

# Sensores y Actuadores

## TST-2022

### **Profesores:**

ING.JORGE E. MORALES

TEC. GONZALO VERA

**AÑO: 2022**



### 1.e) Como implementaría un sensor inteligente de altitud?

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 sensor la temperatura podemos compensar su influencia en la presión y así determinar con mayor exactitud la altitud.

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



### Librería para el sensor de presión BMP180

Se puede descargar del siguiente enlace: [https://github.com/sparkfun/BMP180\\_Breakout/...](https://github.com/sparkfun/BMP180_Breakout/...)

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

Alumno: Fernando Vexenat

Grupo: 01

Materia: Sensores y Actuadores -TST-2022



### **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 altitud (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

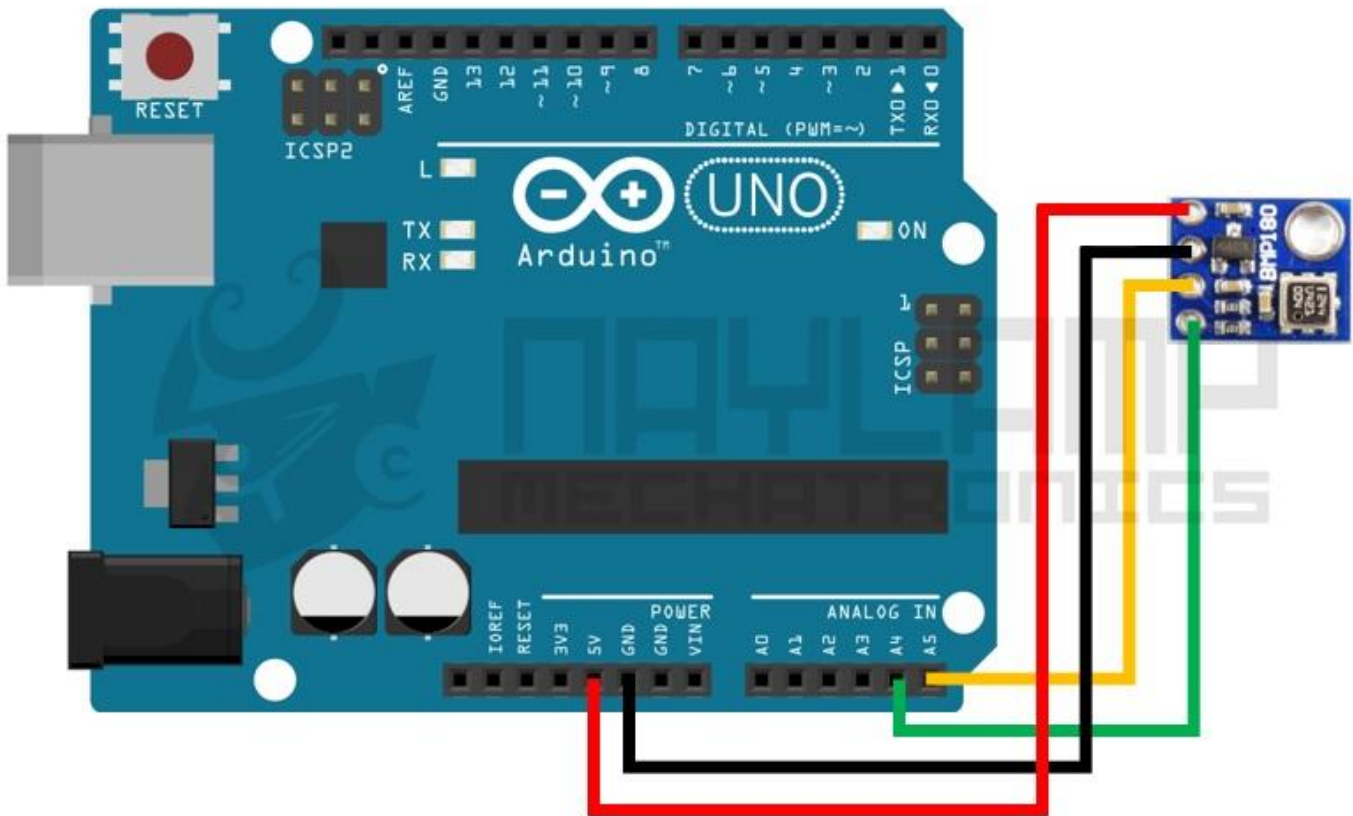
Explicado esto procedemos a realizar nuestros ejemplos:

## Conexiones entre Arduino y módulo BMP180

Las conexiones son como cualquier conexión I2C:

Adaptador LCD a I2C	Arduino Uno, Nano, Mini.	Arduino Mega , DUE
VCC	5V	5V
GND	GND	GND
SCL	A5	21
SDA	A4	20





## 1. Realizar lecturas de presión y temperatura con el BMP180

Realizar esta tarea es sumamente sencillo, basta con iniciar la lectura, esperar el tiempo que dura la lectura y obtener dicho valor. Realizaremos el mismo procedimiento tanto para temperatura y presión.

A continuación se muestra el Sketch correspondiente:

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

```
SFE_BMP180 bmp180;
```

```
void setup()
```



```
{
  Serial.begin(9600);

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

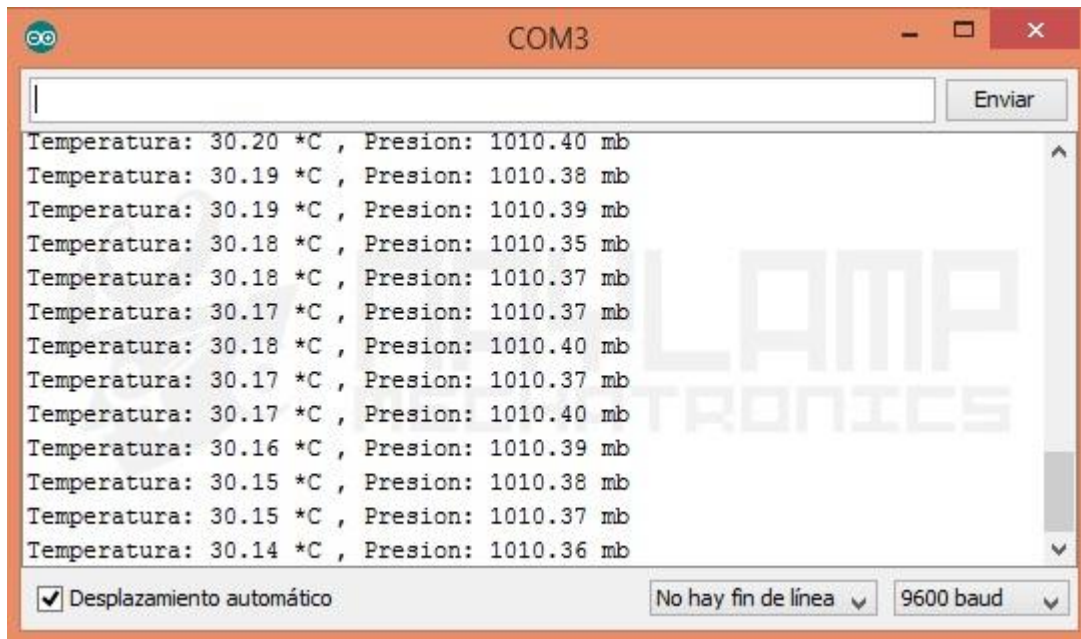
void loop()
{
  char status;
  double T, P;

  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)
    {
      status = bmp180.startPressure(3); // Inicio lectura de presión
      if (status != 0)
      {
        delay(status); // Pausa para que finalice la lectura
        status = bmp180.getPressure(P, T); // Obtenemos la presión
        if (status != 0)
        {
          Serial.print("Temperatura: ");
          Serial.print(T, 2);
          Serial.print(" *C , ");
          Serial.print("Presion: ");
          Serial.print(P, 2);
          Serial.println(" mb");
        }
      }
    }
  }
  delay(1000);
}
```





El resultado que obtenemos por el monitor serial es el siguiente:



## 2. Estimar la Altitud con el BMP180

En este caso calcularemos la altitud de nuestra posición sobre el nivel del mar, para esto es necesario medir tanto la presión como la temperatura de nuestra posición, además como dato necesitamos la presión atmosférica sobre el nivel del mar.

