

LoRaGWIFI

Waterway monocal LoRa



Documentación, esquemas, librerías, ejemplos, etc.:



<https://github.com/raymirabel/LoRaGWIFI>

ÍNDICE DE CONTENIDO

1. ¿Qué es LoRa?	3
2. Características principales.	5
3. Conexiones.	6
4. Microcontrolador.....	7
5. Instalación Arduino Core para ESP8266.....	8
6. Programación.....	11
7. Línea de comandos.....	13
8. Especificaciones técnicas generales.	16
9. Revisiones.	17

1. ¿Qué es LoRa?

LoRa es una tecnología que utiliza un tipo de modulación en frecuencia, como la AM o la FM, pero patentado por Semtech, una importante empresa fabricante de chips de radio.

Esta tecnología de modulación se llama Chirp Spread Spectrum, o CSS, y se usa en comunicaciones militares y espaciales desde hace décadas. La gran ventaja de la misma es que puede lograr comunicaciones a largas distancias (típicamente kilómetros) y tiene gran solidez frente a las interferencias.

Los dispositivos LoRa, se han convertido en una de las principales tecnologías inalámbricas de largo alcance y bajo consumo utilizadas por el “Internet de las cosas” (IoT). La tecnología LoRa permite crear aplicaciones inteligentes de IoT que ayudan a afrontar problemas de la actualidad, como pueden ser la gestión energética, el control de la polución, o la prevención de desastres naturales.

Ventajas

Entre las principales ventajas de LoRa se encuentran las siguientes características:

- Alta tolerancia a las interferencias.
- Alta sensibilidad para recibir datos (-168dB).
- Basado en modulación “chirp”.
- Bajo Consumo (hasta 10 años con una batería).
- Largo alcance 1 a 20km.
- Baja transferencia de datos (hasta 255 bytes).
- Conexión punto a punto (mesh y enrutador se pueden implementar por software).

Banda ISM

LoRa utiliza el espectro de frecuencia sin licencia como parte de la banda de radio ISM (Industrial, Científica y Médica). Estas bandas de radio están reservadas internacionalmente para el uso de la energía de radiofrecuencia para fines industriales, científicos y médicos distintos de las telecomunicaciones.

En Europa se utiliza un plan de frecuencias sin licencia en torno a los 868Mhz, mientras que en Estados Unidos es 915Mhz y en Asia 433Mhz.

El uso de las bandas sin licencia del espectro radioeléctrico, hace que sea fácil para cualquiera configurar su propia red y hacer uso de ella.

Para utilizar la tecnología LoRa con el fin de comunicar dispositivos IoT existen dos alternativas:

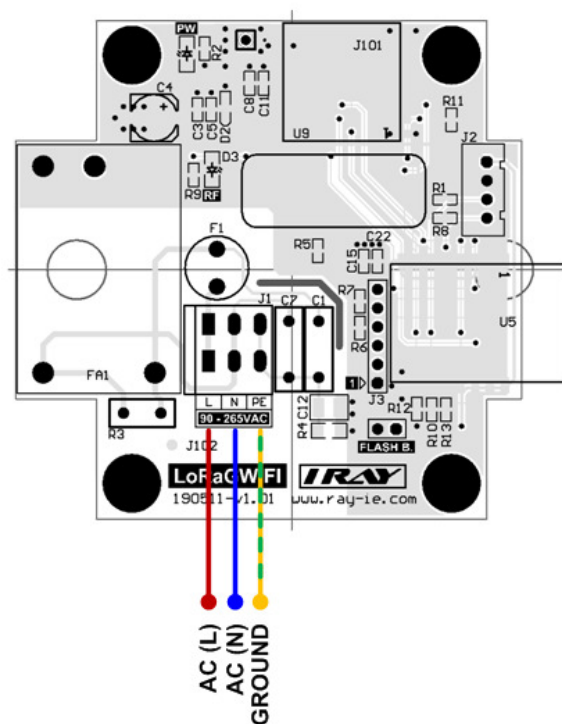
1. Crear una red propia. Será necesario comprar los chips LoRa y desarrollar unos gateways y nodos propios. Como ventaja, el usuario podrá ajustar y adaptar dicha red a sus necesidades, aunque deberá encargarse del mantenimiento de la misma.
2. Usar un operador de red. Actualmente, varios operadores están empezando a ofrecer redes LoRaWAN en ciertas áreas por ejemplo <https://www.thethingsnetwork.org/>. De este modo, pueden contratarse planes de conectividad con estas compañías evitando así el mantenimiento de la red. Esta opción tiene como desventaja que el operador puede dejar de ofertar este servicio.

