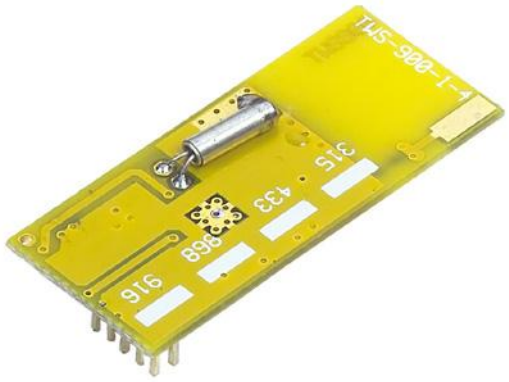
WENSHING®©

TWS-900C RF MODULE

Wireless Low Power RF Transmitter Module (GFSK)



WENSHING Confidential
Do Not Distribution

Version History

Version	Date	Changes
V1.01	May. 05,2009	1st. Edition
V1.02	Aug. 26, 2009	2nd. 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	℃	
Solder Reflow Temperature		265	℃	According to IPC/JEDEC J-STD-020C

WENSHING Confidential Do Not Distribution

Operating Condition

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

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

Electrical Specification

Tc = 25℃, 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

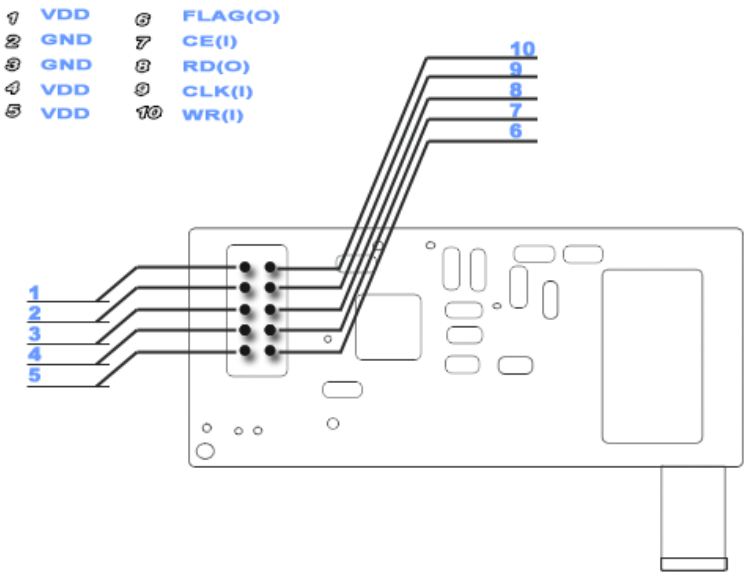
WENSHING Confidential Do Not Distribution

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).

WENSHING Confidential
Distributor

Pin Assignment



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)

Application

1 This instruction is writing for matching TRW-400B transceiver module. If you would like to use TRW-900C, we also have TRW-900C data sheet for reference.

2 Data Rate：2.4/4.8/9.6/19.2/38.4/76.8/153.6K

3 Current Consumption：

3.1 +10dBm output 26.4mA

3.2 0dBm output 15mA

4 Dispose：

4.1 After power program, restore TWS-900C.

4.2 Wait for 1ms, and then dispose TWS-900C.

4.3 After dispose, enable transmit data to TWS-900C at least 5ms time delay.

5 In order to simplify the program, the number of package can not beyond 64 Byte for once. (In principle, it can have illimitably Byte.)

6 Pin Illustration：

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 Frequency Formulation：

Frequency value = actual working frequency*2^16/26000000, frequency value address: 0x0D, 0x0E and 0x0F.

Ex：Working frequency = 434MHz，frequency value：

= 434*2^16/26 = 1093947 = 0x10B13B，then:

0x0D = 0x10

0x0E = 0xB1

0x0F = 0x3B

8 Restore TWS-900C：

8.1 Dispose high/low CE, more than 10us time delay.

8.2 Dispose high CE; have to equal or more than 50us T1 time delay.

8.3 Reset instruction: 0x30.

WENSHING Confidential Do Not Distribution

8.4 When CE become low (After T1 time), suggest RD status:

8.4.1 If it shows 1, keep waiting. (if beyond 20ms, indicate bad module.)

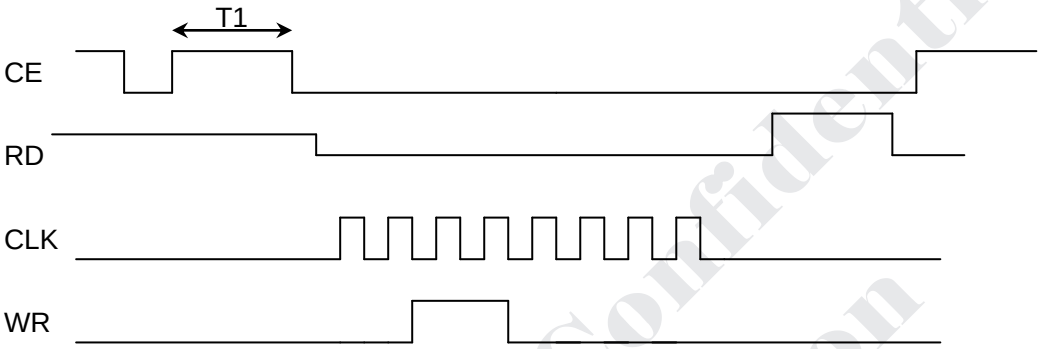
8.4.2 If it shows 0, then transmit reset instruction to RF.

