

NRF24L01 Radio Transceiver

Introduction

The NRF24L01 is a module which includes a radio chip and antenna. It operates at frequencies around 2.4GHz. The modules can be set to one of 125 (0 - 124) channels, with each channel being separated by 1MHz and multiple modules can communicate on a single channel.

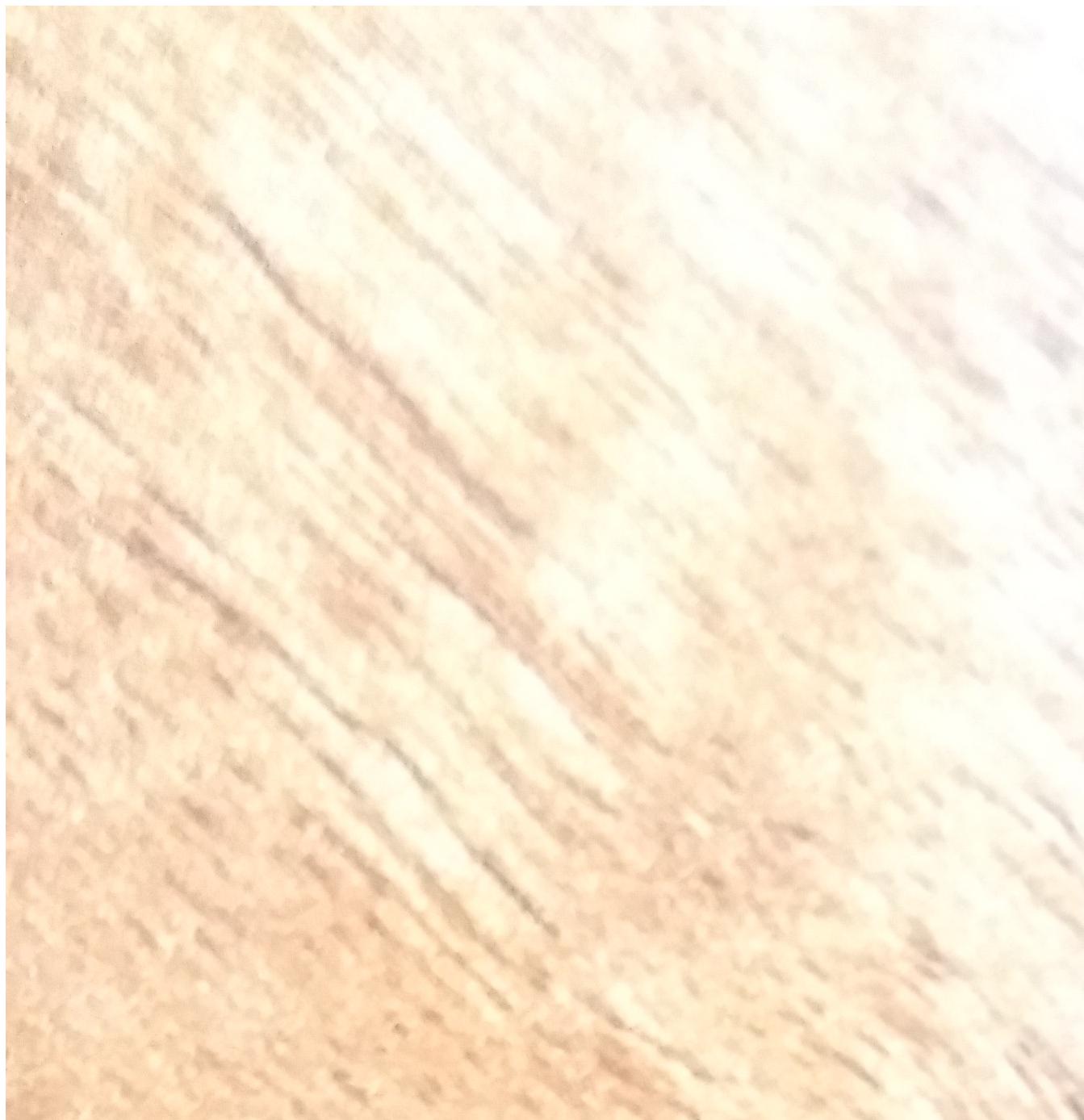
The module is very useful for short range radio communication, small data packets can be sent between multiple data packets for fast wireless communication, great for remote control devices like toys, home automation and robotics.

