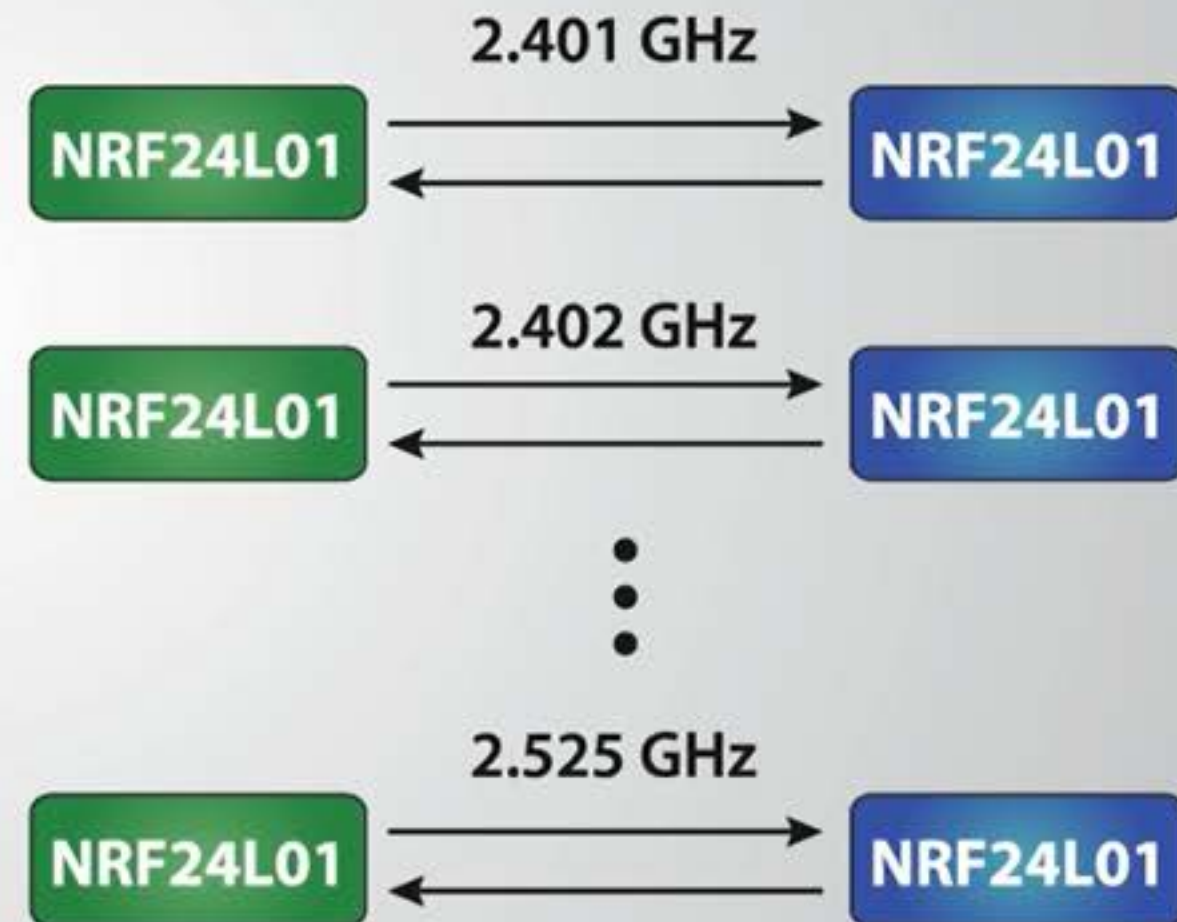# NRF24L01 Transceiver Module

- 2.4 GHz band

- Baud Rate: 250 kbps - 2 Mbps

- Range: 100 m
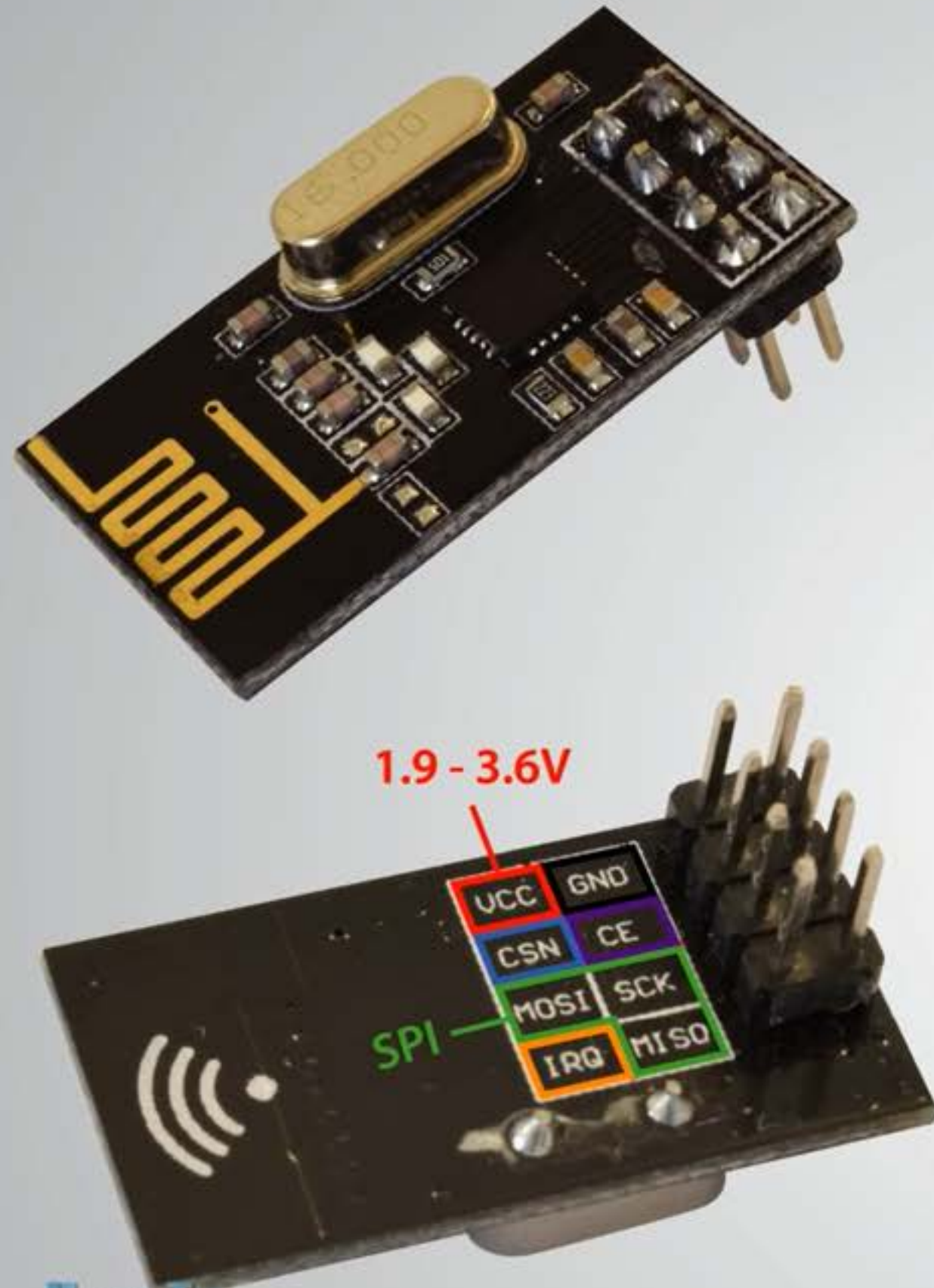

