nRF24L01+PA+LNA 2.4GHz RF Transceiver Module

This module based on Nordic nRF24L01+ with integrated Power Amplifier (PA) and Low-Noise-Amplifier (LNA) for extended range of up to 1,000 meter. This transceiver IC operates in the 2.4GHz band and has many new features! Take all the coolness of the nRF2401A and add some extra pipelines, buffers, and an auto-retransmit feature. You can use it in you project without any wireless design, just need to leave a 8 pins interface to this module, use the SPI interface to control the module on transferring data.
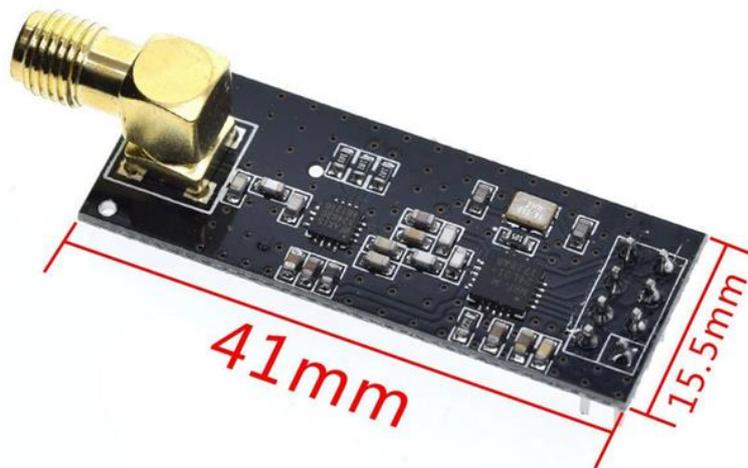
SKU: MDU1087

Brief Data:

- Worldwide license-free 2.4GHz ISM band.
- Integrated RF Power Amplifier and Low Noise Amplifier (LNA).
- Max Output Power: ~20dBm.
- Receiver Sensitivity: ~-95dBm.
- External SMA Antenna.
- 250kbps, 1Mbps and 2Mbps on-air data-rate.
- Communication Range: >1000+ meter (line of sight).
- 1.9 to 3.6V power supply range.
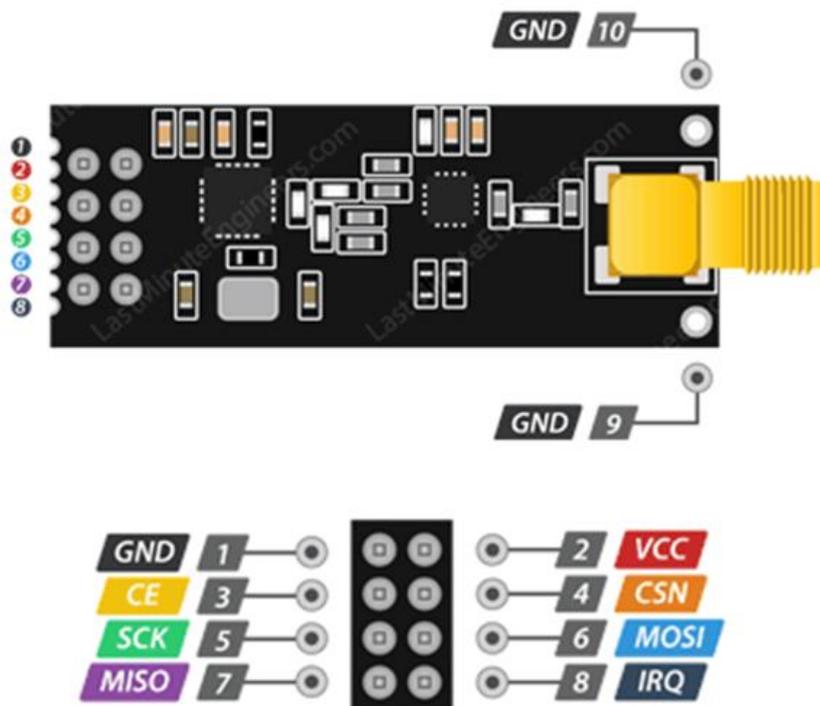- ±60ppm 16MHz crystal.
- 5V tolerant inputs.

Mechanical Dimension:

Unit: mm

41mm

15.5mm

Module Pins Assignment:

nRF24L01+ PA/LNA Pinout

GND 10

GND 9

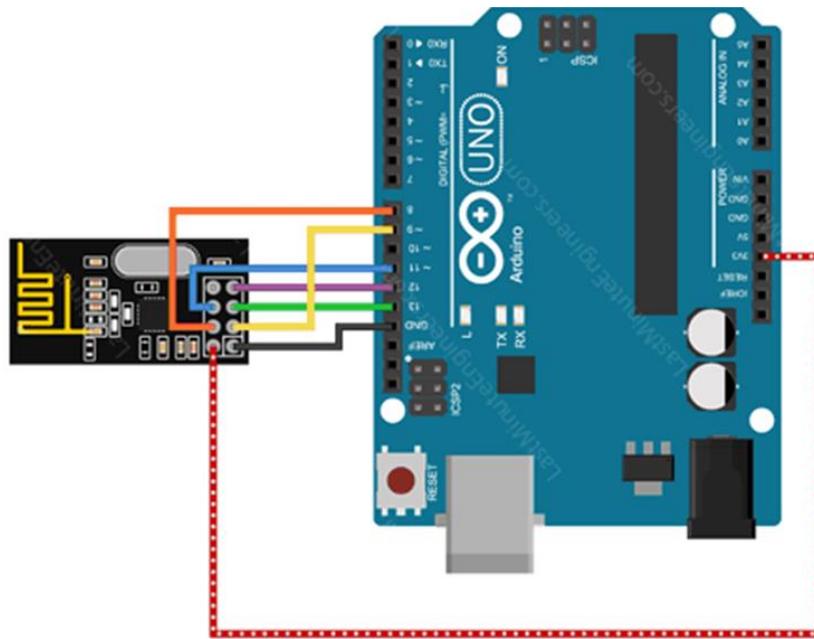| GND | 1 | | | | 2 | VCC |
CE	3				4	CSN
SCK	5				6	MOSI
MISO	7				8	IRQ

Top View Pin Assignment

Connecting NRF24L01+ to Arduino:

We need two modules (with Arduino) to complete this test, one board will act as transmitter and the other as receiver. The schematics for this project is quite simple, connect one NRF24L01 to each of the two Arduino boards that will be used for this project as shown in the schematics below.

NRF24L01+ Pin	Arduino Pin
GND	GND
VCC	+3.3V
CE	D9
CSN	D8
SCK	D13
MOSI	D11
MISO	D12

To interface the Arduino with the module, we'll be using TMRh20's RF24 library, which conveniently packages the low-level communications between the RF module and the MCU into an easy-to-use C++ class.

Download the library from the below link and copy to Arduino IDE libraries folder:

https://github.com/nRF24/RF24

The full documentation on the library can be found in the link.

Arduino Code – For Transmitter:

In our experiment we will just send 'Welcome to Handson Tech' message from the transmitter to the receiver.

Here is the sketch we will be using for our transmitter:

```
//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
```

```
#include <RF24.h>

//create an RF24 object
RF24 radio(9, 8);  // CE, CSN

//address through which two modules communicate.
const byte address[6] = "00001";

void setup()
{
  radio.begin();
  radio.setPALevel(RF24_PA_LOW);//Transmitter RF Power Setting
                                //MIN=-18dBm, LOW=-12dBm, HIGH=-6dBm, MAX=0dBm.
  radio.setChannel(108);          //above most WiFi frequencies. RF Channel setting 0-125
  radio.setDataRate( RF24_250KBPS ) ;

  //set the address
  radio.openWritingPipe(address);

  //Set module as transmitter
  radio.stopListening();
}
void loop()
{
  //Send message to receiver
  const char text[] = "Welcome to Handson Tech";
  radio.write(&text, sizeof(text));

  delay(500);
}
```

Arduino Code – For Receiver:

Here is the sketch we will be using for our receiver

```
//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

//create an RF24 object
RF24 radio(9, 8);  // CE, CSN

//address through which two modules communicate.
const byte address[6] = "00001";

void setup()
{
  while (!Serial);
    Serial.begin(9600);

  radio.begin();
  radio.setPALevel(RF24_PA_LOW);  //Transmitter RF Power Setting
                                //MIN=-18dBm, LOW=-12dBm, HIGH=-6dBm, MAX=0dBm.
  radio.setChannel(108);          //above most WiFi frequencies. RF Channel setting 0-125
  radio.setDataRate( RF24_250KBPS ) ;
    //set the address
  radio.openReadingPipe(0, address);

  //Set module as receiver
  radio.startListening();
}

void loop()
{
  //Read the data if available in buffer
  if (radio.available())
  {
    char text[32] = {0};
    radio.read(&text, sizeof(text));
    Serial.println(text);
  }
}
```

At the end we just print the received message on serial monitor. If you did everything ok and there are no mistakes in connections, you should see something like this in your Serial Monitor.

www.handsontec.com

Web Resources:

- https://www.deviceplus.com/how-tos/arduino-guide/nrf24l01-rf-module-tutorial/
- https://github.com/nRF24/RF24
- https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/
- https://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/
- https://howtomechatronics.com/tutorials/arduino/how-to-build-an-arduino-wireless-network-with-multiple-nrf24l01-modules/

HandsOn Technology provides a multimedia and interactive platform for everyone interested in electronics. From beginner to diehard, from student to lecturer. Information, education, inspiration and entertainment. Analog and digital, practical and theoretical; software and hardware.

Hands*On* Technology support Open Source Hardware (OSHW) Development Platform.

open source hardware

Learn : Design : Share

handsontec.com

The Face behind our product quality…

In a world of constant change and continuous technological development, a new or replacement product is never far away – and they all need to be tested.
Many vendors simply import and sell wihtout checks and this cannot be the ultimate interests of anyone, particularly the customer. Every part sell on Handsotec is fully tested. So when buying from Handsontec products range, you can be confident you're getting outstanding quality and value.

We keep adding the new parts so that you can get rolling on your next project.

Breakout Boards & Modules

Connectors

Electro-Mechanical Parts

Engineering Material

Mechanical Hardware

Electronics Components

P

Power Supply

Arduino Board & Shield

Tools & Accessory