

MANUAL DE USUARIO DEL MÓDULO BZI-RF2GH4

Copyright (c) 2007 Bizintek InnoVA, S.L.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Índice

Índice	3
1. Introducción:	4
2. Especificaciones técnicas	4
3. Descripción general	5
4. Librería para el módulo BZI-RF2GH4 en ensamblador	6
Variables	6
Funciones	8
5. Ejemplos en ensamblador	14
6. Librería para el módulo BZI-RF2GH4 en C para CCS	17
Variables	17
Funciones	18
7. Ejemplos en C para el compilador CCS	23
8. Prácticas	25
Programa 1	25
Programa 2	25
9. Regulador 3.3v (LE33CZ)	25

1. Introducción:

Este manual pretende ser una guía rápida para quienes se disponen a abordar una aplicación haciendo uso del módulo **BZI-RF2GH4**. En ella se irán exponiendo las características principales del dispositivo y se dará una orientación acompañada de ejemplos prácticos para una mejor comprensión.

2. Especificaciones técnicas

Parámetro	Valor	Unidad
Tensión mínima de alimentación	1.9	V
Tensión máxima de alimentación	3.6	V
Potencia máxima de salida	0	dBm
Velocidad máxima de transmisión	2000	Kbps
Corriente en modo transmisión @ 0dbm potencia de salida	11.3	mA
Corriente en modo recepción @ 2000kbps	12.3	mA
Corriente en modo <i>Power Down</i>	900	nA
Frecuencia máxima del bus SPI	8	Mhz
Rango de temperatura	-40 a +85	°C

Tabla 1. Parámetros principales del módulo BZI-RF2GH4

Pines	Nº	Descripción
Vcc	1	Tensión de alimentación del módulo
Vss	2	GND
CE	3	Chip Enable
CSN	4	Chip Select del SPI (Negado)
SCK	5	Reloj del bus SPI
SDI	6	Entrada de datos del bus SPI
SDO	7	Salida de datos del bus SPI
IRQ	8	Salida interrupción (negado)

Tabla 2. Pinout del módulo BZI-RF2GH4

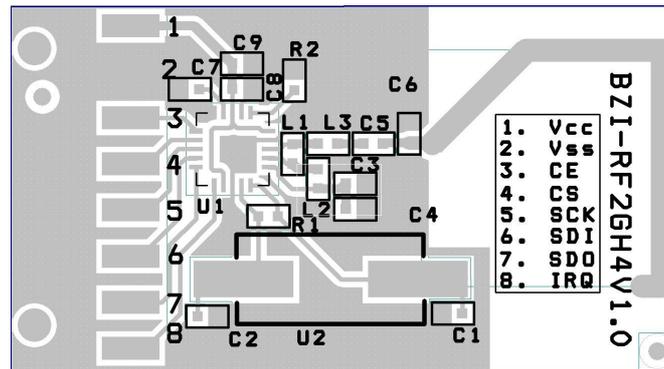


Fig. 1: Disposición de componentes en la placa

3. Descripción general

El módulo de comunicación por radiofrecuencia **BZI-RF2GH4** está basado en el transceptor nRF24L01 fabricado por “Nordic Semiconductors”. En este circuito integrado se ha incorporado toda la lógica necesaria para establecer una comunicación inalámbrica bidireccional con acuse de recibo. La comunicación con el microcontrolador se realiza a través de un bus SPI.

Las principales características del módulo BZI-RF2GH4 son:

- Bajo consumo.
- Frecuencia de trabajo de 2.4GHz,
- Potencia de emisión entre -18 y 0 dBm,
- Velocidad de transmisión entre 1 y 2 Mbps,
- 128 canales de transmisión seleccionables por el bus SPI,

El módulo **BZI-RF2GH4** incorpora, además del CI nRF24L01, toda la electrónica anexa para su correcto funcionamiento y una antena microstrip en la misma placa con la red de adaptación de impedancias. De esta manera el usuario se despreocupa por completo del hardware necesario para implementar la parte de radio de su aplicación.

Como interfaz dispone de cuatro pines accesibles para el bus SPI, dos pines más para el control del módulo y otros dos para la alimentación.

Con el objetivo de facilitar el manejo del módulo se han desarrollado unas librerías que simplifican y acortan el tiempo de desarrollo de cualquier aplicación inalámbrica con estos módulos.

4. Librería para el módulo BZI-RF2GH4 en ensamblador

Con esta librería es posible realizar fácilmente una comunicación entre microcontroladores PIC16F87xⁱ de “Microchip Technology” y el módulo BZI-RF2GH4.

El conjunto de las funciones de la librería ocupan unas 640 palabras de memoria de programa y 35 bytes de memoria de datos.

A su vez es importante tener en cuenta que para llamar a cualquier función de la librería son necesarios tres niveles de la pila libres y el “watchdog” desactivado.

