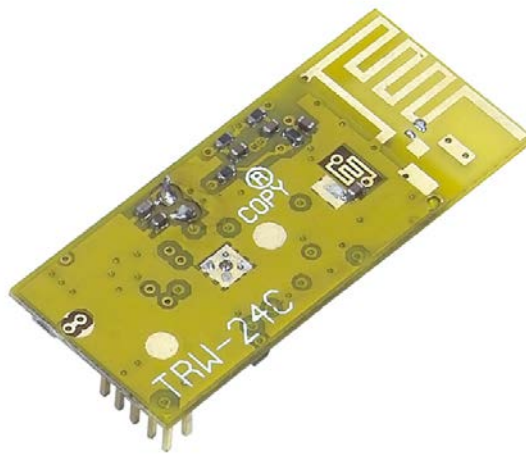

Wireless Low Cost 2.4GHz Hi Power +19dBm RF Transceiver Module



Version History

Version	Date	Changes
V1.01	Jun 14, 2007	1 st . Edition
V1.02	Aug 20, 2007	2 nd . Edition
V1.03	Mar 15, 2008	3 rd . Edition
V1.04	Jan 4, 2009	4 th . Edition
V1.05	Jan 7, 2010	5 th . Edition

Specification

● Remote Control Systems	● Wireless Data Transceiver
● 2.4GHz~2.483GHz ISM/SRD Band	● Remote Metering
● Wireless Security Systems	● Automatic Meter Reading
● Application Range : Remote Metering 、 Wireless Security Systems 、 Automatic Meter 、 Reading 、 Home Automation	

Application

The RF transceiver is integrated with a highly configurable baseband modem. The modem supports various modulation formats and has a configurable data rate up to 250 kbps. The communication range can be increased by enabling a Forward Error Correction option, which is integrated in the modem.

TRW-24C provides extensive hardware support for packet handling, data buffering, burst transmissions, clear channel assessment, link quality indication and wake-on-radio.

The main operating parameters and the 64bytes transmit/receive FIFOs of TRW-24C can be controlled via an SPI interface. In a typical system, the TRW-24C will be used together with a microcontroller and a few additional passive components.

Key Feature

- Small size
- Separate 64bytes RX and TX data FIFOs.
- Efficient SPI interface: All registers can be programmed with one "burst" transfer.
- Programmable output power up to +19dBm.
- High sensitivity (-106dBm at 10kbps, 1% packet error rate).

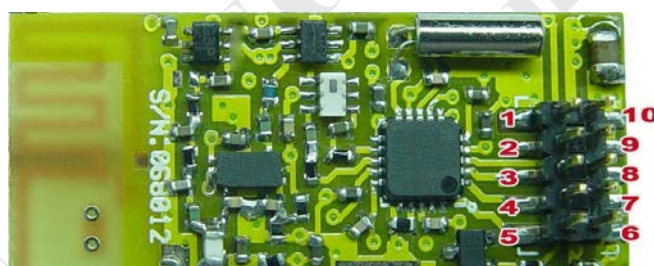
Absolute Maximum Rating

Parameter	Min	Max	Unit	Condition/Note
Supply Voltage	2.7V	3.3V	V	
Voltage on Any Digital Pin	-0.3	VDD+0.3	V	
Input RF Level		+10	dBm	
Storage Temperature Range	-50	+150	°C	
ESD		<500	V	According to JEDEC STD 22, method A114, Human Body Model

