

# Configuración, integración y usos de un ESP32 tasmoteado en PVControl+

## 0. Introducción

Desde hace tiempo ando detrás de conectar al MQTT de PVControl+ algunos sensores, en este caso de temperatura, de diferentes lugares donde tengo la FV, más allá de los que se pueden integrar físicamente a la Rpi o la PCB: habitaciones, el entorno de la construcción, etc. Por otro lado, también me interesaba relacionar condiciones establecidas por sensores de este tipo y vincularlos con relés como, por ejemplo, el de un radiador o, incluso, con aires aa/cc o lo que sea.

Ya había visto que con Python se podía integrar DHT11 o DHT22 a través de un ESP32 (Imagen 1). Pero, dado que no tengo ni idea de programar, busco remedios que me puedan facilitar la vida con PVControl+ mientras aprendo algo de código. Al descubrir que Tasmota se estaba extendiendo también a este tipo de chips y al tener un entorno web bastante intuitivo y fácil de configurar, me lancé a probar.



## 1. Primeros pasos:

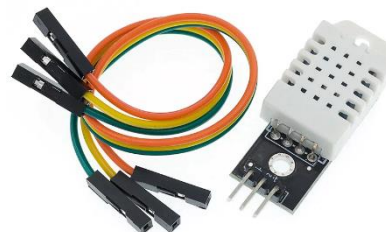
Lo primero que tenemos que hacer es conseguir un ESP32 (Imagen 1). Es muy probable que se pueda hacer también con otros chips, tipo NodeMCU, Wemos, etc. Imagino que sí, pero yo he probado con este. Dónde y cómo conseguir es fácil. En el mismo foro hay enlaces a diferentes tiendas online donde se pueden encontrar a precios asequibles.

Lo segundo que necesitamos es un sensor de temperatura (y humedad) que conectar al módulo. Yo he utilizado un DHT22 (Imagen 2), pero en el futuro probaré para exteriores un BME280 dado que también captura presión atmosférica (llamadme raro).

Un cable USB-microUSB (ojocuidado que sea de datos y no de alimentación). En caso de no tener cable con salida microUSB.

Una versión estable de [Tasmota para ESP32](#) (en el momento en que escribo este tutorial está disponible la 9.5.0 (Michael) enlace [aquí](#)).

Un programa para flashear el chip ([Esp flasher](#) o la propia web de [Tasmota](#) entre otros).



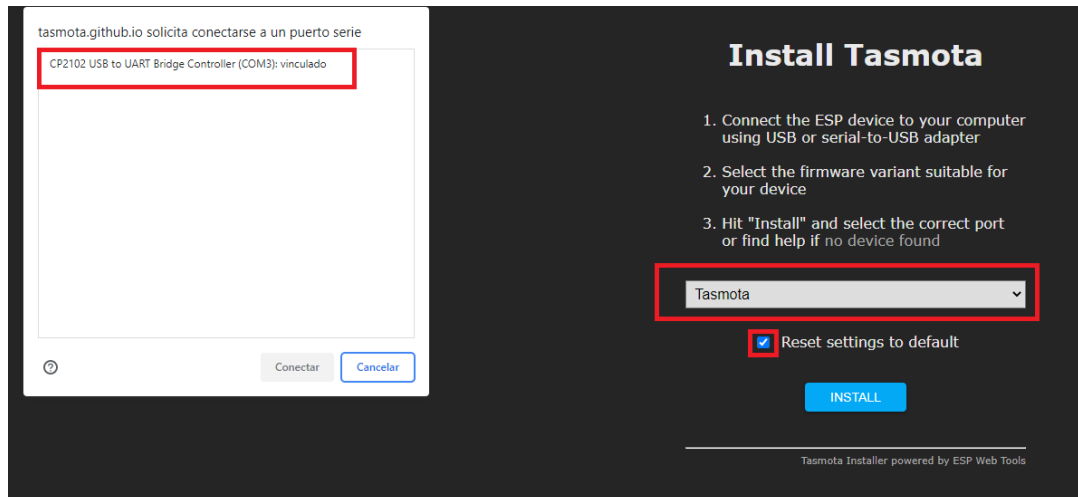
## 2. Flasheando el ESP32

Lo primera que tenemos que hacer es conectar nuestro esp32 al PC a través del USB. Si es la primera vez que conectamos uno de estos dispositivos a nuestro ordenador es muy probable que debamos instalar los drivers. Algunas veces, en las opciones ocultas de las actualizaciones de Windows (que es desde donde he estado trasteando) nos aparece la posibilidad de actualizar o instalar los drivers de los puertos de comunicaciones (COM). En caso de que no sea así, [aquí](#) y [aquí](#) están los enlaces.

