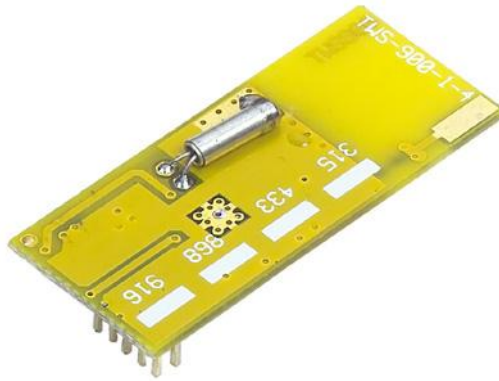


---

Wireless Low Power RF Transmitter Module (GFSK)

---



#### Version History

Version	Date	Changes
V1.01	May. 05,2009	1 <sup>st</sup> . Edition
V1.02	Aug. 26, 2009	2 <sup>nd</sup> . Edition

## Specification

● UHF Wireless Data Transmitter	● Low Power
● Single 1.8V to 3.6V Supply	● Hi Sensitivity: -110dBm (1200bps)
● Antenna on board	● Distance above 300M
● 315 / 433 / 868 and 915 MHz ISM/SRD band systems	
● Application Range : Remote Metering、 Wireless Security Systems 、 Automatic Meter 、 Reading、 Home Automation	

## Absolute Maximum Rating

Under no circumstances must the absolute maximum ratings given in Table 1 be violated. Stress exceeding one or more of the limiting values may cause permanent damage to the device.

Parameter	Min	Max	Unit	Condition
Supply Voltage	-0.3	3.6	V	All supply pins must have the same voltage
Voltage on any digital pin	-0.3	VDD+0.3 Max 3.6	V	
Voltage on the pins RF_P, RF_N and DCOUPL	-0.3	2.0	V	
Input RF level		10	dBm	
Storage Temperature Range	-50	+150	°C	
Solder Reflow Temperature		265	°C	According to IPC/JEDEC J-STD-020C

## Operating Condition

The operating conditions for TWS- 900C listed Table 2 in below.

Parameter	Min	Max	Unit	Condition
Operating Temperature	-10	+70	°C	
Operating Voltage	1.8	3.6	V	All supply pins must have the same voltage

## Electrical Specification

T<sub>c</sub> = 25°C, VDD = 3.0V if nothing else stated. Measured on TI's CC1150 reference design.

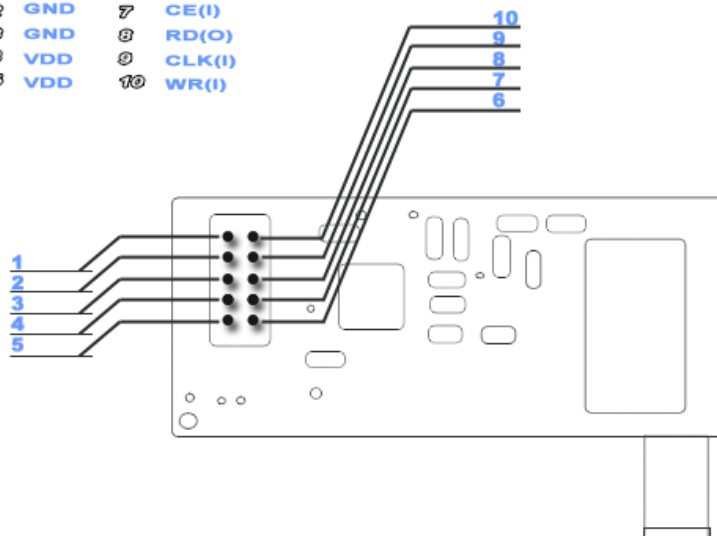
Parameter	Type	Unit	Condition
Current consumption	200	nA	Voltage regulator to digital part off, register values retained (SLEEP state)
	180	μA	Voltage regulator to digital part on, all other modules in power down (XOFF state)
	1.4	mA	Only voltage regulator to digital part and crystal oscillator running (IDLE state)
	8.0	mA	Only the frequency synthesizer running (after going from IDLE until reaching RX or TX states, and frequency calibration states)
Current consumption, 315MHz	26.3 17.6 14.5 11.2	mA	Transmit mode, +10dBm output power Transmit mode, 5dBm output power Transmit mode, 0dBm output power Transmit mode, -10dBm output power
Current consumption, 433MHz	26.4 18.0 14.9 13.4	mA	Transmit mode, +10dBm output power Transmit mode, 5dBm output power Transmit mode, 0dBm output power Transmit mode, -10dBm output power
Current consumption, 868/915MHz	28.7 18.8 15.9 13.7	mA	Transmit mode, +10dBm output power Transmit mode, 5dBm output power Transmit mode, 0dBm output power Transmit mode, -10dBm output power

