
Wireless Hi Power 0.5W RF Transceiver Module for
Narrowband System

**Version History**

Version	Date	Changes
V1.01	Jan.20,2010	1 st . Edition
V1.02	Mar.3,2008	2 nd . Edition

Specification

● UHF Wireless Data Transceiver	● 431MHz ~435MHz ISM
● 402 / 424 / 426 / 429 / 433 / 447 / 449 / 469 MHz Operation	
● Single 4V to 6V Supply	● Up to 0.5W Output Power
● Hi Sensitivity: -122dBm	● AFC Function
● Antenna On Board	● Digital RSSI and Carrier Sense Indicator
● Application Range : Remote Metering 、 Wireless Security Systems 、 Automatic Meter 、 Reading 、 Home Automation	

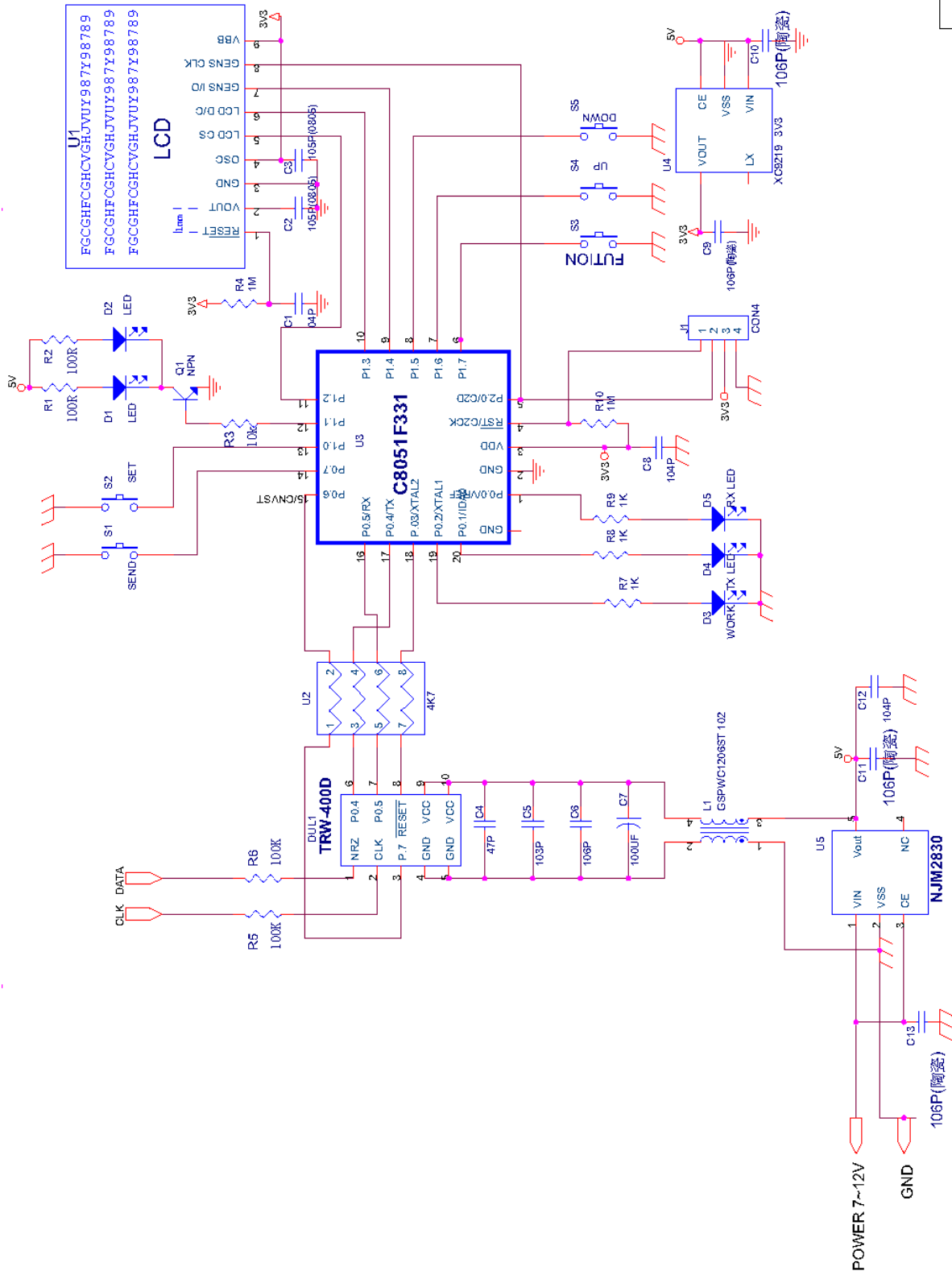
RF Transmit Section

Parameter	Specification			Unit	Condition
	Min	Type	Max		
Frequency Range	431	433.92	435	MHz	
RF Channels		320			12.5KHz Channel
Transmit Data Rate	2.4		153.6	Kbps	2.4K/4.8K/9.6K/19.2K/ 38.4K/76.8K/153.6K
Output Power		+27		dBm	
Current Consumption		250		mA	

