

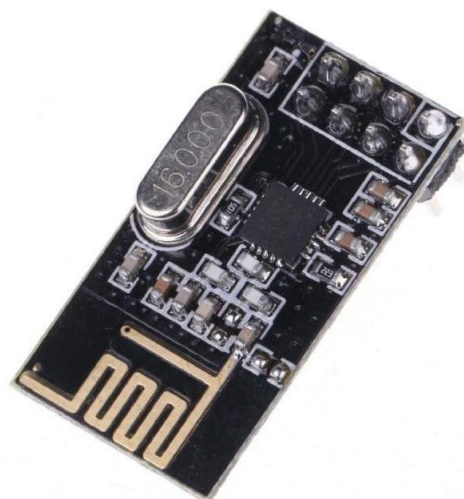
## WIFI modul NRF24L01

### 1. POPIS

Modul umožňuje bezdrátovou komunikaci vývojových kitů Arduino/Genuino (nebo např. Raspberry PI), na frekvenci 2,4 GHz. Tento modul neumožňuje připojení kitu k domácí WiFi síti, komunikace je možná pouze mezi (dvěma či více) moduly.

Základní charakteristika modulu:

- Komunikační rozhraní SPI (serial peripheral interface)
- Integrovaná anténa, komunikace až na vzdálenost 100 m (silně závislé na prostředí)
- Rychlost komunikace až 2Mbps
- 5V tolerantní vstupy
- Kompaktní rozměry



