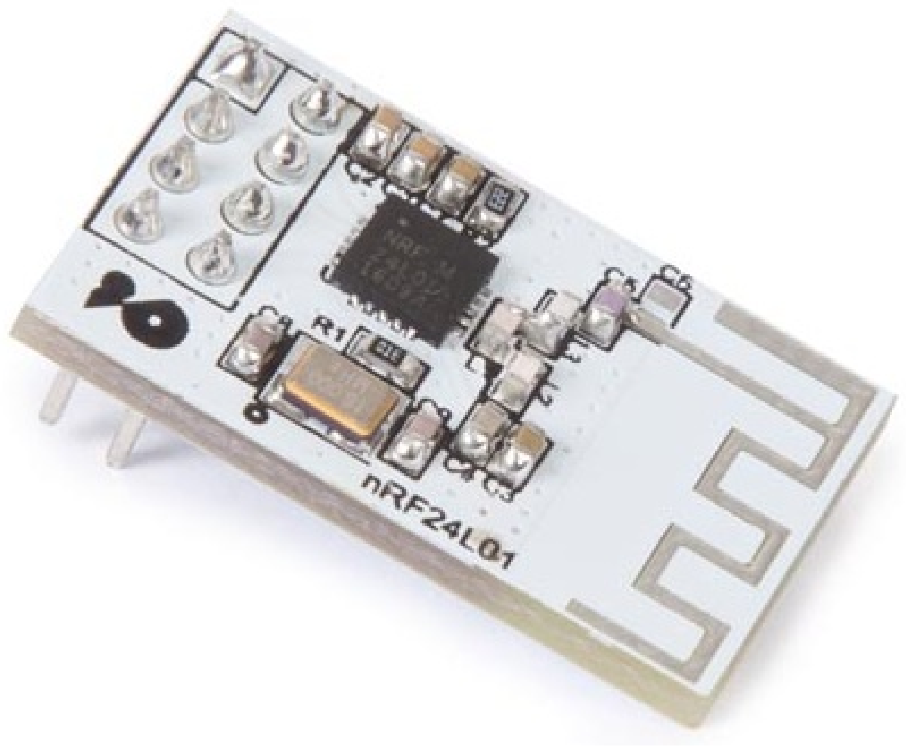
WHADDA WPI322 2.4 GHz NRF24L01 Wireless Transceiver Module User Manual

Contents [hide

WHADDA WPI322 2.4 GHz NRF24L01 Wireless Transceiver Module



Introduction

To all residents of the European Union Important environmental information about this product This symbol on the device or the package indicates that disposal of the device after its lifecycle could harm the environment. Do not dispose of the unit (or batteries) as unsorted municipal waste; it should be taken to a specialized company for recycling. This device should be returned to your distributor or to a local recycling service. Respect the local environmental rules. If in doubt, contact your local waste disposal authorities. Thank you for choosing Whadda! Please read the manual thoroughly before bringing this device into service. If the device was damaged in transit, do not install or use it and contact your dealer.

Safety Instructions

- Read and understand this manual and all safety signs before using this appliance.
- For indoor use only.
- This device can be used by children aged from 8 years and above, and persons with reduced physical, sensory or mental capabilities or lack of experience and knowledge if they have been given supervision or instruction concerning the use of the device in a safe way and understand the hazards involved. Children shall not play with the device. Cleaning and user maintenance shall not be made by children without supervision.

General Guidelines

- Refer to the Velleman® Service and Quality Warranty on the last pages of this manual.
- All modifications of the device are forbidden for safety reasons. Damage caused by user modifications to the device is not covered by the warranty.
- Only use the device for its intended purpose. Using the device in an unauthorized way will void the warranty.
- Damage caused by disregard of certain guidelines in this manual is not covered by the warranty and the dealer will not accept responsibility for any ensuing defects or problems.
- Nor Velleman Group nv nor its dealers can be held responsible for any damage (extraordinary, incidental or indirect) – of any nature (financial, physical…) arising from the possession, use or failure of this product.

- Keep this manual for future reference.

What is Arduino®

Arduino® is an open-source prototyping platform based on easy-to-use hardware and software. Arduino® boards are able to read inputs – light-on sensor, a finger on a button or a Twitter message – and turn it into an output – activating of a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so, you use the Arduino programming language (based on Wiring) and the Arduino® software IDE (based on Processing). Additional shields/modules/components are required for reading a twitter message or publishing online. Surf to www.arduino.cc for more information.

RED Declaration of Conformity

Hereby, Velleman Group nv declares that the radio equipment type WPI322 is in compliance with Directive 2014/53/EU. The full text of the EU declaration of conformity is available at the following internet address: www.velleman.eu.

Product Overview

This module is based on Nordic®'s nRF24L01, a highly integrated, ultra-low power (ULP) 2 Mbps RF transceiver for the 2.4 GHz ISM (Industrial, Scientific and Medical) band. The Nordic® nRF24L01+ integrates a complete 2.4 GHz RF transceiver, RF synthesizer, and baseband logic including the Enhanced ShockBurst™ hardware protocol accelerator supporting a high-speed SPI interface for the application controller.

