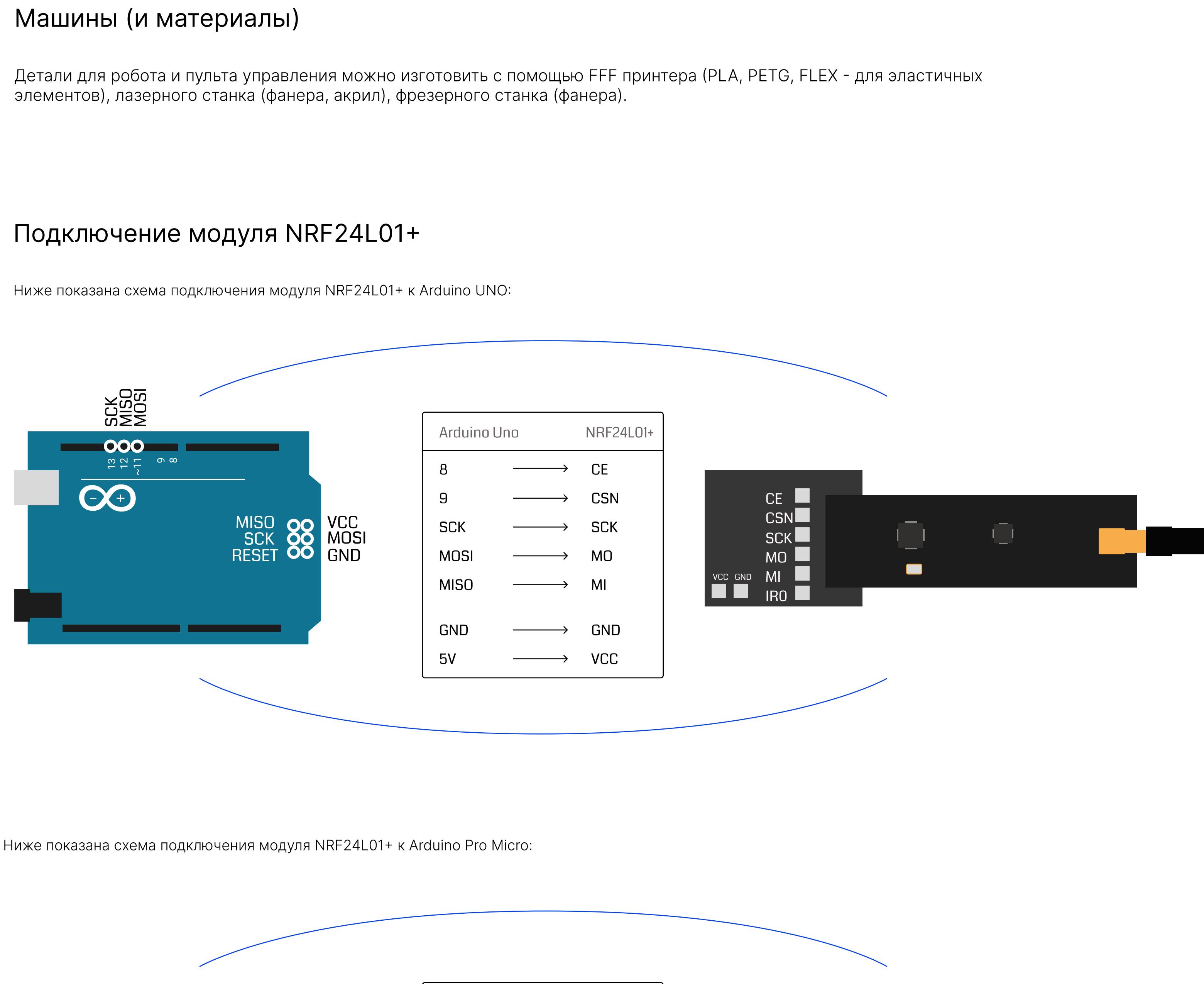


## 1 Электронные компоненты

Стандартный набор включает в себя моторы, контроллер, две аккумуляторные батареи, радио-модули, тумблеры, концевик, потенциометр, RGB LED матрица и джойстик.



