**Curso Proyecto Estación Meteorológica**

**1.- ¿Qué es Arduino?**

**1. ¿QUÉ ES ARDUINO?**

Arduino es una placa o tarjeta controladora, con una serie de entradas y salidas, y que se programa a través del ordenador mediante un lenguaje de programación.

Veamos qué elementos componen una ***Arduino UNO***:



**Alimentación:** Arduino puede estar alimentado por dos vías:

* Conexión **USB** (que proporciona 5 V).
* **Jack** de alimentación (que normalmente será una pila de 9 V o fuente de alimentación, que se recomienda que este entre 7 – 12 V).

Los **pines de alimentación** son para alimentar los circuitos la placa de prototipos o *breadboard* o *protoboard*:

* **3.3 V** proporciona una tensión de 3,3 V, y una intensidad máxima de 50 mA.
* **5 V** proporciona una tensión de 5 V, y una intensidad máxima de 300 mA.
* **GND** es la toma de tierra, o nivel 0 V de referencia.
* **Vin** proporciona la tensión máxima con la que está alimentado Arduino.

**Valores de entrada y de salida:** en función de cómo esté siendo utilizado en pin, tendremos:

* **Salida y entrada digital:** los valores de salida pueden ser o 0 V (**LOW**) o 5 V (**HIGH**), y se interpretara una entrada de entre 0 y 2 V como **LOW** y de entre 3 y 5 V como **HIGH**.
* **Salida analógica:** los valores de salida van desde 0 V a 5 V en un rango de

**0** a **255** (precisión de 8 bits) valores intermedios.

**Entrada analógica:** los valores de entrada van desde 0 V a 5 V en un rango de **0** a **1023** (precisión de 10 bits) valores intermedios.

La intensidad máxima de todos estos pines es de 40 mA.

Normalmente, todo el circuito electrónico que Arduino controlara se monta sobre una placa de prototipos o *breadboard*, y el conexionado se realiza con cables tipo *jumper* (es importante utilizar este tipo de cables porque no suelen romperse en los zócalos):

**2. PROGRAMANDO ARDUINO**



**2.-Programando Arduino**

Todo programa para Arduino presenta una **estructura básica**:

**1ª parte** int x=0; **Declarar** las **variables**.

**2ª parte** void **setup () {…} Configuración** de Arduino.

**3ª parte** void **loop () {…} Comandos** que regirán el

Comportamiento de Arduino.

**1ª** parte**: Declarar las variables**

intx**=**0;

Una variable es un valor que Arduino puede almacenar en su memoria, y que

Posteriormente podrá ser utilizado o modificado.

Los **tipos de variables** más utilizados son:

* int: almacena un **número entero** entre -32769 y 32767 (2 bytes).
* long: almacena un **número entero** muy largo, entre -2147483648 y

2147483647 (4 bytes).

* float: almacena un **número decimal** con un rango entre -3.4028235・1038 y 3.4028235・1038 (4 bytes).
* const: especifica que la variable definida no podrá ser cambiada durante el programa, siendo un siempre un **valor constante**:
* constfloat pi=3.1415;

Es importante saber que es posible declarar una variable sin asignarle un valor inicial, y hacerlo posteriormente durante el transcurso del programa:

Int x; //Declaración de variable

**…**

x=4; //Asignar valor a variable

**Dominio de una variable**: si declaro una variable al comienzo del programa, podre emplear dicha variable en cualquier momento (dentro de cualquier función o bloque de programa), pero si declaro una variable dentro de una función, solo se podrá utilizar en dicha función.