## General Characteristic

Parameter	Min	Type	Max	Unit	Condition
Frequency Range	300		348	MHz	
	400		464	MHz	
	800		928	MHz	
Data rate	1.2		500	Kbps	Modulation formats supported: (Shaped) MSK (also known as differential offset QPSK) up to 500kbps 2-FSK up to 500kbps GFSK and OOK/ASK (up to 250kbps) Optional Manchester encoding (halves the data rate).

## Pin Assignment:

1	VDD	6	FLAG(O)
2	GND	7	CE(I)
3	GND	8	RD(O)
4	VDD	9	CLK(I)
5	VDD	10	WR(I)



Pin	Function
1	VDD
2	GND
3	GND
4	VDD
5	VDD
6	FLAG(O)
7	CE(I)
8	RD(O)
9	CLK(I)
10	WR(I)

## 使用說明

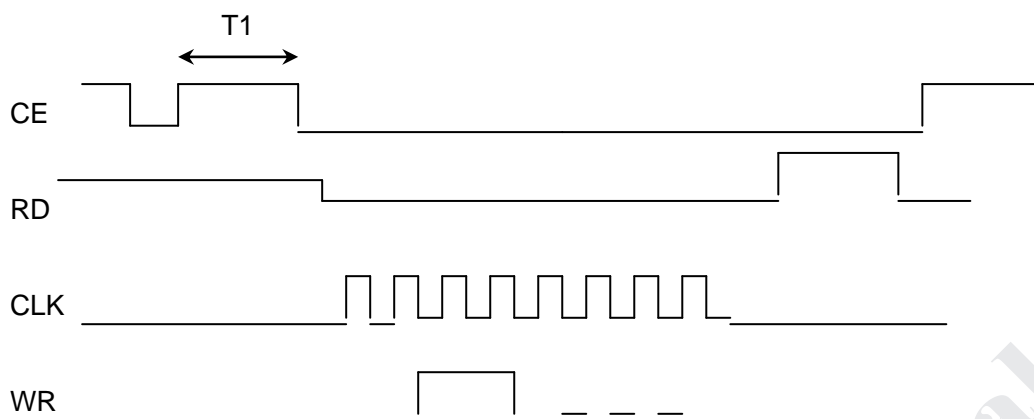
- 1 此說明係配合 TRW-400B 收發雙向模組使用，如果要配合 TRW-900C 使用，請參考 TRW-900C Data Sheet。
- 2 工作速率：2.4/4.8/9.6/19.2/38.4/76.8/153.6K
- 3 工作電流：
  - 3.1 +10dBm 輸出為 26.4mA
  - 3.2 0dBm 輸出為 15mA
- 4 配置說明：
  - 4.1 程式上電後，復位 TWS-900C。
  - 4.2 延時 1ms，配置 TWS-900C。
  - 4.3 配置完成後，至少延時 5ms 才能發送資料給 TWS-900C。
- 5 為了使程式簡單，一次封包的個數不能超過 64 個 BYTE（原則上可以有無限個）。
- 6 引腳說明：
  - 6.1 Pin1：VDD      Pin10：WR (O)
  - 6.2 Pin2：GND      Pin9：CLK (O)
  - 6.3 Pin3：GND      Pin8：RD (I)
  - 6.4 Pin4：VDD      Pin7：FLAG (I)
  - 6.5 Pin5：VDD      Pin6：CE (O)
- 7 頻率的計算公式：