2400 to 2525 MHz
1MHz spacing - 125 channels

NRF24L01 Transceiver Module

- 2.4 GHz band

- Baud Rate: 250 kbps - 2 Mbps

- Range: 100 m

- Max current draw: ~12 mA

- Operating voltage: 1.9 - 3.6 V

- Other pins tolerate 5 V logic

1.9 - 3.6V

VCC GND
CSN CE
MOSI SCK
IRQ MISO

SPI

Arduino	MOSI	MISO	SCK	CS
Mega	51	50	52	53
Uno / Nano	11	12	13	10

Arduino Mega - NRF24L01

3.3V - VCC
GND - GND
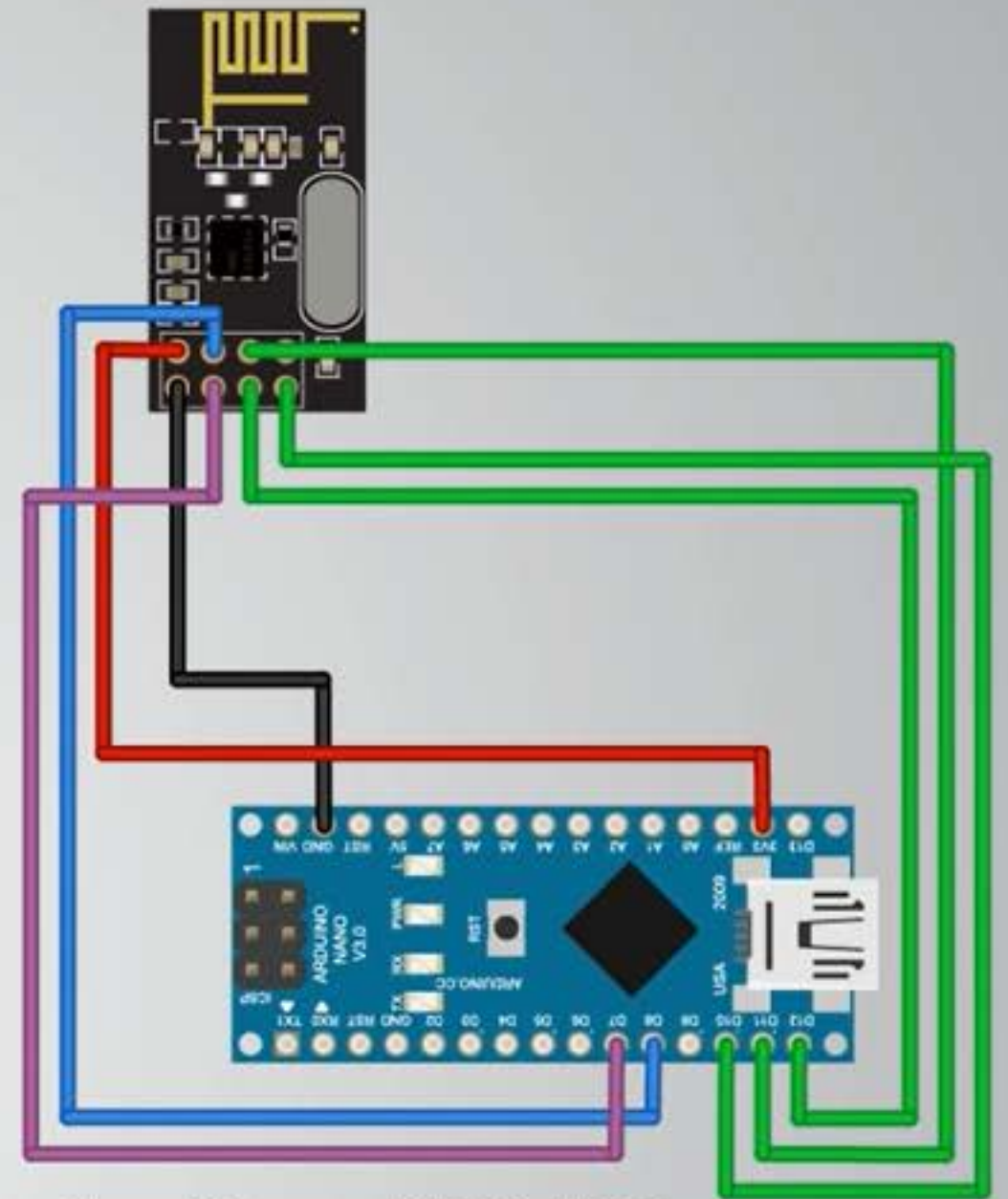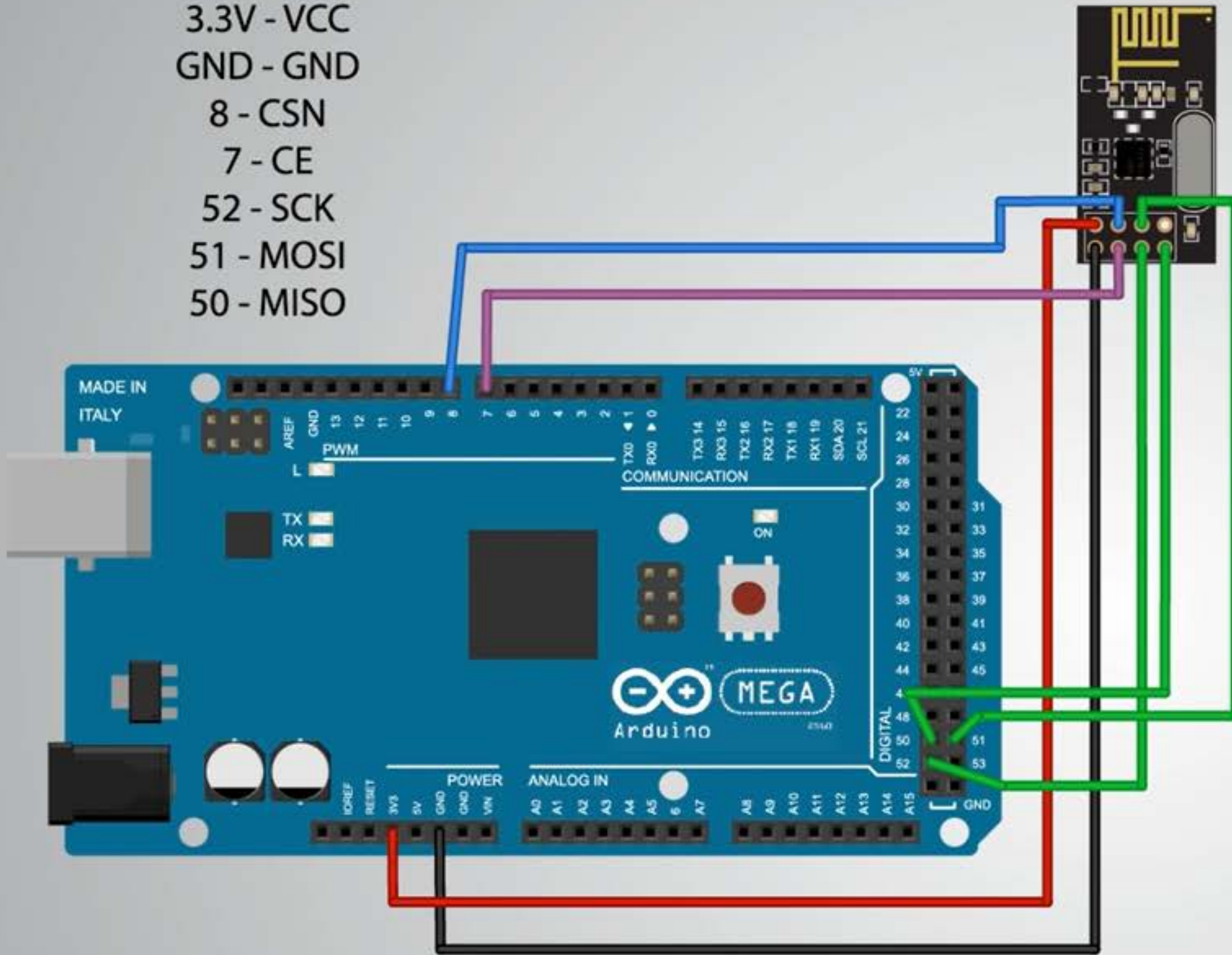8 - CSN
7 - CE
52 - SCK
51 - MOSI
50 - MISO

