Probleme courant NRF24L01

Problèmes courants

Paramètres qui doivent correspondre

Avant de signaler un comportement indésirable, assurez-vous que le les configurations RF24 suivantes correspondent à la fois à la réception et à la transmission Émetteurs-récepteurs nRF24L01 :

1. - `RF24::setAddressWidth()`
2. - `RF24::setChannel()`
3. - `RF24::setDataRate()`
4. - `RF24::setAutoAck()`
5. - `RF24::enableDynamicPayloads()` ou `RF24::disableDynamicPayloads()`
6. - `RF24::enableAckPayload()` ou `RF24::disableAckPayload()` (nécessite auto-ack etfonctionnalités de charges utiles dynamiques)
7. - `RF24::setPayloadSize()` (uniquement i la fonctionnalité de charges utiles dynamiques est désactivée – elle est désactivée par défaut)
8. - `RF24::setCRCLength()` ou `RF24::disableCRC()` (la fonction de confirmation automatique active automatiquement le CRC car il est obligatoire)

En outre, il est utile de considérer une adresse comme un chemin (un itinéraire communément partagé) au lieu d'une destination d'appareil d'identification. Cela signifie que les adresses doivent correspondre pour qu'une charge utile se déplace d'un émetteur-récepteur à un autre.

Cependant, les numéros de tuyau attribués avec l'adresse correspondante n'ont pas correspondre. Vous pouvez considérer les tuyaux comme des places de stationnement pour les paquets, tandis que toutes les charges utiles des paquets vivent dans un tampon FIFO TX ou RX. Rappelez-vous que le Les tampons TX FIFO et les tampons RX FIFO ont tous deux une occupation maximale de 3 charges utiles (indépendamment de la taille maximale de la charge utile de 32 octets).

Utilisation interdite de write*() dans les rappels de la routine de service d'interruption

Étant donné que la bibliothèque RF24 utilise `millis()` pour implémenter un délai d'attente et `delay()` pour les temps d'attente obligatoires, les fonctions suivantes ne peuvent pas être utilisées dans une méthode de rappel ISR :

1. - `RF24::write()`
2. - `RF24::writeBlocking()`
3. - `RF24::writeFast()`
4. - `RF24::startWrite()`
5. - `RF24::txStandBy()` & `RF24::txStandBy(uint32_t, bool)`
6. - `RF24::powerUp()`
7. - `RF24::startListening()`
8. - `RF24::stopListening()`

@voir la note dans la documentation pour `RF24::available()`.

Plus d'informations sur les raisons pour lesquelles vous ne pouvez pas appeler `millis()` (ou `delay()`) à partir d'une fonction de rappel ISR sont disponibles sur [the Arduino docs](https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/&ved=2ahUKEwjMhtSRl5jzAhVUsp4KHWIPCrIQFnoECAoQAQ&usg=AOvVaw1X9H0058Nz7Hck91VIC3bD).

Voici les problèmes les plus courants et leurs solutions

write() renvoie toujours true après setAutoAck(false)

Ne désactivez pas la fonction de confirmation automatique. RF24::write() n'a aucune raison de douter que la charge utile a été livrée si la fonction de confirmation automatique est désactivée. Nous vous recommandons de lire les docs sur RF24::setAutoAck() avant de désactiver le fonction de confirmation automatique.

write() renvoie false lorsque la charge utile a été reçue

Si les paramètres correspondent sur les deux émetteurs-récepteurs d'extrémité, cela ne peut que signifie que le récepteur nRF24L01 n'a pas réussi à envoyer un accusé de réception (ACK) paquet vers le nRF24L01 émetteur. Cela est généralement dû à instabilité (bruit électrique) dans les lignes électriques (VCC et GND) allant à le récepteur nRF24L01.

Si vous ne recevez pas les paquets ACK correctement/de manière fiable sur les débits de données inférieur à 2MBPS, essayez d'ajouter un gros condensateur près du module/puce. Exemples de problèmes : [#264](https://github.com/nRF24/RF24/issues/264) [#211](https://github.com/nRF24/RF24/issues/211).

Pour plus de fiabilité, veuillez utiliser des condensateurs électrolytiques ou au tantale. Céramique les condensateurs peuvent ne pas être assez bons (selon la source de fabrication).