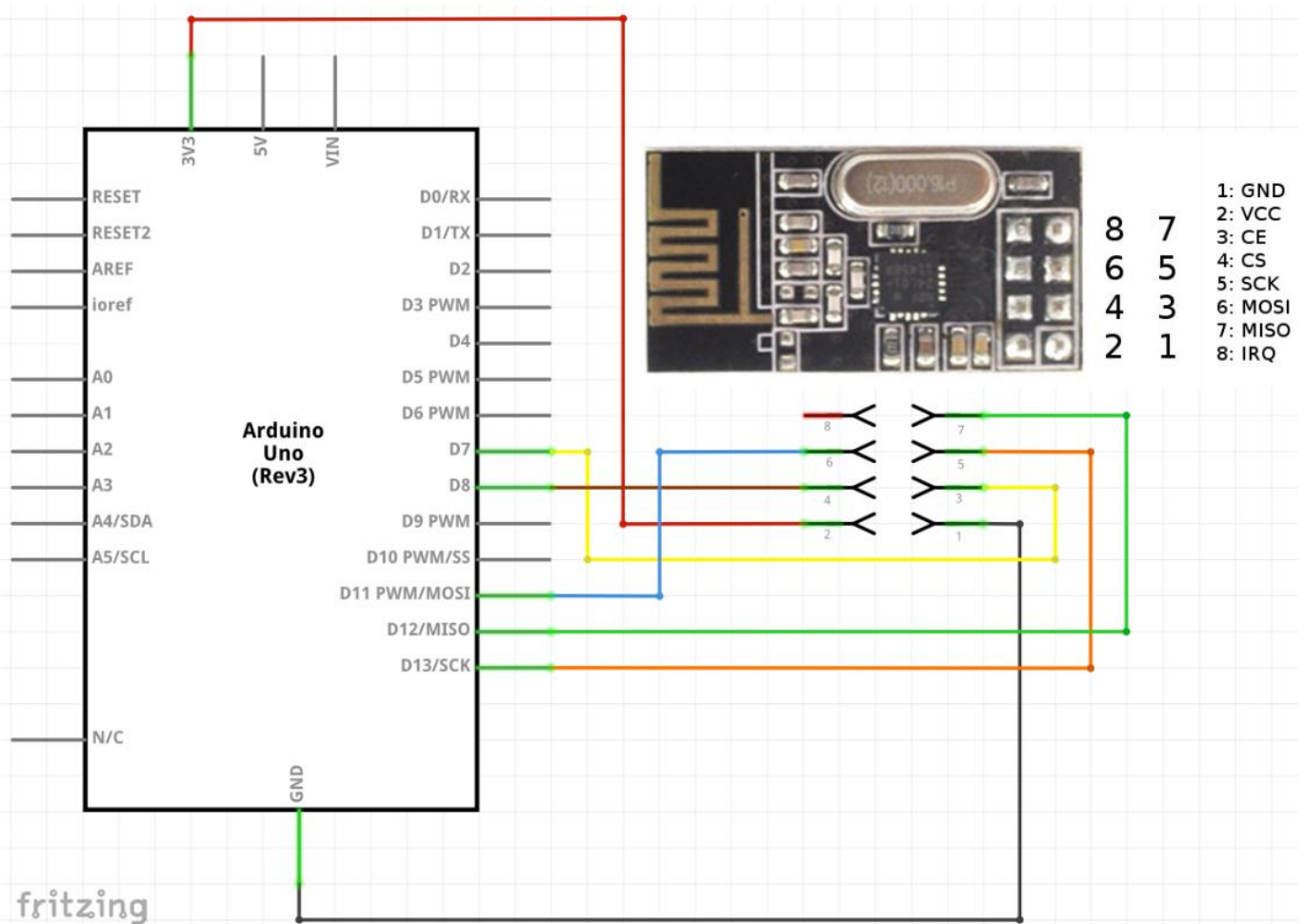
### 2. SPECIFIKACE

<b>Hlavní čip</b>	nRF24L01	<b>Proud (režim standby)</b>	22 uA
<b>Napájecí napětí</b>	1,9–3,6 VDC	<b>Proud (režim powerdown)</b>	900 nA
<b>Počet RF kanálů</b>	126	<b>Start-up (režim standby)</b>	130 us
<b>Komunikační frekvence</b>	2,4 GHz	<b>Start-up (režim powerdown)</b>	1,5 ms
<b>Typ modulace</b>	GFSK	<b>Velikost FIFO bufferu (Rx, Tx)</b>	32 byte
<b>Programovatelné zesílení vysílače (dBm)</b>	0/-6/-12/-18	<b>Rozměry (mm)</b>	28 x 15x 13
<b>Citlivost přijímače</b>	Až -85 dBm	<b>Hmotnost (g)</b>	43



### 3. ZAPOJENÍ

Přijímací i vysílací modul je zapojen stejně. Obrázek převzatý z <http://navody.arduino-shop.cz/navody-k-produktum/arduino-wifi-modul-nrf24l01.html>.



### 4. UKÁZKA PROGRAMU – PŘIJÍMACÍ MODUL

Pro správnou funkci programu nainstalujte do vývojového prostředí Arduino knihovnu [RF24](http://navody.arduino-shop.cz/navody-k-produktum/arduino-wifi-modul-nrf24l01.html). Ukázka je převzata z <http://navody.arduino-shop.cz/navody-k-produktum/arduino-wifi-modul-nrf24l01.html>.

```
// nRF24L01 přijímač
// připojení knihoven
#include <SPI.h>
#include "RF24.h"
// nastavení propojovacích pinů
#define CE 7
```

```

#define CS 8
// inicializace nRF s piny CE a CS
RF24 nRF(CE, CS);
// nastavení adres pro přijímač a vysílač,
// musí být nastaveny stejně v obou programech!
byte adresaPrijimac[]= "prijimac00";
byte adresaVysilac[]= "vysilac00";

void setup() {
  // komunikace přes sériovou linku rychlostí 9600 baud
  Serial.begin(9600);
  // zapnutí komunikace nRF modulu
  nRF.begin();
  // nastavení výkonu nRF modulu,
  // možnosti jsou RF24_PA_MIN, RF24_PA_LOW, RF24_PA_HIGH and RF24_PA_MAX,
  // pro HIGH a MAX je nutný externí 3,3V zdroj
  nRF.setPALevel(RF24_PA_LOW);
  // nastavení zapisovacího a čtecího kanálu
  nRF.openWritingPipe(adresaPrijimac);
  nRF.openReadingPipe(1,adresaVysilac);
  // začátek příjmu dat
  nRF.startListening();
}

void loop() {
  // proměnné pro příjem a odezvu
  int prijem;
  unsigned long odezva;

  // v případě, že nRF je připojené a detekuje
  // přichází data, začni s příjmem dat
  if( nRF.available()){
    // čekání na příjem dat
    while (nRF.available()) {
      // v případě příjmu dat se provede zápis
      // do proměnné příjem
      nRF.read( &prijem, sizeof(prijem) );
    }
    // vytisknutí přijatých dat na sériovou linku
    Serial.print("Prijata volba: ");
    Serial.print(prijem);
    // dekodování přijatých dat
    switch( prijem ) {
      // pro známou hodnotu dat (1,2,3)
      // se odešle odezva:
      case 1:
        // v případě 1 odešli počet milisekund
        // od připojení napájení
        odezva = millis();
        break;
      case 2:
        // v případě 2 počet sekund
        // od připojení napájení
        odezva = millis()/1000;
        break;
      case 3:
        // v případě 3 počet mikrosekund
        // od připojení napájení
        odezva = micros();
        break;
      // v případě ostatních dat bude odezva 0
      default:
        odezva = 0;
        break;
    }
    // ukončení příjmu dat
    nRF.stopListening();
    // odeslání odezvy
    nRF.write( &odezva, sizeof(odezva) );
    // přepnutí do příjmu dat pro další komunikaci
    nRF.startListening();
    // vytištění odezvy po sériové lince
    Serial.print(", odezva: ");

