

OpenDS+ - Derivador de excedentes **versión 1.1.0**

Manual de uso

1.- Que es OpenDS+

OpenDS+ es un derivador de excedentes y gestor energético orientado a los inversores Solax y Fronius en la actualidad. Su función es aprovechar la potencia no usada derivándola a una carga resistiva como un termo o una estufa y mediante relés o a través del protocolo mqtt o http a cualquier tipo de carga mediante sistemas domóticos, enchufes inteligentes, etc., para ello. Con esto conseguimos aprovechar esa energía que es capaz de producir la instalación y la derivamos a otra carga puntualmente y de forma automática. También podemos controlar los parámetros de vertido a red según nuestras necesidades.

El control de potencia de la carga resistiva se realiza mediante un dimmer con protocolo PWM (Gestión de frecuencia de onda) y siempre a esta salida nos vamos a referir con el nombre pwm.

Para trabajar correctamente, los parámetros de control PWM deberán tener un diferencial de unos 200w si el inversor funciona con vertido cero, es decir, si mantenemos el valor de vertido en el inversor a 0, la configuración pwm debería estar como mínimo entre -100 y -200, para que el margen de regulación del inversor no interfiera en el margen de regulación del derivador. Si queremos mantener un vertido cercano a 0 lo ideal sería configurar el derivador con valores -1 y -20 y el vertido del inversor con un mínimo de 200w. En ese caso la regulación del vertido cero la está realizando el derivador.

Aunque no es la función del derivador, durante su uso se mantiene un nivel de vertido programado (Sea vertido cero o el que le indiquemos), por lo que deberemos de configurar en el inversor un vertido de control mayor que el del derivador para que éste pueda detectar que disponemos de más energía.

2.- ¿Donde comprar OpenDS+?

OpenDS+ es un proyecto OpenSource, es decir de código abierto que te puedes hacer tu mismo con unos mínimos conocimientos. En este momento no se vende montado y tampoco está previsto hacerlo.

En el caso de no disponer de los conocimientos necesarios para la grabación si podemos suministrar las placas con el firmware grabado poniéndose en contacto con el autor.

3.- Materiales necesarios.

El sistema está diseñado para trabajar con un modulo DEV-KIT 32 de la empresa Heltec y un módulo ESP01 en el caso de que la versión del wifi de tu equipo sea una versión 2.

Para el control de la salida variable se usa un “dimmer” a través del protocolo PWM, según los esquemas disponibles en el git del proyecto. Dado que es un proyecto vivo y por lo tanto variable, dejamos la pertinente documentación en el Git, y en este manual nos limitaremos a la grabación del firmware, la puesta en marcha y un esquema básico de instalación.

La dirección del GIT es: <https://github.com/iqas/derivador>

Las opciones que sólo estén disponibles en versiones superiores a la 1.0 estarán indicadas con el formato similar el siguiente, indicando a partir de que versión están disponibles: ^(1.1)

OpenDS+ - Derivador de excedentes **versión 1.1.0**

Manual de uso

4.- Firmware

El firmware es el código que necesitamos instalar en los controladores para que realice su función y lo podremos realizar de 2 maneras, de las cuales sólo trataremos una de ellas.

a) Compilando el código fuente:

- Para ello necesitaremos el entorno Arduino con sus librerías y unos mínimos conocimientos de programación, esta opción la dejaremos para un entorno avanzado y no será tratada en este manual.

b) Grabación desde los ficheros binarios:

(1.1) (Este paso sólo es necesario realizarlo una vez, luego se puede actualizar directamente desde la web del derivador, tanto el Esp32/DevKit32 como el módulo M1 basado en esp-01)

- Se han añadido los ficheros binarios ya precompilados para que sea sencillo a un usuario el poder realizar la instalación del firmware.