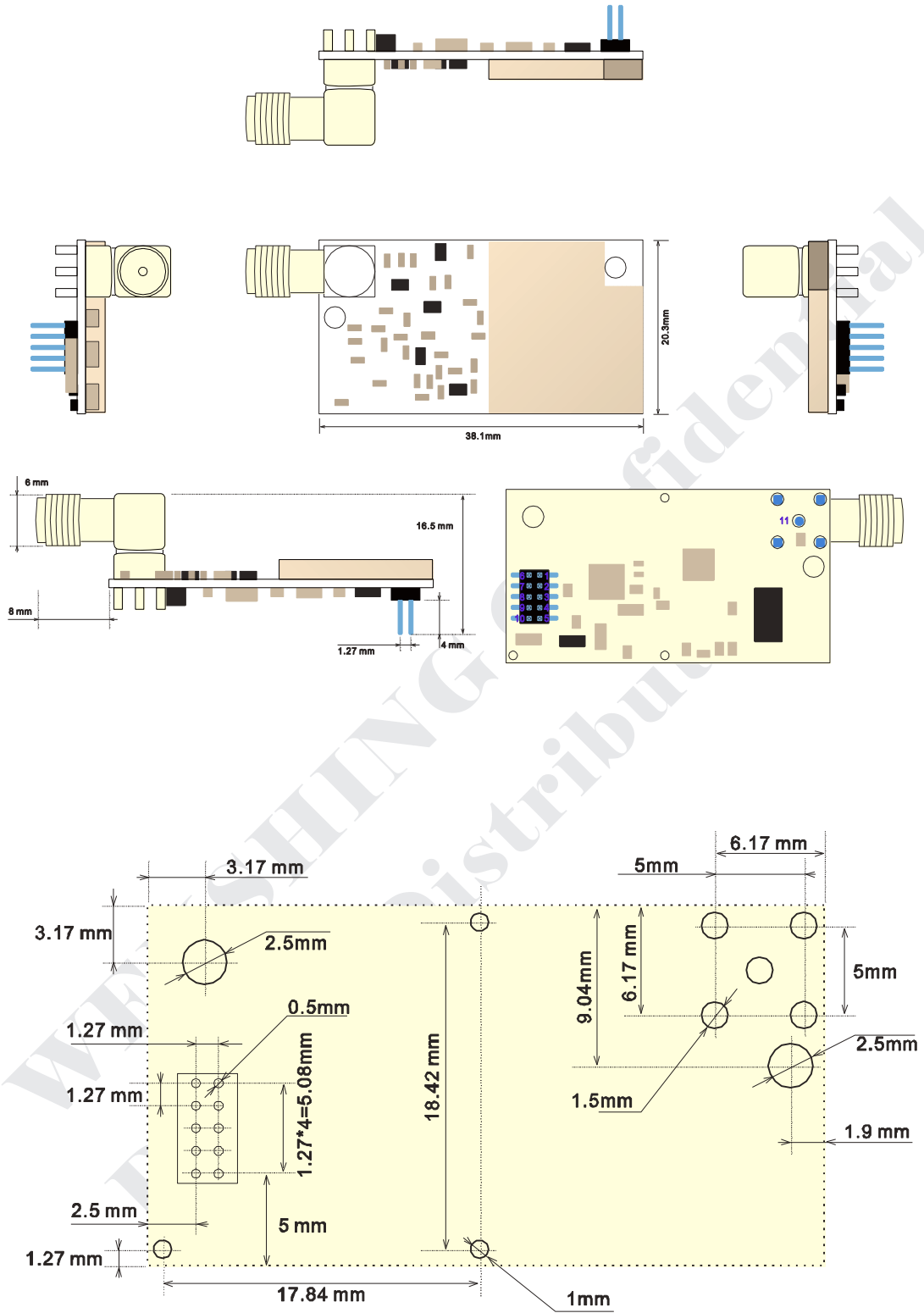
RF Receive Section

Parameter	Specification			Unit	Condition
	Min	Type	Max		
Frequency Range	431	433.92	435	MHz	
RF Channels		320			12.5KHz Channel
Transmit Data Rate	2.4		153.6	Kbps	2.4K/4.8K/9.6K/19.2K/ 38.4K/76.8K/153.6K
Sensitivity		-122		dBm	12.5KHz Channel
		-117		dBm	25KHz Channel
		-98		dBm	500KHz Channel Spacing: 153.6 K Band
Current Consumption		23		mA	