頻率值 = 實際工作頻率\*2<sup>16</sup>/26000000，頻率值的位址為：0x0D,0x0E,0x0F 三個地址。

例：工作頻率為 434M，則頻率值：

$$= 434 * 2^{16} / 26 = 1093947 = 0x10B13B，則：$$

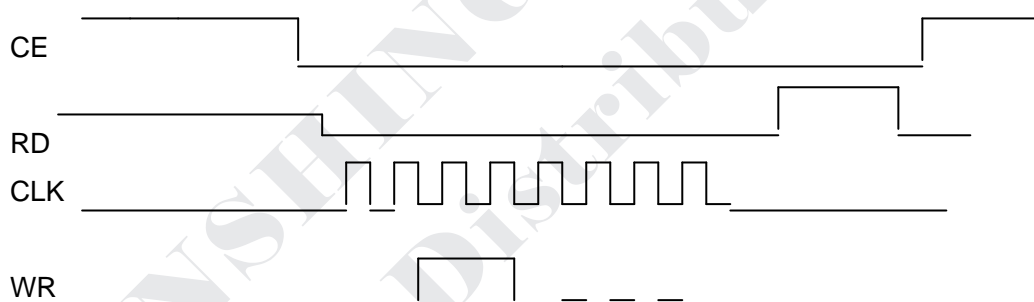
0x0D = 0x10  
0x0E = 0xB1  
0x0F = 0x3B
- 8 復位 TWS-900：
  - 8.1 設置 CE 為高/低，延時 10us 以上。
  - 8.2 設置 CE 為高，延時 T1 的時間大於或等於 50us。
  - 8.3 重定的指令為 0x30。
  - 8.4 當 CE 變成低後（T1 的時間過後），判斷 RD 的狀態：
    - 8.4.1 如為 1，一直等待（大於 20ms 的話，說明模組已損壞）。
    - 8.4.2 如為 0，則把重定指令發送給 RF。
  - 8.5 發送完重定的指後，等待 RD 的狀態：
    - 8.5.1 如為 1，等待其為 0 後，把 CE 高為 1。
    - 8.5.2 如為 0，把 CE 高為 1。
  - 8.6 復位完成，時序如下：



9 寫入命令值給 RF 模組：

- 9.1 配置完成後寫入兩個命令值：0x37/0x33
- 9.2 發送之前寫入兩個命令值：0x3B/0x35
- 9.3 進入低功耗之前寫入兩個命令值：0x39/0x32
- 9.4 從低功耗到正常工作：重定，配置。
- 9.5 當 CE 變低後須等待 RD 變成低後才能傳送資料給 RF。

寫入命令值的時序



10 配置資料給 RF，其格式：位址+資料。

Address	Data Rate						
	2.4	4.8	9.6	19.2	38.4	76.8	153.6
02H	0x06	0x06	0x06	0x06	0x06	0x06	0x06
04H	0x55	0x55	0x55	0x55	0x55	0x55	0x55
05H	0x55	0x55	0x55	0x55	0x55	0x55	0x55
06H							
08H	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0AH	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0DH	0x10	0x10	0x10	0x10	0x10	0x10	0x10
0EH	0xB1	0xB1	0xB1	0xB1	0xB1	0xB1	0xB1
0FH	0x3B	0x3B	0x3B	0x3B	0x3B	0x3B	0x3B

10H	0x86	0x87	0x88	0x89	0x8A	0x8B	0x8C
11H	0x83	0x83	0x83	0x83	0x83	0x83	0x83
12H	0x03	0x03	0x03	0x03	0x03	0x03	0x03
13H	0x02	0x02	0x02	0x02	0x02	0x02	0x02
14H	0xF8	0xF8	0xF8	0xF8	0xF8	0xF8	0xF8
15H	0x02	0x04	0x14	0x24	0x34	0x43	0x53
18H	0x08	0x08	0x08	0x08	0x08	0x08	0x08
22H	0x10	0x10	0x10	0x10	0x10	0x10	0x10
23H	0xA9	0xA9	0xA9	0xA9	0xA9	0xEA	0xEA
24H	0x2A	0x2A	0x2A	0x2A	0x2A	0x2A	0x2A
25H	0x00	0x00	0x00	0x00	0x00	0x00	0x00
26H	0x11	0x11	0x11	0x11	0x11	0x11	0x11
29H	0x59	0x59	0x59	0x59	0x59	0x59	0x59
2CH	0x81	0x81	0x81	0x81	0x81	0x88	0x88
2DH	0x35	0x35	0x35	0x35	0x35	0x31	0x31
2EH	0x0B	0x0B	0x0B	0x0B	0x0B	0x0B	0x0B