Nos descargamos los siguientes drivers para el Heltec32/Esp32 y para el ESP01 (Sólo para Wifi V2 con firm M1):

- Driver para el grabador del ESP-01 (Ver imagen posterior)

<https://github.com/nodemcu/nodemcu-devkit/tree/master/Drivers>

- Driver para el HELTEC DEV-KIT-32

<https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>

Deberemos instalar estos Drivers para que Windows reconozca la electrónica.



With ESP-01



Grabador ESP01



Heltec DEV KIT 32

Después de esto descargamos el programa que se encargará de enviar los datos en forma binaria a la electrónica, lo descargaremos de la página del fabricante del chip en la siguiente dirección:

<https://www.espressif.com/en/support/download/other-tools>

OpenDS+ - Derivador de excedentes **versión 1.1.0**

Manual de uso

Nos descargamos el programa Flash Download Tools, lo ejecutamos y seleccionamos el ESP8266 para grabar el ESP01 y ESP32 para grabar el DevKit 32.

a) Grabacion del Esp32/Dev KIT 32:

Para este sistema deberemos usar la siguiente configuración:

Chip: esp32

Velocidad: 921600

Modo de flash: dio

Frecuencia: 80mhz

0xe000 boot_app0.bin

0x1000 bootloader_dio_80m.bin

0x10000 opendsdv32.bin (Para devkit32) y opendsesp32.bin para Esp32 (sin pantalla)

0x8000 esp232_derivator.ino.partitions.bin

b) ESP-01 (Sólo Solax X1 Boost wifi v2)

(Este firmware es alternativo al original que trae el ESP01 de fábrica, dado que da problemas a algunos usuarios, este firm lo detectará como V3, en todo caso se puede probar primero con el ESP01 sin grabar)

Para este controlador la configuración es la siguiente:

Velocidad: 115200

Modo de flash: dio

^(1.1)Flash: 1024Kb

Frecuencia: 40Mhz.

0x00000 m1.bin

IMPORTANTE:

Usad el firmware adecuado a la tarjeta, las carpetas incluidas indican:

esp32 → Módulo generico esp32 SIN PANTALLA

dev_kit_32 → Módulo esp32 Dev Kit de Heltec con pantalla

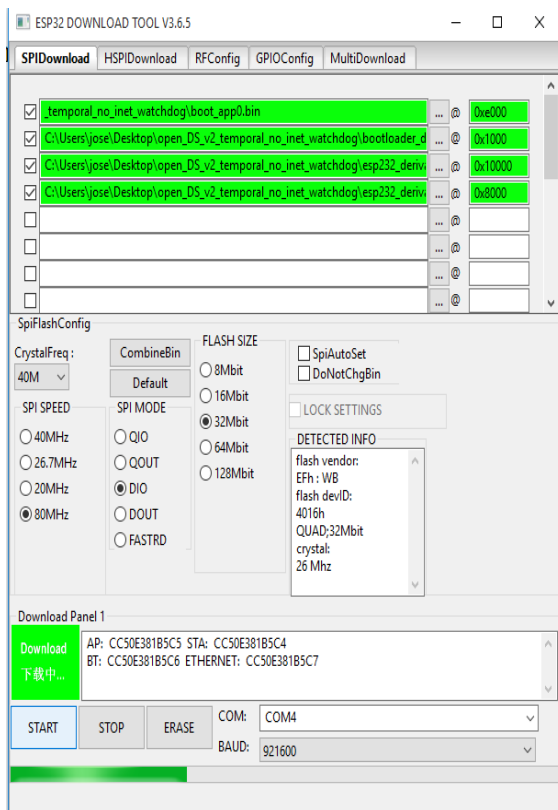
m1 → Módulo ESP01 (M1)

openDSv → Visor para OpenDS+ (Será integrado en OpenDS+ en la próxima versión)