Operating Ambient Temperature Range	-10		+70	°C	

Demo Circuit



Size

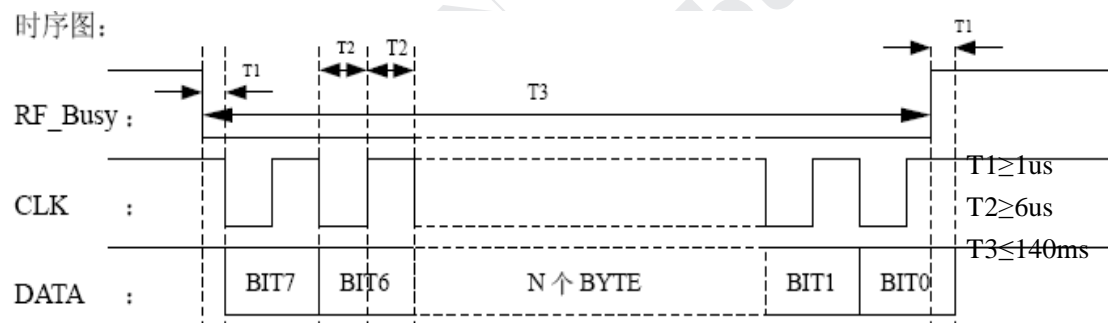


Reference hole position for PCB mounting(Bottom view)

Pin Assignment

Pin	Function	Description
1	NRZ DATA	RF DATA
2	NRZ CLK	RF CLK (work rate)
3	RF BUSY	INPUT/OUTPUT
4	GND	POWER GND
5	GND	POWER GND
6	DATA	INPUT/OUTPUT
7	CLK	INPUT
8	RESET	RF RESET (LO RESET)
9	VCC	POWER VDD
10	VCC	POWER VDD
11	SMA	RF ANTENNA