Para flashear el dispositivo voy a hacerlo a través de la página web, dado que así no es necesario instalar nada y es sencillísimo.

Lo primero que haremos será seleccionar el software a instalar en el menú desplegable. Seleccionaremos “Tasmota” si lo queremos en inglés o bien “Tasmota (Spanish)” si lo queremos en español dentro de las opciones “Release”.

Después, activaremos la opción “reset settings to default”, pulsaremos el botón boot de nuestro ESP32 y le daremos a “install”. Aparecerá una ventanita donde tendremos que seleccionar el puerto (COM) en el que está conectado el ESP32 a través del USB y le damos a conectar. En caso de que no pulsemos el botón boot, nos dirá en letras bien grandes que no lo hemos hecho y que debemos pulsarlo. Cuando empiece a contar el porcentaje de flasheo soltaremos el botón boot. Una vez que se haya flasheado el dispositivo nos aparecerá un mensaje de que todo ha ido bien, lo desconectamos del USB y lo volvemos a conectar.



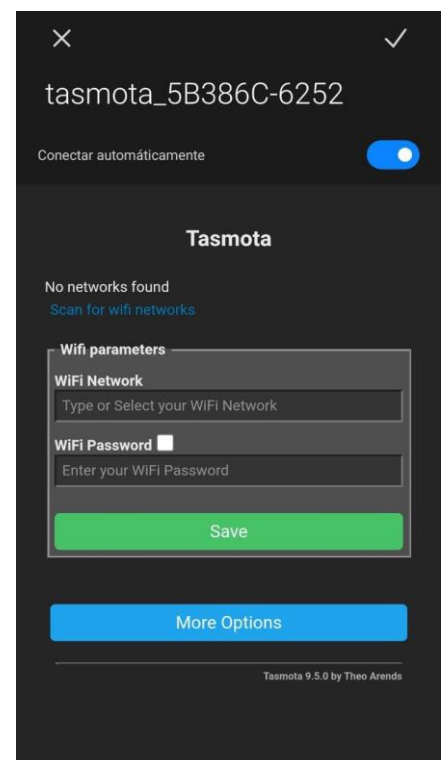
### 3. Conectándonos al dispositivo.

A partir de este momento el dispositivo se habrá puesto en modo “AP” o punto de acceso para poder acceder a él y configurarle la red wifi a la que conectar. Con un dispositivo que cuente con wifi miraremos las redes disponibles y se habrá generado una llamada “tasmota\_xxxxxx-yyyy”, el nombre que tiene ahora mismo nuestro dispositivo.

Una vez que hemos conectado a esa red abrirá la interfaz de la AP, donde tendremos que seleccionar la red donde queremos que se conecte (principalmente la misma donde esté PVControl+) y su contraseña. Intentará conectar y un mensaje nos dirá que lo ha conseguido y la IP que se le ha asignado.

NOTA: Otra opción rápida es usar Tasmotizer o ESPlorer con el dispositivo conectado y enviarle los datos de nuestra wifi y MQTT para no tener que conectarnos al dispositivo y ahorrarnos este paso y el del MQTT, como veremos más adelante.

Hay que advertir también que a día en que se está escribiendo este manual Tasmotizer no tiene soporte para flashear ESP32, pero sí que nos vale para enviarle nuestra configuración.



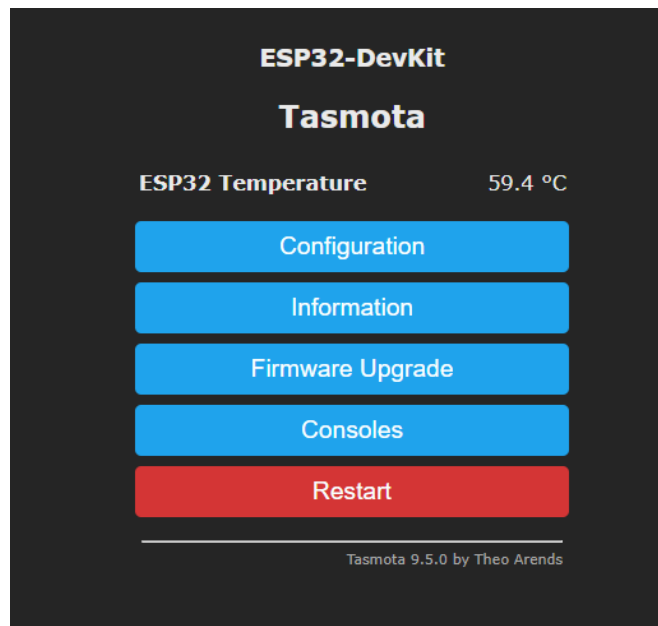
## 4. Configuración del ESP32

Acto seguido, abrimos el navegador e introducimos esa dirección. Entraremos en la interfaz web de Tasmota:

Lo primero que resalta de la interfaz es que por defecto ya nos muestra un sensor de temperatura de la CPU. De los cinco botones pulsaremos en el primero, “Configuración”. Allí nos aparecerá una serie de opciones para configurar. Pulsaremos en el primero: “Configure Module”. Dentro de ese submenú veremos todos los pines que tiene el dispositivo y las funciones que le podemos asignar a cada uno de ellos.

Antes de ello habremos conectado el DHT22 de la siguiente forma: el + a VIN del ESP32, - a cualquier GND y OUT al cualquier pin. En mi caso lo he conectado al pin GPIO 4, pin 24, impreso como D4. En Tasmota seleccionaremos el IO GPIO4 y en el desplegable le asignaremos AM2301 y le damos a “Save”. Con esto se reiniciará el dispositivo.

Una vez que el dispositivo se haya reiniciado, podremos ver cómo se está leyendo la información que nos proporciona el dispositivo conectado: Temperatura, Humedad, y Punto de rocío.



El siguiente paso será configurar el MQTT. Iremos a Configuration, Configure MQTT. Dentro de ese menú tendremos varias opciones a configurar:

Host: Dirección de nuestra rpi donde tenemos PVControl+

Port: el puerto MQTT de nuestro bróker, por defecto el que viene: 1883.

Client: el nombre con el que se reconocerá dentro del broker

User: el usuario de nuestro bróker (por defecto rpi)

Password: el password de nuestro broker (por defecto fv)\*\*\*

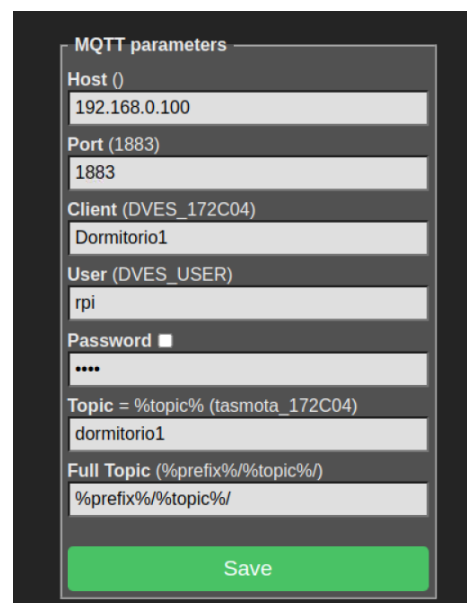
Topic: el topic con el que nuestro dispositivo se describirá al broker.

Fulltopic: lo dejamos tal cual.

Damos a Guardar y el dispositivo se reiniciará.

\*\*\* Durante las pruebas Mleón descubrió que los passwords cortos no los acepta Tasmota, por lo que daba error de conexión al bróker. En ese sentido fue necesario insertar un comando por consola. Para ello iremos a Consoles, Console y pegaremos este comando y le damos a intro:

Backlog mqttthost ipdevuestrohost; mqttport 1883; mqttuser: rpi; mqttpassword fv; topic elquesea