**Poner nombre a las variables**: Por último, una última consideración: a la hora deponer un nombre a una variable es recomendable utilizar alguna palabra que nos ayude a reconocer que se está almacenando en ella, y en caso de utilizar dos o más palabras se suele emplear la notación de **joroba de camello** (poner en mayúscula la primera letra de las siguientes palabras). Ejemplos son:

ledPin estadoAnterior cuentaPulsaciones

miVariable lecturaSensor ledPinAzul

**2ª** parte**: Configuración de Arduino**

void **setup() {…}**

En este bloque habrá que especificar:

* Que **pines** van a ser empleados como **entrada** y cuales como **salida**.

pinMode(2,OUTPUT); //utilizaré el pin 2 como salida Digital.

pinMode(3,OUTPUT); //utilizaré el pin 3 como salida Digital o Analógica.

pinMode(8,INPUT);//utilizaré el pin 10 como entrada Digital.

Las entradas analógicas no hacen falta incluirlas en el setup, puesto que esos pines (A0, A1, A2, A3, A4, A5) solo pueden ser entradas analógicas.

* Si vamos a querer establecer una **conexión** con el **ordenador**.

**Serial**.begin(9600); /\*hay que especificar los baudios (bits por

segundo) a la que va a realizarse dicha

comunicación Arduino-PC \*/

* Si vamos a querer utilizar **número aleatorios**.

randomSeed(0); //se inicia la generación de números aleatorios.

**3ª** parte**: Comandos que regirán el comportamiento de Arduino**

Void **loop () {…}**

En este bloque se deberá escribir todas aquellas instrucciones, ordenes, primitivas, comandos o funciones necesarias para que Arduino funcione según nuestro deseo.

Realmente, este bloque constituye un bucle infinito, ya que Arduino, mientras este alimentada con energía, funcionara haciendo el programa **loop** una y otra vez.

Iremos viendo cuales son estas funciones durante el desarrollo de estos apuntes.

**Manejar Librerías Arduino**

Las librerías son trozos de código hechas por terceros que usamos en nuestro sketch. Esto nos facilita mucho la programación y permite la abstracción haciendo que nuestro programa sea más sencillo de hacer y de entender. En este apartado también veremos cómo escribir o modificar librerías.

Disponemos de infinidad de librerías a nuestra disposición para facilitarnos el trabajo,

todas ellas son open source y disponemos de su código.

Las librerías normalmente incluyen los siguientes archivos comprimidos en un archivo ZIP o dentro de un directorio. Estas siempre contienen:

* Un archivo .cpp (código de C++)
* Un archivo .h o encabezado de C, que contiene las propiedades y métodos o funciones de la librería.
* Un archivo Keywords.txt, que contiene las palabras clave que se resaltan en el IDE (opcional).
* Muy posiblemente la librería incluye un archivo readme con información adicional de lo que hace y con instrucciones de cómo usarla.
* Directorio denominado examples con varios sketchs de ejemplo que nos ayudará a entender cómo usar la librería (opcional).

**Variables por medir**

**1.-**[**Presión Barométrica**](http://www.parro.com.ar/definicion-de-presi%F3n+barom%E9trica)**:** Presión ejercida por la atmósfera de la tierra en un punto dado, equivalente a la presión ejercida por una columna de mercurio.

**2.-Altitud**

Es la distancia vertical entre un punto situado sobre la superficie terrestre o

la atmósfera y el nivel medio del mar.

**3.-Humedad Relativa**

**La humedad relativa**(H.R.)**se define como el porcentaje (%) de vapor de agua que tiene una masa de aire en un momento determinado. Es la relación entre la humedad que tiene, y la máxima que podría tener a esa temperatura. Se expresa en tanto por cien, por ejemplo un 80% de vapor. Se calcula mediante una proporción:**

**Humedad relativa / 100 = gramos de agua que contiene / cantidad máxima de agua en gramos.**

Humedad relativa = humedad absoluta. 100 / humedad de saturación

La H.R. indica la capacidad del aire para absorber agua. Una humedad relativa baja indica que el aire aún puede absorber humedad. Por ejemplo, si la humedad relativa es del 70 %, significa que el aire aún puede absorber hasta un 30% más de vapor de agua.   
La H.R. nos da un índice de lo confortable que es el aire para nosotros ([sensación térmica](http://teleformacion.edu.aytolacoruna.es/AYC/document/atmosfera_y_clima/temperatura/sensacTermica0.htm)). Si es baja, el vapor de agua que exudan nuestros poros se evapora y nos sentimos bien. Si es alta, el sudor se vuelve pegajoso y queda sobre la piel.

