referencia a nivel del mar tomamos como referencia el primer punto.

```

#include <SFE_BMP180.h>
#include <Wire.h>

SFE_BMP180 bmp180;

double Po; //presion del punto inicial para h=0;
char status;
double T,P,A;

```

```

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

  if (bmp180.begin())
  {
    Serial.println("BMP180 iniciado");
    status = bmp180.startTemperature(); //Inicio de lectura de temperatura
    if (status != 0)
    {
      delay(status); //Pausa para que finalice la lectura
      status = bmp180.getTemperature(T); //Obtener la temperatura
      if (status != 0)
      {
        status = bmp180.startPressure(3); //Inicio lectura de presión
        if (status != 0)
        {
          delay(status); //Pausa para que finalice la lectura
          status = bmp180.getPressure(P,T); //Obtener la presión
          if (status != 0)
          {
            Po=P; //Asignamos el valor de presión como punto de referencia
            Serial.println("Punto de referencia establecido: h=0");
          }
        }
      }
    }
  }
  else
  {
    Serial.println("Error al iniciar el BMP180");
    while(1);
  }
}

void loop()
{
  status = bmp180.startTemperature(); //Inicio de lectura de temperatura
  if (status != 0)
  {
    delay(status); //Pausa para que finalice la lectura
    status = bmp180.getTemperature(T); //Obtener la temperatura
    if (status != 0)
    {
      status = bmp180.startPressure(3); //Inicio lectura de presión
      if (status != 0)
      {
        delay(status); //Pausa para que finalice la lectura
        status = bmp180.getPressure(P,T); //Obtener la presión
        if (status != 0)
        {
          A= bmp180.altitude(P,Po); //Calcular altura con respecto al punto de referencia
          Serial.print("h=");
          Serial.print(A);
          Serial.println(" m");
        }
      }
    }
  }
  delay(1000);
}

```