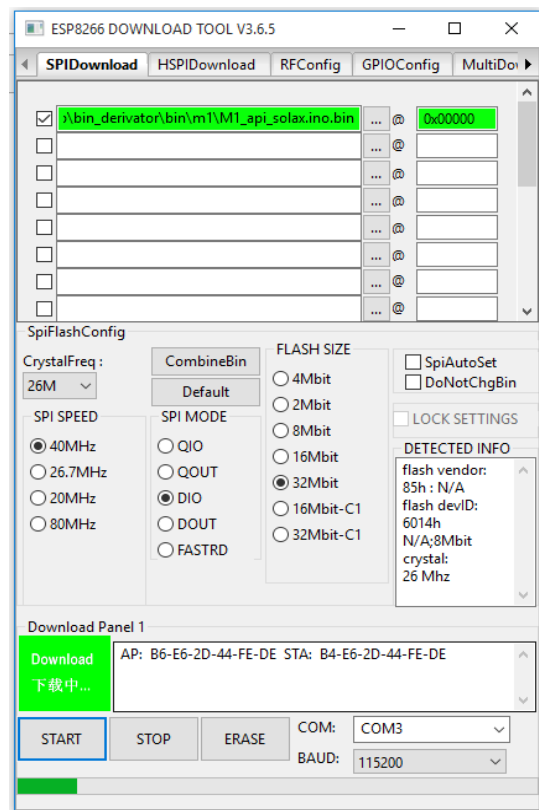
Con lo cual nos quedarían configurados como la siguiente imagen, colocando los ficheros indicados en cada dirección:

OpenDS+ - Derivador de excedentes **versión 1.1.0**

Manual de uso



Dev Kit 32



ESP-01

Pulsamos el botón START y comenzará la grabación, deberemos esperar hasta que finalice la transferencia.

Con esto tenemos grabado el firmware, procederemos al siguiente paso.

(1.1) En el caso de los híbridos seguiremos los pasos como si se tratase de la versión v1.

Grabación desde Linux:

Primero instalaremos el entorno Arduino para esp32 y esp8266, añadiendo en Archivo→ Preferencias→ Gestor de URLs Adicionales de Tarjetas:

http://arduino.esp8266.com/stable/package_esp8266com_index.json,

https://dl.espressif.com/dl/package_esp32_index.json

Luego nos vamos a Herramientas→ Placa→ Gestor de tarjetas y añadimos esp32 y esp8266, con esto ya tenemos las herramientas de grabación esptool.py instaladas.

(Cambiar USB0 por el puerto que uséis, se puede ver en Arduino)

Borramos flash:

`esptool.py --chip esp32 --port /dev/ttyUSB0 --before default_reset --after hard_reset erase_flash`

OpenDS+ - Derivador de excedentes **versión 1.1.0**

Manual de uso

Grabamos:

Para esp32

```
python ~/Arduino/hardware/heltec/esp32/tools/esptool.py --chip esp32 --port /dev/ttyUSB0 --baud 921600 --before default_reset --after hard_reset write_flash -z --flash_mode dio --flash_freq 80m --flash_size detect 0xe000 boot_app0.bin 0x1000 bootloader_dio_80m.bin 0x10000 opensdesp32.bin 0x8000 esp32_derivator_1.1.ino.partitions.bin
```

Para devKit32

```
python ~/Arduino/hardware/heltec/esp32/tools/esptool.py --chip esp32 --port /dev/ttyUSB0 --baud 921600 --before default_reset --after hard_reset write_flash -z --flash_mode dio --flash_freq 80m --flash_size detect 0xe000 boot_app0.bin 0x1000 bootloader_dio_80m.bin 0x10000 opensdsv32.bin 0x8000 esp32_derivator_1.1.ino.partitions.bin
```

Para M1:

```
python ~/Arduino/hardware/heltec/esp8266/esptool.py --port /dev/ttyUSB0 write_flash -fs 1MB -fm dout 0x0 M1_api_solax.ino
```

OpenDS+ - Derivador de excedentes **versión 1.1.0**

Manual de uso

5.- Configuración

Después de instalar el firmware, lo conectamos al PC o directamente al dimmer, pero no es necesario en esta fase que tengamos acceso a la Wifi del Solax.