Dado que todas las funciones utilizan el protocolo SPI, es necesario habilitar el hardware del microcontrolador para este propósito. Para ello basta con añadir unas líneas de código en la configuración inicial del programa.

Variables

RF_STATUS

Esta variable de solo lectura informa de la situación de la comunicación por el módulo de radio.

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Sin uso	CONFIGOK	OFFOK	ONOK	RCVNW	RCVOK	ACK	SNDOK
-							

Bit 7: **Sin uso**

Bit 6: **CONFIGOK:** Muestra si el módulo se ha configurado correctamente.
1 = El módulo se ha configurado correctamente.
0 = Módulo desconfigurado. Imposibilidad de comunicación con el módulo por ausencia o conexión eléctrica incorrecta.

Bit 5: **OFFOK:** Muestra si el módulo se ha apagado correctamente.
1 = El módulo se ha apagado correctamente.
0 = No se ha apagado correctamente. Imposibilidad de comunicación con el módulo por ausencia o conexión eléctrica incorrecta.

ⁱ Excepto el PIC16F870 y PIC16F871.

- Bit 4: **ONOK:** Muestra si el módulo se ha encendido correctamente.
1 = El módulo se ha encendido correctamente.
0 = El módulo no está activo. Imposibilidad de comunicación con el módulo por ausencia o conexión eléctrica incorrecta.
- Bit 3: **RCVNW:** Muestra si todavía quedan datos por leer.
1 = Quedan tramas de datos por leer en la pila del módulo de radio.
0 = Tras la última lectura, la pila de datos del módulo quedó vacía. No hay mensajes en espera.
- Bit 2: **RCVOK:** Informa que se han recibido datos correctamente y están accesibles para ser tratados.
1 = Recepción correcta.
0 = No se han recibido datos o la información recibida es corrupta.
- Bit 1: **ACK:** Muestra si se ha recibido el ACK (confirmación) del receptor tras una transmisión.
1 = El receptor ha confirmado que ha recibido los datos correctamente.
0 = No se ha recibido la confirmación del receptor. Puede ser causa de que no haya recibido la señal o de que tenga la pila llena y no pueda almacenar más mensajes.
- Bit 0: **SNDOK:** Muestra si el último envío de datos se ha realizado.
1 = El módulo de radio ha enviado los datos. Este bit no indica que alguien lo haya escuchado.
0 = No ha sido posible enviar los datos. Puede ser debido a un fallo en la comunicación con el módulo de radio

RF_DATA_OUT_0, RF_DATA_OUT_1,... RF_DATA_OUT_7

Este grupo de variables consta de 8 bytes. En cada transmisión se envía el contenido de los 8 bytes.

RF_DATA_IN_0, RF_DATA_IN_1,... RF_DATA_IN_7

Este grupo de variables consta de 8 bytes. En cada recepción se actualizan estos 8 bytes.

RF_DIR_OUT

Esta variable es de un solo byte. Indica la dirección del dispositivo que se desea mandar el dato.

RF_DIR_IN

Esta variable es de un solo byte. Indica la dirección de los datos recibidos.

RF_DIR

Esta variable es de un solo byte. Indica la dirección propia con la que se configura el módulo.

Funciones

La librería consta de nueve funciones que facilitarán bastante el trabajo a la hora de desarrollar una aplicación de comunicación con los módulos **BZI-RF2GH4**. A continuación se dará una breve descripción de cada una de ellas.

Funciones para el módulo BZI-RF2GH4	
RF_CONFIG	Configura las entradas y salidas del microcontrolador ⁱ así como parámetros del módulo de radio.
RF_CONFIG_SPI	Configura las entradas y salidas del microcontrolador ⁱⁱ así como los parámetros necesarios para utilizar el bus SPI.
RF_ON	Activa el módulo de radiofrecuencia en modo escucha.
RF_OFF	Desactiva el módulo de radiofrecuencia y lo deja en modo de bajo consumo.
RF_SEND	Envía una trama de datos (8 Bytes) a la dirección indicada.
RF_RECEIVE	Comprueba si se ha producido una recepción y de ser así, recoge la trama.
RF_RECEIVE_INT	Realiza la misma función que RF_RECEIVE solo que por interrupción.
RF_INT_EN	Esta rutina habilita en el microcontrolador la interrupción externa para el módulo de radio.

Tabla resumen de las funciones en ensamblador.

RF_CONFIG

Variables de entrada	
RF_DIR	Dirección del dispositivo. Debe ser un valor entre 0x01 y 0xFE.
RF_CHN	Canal a utilizar en la comunicación. Debe ser un valor entre 0x00 y 0x7F (128 canales).
Variables de salida	
FLAGS: CONFIGOK	

Esta función configura el transceptor habilitando su propia dirección de escucha y la dirección de 'broadcast'. A su vez configura otros parámetros como pines del PIC, el

ⁱ El usuario puede cambiar estos pines modificando la parte de la librería donde se define el patillaje

ⁱⁱ Ídem.

canal, la velocidad de transmisión, la potencia de emisión, la longitud de la dirección, la longitud del código CRC, etc.

