

## Проектирование и реализация беспроводной связи между микроконтроллерами с использованием nRF24L01 в диапазоне 2.4 ГГц

*Болтовский Лев Александрович*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

### Аннотация

Целью данной статьи является проектирование и реализация беспроводной связи между микроконтроллерами. В исследовании используются микроконтроллеры Atmega mega328pb и радиомодули nRF24L01+, прошивки написаны с использованием фреймворка Arduino Wiring и библиотек RF24 и util/crc16.h. Результатом исследования станет готовое решение, которое может быть имплементировано в реальные проекты.

**Ключевые слова:** Atmega mega328pb, nRF24L01+, Arduino Wiring

### Design and implementation of wireless communication between microcontrollers using nRF24L01+ in the 2.4 GHz band

*Boltovskiy Lev Alexandrovich*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

The purpose of this paper is to design and implement wireless communication between microcontrollers. The study uses Atmega mega328pb microcontrollers and nRF24L01+ radio modules, the firmware is written using Arduino Wiring framework and RF24 and util/crc16.h libraries. The result of the research will be a ready solution that can be implemented in real projects.

**Keywords:** Atmega mega328pb, nRF24L01+, Arduino Wiring

## 1 Введение

### 1.1 Актуальность исследования

Беспроводная передача данных активно используется в современном мире, растёт количество Интернет вещей и смарт устройств.

Модуль nRF24L01+ отличается своей энергоэффективностью и популярностью, активно используется при создании сетей датчиков, умных домов, различных гаджетов и автоматизированных систем. Данный модуль применим при обучении робототехнике и выступает основным устройством, используемым для реализации беспроводного управления. Работа с данным модулем так же может быть полезна с точки зрения исследования надёжности и устойчивости связи по причине большого количества устройств, работающих в том же частотном диапазоне.

## 1.2 Обзор исследований

Принципы построения сетей, находящихся в диапазоне радиочастот от 2.4 ГГц до 6 ГГц, регламентируются техническим стандартом IEEE 802.11 [1]. Работа с данным стандартом является объектом исследования многих авторов. М. А. Перегудов, А. С. Стешковой, А. В. Щеглов в своём исследовании [2] рассматривают особенности физического и канального уровня компьютерных сетей в рамках этого стандарта, особое внимание уделено помехоустойчивости.

Исследуемый радиомодуль работает в диапазоне ISM (Industrial, Scientific, and Medical). Его ключевой особенностью является отсутствие необходимости в лицензировании регулирующими органами. Более подробно об ISM можно узнать в исследовании В. И. Сидоровой [3].

Пример внедрения в устройство nRF24L01+ приведён Т. Г. Корминим, Р. В. Моховым, К. М. Чайниковым [4]. Использование радиомодуля на предприятии описывается в исследовании В.Н.Усенкова, О.А.Комарова [5].

## 1.3 Цель исследования

Цель исследования - разработать и оценить методы эффективного и надежного обмена данными между микроконтроллерами с использованием радиомодуля nRF24L01 в диапазоне 2.4 ГГц для приложений в беспроводных системах и Интернета вещей.

## 2 Материалы и методы

Для достижения поставленной цели используются модуль nRF24L01+ и микроконтроллер Atmel Mega328pb в видео отладочной платы Arduino Nano. Среда разработки – Arduino IDE.

## 3 Результаты обсуждения

Перед началом работы следует ознакомиться с официальной документацией к чипам для nRF24L01+ [6] и для Atmega Mega 328p [7]. Для отладки и прошивки микроконтроллера используется Arduino IDE [8]. Модуль и плата представлены на рисунке (рис. 1).

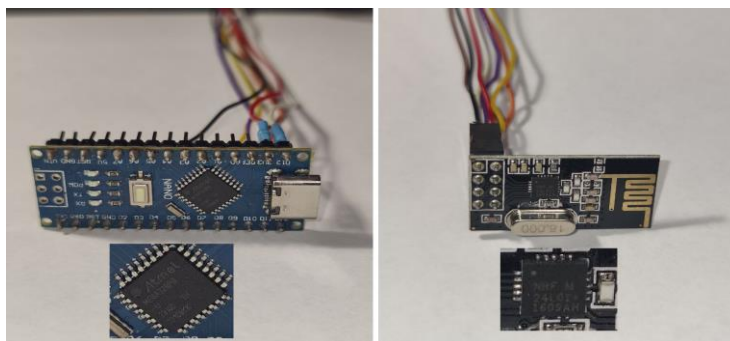


Рисунок 1 – Arduino Nano и nRF24L01+

Для сборки передатчика и приёмника потребуется несколько модулей. Назначение выводов nRF24L01+ представлено на рисунке (рис. 2).