(1.1) Nos conectaremos a la red wifi openDS+ y accederemos al sitio <http://192.168.4.1>, en el nos aparecerá la pantalla del derivador, pulsaremos Red y configuraremos nuestro acceso wifi. Guardamos y al reiniciar se conectará a nuestra wifi y podremos configurar el resto de campos.

(1.1) Para acceder deberemos hacerlo con el usuario: admin password: admin

(1.1) La clave del usuario se puede cambiar en el menú de Config.

(1.0) ~~Al encenderlo nos indicará "Connecting", se está intentando conectar a la wifi con las SSID por defecto, al no conseguir realizar la operación nos indicará que entra en modo SMARTCONFIG, para continuar haremos lo siguiente~~

(1.0) Instalamos la siguiente app en el móvil:

— Para Android: ESP8266 SmartConfig

— Para IOS: Espressif Esptouch

(1.0) ~~La ejecutamos con el móvil conectado a la wifi, ponemos la clave de nuestra wifi, le damos a CONFIRM y se cargarán los datos en el derivador.~~

El derivador se conectará a nuestra Wifi y nos indicará la IP en pantalla.

Nos vamos al explorador y escribimos la IP de la pantalla (Ej: <http://192.168.0.75>).

Nos arrancará y veremos que en la pantalla parpadea la letra "S", lo que nos indica que no tiene conexión con el inversor Solax y nos informará en la web de que la conexión no se puede realizar.

Nos mostrará los parámetros todos a cero, dado que no tiene conexión con el inversor.

En esta pantalla seleccionaremos, mediante el desplegable la versión del stick wifi de que disponemos, estando preseleccionada la versión V2.

Las opciones disponibles son:

- Solax V1 Híbridos (Para inversores offgrid con stick wifi v1 e híbridos v1)
- Solax Wifi V2 (Para inversores con stick wifi v2)
- Solax wifi v2 local (Para inversores con stick wifi v2, sin esp01, no hay conexión a red local)
- Fronius (Para inversores Fronius)

Debajo de los parámetros disponemos información del estado de los relés. Debajo de está información disponemos de botones de acceso al resto de menús que pasamos a enumerar:

The screenshot shows the 'openDS+' web interface. At the top, there's a header with the logo and a dropdown menu set to 'Solax Wifi v2'. Below the header, the title 'Monitorización' is displayed. The main content area shows several data fields with input boxes and units: 'Potencia Solar: 3240.00 W', 'Potencia de Red: -202.00 W', 'Energía Diaria Solar: 5.40 kWh', 'Potencia String 1: 1986 (438V) (4.50A) W', 'Potencia String 2: 1416 (319V) (4.40A) W', and 'PWM: 0 %'. Below these fields are four numbered circular icons (1, 2, 3, 4). At the bottom, there are five buttons: 'Red', 'Config', 'Salidas', 'Reboot', and 'OTA'.

OpenDS+ - Derivador de excedentes **versión 1.1.0**

Manual de uso

- Red: Configuración de las conexiones Wifi, tanto del área local como del Solax.
- Config: Configuración de otros parámetros, en esta versión solo cubre el servicio MQTT
- Salidas: Programación de las salidas, tanto del dimmer pwm como de los relés.
- Reboot: Reinicio del sistema (Pondrá a cero de forma escalonada el pwm antes de reiniciar)
- OTA: Actualización a través de internet, actualiza tanto el esp32 como el es01 en caso de disponer del mismo.

La primera fase de la configuración pasa por realizar las conexiones de red, configurando las mismas mediante el botón "Red"

Pulsamos el botón que pone RED y configuramos las conexiones Wifi 1, la Wifi 2 es opcional, se conectará a ella si falla la Wifi1.

En Datos Inversor deberemos elegir el SSID del inversor si disponemos del stick wifi v2 o Fronius o bien poner la IP del solax si es la v1 y de los híbridos serie SU o de Fronius.

