rhomb.io

Tutorial, manual de uso y ejemplos: RFM69 Rhomb.io Slave Module

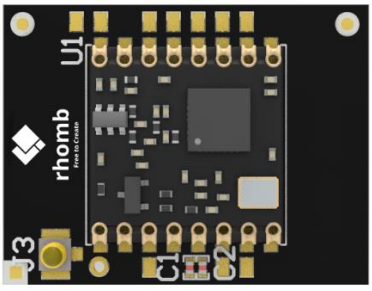
15/05/2018

Fecha	15/05/2018
Autor	Rubén Mir
Versión	1.0
Revisado por	Rubén Mir

TABLA DE CONTENIDO

1. INTRODUCCIÓN

El RFM69 es un Rhomb.io Slave Module basado en el integrado RFM69CW. Se trata de un módulo capaz de transmitir o recibir datos por radio frecuencia. Puede actuar tanto como emisor como receptor por lo que se trata de un transceptor. Puede trabajar en un amplio rango de frecuencias incluyendo 315,433, 868 y 915MHz. Los principales parámetros RF son programables. Está optimizado para un bajo consumo de energía al mismo tiempo que ofrece un alto nivel de potencia en la señal RF.



RFM69 Rhomb.io Slave Module

Entre sus posibles aplicaciones se encuentran:

- Automatizar lecturas de datos
- Redes de sensores inalámbricos

- Domótica
- Alarmas y sistemas de seguridad
- Monitorización y control industrial
- Bus de datos inalámbrico

2. REQUISITOS

2.1. HARDWARE

- 2x PCB de clase 2 o clase 3
- 2x Duino uno Rhomb.io master module (o bien 2x Duino Mega)
- 2x RFM69 Rhomb.io slave modules
- 2x antena (opcionalmente puede emplearse 2 cables a modo de antena)
- 2x Cable micro USB
- 2x Equipos PC (o bien 1x PC y 1x transformador con cable micro USB)

2.2. SOFTWARE

- Arduino IDE
- Librería RadioHead-master
- Sketch: rf69_server_gui_01.ino
- Sketch: rf69_client_gui_01.ino

3. CONEXIONES

Por favor, revisar el siguiente documento para más información:

- Inserción de módulos

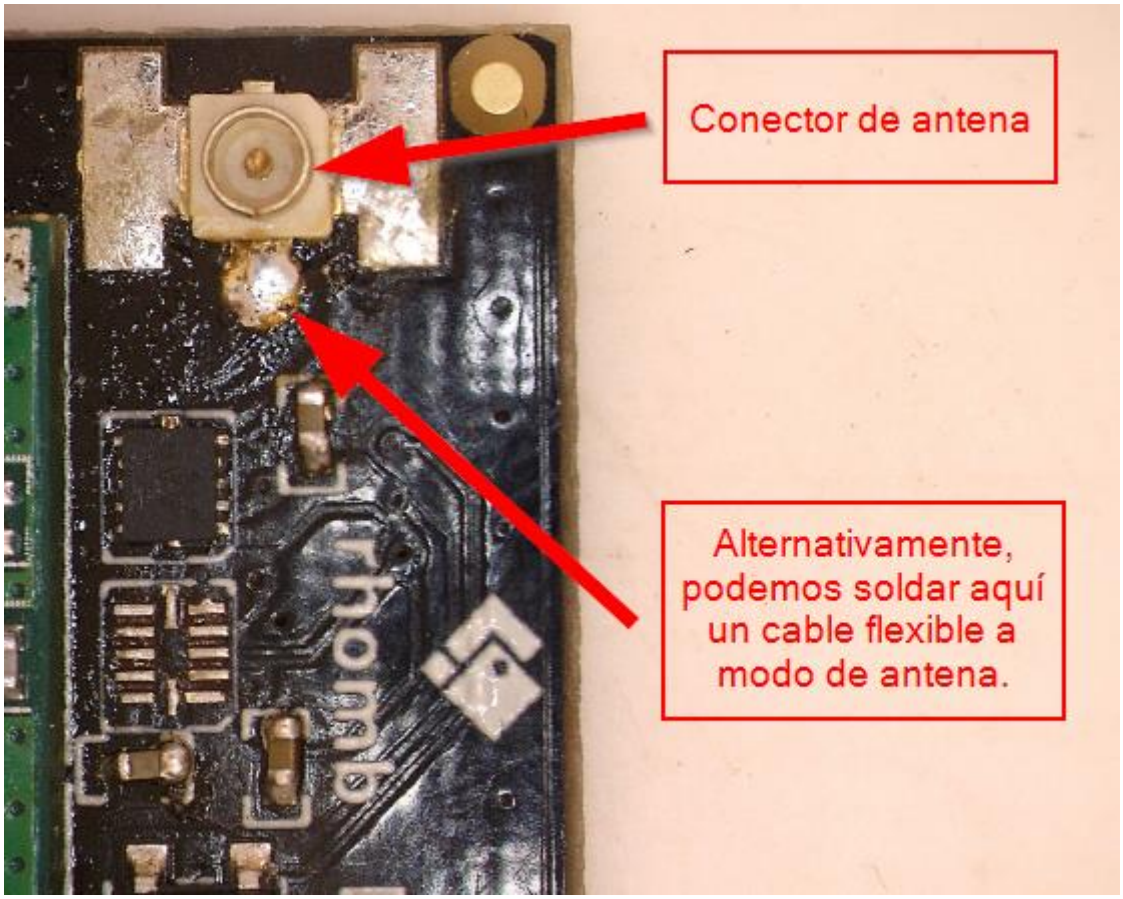
3.1. MÓDULOS

Insertar cada uno de los dos Duino uno/mega Rhomb.io master module en cada una de las dos placas de clase 2 o clase 3.

Insertar cada uno de los dos RFM69 Rhomb.io slave modules en cada una de las dos placas de clase 2 o clase 3.

3.2. ANTENA

El módulo dispone de un conector UMCC macho para antenas RF. La antena debe tener una longitud determinada. Opcionalmente, también puede utilizarse un pequeño trozo de cable flexible soldado a la toma de antena.



Detalle conexión antena en RFM69 Rhomb.io Slave Module

Para la realización de pruebas preliminares se puede emplear un cable, pero para un uso en producción debería emplearse una antena, ya que maximiza la potencia de la señal RF.

Para determinar la longitud del cable se emplea esta fórmula:

$$Longitud\ Antena = \frac{7500}{\text{Frecuencia (MHz)}}$$

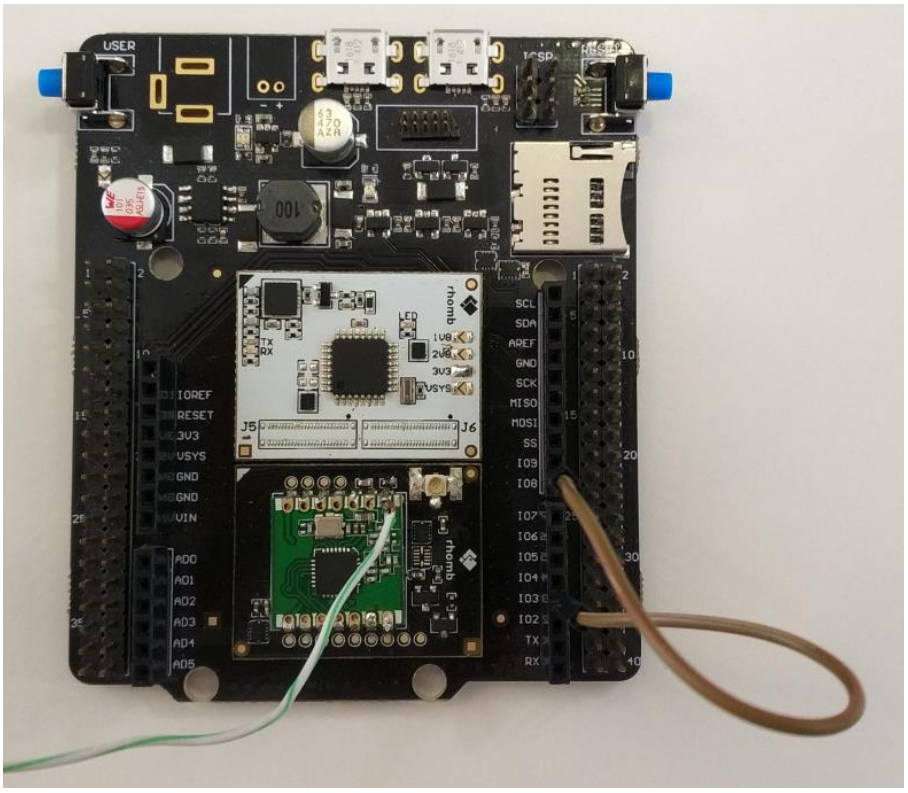
Para este ejemplo vamos a utilizar la frecuencia de 433 MHz por lo que la longitud del cable de nuestra antena sería:

$$Longitud\ Antena = \frac{7500}{433} = 17.32\ cm$$

3.3. Conexiones adicionales

Dependiendo que placa de clase 2 o 3 utilicemos y de en que socket pinchamos el esclavo, debemos realizar las siguientes conexiones por medio de un cable o pin:

- Helios, Phobos o Deimos Socket 1: Puentear I03 con I08
- Deimos socket 2: Puentear I03 con I011



Conjunto de Helios + RFM69 Slave Module. Al tratarse de una Helios, hemos puenteado I03 con I08 con un cable marrón.