```

```
Serial.println(odezva);
}
}
```

```
00101
01001
00001
```

## 5. UKÁZKA PROGRAMU – VYSÍLACÍ MODUL

Pro správnou funkci programu nainstalujte do vývojového prostředí Arduino knihovnu [RF24](#). Ukázka je převzata z <http://navody.arduino-shop.cz/navody-k-produktum/arduino-wifi-modul-nrf24l01.html>.

```
// nRF24L01 vysílač

// připojení knihoven
#include <SPI.h>
#include "RF24.h"
// nastavení propojovacích pinů
#define CE 7
#define CS 8
// inicializace nRF s piny CE a CS
RF24 nRF(CE, CS);
// nastavení adres pro přijímač a vysílač,
// musí být nastaveny stejně v obou programech!
byte adresaPrijimac[] = "prijimac00";
byte adresaVysilac[] = "vysilac00";

void setup() {
  // komunikace přes sériovou linku rychlostí 9600 baud
  Serial.begin(9600);
  // zapnutí komunikace nRF modulu
  nRF.begin();
  // nastavení výkonu nRF modulu,
  // možnosti jsou RF24_PA_MIN, RF24_PA_LOW, RF24_PA_HIGH and RF24_PA_MAX,
  // pro HIGH a MAX je nutný externí 3,3V zdroj
  nRF.setPALevel(RF24_PA_LOW);
  // nastavení zapisovacího a čtecího kanálu
  nRF.openWritingPipe(adresaVysilac);
  nRF.openReadingPipe(1, adresaPrijimac);
  // začátek příjmu dat
  nRF.startListening();
}

void loop() {
  // for smyčka pro postupné odeslání
  // hodnot 0 až 3 pro načtení všech dat
  // z přijímače
  for (int i=0; i < 4; i++) {
    // ukončení příjmu dat
    nRF.stopListening();
    // vytisknutí aktuální volby po sériové lince
    Serial.print("Posílám volbu ");
    Serial.println(i);
    // uložení startovního času komunikace
    unsigned long casZacatek = micros();
    // odeslání aktuální volby, v případě selhání
    // vytištění chybové hlášky na sériovou linku
    if (!nRF.write( &i, sizeof(i) )){
      Serial.println("Chyba při odeslání!");
    }
    // přepnutí do příjmu dat pro další komunikaci
    nRF.startListening();
    // uložení času začátku čekání
    unsigned long casCekaniOdezvy = micros();
    // proměnná s uložením stavu čekání na odezvu
    // od přijímače - "timeout"
    boolean timeout = false;
    // čekací while smyčka na odezvu od přijímače
    while ( ! nRF.available() ){
      // pokud do 200 000 mikrosekund nepříjde odezva,
      // ukončí čekání a nastav timeout

```

```

    if (micros() - casCekaniOdezvy > 200000 ){
        timeout = true;
        break;
    }
}
// kontrola stavu timeoutu
if ( timeout ){
    // v případě vypršení čekací smyčky,
    // vytiskni informaci o chybě spojení
    Serial.println("Chyba při prijmu, vyprseni casu na odezvu!");
}
// v opačném případě ulož přijatou zprávu a vypiš ji po sériové lince
else{
    // proměnná pro uložení přijatých dat
    unsigned long prijataData;
    // příjem dat se zápisem do proměnné prijataData
    nRF.read( &prijataData, sizeof(prijataData) );
    // uložení času konce komunikace
    unsigned long casKonec = micros();
    // výpis dat z komunikace po sériové lince
    // včetně délky trvání spojení
    Serial.print("Odeslana volba: ");
    Serial.print(i);
    Serial.print(", prijata data: ");
    Serial.println(prijataData);
    Serial.print("Delka spojeni: ");
    Serial.print(casKonec - casZacatek);
    Serial.println(" mikrosekund.");
}
// pauza 1 sekunda
delay(1000);
}
}

```