General Characteristics and Electrical Specification

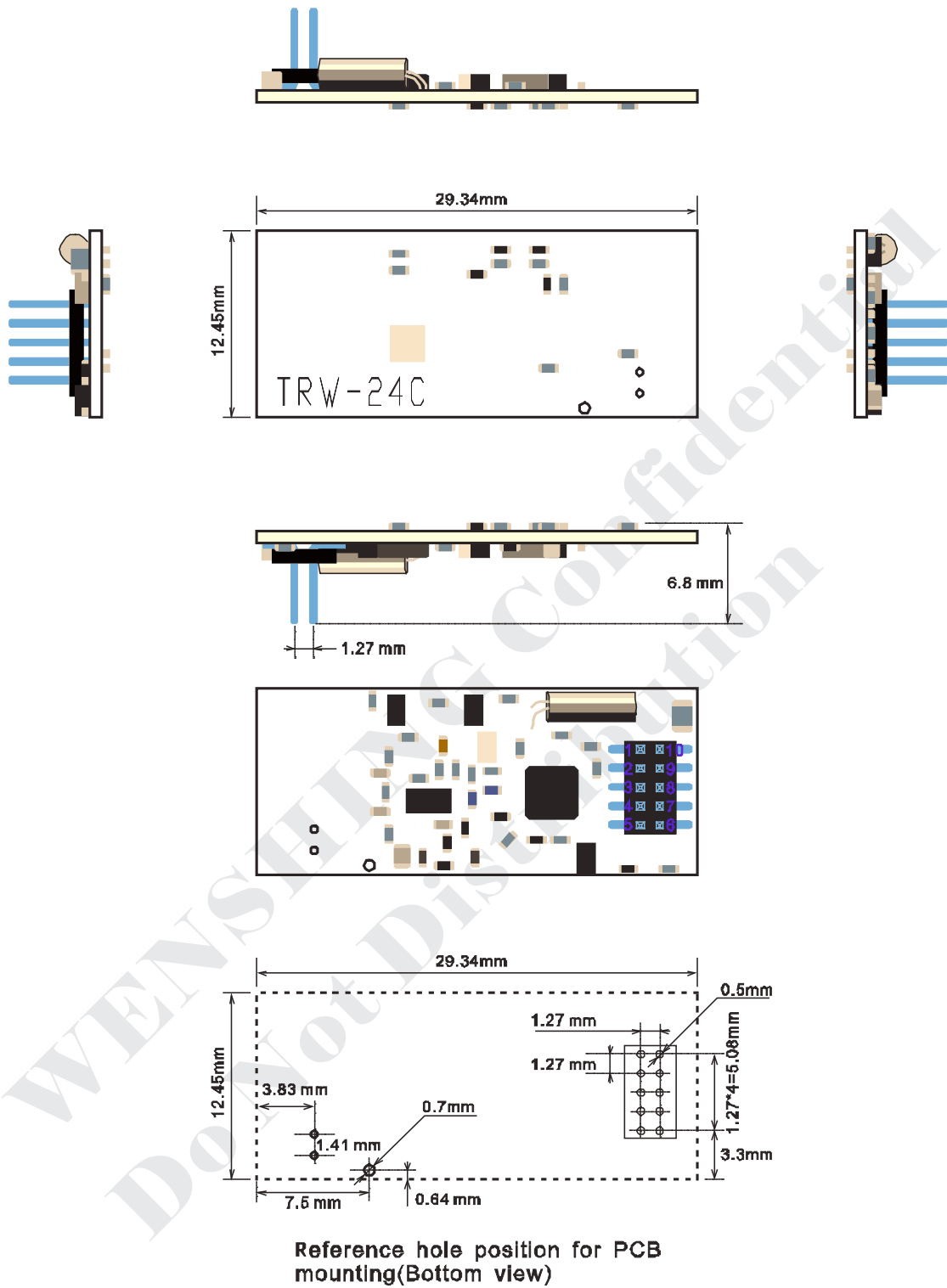
Parameter	Min	Type	Max	Unit	Condition/Note
Frequency Range	2400		2483.5	MHz	
Data Rate	1.2K		250	bps	
Power Down Modes		400		nA	Voltage regulator to digital part off, register values retained (sleep state)
Current Consumption, RX		19		mA	
Current Consumption, TX		90		mA	Transmit mode, +19dBm output power
Receiver Sensitivity	-106			dBm	
Transmit Output power			+19	dBm	Transmit mode

Pin Assignment

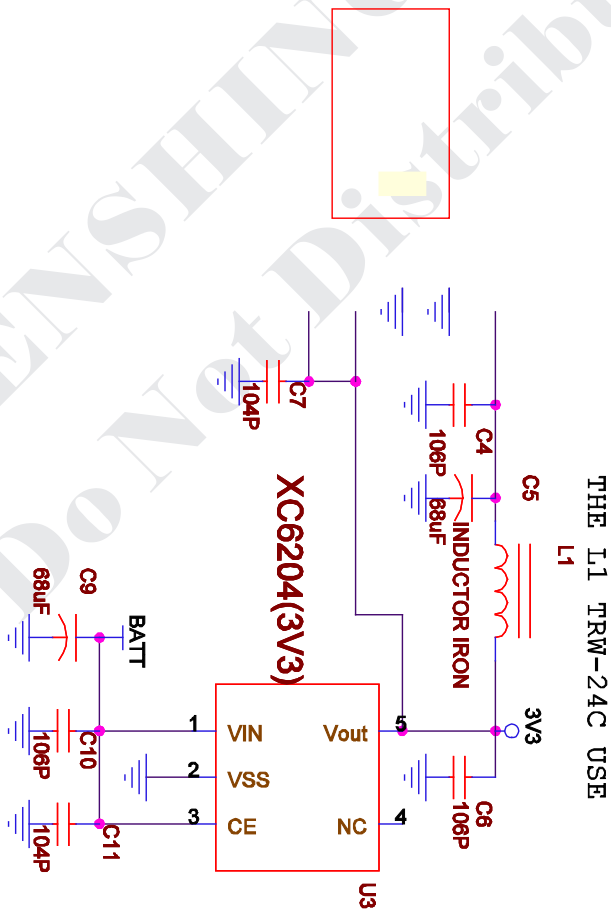
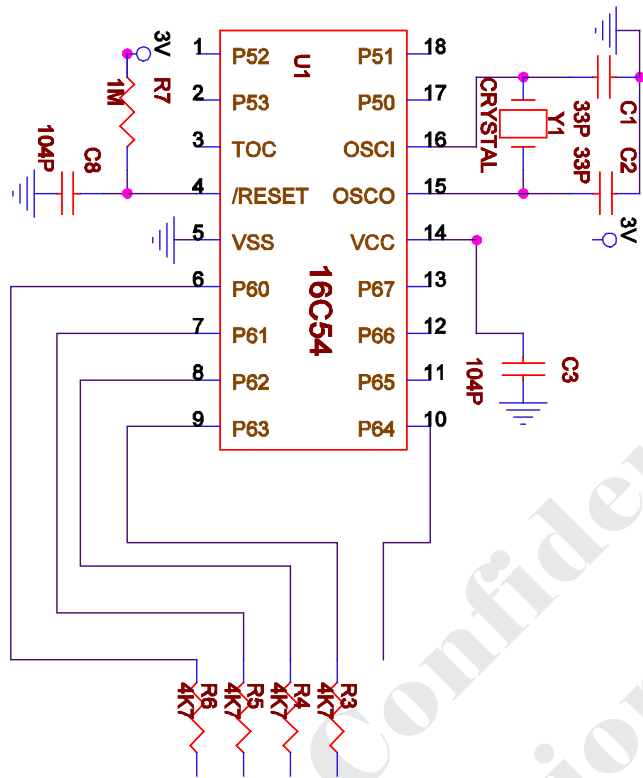