PIN RF	PIN PIC
IRQ	RB0
CSN	RB7
CE	RC2

Tabla 3. Configuración módulo RF puertos PIC

El canal debe ser común a todos los módulos que van a participar en la comunicación. El usuario puede elegir cualquier canal de los 128 disponibles. Sin embargo, si en el entorno existe más de una comunicación entre módulos en diferentes canales, hay que dejar un espaciado de 2 entre los canales a utilizar para evitar interferencias dejando así 32 canales útiles. Otra cuestión a tener en cuenta es la existencia de otras tecnologías que utilizan la banda ISM 2.4GHz (Wifi, Bluetooth, etc.) y que también pueden causar interferencias en alguno de los canales.

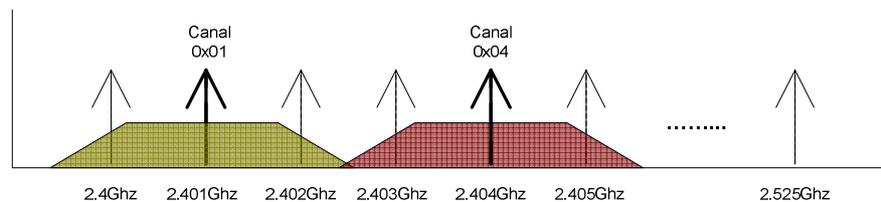


Fig. 2: Canales RF

La dirección que se asigna a cada dispositivo debe ser unívoca dentro de cada canal.

Si la comunicación SPI del PIC está mal configurada, el conecionado se ha realizado incorrectamente o en ausencia del módulo, el bit CONFIGOK de RF_STATUS permanecerá a 0.

RF_CONFIG_SPI

La velocidad del SPI no debe superar los 8 Mhz por lo que la utilización de esta función queda limitada a microcontroladores PIC con una frecuencia menor de 32Mhz. En la función se configuran los diferentes parámetros del módulo SPI y los pines del PIC.

PIN RF	PIN PIC
SCK	RC3
SDI	RC4
SDO	RC5

Tabla 4. Configuración SPI puertos PIC

RF_ON***Variables de salida***

FLAGS: ONOK

Esta rutina activa el módulo de radio en modo escucha para poder recibir datos y/o realizar envíos de datos.

Es importante tener en cuenta que tras la llamada a esta rutina el módulo necesita 2,5ms para estar listo.

Si la comunicación SPI del PIC está mal configurada, el conexionado se ha realizado incorrectamente o en ausencia del módulo, el bit ONOK de RF_STATUS permanecerá a 0.

RF_OFF***Variables de salida***

FLAGS: OFFOK

Esta rutina desactiva el módulo de radio dejándolo en modo de bajo consumo. No borra la configuración establecida.

Si la comunicación SPI del PIC está mal configurada, el conexionado se ha realizado incorrectamente o en ausencia del módulo, el bit OFFOK de RF_STATUS permanecerá a 0.

RF_SEND***Variables de entrada***

RF_DIR_OUT Dirección a la que se quiere enviar los datos (1 byte).

RF_DATA_OUT_0 –
RF_DATA_OUT_7 Variables que van a ser transmitidas (8 bytes).***Variables de salida***

FLAGS: SNDOK y ACK

Esta función envía 8 Bytes de datos a la dirección indicada informando de la correcta recepción en el destinatario. Tras su ejecución el dispositivo volverá al modo de escucha.

Si un mensaje es enviado a la dirección 0x00, este será recibido por todos los módulos que se encuentren en el mismo canal. Se debe tomar en cuenta que el módulo acepta el primer ACK que recibe, por lo tanto no tendremos la certeza de la llegada del dato a todos los dispositivos.

RF_RECEIVE

<i>Variables de salida</i>	
RF_DIR_IN	Dirección de quien ha enviado la trama
RF_DATA_IN_0 – RF_DATA_IN_7	Trama recibida desde la dirección indicada.
RCVOK y RCVNW	

Esta rutina se encarga de comprobar si se ha producido una recepción y de ser así, devuelve los datos recibidos. Asimismo, informa si quedan datos sin leer en la FIFO de recepción del módulo.

Cuando se reciba una trama se debe hacer una comprobación del bit RCVNW de la variable RF_STATUS y si está activo se debe llamar a la función RF_RECEIVE de nuevo tras tratar los datos. El transceptor tiene una pila de tres niveles, por lo que si no se llama a la función recibir antes de que se llene la pila, el dispositivo no podrá recibir más datos.