Actualmente, nuestros dispositivos soportan LoRa para redes privadas y LoRaWAN. Para ello deber cargar el firmware adecuado en cada caso de uso. Dicho firmware está a disposición del usuario para su uso pudiendo modificarlo a sus necesidades. Los dispositivos son fácilmente reprogramables ya que están basados en arquitectura Arduino y por lo tanto editarse desde cualquier entorno o IDE de Arduino.

2. Características principales.

- Microcontrolador ESP8266-ESP12 compatible con **NodeMCU**.
- Conectividad WIFI.
- Compatible LoRa/LoRaWAN 868Mhz (RFM95).
- Alimentación 90-265VAC 50-60Hz
- Caja policarbonato IP67

3. Conexiones.



Para introducir o sacar el cable de la borna, utilice un destornillador plano y presione sobre la tecla de la siguiente forma:



4. Microcontrolador.

LoRaGWIFI se entrega con un firmware ejemplo estándar de funcionamiento. Dicho firmware recibe paquetes LoRa de sensores y los envía a un servidor mediante la conectividad WiFi.

Hay disponible una librería y ejemplos para facilitar la programación si desea modificar o mejorar el firmware actual. Puede encontrar el repositorio de librerías, documentación, manuales, etc, en este enlace:

<https://github.com/raymirabel/LoRaGWIFI.git>

Podemos utilizar cualquiera de los entornos de desarrollo Arduino para modificar el firmware.

La equivalencia entre las E/S de LoRaGWIFI y el procesador ESP8266 son las siguientes:

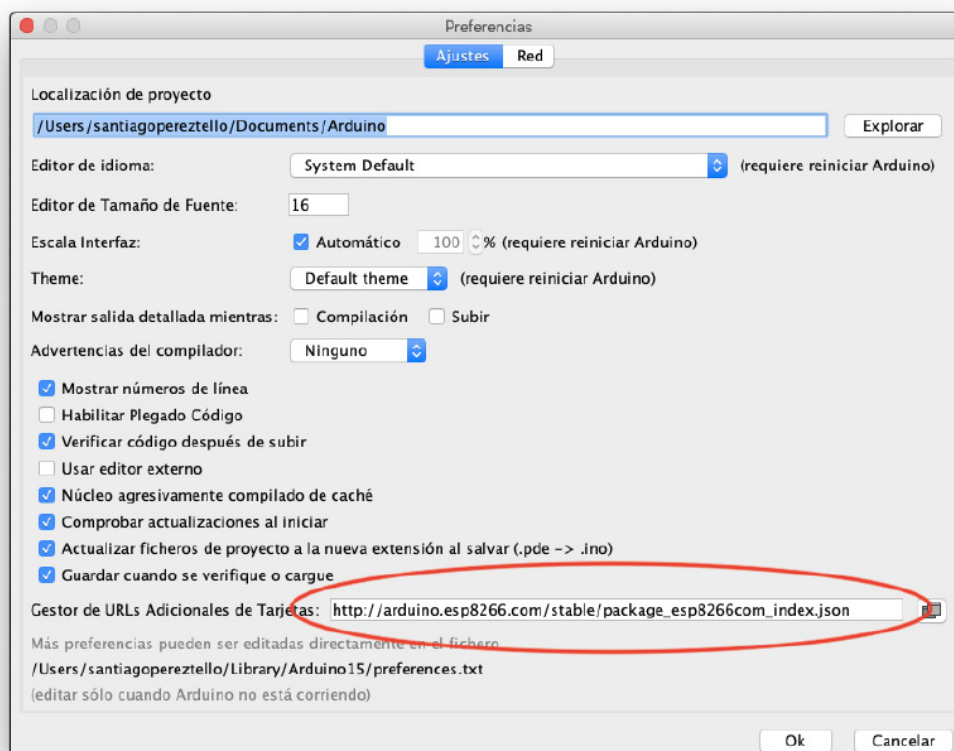
E/S LoRaGWIFI	E/S ESP8266
ESP8266 FLASH BOOT	0
GPIO0 - IRQ RFM95	4
GPIO1 - RFM95	5
CS RFM95	15
LED TEST	16

