

Modulul 13

Utilizare transceiver RF

ASW: nRF24L01 module

# Obiectivele modulului

Utilizarea nivelului ASW, comunicatie prin radio: nRF24L01

# Utilizarea modulului transceiver prin RF

Transceiver = Transmitter + Receiver.

E important de retinut faptul ca acest modul RF nu poate in acelasi timp receptiona si transmite date. Aceste operatii trebuie facute pe rand.

Pentru a începe utilizarea modulului de comunicatie prin radio, trebuie să răspundem la 3 întrebari:

* Ce este nRF24L01 ?
* Cum controlează microcontroller-ul dispozitivul hardware ?
* Cum pot utiliza dispozitivul hardware de comunicatie prin RF ?

Ce este nRF24L01 ?

Modulul NRF24L01 reprezintă un modul de comunicație wireless, în banda 2.4GHz ISM(Industrial, Scientific and Medical). Modulul este low-power, tensiunea de alimentare fiind între 1.9 și 3.6 Volți, iar curentul în timpul transmisiei și al recepției nu depășește 14mA.

Viteza de transmisie maximă posibilă este de 2 Mbps. Dacă se dorește extinderea distanței de transmisie se poate alege o viteză mai mica (1 Mbps). Lungimea pachetului de date este de 32 de octeți, și adresa pe 5 octeți. Un mare avantaj al acestor module constă în faptul că au implementat un cod corector de erori, pe 8 biți, făcând comunicația sigură.

Distanța maximă testată la care s-a reușit comunicația fără a apărea întârzieri majore în recepția mesajelor este în aer liber de 100 de metri.

Cum controlează microcontroller-ul dispozitivul hardware ?

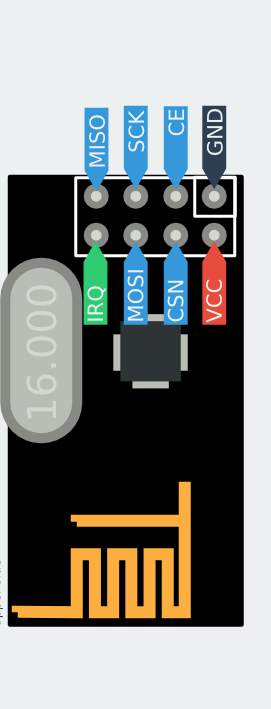
Microcontroller-ul (uC-ul) se foloseste de protocolul de comunicatie numit SPI (Serial Peripheral Interface) pentru a controla modulul RF. Astfel, uC-ul transmite comenzi si date modulului radio si primeste inapoi de la acesta informatii de stare si datele receptionate (daca exista).

Protocolul de comunicatie SPI foloseste 4 pini:

* SCKL (Serial Clock)
* CSN (Chip Select Negative, adica semnalul este activ pe ’0’ logic)
* MISO (Master Input Slave Output)
* MOSI (Master Output Slave Input)

Pentru utilizarea modulului RF mai sunt necesari 2 pini de alimentare: Vcc si GND si 2 pini de control:

* pinul numit CE – Chip Enable, pentru a activa anumite functionalitati ale modulului RF.
* pinului IRQ – Interrupt Request, i se atribuie valoarea logica ’0’ de catre modulul RF atunci cand acesta a terminat de transmis sau cand a primit date.



Cum pot utiliza dispozitivul hardware de comunicatie prin RF ?

Se pun la dispozitie 5 functii pentru utilizarea modului de comunicatie prin RF. Dependintele modulului: fisierul hal\_nrf.h

|  |  |  |  |
| --- | --- | --- | --- |
| **COM\_vStartListening** | | | |
| Funcția pune modulul RF in starea de receptie. | | | |
| **Parametri** | | | |
| **Nume** | **Tip** | **Descriere** | |
| - | - | - | |
| **Valoare returnată** | | | |
| **Tip** | **Descriere** | | |
| void | - | | |
| **Reentrantă** | | | Nu |

|  |  |  |  |
| --- | --- | --- | --- |
| **COM\_vSendMessage** | | | |
| Funcţia furnizeaza data de transmis si pune modulul RF in starea de transmisie. | | | |
| **Parametri** | | | |
| **Nume** | **Tip** | **Descriere** | |
| u8Message | T\_U8 | Data(mesajul) de transmis. | |
| **Valoare returnată** | | | |
| **Tip** | **Descriere** | | |
| void | - | | |
| **Reentrantă** | | | Nu |

|  |  |  |  |
| --- | --- | --- | --- |
| **COM\_vCheckIRQ** | | | |
| Funcția verifica daca s-a primit semnal activ pe pinul IRQ.  Adica, daca s-au primit date SAU s-au transmis datele. | | | |
| **Parametri** | | | |
| **Nume** | **Tip** | **Descriere** | |
| - | - | - | |
| **Valoare returnată** | | | |
| **Tip** | **Descriere** | | |
| void | - | | |
| **Reentrantă** | | | Nu |

|  |  |  |  |
| --- | --- | --- | --- |
| **COM\_vProcessFIFO** | | | |
| Funcţia verifica daca s-au primit date noi. Daca da, atunci proceseaza fiecare mesaj primit, folosind functia descrisa mai jos: COM\_vProcessMessage. | | | |
| **Parametri** | | | |
| **Nume** | **Tip** | **Descriere** | |
| - | - | - | |
|  | | | |
| **Tip** | **Descriere** | | |
| void | - | | |
| **Reentrantă** | | | Nu |

|  |  |  |  |
| --- | --- | --- | --- |
| **COM\_vProcessMessage** | | | |
| In functie de mesajul primit, aici se vor lua actiunile necesare aplicatiei. | | | |
| **Parametri** | | | |
| **Nume** | **Tip** | **Descriere** | |
| u8Message | T\_U8 | Data(mesajul) de procesat. | |
| **Valoare returnată** | | | |
| **Tip** | **Descriere** | | |
| void | - | | |
| **Reentrantă** | | | Nu |

Functiile *COM\_vStartListening*,*COM\_vCheckIRQ*, *COM\_vProcessFIFO* vor fi apelate una dupa cealalta (in aceasta ordine) intr-un task.

Cu cat task-ul se apeleaza mai des, cu atat uC-ul va fi mai rapid instiintat de primirea vreunui mesaj si va actiona la fel de repede.

A se observa ca aceasta ordine are urmatoarea logica:

1. Modulul RF intra in starea de receptie.
2. uC-ul verifica daca modulul RF a primit vreun mesaj. Daca da, il va prelua de la modulul RF si-l va salva intr-o coada cu mesaje (in memoria uC-ului).
3. In cazul in care in coada cu mesaje exista un mesaj nou, uC-ul il va procesa(consuma) imediat.

(a consuma = a procesa + a sterge dupa procesare)

Pentru transmisie, se va folosi functia *COM\_vSendMessage.*

A se retine ca pentru o functionare corespunzatoare, este indicat ca modulul RF sa fie mereu in starea de receptie. Va iesi din aceasta stare doar cand va avea ceva de transmis. Apoi va reveni in starea de receptie.