Les charges utiles reçues/envoyées sont inexactes ou printDetails() génère la ou les valeurs inattendues

Cela est probablement dû au fait que la vitesse SPI est amplifiée jusqu'à 10 MHz par défaut. Nous recommandons:

1. Assurez-vous que les fils ne sont pas desserrés et essayez d'éviter d'utiliser de longs fils. 2. Si le point précédent ne vous aide pas, essayez de réduire la vitesse SPI comme ceci

```cpp
radio RF24 (7, 8, 4000000); // définit la vitesse SPI sur 4 MHz au lieu de 10 MHz par défaut
```

Au début de la bibliothèque RF24, la valeur par défaut était (avant 2014) définie sur 4 MHz.

Mon module PA/LNA ne parvient pas à transmettre

Vous pouvez trouver des variantes de l'émetteur-récepteur nRF24L01 commercialisées sous le nom de “nRF24L01+PA+LNA”. Ces modules se distinguent par le fait qu'ils sont livrés avec une antenne détachable (de type SMA). Ils utilisent un circuit intégré RFX24C01 séparé avec l'antenne pour des fonctionnalités améliorées d'amplification de puissance (PA) et d'amplification à faible bruit (LNA). Bien qu'ils offrent une plus grande portée avec les mêmes fonctionnalités, ils sont sujets à quelques inconvénients moins connus (et moins annoncés):

1. Source d'alimentation plus puissante. Vous trouverez ci-dessous un tableau des exigences de courant annoncées que de nombreux régulateurs 3V de cartes MCU peuvent ne pas être en mesure de fournir (après avoir alimenté les composants internes).

```
| Spécification | Valeur |
|:-------------:|:-----:|
| Courant de mode d'émission (crête) | 115mA |
| Courant de mode de réception (crête) | 45 mA |
| Courant en mode hors tension | 4,2 µA |
```

2. Nécessite une protection contre les interférences électromagnétiques. Le blindage fonctionne généralement mieux lorsqu'il a un chemin vers la terre (broche GND), mais cette connexion à la broche GND n'est pas nécessaire. Il est important que le blindage ne touche aucune pièce conductrice de courant.

1. Les professionnels ont tendance à utiliser une cage/maille de Faraday pour mettre en place un blindage électromagnétique, mais cela peut être coûteux pour ce scénario.
2. Une solution rapide à faire soi-même (comme preuve de concept) consisterait à envelopper le module PA/LNA avec du ruban électrique, puis à enrouler du papier d'aluminium autour du ruban électrique (pour le blindage) tout en faisant très attention à ne pas laisser le papier d'aluminium touchez toutes les pièces conductrices de courant (comme les broches GPIO, le support d'antenne et les joints de soldat pour le support d'antenne). Observer

[![ghetto_shielding_1.png](https://github.com/nRF24/RF24/blob/master/images/ghetto_sheilding_1.png)](https://github.com/nRF24/RF24/blob/master/images/ ghetto_sheilding_1.png)

```
[![ghetto_shielding_2.png](https://github.com/nRF24/RF24/blob/master/images/
ghetto_sheilding_2.png)](https://github.com/nRF24/RF24/blob/master/images/
ghetto_sheilding_2.png)
```

Mon module PA/LNA ne fonctionne pas aussi bien que je l'espérais ou la radio NRF fonctionne mieux en le touchant

Comme décrit ci-dessus, les modules radio (notamment les versions PA+LNA) dépendent d'une source d'alimentation stable. Bien que ces modules puissent fonctionner avec une mauvaise alimentation, ils perdent souvent des paquets ou ne parviennent pas à en recevoir autant qu'un module avec une meilleure source d'alimentation. De plus, cela peut parfois être vu de manière étrange, comme le module radio qui fonctionne mieux au toucher. Là encore, il s'agit probablement d'un problème de stabilité de l'alimentation car il manque un condensateur au module radio (une dépense souvent négligée de la part du fabricant du module).

Ajoutez des condensateurs à proximité des broches VCC et GND de la radio. En règle générale, 10 uF suffisent. En fonction de la disposition de votre circuit, des différences dans les propriétés électrolytiques des condensateurs peuvent être observées, de sorte qu'un condensateur à faible ESR

(Equivalent Series Resistance) est souhaitable.

From:
http://chanterie37.fr/fablab37110/ - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:
http://chanterie37.fr/fablab37110/doku.php?id=start:arduino:nrf24l01:probleme_courant_nrf24l01

Last update: **2023/01/27 16:08**