序號	名稱	類型	說明
1	DR	輸出	平常狀態為 0，當為 1 時，說明 RF 正在傳送或接收資料
2	CS	輸入	低電平有效，為選擇模組
3	MISO	輸出	從模組內讀取的資料，包含資料與配置資訊
4	SCLK	輸入	給 RF 讀寫提供時鐘
5	MOSI	輸入	向模組內寫入的資料，包含資料與配置資訊
6,9,10	VDD		電源正極
7,8	GND		電源負極

Size



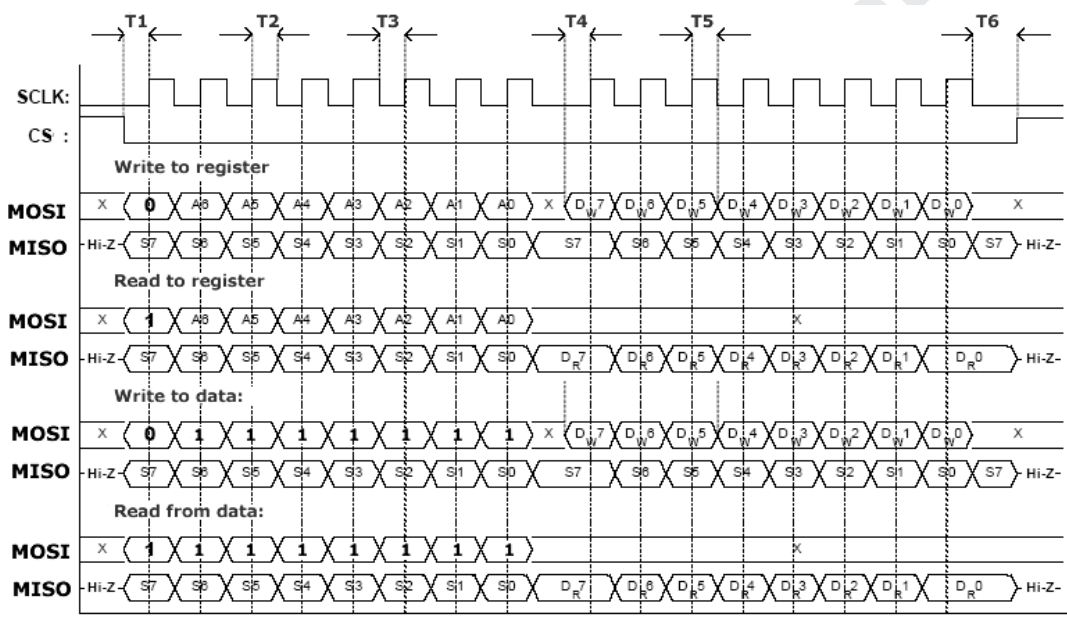
Demo Circuit



時序說明

TRW-24C 採用 SPI 的資料傳送方式，每次傳送時都要加上位址，位址的 bit8 為方向位，當為 0 時，則是寫入資料；當為 1 時，則是讀出資料，在配置時讀出的目的為了驗證寫入的正確性。

1. 配置時序說明：格式：address+data。
2. 資料傳送/接收時序說明：其時序跟配置一樣，只是位元元址跟寫讀的資料不同。
發送格式：0x7F+N 個 bytes (N<65)。
接收格式：0xFF+N 個 bytes (N<61)。
3. 寫入要發送的資料的位址為 0x7F，一次最多可寫入 64 個 bytes 的資料給 RF。讀出收到的資料位元址為 0xFF，一次最多從 RF 讀出 64 個 bytes。