Al no utilizar interrupciones, la probabilidad de pérdida de paquetes, con tráfico elevado, es moderada. Es aconsejable utilizarla sólo en entornos con pocos dispositivos y/o poco tráfico de datos. También se puede solucionar este problema haciendo que los emisores reenvíen la misma trama hasta que la comunicación haya sido correcta, pero en entornos con mucho tráfico las colisiones crecen exponencialmente aumentando considerablemente los tiempos de envío.

RF_RECEIVE_INT

<i>Variables de salida</i>	
RF_DIR_IN	Dirección de quien ha enviado la trama
RF_DATA_IN_0 – RF_DATA_IN_7	Trama recibida desde la dirección indicada.
RCVOK, RCVNW	

Es la rutina óptima de recepción. Es una rutina prácticamente igual a RF_RECEIVE con la diferencia que esta funciona por interrupción. Por ello debe ubicarse dentro del código de la interrupción y haber configurado las interrupciones anteriormente (RF_INTER_EN). Se encarga de comprobar si se ha producido una interrupción externa (RBO) y si es así, limpia el flag de interrupción. La probabilidad de pérdida de paquetes es mínima. Aún así conviene que los emisores reenvíen si el flag de envío no se activa.

RF_INT_EN

Esta rutina se encarga de habilitar la interrupción externa del microcontrolador (RB0) que utiliza el módulo de RF en la recepción de datos. Para ello se configura como entrada el pin RB0. Aunque se puede manejar el módulo sin interrupciones no se garantiza el mínimo tiempo de respuesta.

Diagrama de flujo para envío y recepción de datos

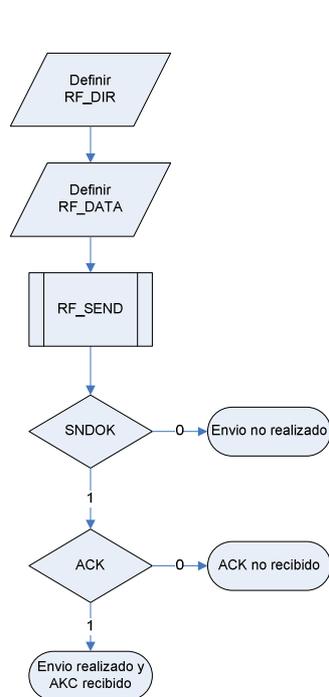


Diagrama 1. Envío de datos en ensamblador

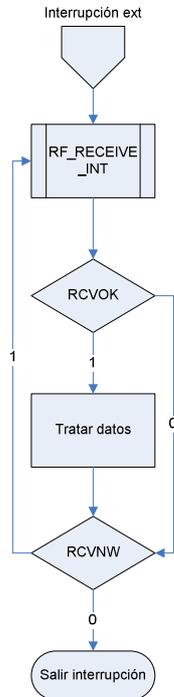


Diagrama 2. Recepción con interrupción en ensamblador

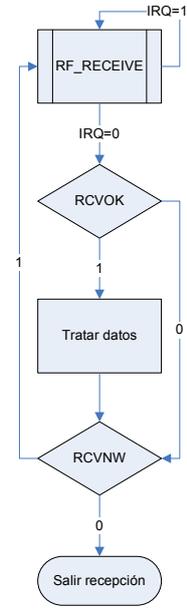


Diagrama 3. Recepción sin interrupción en ensamblador

5. Ejemplos en ensamblador

En este apartado se muestran fragmentos de código a modo de ejemplo para orientar en la utilización de las funciones.

```
;-[Rutina de configuración con interrupción y activación]-----
```

```
  ;Habilitar interrupciones
```

```
  call      RF_INT_EN
```

```
  ;Configurar módulos SPI del PIC
```

```
  call      RF_CONFIG_SPI
```

```
  ;Configurar módulo RF (canal y dirección propias)
```

```
  movlw    0x01      ;Dirección propia
```

```
  movwf    RF_DIR
```

```
  movlw    0x40      ;Canal
```

```
  movwf    RF_CHN
```

```
  call     RF_CONFIG
```

```
  btfss   RF_STATUS,CONFIGOK
```

```
  nop     ;Módulo no configurado
```

```
  ;Activar el módulo RF
```

```
  call    RF_ON
```

```
  btfss  RF_STATUS,ONOK
```

```
  nop    ;Módulo no inicializado
```

```
;------
```

```
;-[Rutina de configuración sin interrupción y activación]-----
```

```
  ;Configurar módulos SPI del PIC
```

```
  call      RF_CONFIG_SPI
```

```
  ;Configurar módulo RF (canal y dirección propias)
```

```
  movlw    0x01      ;Dirección propia
```

```
  movwf    RF_DIR
```

```
  movlw    0x40      ;Canal
```

```
  movwf    RF_CHN
```

```
  call     RF_CONFIG
```

```
  btfss   RF_STATUS,CONFIGOK
```

```
  nop     ;Módulo no configurado
```

```
;Activar el módulo RF
call      RF_ON
btfs     RF_STATUS,ONOK
nop      ;Módulo no inicializado
```

```
;
```

```
;-----[Rutina de envío de datos]-----
```

```
;Preparación de la dirección del receptor
;y de los datos.
```

```
movlw    0x02      ;Dirección del receptor
movwf    RF_DIR_OUT

;Datos a enviar
clrf     RF_DATA_OUT_0
clrf     RF_DATA_OUT_1
clrf     RF_DATA_OUT_2
clrf     RF_DATA_OUT_3
clrf     RF_DATA_OUT_4
clrf     RF_DATA_OUT_5
clrf     RF_DATA_OUT_6
clrf     RF_DATA_OUT_7

call     RF_SEND    ;Enviar trama
btfs     RF_STATUS,SENDOK
nop      ;No enviado
btfs     RF_STATUS,ACK
nop      ;No ACK
```

```
;
```

```
;------[Rutina de recepción de datos con interrupción]-----
```

```
LEER_MAS_DATOS
call     RF_RECEIVE_INT
btfs     RF_STATUS,RCVOK
nop      ;Tratar datos
btfs     RF_STATUS,RCVNW
goto     LEER_MAS_DATOS
goto     SALIR_INTERRUPCIONES
```

```
;
```

