Optimized High Speed Driver for nRF24L01(+) 2.4GHz Wireless Transceiver

Design Goals

This library fork is designed to be...

- More compliant with the manufacturer specified operation of the chip, while allowing advanced users to work outside the recommended operation.
- Utilize the capabilities of the radio to their full potential via Arduino
- More reliable, responsive, bug-free and feature rich
- Easy for beginners to use, with well documented examples and features
- Consumed with a public interface that's similar to other Arduino standard libraries

News

Dec 2015

- ESP8266 support via Arduino IDE
- Particle Photon/Core fork available
- ATTiny2313/4313 support added
- Python 3 support added
- **RF24** added to Arduino library manager
- **RF24** added to PlatformIO library manager

March 2015

- Uses SPI transactions on Arduino
- New layout for easier portability: Break out defines & includes for individual platforms to RF24/utility
- MRAA support added (Galileo, Edison, etc)
- Generic Linux support (SPIDEV) support
- Support for RPi 2 added
- Major Documentation cleanup & update (Move all docs to github.io)

If issues are discovered with the documentation, please report them here

Useful References

- **RF24** Class Documentation
- **Download**
- **Source Code**
- **My Blog:** RF24 Optimization Overview
- **My Blog:** RPi/Linux w/RF24Gateway
- Chip Datasheet

Additional Information and Add-ons

- **RF24Network:** OSI Network Layer for multi-device communication. Create a home sensor network.
- **RF24Mesh:** Dynamic Mesh Layer for RF24Network
- **RF24Ethernet:** TCP/IP Radio Mesh Networking (shares Arduino Ethernet API)
- **RF24Audio:** Realtime Wireless Audio streaming

- All TMRh20 Documentation Main Page

More Information and RF24 Based Projects

- Project Blog: TMRh20.blogspot.com
- Maniacal Bits Blog
- MySensors.org (User friendly sensor networks/IoT)
- RF24Node_MsgProto (MQTT)
- RF24SensorNet
- Home Automation for Geeks
- Original Maniacbug RF24Network Blog Post
- ManiacBug on GitHub (Original Library Author)

Platform Support Pages

- **Arduino** (Uno, Nano, Mega, Due, Galileo, etc)
- **ATTiny**
- **Linux devices**(**RPi** , **Linux SPI userspace device**, **MRAA** supported boards (Galileo, Edison, etc), **LittleWire**)
- **Cross-compilation** for linux devices
- **Python** wrapper available for Linux devices

General µC Pin layout (See the individual board support pages for more info)

The table below shows how to connect the the pins of the NRF24L01(+) to different boards. CE and CSN are configurable.

PIN	NRF24L01	Arduino UNO	ATtiny25/45/85 [0]	ATtiny44/84 [1]	LittleWire [2]	RPI	RPi -P1 Connector
1	GND	GND	pin 4	pin 14	GND	rpi-gnd	(25)
2	VCC	3.3V	pin 8	pin 1	regulator 3.3V required	rpi-3v3	(17)
3	CE	digIO 7	pin 2	pin 12	pin to 3.3V	rpi-gpio22	(15)
4	CSN	digIO 8	pin 3	pin 11	RESET	rpi-gpio8	(24)
5	SCK	digIO 13	pin 7	pin 9	SCK	rpi-sckl	(23)
6	MOSI	digIO 11	pin 6	pin 7	MOSI	rpi-mosi	(19)
7	MISO	digIO 12	pin 5	pin 8	MISO	rpi-miso	(21)
8	IRQ	-	-	-	-	-	-

- [0] https://learn.sparkfun.com/tutorials/tiny-avr-programmer-hookup-guide/attiny85-use-hints
- [1] http://highlowtech.org/?p=1695
- [2] http://littlewire.cc/

Generated on Sat Nov 5 2016 10:02:27 for Optimized High Speed NRF24L01+ Driver Class Documenation by doxygen 1.8.10