8.5 Wait for RD status after finish reset instruction transmitting：

8.5.1 If it shows 1, make CE to 1 after wait for others turn to 0.

8.5.2 If it shows 0, make CE to 1.

8.6 After restoration, the 時序 are as below：



9 Write in orders to RF module：

9.1 Write in two orders after finish dispose：0x37/0x33

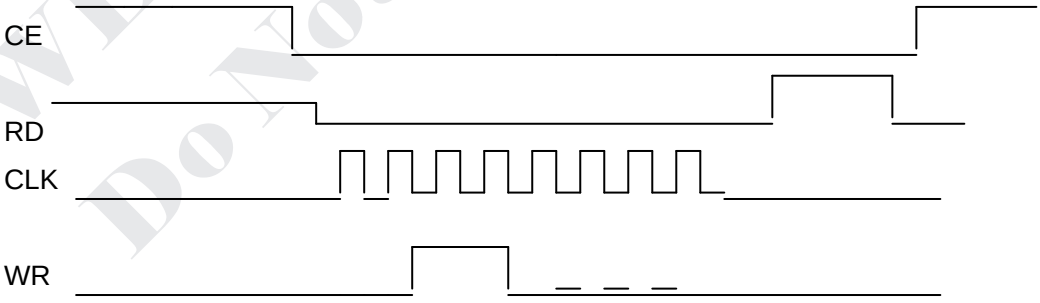
9.2 Write in two orders before transfer：0x3B/0x35

9.3 Write in two orders before enter low power consumption：0x39/0x32

9.4 From low power consumption to normal：Reset, dispose.

9.5 When CE become low, have to wait for RD become low to transfer data to RF.

Graphs of write in orders：



WENSHING Confidential
Do Not Distribution

10 Dispose data to RF, form: address + data.

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 Special description：

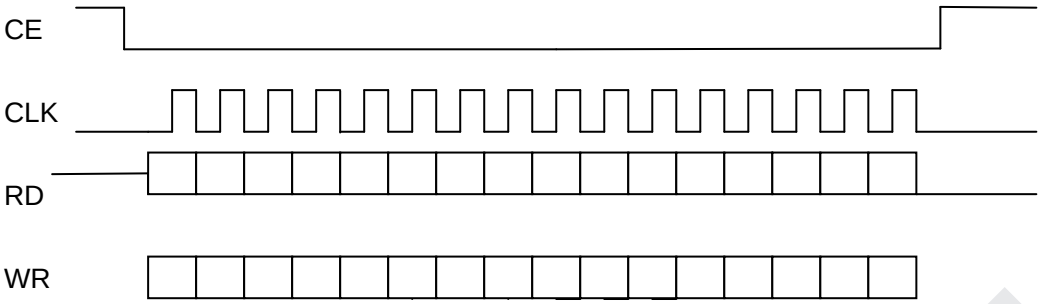
10.1.1 0DH，0EH，0FH are the address for working frequency.

10.1.2 06H is the number of BYTE for once package. It will be 0AH if it transfers 10 units once.

10.2 After reset and write the dispose above in RF, write in two orders：0x37/0x33

10.3 If would like to know the data in 06H is correct or not, change the address to 0x80 and read RD data, as same as reading others address. The graphs are as same as while write in, reference as below：

WENSHING Confidential
Do Not Distribution



10.4 In order to match TRW-400 transceiver module, transfer data have accord with the form as below, or TRW-400 will not receive any data:

10.4.1 Address + Byte*n data + checkup code.

10.4.2 Address (Byte*n) the transfer number have to be the same as RX value.

10.4.3 Check code (include two Byte, the value of these two Byte are the same), 55 ⊕ address ⊕ data. Ex: Transmit address is 4 Byte, its value: 0x12345678. Data is 8 Byte, its value: 0x01020304050607. Check code: 0x5D5D.