Fsclk < 6.5MHz

T1 > 200us

T3 > 50ns

T5 > 20ns

T2 > 50ns

T4 > 80ns

T6 > 20ns

暫存器說明

此表是以基頻為 2410M，250K 的頻道間隔，選擇第 0 個頻道工作 2.4/10K/250K 的工作速率下的一些參考資料：

地址	2.4K	10K	250K	功能說明	
調變方式	FSK	FSK	MSK		
0x0D	0x5C	0x5C	0x5C	設置 TRW-24C 的基頻，工作基頻為 2410M	
0x0E	0xB1	0xB1	0xB1		
0x0F	0x3B	0x3B	0x3B		
0x13	0x43	0x43	0x43	13H[6:4]為前導碼的個數	
0x14	0x3B	0x3B	0x3B	13H[1:0]14H 為頻道間隔的基頻	
0x0A	0x00	0x00	0x00	設置為第幾個頻道	
0x0B	0x08	0x06	0x09	0BH[3:0]IF 的設定	
0x0C	0x00	0x00	0x00	固定	
0x10	0x86	0x78	0x2D	10H[7:4]接收的帶寬設定	
0x11	0x83	0x93	0x3B	設置模組的工作速率	
0x12	0x03	0x03	0x73	12H[7:4]選擇調制方式	
0x15	0x44	0x44	0x01		
0x22	0x10	0x10	0x10		
0x21	0x56	0x56	0xB6		
0x18	0x08	0x08	0x08		
0x19	0x16	0x16	0x1D		
0x1A	0x6C	0x6C	0x1C		
0x1B	0x03	0x43	0xC7		
0x1C	0x40	0x40	0x00		
0x1D	0x91	0x91	0xB2		
0x23	0xA9	0xA9	0xEA		
0x24	0x0A	0x0A	0x0A		
0x25	0x00	0x00	0x00		
0x26	0x11	0x11	0x11		
0x29	0x59	0x59	0x59		
0x2C	0x81	0x81	0x88		
0x2D	0x35	0x35	0x31		
0x2E	0x0B	0x0B	0x0B		
0x08	0x05	0x05	0x05		封包控制
0x02	0x06	0x06	0x06		固定
0x00	0x1B	0x1B	0x1B		
0x06	0xFF	0xFF	0xFF		

命令值說明

命令值	功能說明
0x30	重定 RF 模組
0x32	關閉 RF 的 OSC
0x33	自動鎖頻，鎖頻時間不超過 1ms
0x34	清除接收緩衝區
0x35	清除發送緩衝區
0x36	模組進入靜態模式
0x39	模組進入低功耗模式
0x3A	模組進入接收模式
0x3B	模組進入發送模式
0x7E	發送功率設定
0x7F/0xFF	寫發送/讀接收 緩衝區資料

1. TRW-24C 的基頻設定：0DH，0EH，0FH

$$FREQ[23:0] = F_{carrier} * 2^{16} / 26000000$$

例 1：工作頻率為 2401M，則 $FREQ[23:0] = 240100000 * 2^{16} / 26000000 = 6051997$
 $= 0x5C589D$

則 0DH = 0x5C 0EH = 0x58 0FH = 0x9D

例 2：工作頻率為 2411.5M，則 $FREQ[23:0] = 241150000 * 2^{16} / 26000000 = 6078464$
 $= 0x5CC000$

則 0DH = 0x5C 0EH = 0xC0 0FH = 0x00

2. 頻道與頻道的間隔頻率的設定：13H[1:0]14H

$$f_{CHANNEL} = 26000000 * (256 + 11H[7:0]) * 2^{10}[1:0] / 2^{18}$$

如果要設定頻道的間隔頻率為 250K，則 $14 = 0x3B$ ， $13[1:0] = 3$;

3. 頻道間隔的選擇：0x0A

$$f_{carrier} = f_{BASE} + f_{CHANNEL} * 0AH[7:0]$$

如果基頻為 2401， $f_{CHANNEL} = 250K$ ， $0x0A=00$ 的話，則當前的工作頻率為：2401M

如果基頻為 2401， $f_{CHANNEL} = 250K$ ， $0x0A=04$ 的話，則當前的工作頻率為：2402M

4. 前導碼的設定：13H[6:4]

13H[6:4] = 0 則為 2 個 bytes 的前導碼

13H[6:4] = 1 則為 3 個 bytes 的前導碼

13H[6:4] = 2 則為 4 個 bytes 的前導碼

13H[6:4] = 3 則為 6 個 bytes 的前導碼

13H[6:4] = 4 則為 8 個 bytes 的前導碼

13H[6:4] = 5 則為 12 個 bytes 的前導碼

13H[6:4] = 6 則為 16 個 bytes 的前導碼

13H[6:4] = 7 則為 24 個 bytes 的前導碼

5. 接收帶寬的設定：10H[7:4]，其下對應各值給出的對應頻率

10H[7:4] = 00 812K 10H[7:4] = 01 625K

10H[7:4] = 02 541K 10H[7:4] = 03 464K

10H[7:4] = 04 406K 10H[7:4] = 05 325K

10H[7:4] = 06 270K 10H[7:4] = 07 232K

10H[7:4] = 08 203K 10H[7:4] = 09 162K

10H[7:4] = 10 135K 10H[7:4] = 11 116K

10H[7:4] = 12 102K 10H[7:4] = 13 81K

10H[7:4] = 14 68K 10H[7:4] = 15 58K

6. IF 頻率的設定： $f_{IF} = 25390.625 * 0BH[3:0]$

例如：2.4K 的工作速率，0BH[3:0]=8，則 $f_{IF} = 203.125\text{KHz}$

7. 工作速率的設定：10H[3:0]，11H[7:0]，求最接近值：

$$\text{RATE} = (256 + 11H[7:0]) * 2^{10H[3:0]} * 26000000 / 2^{28}$$

例：如工作速率為 2.4K，則 11H = 0x83, 10H[3:0] = 0xx6

例：如工作速率為 1.2K，則 11H = 0x83, 10H[3:0] = 0xx5

8. 調製選擇方式：12H[7:4]