;---- Rutina de recepción de datos sin interrupción]-----

RECIBIR

call	RF_RECEIVE	
btfs	RF_STATUS,RCVOK	
nop		;Tratar datos
btfs	RF_STATUS,RCVNW	
goto	RECIBIR	

;-----

6. Librería para el módulo BZI-RF2GH4 en C para CCS

Con esta librería es posible realizar fácilmente una comunicación entre microcontroladores PIC16F87x¹ de “Microchip Technology” y el BZI-RF2GH4.

El conjunto de las funciones de la librería ocupan 1k palabras de memoria de programa, y 33 bytes de memoria de datos.

A su vez es importante tener en cuenta que para llamar a cualquier función de la librería es necesario tener el “watchdog” desactivado.

Dado que todas las funciones utilizan el protocolo SPI, es necesario habilitar el hardware del microcontrolador para este propósito. Para ello basta con añadir unas líneas de código en la configuración inicial del programa.

Variables

RF_DATA_OUT

Es una variable de 8 bytes que es utilizada por la función de transmisión. En ella están los datos transmitidos.

RF_DATA_IN

Es una variable de 8 bytes que es utilizada por la función de recepción. En ella están los datos recibidos.

RF_DIR_OUT

Esta variable es de un solo byte. Indica el destino del mensaje de 8 bytes de RF_DATA_OUT. En esta variable escribiremos la dirección del destinatario a la hora de realizar un envío.

RF_DIR_IN

Esta variable es de un solo byte. Indica el origen del mensaje de 8 bytes de RF_DATA_IN. En esta variable podremos ver la dirección de quien nos está enviando la información.

RF_DIR

Esta variable es de un solo byte. Indica nuestra propia dirección.

¹ Excepto el PIC16F870 y PIC16F871.

Funciones

Para manejar el paso de parámetros y el retorno de valores, se utilizan variables externas que hay que modificar antes o verificar después de cada llamada. En cada función se expondrá cuáles son y cómo actúan.

Funciones para el módulo BZI-RF2GH4	
int RF_CONFIG(int canal, int dir)	Configura las entradas y salidas del microcontrolador ⁱ así como parámetros del módulo de radio.
void RF_CONFIG_SPI()	Configura las entradas y salidas del microcontrolador ⁱⁱ así como los parámetros necesarios para utilizar el bus SPI.
int RF_ON()	Activa el módulo de radiofrecuencia en modo escucha.
int RF_OFF()	Desactiva el módulo de radiofrecuencia y lo deja en modo de bajo consumo
int RF_SEND()	Envía una trama de datos (8 Bytes) a la dirección indicada.
int RF_RECEIVE()	Comprueba si se ha producido una recepción y de ser así, recoge la trama.
void RF_INT_EN()	Esta rutina habilita en el microcontrolador la interrupción externa para el módulo de radio.

Tabla resumen de las funciones en ensamblador.

void RF_CONFIG(int canal, int dir)

Variables de entrada	
RF_DIR	Dirección del dispositivo. Debe ser un valor entre 0x01 y 0xFE.
RF_CHN	Canal a utilizar en la comunicación. Debe ser un valor entre 0x00 y 0x7F (128 canales).
Retorno de la Función	
1: Configuración correcta	El módulo se ha configurado correctamente.
0: Configuración incorrecta	El módulo no está configurado. Imposibilidad de comunicación con el módulo por ausencia o conexión eléctrica incorrecta.

Esta función configura el transceptor habilitando su propia dirección de escucha y la dirección de 'broadcast'. A su vez configura otros parámetros como pines del PIC, el

ⁱ El usuario puede cambiar estos pines modificando la parte de la librería donde se define el patillaje

ⁱⁱ Ídem.

canal, la velocidad de transmisión, la potencia de emisión, la longitud de la dirección, la longitud del código CRC, etc.