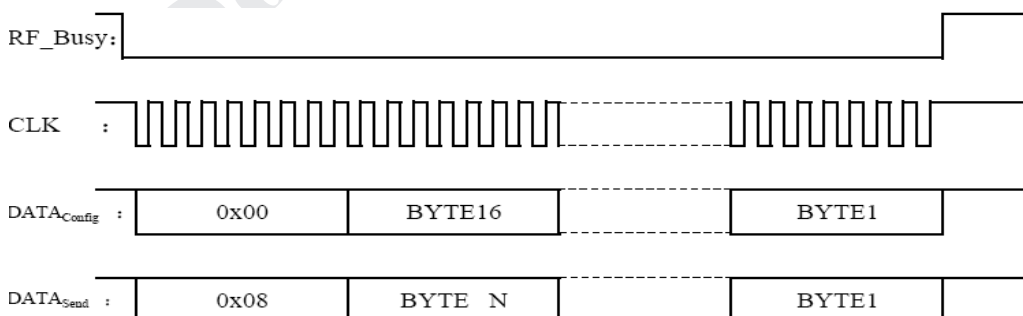
Minimum Time Request



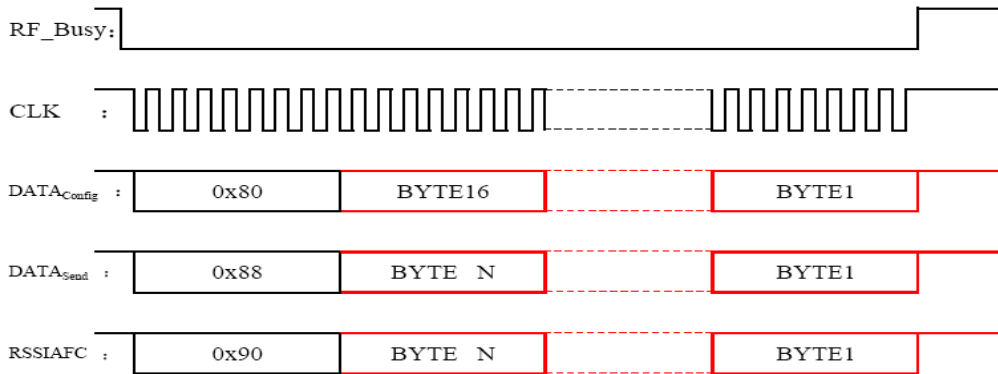
Note:

Sequence Diagram(writing) for configuration and data send:

All black come from user, all RED come from RF.



Sequence Diagram (reading) for configuration and data send



Note: "N" is decided by Configuration Bit6~0 in BYTE16.

Command for write configuration data: 0x00H, command for read configuration data : 0x80H.

Command for data send: 0x08H, command for read data set: 0x88H

Command for red RSSI and AFC:0x90H.

Remark:

1. When it configure, if below CLK number are over $136 = (16+1) * 8$, only need to accept data before 136 CLK and others will automatically mask and wait RF_BUSY become high level.

Send data which wrote are also the same, but LCK number is decided by BIT6~0 of first BYTE of configuration, it's number = $(\text{Bit6} \sim 0 + 1) * 8$.

2. When RF working under transmission, it will detect DATA PIN is in low. You can write and read data in RF at this time, otherwise, RF will be wrong. After send data for one time, it will determine DATA line status every 10us.

3. When module received 0x90 command, please read 2 bytes.

First BYTE is RSSI value, second BYTE is AFC value, and those two are with sign.

If RSSI value transfer to dBm is not sure, we only can say if read value is larger, sense indicator (it has sign differentiate) is stronger.