**W!** Не вращайте вручную вал мотора или сервопривода

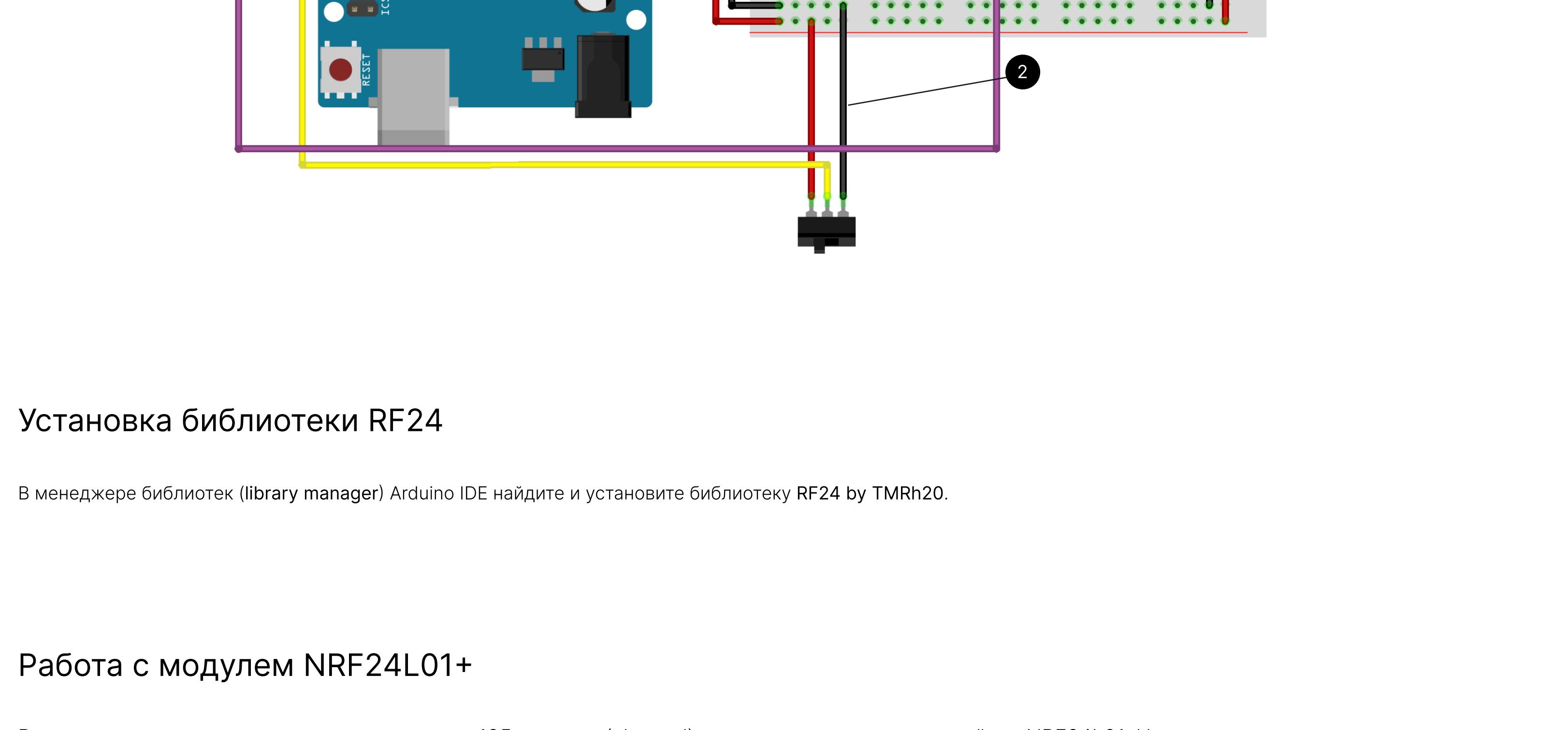
**W!** Перед использованием все аккумуляторы нужно полностью зарядить. Для зарядки аккумуляторов используйте только зарядное устройство в К-020 (-1 этаж)

## 2 Машины (и материалы)

Детали для робота и пульта управления можно изготовить с помощью FFF принтера (PLA, PETG, FLEX - для эластичных элементов), лазерного станка (фанера, акрил), фрезерного станка (фанера).