PIN RF	PIN PIC
IRQ	RB0
CSN	RB7
CE	RC2

Tabla 5. Configuración módulo RF puertos PIC

El canal debe ser común a todos los módulos que van a participar en la comunicación. El usuario puede elegir cualquier canal de los 128 disponibles. Sin embargo, si en el entorno existe más de una comunicación entre módulos en diferentes canales, hay que dejar un espaciamiento de 2 entre los canales a utilizar para evitar interferencias dejando así 32 canales útiles. Otra cuestión a tener en cuenta es la existencia de otras tecnologías que utilizan la banda ISM 2.4GHz (Wifi, Bluetooth, etc.) y que también pueden causar interferencias en alguno de los canales.

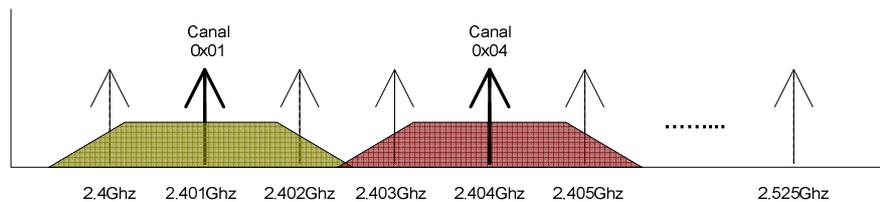


Imagen 1. Canales RF

Antes de llamar a esta función el módulo SPI debe estar configurado.

void RF_CONFIG_SPI()

La velocidad del SPI no debe superar los 8 Mhz por lo que la utilización de esta función queda limitada a microcontroladores PIC con una frecuencia menor de 32Mhz. En la función se configura los diferentes parámetros del módulo SPI, y los pines del PIC.

PIN RF	PIN PIC
SCK	RC3
SDI	RC4
SDO	RC5

Tabla 6. Configuración SPI puertos PIC

int RF_ON()

Retorno de la Función	
1: Activación correcta	El módulo se ha encendido correctamente.
0: Activación incorrecta	El módulo no está activo. Imposibilidad de comunicación con el módulo o, por lo contrario, ausencia o conexión eléctrica incorrecta.

Esta rutina activa el módulo de radio en modo escucha para poder recibir datos y/o realizar envíos de datos.

Es importante tener en cuenta que tras la llamada a esta rutina el módulo necesita 2,5ms para estar listo.

void RF_OFF()

Retorno de la Función	
1: Desactivación correcta	El módulo se ha desactivado correctamente.
0: Desactivación incorrecta	El módulo no se ha desactivado correctamente. Imposibilidad de comunicación con el módulo por ausencia o conexión eléctrica incorrecta.

Esta rutina desactiva el módulo de radio dejándolo en modo de bajo consumo. No borra la configuración establecida.

int RF_SEND()

Variables de entrada	
RF_DATA_OUT	Es una variable de 8 bytes. (RF_DATA_OUT[0 - 7]).
RF_DIR_OUT	
Retorno de la Función	
0: Envío correcto (ACK OK)	Se ha realizado el envío y se ha recibido el ACK del receptor.
1: Recepción incorrecta de ACK (NO ACK)	Se ha enviado la información pero no se ha recibido ACK (receptor mal configurado, canal diferente en el receptor, dirección incorrecta).
2: No enviado	No se ha realizado el envío.

Esta función envía 8 Bytes de datos a la dirección indicada informando de la correcta recepción en el destinatario. Tras su ejecución el dispositivo volverá al modo de escucha.

Si un mensaje es enviado a la dirección 0x00, este será recibido por todos los módulos que se encuentren en el mismo canal. Se debe tomar en cuenta que el módulo acepta el primer ACK que recibe, por lo tanto no tendremos la certeza de la llegada del dato a todos los dispositivos.

int RF_RECEIVE()

<i>Variables de salida</i>	
RF_DATA_IN	Es una variable de 8 bytes. Presenta la información recibida (RF_DATA_IN[0 - 7]).
RF_DIR_IN	Es una variable de byte. Indica la dirección del emisor.
<i>Retorno de la Función</i>	
0	Recepción única. No hay más datos en la pila de recepción.
1	Recepción múltiple. Hay más datos en la pila de recepción. Ocurre cuando el emisor envía más de una trama antes de que el receptor la recoja.
2	No se han recibido datos.

Esta rutina se encarga de comprobar si se ha producido una recepción y de ser así, devuelve los datos recibidos. Asimismo, informa si quedan datos sin leer en la FIFO de recepción del módulo.