10.4.4 Transmit address is 0x7F+0x12345678+0x01020304050607+0x5D5D.

10.4.5 After CE becomes low, check RD line to see if it is 0 or not. Transfer address if it is 0, or keep waiting.

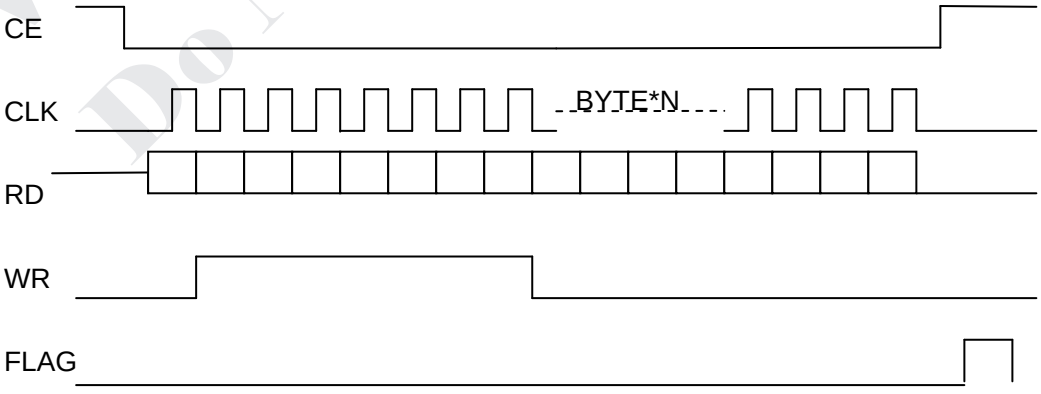
10.4.6 Write in two orders to RF module before transfer data every time. 0x3B/0x35

10.4.7 After all data transferred, check FLAG is 1 or not:

If 1, transfers the next package after it become low.

If 0, transfers the next package after it become low later than high.

Graph of transfer data are as below



WENSHING Confidential
Do Not Distribution

Demo Program

```
*********************************************************************
// RESETxx_xxxC module
// *********************************************************************
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;
while(RD);
CE = 1;
}
// *********************************************************************
// Enter to 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
```

WENSHING Confidential
Do Not Distribution

```
DJNZ R2 ,RESET_Txx_xxxC_0
SETB CE
RET
// ************************************************************************
// When Address.7 = 1，to read out 1 byte data from Txx_xxxC.
// When Address.7 = 0，to read in 1 byte data from Txx_xxxC.
// ************************************************************************
char RW_Txx_xxC(char Address,char data0)
{
Unsigned char i=0;
CE = 0;
While (RD);
SPI0DAT = Address;
While (!SPIF);
SPIF =0;
SPI0DAT = data0;
While (!SPIF);
SPIF =0;
CE = 1;
Return (SPI0DAT);
}
// ************************************************************************
// Key in A, B. The A for address, B for data.
// Function：to read in a disposition unit or data from Txx_xxC.
// ************************************************************************
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
```

WENSHING Confidential
Do Not Distribution

```
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
// ************************************************************************
// Key in A, and A for address.
// Function：to read out a disposition unit or data from Txx_xxC.。
// ************************************************************************
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;
```

WENSHING Confidential
Do Not Distribution

```
while(!SPIF);
SPIF =0;
CE = 1;
}
// ***********************************************************************
// Key in  ：ACC
// Function: to key in a command value from Txx_xxxC module.
// ***********************************************************************
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
// ***********************************************************************
// When Address.7 = 1，to read out 1 string data from Txx_xxxC.
// When Address.7 = 0，to read in 1 string data from Txx_xxxC.
// ***********************************************************************
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;
}
// ***********************************************************************
```

WENSHING Confidential
Do Not Distribution

```
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
// ************************************************************************
// Key in：A，R0，R7
// A is to represent the beginning address
// R0 is to represent the expressing data by beginning address
// R7 is to represent the how many byte will be expressed?
// ************************************************************************
// ************************************************************************
W_Txx_xxxC_String:
CLR CE
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
// ************************************************************************
// Key in：A，R0，R7
```

WENSHING Confidential
Do Not Distribution

```
// A is to represent the beginning address
// R0 is to represent the read out the data of RF preservation site unit
// R7 is to represent the how many byte will be read?
// ********************************************************************
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
// ********************************************************************
// allocation 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
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
```

WENSHING Confidential
Do Not Distribution

```
RW_Txx_xxC(0x06 ,0x0A); //06 = Adress+RF BYTE+2 CRC 4+4+2
RW_Txx_xxC_String(0xC0,&Receive_RF_Data,0x30);
// from 00H unit to read out 0x30 data, the modifies are as follows:
// MOV A ,#0C0H
// MOV R0 ,#30H
// MOV R7 ,#30H
// LCALL R_Txx_xxxC_String
for(i=0;i<8;i++)
Receive_RF_Data[i]=0xC3;
RW_Txx_xxC_String(0x7E,&Receive_RF_Data,0x08);
// Read in 8 byte data to 7EH unit，the data all for 0xC3, the memory address is for 30H~37H
// MOV A ,#7EH
// MOV R0 ,#30H
// MOV R7 ,#07H
// LCALL W_Txx_xxxC_String
Write_Command(0x33);
Write_Command(0x37);
for(i=0;i<30000;i++);
}
// ***************************************************************************
//Allocation: TWS-900C
// ***************************************************************************
void Txx_xxxC_Tx_Mode(void)
{
Write_Command(0x36);
Write_Command(0x3B);
Write_Command(0x35);
RW_Txx_xxC_String(0x7F,&Receive_RF_Data,0x0A);
While (Flag); //This pin should be changed with CE pin, it mean that the sixth pin is Flag, the
seventh pin is CE pin.
While (Flag);}
```

WENSHING Confidential
Do Not Distribution