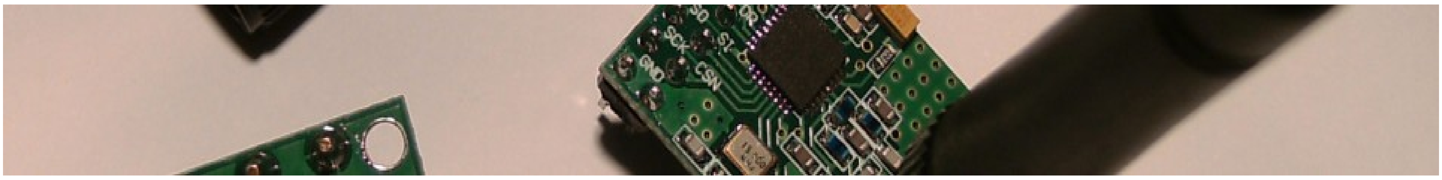


LCD Image Viewer (https://blog.zakkemble.net/lcd-image-viewer/)

Getting Bluetooth modules talking to each other (https://blog.zakkemble.net/getting-bluetooth-modules-talking-to-each-other/)

Jan 01 2013

nRF905 Radio Library for AVR and Arduino

- By Zak Kemble (https://blog.zakkemble.net/author/Zak/) in Arduino (https://blog.zakkemble.net/category/arduino/), AVR (https://blog.zakkemble.net/category/microcontroller/avr/), Libraries (https://blog.zakkemble.net/category/library/), Microcontroller (https://blog.zakkemble.net/category/microcontroller/), Radio (https://blog.zakkemble.net/category/radio/)

The nRF905 is a radio transceiver IC similar to the well known nRF24L01, but operates at 433/898/915MHz instead of 2.4GHz, has a much longer range and a few extra IO pins. However, the nRF905 data rate is only 50Kbps compared to nRF24L01's 2Mbps.

This library offers quite a bit of flexibility: Optional use of interrupts, 2 of the connections to the module are optional since their states can also be accessed by the ICs status register, and supports basic collision avoidance.

NOTE: v3.0.0 of the library was released on 12th September 2017, the default CD pin has changed and the AM pin is now used by the library.

Download:
Arduino (https://github.com/zkemble/nRF905-arduino) – Documentation (http://zkemble.github.io/nRF905-arduino/) (or use the Arduino library manager, search for "nrf905")
AVR (non-Arduino) (https://github.com/zkemble/nRF905) – Documentation (http://zkemble.github.io/nRF905/)

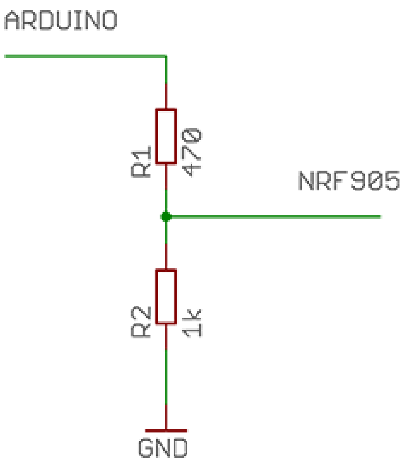
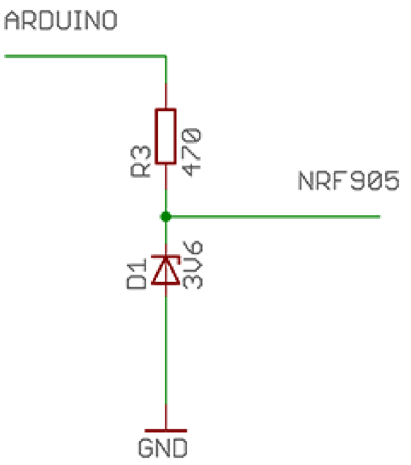
nRF905	ATmega48/88/168/328	Arduino Uno	Description
VCC	3.3V	3.3V	**Power** (3.3V)
CE	D7 (13)	7	**Standby** – High = TX/RX mode, Low = standby
TXE	B1 (15)	9	**TX or RX mode** – High = TX, Low = RX
PWR	B0 (14)	8	**Power-up** – High = on, Low = off
CD	D4 (6)	4	**Carrier detect** – High when a signal is detected, for collision avoidance
AM	D2 (4)	2	**Address Match** – High when receiving a packet that has the same address as the one set for this device, optional since state is stored in register, if interrupts are used (default) then this pin must be connected
DR	D3 (5)	3	**Data Ready** – High when finished transmitting/High when new data received, optional since state is stored in register, if interrupts are used (default) then this pin must be connected
SO	B4 (18)	12	**SPI MISO** (Mega pin 50)

