**Sonoff S20 hack.**

Sonoff S20 es un es un interruptor con control remoto por wifi.

Es la evolución del Sonoff Wi-Fi Smart link.

El S20 no necesita montaje adicional. Viene con un enchufe macho y uno hembra.

Un pulsador y dos leds. Uno verde y uno azul.

En su interior podemos encontrar un pequeño transformador, un relé para controlar la salida de alto voltaje, y un ESP8266 como corazón de la placa.

No tendremos en cuenta el software original que contiene. Lo que nos interesa es crear nuestros propios programas de gestión.

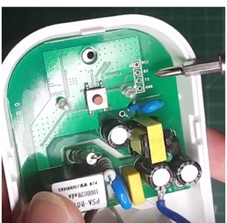
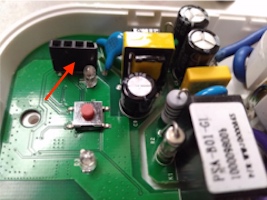
 

Lo primero que debemos hacer es quitar la pegatina de garantía que cubre el agujero con el único tornillo exterior del dispositivo.

Cuando lo abrimos encontramos todos los componentes.

Identificamos el interruptor/pulsador, que tiene dos leds a sus lados, etiquetados como D1 y D3.

Al lado del led D1 tenemos 4 agujeros correspondientes a los pines de comunicación y alimentación.

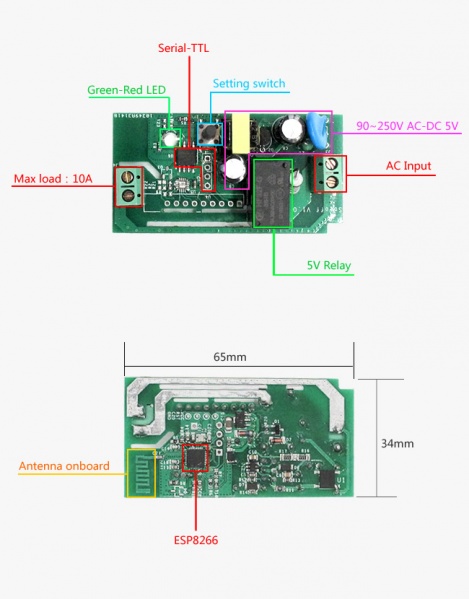
 

https://diyprojects.io/hack-connected-sonoff-s20-plug-esp-easy-rules/

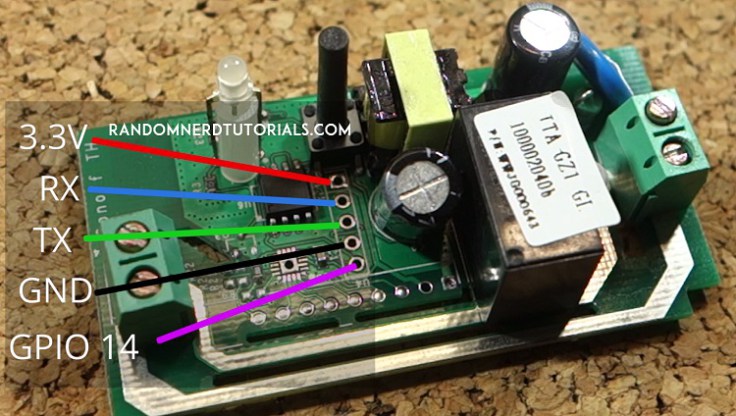
Material necesario: dispositivo sonoff, destornillador, soldador, peine de 4 pines, FTDI, cables.

Quitaremos la carcasa para dejar la placa base a la vista.





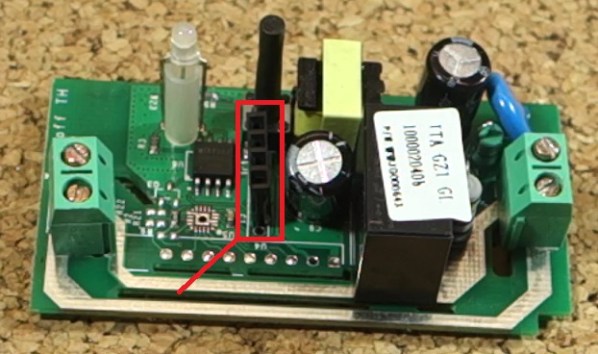
En la parte superior de la placa base encontraremos 5 agujeros preparados para poder soldar pines.

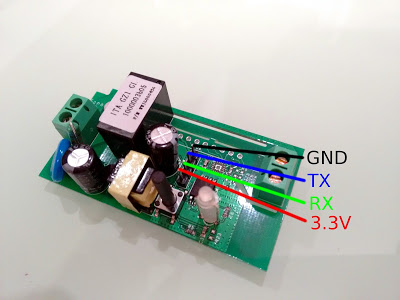


El agujero para el GPIO14 no lo necesitaremos para transferir, pero los otros 4 sí.

Los dos de alimentación (3.3V y GND) y los dos de comunicación serie (RX y TX).

Utilizaremos un “peine” (da igual si macho o hembra) de 4 pines que soldaremos por la parte inferior sobre los agujeros indicados.

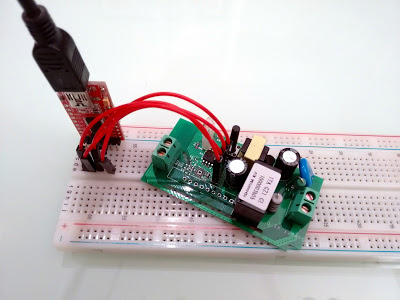




Necesitaremos un FTDI para convertir una salida USB en un puerto serie

Las conexiones serán las siguientes:

|  |  |
| --- | --- |
| **Sonoff** | **FTDI** |
| 3.3V | 3.3V |
| RX | TX |
| TX | RX |
| GND | GND |



Es importante borrar totalmente la memoria del dispositivo. Si no lo hacemos es posible que nuestros desarrollos no se comporten correctamente.

Para ello utilizaremos un script de Python. Por lo tanto necesitaremos tener instalado Python y las herramientas del esp (esptools).

Estas herramientas se instalan desde la línea de comandos:

python -m pip install esptool

Así se descargarán las librerías necesarias.

Sabiendo cuál es el puerto COM al que tenemos conectado el FTDI ejecutamos la siguiente línea:

d:\programas\python27\lib\site-packages\esptool.py --port COM12 erase\_flash

El path de dónde se encuentra la librería lo obtenemos de la instalación

El COM12 debe ser sustituido por el COM correspondiente.

Tarda algo menos de 10 segundos en borrar la memoria.

En el IDE de Arduino se debe configurar de la siguiente manera:

Placa: “Generic ESP8266 Module”

Flash Mode: “DIO”

Flash Frequency: “40MHz”

Upload Using: “Serial”

CPU Frequency: “80 MHz”

Flash Size: “1m(64K SPIFFS)”

Reset Method: “ck”

Upload Speed: “115200”

Los GPIOs disponibles son los siguientes:

|  |  |
| --- | --- |
| **GPIO** | **Función** |
| 0 | Interruptor (pulsador) |
| 1 | TX |
| 3 | RX |
| 12 | Relé |
| 13 | Led |