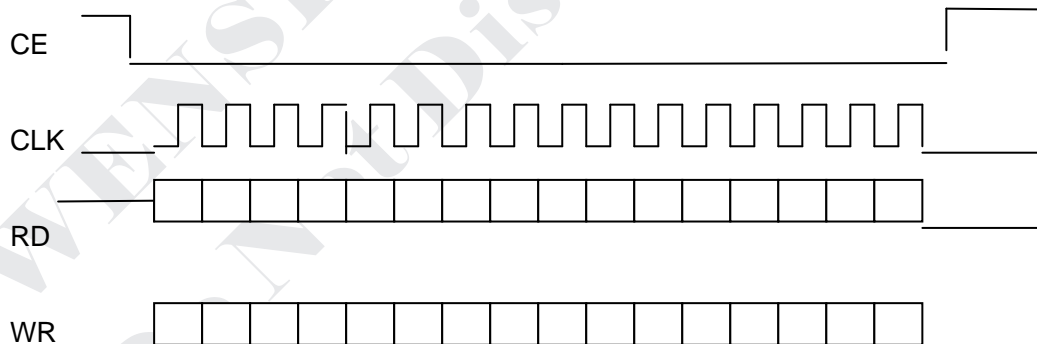
10.1 特殊說明：

10.1.1 0DH, 0EH, 0FH 為工作頻率的位址。

10.1.2 06H 為一次封包的 BYTE 數，如一次傳送 10 個，則為 0AH。

10.2 重定後，向 RF 寫完上表的配置字後，寫入兩個命令值：0x37/0x33。

10.3 如想知道 06H 內的資料是否正確，則把位址改成 0x80，讀 RD 上的資料即可，讀其他位址都一樣。其時序跟寫入一樣，時序如下：



10.4 為了配合 TRW-400 收發雙向模組，發送資料須符合以下格式，否則 TRW-400 將會收不到資料：

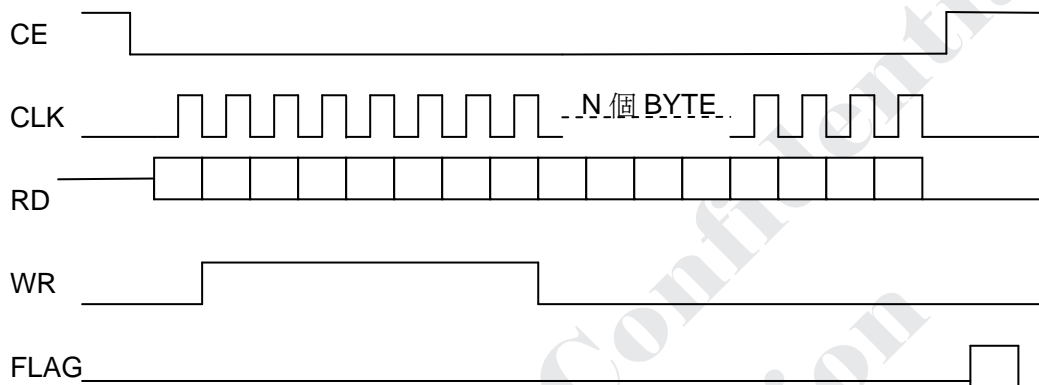
10.4.1 位址+N 個 BYTE 的資料+核對碼。

10.4.2 地址 (n 個 BYTE) 看 RX 設為多少個就發多少個，其值跟 RX 設的一樣。

10.4.3 核對碼 (為兩個 BYTE，兩個 BYTE 值是相同的)， $55 \oplus \text{位址} \oplus \text{資料}$ 。例：發送位址為 4 個 BYTE，其值為 0x12345678，數據為 8 個 BYTE，其值為 0x01020304050607，則核對碼為：0x5D5D。

- 10.4.4 發送地址為 0x7F+0x12345678+0x01020304050607+0x5D5D。
- 10.4.5 當 CE 變成低後，檢查 RD 線是否為 0，如是傳送位址，否則等待。
- 10.4.6 每次要發送資料前寫入兩個命令給 RF 模組。0x3B/0x35。
- 10.4.7 當傳送完所有資料後，檢查 FLAG 是否為 1：
  - 如為 1，等待其變低後就可傳送下一次的封包。
  - 如為 0，等待其變高後，再等待其變低後就可傳送下一次的封包。

發送資料的時序如下



### 應用程式範例

```

*****
// 復位Txx_xxxC模組
// *****
void RESET_Txx_xxxC (void){
  unsigned char i;
  CE = 1;
  for(i=0;i<100;i++);
  CE = 0;
  for(i=0;i<200;i++);
  CE = 1;
  for(i=0;i<200;i++);
  CE = 0;
  while(RD);
  SPI0DAT = 0x30;
  while(!SPIF);
  SPIF = 0;
  CE = 1;
}

