

OpenDS+ - Derivador de excedentes

Manual de uso

1.- Que es OpenDS+

OpenDS+ es un derivador de excedentes orientado a los inversores Solax X1 Boost. Su función es aprovechar la potencia no usada derivandola a una carga resistiva como un termo o una estufa y mediante relés a cualquier tipo de carga. Con esto conseguimos aprovechar esa energía que es capaz de producir la instalación y la derivamos a otra carga puntualmente y de forma automática.

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.

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>

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:

- 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 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>

OpenDS+ - Derivador de excedentes

Manual de uso

Deberemos instalar estos Drivers para que Windows reconozca la electrónica.



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>

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 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 esp232_derivator.ino.bin

0x8000 esp232_derivator.ino.partitions.bin

b) ESP-01

OpenDS+ - Derivador de excedentes

Manual de uso

(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

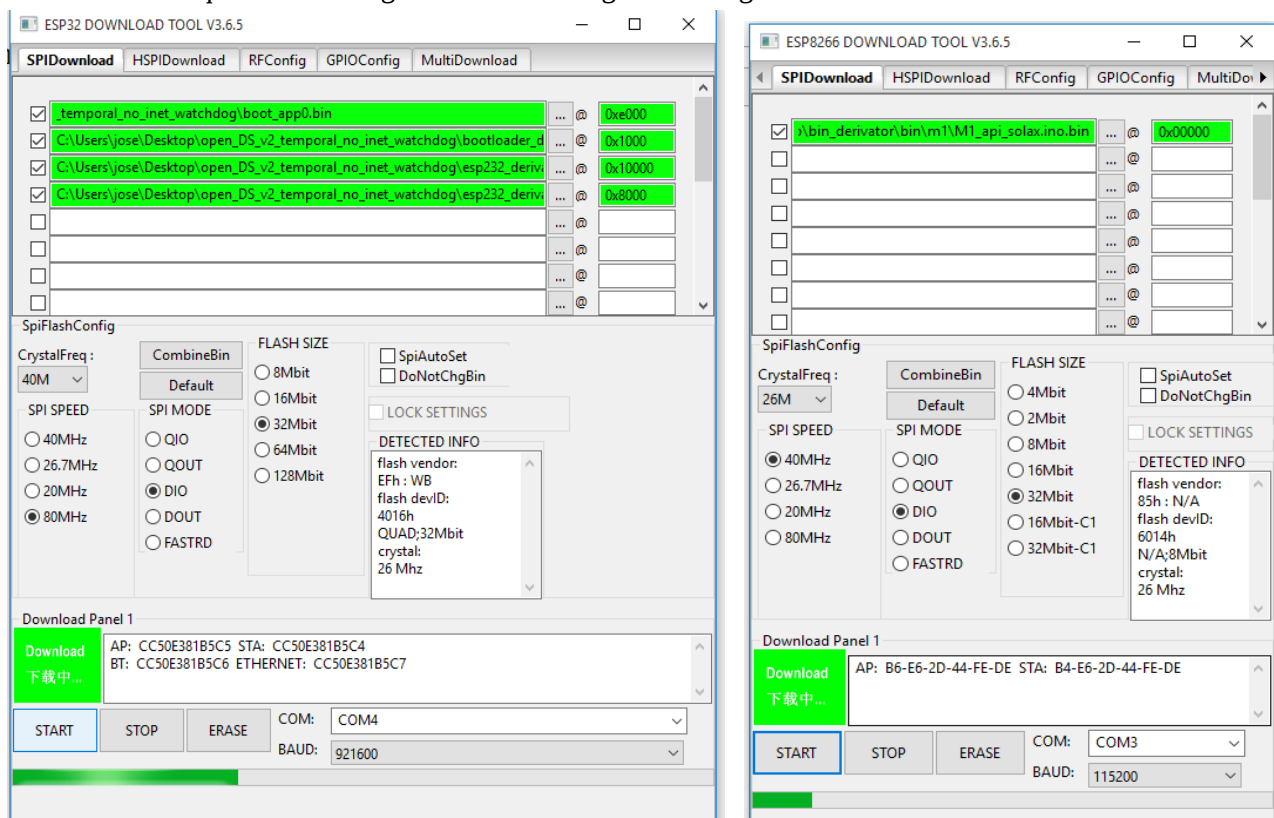
Modo de flash: dio

Flash: 512Kb

Frecuencia: 40Mhz.