EL sketch para obtener la altura es el siguiente:

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

SFE_BMP180 bmp180;

doublePresionNivelMar=1013.25; //presión sobre el nivel del mar en mbar

voidsetup()
{
  Serial.begin(9600);

  if (bmp180.begin())
```



```

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

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

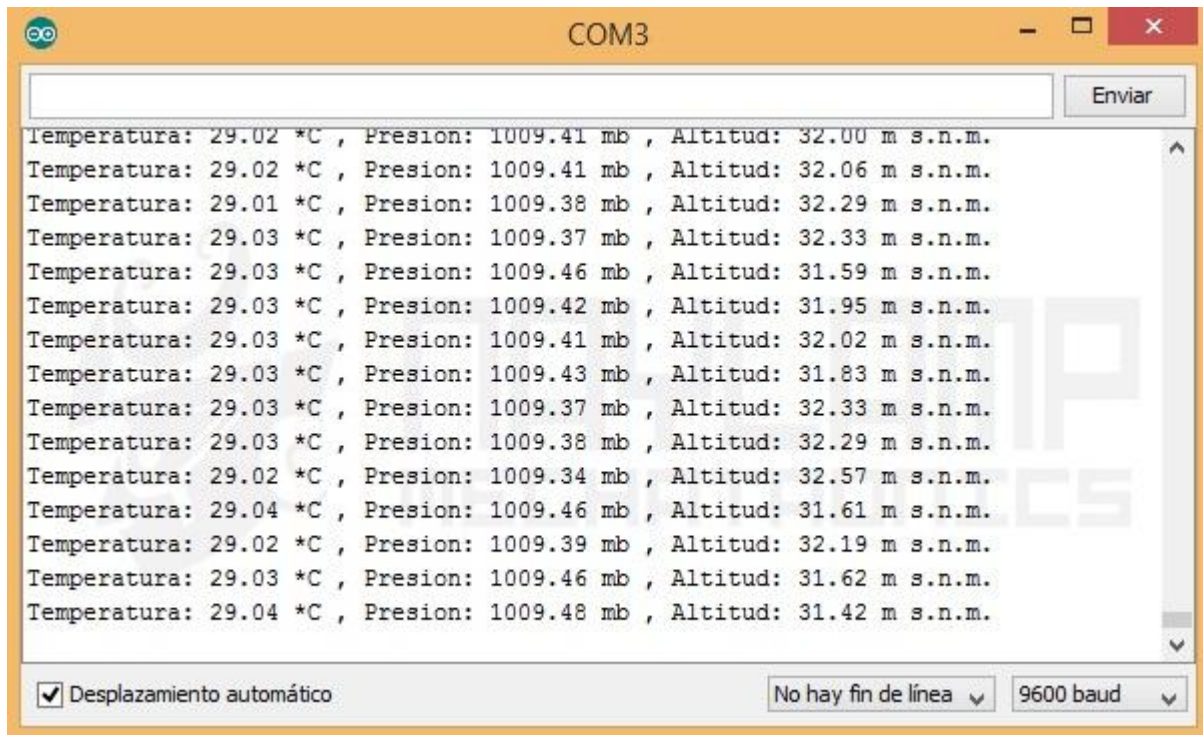
    status = bmp180.startTemperature(); // Inicio de lectura de temperatura
    if (status != 0)
    {
        delay(status); // Pausa para que finalice la lectura
        status = bmp180.getTemperature(T); // Obtener la temperatura
        if (status != 0)
        {
            status = bmp180.startPressure(3); // Inicio lectura de presión
            if (status != 0)
            {
                delay(status); // Pausa para que finalice la lectura
                status = bmp180.getPressure(P, T); // Obtenemos la presión
                if (status != 0)
                {
                    Serial.print("Temperatura: ");
                    Serial.print(T);
                    Serial.print(" *C , ");
                    Serial.print("Presión: ");
                    Serial.print(P);
                    Serial.print(" mb , ");
                    //-----Calculamos la altitud-----
                    A = bmp180.altitude(P, PresionNivelMar);
                    Serial.print("Altitud: ");
                    Serial.print(A);
                    Serial.println(" m s.n.m.");
                }
            }
        }
    }
    delay(1000);
}

```