## 5. Integrando nuestro ESP32 en PVControl+

### 5.1 Con la última imagen publicada.

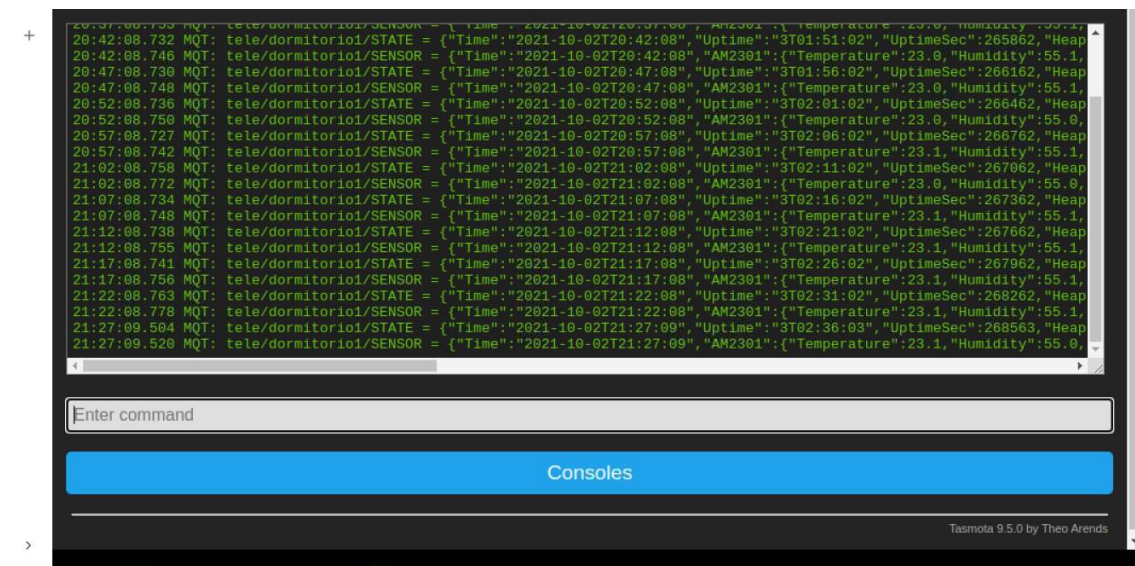
Una vez que nuestro dispositivo está funcionando, si tenemos instalada la última versión de la imagen de PVControl\* (altamente recomendable) debemos de modificar un mar de cosas en PARAMETROS\_FV.py

Abriremos nuestro programa con Geany o cualquier otro editor que utilicemos y modificaremos las siguientes líneas\*\*\*:

\*\*\* hablo de las líneas de mi archivo. Contrastad con vuestro archivo.

Línea 121: `usar_mqtt: 1`

Línea 122: `mqtt_suscripciones=` entre los corchetes copiaremos los topics de los dispositivos que queramos integrar y que habremos copiado de la consola de tasmota. Vemos dos ejemplos en las imágenes siguientes.



En este caso nos suscribiríamos a: `tele/dormitorio1/STATE` y `tele/dormitorio1/SENSOR`. Como vemos, dormitorio1 coincide con el nombre que le dimos al topic en la configuración de MQTT. Veamos ahora en Parametros\_FV.py:

Después, lo que podremos hacer es asignarle un valor de los que ya hay establecidos en PVControl+. En mi caso, que he agregado dos dht22, se los he atribuido tanto a Aux1 como a Aux2.

```
Aux1_sensor... = "d,['MQTT']['PVControl/DatosFV/Temp/Salon/SENSOR']['AM2301']['Temperature']".... # ADS, etc...
Aux2_sensor... = "d,['MQTT']['tele/dormitorio1/SENSOR']['AM2301']['Temperature']".... # ADS, etc...
Vred_sensor... = ""..... # Sensor Voltaje de red (d_huawei['Vred'],...)
```

En ello lo que decimos es que, el valor que asignamos es el que, dentro de ese topic, tomamos el valor del pin que asignamos más arriba ['AM2301'] discriminamos ['Temperature']. Podríamos discriminar también ['Humidity'], pero en este caso lo hemos desechado.

Guardaremos y reiniciaremos el servicio FV (en la consola “`sudo systemctl restart fv`”).

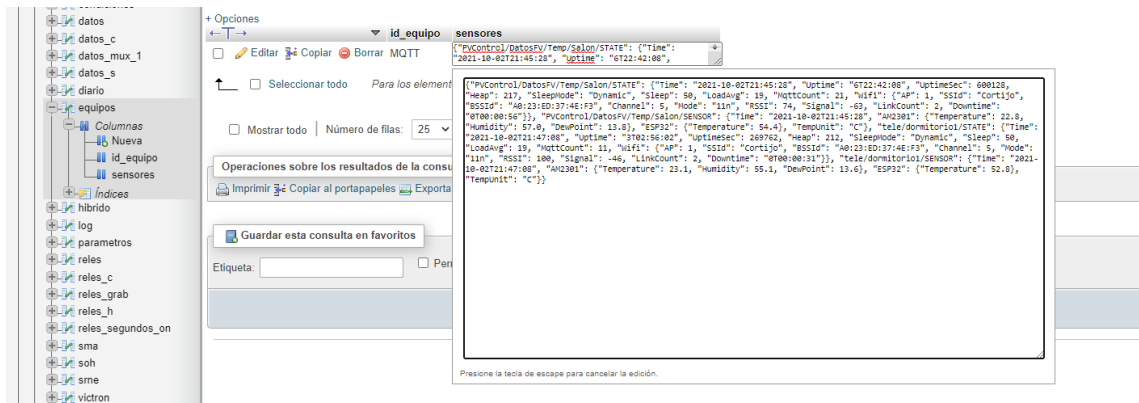
```
usar_mqtt = 1 # 1 = activa fv_mqtt.py para la suscripcion/lectura de los topics que se definen en mqtt_suscripciones
mqtt_suscripciones=["PVControl/DatosFV/Temp/Salon/SENSOR", "PVControl/DatosFV/Temp/Salon/STATE", "tele/dormitorio1/SENS
# -----
```