```



```

// *****
// 輸入為ACC = 30H
// *****

RESET_Txx_xxxC:
SETB CE
MOV R2 ,#10
DJNZ R2 ,\$
CLR CE
DJNZ R2 ,\$
SETB CE
DJNZ R2 ,\$
CLR CE
JB RD ,\$
MOV R2 ,#8
RESET_Txx_xxxC_0:
CLR CLK
CLR WR
JNB ACC.7 ,RESET_Txx_xxxC_1
SETB WR
RESET_Txx_xxxC_1:
SETB CLK
RL A
DJNZ R2 ,RESET_Txx_xxxC_0
JB RD ,\$
SETB CE
RET
// *****
// 當Address.7 = 1時，從Txx_xxxC讀出一個Byte的資料。
// 當Address.7 = 0時，向Txx_xxxC寫入一個Byte的資料。
// *****

char RW_Txx_xxC(char Address,char data0)
{
unsigned char i=0;
CE = 0;
while(RD);
SPI0DAT = Address;
while(!SPIF);
SPIF =0;
http://www.wenshing.com.tw ; http://www.rf.net.tw

```

```

SPI0DAT = data0;
while(!SPIF);
SPIF =0;
CE = 1;
return(SPI0DAT);
}
// *****
// 輸入A，B，其中A為位址，B為資料
// 功能：向Txx_xxC寫入一個配置單元或資料。
// *****

W_Txx_xxC:
CLR CE
JB RD,$
MOV R2,#8
ANL A,#07FH
W_Txx_xxC_0:
CLR CLK
CLR WR
JNB ACC.7,W_Txx_xxC_1
SETB WR
W_Txx_xxC_1:
SETB CLK
RL A
DJNZ R2,W_Txx_xxC_0
MOV R2,#8
MOV A,B
W_Txx_xxC_2:
CLR CLK
CLR WR
JNB ACC.7,W_Txx_xxC_3
SETB WR
W_Txx_xxC_3:
SETB CLK
RL A
DJNZ R2,W_Txx_xxC_2
SETB CE
RET
// *****

```

```

// 輸入A，其中A為位址
// 功能：從Txx_xxC讀出一個配置單元或資料。
// *****
R_Txx_xxC:
CLR CE
JB RD,$
MOV R2,#8
ORL A,#080H
R_Txx_xxC_0:
CLR CLK
CLR WR
JNB ACC.7,R_Txx_xxC_1
SETB WR
R_Txx_xxC_1:
SETB CLK
RL A
DJNZ R2,R_Txx_xxC_0
MOV R2,#8
R_Txx_xxC_2:
CLR CLK
CLR ACC.7
JNB RD,R_Txx_xxC_3
SETB ACC.7
R_Txx_xxC_3:
SETB CLK
RL A
DJNZ R2,W_Txx_xxC_2
SETB CE
RET
// *****
void Write_Command(char command)
{
CE = 0;
while(RD);
SPI0DAT = command;
while(!SPIF);
SPIF =0;
CE = 1;
http://www.wenshing.com.tw; http://www.rf.net.tw

```

```

}
// *****
// 輸入：ACC
// 功能：向Txx_xxxC模組寫入一個命令值
// *****

Write_Command:
CLR CE
JB RD,$
MOV R2,#8
Write_Command_0:
CLR CLK
CLR WR
JNB ACC.7,Write_Command_1
SETB WR
Write_Command_1:
SETB CLK
RL A
DJNZ R2,Write_Command_0
SETB CE
RET
// *****
// 當Address.7 = 1時，從Txx_xxxC讀出一串數據。
// 當Address.7 = 0時，向Txx_xxxC寫入一串資料。
// *****

void RW_Txx_xxC_String(char Address,char *data0,char x)
{
char i;
CE = 0;
while(RD);
SPI0DAT = Address;
while(!SPIF);
SPIF =0;
for(i=0;i<x;i++)
{
SPI0DAT = *data0;
while(!SPIF);
SPIF =0;
*data0++ = SPI0DAT;
}
}