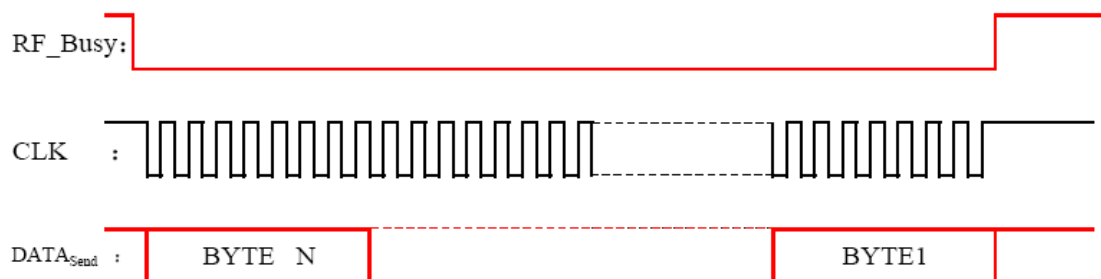
AFC value transfer to frequency is as below:

$$F_{AFC} = AFC * \text{Work rate} / 16$$

Remark:

Receiving Data Sequence Diagram :

CLK are all from user, RF_Busy/DATA come from RF automatically.



Remark:

1. Read receiving data is not input from command directly. When CLK which input reach N*8, RF_Busy will automatically become high.
2. When RF_Busy is low, you can give configuration data to RF. User also need to input LCK to RF in 1ms, otherwise, this data will be loss.
3. First BYTE address(first BYTE) value doesn't during 40H~77H or 80H~FFH.

When using TRW-400D Transceiver Module, please use main transmit mode which is similar to SPI protocol to send 16 bytes to TRW-400D for configuration.

Configuration address is 0x00, please check more details of 16 bytes as below:

BYTE16	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Select mode and package length.
	CM		Length of data						
BYTE15	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	ID setting ,work Rate and Tx/Rx Select
	RXEN	Address BYTE				Data Rate			
BYTE14	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Receive Frequency setting.
	FREQ_2R								
BYTE13	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
	FREQ_1R								Transmit Frequency setting.
BYTE12	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
	FREQ_0R								
BYTE11	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Transmit Frequency setting.
	FREQ_2T								
BYTE10	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
	FREQ_1T								Preamble setting, fixed length on 4 bytes .
BYTE9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
	FREQ_0T								
BYTE8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Preamble setting, fixed length on 4 bytes .
	Preamble BYTE								
BYTE7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Setting ID code.
	Address BYTE 6								
BYTE6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
	Address BYTE 5								
BYTE5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
	Address BYTE 4								
BYTE4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
	Address BYTE 3								
BYTE3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
	Address BYTE 2								
BYTE2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
	Address BYTE 1								
BYTE1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
	Address BYTE 0								

BYTE Description

1.1 BYTE16 :

Bit7(CM) mode selection:

1----- Direct mode. Send and receive data from RF CLK and RF DATA line, not through MCU.

This mode can save SPI sending time and increase sending rate.

0----- Indirect mode. Send data and receive data through MCU.

Bit6~0 package length should be between 1~100 bytes.

1.2 BYTE15 :

Bit7(RXEN) receive and transmit selection position:

1----- RF Working is on receiving mode.

0----- RF Working is on transmitting mode.

Bit6~4(Adress_Byte) ID code setting:

At most is 7 bytes; if it is less than 7 bytes, please select low.

If ID number set on 4, ID code save on: BYTE4 , BYTE1, high is in front.

Attention:

ID data and ID code of pairing receiver and transmitter module must be the same.

Bit3 RF module low power consumption:

0----- RF chip works on low power consumption.

1----- RF chip works on normal.

For Bit2~0 RF working rate, there are several for your selection as below:

2.4K/4.8K/9.6K/19.2K/38.4K/76.8K/153.6Kbps, 6 types for your selection.

Ex: [Bit2~0]=4 when choose 38.4Kbps.

1.3 BYTE14~12 is the value of receiving frequency:

BYTE14 is the highest byte of receiving frequency, BYTE12 is the lowest byte. Its formula is as below:

$$\text{FREQ} = \{[(f_{SF}-307.2K) - (3/4)*7372800]*32768 - 3686400\} / 3686400$$

f_{SF} is the actual working frequency. FREQ is the value of write in BYTE 14~12 (have to use hexadecimal method), whatever the lowest is 0 or 1, all replace by 1.