Lo más interesante para el clima es conocer la humedad relativa: una masa de aire saturada, o cercana a la saturación (con alta humedad relativa) es una masa de aire húmeda y las plantas pueden aprovechar mejor el agua ya que a través de sus hojas se evapora en menor cantidad.

**Temperatura**

Es una [magnitud](https://es.wikipedia.org/wiki/Magnitud_f%C3%ADsica) referida a las nociones comunes de [calor](https://es.wikipedia.org/wiki/Calor) medible mediante un [termómetro](https://es.wikipedia.org/wiki/Term%C3%B3metro). En física, se define como una [magnitud escalar](https://es.wikipedia.org/wiki/Magnitud_escalar) relacionada con la [energía interna](https://es.wikipedia.org/wiki/Energ%C3%ADa_interna) de un sistema termodinámico, definida por el [principio cero de la termodinámica](https://es.wikipedia.org/wiki/Principio_cero_de_la_termodin%C3%A1mica)

**4.-Temperatura en Grados Celcius**

Es la unidad termométrica cuyo 0 se ubica 0,01 grados por debajo del punto triple del agua y su intensidad calórica equivale a la de kelvin.

El grado Celsius pertenece al Sistema Internacional de Unidades, con carácter de unidad accesoria, a diferencia del kelvin, que es la unidad básica de temperatura en dicho sistema, aunque es igual a la Celsius.

**5.-Temperatura en** [**Fahrenheit**](https://es.wikipedia.org/wiki/Grado_Fahrenheit)**.**

El grado Fahrenheit (representado como ℉) es una escala de temperatura propuesta por Daniel Gabriel Fahrenheit en 1724. La escala establece como las temperaturas de congelación y ebullición del agua, 32 ℉ y 212 ℉, respectivamente. El método de definición es similar al utilizado para el grado Celsius (℃).

**6.-Temperatura en grados Kelvin**

Unidad de temperatura de la escala creada por William Thomson Kelvin, en el año 1848, sobre la base del grado Celsius, estableciendo el punto cero en el cero absoluto (−273,15 °C) y conservando la misma dimensión. Kelvin, a sus 24 años introdujo la escala de temperatura termodinámica, y la unidad fue nombrada en su honor.

Es una de las unidades del Sistema Internacional de Unidades y corresponde a una fracción de 1/273,16 partes de la temperatura del punto triple del agua.2​ Se representa con la letra K, y nunca "°K". Actualmente, su nombre no es el de "grados kelvin", sino simplemente "kelvin".2​

**7.- Punto de Rocío**

**.**

El rocío es un fenómeno físico en el cual la humedad se condensa en formas de gotas de agua, este fenómeno es causado por la diminución brusca de la temperatura, o el contacto con superficies frías.

El punto de rocío o dew point es la temperatura a la cual ha de bajar una masa de aire para que la humedad relativa llegue al 100%, es decir, para que esta masa de aire quede saturada.

El poder que tiene el aire de contener vapor de agua no es constante, sino que depende de la temperatura. Como más caliente es el aire, más cantidad de vapor de agua puede contener. Si una masa de aire a una determinada temperatura tiene, por ejemplo, la mitad del vapor de agua que puede contener a esta temperatura, podemos decir que la humedad relativa es del 50%. Si ahora bajamos progresivamente la temperatura 20 de este aire, la cantidad de vapor que podrá contener es cada vez menor, y por tanto la humedad relativa subirá. Cuando la humedad relativa llega al 100% la temperatura del aire coincide con el punto de rosada.

