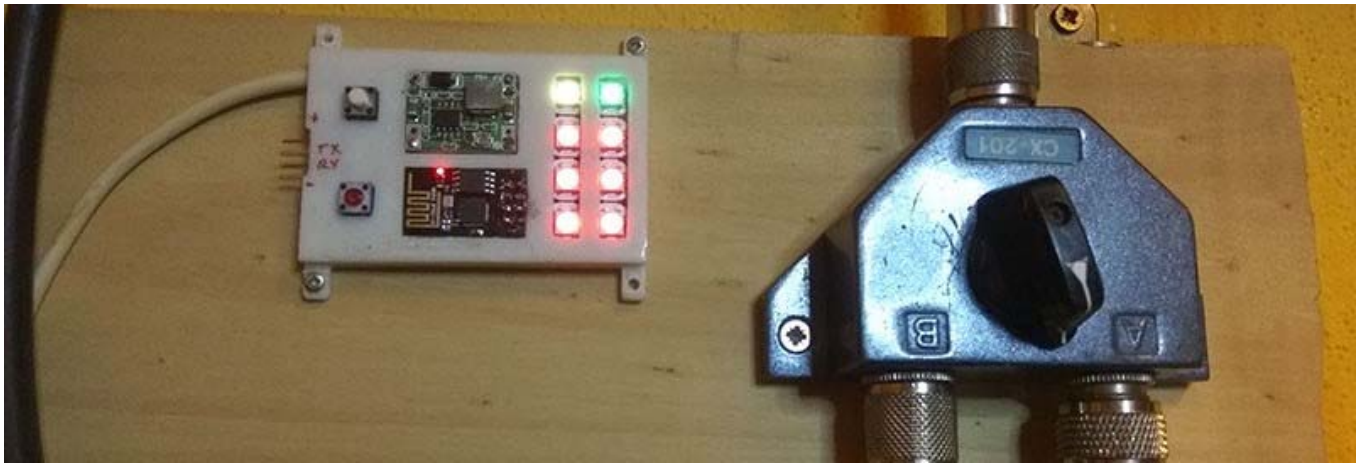
ESP8266 SolarData-RGB



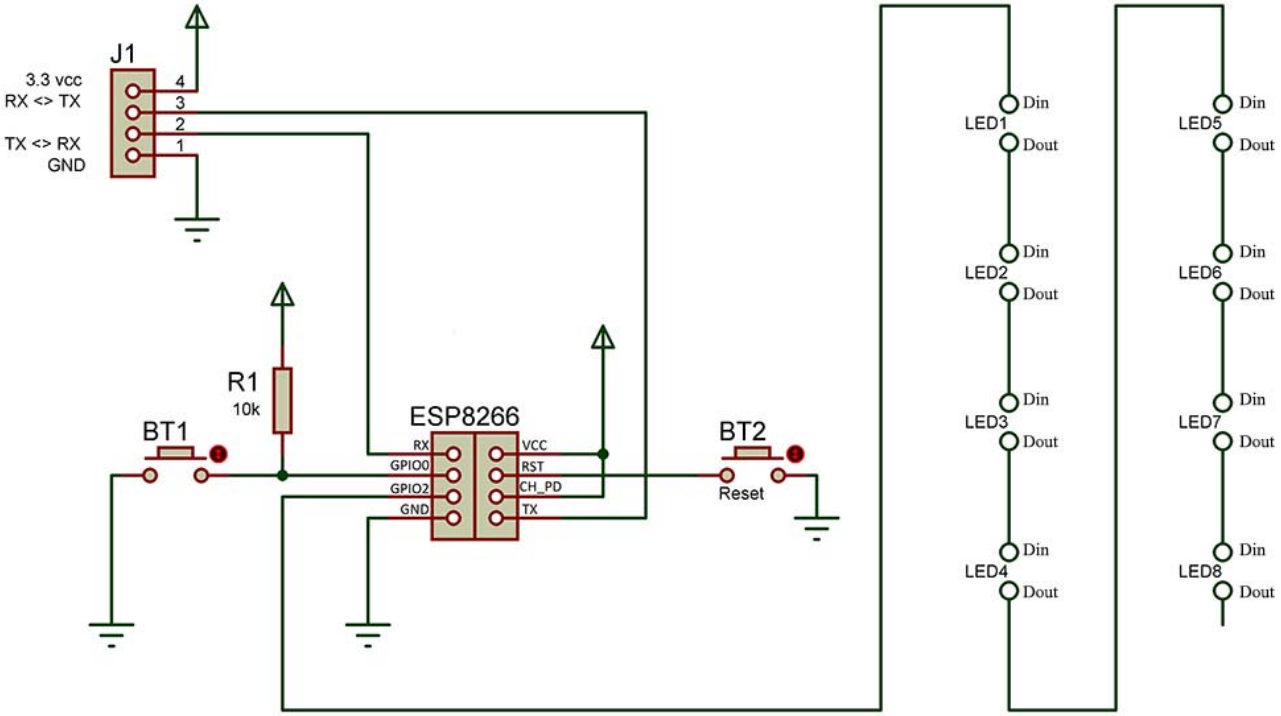
Se trata de un sencillo circuito que nos muestra en una tira LED WS2812B, por medio de colores, el estado de las bandas HF de radioaficionados, obteniendo dicha información desde la página web hamqsl.com.