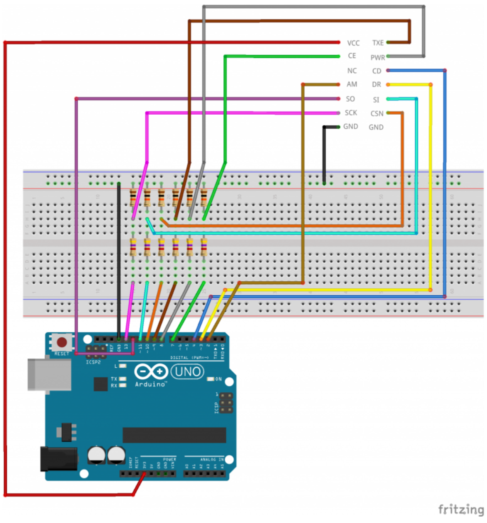
SI	B3 (17)	11	**SPI MOSI** (Mega pin 51)
SCK	B5 (19)	13	**SPI SCK** (Mega pin 52)
CSN	B2 (16)	10	**SPI SS**
GND	GND	GND	**Ground**

Some of the module pin names differ from the IC pin names in the datasheet:

Module	**IC**
CE	TRX_EN
TXE	TX_EN

The nRF905 is not 5V compatible, so some level conversions will need to be done with the Arduino outputs. A simple voltage divider or resistor and zener diode will do the trick. Only TXE, CE, PWR, SI, SCK and CSN pins need level conversion (not CD, AM, DR and SO).

Divider	**Zener**
 (https://blog.zakkemble.net/wp-content/uploads/2020/11/res_div.png)	 (https://blog.zakkemble.net/wp-content/uploads/2013/01/zener-1.png)



(http://blog.zakkemble.co.uk/wp-content/uploads/2013/01/nrf905_bb-1.png)

The nRF905 has 511 channels ranging 422.4MHz – 473.5MHz in 100KHz steps on the 433MHz band and 844.8MHz – 947MHz in 200KHz steps on the 868/915MHz band (remember to check which frequencies are legal in your country!), but each channel overlaps adjacent channels so there are only a total of 170 usable channels at once.

Searching for nRF905, PTR8000 and PTR8000+ should yield some results for modules on AliExpress, Ebay and DealExtreme. You should be able to get 2 for around £10.

Callbacks

Since v3.0.0 this library uses callbacks which are ran when events occur. If the option to use interrupts is enabled then the callbacks will run from the interrupt routine so be sure that any global variables you use in them are declared 'volatile', just as you would when dealing with normal ISRs. These events will wake the microcontroller if it is sleeping.

Event	Callback	Notes
New packet incoming	NRF905_CB_ADDRMATCH	
Valid packet received	NRF905_CB_RXCOMPLETE	
Invalid packet received	NRF905_CB_RXINVALID	
Packet transmission complete	NRF905_CB_TXCOMPLETE	This only works if the nextMode is NRF905_NEXTMODE_STANDBY when calling nRF905_TX()

Nitty-gritty radio stuff

The actual air data-rate of nRF905 is 100Kbps, but the data is Manchester encoded which halves the throughput to 50Kbps. The modulation is GFSK with ±50KHz deviation. The radio also adds a few extra bits of data to the address and payload; a preamble and CRC.

Transmitted packet

Preamble	Address	Payload	CRC
10 bits	**1 or 4 bytes**	**1 – 32 bytes**	**0 – 2 bytes**

The address is also used as the syncword and should have as many level shifts as possible. 10100110 01100101 00011010 11011010 (2791643866) would be a good address to use, but 00000000 00000000 00000000 0010000 (32) would not be.

The CRC is used to detect errors in the received data. Having a CRC doesn't eliminate all bad packets, there is always a small chance of a bad packet passing the CRC. Using larger CRCs helps to reduce that chance.
When the CRC is set to 16 bit the radio uses the CRC16-CCITT-FALSE (0xFFFF) algorithm. I'm not sure what it uses for 8 bit CRCs.

Transmission time

The sizes of the address, payload and CRC can be adjusted for a balance between throughput and latency.

Example configurations

Address size	Payload size	CRC size	=	Throughput (bps)	Latency (ms)
4	32	2	=	36940	6.93
1	4	1	=	17680	1.81
1	1	0	=	6838	1.17

Throughput is how much useful data can be sent (the payload part), assuming no delays for writing and reading the payload.
Latency is how long it takes for the transmission to complete.
Switching from standby mode to receive or transmit mode takes a maximum of 650us and switching between receive and transmit modes takes a maximum of 550us.

Transmission time can be worked out with:

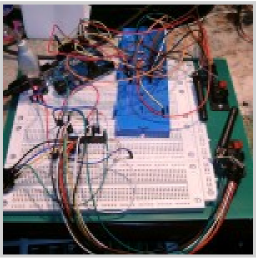
$$t = t_{startup} + t_{preamble} + ((N_{address} + N_{payload} + N_{CRC}) / BR)$$

$t_{startup}$ is the time to switch mode as stated above (650us / 550us).
$t_{preamble}$ is 200us for the 10 bit preamble.
$N_{address}$, $N_{payload}$ and N_{CRC} are the address, payload and CRC sizes in bits.
BR is the bit rate which is 50,000.