12H[7:4] = 0，說明選擇 2-FSK 模式。

12H[7:4] = 1，說明選擇 MSK 模式。

9. 調製度的設定：15H

a. 在 MSK 時，其值永遠都為 1。

b. 在 2-FSK 時，其調製度的計算公式如下，求最近值。

$$f_{dev} = (8 + 15H[2:0]) * 2^{15H[6:4]} * 198$$

10. 封包控制 08H

08H = 0x04，說明封包的長度是固定的，其長度的值存於 06H 中。

例：當 06H 中的值為 0x70 時，說明一個封包在發射和接收的 byte 的為 112 個，在發射和接收過程中需要分段發送跟接收才能把資料收完。

發射格式：0x7F+N 個 bytes (N = 06H 中的值)

08H = 0x05，說明封包的長度是可變的，其長度的值為發射的第一個 byte 的資料，06H 中的值為 0xFF。

發射格式：0x7F+封包長度值+N 個 bytes 數據(N = 封包的長度)

註：

- a. 當在發射狀態時，如果封包的長度 ≤ 64 ，則一次可以把資料寫入 TRW-24C 中，如果封包的長度 > 64 ，則需要分段傳送資料給 TRW-24C，因為 TRW-24C 內只有一個 64 個 bytes 的緩衝區。具體分段的方法如下：

10.1.1 第一次先寫入 64 個 bytes 給 TRW-24C。

10.1.2 等待 DR 腳變成高。

10.1.3 讀 0xFA 的值，如果讀到的值為 0x10，說明緩衝區中還有 16 個 bytes 沒送完，如果讀到的值為 0x20，說明緩衝區中還有 32 個 bytes 沒送完。

10.1.4 根據讀到 0xFA 的值，用戶可決定什麼時候再次傳送資料給 TRW-24C，但不能讀到其等於 0，否則會有錯。

10.1.5 如果再次傳送資料還沒有把資料送完，重複 10.1.13 與 10.1.14。

- b. 當在接收狀態時，等待其 DR 線變成高，如果封包的長度小於 61，則等待 DR 線變低後多讀兩個 bytes 的資料出來，多讀的兩個資料，前一個為 RSSI 的值，後一個為 CRC 的值，當後一個 CRC 的值 bit7 為 1 時，說明封包正確，否則錯誤，RSSI 值是一個有符號的值，如果讀到的數值越大，說明接收信號越好。請注意：0xFF 為 $-1 < 0$ 的。如果封包的長度大於 60，則具體讀數方法如下：

10.2.1 先等待 DR 線變成 1。

10.2.2 讀 0xFB 中的值如果等於 51，則從 TRW-24C 模組中讀出 50 個 bytes 資料(注 51/50 可由用戶自己定義，只是要比 0xFB 中的值小 1 就好了)。

10.2.3 重複 10.2.2，如果在所在讀 0xFB 時，其 DR 線變成低，說明此次的資料不夠 50 個 bytes，此時須退出讀 0xFB 的狀態，讀出剩餘的資料（註：讀出的資料比實際要多兩個）。

10.2.4 當 DR 線變成低後，怎樣知道還要讀多少的資料：

在固定封包的狀態下，由 0x06 中的值來決定，如果設為 N，則讀出的資料為 N+2 個 bytes。

在可變封包的狀態下，由讀到的第一個 byte 的值來決定，如果收到為 N，則讀出的資料為 N+3 個 bytes。

11. 功率設定：

當設定值為 0xFF/0xFE/0xBB/0xA9/0x7F/0x6E/0x97/0xC6/0x8D/0x55/0x93/0x46 對應的 dB 值為+19/+18/+16/+14/+12/+10/+8/+6/+4/+2/+0/-2dBm

設定功率的位址為 0x7E+8 個 bytes 的資料，8 個 bytes 設成一樣，選擇以上的功率值。

讀設定功率的值為 0xFE+8 個 bytes 的資料，8 個 bytes 是否跟設定的一樣。

程式說明

- | | |
|----------------|-------------------------------|
| 1 · 重定 TRW-24C | 副程式： Reset-TRW-24C 。 |
| 2 · 配置 TRW-24C | 副程式： Config-TRW-24C 。 |
| 3 · 選擇工作模式： | |
| 發射模式： | 副程式： TRW-24C-TxMode 。 |
| 接收模式： | 副程式： TRW-24C-RxMode 。 |
| 空閒模式： | 副程式： TRW-24C-TdleMode 。 |
| 4 · 資料傳遞： | |
| 發送資料 | 副程式： TRW-24C-Send-Data 。 |
| 接收資料 | 副程式： TRW-24C-Receive-Data 。 |
| 5 · 其他模式： | |
| 設置功率 | 副程式： TRW-24C-Set-Power 。 |
| 低功耗模式 | 副程式： TRW-24C-Low Power Mode 。 |