5. Instalación Arduino Core para ESP8266.

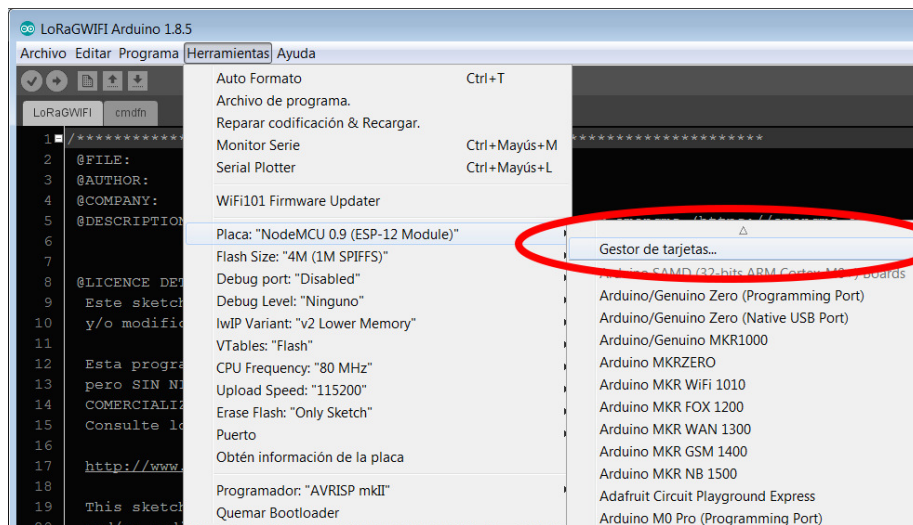
El primer paso que tendremos que realizar antes de programar nuestro LoRaGWIFI será instalar en el gestor de tarjetas de nuestro IDE Arduino el “Arduino core for ESP8266 wifi chip” que básicamente nos permitirá configurar los parámetros y seleccionar entre las distintas placas que existen con el chip ESP8266.

Para ello abrimos las preferencias de nuestro entorno de Arduino y establecemos el siguiente enlace como se muestra en la siguiente imagen:

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

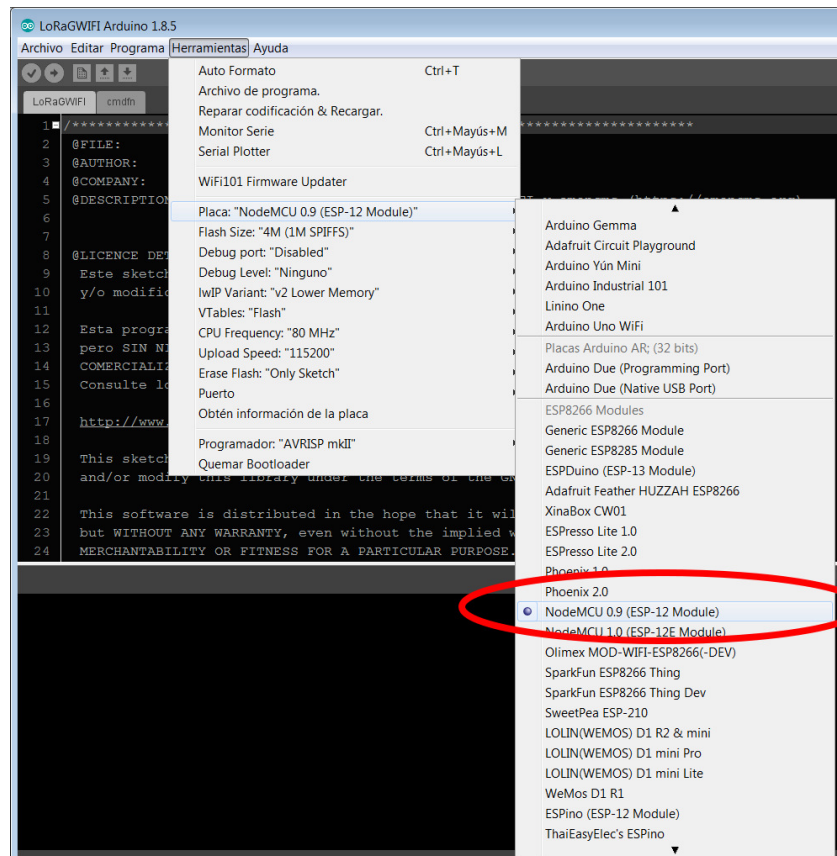


Presionamos “Ok”. Tras este paso accedemos al gestor de tarjetas de Arduino y buscamos “ESP8266”:



Instalamos “esp8266 by ESP8266 Community”, la versión más reciente.

Una vez instalado el core ESP8266 para Arduino seleccione la tarjeta NodeMCU 0.9 (ESP-12 MODULE):



6. Programación.

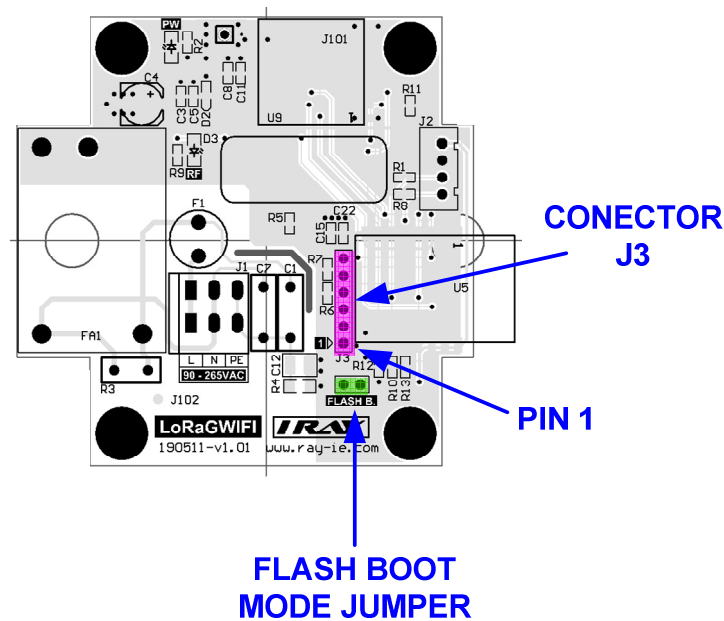
(*) NOTA IMPORTANTE: Asegúrese que tiene seleccionado, dentro del entorno de desarrollo, como placa el modelo "NodeMCU".

Para la programación de nuestro LoRaGWIFI usaremos un cable USB comercial con salida serie TTL. Nosotros recomendamos, el modelo TL-232R-5V de la firma FTDI:



El pinout del conector J3 es el siguiente:

PIN	SEÑAL	COLOR
1	GND	NEGRO
2	CTS (NO UTILIZADO)	MARRON
3	NO UTILIZADO	ROJO
4	TXD (RX del microcontrolador)	NARANJA
5	RXD (TX del microcontrolador)	AMARILLO
6	RTS (RESET del microcontrolador)	VERDE



(*) **NOTA IMPORTANTE:** Antes de conectar el cable de programación, asegúrese que tiene bien posicionado el conector: PIN1 = GND = CABLE COLOR NEGRO.