4. EJEMPLO DE USO

Para llevar a cabo este ejemplo, es recomendable la lectura previa de los siguientes documentos:

- Flashear módulos master Duino Series en placas clase II y III
- Como importar librerías al IDE de Arduino

4.1. OBJETIVO

En este ejemplo se van a transmitir datos de forma inalámbrica desde una PCB a otra mediante radio frecuencia. Se va a emplear la frecuencia de 433MHz. Durante la prueba, va a poder medirse la intensidad de la señal que se recibe en cada momento.

4.2. PROCEDIMIENTO

Tenemos dos sketches, servidor y cliente. En cada uno, existe un bloque de código que debemos modificar en función de que placa estamos utilizando. La modificación consiste en comentar o descomentar una línea del código.

- Para comentar una línea escribiremos // al principio de la línea
- Para descomentar una línea, borraremos //

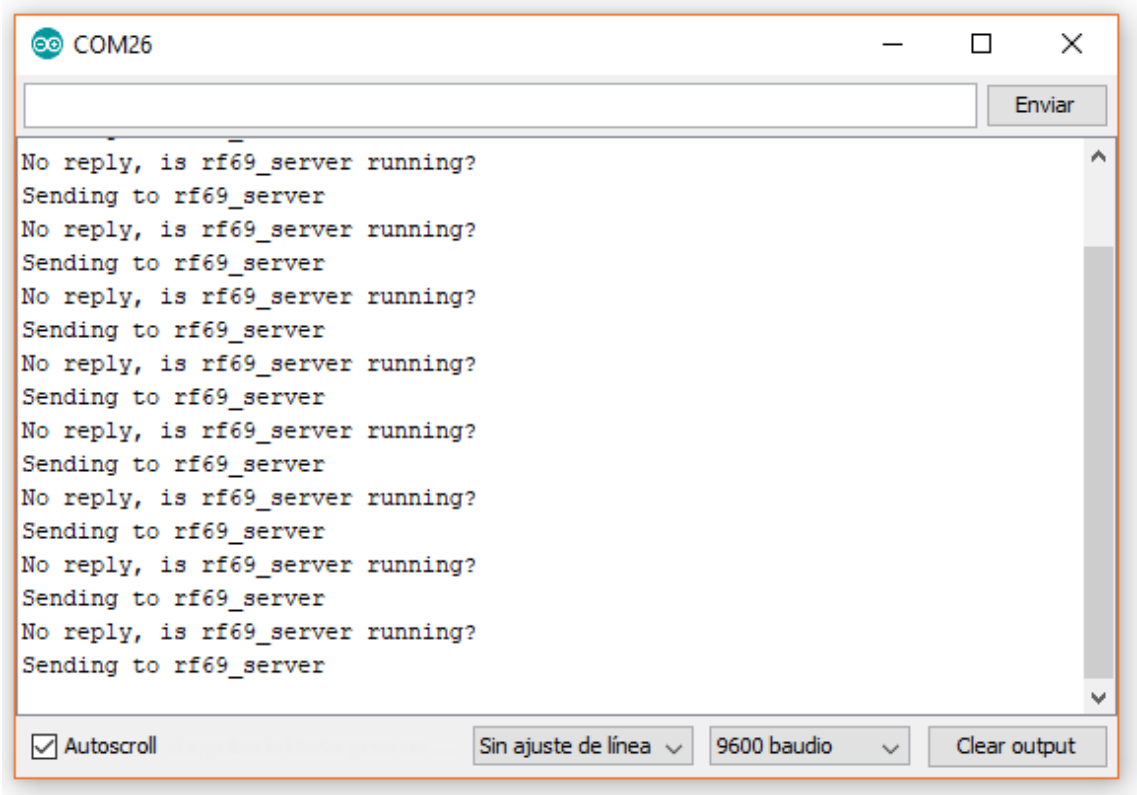
```
23 /********** rhomb.io ***********/
24 // Si usamos RF69 en Helios, Phobos o Deimos-Socket_1 hay que puentear IO3 con IO8
25 // para que funcione la interrupción.
26 // Si usamos RF69 en Deimos-Socket_2 hay que puentear IO3 con IO11
27 // para que funcione la interrupción.
28 RH_RF69 rf69 (10, 3); // Si usamos Duino Uno y RFM69 en Helios, Phobos o Deimos-Socket_1
29 //RH_RF69 rf69 (15, 3);  // Si usamos Duino Mega y RFM69 en Deimos-Socket_2
30 //RH_RF69 rf69 (53, 3); // Si usamos Duino Mega y RFM69 en Helios, Phobos o Deimos-Socket_1
31 /********** rhomb.io ***********/
```