Cuando se reciba una trama se debe hacer una comprobación de la salida de la función. Si la función devuelve un 1 se debe llamar a la función RF_RECEIVE() de nuevo pero antes es necesario tratar los datos o se perderán. El transceptor tiene una pila de tres niveles, por lo que si no se llama a la función RF_RECEIVE() antes de que se llene la pila, el dispositivo no podrá recibir más datos.

void RF_INT_EN()

Esta rutina se encarga de habilitar la interrupción externa del microcontrolador que utiliza el módulo de RF en la recepción de datos. Para ello se configura como entrada el pin RB0. Aunque se puede manejar el módulo sin interrupciones no se garantiza el mínimo tiempo de respuesta.

Diagrama de flujo para envío y recepción de datos

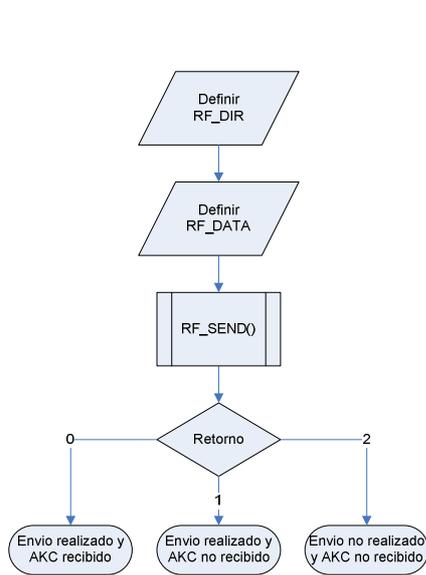


Diagrama 4. Envío de datos en C

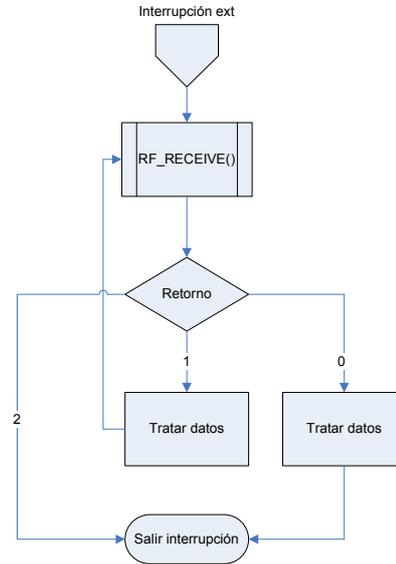


Diagrama 5. Recepción con interrupción en C

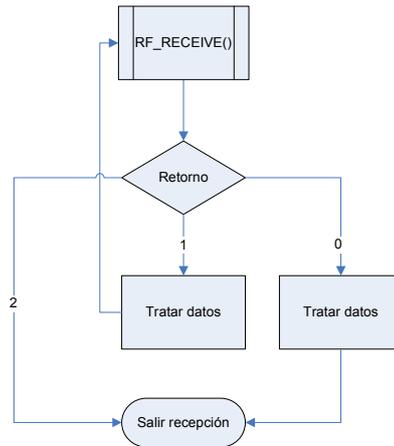


Diagrama 6. Recepción sin interrupción en C

7. Ejemplos en C para el compilador CCS

En este apartado se muestran fragmentos de código a modo de ejemplo para orientar en la utilización de las funciones.

```
//-----[Rutina de configuración y activación con interrupción]-----  
//Habilitar interrupciones  
RF_INT_EN();  
  
//Configurar módulos SPI del PIC  
RF_CONFIG_SPI();  
  
//Configurar módulo RF (canal y dirección)  
if(RF_CONFIG(0x40,0x01)==0){  
//Módulo no configurado  
}  
  
//Activar el módulo RF  
if(RF_ON()==0){  
//Módulo no inicializado  
}  
//-----  
  
//-----[Rutina de configuración y activación sin interrupción]-----  
//Configurar módulos SPI del PIC  
RF_CONFIG_SPI();  
  
//Configurar módulo RF (canal y dirección)  
if(RF_CONFIG(0x40,0x01)==0){  
//Módulo no configurado  
}  
  
//Activar el módulo RF  
if(RF_ON()==0){  
//Módulo no inicializado  
}  
//-----  
  
//-----[Rutina de envío de datos]-----  
//Preparación de la dirección del receptor  
//y de los datos.  
RF_DIR_OUT=0x02;  
for(i=0;i<8;i++)
```

```
{
RF_DATA_OUT[i]=0x00;
}

//Envío de los datos
ret=RF_SEND();
if(ret==0){
//Envío realizado y ACK recibido
}
else if(ret==1){
//Envío realizado y ACK no recibido
}
else{
//Envío no realizado
}
//-----

//-----[Rutina de recepción con interrupción]-----
//Interrupción del módulo RF
#int_ext
void int_externo() {

    while(RF_RECEIVE()!=2){
        // Sustituir por código necesario para tratar los datos
    }
}
//-----

//-----[Rutina de recepción sin interrupción]-----
//Interrupción del módulo RF
while(1){

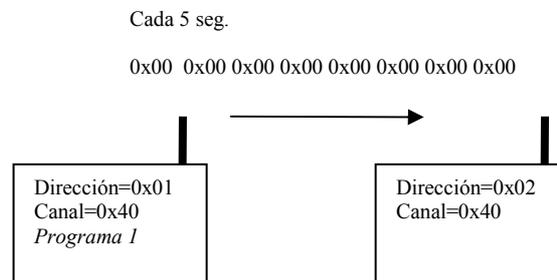
    while(RF_RECEIVE()!=2){
        // Sustituir por código necesario para tratar los datos
    }
}
//-----
```

8. Prácticas

El usuario tiene disponibles 2 programas ejemplo en ensamblador y en C.