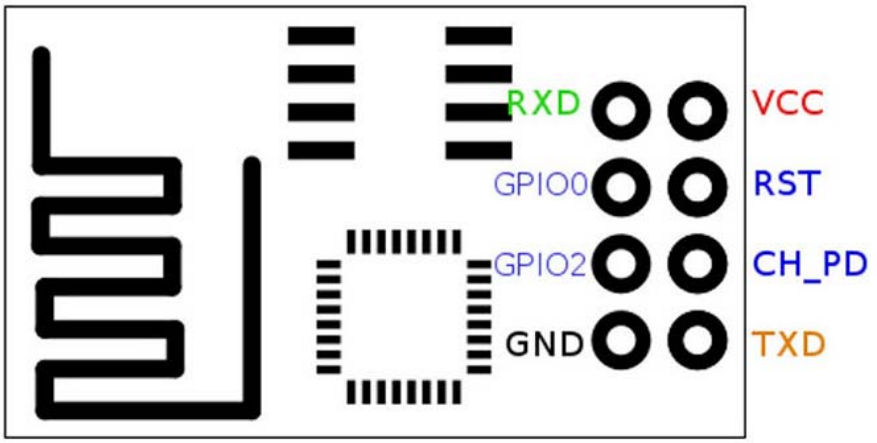
Para el control y conexión a la red wifi, se utiliza el módulo ESP8266, utilizado en este circuito el primer modelo ESP-01, que es el mas simple y pequeño de la serie pero suficiente para este proyecto.

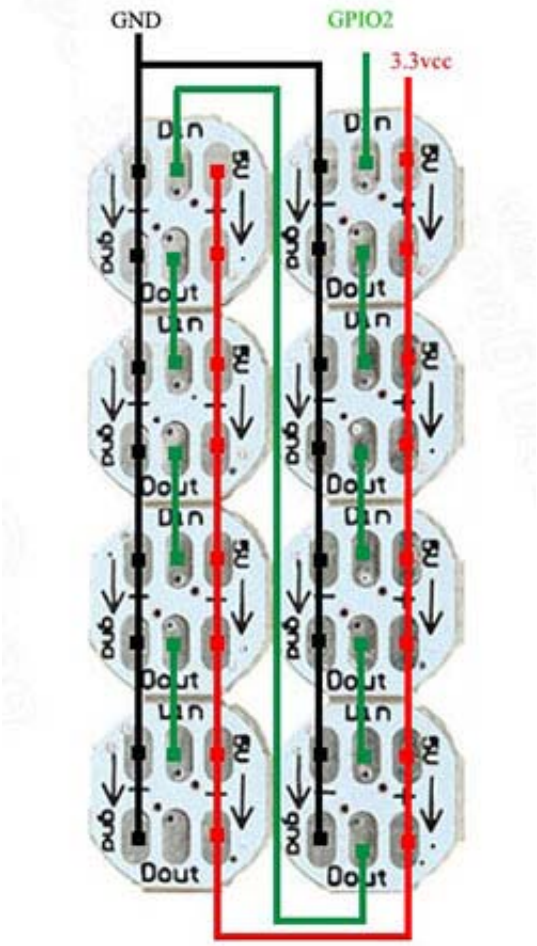
Material

- Modulo ESP8266 ESP-01.
- 8 LEDs RGB WS2812, tipo neopixel o similares.
- 2 pulsadores NA (normalmente abiertos)
- 1 resistencia de 10K.
- Conversor USB a TTL de 3.3v (solo para la programación)
- Fuente de alimentación de 3.3v para su instalación definitiva.