Library

- There are a couple of options to use here, but I am using "NRFLite" by Dave Parson, which can be installed via the Library manager
- This library has a lot of good examples that I recommend you check out to see the capabilities of the module, there is also a YouTube tutorial:
<https://www.youtube.com/watch?v=tWEgvS7Sj-8&t=223s>

Connections

- I have provided a breakout board which makes the NRF24L01 easy to connect to a breadboard, it needs to be soldered. Pay attention to the image:







IRQ

MISO

MOSI

SCK

CSN

CE

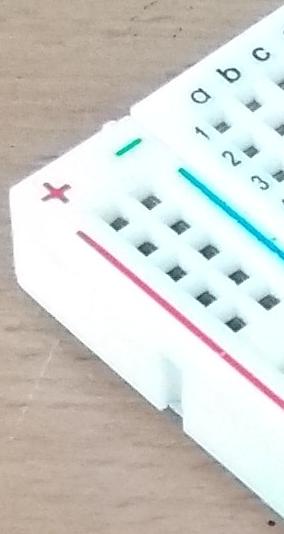
3V3



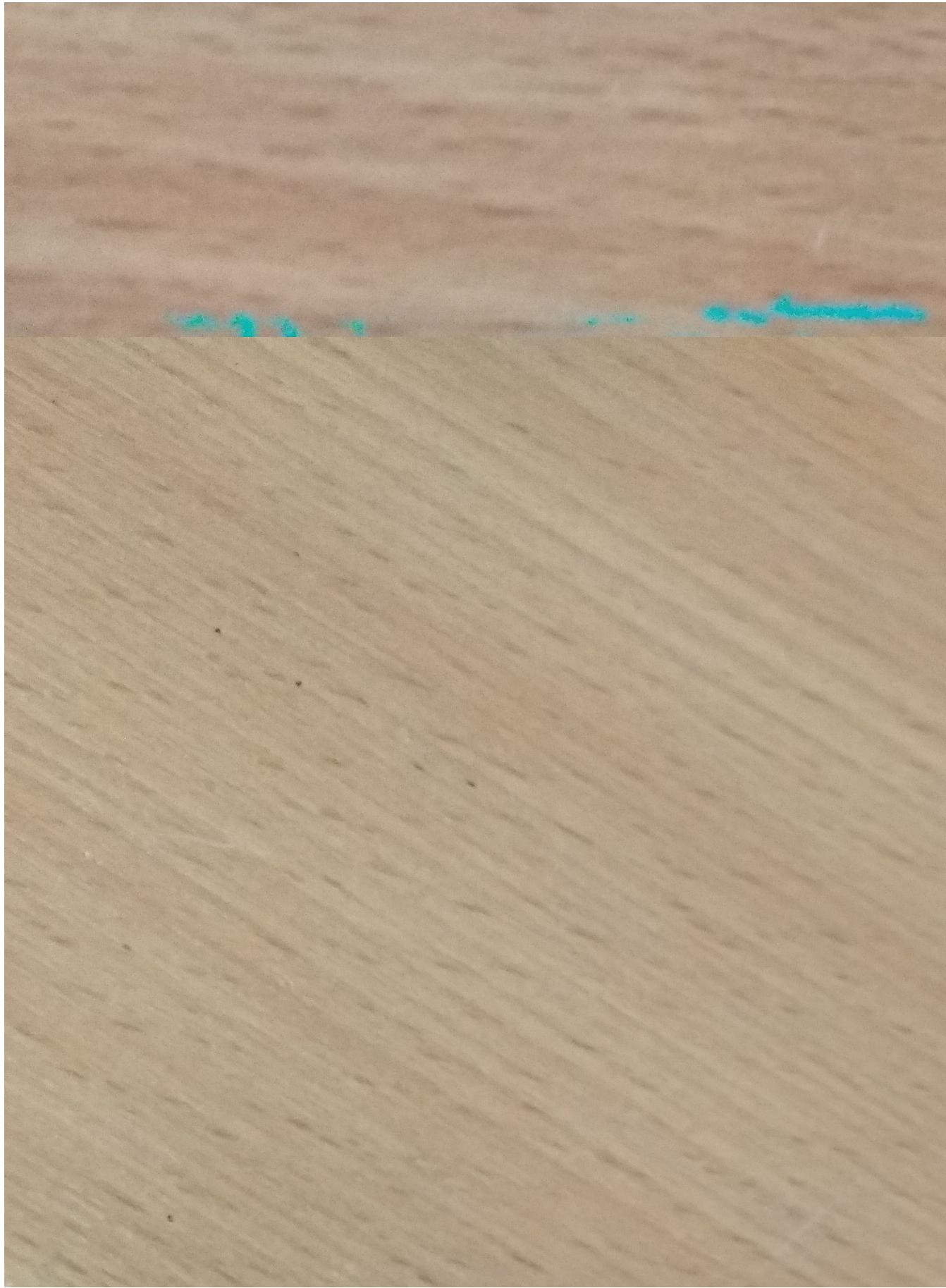
- **WARNING** This is a 3v3 device, not 5V!! Connecting to 5V will permanently damage the module, use the 3v3 pin on the Arduino to power the module
- Connect the module like so:

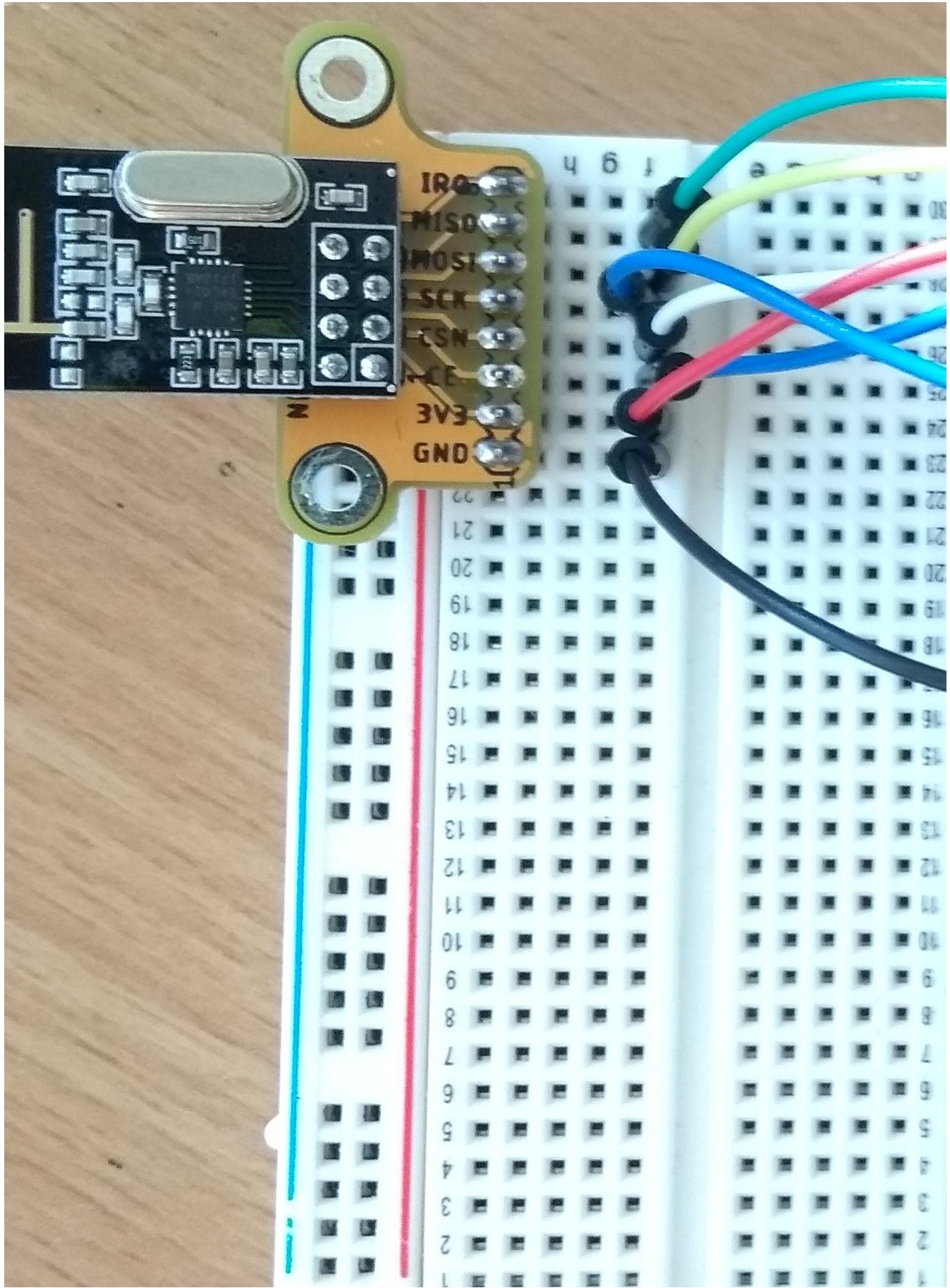
NRF24L01 Arduino Uno

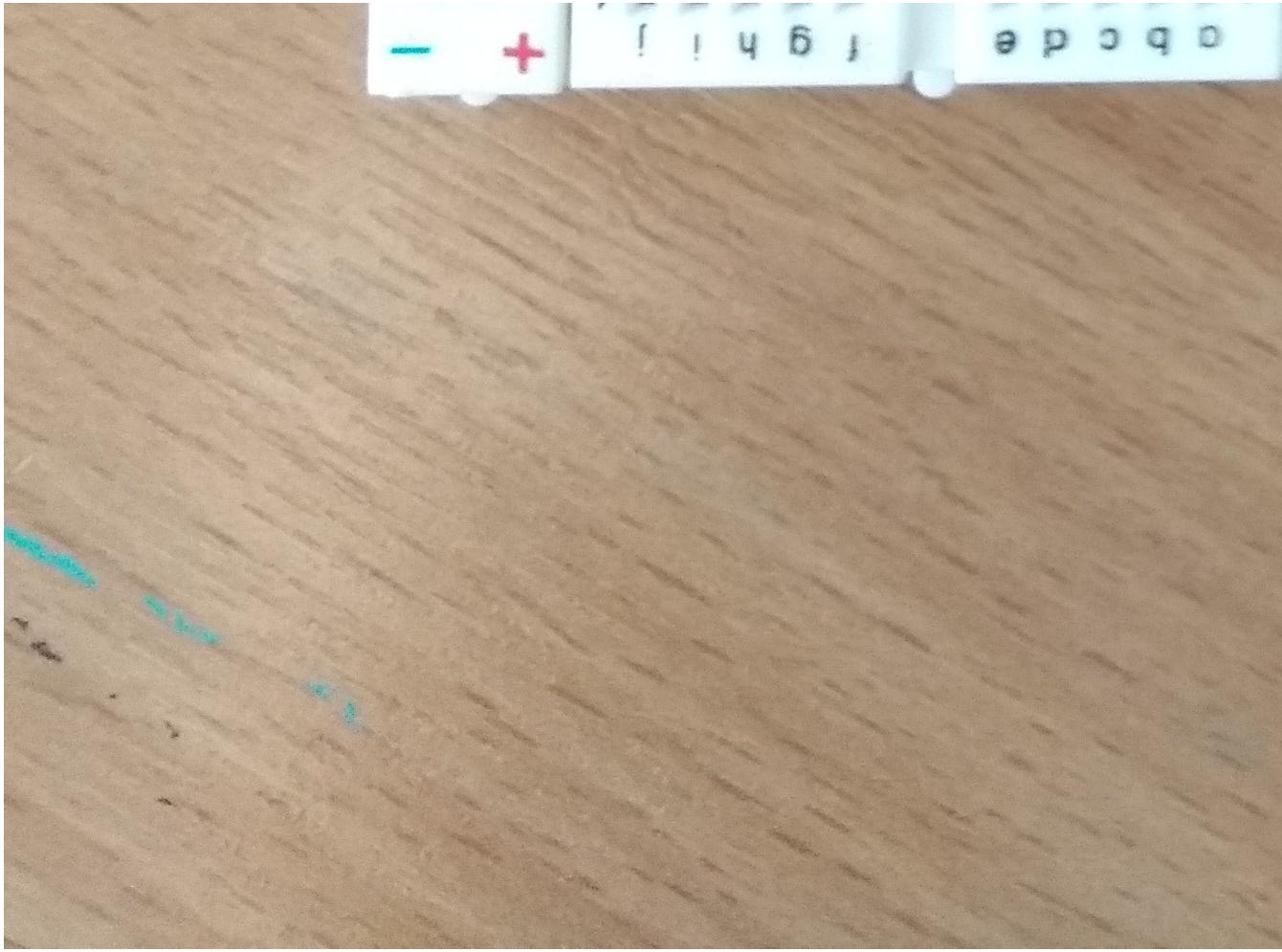
3v3	3v3
GND	GND
CE	9
CSN	10
MOSI	11
MISO	12
SCK	13
IRQ	Not required











