

# **Nodo\_DHT\_Doxy**

Andrés Felipe Fernández Ríos  
Elkin Alejandro Ledesma Narvaez  
Angie Tatiana Perez Muñoz

Versión 1.0.0

## Índice de clases

### Lista de clases

Lista de clases, estructuras, uniones e interfaces con breves descripciones:

**dht\_message\_t (Estructura de datos enviada vía ESP-NOW con temperatura y humedad ) .....3**

## Índice de archivos

### Lista de archivos

Lista de todos los archivos con breves descripciones:

**C:/Users/Andres**

**Fernandez/Documents/Arduino/sketch\_may7a/Nodo\_TemperaturaHumedad/Nodo\_TemperaturaHumedad.ino .....4**

# Documentación de clases

## Referencia de la estructura `dht_message_t`

Estructura de datos enviada vía ESP-NOW con temperatura y humedad.

### Atributos públicos

- float **temp**
  - float **hum**
- 

### Descripción detallada

Estructura de datos enviada vía ESP-NOW con temperatura y humedad.

Definición en la línea **37** del archivo **Nodo\_TemperaturaHumedad.ino**.

---

### Documentación de datos miembro

#### **float dht\_message\_t::hum**

Humedad relativa en %%

Humedad relativa en porcentaje.

Definición en la línea **39** del archivo **Nodo\_TemperaturaHumedad.ino**.

#### **float dht\_message\_t::temp**

Temperatura en °C

Temperatura en grados Celsius.

Definición en la línea **38** del archivo **Nodo\_TemperaturaHumedad.ino**.

---

**La documentación de esta estructura está generada del siguiente archivo:**

- C:/Users/Andres  
Fernandez/Documents/Arduino/sketch\_may7a/Nodo\_TemperaturaHumedad/Nodo\_Temperat  
uraHumedad.ino

# Documentación de archivos

## Referencia del archivo C:/Users/Andres Fernandez/Documents/Arduino/sketch\_may7a/Nodo\_TemperaturaHumedad/Nodo\_TemperaturaHumedad.ino

```
#include <esp_now.h>
#include <WiFi.h>
#include <DHT.h>
#include "esp_wifi.h"
```

### Clases

struct **dht\_message\_t** *Estructura de datos enviada vía ESP-NOW con temperatura y humedad.*

### defines

- **#define DHTPIN 5**  
*Pin digital conectado al sensor DHT22.*
- **#define DHTTYPE DHT22**  
*Tipo de sensor DHT utilizado (DHT11, DHT22, etc.)*

### Funciones

- **DHT dht (DHTPIN, DHTTYPE)**  
*Instancia del sensor DHT.*
- **void OnDataSent (const uint8\_t \*mac\_addr, esp\_now\_send\_status\_t status)**  
*Callback llamado tras un intento de envío ESP-NOW.*
- **void DHTSensorTask (void \*parameter)**  
*Tarea que lee el sensor DHT22 y envía los datos.*
- **void setup ()**  
*Función de configuración inicial.*
- **void loop ()**  
*Bucle principal (vacío, ya que todo corre en tareas).*

### Variables

- **uint8\_t receiverMAC [] = {0x5C, 0x01, 0x3B, 0x72, 0xF2, 0xCC}**  
*Dirección MAC del nodo receptor al que se envían los datos.*

## Documentación de «define»

### #define DHTPIN 5

Pin digital conectado al sensor DHT22.

Definición en la línea **15** del archivo **Nodo\_TemperaturaHumedad.ino**.

### #define DHTTYPE DHT22

Tipo de sensor DHT utilizado (DHT11, DHT22, etc.)

Definición en la línea **17** del archivo **Nodo\_TemperaturaHumedad.ino**.

---

## Documentación de funciones

### DHT dht (DHTPIN , DHTTYPE )

Instancia del sensor DHT.

### void DHTSensorTask (void \* parameter)

Tarea que lee el sensor DHT22 y envía los datos.

- Lee temperatura y humedad.
- Verifica lecturas válidas (no NaN).
- Empaqueta y envía los datos al receptor.
- Espera 10 segundos antes de la siguiente iteración.

#### Parámetros

<i>parameter</i>	Parámetro de tarea no usado.
------------------	------------------------------

Definición en la línea **70** del archivo **Nodo\_TemperaturaHumedad.ino**.

### void loop ()

Bucle principal (vacío, ya que todo corre en tareas).

Definición en la línea **160** del archivo **Nodo\_TemperaturaHumedad.ino**.

### void OnDataSent (const uint8\_t \* mac\_addr, esp\_now\_send\_status\_t status)

Callback llamado tras un intento de envío ESP-NOW.

Muestra por Serial si el envío fue exitoso o falló.

#### Parámetros

<i>mac_addr</i>	Dirección MAC del peer destino.
<i>status</i>	Estado del envío (ESP_NOW_SEND_SUCCESS o fallo).

Definición en la línea **53** del archivo **Nodo\_TemperaturaHumedad.ino**.

### void setup ()

Función de configuración inicial.