Circuito


J1
3.3 vcc
RX <> TX
TX <> RX
GND
4
3
2
1
R1
10k
BT1
ESP8266
RX
GPIO0
GPIO2
GND
VCC
RST
CH_PD
TX
BT2
Reset
LED1
LED2
LED3
LED4
LED5
LED6
LED7
LED8
Din
Dout


RXD
GPIO0
GPIO2
GND
VCC
RST
CH_PD
TXD



Programación

1. Instalar la última versión de Arduino.
2. Instalar la tarjeta del ESP8266.
   - Abrir la aplicación de aplicación de Arduino e ir a Preferencias desde el menú Archivo.
   - Introducir `http://arduino.esp8266.com/stable/package_esp8266com_index.json` en "Gestor de URLs Adicionales de Tarjetas" (puede introducir varias direcciones separandolas por comas ',').
   - Abrir *Gestor de tarjetas* desde el menú Herramientas/Placa. Buscamos por ESP8266 e instalamos.
   - Podéis encontrar mas información u otros modos de instalación en su página de GitHub.
3. Instalación de la librería que controla los LED's.
   - Desde el menú Programa/Incluir Librería, abrir *Gestionar Librerías*. Buscar e instalar la librería llamada **Adafruit NeoPixel** by Adafruit.
4. Seleccionar el módulo *Generic ESP8266 Module* para el ESP-01; seleccionar el puerto COM de su conversor USB-TTL; en *Upload speed* seleccionar 115200.
5. Para subir el programa al ESP8266 ESP-01, primero debemos iniciarlo en modo programación, para ello, siguiendo el circuito propuesto:
   - Conectar la alimentación, Tx y Rx a nuestro USB-TTL **(importante que éste sea de 3.3v, tanto alimentación como bus de datos)**.
   - Pulsar y mantener pulsado el pulsador de reset "BT2".
   - Pulsar y mantener pulsado el pulsador 1 "BT1".

- Soltar pulsador de reset "BT2".
- Esperar 2 segundos y soltar el pulsador 1 "BT1".
- Subir el programa desde la aplicación de Arduino.

Se utiliza la librería ***WiFiManager*** incluida directamente en el proyecto, por lo que no hace falta sea instalada. Se ha realizado de este modo ya que ésta es la que nos crea la web de configuración que por defecto viene en inglés. La incluida, esta modificada para que la web esté en español.

Para mas información sobre ***WiFiManager***, acceder a su página de GitHub.

Funcionamiento y puesta en marcha.

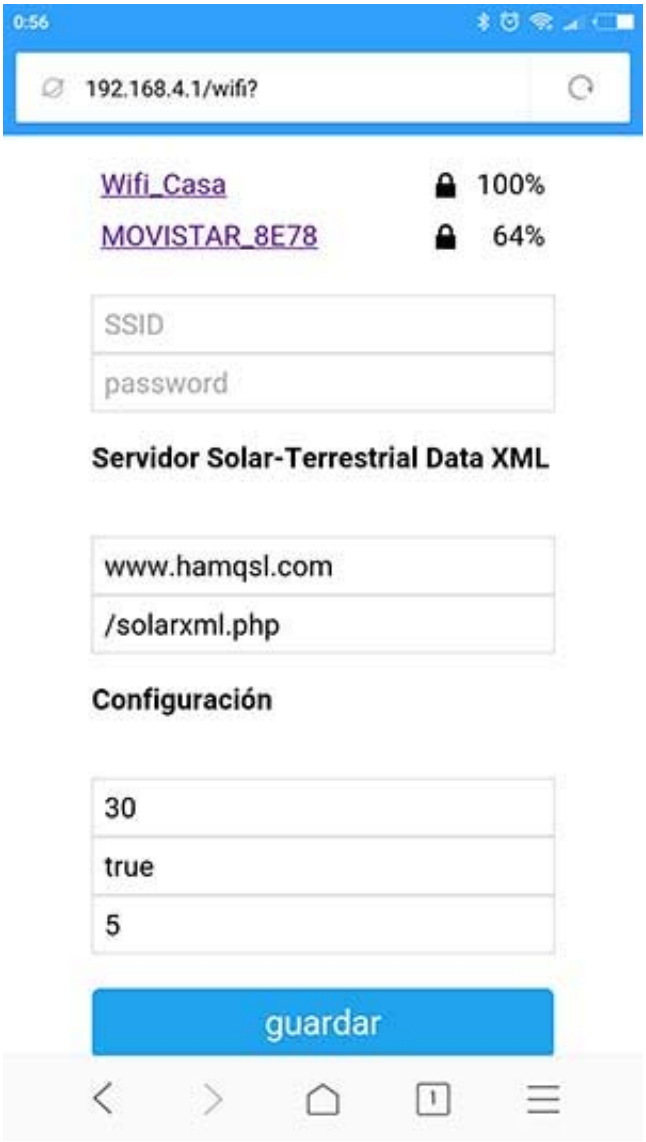
Una vez subido el *sketch* a nuestro ESP8266 y si todo ha ido bien, deberíamos ver como se iluminan todos los leds en azul durante unos segundos pasando a continuación a un color morado.