Examples

Basic Transmit

This example shows how to send a basic message from the NRF24L01, and should be paired with an Arduino running the Basic Receive example below:

- please set the radio channel in the example sketch to your group number (found on Moodle) otherwise radio traffic is going to cause problems.
- Each transceiver module on a channel should have a unique ID, which should just be a number between 0 - 255 (one byte). You need to set the transmit radio's ID for this Arduino, and also the ID of the receiver radio you want to send to.

```
/* NRF24L01 Transmit Example
 * This example uses Dave Parson's NRFLite Library
 *
 * Example by Aidan Taylor 2018 for Engineering Science
 * Cardiff Metropolitan University
 */
/*
Connections:

Radio    Arduino
CE      -> 9
CSN     -> 10 (Hardware SPI SS)
MOSI    -> 11 (Hardware SPI MOSI)
MISO    -> 12 (Hardware SPI MISO)
SCK     -> 13 (Hardware SPI SCK)
IRQ     -> No connection
VCC     -> No more than 3.6 volts
GND     -> GND
*/
// Libraries:
#include <SPI.h>
#include <NRFLite.h>

// Radio Variables
byte chan = 1;
byte radio_id = 1; // this radio's ID
byte send_id = 0; // the ID of the receiver radio
byte cepin = 9;
byte cspin = 10;
byte radio_chan = 0; // please change this number to your group number

NRFLite radio; // this allows us to talk to the library in the sketch
byte counter = 0; // a variable to use to send some data
```

```

void setup() {
    Serial.begin(115200);

    // Test the radio is connected properly:
    if (!radio.init(radio_id, cepin, cspin, NRFLite::BITRATE2MBPS, radio_chan)) {
        Serial.println("Cannot communicate with radio");
        while(true); // Wait here forever.
    }
}

void loop() {
    Serial.print("Loop count: ");
    Serial.println(counter);
    Serial.println("Sending loop count...");

    if(radio.send(send_id, &counter, sizeof(counter))) { // & must be used in front of variable
        Serial.println("...Success");
    }
    else {
        Serial.println("...Failed");
    }

    counter++; // Add 1 to the counter variable
    delay(5000); // Wait 5 seconds
}

```

Basic Receive

This example shows how to receive a basic message from the NRF24L01, and should be paired with an Arduino running the Basic Transmit example above:

- please set the radio channel in the example sketch to your group number (found on Moodle) otherwise radio traffic is going to cause problems.
- Each transciever module on a channel should have a unique ID, which should just be a number between 0 - 255 (one byte). You need to set this radio's ID so that the transmitter sends to the correct target ID.

```

/* NRF24L01 Receive Example
 * This example uses Dave Parson's NRFLite Library
 *
 * Example by Aidan Taylor 2018 for Engineering Science
 * Cardiff Metropolitan University
 */

/*
Connections:

Radio      Arduino
CE         -> 9
CSN        -> 10 (Hardware SPI SS)
MOSI       -> 11 (Hardware SPI MOSI)
MISO       -> 12 (Hardware SPI MISO)
SCK        -> 13 (Hardware SPI SCK)
IRQ        -> No connection
VCC        -> No more than 3.6 volts
GND        -> GND
*/

#include <SPI.h>
#include <NRFLite.h>

// Radio Variables
byte radio_id = 0;
byte cepin = 9;
byte cspin = 10;
byte radio_chan = 0; // please change this number to your group number

NRFLite radio;

boolean data_rx = 0; // this variable will be updated with the incoming message

void setup() {
    Serial.begin(115200);

    // Check the radio is connected correctly:
    if (!radio.init(radio_id, cepin, cspin, NRFLite::BITRATE2MBPS, radio_chan)) {
        Serial.println("Cannot communicate with radio");
        while (1); // Wait here forever.
    }
}

void loop() {
    while(radio.hasData()) {
        radio.readData(&data_rx); // & must be placed in front of the variable

        String msg = "Message Received: ";
        msg += data_rx; // += will append the variable to the string msg

        Serial.println(msg); // print the message
    }
}