2 Datos Inversor



3 Conexión de red

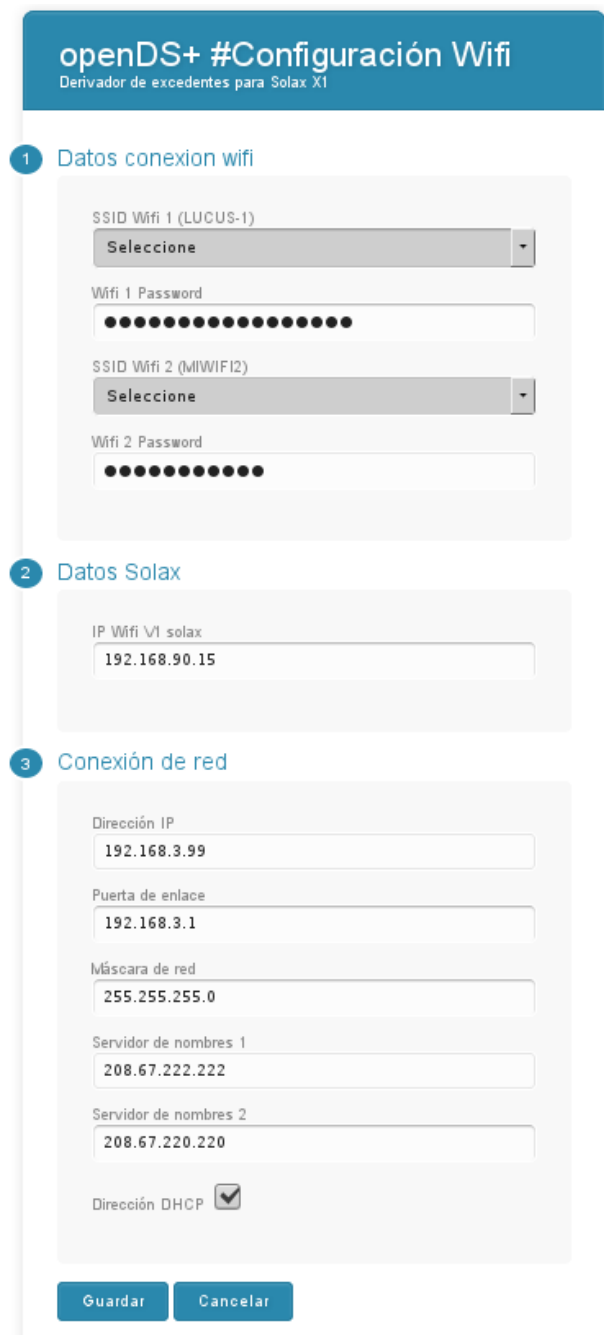
Esta operación la deberemos realizar con cobertura de la Wifi del Solax y con el esp01 instalado en caso de usar la v2. La S de la pantalla debería dejar de parpadear al conectarse al Solax.

La tercera opción es la configuración de red:

- Dirección IP: dirección en nuestra red local
- Puerta de enlace: Dirección ip de la puerta de salida, generalmente nuestro router
- Servidores de nombre: Podemos cambiarlos si lo deseamos, por defecto están los de openDNS

Config:

Dirección DHCP, si lo tenemos seleccionado (Opción por defecto), el router nos dará una IP que puede variar de una vez a otra y no se tendrán en cuenta los parámetros anteriores.



OpenDS+ - Derivador de excedentes **versión 1.1.0**

Manual de uso

Si no está seleccionada se tomarán los parámetros anteriores para la configuración de red.

Al pulsar Guardar, almacenará la configuración de red en la memoria EEPROM y se reiniciará el dispositivo.

Guardamos cambios y se reinicia en 5 segundos.