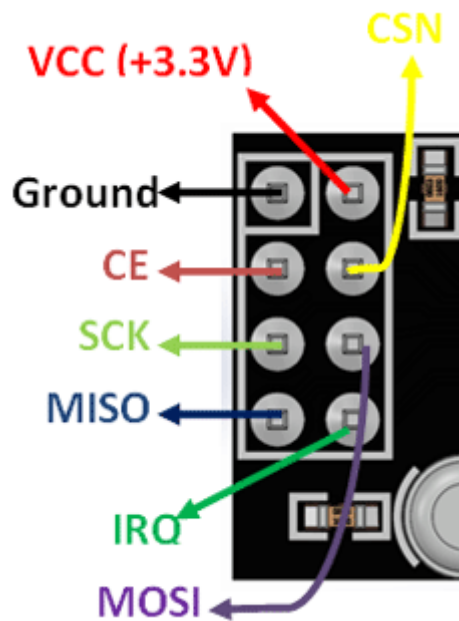


Рисунок 2 – Распиновка радиомодуля

Схема подключения представлена в таблице (табл. 1).

Таблица 1 – Подключение

nRF24L01	Назначение	Подключение
1	Ground	GND
2	VCC (3.3)	3.3V
3	CE	D9
4	CSN	D10
5	SCK	D13
6	MOSI	D11
7	MISO	D12
8	IRQ	Не используется

Собранные устройства представлены на рисунке (рис. 3).

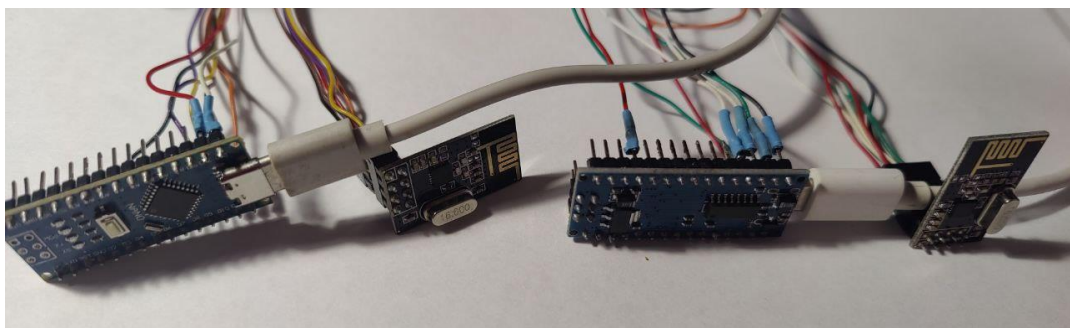


Рисунок 3 – Собранные приёмник и передатчик

Код для приёмника и передатчика представлен на рисунке (рис. 4).

```

1 // Передатчик
2
3 #include <SPI.h>
4 #include <nRF24L01.h>
5 #include <RF24.h>
6
7 // Настройка пинов CE и CSN
8 RF24 radio(9, 10);
9
10 // Адрес канала передачи, должен совпадать на передатчике и приемнике
11 const byte address[6] = "00001";
12
13 void setup() {
14   Serial.begin(9600);
15   radio.begin();
16   radio.openWritingPipe(address); // Открытие канала для передачи данных
17   radio.setPALevel(RF24_PA_MIN); // Уровень мощности передатчика
18   radio.stopListening(); // Отключение режима приема
19 }
20
21 void loop() {
22   const char text[] = "Hello, world!";
23   radio.write(&text, sizeof(text)); // Отправка данных
24   Serial.println("Сообщение отправлено: Hello, world!");
25   delay(1000); // Задержка перед следующей отправкой
26 }

```

```

1 // Приёмник
2
3 #include <SPI.h>
4 #include <nRF24L01.h>
5 #include <RF24.h>
6
7 // Настройка пинов CE и CSN
8 RF24 radio(9, 10);
9
10 // Адрес канала приема, должен совпадать на передатчике и приемнике
11 const byte address[6] = "00001";
12
13 void setup() {
14   Serial.begin(9600);
15   radio.begin();
16   radio.openReadingPipe(0, address); // Открытие канала для приема данных
17   radio.setPALevel(RF24_PA_MIN); // Уровень мощности приемника
18   radio.startListening(); // Включение режима приема
19 }
20
21 void loop() {
22   if (radio.available()) { // Проверка наличия входящих данных
23     char text[32] = ""; // Буфер для хранения полученного текста
24     radio.read(&text, sizeof(text)); // Чтение данных
25     Serial.print("Получено сообщение: ");
26     Serial.println(text);
27   }
28 }

```

Рисунок 4 – Код приёмника и передатчика

Для обмена данными между микроконтроллерами с использованием модуля nRF24L01+ можно использовать библиотеку RF24, которая упрощает работу с этим радиомодулем.

Передатчик отправляет строку текста "Hello, world!" каждую секунду. Данные передаются на указанный адрес, который должен совпадать с адресом приемника. Приемник постоянно прослушивает канал и, если данные доступны, принимает их и выводит на монитор порта.

Данный код обладает недостатком: в нём нет проверки целостности отправляемого сообщения.