應用程式範例

```
/* ***** */
// 功能描述：如何使用TRW-24C模組
// 程式來源：文星電子股份有限公司
// 文星網址：http://www.wenshing.com.tw
// 更新日期：2007/01/16
// 版本：Ver1.00
/* ***** */

#include <C8051F320.H>
/* ***** */
/* ***** */
/* ***** */
/* ***** */

http://www.wenshing.com.tw ; http://www.rf.net.tw
```

```

/* ***** */
const unsigned char code TRW-24C-Table[68];
/* ***** */

sbit CLK = P1^1;
sbit MISO = P1^2;
sbit MOSI = P1^3;
sbit CS = P1^6;
sbit DR = P1^4;
sbit TX_LED = P2^1;
/* ***** */

/* 函數名：W-TRW-24C-byte */
/* 功能描述：向TRW-24C寫入一個byte的資料 */
/* 輸入：x */
/* 返回：無 */
/* ***** */

void W-TRW-24C-byte(char x)
{
    unsigned char i;
    for(i=0;i<8;i++)
    {
        CLK = 0;
        MOSI = 0;
        if(x&0x80)
            MOSI = 1;
        x<<=1;
        CLK = 1;
    }
    CLK = 0;
}
/* ***** */

/* 函數名：R-TRW-24C-byte */
/* 功能描述：從TRW-24C讀出一個byte的資料 */
/* 輸入：無 */
/* 返回：x */
/* ***** */

char R-TRW-24C-byte(void)
{
    unsigned char i,x;
    for(i=0;i<8;i++)

```

```

{
CLK = 0;
x <<= 1;
x &= 0xFE;
if(MISO)
x|= 0x01;
CLK = 1;
}
CLK = 0;
return(x);
}
/* ***** */
/* 函數名：Config-TRW-24C-byte */
/* 功能描述：配置TRW-24C模組的一個byte */
/* 調用函數：W-TRW-24C-byte */
/* R-TRW-24C-byte */
/* 函數說明：當address的第七位為1時，讀取其功能描述：配置TRW-24C的一個byte */
/* 輸入：address */
/* 返回：c_data */
/* ***** */
char Config-TRW-24C-byte(char address,char c_data)
{
CS = 0;
while(MISO); // 判斷MISO是否為低電平，否則等待變成低電平後退出
W-TRW-24C-byte(address);
if(address&0x80)
c_data = R-TRW-24C-byte();
else
W-TRW-24C-byte(c_data);
CS = 1;
return(c_data);
}
/* ***** */
/* 函數名：Reset-TRW-24C */
/* 功能描述：復位TRW-24C */
/* 調用函數：W-TRW-24C-byte */
/* 輸入：無 */
/* 返回：無 */
/* ***** */

```

```

void Reset-TRW-24C(void)
{
unsigned char i;
CS = 1;
for(i=0;i<100;i++);
CS = 0;
for(i=0;i<200;i++);
CS = 1;
for(i=0;i<200;i++);
CS = 0;
while(MISO); // 判斷MISO是否為低電平，否則等待變成低電平後退出
W-TRW-24C-Byte(0x30);
CS = 1;
}
/* ***** */
/* 函數名 : TRW-24C-Set-Power */
/* 功能描述 : 設置TRW-24C模組功率 */
/* 調用函數 : W-TRW-24C-byte */
/* 函數說明 : 一次要寫8個bytes的數據 */
/* 輸入 : x */
/* 返回 : 無 */
/* ***** */
void TRW-24C-Set-Power(char x)
{
unsigned char i;
CS = 0;
while(MISO);
for(i=0;i<8;i++)
W_TRW_24C_byte(x);
CS = 1;
}
/* ***** */
/* 函數名 : Config-TRW-24C */
/* 功能描述 : 配置TRW-24C */
/* 調用函數 : Config-TRW-24C-byte */
/* : W-TRW-24C-byte */
/* 輸入 : 無 */
/* 返回 : 無 */
/* ***** */