Configuración:

En este apartado disponemos de los campos para realizar la configuración del servicio MQTT, con los siguientes campos:

Mqtt Activo: Seleccionaremos para activar el servicio

Mqtt Broker: Dirección ip del broker mqtt (Testado con mosquitto)

Mqtt Usuario: Nombre del usuario mqtt

Mqtt Password: Clave del usuario mqtt

Temas: Temas para cada salida de relé

En cuanto pulsemos Guardar el dispositivo puede tener un cierto retardo si no estaba activo el servicio ya que tiene que registrarse en el broker.

Cambio de password:

Nos permite cambiar la configuración de la clave de acceso, por defecto admin – admin.

Api remota:

Con este servicio enviamos una solicitud http GET a un servidor remoto para que trate los datos. Sus funciones pueden variar desde un sistema domótico para controlar los parámetros a cualquier script de almacenamiento de los mismos para estadísticas.

Funcionamiento del sistema:

Deberemos configurar la dirección http con todos los parámetros necesarios, como ejemplo, para enviar la potencia solar en la variable GET potencia, al script data.php del servidor server.micasa.es tendríamos la siguiente línea (El sistema ya añade http:// a la línea por lo que no se debe incluir:

server.micasa.es/data.php?potencia=%wsolar%

The screenshot shows the 'openDS+ #Configuración' interface, specifically the '1 Datos MQTT' section. It contains the following fields and controls:

- MQTT Activo:** A checkbox that is currently unchecked.
- MQTT Broker:** A text input field containing '192.168.0.2'.
- MQTT Usuario:** A text input field containing 'MQTT_user'.
- MQTT Password:** A password input field with 12 dots representing the masked characters.
- MQTT Puerto:** A text input field containing '1883'.
- Temas:** Four text input fields labeled 'Tema R1' through 'Tema R4', each containing a topic string: '/solax/relay/1/cmdnd/POWER', '/solax/relay/2/cmdnd/POWER', '/solax/relay/3/cmdnd/POWER', and '/solax/relay/4/cmdnd/POWER' respectively.
- Buttons:** At the bottom, there are two buttons: 'Guardar' (Save) and 'Cancelar' (Cancel).

OpenDS+ - Derivador de excedentes **versión 1.1.0**

Manual de uso

En este caso la var %wsolar% se reemplaza por el valor de la potencia que genera el inversor y haríamos una petición a un api de un sistema del estilo <http://server.micasa.es/data.php?potencia=456> si la potencia generada en ese momento correspondiese a ese valor.

Las variables aceptadas son las siguientes:

%pv1c% -> corriente string 1
%pv2c% -> corriente string 2
%pv1v% -> tension string 1
%pv2v% -> tension string 2
%gridv%" -> tensión de red
%wsolar% -> potencia solar
%wtoday% -> potencia solar diaria
%wgrid% -> potencia solar
%pw1% -> potencia string 1
%pw2% -> potencia string 2
%wtogrid% -> energía enviada a red



The screenshot shows a configuration window titled 'API remota' with a blue header bar. Below the header, there is a label 'Web Api' followed by a text input field. At the bottom of the window, there are two blue buttons: 'Guardar' (Save) and 'Cancelar' (Cancel).

OpenDS+ - Derivador de excedentes versión 1.1.0

Manual de uso

Salidas:

En este menú configuramos las cargas del dispositivo, que pueden ser enlaces físcos a las salidas GPIO o bien enlaces virtuales mediante el protocolo mqtt.

1.- PWM:

En este menú configuramos los niveles de consumo de red a mantener, teniendo en cuenta que los valores negativos son consumo de red y los valores positivos son vertido a red. Así tenemos que si le indicamos el valor mínimo a -10w y el valor máximo a -30w, el dispositivo intentará mantener estos valores.

Cabe destacar que es recomendable que el valor mínimo y el valor de vertido a red del inversor tengan al menos una diferencia de 50w, siendo lo idóneo entre 100 y 200w.

Durante el apagado hay un retardo de 6 segundos aproximadamente para que se estabilice después de un encendido.