## 3 Подключение модуля NRF24L01+

Ниже показана схема подключения модуля NRF24L01+ к Arduino UNO:



## 4 Пример пульта управления

Ниже показана схема подключения радиомодуля NRF24L01+ (1), тумблера (2), джойстика (3), потенциометра (4) и кнопки (5) к Arduino UNO:



## 5 Установка библиотеки RF24

В менеджере библиотек (library manager) Arduino IDE найдите и установите библиотеку RF24 by TMRh20.

## 6 Работа с модулем NRF24L01+

В одном пространстве можно развернуть до 125 каналов (channel) связи между парами устройств NRF24L01. На каждом канале можно разместить 1 приемник и до 6 передатчиков или труб (pipe).

Канал устанавливается с помощью метода `.setChannel(n)`, где `n` - номер канала в диапазоне от 0 до 125. Частота канала вычисляется по формуле:  $2400 \text{ MHz} + n$ :

```
radio.setChannel(3);
```

Скорость передачи данных устанавливается с помощью метода `.setDataRate(rate)`, где `rate` принимает одно из следующих значений: RF24\_250KBPS, RF24\_1MBPS, RF24\_2MBPS:

```
radio.setDataRate(RF24_1MBPS);
```

Мощность передатчика (чем больше расстояние между устройствами, тем большая мощность необходима) устанавливается с помощью метода `.setPAlevel(level)`, где `level` принимает одно из следующих значений: RF24\_PA\_MIN, RF24\_PA\_LOW, RF24\_PA\_HIGH, RF24\_PA\_MAX:

```
radio.setPAlevel(RF24_PA_HIGH);
```

Level	dBm
RF24_PA_MIN	-18
RF24_PA_LOW	-12
RF24_PA_HIGH	-6
RF24_PA_MAX	0

Для того чтобы два NRF устройства могли найти друг друга нужно установить для них одинаковый адрес трубы. Адрес может состоять из символов A, B, C, D, E, F и цифр от 0 до 9. Труба с уникальным адресом может быть открыта только для записи или только для чтения. Адрес трубы имеет вид:



Адрес трубы устанавливается с помощью метода `.openWritingPipe(address)` для передатчика и метода `.openReadingPipe(id, address)` для приемника, где `id` - номер трубы (с 0 по 5), `address` - адрес трубы:

```
radio.openWritingPipe(0x1234567890LL);
```

```
radio.openReadingPipe(1, 0x1234567890LL);
```

## Пример кода для передатчика (пульта)

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
```

```
RF24 radio(8, 10); // CSN, CE
int data[2];
```

```
void setup() {
    radio.begin();
    radio.setChannel(1);
    radio.setDataRate(RF24_1MBPS);
    radio.setPAlevel(RF24_PA_HIGH);
    radio.openWritingPipe(0x1234567890LL);
}
```

```
void loop() {
    data[0] = 0;
    data[1] = 1;

    radio.write(&data, sizeof(data));
    delay(1000);
}
```

## Пример кода для приемника (робот)

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
```

```
RF24 radio(8, 9); // CSN, CE
int data[2];
```

```
void setup() {
    Serial.begin(9600);
}
```

```
radio.begin();
radio.setChannel(1);
radio.setDataRate(RF24_1MBPS);
radio.setPAlevel(RF24_PA_HIGH);
radio.openReadingPipe(0, 0x1234567890LL);
radio.startListening();
```

```
void loop() {
    if (radio.available()) {
        radio.read(&data, sizeof(data));
        Serial.println(data[0]);
    }
}
```

## 7 Документирование

Необходимо создать репозиторий на GitHub с подробным описанием робота (все схемы, итерации, проблемы и решения, компоненты, видео и фото работы).

Пример