El color morado, significa que nuestro circuito a entrado en modo AP (*"Acces Point" o "Punto de Acceso"*), creando una red Wifi propia con el nombre AP_SolarData.

Condigurar nuestra red wifi y otras opciones.

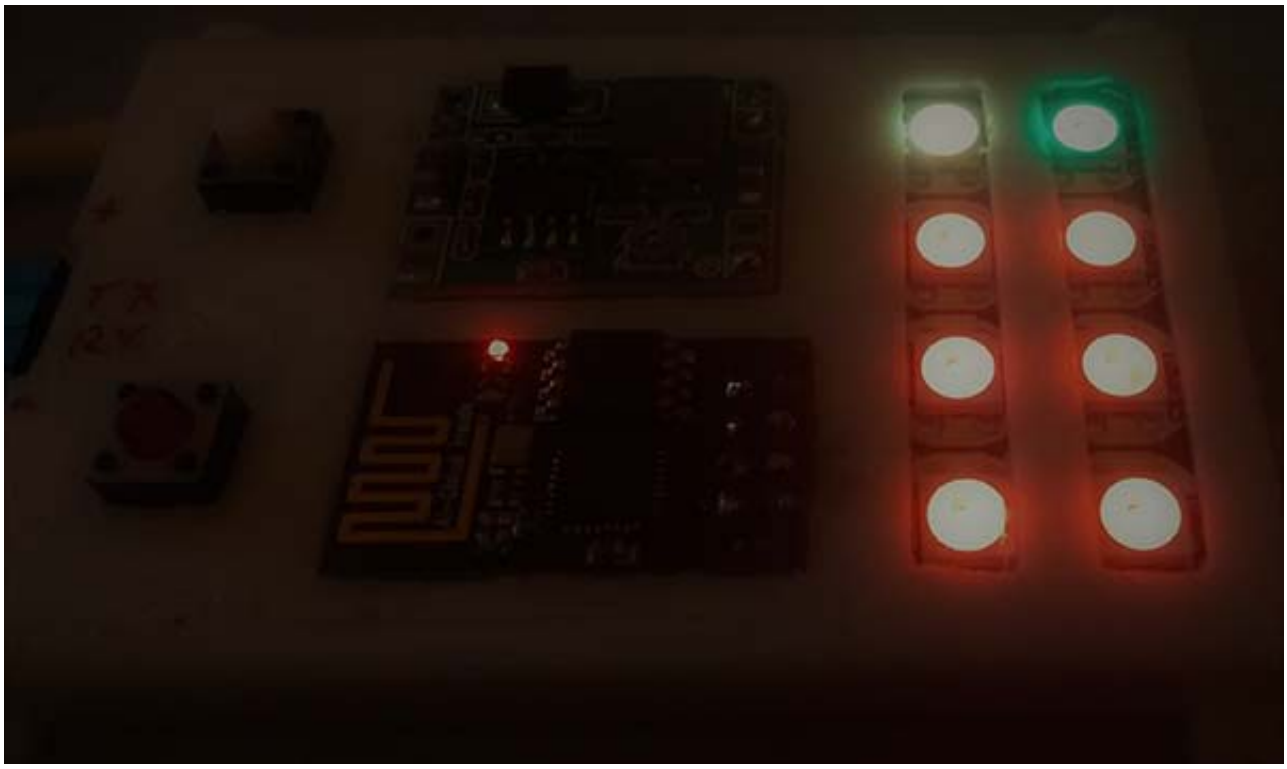
- Conectarse con una tablet, pc, teléfono o similar a la red wifi creada *AP_SolarData*, es una red abierta.
- Abrir un navegador web, en caso de no redireccionarnos directamente, acceder a la dirección 192.168.4.1
- En la siguiente pantalla, pulsar en *Configurar WIFI*.