2.- Configuración Salidas Auto:

Aquí configuraremos los niveles (%) en los cuales una determinada salida de relé, sea física o virtual se inicia. Es decir si indicamos un valor de 30 en el R3, el relé R3 se activará cuando el dimmer PWM tenga una salida igual o superior a este valor. Este relé se mantendrá encendido mientras el valor PWM supere el 5%.

Indicar que a su vez, si el servicio está activo se notificará al tema el valor "ON" para el encendido del relé y el valor "OFF" para su apagado mediante el servicio mqtt.

Si no configuramos estos valores, por defecto las salidas se activarán al superar un valor de 170 sobre 180 en la variable PWM lo cual equivale a un 94% aproximadamente.

3.- Configuración manual:

Estas opciones están destinadas a activar las salidas de forma manual, aún cuando no tenemos excedentes.

openDS+ #Configuración Salidas

Derivador de excedentes para Solax X1

1 Configuración Dimmer PWM

PWM Dimmer Activo ☒

Consumo de red, valor mínimo

Consumo de red, valor máximo

2 Configuración Salidas Auto

R1 Inicio desde % sobre PWM

R2 Inicio desde % sobre PWM + R1

R3 Inicio desde % sobre PWM + R2

R4 Inicio desde % sobre PWM + R3

3 Configuración Manual

Relé 1 Activo ☐ Relé 2 Activo ☐

Relé 3 Activo ☐ Relé 4 Activo ☐

OpenDS+ - Derivador de excedentes **versión 1.1.0**

Manual de uso

6.- Datos suministrados

1.- Pantalla:

Los datos que aparecen en pantalla son los siguientes:

Solar	–	S W M v	–	Red
0				0
PWM 17% Relay 1 2 3 4				

S: Conexión con Solax, parpadea cuando falla y no consigue conectarse

W: Conexión Wifi, parpadea si la perdemos

M: Conexión Mqtt, parpadea si no hay conexión con el broker

v: Tipo de versión, 0- Solax Wifi v2 modo local, 1-Solax Wifi V1, 2-Solax Wifi V2 con ESP01 firm M1, 11- Fronius

PWM: Tanto por ciento de energía que estamos derivando al dimmer de la posible.

Relay: Indica el funcionamiento de los relés (parpadeando = funcionales)

2.- Api interna

Disponemos de los datos del inversor mediante una petición HTTP GET a la dirección del dispositivo, por ejemplo <http://192.168.90.107/?api>

Esta llamada nos retorna una cadena en formato JSON con los siguientes campos en la variable data:

1) *httpcode*: Código de respuesta de la petición al Solax

2) *solax_pv1c*: Intensidad del String 1

3) *solax_pv2c*: Intensidad del String 2

4) *solax_pv1v*: Tensión del String 1

5) *solax_pv2v*: Tensión del String 2

6) *solax_pw1*: Potencia del String 1

7) *solax_pw2*: Potencia del String2

8) *solax_gridv*: Tensión de red

9) *solax_wsolar*: Potencia solar

10) *solax_wtoday*: Energía diaria producida

11) *solax_wgrid*: Potencia de red

12) *solax_wtgrid*: Energía enviada a red

(1.1) 13) *solax_pwm*: Valor pwm (0-180)

(1.1) 14) *status0*: Estado de las salidas y conexiones (RL4|RL3|RL2|RL1|merror|werror|serror)

(1.1) 15) *status1*: Reservado

3.- MQTT

Los temas mqtt suscritos con la información aparte de los programables son los siguientes:

solax/pwm -> PWM Value (0-180)

solax/pv1c -> corriente string 1

solax/pv2c -> corriente string 2

solax/pv1v -> tension string 1

solax/pv2v -> tension string 2

solax/pw1 -> potencia string 1

solax/pw2 -> potencia string 2

solax/gridv -> tension de red

solax/wsolar -> Potencia solar

solax/wtoday -> Potencia solar diaria

solax/wgrid -> Potencia de red (Negativo: de red
- Positivo: a red)