Для передачи данных с проверенной целостностью можно использовать алгоритм контрольной суммы, например CRC (Cyclic Redundancy Check), который добавляет в пакет данных специальное значение, позволяющее приемнику проверить правильность принятой информации. Перед отправкой данных передатчик вычисляет CRC и

добавляет его к данным. Контрольная сумма (CRC) помогает убедиться, что данные не были искажены во время передачи. Приемник получает данные и вычисляет CRC на своей стороне. Если вычисленная CRC совпадает с принятой CRC, данные считаются корректными. Если CRC не совпадают, данные считаются поврежденными, и могут быть отклонены или запрошены повторно.

Для вычисления CRC можно использовать библиотеку `util/crc16.h`, которая доступна для Arduino и предоставляет простые функции для работы с CRC. Пример реализации подсчёта контрольной суммы на передатчике представлен на рисунке (рис. 5).

```
31 packet.crc = _crc16_update(0, packet.number & 0xFF); // Вычисление CRC
32 packet.crc = _crc16_update(packet.crc, (packet.number >> 8) & 0xFF);
33
34 // Отправка данных с CRC
35 bool success = radio.write(&packet, sizeof(packet));
36 if (success) {
37     Serial.print("Отправлено число: ");
38     Serial.print(packet.number);
39     Serial.print(" с CRC: ");
40     Serial.println(packet.crc, HEX);
41 } else {
42     Serial.println("Ошибка отправки данных.");
43 }
```

Рисунок 5 – Проверка контрольной суммы на передатчике

Аналогичную работу необходимо выполнить и на стороне приёмника (рис. 6).

```
31 // Вычисление CRC на стороне приемника для проверки целостности
32 uint16_t crc_check = _crc16_update(0, packet.number & 0xFF);
33 crc_check = _crc16_update(crc_check, (packet.number >> 8) & 0xFF);
34
35 // Сравнение CRC
36 if (crc_check == packet.crc) {
37     Serial.print("Получено число: ");
38     Serial.print(packet.number);
39     Serial.print(" с CRC: ");
40     Serial.println(packet.crc, HEX);
41 } else {
42     Serial.println("Ошибка целостности данных!");
43 }
44 }
45 }
46 }
```

Рисунок 6 – Проверка контрольной суммы на приёмнике

В передатчике CRC вычисляется с помощью функции `_crc16_update`, которая обновляет сумму по каждому байту числа. В нашем случае число (int) делится на два байта, и CRC вычисляется для каждого байта отдельно. Передатчик отправляет пакет данных (число и CRC) через радиомодуль nRF24L01. Приемник получает пакет данных и заново вычисляет CRC для принятого числа.

Если вычисленное значение CRC совпадает с CRC, пришедшим вместе с данными, данные считаются корректными и выводятся на экран. Если CRC не совпадает, приемник выводит сообщение об ошибке целостности, указывая на возможное повреждение данных во время передачи.

Эти меры позволяют проверить целостность передаваемых данных и обеспечить надежную передачу информации между микроконтроллерами.

#### 4 Выводы

Использование радиомодуля nRF24L01 позволяет эффективно и надежно организовать обмен данными между микроконтроллерами, а внедрение методов проверки целостности данных значительно повышает устойчивость системы к помехам и ошибкам передачи, что делает данный подход перспективным для разработки беспроводных сетей и приложений Интернета вещей.

#### Библиографический список

1. IEEE [Электронный ресурс]. URL: <https://www.ieee.org/> (дата обращения: 04.09.2024).
2. Перегудов М. А. Описательная модель канального уровня сетей цифровой радиосвязи семейства стандартов IEEE 802.11 // Системы управления, связи и безопасности. 2020. № 3. С. 203-221.
3. Сидорова В. И. Некоторые особенности диапазона ISM // Молодой ученый. 2022. № 22(417). С. 26-28.
4. Кормин Т. Г. Прием и передача данных на Arduino через радио-модуль nrf24l01 // Фундаментальные и прикладные научные исследования. 2018. С. 74-78.
5. Усенков В. Н. Построение низкоскоростной беспроводной связи для малых производственных предприятий на базе микросхемы NRF24L01 // Инфокоммуникации и информационная безопасность 2016. С. 202-208.
6. PRELIMINARY PRODUCT SPECIFICATION [Электронный ресурс]. URL: [https://www.sparkfun.com/datasheets/Components/nRF24L01\\_prelim\\_prod\\_spec\\_1\\_2.pdf](https://www.sparkfun.com/datasheets/Components/nRF24L01_prelim_prod_spec_1_2.pdf) (дата обращения: 04.09.2024).
7. ATMEGA328P [Электронный ресурс]. URL: <https://www.microchip.com/product/ATmega328P> (дата обращения: 04.09.2024).
8. Arduino IDE [Электронный ресурс]. URL: <https://www.arduino.cc/en/software> (дата обращения: 04.09.2024).