```

Button Transmit

This example requires a button attached to pin 2. A message is sent over radio when the button is pressed. The message can be used to turn on the LED in the next example:

```

/* NRF24L01 Transmit on Button Press Example
 * This example uses Dave Parson's NRFLite Library
 *
 * Example by Aidan Taylor 2018 for Engineering Science
 * Cardiff Metropolitan University

```

```

/*
/*
Connections:
Radio    Arduino
CE      -> 9
CSN     -> 10 (Hardware SPI SS)
MOSI    -> 11 (Hardware SPI MOSI)
MISO    -> 12 (Hardware SPI MISO)
SCK     -> 13 (Hardware SPI SCK)
IRQ     -> No connection
VCC     -> No more than 3.6 volts
GND     -> GND
*/
// Libraries:
#include <SPI.h>
#include <NRFLite.h>

// Radio Variables
byte chan = 1;
byte radio_id = 1; // this radio's ID
byte send_id = 0; // the ID of the receiver radio
byte cepin = 9;
byte cspin = 10;
byte radio_chan = 0; // please change this number to your group number

NRFLite radio; // this allows us to talk to the library in the sketch

// Button Variables:
byte buttonpin = 2;
boolean buttonstate = 1;
boolean lastbuttonstate = 1;

boolean mode = 0; // Mode can be the variable to be transmitted

void setup() {
  Serial.begin(115200);
  pinMode(buttonpin, INPUT_PULLUP);

  delay(1000);

  // Test the radio is connected properly:
  if (!radio.init(radio_id, cepin, cspin, NRFLite::BITRATE2MBPS, radio_chan)) {
    Serial.println("Cannot communicate with radio");
    while(true); // Wait here forever.
  }
}

void loop() {
  buttonstate = digitalRead(buttonpin);

  // check if button state has changed (high to low or low to high)
  if(buttonstate != lastbuttonstate) {
    lastbuttonstate = !lastbuttonstate; // reverse logic if true

    // if the button has been pressed down (low)
    if(buttonstate == LOW) {
      mode = !mode; // reverse logic of mode

      // Let's print mode as a message:
      if(mode == 1) {
        Serial.println("Turn on the LED");
      }
      else {
        Serial.println("Turn off the LED");
      }

      // Send mode variable as a packet over radio:
      if(radio.send(send_id, &mode, sizeof(mode))) { // & must be used in front of variable
        Serial.println("...Success");
      }
      else {
        Serial.println("...Failed");
      }
    }
  }
}
}

```

LED Receive

This sketch requires an LED attached to pin 2. On receipt of a message the LED will be activated or deactivated

```

/* NRF24L01 Radio controlled LED example
 * This example uses Dave Parson's NRFLite Library
 *
 * Example by Aidan Taylor 2018 for Engineering Science
 * Cardiff Metropolitan University
 */

/*
Connections:
Radio    Arduino
CE      -> 9
CSN     -> 10 (Hardware SPI SS)
MOSI    -> 11 (Hardware SPI MOSI)
MISO    -> 12 (Hardware SPI MISO)
SCK     -> 13 (Hardware SPI SCK)
IRQ     -> No connection
VCC     -> No more than 3.6 volts
GND     -> GND
*/

```

```

*/
#include <SPI.h>
#include <NRFLite.h>

// Radio Variables
byte radio_id = 0;
byte cepin = 9;
byte cspin = 10;
byte radio_chan = 0; // please change this number to your group number

NRFLite radio;

boolean data_rx = 0; // this variable will be updated with the incoming message

// LED
byte ledpin = 2;

void setup() {
  Serial.begin(115200);
  pinMode(ledpin, OUTPUT);

  // Check the radio is connected correctly:
  if (!radio.init(radio_id, cepin, cspin, NRFLite::BITRATE2MBPS, radio_chan)) {
    Serial.println("Cannot communicate with radio");
    while (1); // Wait here forever.
  }
}

void loop() {
  while(radio.hasData()) {
    radio.readData(&data_rx); // & must be placed in front of the variable

    String msg = "Message Received: ";
    if(data_rx == 1) {
      msg += "turn on the LED";
      digitalWrite(ledpin, HIGH);
    }
    else {
      msg += "turn off the LED";
      digitalWrite(ledpin, LOW);
    }

    Serial.println(msg); // print the message
  }
}

```