El punto de rocío puede ser fácilmente calculado una vez tenemos la temperatura y la humedad. Una de las posibles aproximaciones, formulada por Hardy B. para calcular el dew point es la que se muestra a continuación en la siguiente fórmula.

****

Constantes derivadas de la ecuación de Hardy B, (Temas de Físco química)

Donde:

/

T es la temperatura seca

HR la humedad relativa

**Lcd i2c**

El controlador de LCD I2C es un dispositivo que nos permite controlar una pantalla a través del bus I2C, usando únicamente dos cables.

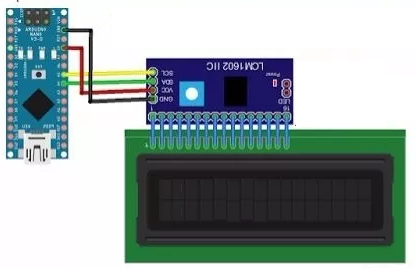
En [esta entrada](https://www.luisllamas.es/arduino-lcd-hitachi-hd44780/) aprendimos a manejar un display LCD Hitachi con controlador HD44780, una familia de pantallas barata y sencillas de emplear.

Pero usar esta pantalla directamente desde Arduino requería emplear una gran cantidad de pines de Arduino, lo que supone un enorme desperdicio de recursos, que deberían estar ocupados en cosas mucho más importantes que encender un simple display.

LCD es el acrónimo de Liquid Crystal Display (en español Pantalla de Cristal Líquido).

Hay una **amplia gama de pantallas LCDs** que se pueden utilizar con Arduino. se pueden diferenciar por el número de filas y columnas, su tamaño.

Por ejemplo, una pantalla LCD de 16×1 tendrá una fila de 16 caracteres es decir, solo podremos mostrar 16 caracteres simultáneamente, al igual que un LCD de 20×4 tendrá 4 filas de 20 caracteres cada una.

****

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

// Constructor de la librería de LCD 16x2

LiquidCrystal\_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

void setup()

{

lcd.begin(16, 2); // Indicar a la libreria que tenemos conectada una pantalla de 16x2

lcd.home (); // Mover el cursor a la primera posición de la pantalla (0, 0)

}

void loop()

{

lcd.setCursor ( 0, 0 ); // Imprimir "Hola Mundo" en la primera linea

lcd.print(" Hola Mundo "); // Mover el cursor a la segunda linea (1) primera columna

lcd.setCursor ( 0, 1 ); // Imprimir otra cadena en esta posicion

lcd.print(" Instituto Tec."); // Esperar un segundo

delay(1000);

}

**Sensor dht11 serie**

El DHT11 es un sensor que permiten realizar la **medición simultánea de temperatura y humedad**.

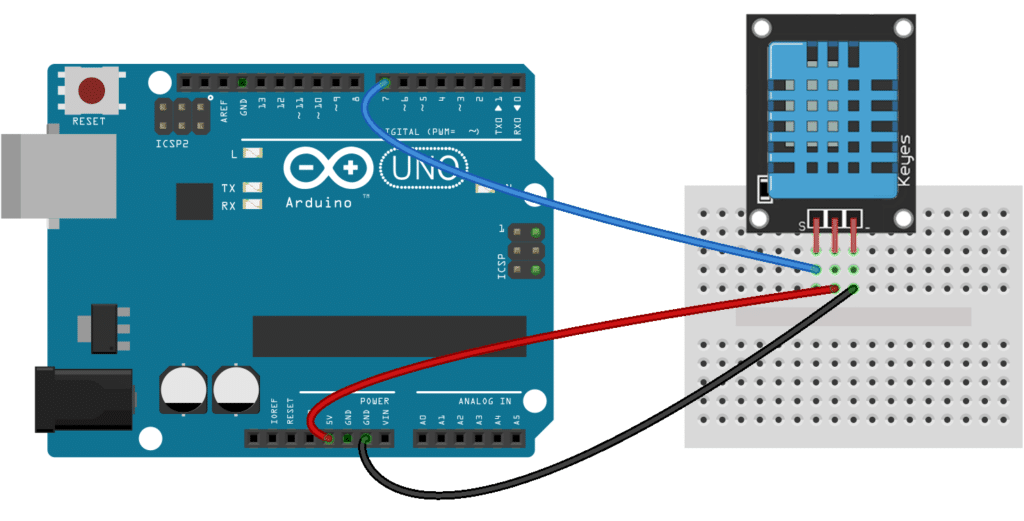
Este sensor dispone de un procesador interno que realiza el proceso de medición, proporcionando la medición mediante una señal digital, por lo que **resulta muy sencillo obtener la medición desde un microprocesador como Arduino**.