```

```

}
CE = 1;
}
// *****

Write_8Bit:
MOV R2 ,#8
Write_8Bit_0:
CLR CLK
CLR WR
JNB ACC.7 ,Write_8Bit_1
SETB WR
Write_8Bit_1:
SETB CLK
RL A
DJNZ R2 ,Write_8Bit_0
RET
// *****

Read_8Bit:
MOV R2 ,#8
Read_8Bit_0:
CLR CLK
CLR ACC.7
JNB RD ,Read_8Bit_1
SETB ACC.7
Read_8Bit_1:
SETB CLK
RL A
DJNZ R2 ,Read_8Bit_0
RET
// *****

// 輸入：A，R0，R7
// A 代表起始地址
// R0 代表要傳送資料的起始位址
// R7 代表要傳送多少個Byte。
// *****
// *****

W_Txx_xxxC_String:
CLR CE
http://www.wenshing.com.tw; http://www.rf.net.tw

```

```

LCALL Write_8Bit
W_Txx_xxxC_String_0:
MOV A ,@R0
INC R0
LCALL Write_8Bit
DJNZ R7 ,W_Txx_xxxC_String_0
SETB CE
RET
// *****
// 輸入：A，R0，R7
// A 代表起始地址
// R0 代表讀出RF的資料保存的間址單元
// R7 代表要讀多少個Byte。
// *****
R_Txx_xxxC_String:
CLR CE
LCALL Write_8Bit
R_Txx_xxxC_String_0:
LCALL Read_8Bit
MOV A ,@R0
INC R0
DJNZ R7 ,R_Txx_xxxC_String_0
SETB CE
RET
// *****
// 配置TWS-900C
// *****
void Config_TWS-900C(void)
{
int i;
RW_Txx_xxC(0x0D ,0x10); //0D
RW_Txx_xxC(0x0E ,0xB1); //0E
RW_Txx_xxC(0x0F ,0x3B); //0F
RW_Txx_xxC(0x04 ,0x55); //04
RW_Txx_xxC(0x05 ,0x55); //05
RW_Txx_xxC(0x10 ,0x87); //10
RW_Txx_xxC(0x11 ,0x83); //11
RW_Txx_xxC(0x12 ,0x03); //12
http://www.wenshing.com.tw ; http://www.rf.net.tw

```

```

RW_Txx_xxC(0x13 ,0x22); //13
RW_Txx_xxC(0x14 ,0xF8); //14
RW_Txx_xxC(0x0a ,0x00); //0A
RW_Txx_xxC(0x15 ,0x12); //15
RW_Txx_xxC(0x22 ,0x10); //22
RW_Txx_xxC(0x18 ,0x08); //18
RW_Txx_xxC(0x23 ,0xA9); //23
RW_Txx_xxC(0x24 ,0x2A); //24
RW_Txx_xxC(0x25 ,0x00); //25
RW_Txx_xxC(0x26 ,0x11); //26
RW_Txx_xxC(0x29 ,0x59); //29
RW_Txx_xxC(0x2A ,0x81); //2A
RW_Txx_xxC(0x2B ,0x35); //2B
RW_Txx_xxC(0x2C ,0x0B); //2C
RW_Txx_xxC(0x08 ,0x04); //08
RW_Txx_xxC(0x02 ,0x06); //02
RW_Txx_xxC(0x06 ,0x0A); //06 = Adress+RF Byte+2個CRC 4+4+2
RW_Txx_xxC_String(0xC0,&Receive_RF_Data,0x30);
// 從00H單元讀出0x30個資料,更改如下:
// MOV A ,#0C0H
// MOV R0 ,#30H
// MOV R7 ,#30H
// LCALL R_Txx_xxC_String
for(i=0;i<8;i++)
Receive_RF_Data[i]=0xC3;
RW_Txx_xxC_String(0x7E,&Receive_RF_Data,0x08);
// 向7EH單元寫入8個Byte的資料,其資料都為0xC3,存於位址為30H~37H
// MOV A ,#7EH
// MOV R0 ,#30H
// MOV R7 ,#07H
// LCALL W_Txx_xxC_String
Write_Command(0x33);
Write_Command(0x37);
for(i=0;i<30000;i++);
// *****
// 配置TWS-900C
// *****
void Txx_xxC_Tx_Mode(void)
http://www.wenshing.com.tw ; http://www.rf.net.tw

```

```
{  
Write_Command(0x36);  
Write_Command(0x3B);  
Write_Command(0x35);  
RW_Txx_xxC_String(0x7F,&Receive_RF_Data,0x0A);  
while(!Flag); //此腳應與CE腳對調，也就是6腳為Flag,7腳為CE腳。  
while(Flag);  
}
```

WENSHING Confidential  
Do Not Distribution