For config 1, switching from standby to transmit:

$$t = 0.00065 + 0.0002 + ((32 + 256 + 16) / 50000)$$
$$t = 0.00693 \text{ seconds (6.93ms)}$$

A note to developers: Before v3.0.0 this library would load the address bytes in reverse order, keep that in mind if you're using an old version of the library to interface with other radio systems!



(https://blog.zakkemble.net/wp-content/uploads/2012/12/PICT1443.jpg)

(https://blog.zakkemble.net/wp-content/uploads/2012/12/modules.jpg)

610 comments

3 pings

Skip to comment form

1.
   - Chris on July 28, 2019
     # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-208980)
     Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=208980#respond)

   Hello Zak,
   First of all thank you for this library. Great work!
   The communication of my modules works flawlessly. I have also connected 2 Sevomotors to the Mega (PIN 40 and 42 with separate power supply) which permanently ticks when the communication starts. Do you have any idea why this could be?
   Many thanks and best regards,
   Chris

   1.
      - Zak Kemble () on July 28, 2019
        Author
        # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-208982)
      - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=208982#respond)

      Hey Chris, thanks! You'll have to post your code before I can see what's up! -> https://pastebin.com/ (https://pastebin.com/)

2.
   - Tim on September 27, 2019
     # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209049)
     Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209049#respond)

   Hi Zak, I've built a few of these links running at 4800 baud to handle 30 bytes of data every 250mS running at 433MHz carrier.

   However on one of the links it runs ok for about 20 seconds then hangs no tx or rx activity. This problem can happen at either end and I suspect it may actually be nRf905 module related.

   I've tried disabling IRQ and other variants but always the same result.

   Pressing the reset button on the tx Arduino fixes the problem. Maybe there's an overflow problem somewhere ?

   Any pointers on how to debug ?

   I'll try slowing down the refresh rate from 250mS to 1S and see what happens

3.
   - Tim on September 27, 2019
   - # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209050)
   - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209050#respond)

   Hi Zak, me again. Just reduced the update rate to 1 second and still the same problem. However looking at the MOSI pin on the nRF905 when the sending end stops transmitting there is not the normal activity on this pin – looks as though some is corruption.
   Then it'll start up again.

   Time for a beer – will look at this again tomorrow as I'm well down a 'rabbit hole' here 🙂

4.
   - Tim on September 28, 2019
   - # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209051)
   - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209051#respond)

   Hi Zak,

   further to my last e mail about a problem I was having – I've managed to fix it.
   I'm running the wireless UART example and found that it would hang in the rx loop and not detect incoming traffic unless the device was reset.
   This problem appears with both the Arduino and AVR code on 2 out of 5 links.
   I fixed it by putting a watch timer set to 1 second which gets reset on transmission or reception of data.
   Thus when the rx stops getting traffic the watch dog kicks in and restarts the link.
   Bit of a bodge as I'm not sure how to get to the root cause.
   Before this fix when the rx hung the carrier detect line didn't work – so suspect it could be the nRf905 losing it settings and needing to be re-initialized ?

   Regards Tim

   1.
      - Zak Kemble () on September 28, 2019
        Author
        # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209052)
      - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209052#respond)

      Hey Tim, the wireless link example isn't really very reliable, but it shouldn't cause things to hang. Things will get even more wack if you've got 5 modules with the wireless link example all talking to each other, since it's only meant for 2 modules! I don't think the modules are loosing their settings, what state are the TXE, DR and CE pins in when it hangs?

      1.
         - Tim on September 28, 2019
           # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209053)
         - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209053#respond)

         Hi Zak,

         thanks for the reply – just to clarify I'm only using them in pairs but have seen this problem on a couple of pairs and the problem moves with a particular unit.
         Each pair is far enough away not to cause mutual interference and operate on a clear frequency in the 70cm 433MHz band for which I'm licensed.
         By wrapping some extra code around yours combined with the watch dog timer it's now fairly robust.
         I'll check the TXE, DR and CE lines when it hangs and report back.

         Best regards Tim

5.
   - Tim on September 29, 2019
   - # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209054)
   - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209054#respond)

   Hi Zak,

   when the rx hangs all the lines are as they should be – a bit more testing seems to indicate the rx crash is possibly being caused by something on the tx side.

   Maybe irq collisions between UART and SPI ?

I'm sending a total of 30 characters including a string terminator and I suspect this might be the root cause as this is the maximum allowed #define MAX_PACKET_SIZE (NRF905_MAX_PAYLOAD – 2).

Wonder why maxpayload is -2 ?

As I'm always sending a fixed length message I'm wondering if there's a better way to assemble the message in the nano ready for the nRF905 to process it ?

Currently I'm sending the message character by character at 4800 – just for fun I'll try 9600.