Ex: while configuration is in 433.92MHz

$$\begin{aligned}\text{FREQ} &= (433920000-307200-0.75*7372800*32768-3686400) / 3686400 \\ &= 3805183 = 0x3A0FFF\end{aligned}$$

1.4 BYTE11~9 is the value of transmitting frequency:

BYTE11 is the highest byte of transmitting frequency, the lowest byte is BYTE9. Its formula is as below:

$$\begin{aligned}\text{FREQ} &= [(433920000-0.75*7372800) * 32768 - 3686400] / 3686400 \\ &= 3807913 = 0x3A1AA9\end{aligned}$$

f_{SF} is the actual working frequency. FREQ is the value write in BYTE 11~9 (have to use hexadecimal method), whatever lowest is 0 or 1, all replace by 1.

Ex: while configuration is 433.92MHz

$$\text{FREQ} = [(433920000 - 0.75 * 7372800) * 32768 - 3686400] / 3686400 \\ = 3807913 = 0x3A1AA9$$

1.5 BYTE8 is for the number of preamble code, the value is between 4~255.

1.6 BYTE7~1 are the ID numbers for distinguish from transmitter and receiver. The value has to be the same for a relative pair of transmitter and receiver.

When configuration RF, data writing to RF are as follows:

C0+BYTE16+BYTE15+...+BYTE2+BYTE1

NRZ DATA :

When CM=1, it is the data line of RF.

When CM=0, do not need this line.

NRZ CLK :

The data rate of RF: 2.4, 4.8, 9.6, 19.2, 38.4, 76.8, 153.6Kbps.

RF_Busy :

Transmitting status:

= 0, modules is under busy status, control chip can't read and write module, otherwise, it will has error.

= 1, it can read/write data and configuration data.

Receiving mode:

= 0, indicate module already received data and wait for MCU read data.

When this line keeps 1ms and MCU doesn't read number, this data will get lost.

= 1 indicate RF module waiting for data receiving and can write configuration data into module.

RESET: Module reset pin, low level reset.

Data: RF module control signal and signal line.

CLK: RF control signal and signal line. When it is free, this line is high level.

Remark:

The time for T1 is over 1us, time for T2 is over 6us , time for T3 can't over 140ms(data will not deal if connection is over time).

CLK is low, it write into data, CLK is high, it read data.

1. Under (read/write) transmitting data and (read/write) configuration status:

Clock of RF_BUSY, CLK and DATA will produce from user.

a) Write configuration command: 0xxx0xxxB

Format: 00H+16 bytes configuration information.

b) Write transmitting data command: 0xxx1xxxB

Format: 08H+N bytes of data.

- c) Read configuration command: 1xx00xxxB
Format: 80H+16 bytes configuration data.
- d) Read transmitted data command: 1xx01xxxB
Format: 88H+N bytes data.
Remark:
When it is on configuration, below CLK number should be over 136 CLK data, behind data will be mask automatically and wait RF_BUSY become high level.
When it write into transmitted data, it is also the same, but CLK numbers will decided by BIT6~0 of first byte of configuration, number= (Bit6~0+1) *8 ◦
- e) When MCU detect DATA line is low, it can give configuration and send data to RF.

2. Under receive status:

RF_BUSY, data produce from RF , CLK produce from user.

- a) Read configuration command: 1xx00xxxB
Format: 80H+16 bytes configuration information.
- b) Write configuration command: 0xxx0xxxB
Format: 00H+16 bytes configuration information.
- c) Read RSSI & AFC command: 1xx1xxxxB
Format: 90H+AFC+RSSI
Before received data, value of read RSSI & AFC are 0.
In direct mode, it can read AFC & RSSI value.
- d) Read received data is not through command input. When you input CLK into internal to identify data, RF_BUSY will automatically become high.
- e) When RF_BUSY is low, you can give configuration data to RF.