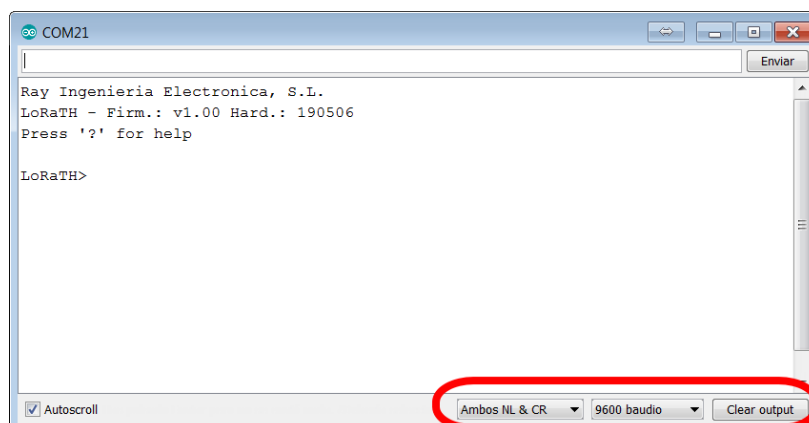
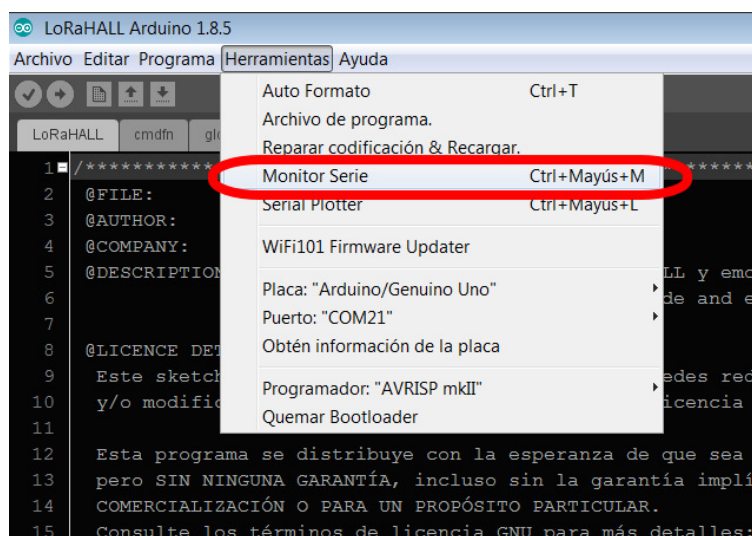
Antes de iniciar la carga del programa debe poner el microcontrolador en modo FLASH BOOT. Para ello conecte el jumper FLASH BOOT MODE antes de encender el gateway. Una vez encendido el equipo puede proceder con la carga del programa. No olvide desconectar el jumper una vez finalizada la carga.

7. Línea de comandos.

LoRaGWIFI se entrega con un firmware ejemplo de estándar de funcionamiento llamado "**LoRa SERVER RAW**". Este ejemplo recibe paquetes LoRa cuando un sensor transmite una estructura de datos donde se encuentran los parámetros de las medidas del sensor como distancia, temperatura, humedad, luz, nivel de batería, etc. Una vez recibidos los datos estos son transmitidos inmediatamente al servidor de datos [emoncms](#).

El firmware "**LoRa NODE RAW**" incluye una línea de comandos para configurar parámetros y consultar estados.

Para acceder a la línea de comandos debe conectar un cable compatible SERIE-TTL al conector J3 de LoRaGWIFI y configurar el terminal serie a 9600,N,8,1. Consulte el apartado 6, programación, para más información de cómo conectar un cable serie TTL. Puede usar como terminal serie el que incluye el IDE de Arduino:



Al iniciar el terminal serie aparecerá el siguiente mensaje de bienvenida donde nos indica la versión del firmware y hardware:

```
Ray Ingenieria Electronica, S.L.  
LoRaGWIFI - Firm.: v1.00 Hard.: 190511  
Press '?' for help
```

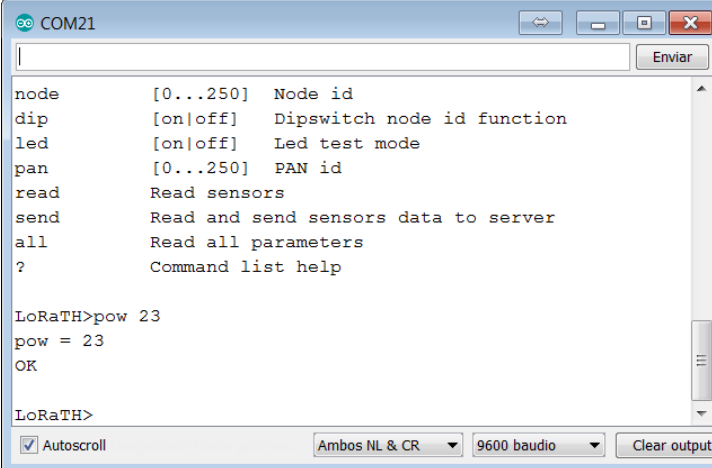
Pulse ? para consultar la lista de comandos:

ssid	[32 char]	Wifi SSID
pass	[32 char]	Wifi password
key	[32 char]	Emoncms APIKEY
pow	[5...23]	RF power (5:min...23:max)
pan	[0...250]	PAN id
clear		Clear Wifi configuration
sta		Read Wifi status
all		Read all parameters
?		Command list help