Regards Tim

6.
- Tim on September 29, 2019
- # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209055)
- Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209055#respond)

Hi Zak,

still plugging away at this and learning / discovering as I go along. As I suspected sometimes the tx gets corrupted and either sends jibberish or only part of the packet – maybe roughly once out of a 100 packets.
I see you mention in your code about needing a 700uS delay going from tx to rx to avoid this so I set it to 1mS then 2mS and seems to have helped but not perfect.
I also tried to remove your 'ack code' – seemed pretty obvious how it worked but strangely it shortened the tx data packet.
Time for another break and reflection !
I'm thinking it maybe the nRF905 chip – perhaps a counterfeit device ?
I have another batch of nRF905's coming so will see if that helps.

Regards Tim

7.
- Tim on September 30, 2019
- # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209057)
- Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209057#respond)

Ok, I'm 99% certain it's a dodgy chip problem as my code works fine on 4 out of 5 links and never misses a beat.

On the plus side it's caused me to refine the protocol and generally harden the code to cope with drop outs and missed data.

I'm going to install the next batch of nRF905 modules in sockets for easy test and replace meant !

Thanks for a great piece of code and listening to my trials and tribulations !

8.
- saniul on October 2, 2019
- # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209059)
- Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209059#respond)

Hi Zak,
I need your help. I need to transmit a constant data through the nrf905 transceiver module at 915mhz band. I have a arduino uno and a nano. can you please tell me how to set up the whole thing and code for transmitting a constant data. I need these for my final year project. Another thing, Can it jam a mobile signal which operates under 915mhz frequency band? I know it's illegal to make a jammer but I am working on this project which needs to jam a signal, Which I would use with my bomb disposing robot.

1.
   - Zak Kemble () on October 2, 2019
     Author
     # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209060)
   - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209060#respond)

   lol I'm not going to do your school work for you.

9.
- Geoffrey Brown on December 4, 2019
- # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209107)
- Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209107#respond)

There are a number of NRF905 + STC15F204 combinations available from various places:

https://www.aliexpress.com/item/32931569323.html?spm=a2g0o.productlist.0.0.16195be05Nt8Xe&algo_pvid=ba66008d-0bfd-4ec7-ab0b-c23d5750a36b&algo_expid=ba66008d-0bfd-4ec7-ab0b-c23d5750a36b-25&btsid=7bee88fa-c4ed-4600-84b8-b34c31622546&ws_ab_test=searchweb0_0,searchweb201602_2,searchweb201603_55 (https://www.aliexpress.com/item/32931569323.html?spm=a2g0o.productlist.0.0.16195be05Nt8Xe&algo_pvid=ba66008d-0bfd-4ec7-ab0b-c23d5750a36b&algo_expid=ba66008d-0bfd-4ec7-ab0b-c23d5750a36b-25&btsid=7bee88fa-c4ed-4600-84b8-b34c31622546&ws_ab_test=searchweb0_0,searchweb201602_2,searchweb201603_55)

The STC module appears to have defined pins for certain functions (presumably for TxE, DR, CD etc).

https://1drv.ms/u/s!Am_AnSJTVHLYmo8ib-sQB_OszURpew?e=yHpupr (https://1drv.ms/u/s!Am_AnSJTVHLYmo8ib-sQB_OszURpew?e=yHpupr)

It appears that the device uses a serial interface perhaps for a lot of functions and the pin assignments indicate to me that there is no SPI interface as on a mega these use D50-D52. The fact that it references P34 and P55 indicates this is intended for connection to a Arduino Mega only as other processors, to my knowledge, don't have these pins. I think the DR and AM pins need to go to external interrupt lines so the fact that none of the labelled ports, to my knowledge.

I suspect it was perhaps produced for a company for a specific need hence the reason that there is little available on it on the internet.

Anyway, if you can shed any light on this it would be much appreciated.

Geoffrey Brown, Papamoa, NZ.

1.
   - Zak Kemble () on December 10, 2019
     Author
     # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209114)
   - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209114#respond)

   Hi Geoffrey, those STC15F204/NRF905 modules are plain serial-RF bridges, and are not compatible with this library (it runs its own code on the STC15F204). Those extra pin holes appear to be extra GPIOs from the STC15F204 microcontroller rather than connections to the nRF905 module. I don't think they have any special protocol, just send it serial data at the correct baud rate and out pops the data wherever other modules are that are in range.

10.
   - Andrei on December 9, 2019
     # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209110)
     Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209110#respond)

Is it possible to change pins 2,3,4 and 7 to analog pins? As far as I know, analog pins can be used as digital, but not the ones with pwm.

Thanks!

1.
   - Zak Kemble () on December 10, 2019
     Author
     # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209112)
   - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209112#respond)

   Hey Andrei! Pins 4 and 7 can be moved around anywhere (change things in nRF905_config.h), but 2 and 3 must be interrupt pins unless you disable interrupts (NRF905_INTERRUPTS 0).