3. The functions are all the same as below, only add save electricity function:

- a) When BIT3~BIT0 of BYTE2 is equal to 0111b, meanwhile, RF entry into stop status.
Current is less than 1mA.
Wake up method:
Give RESET line of RF a low pulse for 20ms. After it normal working, it will automatically working and under the status of writing Bit7 of BYTE2, but rate will become 4.8K.
Note:
When BIT2~0 is 111, there is no any relative rate to select.
When it is other value, relative rate is as below: 4K/4.8K/9.6K/19.2K/38.4K/76.8K/153.6K
- b) When BIT3 of BYTE2 is 0, Bit2~0 is not 111 and RF WILL entry into standby status.
Give configuration data, it can work on transmitted status or received status.
Address of first BYTE should be not between 40H~77H or 80H~EFH.

4. Address of first BYTE should be now between 40H~77H or 80H~EFH.

; -----
;
; -----

Read_Config EQU 80H ;
Read_Send EQU 88H ;

```

;-----
;-----
ORG      0000H      ;
NOP      ;
NOP      ;
SJMP    RESET      ;
;-----
RESET:
    LCALL Delay200ms ;
;-----
    SETB  RF_Busy   ;
    SETB  CLK       ;
    SETB  DATA     ;
;-----
; Transmitted status
;-----
Config_TX_State:
    MOV    DPTR    ,#Config_Tx_Table ;
    MOV    R2      ,#17              ;
    LCALL  Config_Send_Data          ;
    MOV    R2      ,#16              ;
    MOV    A       ,#Read_Config     ;
    ; This is not necessary, just in order
    ; to verify wrote data is correct or not.
    LCALL  Read_Config_Send         ;
;-----
Send_DATA_Loop:
    LCALL  Delay20us                ; Only need to extend 20us.
    JB    DATA    ,$               ;
    MOV    DPTR    ,#Send_Data      ;
    MOV    R2      ,#5              ;
    LCALL  Config_Send_Data         ;
    LCALL  Delay20us                ; This is not necessary
    MOV    A       ,#Read_Send      ;
    MOV    R2      ,#4              ;
    LCALL  Read_Config_Send         ;
    AJMP  Send_DATA_Loop           ;
;-----
; *****
; *****
; Received status
;-----
    MOV    DPTR    ,#Config_Rx_Table ;
    MOV    R2      ,#17              ;
    LCALL  Config_Send_Data         ;
;-----
Receive_Data_Loop:

```

```

JB      RF_BUSY,$          ;
MOV     R2      ,#4        ;
LCALL  Read_DATA          ;
AJMP   Receive_Data_Loop  ;
;-----
; Standby status
;-----
MOV     DPTR   ,#Config_Idle_Table ;
MOV     R2     ,#17        ;
LCALL  Config_Send_Data      ;
AJMP   Config_TX_State      ;
AJMP   Config_RX_State     ; OR
;-----
; Stop status
;-----
MOV     DPTR   ,#Config_Stop_Table ;
MOV     R2     ,#17        ;
LCALL  Config_Send_Data      ;
CLR    RESET          ;
LCALL  Delay20ms        ;
SETB   RESET          ;
AJMP   $              ;
;-----
;-----
;-----
Read_Config_Send:
JNB    DATA    , $      ;
CLR    RF_Busy          ; Transfer to output status.
Read_Config_Send_000:
MOV    R3      ,#8      ;
Read_Config_Send_001:
CLR    CLK          ;
CLR    DATA        ;
JNB    ACC.7 ,Read_Config_Send_002 ;
SETB   DATA        ;
Read_Config_Send_002:
CLR    CLK          ;
LCALL  Delay10Cycle ;
LCALL  Delay10Cycle ;
LCALL  Delay10Cycle ;
SETB   CLK          ;
RL     A            ;
LCALL  Delay10Cycle ;
LCALL  Delay10Cycle ;