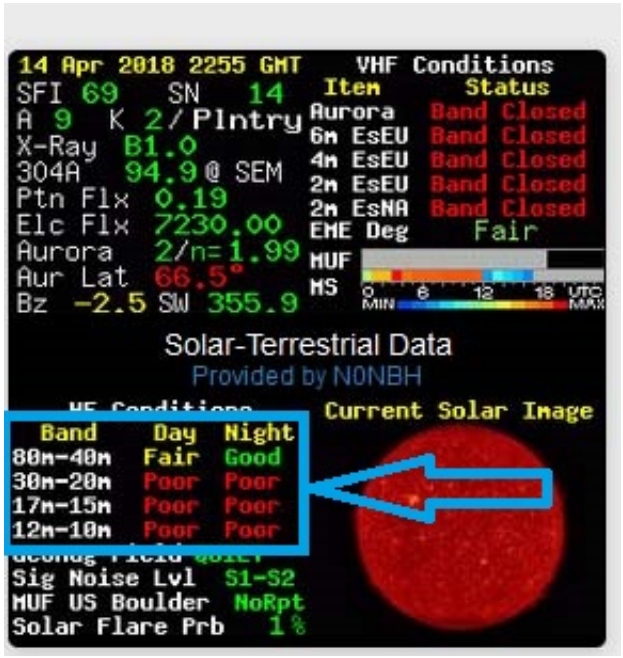
- Nos aparecerá una página como la siguiente:



- En el listado, pulsar sobre nuestra red e introducir la contraseña de acceso.
- El apartado *Servidor Solar-Terrestrial Data XML* son el host y la url de acceso al XML que nos sirve los datos, no deberían modificarse salvo que cambie la ubicación del archivo.
- En el apartado *Configuración*, podemos seleccionar algunos parámetros a nuestro gusto:
  - El Primero (30) es la intensidad de iluminación de los LEDs, podemos poner un valor comprendido entre 0 y 255.
  - El segundo nos permite escribir *true* o *false*
    - **true**: Se apagarán los LEDs transcurridos unos segundos. Pulsando el "BT2" se realiza una nueva solicitud de datos a la web y reilumina los leds según correspondan.
    - **false**: Se mantienen los LEDs encendidos. Pulsando el "BT2" se realiza una nueva solicitud de datos a la web y reilumina los leds según correspondan.
    - En cualquier caso, el módulo se desconecta de la red y pasa a bajo consumo una vez cargados los datos.
  - El tercer apartado nos permite configurar (en caso de ser true el apartado anterior) los segundos que se mantendrán los LED's iluminados.
- Pulsar sobre *guardar* y si todo ha ido bien nuestro ESP debería reiniciarse, conectarse a nuestra wifi y mostrar los colores según corresponda. Si la iluminación de los LEDs sigue siendo morada, significa que no ha conseguido conectarse a nuestra wifi, por lo que habrá

entrado en modo AP para volver a configurar.





Cambiar configuración.

Una vez nuestro ESP tiene los datos guardados y consigue entrar a nuestra wifi, éste no vuelve a ponerse en modo AP, solo se activa este modo si no es capaz de conectarse a nuestra wifi o si reiniciamos los datos.

Para hacer un reinicio de datos para volver a entrar en modo configuración hay que:

- Reiniciar el ESP con el pulsador 2 "BT2" y soltarlo.
- Cuando la iluminación está en AZUL, pulsar el BT1.

Ahora debería iluminarse en morado, ya está listo para volver a configurar desde la wifi *AP_SolarData*.

Iluminación AZUL es una ventana de 2,5 segundos que nos permite reiniciar los valores a nuestro ESP pulsando el BT1

Mas información en la web de GitHub: https://github.com/DonJaume/ESP8266_SolarData_RGB