11.
   - Geoffrey Brown on January 25, 2020
     # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209155)
     Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209155#respond)

I believe I have uncovered a little issue with this library, in particular, in regards the nRF905_TX routine.

I, like some others have mentioned, have had a situation whereby the communications have come to an end and the Arduino Mega, in my case, needed to be restarted.

The issue I believe is as follows:

```
uint8_t nRF905_TX(uint32_t sendTo, void* data, uint8_t len, nRF905_nextmode_t nextMode)
{
// TODO check DR is low?

NRF905_NO_INTERRUPT()
{
if(nRF905_airwayBusy())
return 0;

setAddress(sendTo, NRF905_CMD_W_TX_ADDRESS);

// Load new payload
if(data != NULL)
{
NRF905_ATOMIC()
{
CHIPSELECT()
{
spi_transfer_nr(NRF905_CMD_W_TX_PAYLOAD);
for(uint8_t i=0;i<len;i++)
spi_transfer_nr(((uint8_t*)data)[i]);
}
}
}

if(!POWERED_UP())
{
STANDBY_ENTER();
POWER_UP();
delay_ms(3);
}

if(nRF905_airwayBusy()) //PROBLEM LOCATION ACCORDING TO GEOFFREY
return 0;

// Put into transmit mode
MODE_TX();

// Pulse standby pin to start transmission
STANDBY_LEAVE();

if(nextMode == NRF905_NEXTMODE_RX)
{
// The datasheets says that the radio can switch straight to RX mode after
// a transmission is complete by clearing TX_EN while transmitting, but
// if this is done within ~700us the transmission seems to get corrupt.
delay_us(700);
MODE_RX();
}
else if(nextMode == NRF905_NEXTMODE_STANDBY)
{
delay_us(14);
STANDBY_ENTER();
}
// else NRF905_NEXTMODE_TX
}

return 1;
}
```

What happens is that the CHIPSELECT() part that is inside the NRF905_ATOMIC section does two things as seen in its associated define (if I have things right):

```
#define CHIPSELECT(standby) standby; for(uint8_t _cs = cselect(); _cs; _cs = cdeselect())
```

```
#define STANDBY_LEAVE() (digitalWrite(NRF905_TRX_EN, HIGH))
#define STANDBY_ENTER() (digitalWrite(NRF905_TRX_EN, LOW))
#define STANDBY (STANDBY_ENTER())
```

It enters standby mode (CE line goes low) as the first step. Note that in Zac's code NRF905_TRX_EN is synonymous with CE (chip enable). The second part (in the for loop) toggles the CSN pin.
So if we look at the first part of the code we have:

```
if(data != NULL)
{
NRF905_ATOMIC()
{
CHIPSELECT()
{
spi_transfer_nr(NRF905_CMD_W_TX_PAYLOAD);
for(uint8_t i=0;i<len;i++)
spi_transfer_nr(((uint8_t*)data)[i]);
}
}
}
```

And NRF905_ATOMIC has the following define:

```
#define NRF905_ATOMIC() for(uint8_t _cs2=interrupt_off(); _cs2; _cs2=interrupt_on())
```

So this NRF905_ATOMIC part of the loop turns interrupts off then on again at the end of the loop. The CHIPSELECT inside this turns the NRF905 into standby mode (CE low) and pulses the CSN line low while SPI communications are occurring:

Now if COLLISION_AVOID is enabled (default) the second nrf905_airwayBusy check is made. This is marked as //PROBLEM LOCATION in code above and reproduced below:

```
…
if(nRF905_airwayBusy())
return 0;

// Put into transmit mode
MODE_TX();

// Pulse standby pin to start transmission
STANDBY_LEAVE();
```

The end result is that if the NRF905 is busy receiving a packet at this point in the code (nrf905_airwayBusy is true) the nRF905_TX routine causes the device to return with the NRF905 in a standby state (from the CHIPSELECT loop) and the STANDBY_LEAVE never occurs as this code is after the return 0 of the second nRF905_airwayBusy and so is never reached. The upshot is that the nRF905_TX routine returns having left the NRF905 in a standby mode and the packet transmit has not started. The device will not receive another packet even if the device was in receive mode before the nRF905_TX routine was called (refer nRF905_RX()). If the code abandons the transmit of the packet then we have the signals left in the following state according to my multimeter:

TXE 0V (read mode)
CE 0V (also TRX_EN in code; device in standby) – this is the problem
PWR 3.2V (device powered on)
AM 0V (no address matched because device in standby mode)
DR 0V (no data ready because nothing happening)

The interrupts are left enabled (from the NRF905_ATOMIC loop completing fine). The CSN line is also left in the correct state because the CHIPSELECT loop completes as well.

If the packet being sent was in response to a packet received (the last one) and if this issue occurs in nRF905_TX and the nRF905_TX is abandoned in the code for whatever reason then the nRF905_TX leaves the system in an mode where nothing will be received again until nRF905_RX() is called.

