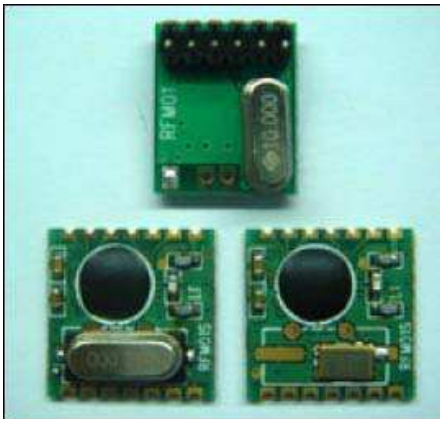
ISM BAND FSK RECEIVER MODULE

RFM01

（the purpose of this spec covers mainly for the physical characteristic of the module, for register configure and its related command info please refer to RF01 data sheets）

RFM01



General Introduction

RFM01 is a low costing ISM band receiver module implemented with unique PLL and zero IF design approach. It works with FSK modulated signal ranges from 315/433/868/915MHZ bands, comply with FCC, ETSI regulation. The SPI interface is used to communicate with microcontroller for parameter setting. RFM01 works with RFM02 transmitter module. At 433MHZ band, the pair of module can work up to 300m in the free open air.

Features:

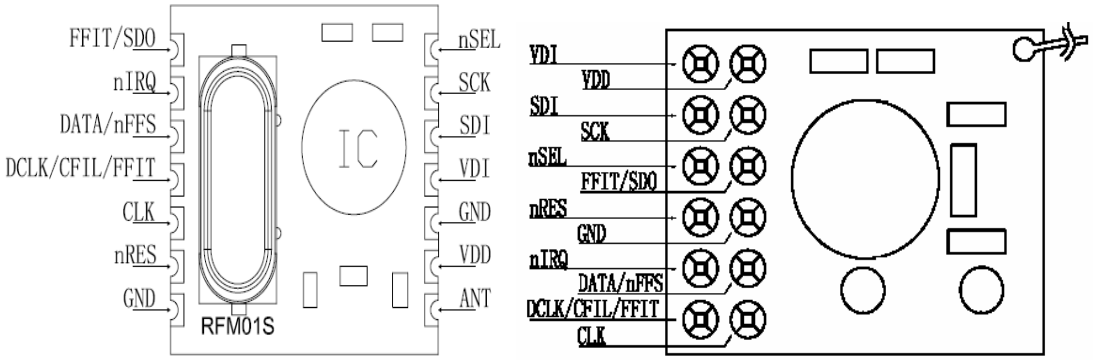
- Low costing, high performance and price ratio
- Tuning free during production
- FSK reception
- PLL and zero IF technology
- Fast PLL lock time
- High resolution PLL with 2.5 KHz step
- High data rate (up to 115.2 kbps with internal demodulator, with external RC filter highest data rate is 256 kbps)
- Differential antenna input
- Automatic antenna tuning
- Programmable receiver bandwidth (from 67 to 400 kHz)
- Analog and digital signal strength indicator (ARSSI/DRSSI)
- AFC
- DQD
- Internal demodulator
- SPI interface
- Clock and reset signal output for external MCU use
- 16 bits FIFO
- Low power mode (<0.5mA averaged current consumption)
- 10MHz crystal for PLL timing
- Wakeup timer
- Low battery detection
- Programmable capacitor bank
- 2.2V - 5.4V power supply
- Low power consumption
- Stand by current less than 0.3uA

Typical Application：

- Remote control
- Remote sensor
- Wireless data collection
- Home security system
- Toys
- Tire pressure monitoring system

Pin Definition：

SMD **DIP**



definition	Type	Function
VDI	DO	Valid data indicator
VDD	S	Positive power supply
SDI	DI	SPI data input
SCK	DI	SPI clock input
nSEL	DI	Chip select (active low)
FFIT/SDO	DO	FIFO fill interrupt(active low) or status read data output
nRES	DO	Reset output（active low）
GND	S	Power ground
nIRQ	DO	Interrupts request output（active low）
DATA/nFFS	DO/DI	Data input(non FIFO mode)/ FIFO select
DCLK/CFIL/FFIT	DO/AIO/DO	Clock output (no FIFO)/ external filter capacitor(analog mode)/ FIFO interrupts(active high)when FIFO level set to 1, FIFO empty interruption can be achieved
CLK	DO	Clock output for external microcontroller

Electrical Parameter：

Maximum（not at working mode）

symbol	parameter	minimum	maximum	Unit
V_{dd}	Positive power supply	-0.5	6.0	V
V_{in}	All pin input level	-0.5	Vdd+0.5	V
I_{in}	Input current except power	-25	25	MA
ESD	Human body model		1000	V
T_{st}	Storage temperature	-55	125	℃
T_{ld}	Soldering temperature(10s)		260	℃

Recommended working range

symbol	parameter	minimum	maximum	Unit
V_{dd}	Positive power supply	2.2	5.4	V
T_{op}	Working temperature	-40	85	℃

DC characteristic

symbol	parameter	Remark	minimum	typical	maximum	Unit
I_{dd}	Current consumption	315,433MHz band 868,915MHz band		9 10.5	11 12.5	mA
I_x	Stand by current	Crystal and base band on		3．0	3．5	mA
I_{pd}	Sleep mode current	All blocks off		0.3		uA
I_{lb}	Low battery detection			0.5		uA
V_{lb}	Low battery step	0.1V per step	2.2		5.3	V
V_{lba}	Low battery detection accuracy			75		mV
V_{il}	Low level input				0.3*V_{dd}	V
V_{ih}	High level input		0.7*V_{dd}			V
I_{il}	Leakage current	V_{il}=0V	-1		1	uA
I_{ih}	Leakage current	V_{ih}=V_{dd},V_{dd}=5.4V	-1		1	uA
V_{ol}	Low level output	I_{ol}=2mA			0.4	V
V_{oh}	High level output	I_{oh}=-2mA	V_{dd}-0.4			V

AC characteristic

symbol	parameter	remark	min	typical	max	Unit
f_{ref}	PLL frequency	Parallel fundamental	8	10	12	MHz
f_{LO}	frequency (10MHz crystal used)	315 MHz band,2.5KHz step 433 MHz band,2.5KHz step 868 MHz band,5KHz step 915 MHz band,7.5KHz step	310.24 430.24 860.48 900.72		319.75 439.75 879.51 929.27	MHz

f_{LO}	frequency (8MHZ crystal used)	315 MHz band,2.5KHz step 433 MHz band,2.5KHz step 868 MHz band,5KHz step 915 MHz band,7.5KHz step	248.19 344.19 688.38 720.57		255.80 351.80 703.61 743.41	MHz
f_{LO}	frequency (12MHZ crystal used)	315 MHz band,2.5KHz step 433 MHz band,2.5KHz step 868 MHz band,5KHz step 915 MHz band,7.5KHz step	372.28 516.28 1032.5 1080.8		383.71 527.71 1055.4 1115.1	MHz
BW	Receiver bandwidth	1 2 3 4 5 6	60 120 180 240 300 360	67 134 200 270 350 400	75 150 225 300 375 450	kHz
t_{lock}	PLL lock time	After 10MHz step hopping, frequency error <10 kHz		20		us
$T_{st, p}$	PLL start time	After crystal stabilized		250		us
BR	Data rate	With internal digital demodulator			115.2	kbps
BRA	Data rate	With external RC filter			256	kbps
P_{min}	sensitivity	BW=134KHz,BR=1.2kbps		-109	-100	dBm
AFC_{range}	AFC working range	δF_{fsk}: received signal modulation depth		0.8* δF_{fsk}		
RS_A	RSSI accuracy			±5		dB
RS_R	RSSI range			46		dB
C_{ARSSI}	ARSSI filter			1		nF
RS_{STEP}	RSSI programmable step			6		dB
RS_{RESP}	DRSSI response time	RSSI output high after valid , C_{ARRSI}=5nF		500		us
C_{XL}	Capacitor bank	Programmable step with 0.5pF step, +/- 10%	8.5		16	pF
T_{POR}	PWR time	V_{dd} reach 90%		50	100	mS
T_{PBT}	Wakeup timer period	Calibrated each 30s	0．96		1．08	mS
$T_{WAKE-UP}$	Programmable wakeup time		1		5*10E11	mS
T_{SX}	Crystal start up time	Crystal ESR < 100 Ohms			5	mS
$C_{IN,D}$	Load capacitance				2	pF
$T_{r,F}$	Output rising edge	With 15PF load			10	ns

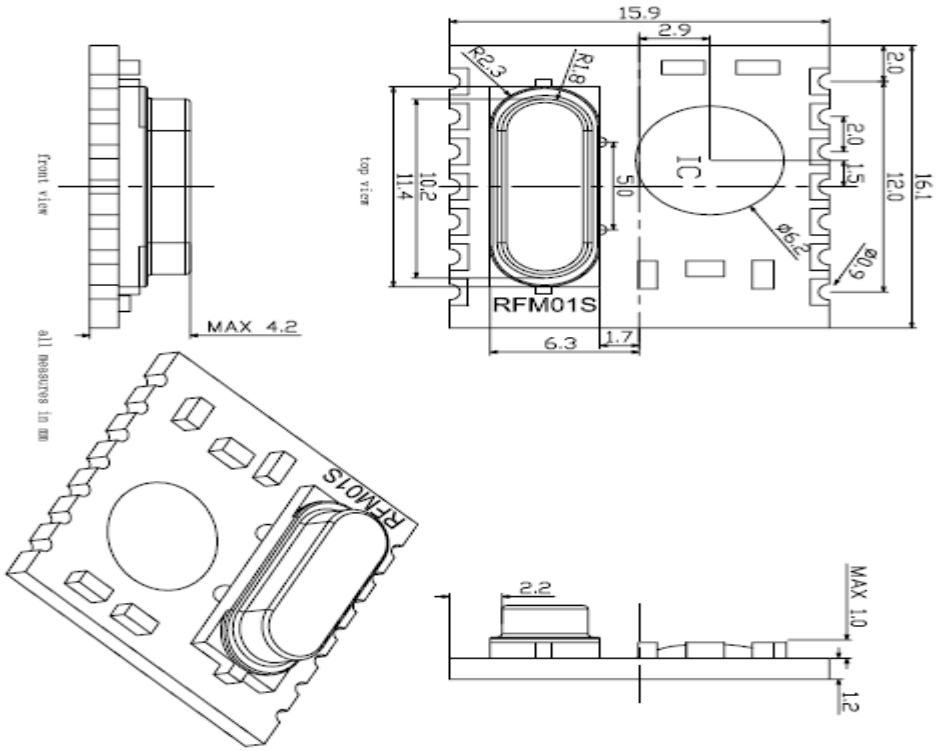
Field testing range

Band	Test condition	Distance
433MHz band	Receiver bandwidth =134KHz, data rate=1.2kbps, transmitter modulation=60KHZ（matches with RF02B） In free open area	>300M
868MHz band	Receiver bandwidth=134KHz,data rate =1.2kbps Transmitter modulation=60KHZ（matches with RFM02B）in free open area	>200M
915MHz band	Receiver bandwidth=134KHz,data rate =1.2kbps Transmitter modulation=60KHZ（matches with RFM02B）in free open area	>200M

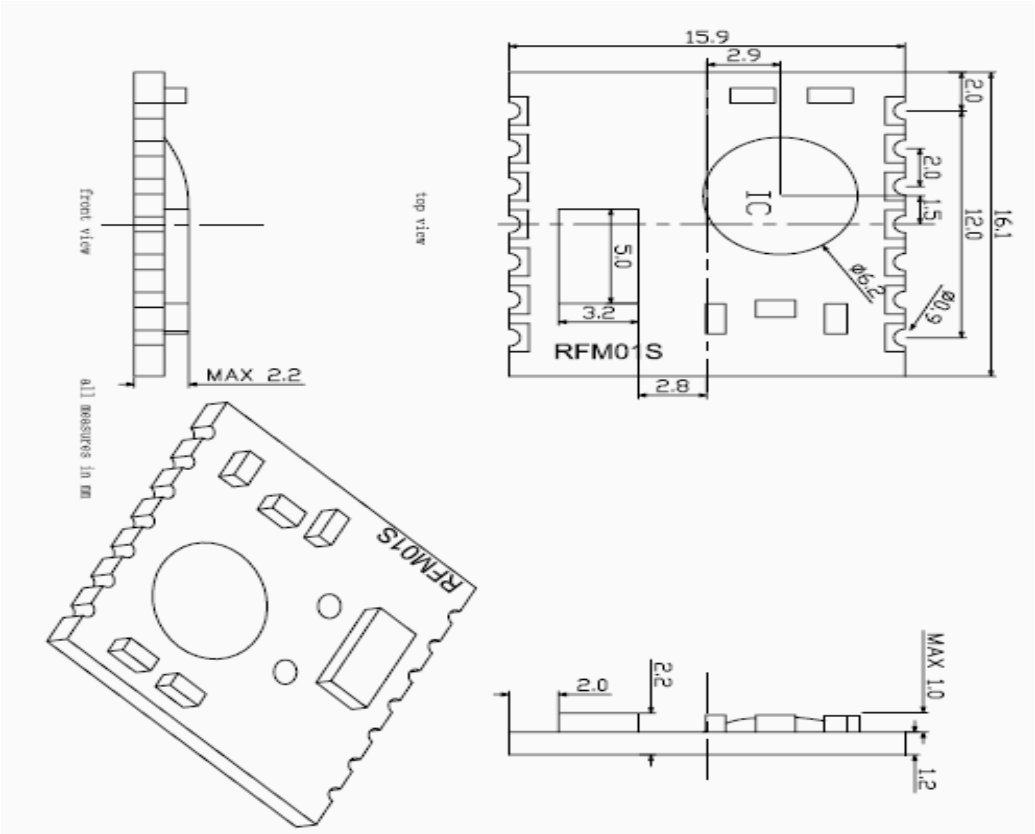
Mechanical Dimension

（units in mm）

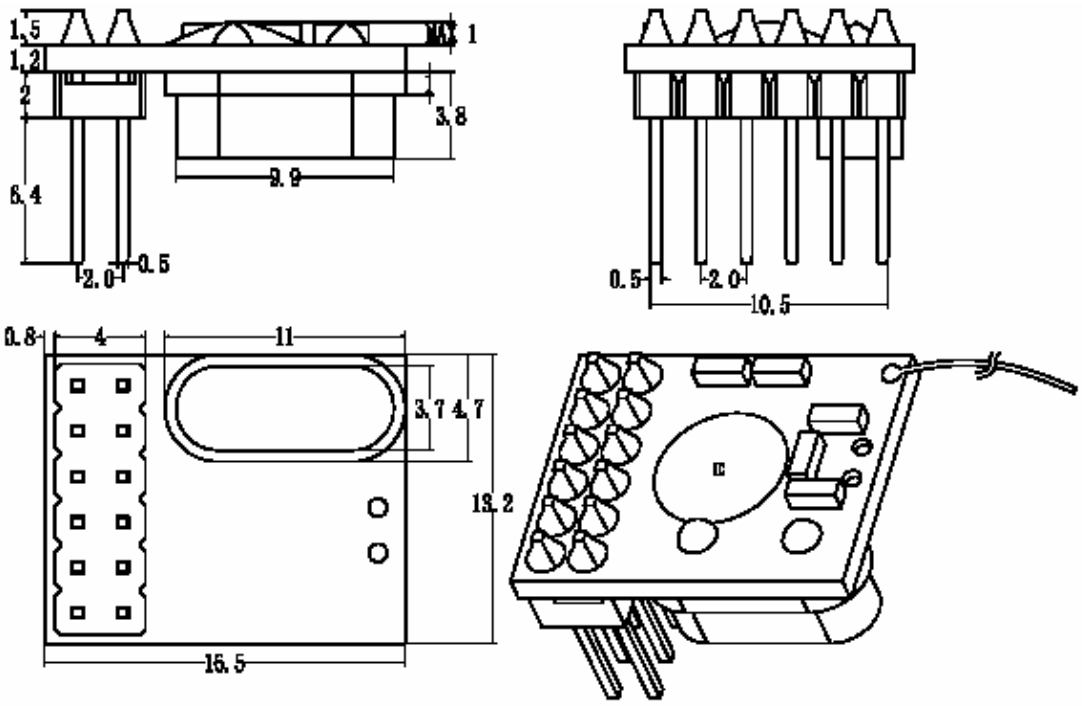
SMD PACKAGE（S1）



SMD PACKAGE（S2）

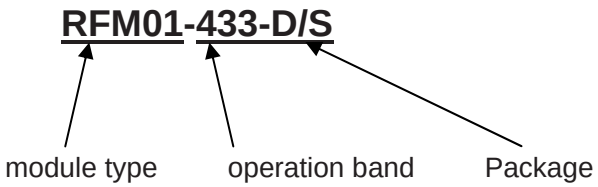


DIP PACKAGE（D）



Module Model Definition

model=module-operation band



Note: SOP packages is divided into two kinds based on thickness: 1. thickness is 4.2mm, 2. thickness is 2.2mm

example：1，RFM01 module at 433MHz band ,DIP: RFM01-433-D。

2，RFM01 module at 868MHZ band,SMD, thickness at 4.2mm: RFM01-868-S1。

Marking difference:

（color marks the difference for frequency）

RFM01 receiver band	Color
433MHz band	Black
868MHz band	red
915MHz band	green

HOPE MICROELECTRONICS CO.,LTD
Rm B.8/F LiJingGe Emperor Regency 6012 ShenNan Rd., Shenzhen,China
Tel: 86-755-82973805
Fax: 86-755-82973550
Email: sales@hoperf.com
trade@hoperf.com
Website: http://www.hoperf.com
http://www.hoperf.cn
http://hoperf.en.alibaba.com

RF01 programming guide

1. Brief description

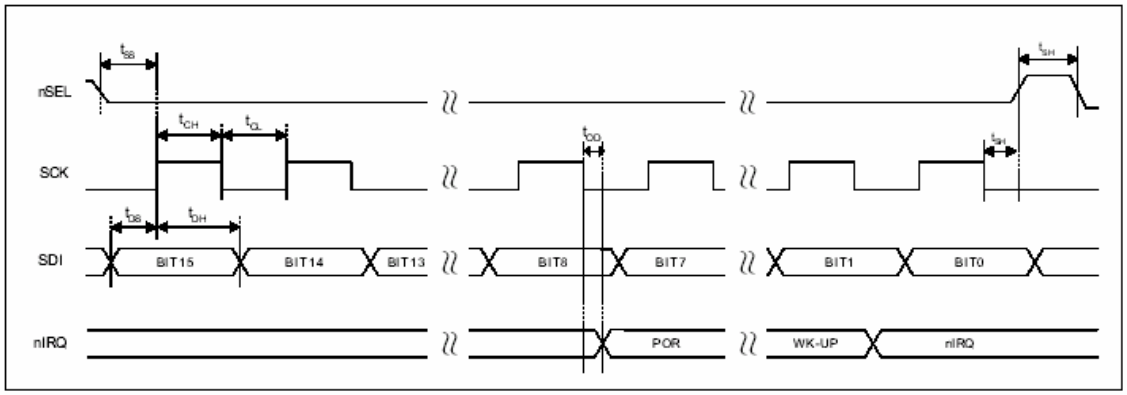
RF01 is a low cost FSK receive IC witch integrated all RF functions in a single chip. It only need a MCU, a crystal, a decouple capacitor and antenna to build a hi reliable FSK receiver. The operation frequency can cover 300 to 1000MHz.

RF01 supports a command interface to setup frequency, deviation, output power and also data rate. No need any hardware adjustment when using in frequency-hopping applications

RF01 can be used in applications such as remote control toys, wireless alarm, wireless sensor, wireless keyboard/mouse, home-automation and wireless data collection.

2. Commands

1. Timing diagram



2. Configuration Setting Command

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	0	0	b1	b0	eb	et	ex	x3	x2	x1	x0	i2	i1	i0	dc	893Ah

b1..b0: select band

b1	b0	band[MHz]
0	0	315
0	1	433
1	0	868
1	1	915

eb：Enable low battery detection function

et：Enable wake-up timer

ex：Enable crystal oscillator

x3..x0: select crystal load capacitor

x3	x2	x1	x0	load capacitor [pF]
0	0	0	0	8.5
0	0	0	1	9.0
0	0	1	0	9.5
0	0	1	1	10.0
……		……		
1	1	1	0	15.5
1	1	1	1	16.0

i2..i0:select baseband bandwidth

i2	i1	i0	Baseband Bandwidth [kHz]
0	0	0	reserved
0	0	1	400
0	1	0	340
0	1	1	270
1	0	0	200
1	0	1	134
1	1	0	67
1	1	1	reserved

dc：Disable signal output of CLK pin

3. Frequency Setting Command

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	0	1	0	f11	f10	f9	f8	f7	f6	f5	f4	f3	f2	f1	f0	A680h

f11..f0: Set operation frequency

315band: Fc=310+F*0.0025 MHz

433band: Fc=430+F*0.0025 MHz

868band: Fc=860+F*0.0050 MHz

915band: Fc=900+F*0.0075 MHz

Fc is carrier frequency, F is frequency parameter and 36≤F≤3903

4. Receiver Setting Command

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	1	0	0	0	0	0	0	d1	d0	g1	g0	r2	r1	r0	en	C0C1h

d1..d0: select VDI source

d1	d0	VDI output
0	0	Digital RSSI output(DRSSI)
0	1	Data quality detection output (DQD)
1	0	Clock recovery lock output
1	1	Always on

g1..g0: select LNA gain

g1	g0	LNA gain (dBm)
0	0	0
0	1	-14
1	0	-6
1	1	-20

r2..r0: select DRSSI threshold

r2	r1	r0	RSSIsetth [dBm]
0	0	0	-103
0	0	1	-97
0	1	0	-91
0	1	1	-85
1	0	0	-79
1	0	1	-73
1	1	0	-67
1	0	1	-61

The actual DRSSI threshold is related to LNA setup:

$RSSI_{th} = RSSI_{setth} + G_{LNA.}$

en: Enable the receiver

5. Wake-Up Timer Command

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	1	1	r4	r3	r2	r1	r0	m7	m6	m5	m4	m3	m2	m1	m0	E196h

The wake-up period is determined by:

$T_{wake-up} = M * 2^{R}$ [ms]

For continual operation, bit ‘et’ must be cleared and set

6. Low Duty-Cycle Command

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	1	0	0	1	1	0	0	d6	d5	d4	d3	d2	d1	d0	en	CCOEh

d6..d0: Set duty cycle

D.C.= (D * 2 +1) / M *100%

en：Enable low duty cycle mode

7. Low Battery Detector and Microcontroller Clock Divider Command

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	1	0	0	0	0	1	0	d2	d1	d0	t4	t3	t2	t1	t0	C200h

d2..d0: select frequency of CLK pin

d2	d1	d0	Clock frequency[MHz]
0	0	0	1
0	0	1	1.25
0	1	0	1.66
0	1	1	2
1	0	0	2.5
1	0	1	3.33
1	1	0	5
1	1	1	10

CLK signal is derive form crystal oscillator and it can be applied to MCU clock in to save a second crystal.

If not used, please set bit “dc” to disable CLK output

To integrate the load capacitor internal can not only save cost, but also adjust reference frequency by software

t4..t0: Set threshold voltage of Low battery detector：

Vlb=2.2+T*0.1 [V]

8. **AFC Command**

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	1	0	0	0	1	1	0	a1	a0	rl1	rl0	st	fi	oe	en	C6F7h

a1..a0: select AFC auto-mode：

a1	a0	
0	0	Controlled by MCU
0	1	Run once at power on
1	0	Keep offset when VDI hi
1	1	Keeps independently from VDI

rl1..rl0: select range limit

r1	r0	range（fres）
0	0	No restriction
0	1	+15/-16
1	0	+7/-8
1	1	+3-4

fres

315，433band: 2.5kHz

868band: 5kHz

915band: 7.5kHz

st: st goes hi will store offset into output register

fi: Enable AFC hi accuracy mode

oe: Enable AFC output register

en: Enable AFC funcition

9. **Data Filter Command**

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	1	0	0	0	1	0	0	al	ml	1	s1	s0	f2	f1	f0	C42Ch

al: Enable clock recovery auto-lock

ml: Enable clock recovery fast mode

s1..s0: select data filter type

s1	s0	Filter type
0	0	OOK
0	1	Digital filter
1	0	reserved

f1..f0: Set DQD threshold

10. Data Rate Command

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	1	0	0	1	0	0	0	cs	r6	r5	r4	r3	r2	r1	r0	C823h

r7..r0: Set data rate

BR=10000000/29/（R+1）/（1+cs*7）

11. Output and FIFO mode Command

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	1	0	0	1	1	1	0	f3	f2	f1	f0	s1	s0	ff	fe	CE85h

f3..f0: Set FIFO interrupt level

s1..s0: select FIFO fill start condition

s1	s0	
0	0	VDI
0	1	Sync-word
1	0	VDI & Sync-word
1	1	Always

ff: Enable FIFO fill

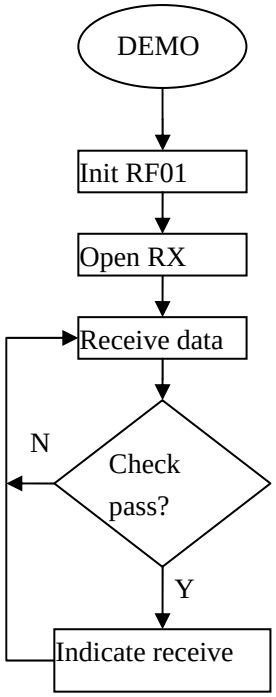
fe: Enable FIFO function

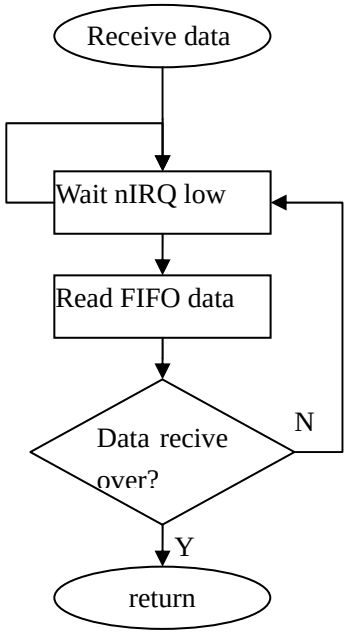
12. Status Read Command

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	-

This command starts with a 0 and be used to read internal status register

3. Demo flow diagram

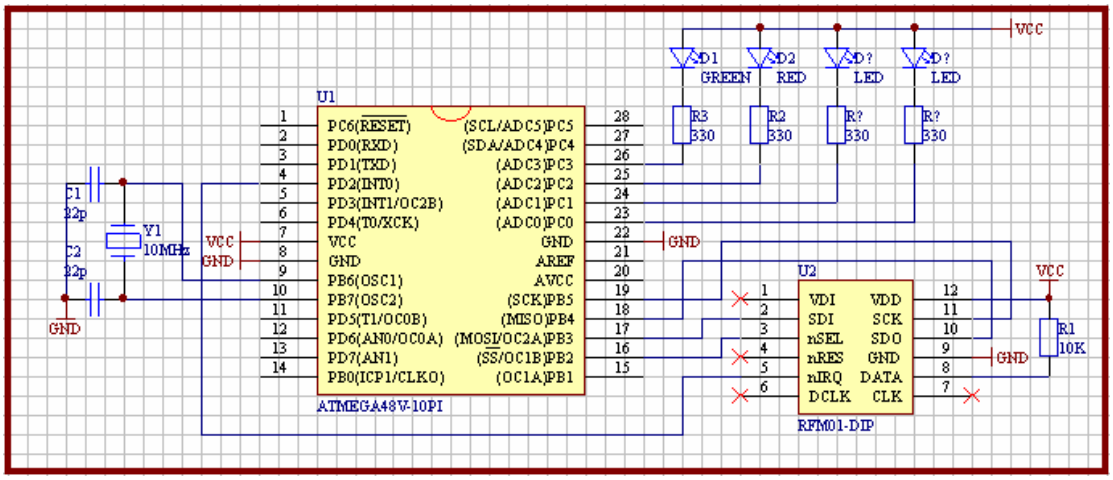




Note: After RF01 initialization, Open FIFO receive mode and wait nIRQ low, only then MCU can

read received and stored in FIFO data. For next package receive, please reset FIFO.

4. Example 1（for AVR microcontroller）




```
/*********************************************************************
    copyright (c) 2006
Title:              RF01 simple example based on AVR C
Current version:    v1.0
Function:           Package Receive Demo
processor           ATMEGA48
Clock:              10MHz Crystal
Operate frequency:  434MHz
Data rate:          4.8kbps
Package size:       23byte
Author:             Tank
Company:            Hope microelectronic Co.,Ltd.
Contact:            +86-0755-86106557
E-MAIL:             hopefsk@hoperf.com
Date:               2006-10-24

                Connections

      ATMEGA48 SIDE       RF01 SIDE
        SCK---------------->SCK
       MISO<----------------SDO
       MOSI---------------->SDI
         SS---------------->nSEL
                            DATA:Pull up to VDD
```

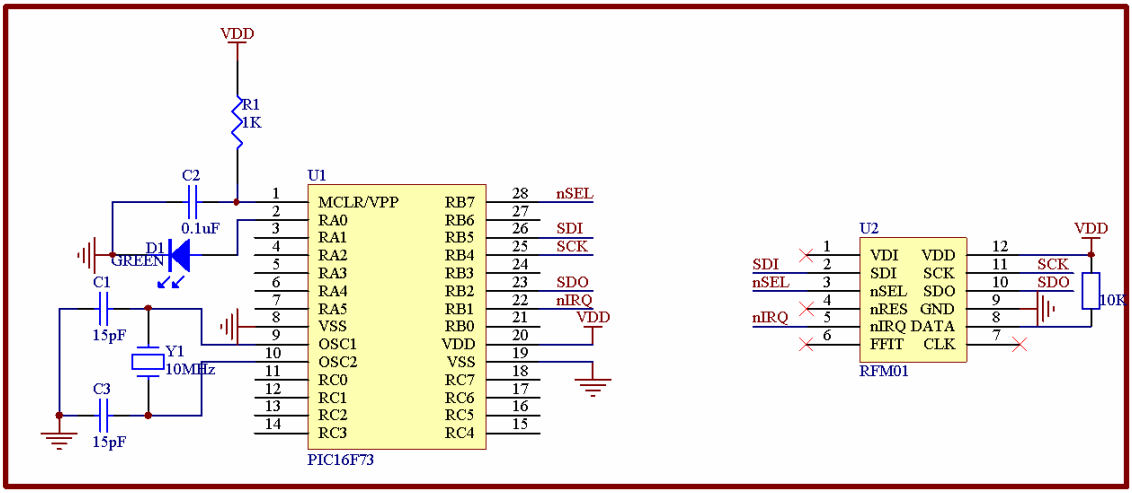
```
        INT0<----------------nIRQ

        PC0~PC3: LED0~LED3

*******************************************************************/
#include <mega48.h>

#define DDR_IN          0
#define DDR_OUT         1

#define PORT_SEL        PORTB
#define PIN_SEL         PINB
#define DDR_SEL         DDRB

#define PORT_SDI        PORTB
#define PIN_SDI         PINB
#define DDR_SDI         DDRB

#define PORT_SCK        PORTB
#define PIN_SCK         PINB
#define DDR_SCK         DDRB

#define PORT_SDO        PORTB
#define PIN_SDO         PINB
#define DDR_SDO         DDRB

#define PORT_LED        PORTC
#define DDR_LED         DDRC

#define PB7             7//--\
#define PB6             6//  |
#define RFXX_SCK        5//  |
#define RFXX_SDO        4//  |RF_PORT
#define RFXX_SDI        3//  |
#define RFXX_SEL        2//  |
#define RFXX_DATA       1//  |
#define PB0             0//--/

#define SEL_OUTPUT()    DDR_SEL |= (1<<RFXX_SEL)
#define HI_SEL()        PORT_SEL|= (1<<RFXX_SEL)
#define LOW_SEL()       PORT_SEL&=~(1<<RFXX_SEL)

#define SDI_OUTPUT()    DDR_SDI |= (1<<RFXX_SDI)
#define HI_SDI()        PORT_SDI|= (1<<RFXX_SDI)
```

```
#define LOW_SDI()       PORT_SDI&=~(1<<RFXX_SDI)

#define SDO_INPUT()     DDR_SDO&= ~(1<<RFXX_SDO)
#define SDO_HI()        PIN_SDO&(1<<RFXX_SDO)

#define SCK_OUTPUT()    DDR_SCK |= (1<<RFXX_SCK)
#define HI_SCK()        PORT_SCK|= (1<<RFXX_SCK)
#define LOW_SCK()       PORT_SCK&=~(1<<RFXX_SCK)

#define LED_OUTPUT()    DDR_LED |=0x0F
#define LED0_ON()       PORT_LED&=~(1<<0)
#define LED0_OFF()      PORT_LED|= (1<<0)
#define LED0_TRG()      PORT_LED^= (1<<0)

#define LED1_ON()       PORT_LED&=~(1<<1)
#define LED1_OFF()      PORT_LED|= (1<<1)
#define LED1_TRG()      PORT_LED^= (1<<1)

#define LED2_ON()       PORT_LED&=~(1<<2)
#define LED2_OFF()      PORT_LED|= (1<<2)
#define LED2_TRG()      PORT_LED^= (1<<2)

#define LED3_ON()       PORT_LED&=~(1<<3)
#define LED3_OFF()      PORT_LED|= (1<<3)
#define LED3_TRG()      PORT_LED^= (1<<3)

unsigned char RF_RXBUF[22];
void RFXX_PORT_INIT(void){
  HI_SEL();
  HI_SDI();
  LOW_SCK();
  SEL_OUTPUT();
  SDI_OUTPUT();
  SDO_INPUT();
  SCK_OUTPUT();
}
unsigned int RFXX_WRT_CMD(unsigned int aCmd){
  unsigned char i;
  unsigned int temp;
  LOW_SCK();
  LOW_SEL();
  for(i=0;i<16;i++){
    temp<<=1;
    if(SDO_HI()){
```

```
      temp|=0x0001;
    }
    LOW_SCK();

    if(aCmd&0x8000){
      HI_SDI();
    }else{
      LOW_SDI();
    }
    HI_SCK();

    aCmd<<=1;
  };
  LOW_SCK();
  HI_SEL();
  return(temp);
}
unsigned char RF01_RDFIFO(void){
  unsigned char i,Result;
  LOW_SCK();
  LOW_SDI();
  LOW_SEL();
  for(i=0;i<16;i++){//skip status bits
    HI_SCK();
    HI_SCK();
    LOW_SCK();
    LOW_SCK();
  }
  Result=0;
  for(i=0;i<8;i++){//read fifo data byte
    Result<<=1;
    if(SDO_HI()){
      Result|=1;
    }
    HI_SCK();
    HI_SCK();
    LOW_SCK();
    LOW_SCK();
  };
  HI_SEL();
  return(Result);
}
void main(void)
{
```

```
  unsigned int intI,intJ;
  unsigned char i,j,ChkSum;
  for(intI=0;intI<10000;intI++)for(intJ=0;intJ<123;intJ++);
  RFXX_PORT_INIT();

  RFXX_WRT_CMD(0x0000);
  RFXX_WRT_CMD(0x898A);//433BAND,134kHz
  RFXX_WRT_CMD(0xA640);//434MHz
  RFXX_WRT_CMD(0xC847);//4.8kbps
  RFXX_WRT_CMD(0xC69B);//AFC setting
  RFXX_WRT_CMD(0xC42A);//Clock recovery manual control,Digital filter,DQD=4
  RFXX_WRT_CMD(0xC240);//output 1.66MHz
  RFXX_WRT_CMD(0xC080);
  RFXX_WRT_CMD(0xCE88);//use FIFO
  RFXX_WRT_CMD(0xCE8B);
  RFXX_WRT_CMD(0xC081);//OPEN RX
  DDRB|=(1<<RFXX_DATA);

  DDRD&=~(1<<2);


  LED_OUTPUT();

  i=0;
  while(1){
    while(!(PIND&(1<<2))){//polling the nIRQ data
      RF_RXBUF[i++]=RF01_RDFIFO();//read FIFO data
      if(i==18){
        i=0;
        RFXX_WRT_CMD(0xCE88);    //reset FIFO for next frame recognition
        RFXX_WRT_CMD(0xCE8B);
        ChkSum=0;
        for(j=0;j<16;j++){
          ChkSum+=RF_RXBUF[j];  //calculate checksum
        }
        if(ChkSum==RF_RXBUF[16]){//frame check
          LED0_TRG();//receive indication
        }
      }
    }
  }
}
```

5. Example 2（for PIC microcontroller）




```
/*************************************************************
    copyright (c) 2006
Title:              RF01 simple example based on PIC C
Current version:    v1.0
Function:           Package receive Demo
Processor           PIC16F73
Clock:              10MHz Crystal
Operate frequency:  434MHz
Data rate:          4.8kbps
Package size:       23byte
Author:             Robben
Company:            Hope microelectronic Co.,Ltd.
Contact:            +86-0755-86106557
E-MAIL:             hopefsk@hoperf.com
Date:               2006-11-14
*************************************************************/
#include "pic.h"
typedef unsigned char uchar;
typedef unsigned int  uint;

#define SDI           RB5
#define SCK           RB4
#define SDO           RB2
```

```
#define nIRQ          RB1
#define nSEL          RB7
#define LED           RA0
#define LED_OUT()     TRISA0=0
#define nIRQ_IN()     TRISB1=1
#define SDI_OUT()     TRISB5=0
#define SCK_OUT()     TRISB4=0
#define SDO_IN()      TRISB2=1
#define DATA_IN()     TRISB1=1
#define nSEL_OUT()    TRISB7=0

void RF1_Init( void );
void Write0( void );
void Write1( void );
void Delayus( uint us );
void WriteCMD( uint CMD );
uchar RF01_RDFIFO(void);
void Delays(void);
__CONFIG(0x3FF2);
bank1 uchar RF_RXBUF[19];

void RF1_Init( void )
{
  nSEL=1;
  SDI=1;
  SCK=0;
  nSEL_OUT();
  SDI_OUT();
  SDO_IN();
  nIRQ_IN();
  SCK_OUT();
  LED_OUT();
  LED=0;
}

void main()
{
  uchar i=0,j=0;
  uint  CheckSum;
  Delays();
  RF1_Init( );
  WriteCMD(0x0000);
  WriteCMD(0x898A);//433BAND,134kHz
  WriteCMD(0xA640);//434MHz
```

```
  WriteCMD(0xC847);//4.8kbps
  WriteCMD(0xC69B);//AFC setting
  WriteCMD(0xC42A);//Clock recovery manual control,Digital filter,DQD=4
  WriteCMD(0xC240);//output 1.66MHz
  WriteCMD(0xC080);
  WriteCMD(0xCE88);//use FIFO
  WriteCMD(0xCE8B);
  WriteCMD(0xC081);//OPEN RX
  while(1)
   {
     while(!nIRQ)
     {
      RF_RXBUF[i++]=RF01_RDFIFO();
      if(i==17)
       {
        i=0;
        WriteCMD(0xCE88);
        WriteCMD(0xCE8B);
        CheckSum=0;
        for(j=0;j<16;j++)
         CheckSum+=RF_RXBUF[j];  //add 0x30-----0x3F
        CheckSum&=0x0FF;
        if(CheckSum==RF_RXBUF[16])
         {
           LED=1;
         }
        Delayus(1);
        LED=0;
        }
       }
   }
}

void Write0( void )
{
  SDI=0;
  SCK=0;
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
```

```
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  SCK=1;
  NOP();
}

void Write1( void )
{
  SDI=1;
  SCK=0;
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  SCK=1;
  NOP();
}

void WriteCMD( uint CMD )
{
  uchar n=16;
  SCK=0;
  nSEL=0;
  while(n--)
   {
```

```
    if(CMD&0x8000)
     Write1();
    else
     Write0();
    CMD=CMD<<1;
   }
  SCK=0;
  nSEL=1;
}

uchar RF01_RDFIFO(void)
{
  uchar i,Result;
  SCK=0;
  SDI=0;
  nSEL=0;
  for(i=0;i<16;i++)
  {                      //skip status bits
    SCK=1;
    NOP();
    NOP();
    SCK=0;
    NOP();
    NOP();
  }
  Result=0;
  for(i=0;i<8;i++)
   {                     //read fifo data byte
    Result=Result<<1;
    if(SDO)
    {
      Result|=1;
    }
    SCK=1;
    NOP();
    NOP();
    SCK=0;
    NOP();
    NOP();
   }
  nSEL=1;
  return(Result);
}
```

```
void Delayus( uint us )
{
  uint i;
  while( us-- )
    {
       i=1000;
       while( i-- )
         {
           NOP();
         }
    }
}

void Delays(void)
{
  uchar i=10;
  while(i--)
   {
     Delayus(1);
   }
}
```

HOPE MICROELECTRONICS CO.,LTD
Address: Rm B.8/F LiJingGe Emperor Regency 6012 ShenNan Rd, Shenzhen, China
Tel: 86-755-82973805
Fax: 86-755-82973550
Email: sales@hoperf.com
trade@hoperf.com
Website: http://www.hoperf.com
http://www.hoperf.cn
http://hoperf.en.alibaba.com