- Inicia Serial.
- Inicializa el sensor DHT.
- Configura WiFi en modo estación.
- Fuerza el canal WiFi para ESP-NOW.
- Inicia ESP-NOW, registra callback, y añade peer receptor.
- Crea la tarea DHTSensorTask.

Definición en la línea **109** del archivo **Nodo\_TemperaturaHumedad.ino**.

---

## Documentación de variables

**uint8\_t receiverMAC[] = {0x5C, 0x01, 0x3B, 0x72, 0xF2, 0xCC}**

Dirección MAC del nodo receptor al que se envían los datos.

Definición en la línea **26** del archivo **Nodo\_TemperaturaHumedad.ino**.

## Nodo\_TemperaturaHumedad.ino

Ir a la documentación de este archivo.

```
00001
00002
00003 #include <esp_now.h>
00004 #include <WiFi.h>
00005 #include <DHT.h>
00006 #include "esp_wifi.h"
00007
00008 #define DHTPIN 5
00009 #define DHTTYPE DHT22
00010
00011 DHT dht(DHTPIN, DHTTYPE);
00012
00013 static const uint8_t WIFI_CHANNEL = 6;
00014
00015 uint8_t receiverMAC[] = {0x5C, 0x01, 0x3B, 0x72, 0xF2, 0xCC};
00016
00017 typedef struct {
00018     float temp;
00019     float hum;
00020 } dht_message_t;
00021
00022 static dht_message_t sensorSend;
00023
00024 void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
00025     Serial.printf("[ESP-NOW] Envío a %02X:%02X:%02X:%02X:%02X:%02X -> %s\n",
00026         mac_addr[0], mac_addr[1], mac_addr[2],
00027         mac_addr[3], mac_addr[4], mac_addr[5],
00028         status == ESP_NOW_SEND_SUCCESS ? "OK" : "FALLÓ");
00029 }
00030
00031 void DHTSensorTask(void *parameter) {
00032     (void) parameter; // Evitar advertencia de parámetro no usado
00033
00034     for (;;) {
00035         float t = dht.readTemperature();
00036         float h = dht.readHumidity();
00037
00038         if (isnan(t) || isnan(h)) {
00039             Serial.println("[DHT] Error al leer sensor");
00040         } else {
00041             // Mostrar lecturas por Serial
00042             Serial.printf("[DHT] Temp: %.1f°C | Hum: %.1f%%\n", t, h);
00043
00044             // Preparar mensaje
00045             sensorSend.temp = t;
00046             sensorSend.hum = h;
00047
00048             // Enviar estructura via ESP-NOW
00049             esp_err_t res = esp_now_send(receiverMAC, (uint8_t *)&sensorSend,
00050                 sizeof(sensorSend));
00051             if (res != ESP_OK) {
00052                 Serial.printf("[Error] Fallo al enviar datos (err %d)\n", res);
00053             }
00054         }
00055
00056         // Esperar 10 segundos
00057         vTaskDelay(pdMS_TO_TICKS(10000));
00058     }
00059 }
00060
00061 void setup() {
00062     // Inicializar Serial para depuración
00063     Serial.begin(115200);
00064     delay(100);
00065
00066     // Iniciar sensor DHT
00067     dht.begin();
00068
00069     // Configurar WiFi en modo estación
00070     WiFi.mode(WIFI_STA);
00071 }
```

```

00119 Serial.printf("[Setup] MAC STA: %s\n", WiFi.macAddress().c_str());
00120
00121 // Configurar canal para ESP-NOW
00122 esp_wifi_set_channel(WIFI_CHANNEL, WIFI_SECOND_CHAN_NONE);
00123 Serial.printf("[Setup] Canal WiFi: %d\n", WIFI_CHANNEL);
00124
00125 // Iniciar ESP-NOW
00126 if (esp_now_init() != ESP_OK) {
00127     Serial.println("[Error] No se pudo iniciar ESP-NOW");
00128     return;
00129 }
00130
00131 // Registrar callback de envío
00132 esp_now_register_send_cb(OnDataSent);
00133
00134 // Configurar peer receptor
00135 esp_now_peer_info_t peerInfo = {};
00136 memcpy(peerInfo.peer_addr, receiverMAC, 6);
00137 peerInfo.channel = WIFI_CHANNEL;
00138 peerInfo.encrypt = false;
00139
00140 if (esp_now_add_peer(&peerInfo) != ESP_OK) {
00141     Serial.println("[Error] Fallo al agregar peer receptor");
00142     return;
00143 }
00144
00145 // Crear tarea DHTSensorTask en Core 1
00146 xTaskCreatePinnedToCore(
00147     DHTSensorTask, /* Función de la tarea */
00148     "DHTSensorTask", /* Nombre de la tarea */
00149     2048, /* Tamaño de stack */
00150     NULL, /* Parámetro */
00151     1, /* Prioridad */
00152     NULL, /* Handle */
00153     1 /* Núcleo */
00154 );
00155 }
00156
00160 void loop() {
00161     vTaskDelay(portMAX_DELAY);
00162 }

```