What is needed is one of three solutions:

1) Don't perform the second check of nRF905_airwayBusy because you already did it at start of nRF905_TX routine – my preferred answer. The second nRF905_airwayBusy and the return is what is causing this issue.

2) In the second nRF905_airwayBusy check do the following:

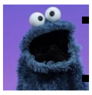
```
if(nRF905_airwayBusy()) {
STANDBY_LEAVE();
return 0;
}
```

3) After any failed nRF905_TX immediately call nRF905_RX() to restore an operational receive mode.

This will undo the effect of the CHIPSELECT having entered standby mode.

Thoughts…

1.

- 
  - Zak Kemble () on January 25, 2020
  - Author
  - # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209156)
- Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209156#respond)

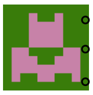
Hi Geoffrey, I get where you're coming from about the CHIPSELECT() standby stuff, however this is not the case. The first parameter of the `CHIPSELECT()` statement is blank, which means that "standby" in the #define is also blank, and does not enter standby mode. To enter standby mode it would have to be written as `CHIPSELECT(STANDBY)`, but this was only used in older verions of the library in a few places.
The only way nRF905_TX() might assert the standby pin is by setting nextMode to NRF905_NEXTMODE_STANDBY.

1.

-
  - Geoffrey Brown on March 2, 2020
  - # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209196)
- Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209196#respond)

Many thanks for that feedback. Very useful. Have some more learning on #defines to do…

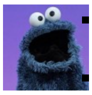
12.

- 
  - Michael on October 29, 2020
  - # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209459)
  - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209459#respond)

hi,
Brand new to all of this.
I wired as per the diagram and opened the supplied code in the IDE. Tried to upload but got errors with the include statement . copied all of the nRF905 folder to the libraries but no luck. Copied the individual files into the same folder as the sketch and it seems better. i now get:

```
In file included from nRF905.c:15:0:
/usr/share/arduino/libraries/SPI/SPI.h:36:1: error: unknown type name 'class'
class SPIClass {
^
/usr/share/arduino/libraries/SPI/SPI.h:36:16: error: expected '=', ',', ';', 'asm' or '__attribute__' before '{' token
class SPIClass {
^
```

1.

- 
  - Zak Kemble () on October 29, 2020
  - Author
  - # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209460)
- Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209460#respond)

You're using the files for non-Arduino stuff, only use the stuff in the arduino folder. Also, try using the new library here – https://github.com/zkemble/nRF905-arduino (https://github.com/zkemble/nRF905-arduino)

1.

-
  - Michael on October 29, 2020
  - # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209461)
- Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209461#respond)

I really appreciate the reply Zak. Have downloaded the files from Github however they seem to be only examples. I copied the nRF905-master/arduino/nRF905 library files to ~/Arduino/library/nRF905 and also to /usr/share/arduino/library/nRF905 however it did not much difference.

I then tried to run the lowpwr_note_baseStation and got the following error:

lowpwr_node_baseStation.ino:20:22: fatal error: LowPower.h: No such file or directory
compilation terminated.

Sorry for my lack of understanding.

1.
   - Zak Kemble () on October 30, 2020
     Author
     # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209462)
   - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209462#respond)

   I'm not sure what you mean by they're only examples, but to fix your error you also need this library https://github.com/rocketscream/Low-Power (https://github.com/rocketscream/Low-Power)

2.
   - Michael on October 30, 2020
     # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209463)
   - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209463#respond)

   Hi again Zak,
   Well, I have found one very silly thing I should have though about earlier – my preferences were pointing to the wrong directory!! No wonder it could not find the library.
   It seems to be compiling and uploading OK now. Now I just need to waiting on my second nRF905 arriving before I can test it.

13.
   - Andrei on January 3, 2021
     # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209523)
     Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209523#respond)

I'm trying to compile wireless serial example, but i get this errors…

```
 wirelessSerialLink:40:18: error: 'NRF905_ADDR_SIZE' was not declared in this scope
byte dstAddress[NRF905_ADDR_SIZE];
^~~~~~~~~~~~~~~~
C:\Users\Andrei\Google
Drive\Arduino\libraries\nRF905_Radio_Library\examples\wirelessSerialLink\wirelessSerialLink.ino:40:18:
note: suggested alternative: 'NRF905_ADDR_SIZE_RX'
byte dstAddress[NRF905_ADDR_SIZE];
^~~~~~~~~~~~~~~~
NRF905_ADDR_SIZE_RX
C:\Users\Andrei\Google
Drive\Arduino\libraries\nRF905_Radio_Library\examples\wirelessSerialLink\wirelessSerialLink.ino:
In function 'void NRF905_CB_RXCOMPLETE()':
wirelessSerialLink:52:2: error: 'nRF905_read' was not declared in this scope
nRF905_read((byte*)newData, sizeof(newData));
^~~~~~~~~~~
```

(*ADMIN EDIT* Shortened error messages)

I've used your library in the past, but it was working fine. Any suggestion?