```

```

    LCALL Delay10Cycle ;
    DJNZ R3 ,Read_Config_Send_001;
    SETB DATA ; Transfer to input status.
    MOV R3 ,#8 ;
Read_Config_Send_100:
    CLR CLK ;
    LCALL Delay10Cycle ;
    LCALL Delay10Cycle ;
    LCALL Delay10Cycle ;
    SETB CLK ;
    CLR ACC.7 ;
    JNB DATA ,Read_Config_Send_101;
    SETB ACC.7 ;
Read_Config_Send_101:
    LCALL Delay10Cycle ;
    LCALL Delay10Cycle ;
    LCALL Delay10Cycle ;
    RL A ;
    DJNZ R3 ,Read_Config_Send_100;
    MOV R3 ,#8 ;
    MOV @R0 ,A ;
    INC R0 ;
    DJNZ R2 ,Read_Config_Send_100;
    SETB DATA ; Transfer to input status.
    SETB RF_BUSY ; Transfer to input status.
    RET ;
;-----
;-----
Read_DATA:
    MOV R3 ,#8 ;
Read_DATA_Loop:
    CLR CLK ;
    LCALL Delay10Cycle ;
    LCALL Delay10Cycle ;
    LCALL Delay10Cycle ;
    SETB CLK ;
    CLR ACC.7 ;
    JNB DATA ,Read_DATA_Loop_000 ;
    SETB ACC.7 ;
Read_DATA_Loop_000:
    LCALL Delay10Cycle ;
    LCALL Delay10Cycle ;
    LCALL Delay10Cycle ;
    RL A ;

```

```

DJNZ R3 ,Read_DATA_Loop ;
MOV @R0 ,A ;
INC R0 ;
DJNZ R2 ,Read_DATA ;
RET ;

```

; This sub-program can use on configuration TX/RX and send data.

Config_Send_Data:

```

JNB DATA , $ ;
CLR RF_Busy ; Transfer to output status.

```

Config_Send_Data_000:

```

CLR A ;
MOVC A ,@A+DPTR ;
INC DPTR ;
MOV R3 ,#8 ;

```

Config_Send_Data_001:

```

CLR CLK ;
CLR DATA ;
JNB ACC.7 ,Config_Send_Data_002;
SETB DATA ;

```

Config_Send_Data_002:

```

CLR CLK ;
LCALL Delay10Cycle ;
LCALL Delay10Cycle ;
LCALL Delay10Cycle ;
SETB CLK ;
RL A ;
LCALL Delay10Cycle ;
LCALL Delay10Cycle ;
LCALL Delay10Cycle ;
DJNZ R3 ,Config_Send_Data_001;
DJNZ R2 ,Config_Send_Data_000;
SETB RF_BUSY ; Transfer to input status
SETB DATA ; Transfer to input status
RET ;

```

; Configuration result:

; Rate: 4.8K

; Frequency: 433.9

; Send BYTE number: 4

; Received BYTE number: 4

; Preamble code: fixed on 4.

; Address code: 4

```
;-----  
Config_Tx_Table:  
    DB 000H,004H,039H,03AH,00FH,04FH,03AH,019H,0F9H ;  
    DB      004H,034H,056H,078H,090H,012H,034H,03FH ;  
;-----  
Config_Rx_Table:  
    DB 000H,004H,0B9H,03AH,00FH,04FH,03AH,019H,0F9H ;  
    DB      004H,034H,056H,078H,090H,012H,034H,03FH ;  
;-----  
Config_Idle_Table:  
    DB 000H,004H,031H,03AH,00FH,04FH,03AH,019H,0F9H ;  
    DB      004H,034H,056H,078H,090H,012H,034H,03FH ;  
;-----  
Config_Stop_Table:  
    DB 000H,004H,037H,03AH,00FH,04FH,03AH,019H,0F9H ;  
    DB      004H,034H,056H,078H,090H,012H,034H,03FH ;  
;-----  
Send_Data:  
    DB 008H,012H,034H,056H,067H  
;-----  
; end
```