Fragmento de código a personalizar en cada sketch según placa utilizada

1. Descomentar la línea apropiada para nuestra placa.
2. Dejar comentadas el resto.

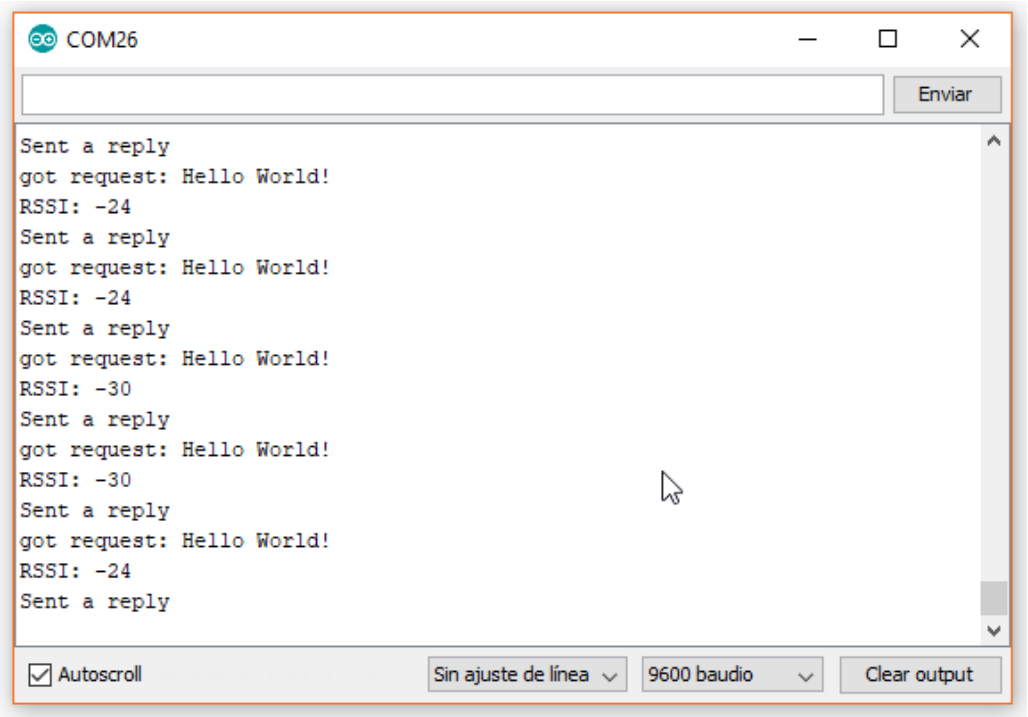
En este ejemplo la línea 28 está descomentada ya que estamos empleando Duino Uno y RFM69 en Helios.

3. Flashear en uno de los masters el sketch: rf69_server_gui_01.ino
4. Flashear en el otro master el sketch: rf69_client_gui_01.ino
5. Dejar el cliente conectado al PC o bien dejarlo conectado a un alimentador con conexión micro USB. En este momento, el cliente, comienza a enviar tramas de datos a un posible server que los reciba. Si abrimos el terminal serie del IDE de Arduino (Ctrl+Mayus+M), veremos que está enviando tramas, pero nadie las recibe. Dejamos el cliente conectado y emitiendo.



Monitor serie del cliente cuando no existe un server escuchando.

6. Conectar el server a un PC con el IDE de Arduino, seleccionar el puerto COM correspondiente (Herramientas > Puerto) y abrir el terminal serie (Ctrl+Mayus+M). En este momento, debería verse como se reciben las tramas, y como a su vez, se envía un mensaje de respuesta al cliente con cada trama que recibe. Cada trama recibida muestra la potencia de la señal recibida (RSSI).