A continuación se muestra los datos recibidos por el monitor serial.



### 3. Calculando la altura entre dos puntos

Para este caso tomaremos un punto inicial, para nuestra altura  $h=0$ , y a partir de aquí, conforme nos desplazamos verticalmente mediremos la altura, en este caso la precisión es de aproximadamente 0,5m. Para esto necesitamos medir tanta presión y temperatura en el punto de la posición inicial, como en los demás posiciones:

EL sketch para realizar esto es:

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

SFE_BMP180 bmp180;

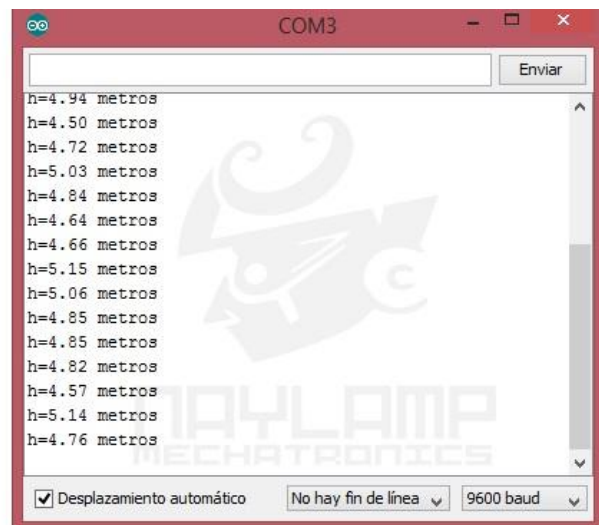
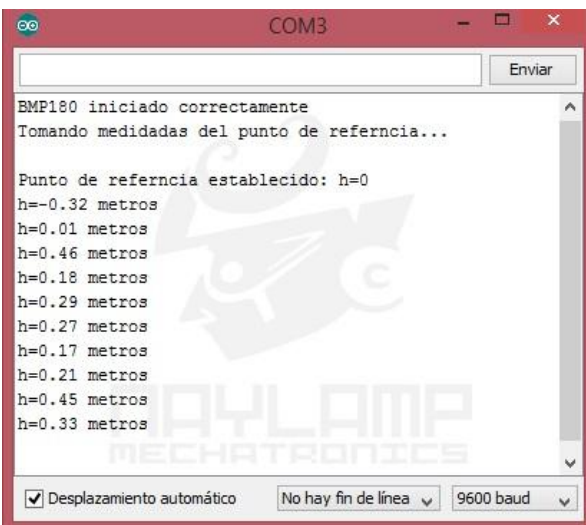
double Po; //presión del punto inicial para h=0;
charstatus;
doubleT,P,A;
```





```
status = bmp180.startPressure(3);//Inicio lectura de presión
if (status != 0)
{
    delay(status);//Pausa para que finalice la lectura
    status = bmp180.getPressure(P,T);//Obtenemos la presión
    if (status != 0)
    {
        //-----Calculamos la altura con respecto al punto de referencia-----
        A= bmp180.altitude(P,Po);
        Serial.print("h=");
        Serial.print(A);
        Serial.println(" metros");
    }
}
}
}
}
delay(1000);
}
```

A continuación mostramos la salida del monitor serial, en la primera imagen lecturas de nuestro punto de referencia y en la segunda imagen lecturas después de subir dos pisos.



En este caso nuestro punto de referencia es la posición inicial en donde encendemos o reiniciamos nuestro Arduino. Pero podemos trabajarlo como un punto fijo en caso sea necesario, para esto tenemos que tener como dato la presión de dicho punto.