Arduino Uno/ Nano - NRF24L01

3.3V - VCC
GND - GND
8 - CSN
7 - CE
13 - SCK
11 - MOSI
12 - MISO

How To MECHATRONICS
www.HowToMechatronics.com

File Edit Sketch Tools Help

Transmitter_Example_1

```
/*
 * Arduino Wireless Communication Tutorial
 *      Example 1 - Transmitter Code
 *
 * by Dejan Nedelkovski, www.HowToMechatronics.com
 *
 * Library: TMRh20/RF24, https://github.com/tmrh20/RF24/
 */


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


RF24 radio(7, 8); // CNS, CE


const byte address[6] = "00001";


void setup() {
  radio.begin();
  radio.openWritingPipe(address);
  radio.setPALevel(RF24_PA_MIN);
  radio.stopListening();
}
```

File Edit Sketch Tools Help

Receiver_Example_1

```
/*
 * Arduino Wireless Communication Tutorial
 *       Example 1 - Receiver Code
 *
 * by Dejan Nedelkovski, www.HowToMechatronics.com
 *
 * Library: TMRh20/RF24, https://github.com/tmrh20/RF24/
 */


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


RF24 radio(7, 8); // CNS, CE


const byte address[6] = "00001";


void setup() {
  Serial.begin(9600);
  radio.begin();
  radio.openReadingPipe(0, address);
  radio.setPALevel(RF24_PA_MIN);
  radio.startListening();
```

File Edit Sketch Tools Help

Transmitter_Example_1

```
/*
 * Arduino Wireless Communication Tutorial
 *       Example 1 - Transmitter Code
 *
 * by Dejan Nedelkovski, www.HowToMechatronics.com
 *
 * Library:                ps://github.com/tmr
 */


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


RF24 radio(7, 8); // CNS, CE


const byte address[6] = "00001";


void setup() {
  radio.begin();
  radio.openWritingPipe(address);
  radio.setPALevel(RF24_PA_MIN);
  radio.stopListening();
}
```

NRF24L01

"123AB"

"00001"

NRF24L01   NRF24L01

"ABCDE"

NRF24L01

File Edit Sketch Tools Help

Receiver_Example_1

```
/*
 * Arduino Wireless Communication Tutorial
 *       Example 1 - Receiver Code
 *
 * by Dejan Nedelkovski, www.HowToMechatronics.com
 *
 * Library: TMRh20/RF24, https://github.com/tmrh20/RF24/
 */


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


RF24 radio(7, 8); // CNS, CE


const byte address[6] = "00001";


void setup() {
  Serial.begin(9600);
  radio.begin();
  radio.openReadingPipe(0, address);
  radio.setPALevel(RF24_PA_MIN);
  radio.startListening();
```

Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM4

Transmitter_Example_1

```
* Library: TMRh20/RF24, https://github.com/tmrh20/RF24/
*/


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


RF24 radio(7, 8); // CNS, CE


const byte address[6] = "00001";


void setup() {
  radio.begin();
  radio.openWritingPipe(address);
  radio.setPALevel(RF24_PA_MIN);
  radio.stopListening();
}


void loop() {
  const char text[] = "Hello World";
  radio.write(&text, sizeof(text));
  delay(1000);
}
```

Receiver_Example_1

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


RF24 radio(7, 8); // CNS, CE


const byte address[6] = "00001";


void setup() {
  Serial.begin(9600);
  radio.begin();
  radio.openReadingPipe(0, address);
  radio.setPALevel(RF24_PA_MIN);
  radio.startListening();
}


void loop() {
  if (radio.available()) {
    char text[32] = "";
    radio.read(&text, sizeof(text));
    Serial.println(text);
  }
}
```

File Edit Sketch Tools Help

File Edit Sketch Tools Help

Transmitter_Example_1

Receiver_Example_1

```
* Library: TMRh20/RF24, https://github.com/tmrh20/RF24/
*/

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

RF24 radio(7, 8); // CNS, CE

const byte address[6] = "00001";

void setup() {
  radio.begin();
  radio.openWritingPipe(address);
  radio.setPALevel(RF24_PA_MIN);
  radio.stopListening();
}

void loop() {
  const char text[] = "Hello World";
  radio.write(&text, sizeof(text));
  delay(1000);
}
```
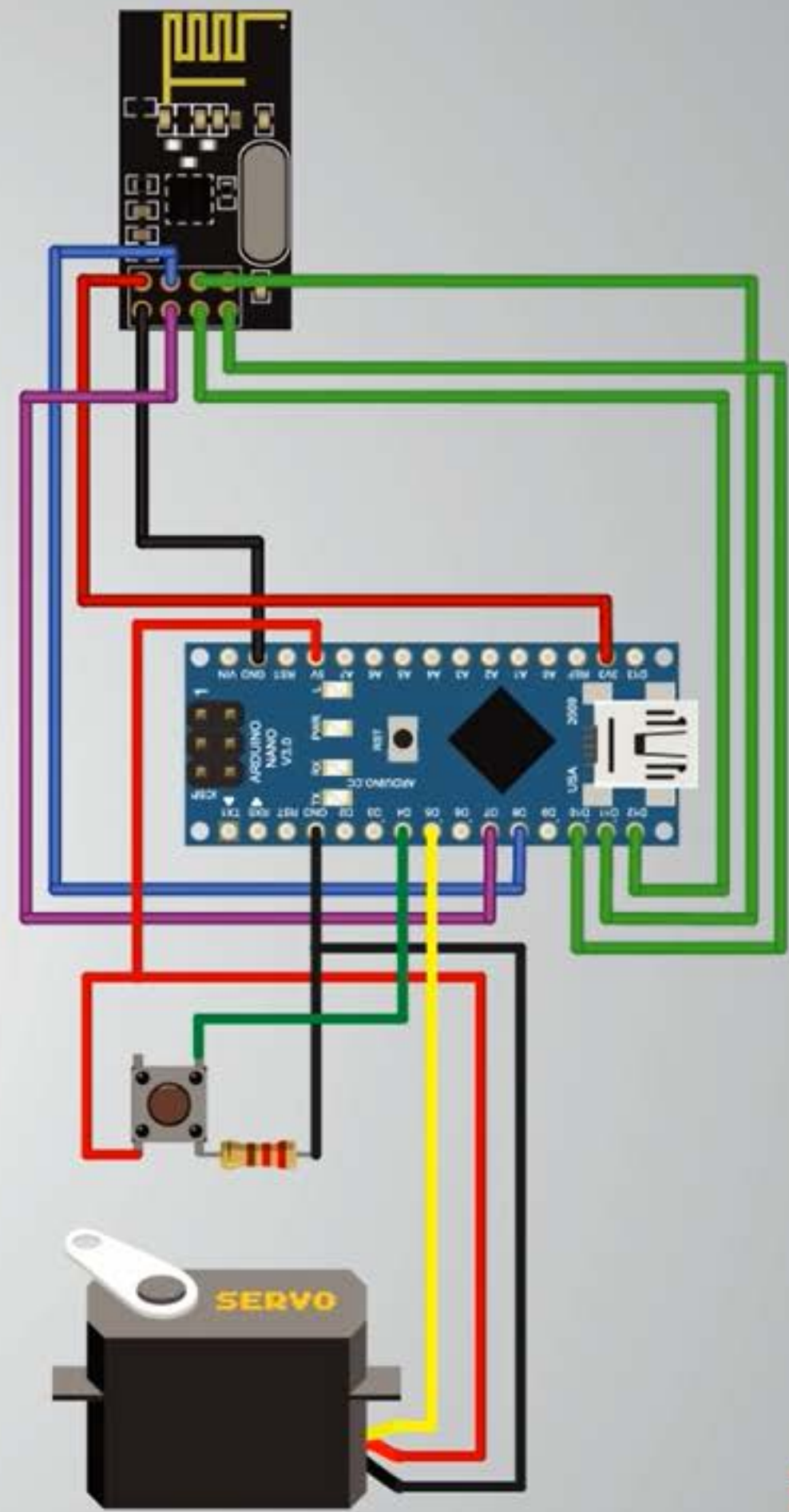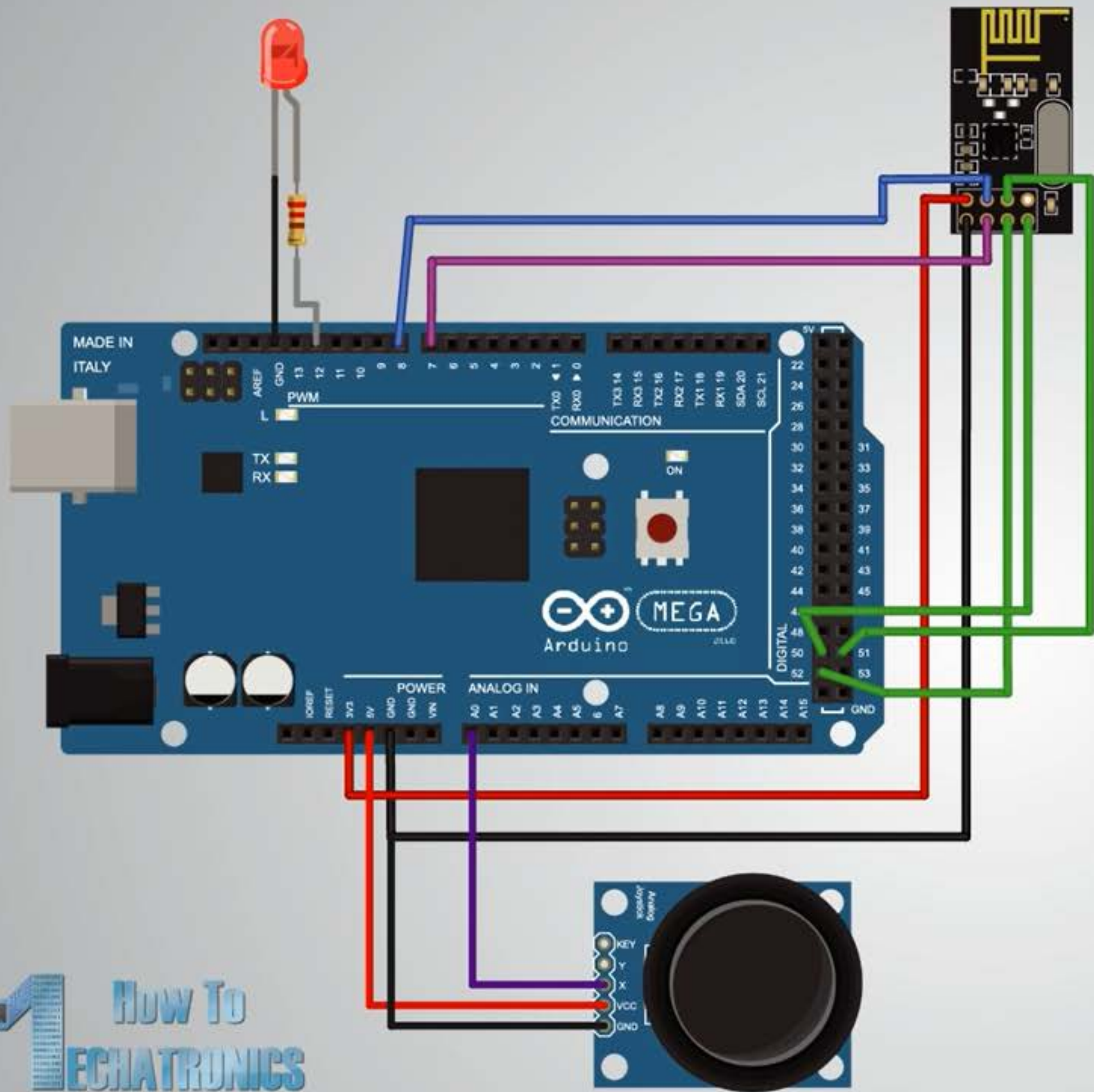
COM6

Send

Hello World
Hello World
Hello World
Hello World
Hello World

☑ Autoscroll   Both NL & CR ▾   9600 baud ▾