```

```

void Config-TRW-24C(void)
{
    unsigned char i,x;
    do
    {
        for(i=0;i<68;)
        Config-TRW-24C-byte(TRW-24C-Table[i++],TRW-24C-Table[i++]);
        TRW-24C-Set-Power(0xFF); // 設置最高功率發射
        CS = 0;
        while(MISO);
        W-TRW-24C-byte(0x33);
        CS = 1;
        CS = 0;
        while(MISO);
        W-TRW-24C-byte(0x37);
        CS = 1;
        for(i=0;i<200;i++);
        i = Config-TRW-24C-byte(0x80,x);
    }while(i!=0x1B);
    }
    /* ***** */
    /* 函數名：TRW-24C-Tx Mode */
    /* 功能描述：配置TRW-24C工作在發射模式 */
    /* 函數說明：此函數可使TRW-24C直接從接收模式或空閒模式切回發射模式 */
    /* 調用函數：W-TRW-24C-byte */
    /* 輸入：無 */
    /* 返回：無 */
    /* ***** */
    void TRW-24C-Tx Mode(void)
    {
        CS = 0;
        while(MISO);
        W-TRW-24C-byte(0x36);
        CS = 1;
        CS = 0;
        while(MISO);
        W-TRW-24C-byte(0x3B);
        CS = 1;
        CS = 0;
    }

```

```

while(MISO);
W-TRW-24C-byte(0x35);
CS = 1;
}
/* ***** */
/* 函數名：TRW-24C-Rx Mode */
/* 功能描述：配置TRW-24C模組工作在接收模式 */
/* 函數說明：此函數可使TRW-24C模組直接從發射模式或發射模式切回發射模式 */
/* 調用函數：W-TRW-24C-byte */
/* 輸入：無 */
/* 返回：無 */
/* ***** */
void TRW-24C-Rx Mode(void)
{
CS = 0;
while(MISO);
W-TRW-24C-byte(0x36);
CS = 1;
CS = 0;
while(MISO);
W-TRW-24C-byte(0x3A);
CS = 1;
CS = 0;
while(MISO);
W-TRW-24C-byte(0x34);
CS = 1;
}
/* ***** */
/* 函數名：TRW-24C-Idle Mode */
/* 功能描述：配置TRW-24C模組工作在空閒模式 */
/* 函數說明：此函數可使TRW-24C模組直接從發射模式或接收模式切回空閒模式，應用場合屬於不想接收，但要求快速返回發射或接收狀態 */
/* 調用函數：W-TRW-24C-byte */
/* 輸入：無 */
/* 返回：無 */
/*
*****
*/
void TRW-24C-Idle Mode(void)

```



```

{
CS = 0;
while(MISO);
W-TRW-24C-byte(0x36);
CS = 1;
}
/* ***** */
void Send_FIFO_Pointer(void)
{
unsigned char i;
while(Config-TRW-24C-byte(0xFA,0x00)>0x10)
for(i=0;i<200;i++);
}
/* ***** */
/* 函數名：TRW-24C-Send-Data */
/* 功能描述：讓TRW-24C發送資料 */
/* 函數說明：固定發送資料88H，在發送過程中，盡可能的避免一次性多個bytes的資料為
0x00,0xFF，否則接收有錯或收不到。
/* 調用函數：W-TRW-24C-byte */
/* 輸入：x指一個封包需要發送的byte數，以可變封包為例，但一次封包小於110個bytes */
/* 返回：無 */
/* ***** */
void TRW-24C-Send-Data(unsigned char x)
{
unsigned char i;
TRW-24C-Tx Mode();
CS = 0;
while(MISO);
W-TRW-24C-byte(0x7F);
W-TRW-24C-byte(x);
if(x<65)
{
for(i=0;i<x;i++)
W-TRW-24C-byte(0x88); // 發送88H的資料
CS = 1;
}
else
{
for(i=0;i<60;i++)

```

```

W-TRW-24C-byte(0x88); // 發送88H的資料
CS = 1;
while(!DR);
Send_FIFO_Pointer();
CS = 0;
while(MISO);
W-TRW-24C-byte(0x7F);
for(i=0;i<(x-60);i++)
W-TRW-24C-byte(0x88); // 發送88H的資料
CS = 1;
}
while(!DR); // 等待發射完成，發射完成後，自動進入IDLE模式，如果發送完後想進入接收模式
的話，就寫TRW-24C-Rx Mode()函數
}
/* ***** */
/* 函數名：Receive_FIFO_Pointer */
/* 功能描述：讀TRW-24C接收資料指標，當收到51個資料時，退出 */
/* 調用函數：W-TRW-24C-byte */
/* 輸入：無 */
/* 返回：0 ,說明不到50個Bytes的資料就接收結束 */
/* 返回：1 ,說明有收到50個bytes的資料 */
/* ***** */
unsigned char Receive_FIFO_Pointer(void)
{
unsigned char i = 0;
while(Config-TRW-24C-byte(0xFB,0x00)<51)
{
TX_LED = ~TX_LED;
for(i=0;i<200;i++)
if(!DR)
i = 210;
if(i==211)
break; //退出while語句
}
if(i==211)
return(0);
else
return(1);
}