solax/wtgrid -> Potencia enviada a red

OpenDS+ - Derivador de excedentes **versión 1.1.0**

Manual de uso

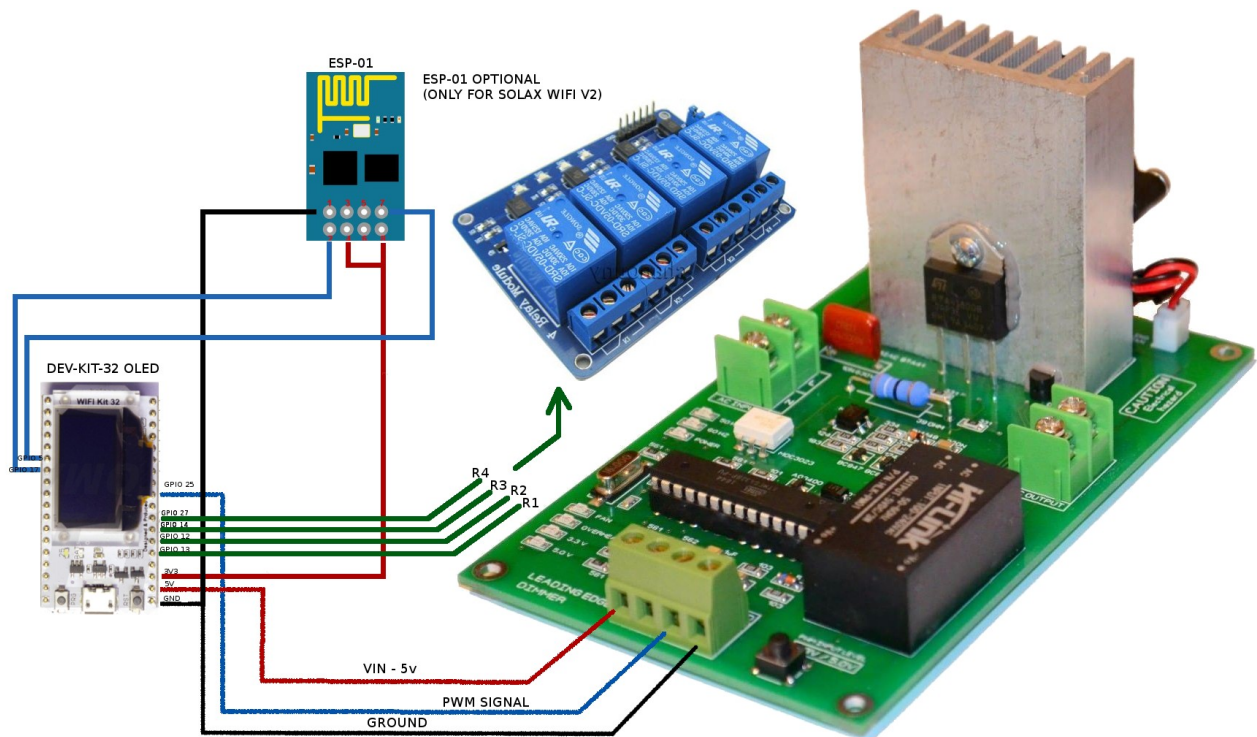
(1.1) Los valores pasan a ser los siguientes:

invert/pwm -> PWM Value (0-180)
invert/pv1c -> corriente string 1
invert/pv2c -> corriente string 2
invert/pv1v -> tension string 1
invert/pv2v -> tension string 2
invert/pw1 -> potencia string 1

invert/pw2 -> potencia string 2
invert/gridv -> tension de red
invert/wsolar -> Potencia solar
invert/wtoday -> Potencia solar diaria
invert/wgrid -> Potencia de red (Negativo: de red - Positivo: a red)
invert/wtograd -> Potencia enviada a red

7.- Conexionado

Para el conexionado seguiremos el esquema indicado a continuación para un Heltec Wifi Kit 32. para otros modelos como el ESP32, se deberá verificar los pines en el código fuente.



Manual de uso

Conexionado de la placa ESP32:

