User Manual of NRF24L01 Breakout Board

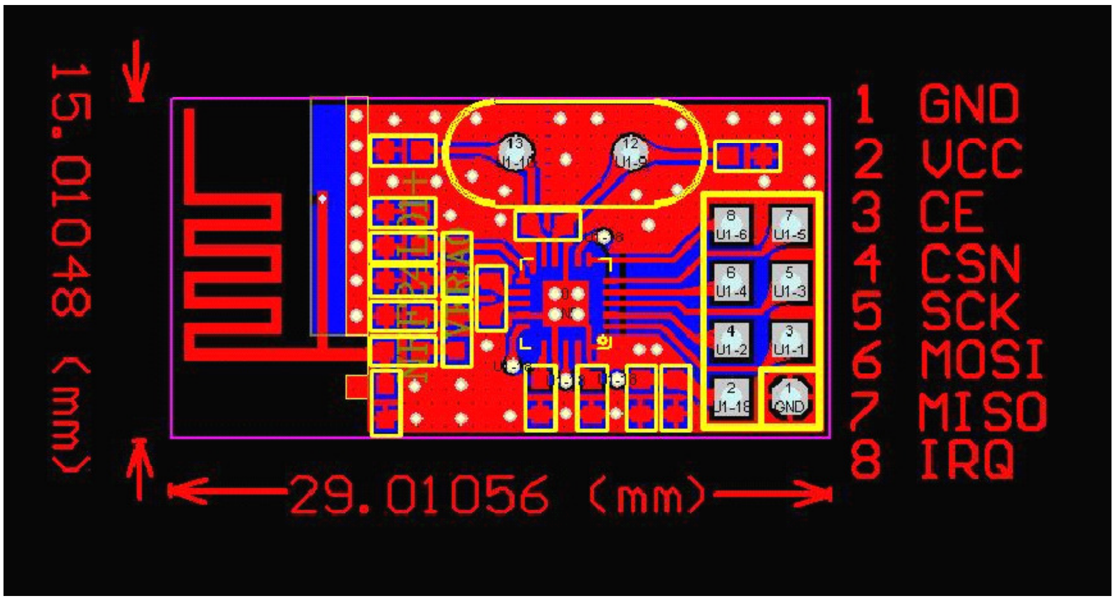
LinkSprite Technologies, Inc

December 2010

1. Introduction

1. 2.4GHz ISM frequency band
2. Max data rate 2Mbps, GFSK modulation, robust anti-interference, especially ideal for industry application.
3. 126 channels, multiple points and frequency hopping
4. Embedded hardware CRC and star topology address control
5. Low power 1.9V- 3.6V. Current 22uA in idle mode, 900nA in sleep mode
6. On-board 2.4Ghz antenna, size 15mm X 29mm
7. Firmware programmed address. Only output data when being addressed and can generate interrupt. It can be used directly with most MCUs.
8. On-board regulator
9. 2.54MM header interface
10. Automatic packet handling, and auto packet transaction handling in Enhanced ShockBurst mode. Optional built-in packet acknowledgment mechanism to reduce packet loss.
11. For 5V MCU, please connect a resistor of 2K in between the IOs of the module and MCU to reduce the current. For 3.3V MCU, the IOs of MCU can be connected directly to those of RF24L01 module.

2. Interface Circuit

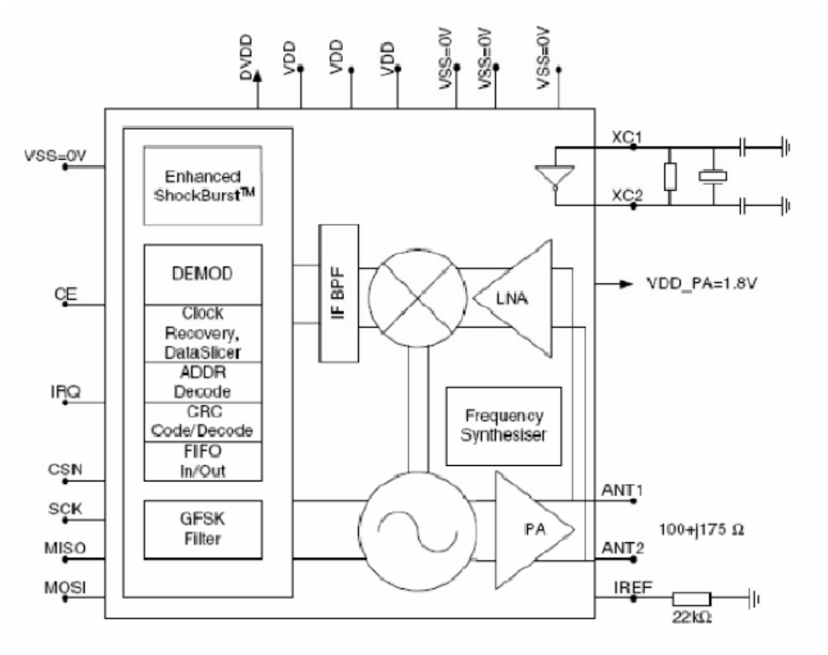


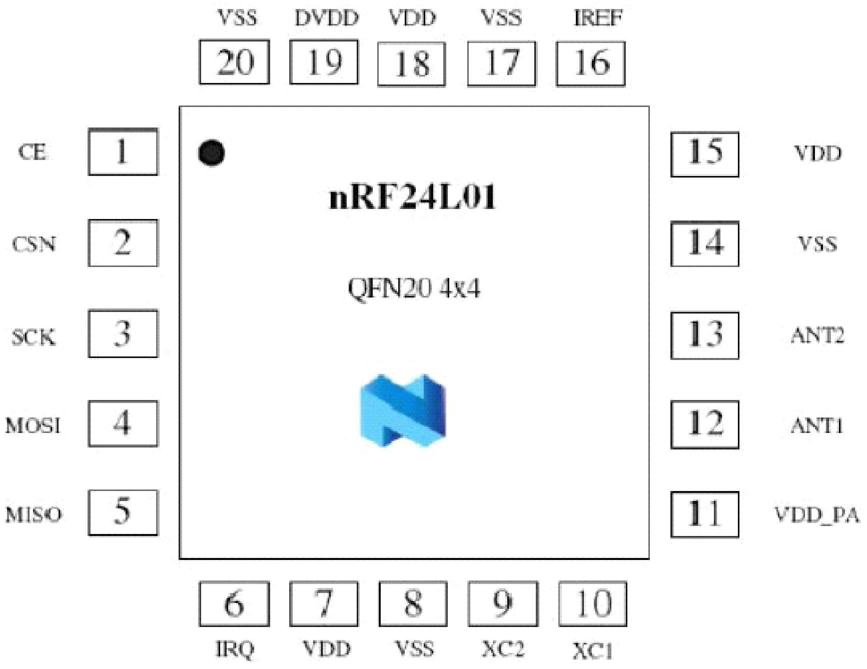
Note:

1. VCC is between 1.9V and 3.6V. Nominal 3.3V.
2. Except VCC and GND, other pins can be connected to 5V MCU with a resistor to limit the current.

3. Diagram of NRF24L01 Module and Pins Description

NRF24L01 module uses nRF24L01 chipset from Nordic.





Pin	Name	Pin function	Description
1	CE	Digital Input	Chip Enable Activates RX or TX mode
2	CSN	Digital Input	SPI Chip Select
3	SCK	Digital Input	SPI Clock
4	MOSI	Digital Input	SPI Slave Data Input
5	MISO	Digital Output	SPI Slave Data Output, with tri-state option
6	IRQ	Digital Output	Maskable interrupt pin
7	VDD	Power	Power Supply (+3V DC)
8	VSS	Power	Ground (0V)
9	XC2	Analog Output	Crystal Pin 2
10	XC1	Analog Input	Crystal Pin 1
11	VDD_PA	Power Output	Power Supply (+1.8V) to Power Amplifier
12	ANT1	RF	Antenna interface 1
13	ANT2	RF	Antenna interface 2
14	VSS	Power	Ground (0V)
15	VDD	Power	Power Supply (+3V DC)
16	IREF	Analog Input	Reference current
17	VSS	Power	Ground (0V)
18	VDD	Power	Power Supply (+3V DC)
19	DVDD	Power Output	Positive Digital Supply output for de-coupling purposes
20	VSS	Power	Ground (0V)

4. Diagram of NRF24L01 Module and Pins Description

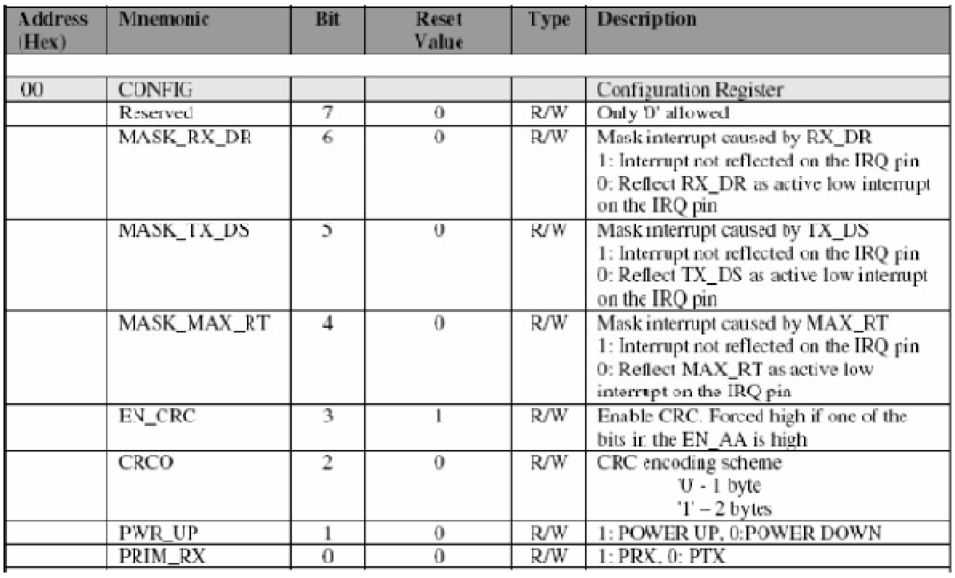
There are four working modes of NRF24L01:

- Transceiver mode
- Configuration mode
- Idle mode
- Sleep mode

The working mode is determined by three things: PWR_UP register, PRIM_RX register, and CE. It's shown as below:

Mode	PWR_UP register	PRIM_RX register	CE	FIFO state
RX mode	1	1	1	-
TX mode	1	0	1	Data in TX FIFO
TX mode	1	0	1→0	Stays in TX mode until packet transmission is finished
Standby-II	1	0	1	TX FIFO empty
Standby-I	1	-	0	No ongoing packet transmission
Power Down	0	-	-	-

5. NRF24L01 Configuration



Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
00	CONFIG				Configuration Register
	Reserved	7	0	R/W	Only '0' allowed
	MASK_RX_DR	6	0	R/W	Mask interrupt caused by RX_DR 1: Interrupt not reflected on the IRQ pin 0: Reflect RX_DR as active low interrupt on the IRQ pin
	MASK_TX_DS	5	0	R/W	Mask interrupt caused by TX_DS 1: Interrupt not reflected on the IRQ pin 0: Reflect TX_DS as active low interrupt on the IRQ pin
	MASK_MAX_RT	4	0	R/W	Mask interrupt caused by MAX_RT 1: Interrupt not reflected on the IRQ pin 0: Reflect MAX_RT as active low interrupt on the IRQ pin
	EN_CRC	3	1	R/W	Enable CRC. Forced high if one of the bits in the EN_AA is high
	CRCO	2	0	R/W	CRC encoding scheme '0' - 1 byte '1' – 2 bytes
	PWR_UP	1	0	R/W	1: POWER UP, 0:POWER DOWN
	PRIM_RX	0	0	R/W	1: PRX, 0: PTX

6. Sample Code

```
#include <reg51.h>
//<nRF2401_Pins>
sbit MISO =P1^3;
sbit MOSI =P1^4;
sbit SCK =P1^5;
sbit CE =P1^6;
sbit CSN =P3^7;
sbit IRQ =P1^2;
sbit LED2 =P3^5;
sbit LED1 =P3^4;
sbit KEY1 =P3^0;
sbit KEY2 =P3^1;
// SPI(nRF24L01) commands
#define READ_REG 0x00 // Define read command to register
#define WRITE_REG 0x20 // Define write command to register
#define RD_RX_PLOAD 0x61 // Define RX payload register address
#define WR_TX_PLOAD 0xA0 // Define TX payload register address
#define FLUSH_TX 0xE1 // Define flush TX register command
#define FLUSH_RX 0xE2 // Define flush RX register command
#define REUSE_TX_PL 0xE3 // Define reuse TX payload register command
```

```
#define NOP 0xFF // Define No Operation, might be used to read status register
//***************************************************//
// SPI(nRF24L01) registers(addresses)
#define CONFIG 0x00 // 'Config' register address
#define EN_AA 0x01 // 'Enable Auto Acknowledgment' register address
#define EN_RXADDR 0x02 // 'Enabled RX addresses' register address
#define SETUP_AW0x03 // 'Setup address width' register address
#define SETUP_RETR 0x04 // 'Setup Auto. Retrans' register address
#define RF_CH 0x05 // 'RF channel' register address
#define RF_SETUP 0x06 // 'RF setup' register address
#define STATUS 0x07 // 'Status' register address
#define OBSERVE_TX 0x08 // 'Observe TX' register address
#define CD 0x09 // 'Carrier Detect' register address
#define RX_ADDR_P0 0x0A // 'RX address pipe0' register address
#define RX_ADDR_P1 0x0B // 'RX address pipe1' register address
#define RX_ADDR_P2 0x0C // 'RX address pipe2' register address
#define RX_ADDR_P3 0x0D // 'RX address pipe3' register address
#define RX_ADDR_P4 0x0E // 'RX address pipe4' register address
#define RX_ADDR_P5 0x0F // 'RX address pipe5' register address
#define TX_ADDR 0x10 // 'TX address' register address
#define RX_PW_P0 0x11 // 'RX payload width, pipe0' register address

#define RX_PW_P1 0x12 // 'RX payload width, pipe1' register address
#define RX_PW_P2 0x13 // 'RX payload width, pipe2' register address
#define RX_PW_P3 0x14 // 'RX payload width, pipe3' register address
#define RX_PW_P4 0x15 // 'RX payload width, pipe4' register address
#define RX_PW_P5 0x16 // 'RX payload width, pipe5' register address
#define FIFO_STATUS 0x17 // 'FIFO Status Register' register address
//------------------------------------------------------------
// Write one byte to 24L01, and read one byte out
uchar SPI_RW(uchar byte)
{
uchar bit_ctr;
for(bit_ctr=0;bit_ctr<8;bit_ctr++) // output 8-bit
{
MOSI = (byte & 0x80); // output 'byte', MSB to MOSI
byte = (byte << 1); // shift next bit into MSB..
SCK = 1; // Set SCK high..
byte |= MISO; // capture current MISO bit
SCK = 0; // ..then set SCK low again
}
return(byte); // return read byte
}
// write one byte to register, and return the status
```

```
uchar SPI_RW_Reg(BYTE reg, BYTE value)
{
uchar status;
CSN = 0; // CSN low, init SPI transaction
status = SPI_RW(reg); // select register
SPI_RW(value); // ..and write value to it..
CSN = 1; // CSN high again
return(status); // return nRF24L01 status byte
}
// read number of bytes data
uchar SPI_Read_Buf(BYTE reg, BYTE *pBuf, BYTE bytes)
{
uchar status,byte_ctr;
CSN = 0; // Set CSN low, init SPI tranaction
status = SPI_RW(reg); // Select register to write to and read status byte
for(byte_ctr=0;byte_ctr<bytes;byte_ctr++)
pBuf[byte_ctr] = SPI_RW(0); //
CSN = 1;
return(status); // return nRF24L01 status byte
}
// Write number of bytes to buffer
uchar SPI_Write_Buf(BYTE reg, BYTE *pBuf, BYTE bytes)
{
uchar status,byte_ctr;
CSN = 0;
status = SPI_RW(reg);
for(byte_ctr=0; byte_ctr<bytes; byte_ctr++) //
SPI_RW(*pBuf++);
CSN = 1; // Set CSN high again
return(status); //
}
// receiving function, 1 means received
unsigned char nRF24L01_RxPacket(unsigned char* rx_buf)
{
unsigned char revale=0;
// set in RX mode
SPI_RW_Reg(WRITE_REG + CONFIG, 0x0f); // Set PWR_UP bit, enable CRC(2 bytes) &
Prim:RX. RX_DR enabled..
CE = 1; // Set CE pin high to enable RX device
dalay130us();
sta=SPI_Read(STATUS); // read register STATUS's value
if(RX_DR) // if receive data ready (RX_DR) interrupt
{
CE = 0; // stand by mode
SPI_Read_Buf(RD_RX_PLOAD,rx_buf,TX_PLOAD_WIDTH);// read receive payload from
RX_FIFO buffer
revale =1;
}
SPI_RW_Reg(WRITE_REG+STATUS,sta);// clear RX_DR or TX_DS or MAX_RT interrupt
flag
return revale;
}
// transmit function
void nRF24L01_TxPacket(unsigned char * tx_buf)
{
CE=0;
//SPI_Write_Buf(WRITE_REG + TX_ADDR, TX_ADDRESS, TX_ADR_WIDTH); // Writes
TX_Address to nRF24L01
//SPI_Write_Buf(WRITE_REG + RX_ADDR_P0, TX_ADDRESS, TX_ADR_WIDTH); //
RX_Addr0 same as TX_Adr for Auto.Ack
SPI_Write_Buf(WR_TX_PLOAD, tx_buf, TX_PLOAD_WIDTH); // Writes data to TX payload
SPI_RW_Reg(WRITE_REG + CONFIG, 0x0e); // Set PWR_UP bit, enable CRC(2 bytes) &
Prim:TX. MAX_RT & TX_DS enabled..
CE=1;
dalay10us();
CE=0;
}
// configuration
void nRF24L01_Config(void)
{
//initial io
CE=0; // chip enable
CSN=1; // Spi disable
SCK=0; // Spi clock line init high
CE=0;
SPI_RW_Reg(WRITE_REG + CONFIG, 0x0f); // Set PWR_UP bit, enable CRC(2 bytes) &
Prim:RX. RX_DR enabled..
SPI_RW_Reg(WRITE_REG + EN_AA, 0x01);
SPI_RW_Reg(WRITE_REG + EN_RXADDR, 0x01); // Enable Pipe0
SPI_RW_Reg(WRITE_REG + SETUP_AW, 0x02); // Setup address width=5 bytes
SPI_RW_Reg(WRITE_REG + SETUP_RETR, 0x1a); // 500us + 86us, 10 retrans...
SPI_RW_Reg(WRITE_REG + RF_CH, 0);
SPI_RW_Reg(WRITE_REG + RF_SETUP, 0x07); // TX_PWR:0dBm, Datarate:1Mbps,
LNA:HCURR
SPI_RW_Reg(WRITE_REG + RX_PW_P0, RX_PLOAD_WIDTH);
SPI_Write_Buf(WRITE_REG + TX_ADDR, TX_ADDRESS, TX_ADR_WIDTH);
SPI_Write_Buf(WRITE_REG + RX_ADDR_P0, TX_ADDRESS, TX_ADR_WIDTH); CE=1; //
}
```

LinkSprite Technologies, Inc.

Add：1067 S Hover St, Unit E-186,Longmont, CO 80501

Tel：   720-204-8599

Email：sales@linksprite.com

Technical questions: support@linksprite.com

Web：www.linksprite.com