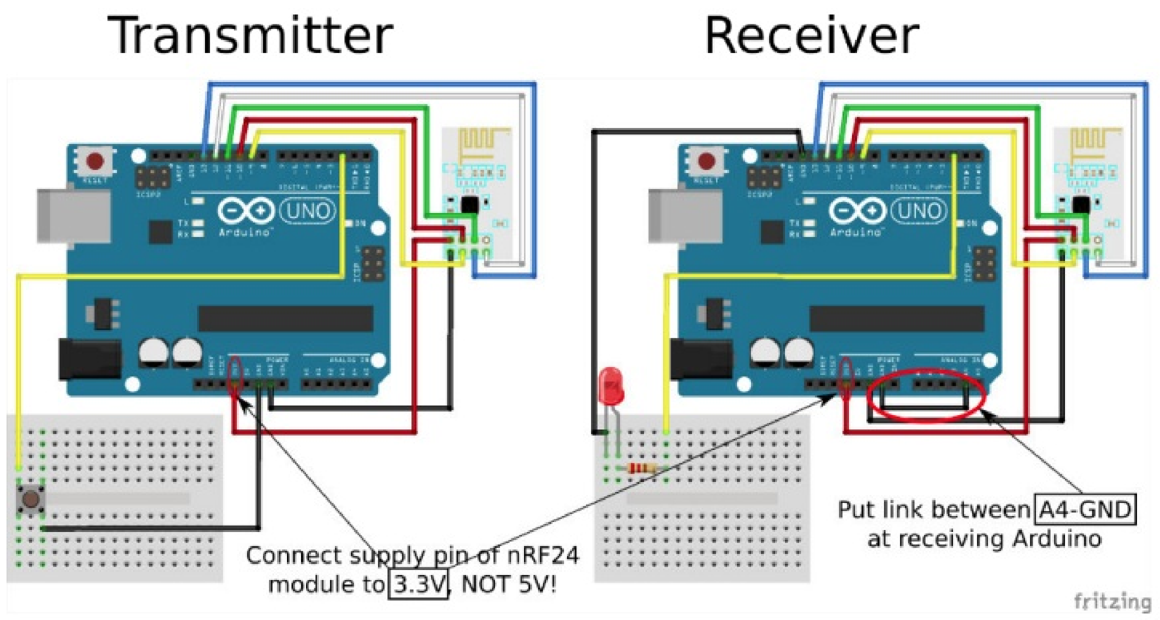
Specifications

- **power supply:** 1.9-3.6 V
- **IO port working voltage:** 0-3.3 V
- **transmitting rate:** +7 dB
- **receiving sensitivity:** < 90 dB
- **transmission range:** 250 m in open area
- **dimensions:** 15 x 29 mm

Features

- worldwide license-free 2.4 GHz ISM band operation
- 250 kbps, 1 Mbps and 2 Mbps on-air data-rate options
- Enhanced ShockBurst™ hardware protocol accelerator
- ultra-low power consumption – months to years of battery lifetime

Connection
You will need two Arduino® boards and at least two RF modules, one to transmit and the other to receive. If you use a standard Arduino® board, you should use the Arduino® board's 3.3 V pin (VDD3V3) to provide power (the nRF24L01 module has 1.9-3.6 V power voltage level). Please note: do not use 5 V pin (VDD5V) to provide power as this may destroy the module.


Transmitter
Receiver
Connect supply pin of nRF24 module to 3.3V, NOT 5V!
Put link between A4-GND at receiving Arduino
fritzing

RF module	**Arduino® Uno compatible board**
GND	GND
VCC	3.3V
CE	D9
CSN	D10
SCK	D13
MOSI	D11
MISO	D12

Upload the included example program to both the transmitting and receiving Arduino®. Make sure to install a jumper wire between analogue pin A4 and GND on the Arduino® board that needs to act as the receiver.

Example
This is an example of how to use the RF24 library to control an LED wirelessly from another Arduino® using two NRF24L01 modules. Start by wiring up two Arduino®s with the NF24L01 according to the provided schematic. Connect on one Arduino® a button to digital pin 2. Connect on the other Arduino® an LED (through an appropriate current limiting resistor, 220 ohms is recommended), and a jumper between analog pin 4 and GND to select this Arduino® as the receiver. You can upload the same sketch to both Arduino®s. Once completed and running, the LED on the receiving side should toggle on/off when pressing the button connected to the transmitting Arduino®. Example based on led_remote from nRF24 library Copyright (C) 2011 J. Coliz <maniacbug@ymail.com> This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License version 2 as published by the Free Software Foundation. Modified by Midas Gossye (Velleman Group nv)
*