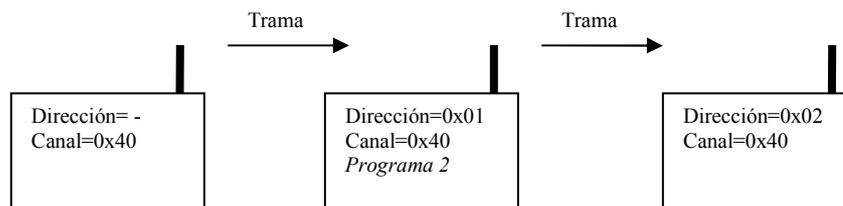
Programa 1

Envío de una trama con ceros a la dirección 0x02 cada 5 segundos. También tiene implementada la rutina de recepción con interrupción.



Programa 2

Al recibir una trama desde cualquier dirección la retransmite a la dirección 0x02. Este programa recibe las tramas sin interrupción, chequeando continuamente si se produce una nueva recepción.



9. Regulador 3.3v (LE33CZ)

Con el módulo RF se distribuye también un regulador lineal de salida a para la alimentación. En las siguientes tablas se describen las principales características eléctricas y el pinout.

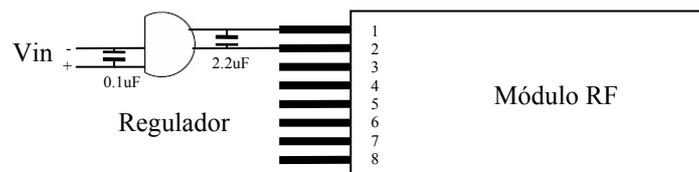
Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit	
V_O	Output Voltage	$I_O = 10 \text{ mA}, V_I = 5.3 \text{ V}$	3.234	3.3	3.366	V	
		$I_O = 10 \text{ mA}, V_I = 5.3 \text{ V}, T_a = -25 \text{ to } 85^\circ\text{C}$	3.168		3.432		
V_I	Operating Input Voltage	$I_O = 100 \text{ mA}$			18	V	
I_O	Output Current Limit		150			mA	
ΔV_O	Line Regulation	$V_I = 4 \text{ to } 18 \text{ V}, I_O = 0.5 \text{ mA}$		3	20	mV	
ΔV_O	Load Regulation	$V_I = 4.3 \text{ V}, I_O = 0.5 \text{ to } 100 \text{ mA}$		3	25	mV	
I_d	Quiescent Current	$V_I = 4.3 \text{ to } 18 \text{ V}, I_O = 0 \text{ mA}$	ON MODE		0.5	1	mA
		$V_I = 4.3 \text{ to } 18 \text{ V}, I_O = 100 \text{ mA}$			1.5	3	
		$V_I = 6 \text{ V}$	OFF MODE		50	100	μA
SVR	Supply Voltage Rejection	$I_O = 5 \text{ mA}$ $V_I = 5.3 \pm 1 \text{ V}$	$f = 120 \text{ Hz}$	80		dB	
			$f = 1 \text{ KHz}$	75			
			$f = 10 \text{ KHz}$	60			
eN	Output Noise Voltage	$B = 10 \text{ Hz to } 100 \text{ KHz}$		50		μV	
V_d	Dropout Voltage	$I_O = 100 \text{ mA}$		0.2	0.4	V	
		$I_O = 100 \text{ mA}, T_a = -40 \text{ to } 125^\circ\text{C}$			0.5		
V_{IL}	Control Input Logic Low	$T_a = -40 \text{ to } 125^\circ\text{C}$			0.8	V	
V_{IH}	Control Input Logic High	$T_a = -40 \text{ to } 125^\circ\text{C}$	2			V	
I_I	Control Input Current	$V_I = 6 \text{ V}, V_C = 6 \text{ V}$		10		μA	
C_O	Output Bypass Capacitance	$\text{ESR} = 0.1 \text{ to } 10 \Omega, I_O = 0 \text{ to } 100 \text{ mA}$	2	10		μF	

Tabla 7. Principales características del regulador

Pines	Nº	Descripción
Vout	1	Tensión de salida
GND	2	Tierra
Vin	3	Tensión de entrada

Tabla 8. Pinout regulador


La conexión del regulador con el módulo se realizará de la siguiente manera.


Vista superior