## Manual de integración de dispositivos tipo Esp32 con Tasmota en PVControl+

Lo siguiente que haremos es ver si se ha integrado bien este valor. Para ello abriremos nuestra bd: (<http://ipdelarpi/pmyadmin/index.php>). recordemos que el usuario por defecto es rpi y la contraseña fv.

Dentro ya de nuestra base de datos, pincharemos en control solar y después en tablas. Se despegarán todas las tablas que tenemos creadas y seleccionaremos equipos, columnas y sensores.

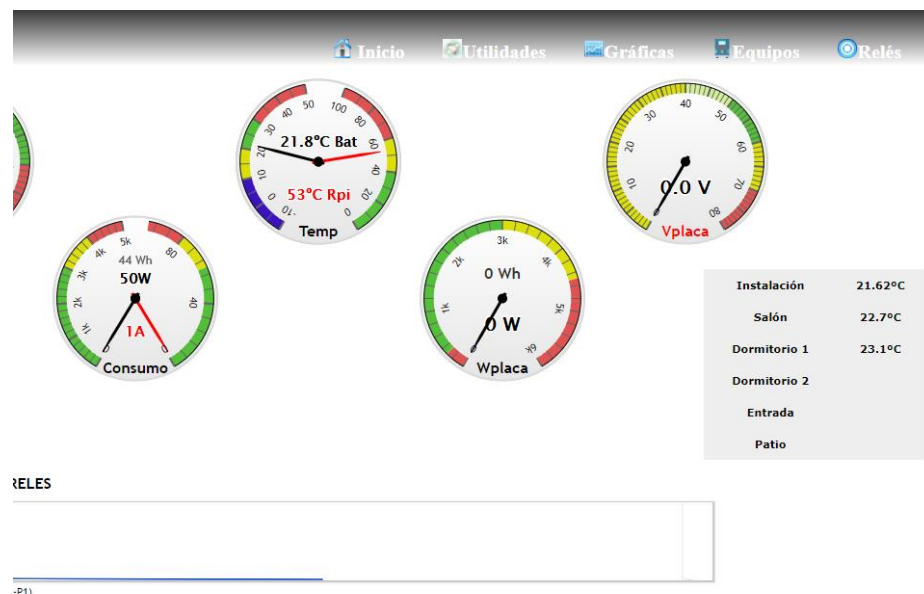
Si se ha integrado bien en la columna sensores deberíamos de ver datos similares a los de la siguiente imagen:



Es posible que nos dé algún error. Es posible que tengamos que ampliar el número de caracteres de esa casilla. Para ello pulsamos en el botón Examinar y, después en longitud/valores los ampliamos y le damos a Guardar. En mi caso lo he puesto en 5000.



Ya podríamos utilizar al gusto este valor. En mi caso he añadido, por ahora, dos DHT22 en las habitaciones de donde tengo la FV. En mi página principal de PVControl+ quedaría tal que así:





## 6. Uso con la PCB de Excedentes

**NOTA: POR AHORA, ESTO SOLO ES VÁLIDO PARA SSR POR ÁNGULO DE FASE.**

Para poder usar los excedentes debemos instalar una versión diferente de Tasmota. A la hora de seleccionar qué versión flashear en vez de elegir una estable seleccionaremos una de desarrollo (Development), bien Tasmota, bien Tasmota(Spanish).

Seguiremos el proceso de flasheado e instalación como se indican en los puntos 2 y 3.

Cuando asignemos la función del pin, seleccionaremos PWM y le daremos a guardar.

En cuanto a la configuración de los topics, en este caso tendremos que asignarle el topic /PVControl/Reles/5x. En caso de asignar el 53, por ejemplo, podríamos controlar los relees 531-538 (límite de canales que tiene el ESP32)