1.
   - Zak Kemble () on January 3, 2021
     Author
     # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209524)
   - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209524#respond)

   Hey Andrei, that's expected. The wireless serial example has a comment at the top that says "DONT USE THIS EXAMPLE, IT STILL NEEDS TO BE UPDATED FOR THE NEW VERSION OF THE NRF905 LIBRARY (v4)".

1.
   - Andrei on January 3, 2021
     # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209525)
   - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209525#respond)

   :)) I'm sure you are tired of ignorants.
   Sorry for bothering you. I needed the code, I knew it as a working code and took it…surprise.

   1.
      - Zak Kemble () on January 3, 2021
        Author
        # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209526)
      - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209526#respond)

      It's fine 🙂 If you really need the old stuff working again then you can get v3.1.0 from here https://github.com/zkemble/nRF905/releases/tag/v3.1.0 (https://github.com/zkemble/nRF905/releases/tag/v3.1.0)

14.
   - Pablo on April 13, 2021
     # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209599)
     Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209599#respond)

   Hi, are you usign Manchester Encoded data before GFSK Modulation in "lowpwr_node_sensor" example?

   Thanks and amazing repository! 🙂

   1.
      - Zak Kemble () on April 14, 2021
        Author
        # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209600)
      - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209600#respond)

      Hi Pablo, data is automatically manchester encoded and decoded by the radio IC.

15.
   - Jarosław on April 21, 2021
     # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209602)
     Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209602#respond)

   Hello.
   I want to pair 2 devices. I have a problem with addressing. I generate random addresses for devices, and this is not a problem. The problem is to assign a generated UINT32_T number in HEX format to the variable in which I want to keep the address. I tried through string, but without the effect. There is still a problem with the data type.
   Can you help me?
   Greetings.

   p.s. code:
   uint32_t RXADDR = 0xE7E7E7E7; //this work
   uint32_t buforWysylany.adresMiernika = random(268435456,368435456);
   RXADDR = "0x"+ (buforWysylany.adresMiernika ,HEX); //this is my problem

   1.
      - Zak Kemble () on April 21, 2021
        Author
        # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209603)
      - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209603#respond)

      Hi Jaroslaw, the 0x part tells the compiler that its a hex value, which tells it to convert 0xE7E7E7E7 to 3890735079 decimal during compilation. All you need is `RXADDR = buforWysylany.adresMiernika;` without any string conversion.

      1.
         - Jarosław on April 21, 2021
           # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209604)
         - Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209604#respond)

Hmmm.
I think I tried so. I will try again.
Thank you for the quick reply.

16.
- henry on April 23, 2021
- # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209606)
- Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209606#respond)

Hi all,

i would like to be able to send a single message to all my NRF905 and get a message back.

Is this possible or i need to send individual message to each radio devices ?

BR/

1.
- Zak Kemble () on April 23, 2021
- Author
- # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209607)
- Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209607#respond)

Hi Henry, if all of the receiving radios have the same address then they will all receive the same message. However, if you also want all of them to reply to that message then you will have to make sure that only 1 radio transmits at a time, otherwise the transmissions will corrupt each other. The library does include some collision avoidance where it checks for other transmissions before transmitting, but it may not be 100% effective. It's probably easier to send a message to each radio one at a time (each radio will need a unique address).

1.
- henry on April 23, 2021
- # (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209608)
- Reply (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/?replytocom=209608#respond)

Hi Zak,

thanks for a such prompt answer !

No unfortunately not, each device will have its own address. i was thinking, at "switch on" of the server, to interrogate all the "client" devices in order to know which ones are up and running and optionally to ask them for the revision of their software.
I guess i will stick to a certain range of addressees and ping them all sequentially. Unless you come up with good idea 🙂

BR/

« Prev (/nrf905-avrarduino-librarydriver/?cpage=4#comments) | 1 (/nrf905-avrarduino-librarydriver/?cpage=1#comments) | … | 3 (/nrf905-avrarduino-librarydriver/?cpage=3#comments) | 4 (/nrf905-avrarduino-librarydriver/?cpage=4#comments) | 5

Leave a Reply

Your email address will not be published.