A continuación detallamos cada comando:

- **ssid:** establece el parámetro SSID para la conexión a una red WIFI. Es necesario reiniciar el equipo para que los cambios tomen efecto.
- **pass:** establece la contraseña para la conexión a una red WIFI. Es necesario reiniciar el equipo para que los cambios tomen efecto.
- **key:** establece la clave APIKEY del servidor emoncms.
- **pow:** establece la potencia de transmisión LoRa.
- **pan:** identificador PAN o red de área personal. Solo los nodos y servidor con el mismo identificador PAN comunicarán entre sí. Este identificador es útil para separar redes dentro de un mismo área.
- **clear:** borra los parámetros ssid, pass y key.
- **sta:** lee el estado de la conexión wifi.
- **all:** lee y muestra todos los valores de la lista de comandos

Para modificar un valor de un comando escriba el comando y el nuevo valor. Por ejemplo, para modificar la potencia de transmisión al máximo escriba **pow 23** (y retorno de carro):



```
node      [0...250]  Node id
dip        [on|off]  Dipswitch node id function
led        [on|off]  Led test mode
pan        [0...250]  PAN id
read       Read sensors
send       Read and send sensors data to server
all        Read all parameters
?          Command list help

LoRaTH>pow 23
pow = 23
OK

LoRaTH>
```

COM21

Enviar

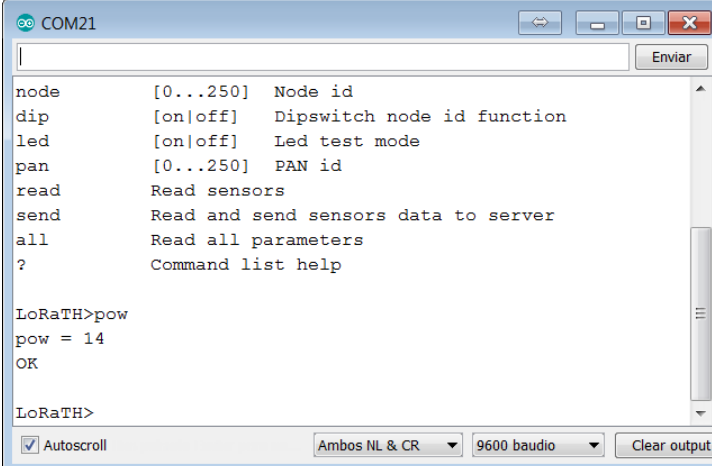
☒ Autoscroll

Ambos NL & CR

9600 baudio

Clear output

Para consultar un valor de un comando escriba el comando y pulse retorno de carro.



```
node      [0...250]  Node id
dip        [on|off]  Dipswitch node id function
led        [on|off]  Led test mode
pan        [0...250]  PAN id
read       Read sensors
send       Read and send sensors data to server
all        Read all parameters
?          Command list help

LoRaTH>pow
pow = 14
OK

LoRaTH>
```

COM21

Enviar

☒ Autoscroll

Ambos NL & CR

9600 baudio

Clear output

8. Especificaciones técnicas generales.

- Alimentación: 90-265VAC, 50-60Hz
- Consumo máximo: 4W
- Microcontrolador: ESP8266 ESP-12
- Transceiver LoRa: RFM95
- Temperatura de funcionamiento: -10 ~ 60 °C
- Grado de protección IP: IP67
- Ancho: 85 mm
- Alto: 85 mm
- Fondo: 55 mm
- Peso : 230 g.

9. Revisiones.

FECHA	VERSION	NOTAS
01/08/2019	1.00	- Documento inicial.