```

```

/* ***** */
/* 函數名：TRW-24C-Receive-Data */
/* 功能描述：讓TRW-24C接收資料 */
/* 函數說明：以可變封包為例 */
/* 調用函數：W-TRW-24C-byte */
/* 輸入：無 */
/* 返回：0 ,說明沒有收到資料 */
/* 返回：1 ,說明有收到錯誤的資料 */
/* 返回：2 ,說明有收到正確的資料 */
/* ***** */

char TRW-24C-Receive-Data(void)
{
    unsigned char i = 0, x = 0, y = 0, z, CRC_Value;
    unsigned char xdata RF_Buffer[100], RF_Pointer = 0;
    while(DR)
    {
        if(Receive_FIFO_Pointer())
        {
            x++;
            CS = 0;
            while(MISO);
            W-TRW-24C-byte(0xFF);
            i = 0;
            if(x==0)
            {
                RF_Pointer = 0;
                y = R_TRW-24C_byte();
                y ++ ; // 在接收資料時，要比實際發射的資料多兩個，前一個為RSSI的值，後一個BIT7為CRC
                值，如CRC正確，則BIT7 = 1，否則為0
                RF_Buffer[RF_Pointer++] = y-1;
            }
            for(;i<50;i++)
                RF_Buffer[RF_Pointer++] = R_TRW_24C_byte();
            CS = 1;
            z = y-50;
        }
        else
        {
            CS = 0;

```

```

while(MISO);
W_TRW_24C_byte(0xFF);
if(x==0)
{
RF_Pointer = 0;
y = R_TRW_24C_byte();
y ++;
RF_Buffer[RF_Pointer++] = y-1;
for(i=0;i<y;i++)
RF_Buffer[RF_Pointer++] = R_TRW_24C_byte();
}
else
{
for(i=0;i<z;i++)
RF_Buffer[RF_Pointer++] = R_TRW_24C_byte();
}
CRC_Value = R_TRW_24C_byte();
RF_Buffer[RF_Pointer++] = CRC_Value;
CS = 1;
z = 0x7F;
}
}
if(z == 0x7F)
{
if(CRC_Value&0x80)
return(2);
else
return(1);
}
else
return(0);
}
/* ***** */
/* 函數名：TRW-24C-Low Power Mode */
/* 功能描述：讓TRW-24C工作在低功耗狀態 */
/* 調用函數：W-TRW-24C-byte */
/* 輸入：無 */
/* 返回：無 */
/* ***** */

```

```

void TRW-24C_Low Power Mode(void)
{
CS = 0;
while(MISO);
W_TRW-24C-byte(0x36);
CS = 1;
Config-TRW-24C-Byte(0x00,0x6F);
CS = 0;
while(MISO);
W_TRW-24C-byte(0x39);
CS = 1;
}
/* ***** */

void Init_MCU_Status (void)
{
P1MDOUT = 0x4B;
P2MDOUT = 0x01;
P1 = 0xF4;
XBR1 = 0x40;
OSCICN |= 0x03; // 工作頻率為24MHz
}
/* ***** */

/* 函數名：Main函數 */
/* MCU Body：C8051F32x */
/* 輸入：無 */
/* 返回：無 */
/* ***** */

void main(void)
{
unsigned int x;
unsigned char i;
PCA0MD = 0x00;
for(x=0;x<30000;x++);
Init_MCU_Status();
Reset-TRW-24C();
Config-TRW-24C();
for(x=0;x<1000;x++);
while(1)
{
http://www.wenshing.com.tw ; http://www.rf.net.tw

```

```

TRW-24C-Send-Data(90);
TRW-24C-Rx Mode();
for(x=0;x<60000;x++)
TRW-24C-Receive-Data();
}
}
const unsigned char code TRW_24C_Table[68] =
{
0x0D,0x5C, // 2.4K 2-FSK 2410M
0x0E,0x4e,
0x0F,0xc5,
0x0B,0x08,
0x0C,0x00,
0x10,0x86,
0x11,0x83,
0x12,0x03,
0x13,0x43,
0x14,0x3B,
0x0A,0x00,
0x15,0x44,
0x22,0x10,
0x21,0x56,
0x18,0x08,
0x19,0x16,
0x1A,0x6C,
0x1B,0x03,
0x1C,0x40,
0x1D,0x91,
0x23,0xA9,
0x24,0x0A,
0x25,0x00,
0x26,0x11,
0x29,0x59,
0x2C,0x81,
0x2D,0x35,
0x2E,0x0B,
0x08,0x05,
0x07,0x04,
0x02,0x06,

```

```
0x00,0x1B,  
0x09,0x00,  
0x06,0xFF  
/* 0x0D,0x5C, ; 250K MSK 2410  
0x0E,0xB1,  
0x0F,0x3B,  
0x0B,0x09,  
0x0C,0x00,  
0x10,0x2D,  
0x11,0x3B,  
0x12,0x73,  
0x13,0x32,  
0x14,0xF8,  
0x0A,0x00,  
0x15,0x01,  
0x22,0x10,  
0x21,0x56,  
0x18,0x08,  
0x19,0x1D,  
0x1A,0x1C,  
0x1B,0xC7,  
0x1C,0x00,  
0x1D,0xB2,  
0x23,0xEA,  
0x24,0x0A,  
0x25,0x00,  
0x26,0x11,  
0x29,0x59,  
0x2C,0x88,  
0x2D,0x31,  
0x2E,0x0B,  
0x08,0x05,  
0x07,0x04,  
0x02,0x06,  
0x00,0x1B,  
0x09,0x00,  
0x06,0xFF  
*/;
```