**Estudio del anemómetro para la estación meteorológica:**

Descargamos un diseño de anemómetro desde *Thingiverse* (buscar: [Anemometer](https://www.thingiverse.com/thing:2559929)). Hemos comprobado que utiliza un rodamiento de 8 x 22 x 7 mm. Aunque aún no tenemos el rodamiento, hemos laminado con *Repetier* las tres piezas que forman el anemómetro con la intención de fabricarlas con la impresora 3D.

Hemos imprimido las partes del anemómetro en PLA y las hemos montado junto con un rodamiento, dándose el caso de que el saliente interior de la cazoleta central no queda encajado en el interior del rodamiento, por lo que hay que sujetarlo con un tornillo y una arandela que impide que se salga. De este modo, el rodamiento no funciona como tal, sino que solo sirve de apoyo superficial para la cazoleta central, lo que podría conseguirse con un anillo y una arandela metálica, en vez de utilizar el rodamiento.

**Estudio del pluviómetro para la estación meteorológica:**

También hemos buscado algún diseño interesante de pluviómetro, en *Thingiverse*, y hemos encontrado algunos que serían relativamente fáciles de construir. La mayoría de ellos están basados en un recipiente basculante que recoge el agua de lluvia y cuando se llena bascula hacia un lado, se llena de nuevo en el lado contrario y vuelve a bascular. Así una y otra vez. Para medir con este pluviómetro se usará un sensor hall como el de esta referencia: [enlace](https://www.vellemanformakers.com/product/hall-holzer-magnetic-switch-module-2-pcs-vma313/).

**Estudio del sensor DHT22 para la medida de la temperatura y la humedad relativa del aire:**

Utilizamos un sensor de temperatura y humedad para medir la humedad relativa del aire y también la temperatura. Lo hemos probado, después de instalar las librerías necesarias para el tipo de sensor, un DHT22, y hemos conseguido medir la humedad y la temperatura a través de él. Se ha usado como control un Arduino UNO estándar.

Las librerías se han descargado de:

* Adafruit DHT Humidity & Temperature Sensor Library en https://github.com/adafruit/DHT-sensor-library,
* Adafruit Unified Sensor Driver en https://github.com/adafruit/Adafruit\_Sensor

Sensor de temperatura y humedad DHT22

La instalación ha sido manual dentro del IDE de Arduino en Programa|Incluir Librería|Añadir biblioteca .ZIP de ambos ficheros. Es importante comprobar después que las librerías están en el menú Incluir Librería. Su uso consiste en incluirlas en el sketch al principio con el menú anterior Incluir Librería.

Ejemplo de uso:

#include <DHT.h>  
#include <DHT\_U.h>

#define DHTTYPE DHT22   // DHT 22  (AM2302), AM2321

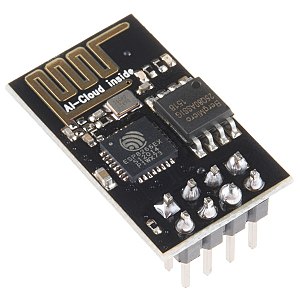
// Conexión:  
// Pin 1, izda., a +5V,  
// Pin 2 (data) a DHTPIN,  
// Pin 3 no conectado,  
// Pin 4, dcha., a GND,  
// Resistencia 10K entre pin 2 y 1,  
   
const int DHTPin = 4; //pin lectura,  
DHT dht(DHTPin, DHTTYPE); //clase,  
   
void setup() {  
   Serial.begin(9600);  
   Serial.println("DHTxx test!");  
   dht.begin();  
}  
   
void loop() {  
   // Esperar 1 seg. entre medidas:  
   delay(1000);  
   
   // La lectura T y H (datasheet: consultar) lleva unos 250 ms:  
   float h = dht.readHumidity();  
   float t = dht.readTemperature();

   // Si fallo lectura:  
   if (isnan(h) || isnan(t)) {  
      Serial.println("Fallo de lectura!");  
      return;  
   }

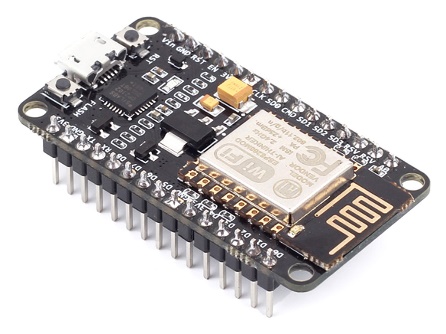
   //Salida:  
   Serial.print("Humedad: ");  
   Serial.print(h);  
   Serial.print(" %\t");  
   Serial.print("Temperatura: ");  
   Serial.print(t);  
   Serial.print(" \*C\n");  
}

Salida del programa por el puerto serie:

*Humedad: 50.00 % Temperatura: 22.60 \*C*

[](https://en.wikipedia.org/wiki/File:ESP-01.jpg)**Estudio de la comunicación WiFi entre Arduino y un ordenador:**

Hemos realizado muchas pruebas con el módulo ESP8266, para la comunicación Wifi de Arduino, y hemos tenido muchos problemas. Tras consultar variada documentación parece que podrían resolverse añadiendo un adaptador de nivel de 3,3V a 5V entre el módulo Wifi y la controladora, pues los terminales del módulo wifi están diseñados para trabajar a 3,3V, pero los terminales de Arduino Uno trabajan con 5V.

Alimentamos con una fuente de tensión externa, conectando un módulo adaptador de niveles entre el módulo WIFI y Arduino UNO para adaptar las señales a 3,3V y 5V, respectivamente. También hemos probado a utilizar casi todas las velocidades de comunicación, obteniendo los mejores resultados con 115200 baudios, resultando con muchos errores también.

Módulo ESP8266 de Expressif utilizado

Como solución alternativa que implica el cambio de plataforma microcontroladora, pero no probada, se muestra un NODE-MCU LUA. Este dispositivo incluye un módulo WIFI.

**Estudio del sensor de presión atmosférica:**

Para medir la presión atmosférica hemos decidido usar un sensor tipo BME280 con esta referencia: [enlace](https://www.vellemanformakers.com/product/bme280-temperature-humidity-and-pressure-sensor-vma335/).

Nos informarnos e intentamos hacer funcionar un sensor BME280 de Velleman con una controladora NODE-MCU LUA con módulo ESP12E. Aún no tenemos el sensor BME280, de presión atmosférica y temperatura, pero hemos compilado un ejemplo obtenido de Internet, tras instalar las librerías correspondientes y la tarjeta controladora en el IDE de Arduino. Hasta llegar aquí hemos realizado numerosas pruebas con resultados erróneos.

El proceso de compilación finalmente exitoso ha sido:

Módulo BME280

* Primero: NodeMCU 1.0(ESP-12E Module) previamente instalado con *http://arduino.esp8266.com/stable/package\_esp8266com\_index.json* en Preferencias del IDE de Arduino.
* Segundo: añadida la librería y ejemplos BME280-2.3.0.zip desde *https://www.arduinolibraries.info/libraries/bme280*. Esta librería es la recomendada en el Manual de usuario del fabricante del sensor.
* Tercero: se cargó un ejemplo de test desde Archivo|Ejemplos del IDE Arduino de los de la librería BME280.

Recursos:

* Manual usuario BM280 de Welleman *en https://www.planetaelectronico.com/datasheets/20075-vma335\_a4v01.pdf*
* Ejemplo uso de BME280 y Node MCU Lua en *https://blog.alexellis.io/iot-nodemcu-sensor-bme280/*