Las características del DHT11 especialmente en rango de medición y precisión.

* Medición de temperatura entre 0 a 50, con una precisión de 2ºC
* Medición de humedad entre 20 a 80%, con precisión del 5%.
* Frecuencia de muestreo de 1 muestras por segundo (1 Hz)

**El DHT11 es un sensor** que podemos usar con fines de formación, pruebas, o en proyectos.

**Conexiones del Circuito**



**Realizando la medición de temperatura Y humedad**

/=============================================

// Programa para medir Humedad y Temperatura

// Revisión 1.0

// M.I. Victor Manuel Mora Romo

// Instituto Tecnológico de Aguascalientes

//=============================================

#include "DHT.h"

#define DHTPIN 2

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

float hi;

void setup()

{

Serial.begin(9600);

Serial.println("=============================================");

Serial.println(" Programa para medir Humedad y Temperatura ");

Serial.println(" Revisión 1.0 ");

Serial.println(" M.I. Victor Manuel Mora Romo ");

Serial.println(" Instituto Tecnológico de Aguascalientes ");

Serial.println("=============================================");

dht.begin();

}

void loop()

{

delay(2000);

// Reading temperature or humidity takes about 250 milliseconds!

// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

float h = dht.readHumidity(); // Read temperature as Celsius (the default)

float c = dht.readTemperature();

float f= ((9/5.0)\*c)+32;

float k= c+273.15;

if (isnan(h) || isnan(c)) // Check if any reads failed and exit early (to try again).

{

Serial.println("Failed to read from DHT sensor!");

return;

}

Serial.print("Humidity: ");

Serial.print(h);

Serial.println(" ");

Serial.print("Temp. C: ");

Serial.print(c);

Serial.print(" \*C ");

Serial.print("Temp. F: ");

Serial.print(f);

Serial.print(" \*F ");

Serial.print("Temp. K: ");

Serial.print(k);

Serial.println(" \*K ");

}

**Sensor de presión barométrica BMP180**

El sensor BMP180 está diseñado para leer la presión atmosférica, que es la fuerza que ejerce el aire (atmósfera) sobre la superficie de la tierra. En otras palabras se debe a la columna de aire que esta sobre determinada área, por esto, si medimos la presión atmosférica en puntos más altos la presión baja debido a que hay menos cantidad de aire sobre nosotros. 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, por lo que podemos discriminar la humedad y viento que también influyen en la presión.

Lo que mide el sensor BMP180 es la presión absoluta (Barométrica) y la temperatura, al sensar la temperatura podemos compensar la influencia de ésta en la presión, y con un valor más exacto de la presión podemos calcular la altitud.

Este sensor vine en diversos módulos que nos ayudan en su conexión, el que usaremos para este tutorial es el siguiente modulo:

Este módulo tiene un pequeño regulador de voltaje de 3.3V, por lo que podemos alimentarlo con los 5V de Arduino, las dimensiones del sensor son pequeños y es Ultra-bajo consumo de energía, pudiendo utilizar en muchas aplicaciones.

**Librería para el sensor de presión  BMP180**

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(Sobremuetreo);**

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);**

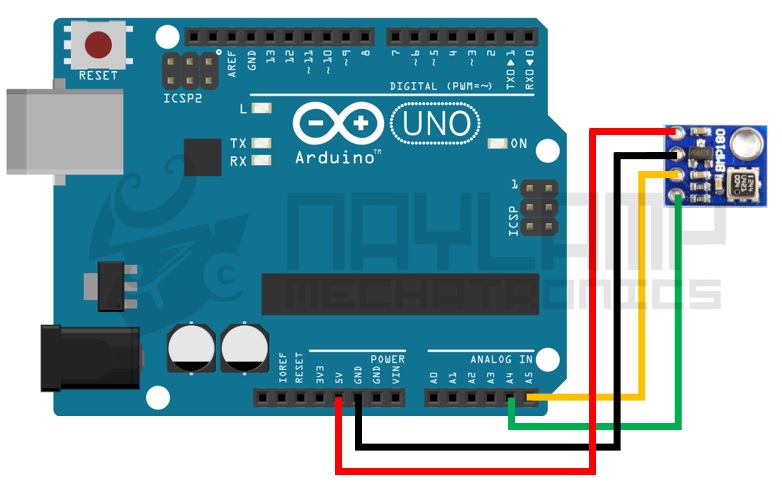
Esta función realiza el cálculo inverso a altitude(P, Po) , Dado una presión P (en mbar) y una altitud A (en metros) calcula la presión al nivel del mar o punto desde donde se mide la altura. Retorna el valor de la presión en mbar

Explicado esto, empecemos a realizar nuestros ejemplos:

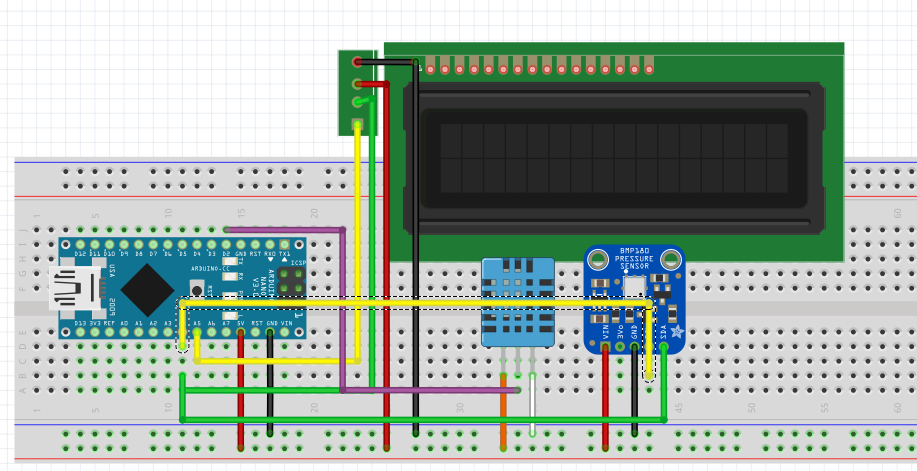
**Conexiones entre Arduino y módulo BMP180**

Las conexiones son como cualquier conexión I2C:

| **Adaptador** | **Arduino Uno, Nano, Mini.** |
| --- | --- |
| VCC | 5V |
| GND | GND |
| SCL | A5 |
| SDA | A4 |



**Realizando Lecturas de presión y temperatura y más variables**



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

A continuación se muestra el Sketch para realizar esta tarea:

#include <SFE\_BMP180.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include "DHT.h"

#define DHTPIN 2

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

float hi;

int Tiempo=2000;

// Constructor de la librería de LCD 16x2

SFE\_BMP180 bmp180;

double PresionNivelMar=1013.25; //presion sobre el nivel del mar en mbar

double Po; //presion del punto inicial para h=0;

char status;

double T,P,A,B,atm;

LiquidCrystal\_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

void setup()

{

// Serial.begin(9600);

lcd.begin(16, 2); // Indicar a la libreria que tenemos conectada una pantalla de 16x2

lcd.home (); // Mover el cursor a la primera posición de la pantalla (0, 0)

lcd.clear();

if (bmp180.begin())

Serial.println("BMP180 iniciado correctamenten");

else

{

Serial.println("Error al iniciar el BMP180");

while(1); // bucle infinito

}

}

void loop()

{

// Reading temperature or humidity takes about 250 milliseconds!

// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

float h = dht.readHumidity(); // Read temperature as Celsius (the default)

float rt = dht.readTemperature();

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)

{

lcd.clear();

lcd.setCursor(1,0);

lcd.print("Meteorologica");

delay(100);

//----Mostrar Presión Monitor Serial y LCD

atm= P\*0.000987; // Conversión de mbar a atm

lcd.setCursor(0,1);

lcd.print("Pres:");

lcd.setCursor(6,1);

lcd.print(atm,2);

lcd.print(" atm");

delay(2000);

//----Mostrar Altitud en metros sobre nivel del mar Monitor Serial y LCD

//-------Calculamos la altitud--------

A= bmp180.altitude(P,PresionNivelMar);

lcd.setCursor(0,1);

lcd.print("Alt :");

lcd.print(A,2);

lcd.print("msnm");

delay(2000);

}

}

}

}

delay(1000);

///////////////////Sensor DHT11

lcd.setCursor (0,0); // Imprimir "Hola Mundo" en la primera linea

lcd.print("Humidity: ");

lcd.print(h);

float c=rt/1.001;

lcd.setCursor (0,1); // Imprimir "Hola Mundo" en la primera linea

lcd.print("Temp C: ");

lcd.print(c,2);

lcd.print(" C ");

delay(Tiempo);

float f= ((9/5.0)\*c)+32;

float k= c+273.15;

lcd.setCursor (0,1); // Imprimir "Hola Mundo" en la primera linea

lcd.print("Temp F: ");

lcd.print(f);

lcd.print(" F ");

delay(Tiempo);

lcd.setCursor (0,1); // Imprimir "Hola Mundo" en la primera linea

lcd.print("Temp K: ");

lcd.print(k);

lcd.println(" K ");

delay(Tiempo);

//Sensación térmica por efecto de la humedad

lcd.clear();

lcd.setCursor (0,0);

lcd.println("STH: ");

if (c>19)

{

//El cálculo de la sensación térmica es en grados fahrenheit

float STH=-42.379+(2.04901523\*f) + (10.14333127\*h) - (0.22475541\*f\*h) - .00683783\*(f\*f)- .05481717\*(h\*h)+ .00122874\*(f\*f)\*h + .00085282\*f\*(h\*h)-.00000199\*(f\*f)\*(h\*h);

lcd.setCursor (5,0);

//conversión de STH a °C

float STHC=(STH-32)\*(5/9.0);

lcd.print(STHC);

// delay(Tiempo);

}

//no sirve para temperaturas bajas como por ejemplo 19 ºC.

else

{

lcd.setCursor (5,0);

lcd.print("N.A. ");

// delay(Tiempo);

}

///////////

//////Calculo de densidad del aire(Utilizar la presion en atm y la temperatura en Kelvin)

float DA=(atm\*29)/(0.0821\*k);

lcd.setCursor(0,1);

lcd.print("D.A.: ");

lcd.setCursor(4,1);

lcd.print(DA);

lcd.print(" kg/m^3");

delay(Tiempo);

lcd.clear();

//Cáculo de punto de Rocio

//Constantes derivadas de la ecuación de Hardy B, (Temas de Físco química)

float a=6.112;

float b=17.62;

float l=243.12;

float pr1= l\*(log(h/100.00)+((b\*c)/(l+c)));

float pr2= b-(log(h/100.00)+((b\*c)/(l+c)));

float prt=pr1/pr2;

lcd.setCursor ( 0, 0 );

lcd.print("P.R.: ");

lcd.print(prt,2);

lcd.print(" \*C");

delay(Tiempo);

}