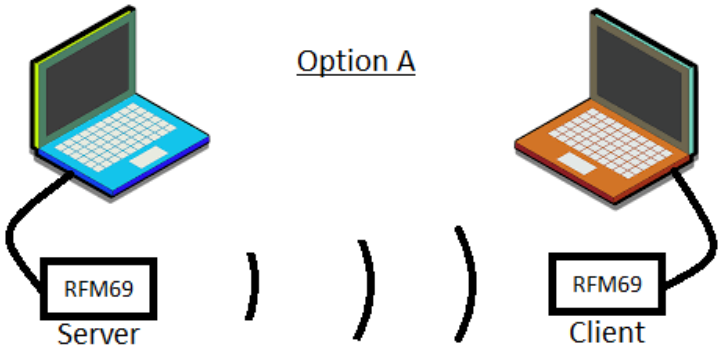


Monitor serie del server cuando recibe las tramas del cliente

Cuando tenemos conectado tanto el servidor como el cliente, aparecerá el texto de la imagen en el monitor serie del servidor. Si el terminal aparece en blanco, es síntoma de que el cliente no está transmitiendo correctamente.

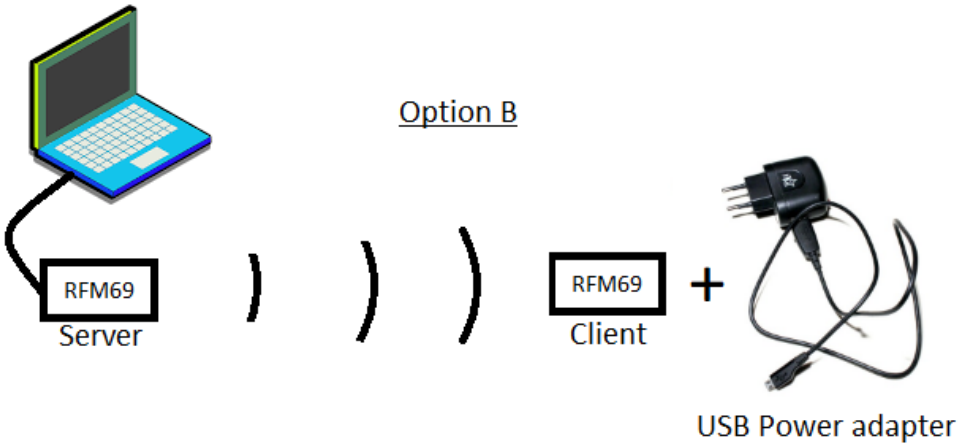
En este momento, podemos alejar el cliente del server y realizar pruebas de campo para verificar la potencia de recepción en el entorno real en el que se vaya a utilizar el módulo.

La idea es tener el cliente conectado a un PC y tener el server conectado a un segundo PC. Podemos abrir el monitor serie del IDE de Arduino en cada uno de ellos para monitorizar la operación.



Planteamiento con 2 PC

Si solo disponemos de un único PC, conectamos el cliente a un alimentador USB y utilizamos el PC para el servidor. No podremos ver el monitor serie del cliente, pero no es necesario.

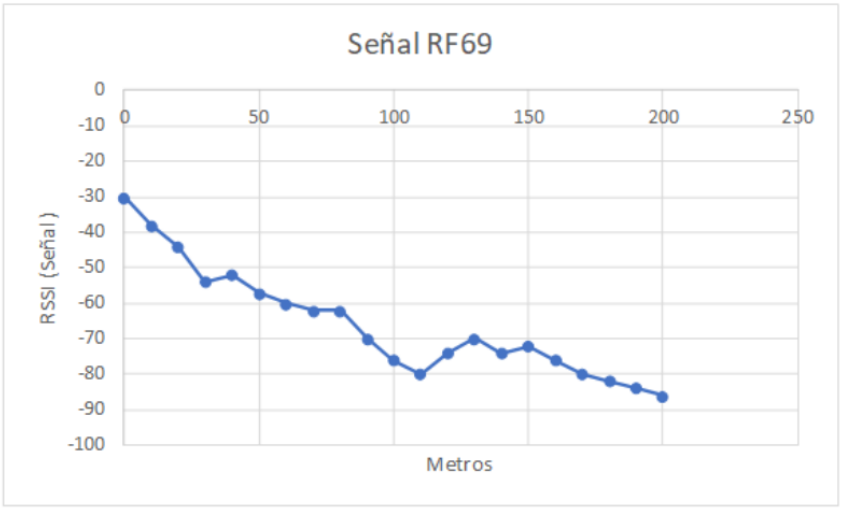


Planteamiento con 1 PC para el server y un alimentador USB para el cliente

4.3. RESULTADOS

Durante las pruebas de laboratorio realizadas por Rhomb.io hemos obtenido los siguientes resultados.

Metros	RSSI
0	-30
10	-38
20	-44
30	-54
40	-52
50	-57
60	-60
70	-62
80	-62
90	-70
100	-76
110	-80
120	-74
130	-70
140	-74
150	-72
160	-76
170	-80
180	-82
190	-84
200	-86



5. DISCLAIMER