0x00000 M1_api_solax.ino.bin

Con lo cual nos quedarían configurados como la siguiente imagen:



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.

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.

OpenDS+ - Derivador de excedentes

Manual de uso

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

Instalamos la siguiente app en el móvil:

- Para Android: ESP8266 SmartConfig
- Para IOS: Espressif Esptouch

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.

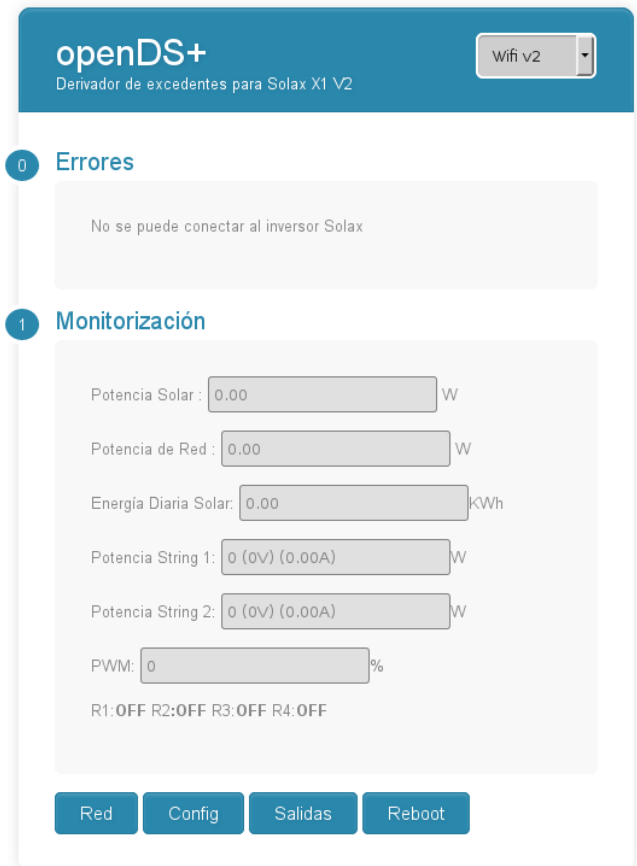
Nos mostrará los parámetros todos a cero, dado que no tiene conexión con el inversor.

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

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

- 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)

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



OpenDS+ - Derivador de excedentes

Manual de uso

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, la Wifi1 está ya con los datos del SmartConfig

En Datos Solax deberemos elegir el SSID del inversor Solax si disponemos del stick wifi v2 o bien poner la IP del solax si es la V1.

2 Datos Solax

SSID Wifi Solax (SOLAX)

Seleccione

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

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.

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.

openDS+ #Configuración Wifi

Derivador de excedentes para Solax X1

1 Datos conexion wifi

SSID Wifi 1 (LUCUS-1)

Seleccione

Wifi 1 Password

●●●●●●●●●●●●●●●●

SSID Wifi 2 (MIWIFI2)

Seleccione

Wifi 2 Password

●●●●●●●●●●

2 Datos Solax

IP Wifi V1 solax

192.168.90.15

3 Conexión de red

Dirección IP

192.168.3.99

Puerta de enlace

192.168.3.1

Máscara de red

255.255.255.0

Servidor de nombres 1

208.67.222.222

Servidor de nombres 2

208.67.220.220

Dirección DHCP ☒

Guardar

Cancelar

OpenDS+ - Derivador de excedentes

Manual de uso

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 (Testeado 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.

openDS+ #Configuración

Derivador de excedentes para Solax X1

1 Datos MQTT

MQTT Activo ☐

MQTT Broker

MQTT Usuario

MQTT Password

MQTT Puerto

Tema R1

Tema R2

Tema R3

Tema R4

OpenDS+ - Derivador de excedentes

Manual de uso

Salidas:

En este menú configuramos las cargas del dispositivo, que pueden ser enlaces físicos 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.

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

Guardar

Cancelar

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

Guardar

Cancelar

3

Configuración Manual

Relé 1 Activo ☐ Relé 2 Activo ☐

Relé 3 Activo ☐ Relé 4 Activo ☐

Guardar

Cancelar

openDS+ versión 1.0 - <https://github.com/iqas/derivador>

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, 1-Wifi V1, 2-Wifi V2 con ESP01, 3-Wifi V2 con ESP01 firm M1

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

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_wtograd: Energía 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.