Your message

Name

Email

Website (optional)

Are you human? *

7 − five = ☐ ⟳

☐ **Notify me of followup comments via e-mail. You can also subscribe (https://blog.zakkemble.net/comment-subscriptions/?srp=569&srk=1a0c27cfa99f2c863a70b797971e8200&sra=s&srsrc=f) without commenting. Please do not use @outlook or @hotmail addresses.**

Submit Comment

Search for:

Search 🔍

Name Zak Kemble
Location UK
Email contact@zakkemble.net

Follow Me

(https://www.youtube.com/user/zakkemble)

(https://github.com/zkemble/)

(https://twitter.com/ZakKemble)

(https://hackaday.io/zakkemble)

(http://forum.zakkemble.net)

(http://feeds.zakkemble.net/ZaksElectronicsBlog)

(https://feedburner.google.com/fb/a/mailverify?uri=ZaksElectronicsBlog&loc=en_US)

Donate

Found something useful here? Donate (https://donate.zakkemble.net/blog/)

1

2

3

4

5

6

7

8

Recent Posts

- Repairing a Samsung HT-C460 Home Cinema System (https://blog.zakkemble.net/repairing-a-samsung-ht-c460-home-cinema-system/)
- AVR-GCC 11.1.0 for Windows 32 and 64 bit (https://blog.zakkemble.net/avr-gcc-builds/)
- Raspberry Pi 4 PCI-Express Bridge “Chip” (https://blog.zakkemble.net/rpi4-pci-express-bridge-chip/)
- Smart Power Bank Keep-Alive (https://blog.zakkemble.net/smart-power-bank-keep-alive/)
- Remote Mail Notifier (and GPS Tracker) (https://blog.zakkemble.net/remote-mail-notifier-and-gps-tracker/)

Recent Comments

- HBT-workbench: QCX+ für 30m – DL1HBT (https://dl1hbt.wordpress.com/2021/05/07/hbt-workbench-qcx-fur-30m/) on AVRDUDESS – A GUI for AVRDUDE (https://blog.zakkemble.net/avrdudess-a-gui-for-avrdude/comment-page-5/#comment-209612)
- Inacio Morais on AVRDUDESS – A GUI for AVRDUDE (https://blog.zakkemble.net/avrdudess-a-gui-for-avrdude/comment-page-5/#comment-209611)
- henry on nRF905 Radio Library for AVR and Arduino (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209608)
- Zak Kemble on nRF905 Radio Library for AVR and Arduino (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209607)
- henry on nRF905 Radio Library for AVR and Arduino (https://blog.zakkemble.net/nrf905-avrarduino-librarydriver/comment-page-5/#comment-209606)

Archives

- April 2021 (https://blog.zakkemble.net/2021/04/)
- June 2020 (https://blog.zakkemble.net/2020/06/)
- February 2020 (https://blog.zakkemble.net/2020/02/)
- January 2020 (https://blog.zakkemble.net/2020/01/)
- September 2019 (https://blog.zakkemble.net/2019/09/)

- September 2018 (https://blog.zakkemble.net/2018/09/)
- December 2017 (https://blog.zakkemble.net/2017/12/)
- August 2017 (https://blog.zakkemble.net/2017/08/)
- March 2016 (https://blog.zakkemble.net/2016/03/)
- August 2015 (https://blog.zakkemble.net/2015/08/)
- March 2014 (https://blog.zakkemble.net/2014/03/)
- September 2013 (https://blog.zakkemble.net/2013/09/)
- August 2013 (https://blog.zakkemble.net/2013/08/)
- May 2013 (https://blog.zakkemble.net/2013/05/)
- April 2013 (https://blog.zakkemble.net/2013/04/)
- February 2013 (https://blog.zakkemble.net/2013/02/)
- January 2013 (https://blog.zakkemble.net/2013/01/)
- October 2012 (https://blog.zakkemble.net/2012/10/)
- September 2012 (https://blog.zakkemble.net/2012/09/)
- July 2012 (https://blog.zakkemble.net/2012/07/)
- May 2012 (https://blog.zakkemble.net/2012/05/)
- November 2011 (https://blog.zakkemble.net/2011/11/)
- February 2011 (https://blog.zakkemble.net/2011/02/)
- January 2011 (https://blog.zakkemble.net/2011/01/)
- July 2010 (https://blog.zakkemble.net/2010/07/)

Categories

- 555 (https://blog.zakkemble.net/category/555/)
- Arduino (https://blog.zakkemble.net/category/arduino/)
- AVR (https://blog.zakkemble.net/category/microcontroller/avr/)
- GPS (https://blog.zakkemble.net/category/gps/)
- LCD (https://blog.zakkemble.net/category/lcd/)
- LEDs (https://blog.zakkemble.net/category/leds/)
- Libraries (https://blog.zakkemble.net/category/library/)
- Linux (https://blog.zakkemble.net/category/linux/)
- Microcontroller (https://blog.zakkemble.net/category/microcontroller/)
- Power (https://blog.zakkemble.net/category/power/)
- Radio (https://blog.zakkemble.net/category/radio/)
- Raspberry Pi (https://blog.zakkemble.net/category/raspberry-pi/)
- Software (https://blog.zakkemble.net/category/software/)
- Uncategorized (https://blog.zakkemble.net/category/uncategorized/)

Meta

- Log in (https://blog.zakkemble.net/wp-login.php)
- Entries feed (https://blog.zakkemble.net/feed/)
- Comments feed (https://blog.zakkemble.net/comments/feed/)
- WordPress.org (https://wordpress.org/)

^