- #include <SPI.h>

- #include “nRF24L01.h”
- #include “RF24.h”
- #include “printf.h”
- // Hardware configuration
- // Set up nRF24L01 radio on SPI bus plus pins 9 & 10 (CE & CS) RF24 radio(9,10);
- // sets the role of this unit in hardware. Connect to GND to be the ‘led’ board receiver
- // Leave open to be the ‘remote’ transmitter const int role_pin = A4;
- // Pin on the remote for button const uint8_t button_pin = 2; const uint8_t num_button_pins = sizeof(button_pin);
- // Pin on the LED board const uint8_t led_pin = 2; const uint8_t num_led_pins = sizeof(led_pin);
- // Topology
- // Single radio pipe address for the 2 nodes to communicate. const uint64_t pipe = 0xE8E8F0F0E1LL;
- // Role management
- // Set up role. This sketch uses the same software for all the nodes in this // system. Doing so greatly simplifies testing. The hardware itself specifies // which node it is.
- // This is done through the role_pin
- // The various roles supported by this sketch typedef enum { role_remote = 1, role_led } role_e;
- // The debug-friendly names of those roles const char* role_friendly_name[] = { “invalid”, “Remote”, “LED Board”};
- // The role of the current running sketch role_e role;
- // Payload uint8_t button_state; uint8_t led_state;
- // Setup void setup(void) {
- // Role
- // set up the role pin pinMode(role_pin, INPUT); // set role pin as input digitalWrite(role_pin,HIGH); // enable internal pull-up role pin delay(20); // slight delay just to get a solid reading on the role pin
- // read the address pin, establish our role (transmitter or receiver) if (digitalRead(role_pin)) role = role_remote; else role = role_led;
- // Print preamble
    - Serial.begin(115200); // Begin serial communication for debugging
    - printf_begin();
    - printf(“\n\rRF24/examples/led_remote/\n\r”);
    - printf(“ROLE: %s\n\r”,role_friendly_name[role]);
- // Setup and configure rf radio radio.begin();
- // Open pipes to other nodes for communication
- // This simple sketch opens a single pipes for these two nodes to communicate // back and forth. One listens on it, the other talks to it. if (role == role_remote) { radio.openWritingPipe(pipe); }
  else { radio.openReadingPipe(1,pipe); }
- // Start listening if (role == role_led) radio.startListening();
- // Dump the configuration of the rf unit for debugging radio.printDetails();
- // Set up button / LED
- // Set pull-up resistor for button pinMode(button_pin,INPUT); digitalWrite(button_pin,HIGH);
- // Turn LED ON until we start getting button responses
    - pinMode(led_pin,OUTPUT);

    - led_state = HIGH;
    - digitalWrite(led_pin,led_state); } Loop oid loop(void) {
- // Remote role. If the state of any button has changed, send the whole state of
- // all buttons.
- // if (role == role_remote) {
- // Get the current state of buttons, and
- // Test if the current state is different from the last state we sent bool different = false; uint8_t state = ! digitalRead(button_pin); if (state != button_state) { different = true; button_state = state; }
- // Send the state of the buttons to the LED board if (different) { printf("Now sending…"); Serial.println(button_state);
- // Send the payload, 1 byte, by passing address of button_state variable (&button_state indicates address of button_state variable) bool ok = radio.write(&button_state, 1); if (ok) printf("ok\n\r"); else printf("failed\n\r"); }
- // Try again in a short while delay(20);
- // LED role. Receive the state of all buttons, and reflect that in the LEDs if (role == role_led) {
- // if there is data ready if (radio.available()) { // Dump the payloads until we've gotten everything while (radio.available()) {
- // Fetch the payload, 1 byte, and put result in button_state variable (&button_state indicates address of button_state variable) radio.read(&button_state, 1);
- // Spew it printf("Got buttons\n\r"); Serial.println(button_state);
- // For each button, if the button now on, then toggle the LED if (!button_state) { led_state ^= HIGH; digitalWrite(led_pin,led_state); }

Power on both Arduino®s. You can confirm if the RF modules are properly connected by opening the serial monitor (set to 115200 baud) and checking the RX and TX addresses are set to a non-zero value. Pressing the button connected to the transmitting Arduino® should toggle on/off the LED connected to the receiving Arduino®. For more info, please, consult **http://playground.arduino.cc/InterfacingWithHardware/Nrf24L01**.

- **whadda.com**

Documents / Resources

	WHADDA WPI322 2.4 GHz NRF24L01 Wireless Transceiver Module [pdf] User Manual WPI322 2.4 GHz NRF24L01 Wireless Transceiver Module, WPI322, 2.4 GHz NRF24L01 Wireless Transceiver Module, NRF24L01 Wireless Transceiver Module, Wireless Transceiver Module, Transceiver Module

References

- **Arduino Playground - Nrf24L01**
- **Whadda - Exciting Electronics**
- **Arduino - Home**

- Arduino - Home
- Velleman â Wholesaler and developer of electronics