```
#include <SPI.h>
                              01.h>

                              // CNS, CE

                              ss[6] = "00001";

                              600);

                              ingPipe(0, address);
                              el(RF24_PA_MIN);
                              tening();

                              lable()) {
                              ] = "";
                              ext, sizeof(text));
      Serial.println(text);
  }
}
```

Done Saving.

Arduino Nano, ATmega328 on COM6

Arduino Nano, ATmega328 on COM6

www.HowToMechatronics.com

fritzing

Transmitter_Example_2 | Arduino 1.6.5

Transmitter_Example_2

```
/*
* Arduino Wireless Communication Tutorial
*      Example 2 - Transmitter Code
*
* by Dejan Nedelkovski, www.HowToMechatronics.com
*
* Library: TMRh20/RF24, https://github.com/tmrh20/RF24/
*/


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


#define led 12

RF24 radio(7, 8); // CNS, CE
const byte addresses[][6] = {"00001", "00002"};
boolean buttonState = 0;

void setup() {
  pinMode(12, OUTPUT);
  radio.begin();
  radio.openWritingPipe(addresses[1]); // 00001
```

Arduino Nano, ATmega328 on COM5

www.HowToMechatronics.com

Receiver_Example_2 | Arduino 1.6.5

Receiver_Example_2

```
/*
* Arduino Wireless Communication Tutorial
*      Example 2 - Receiver Code
*
* by Dejan Nedelkovski, www.HowToMechatronics.com
*
* Library: TMRh20/RF24, https://github.com/tmrh20/RF24/
*/


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


#define button 4

RF24 radio(7, 8); // CNS, CE
const byte addresses[][6] = {"00001", "00002"};
Servo myServo;
boolean buttonState = 0;

void setup() {
  pinMode(button, INPUT);
```

Arduino Nano, ATmega328 on COM5

Transmitter_Example_2

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


#define led 12


RF24 radio(7, 8); // CNS, CE
const byte addresses[][6] = {"00001", "00002"};
boolean buttonState = 0;


void setup() {
  pinMode(12, OUTPUT);
  radio.begin();
  radio.openWritingPipe(addresses[1]); // 00001
  radio.openReadingPipe(1, addresses[0]); // 00002
  radio.setPALevel(RF24_PA_MIN);
}


void loop() {
  delay(5);

  radio.stopListening();
  int potValue = analogRead(A0);
```

Arduino Nano, ATmega328 on COM6

Receiver_Example_2

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


#define button 4


RF24 radio(7, 8); // CNS, CE
const byte addresses[][6] = {"00001", "00002"};
Servo myServo;
boolean buttonState = 0;


void setup() {
  pinMode(button, INPUT);
  myServo.attach(5);
  radio.begin();
  radio.openWritingPipe(addresses[0]); // 00002
  radio.openReadingPipe(1, addresses[1]); // 00001
  radio.setPALevel(RF24_PA_MIN);
}


void loop() {
  delay(5);
```

Arduino Nano, ATmega328 on COM6

Transmitter_Example_2

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


#define led 12

RF24 radio(7, 8); // CNS, CE
const byte addresses[][6] = {"00001", "00002"};
boolean buttonState = 0;

void setup() {
  pinMode(12, OUTPUT);
  radio.begin();
  radio.openWritingPipe(addresses[1]); // 00001
  radio.openReadingPipe(1, addresses[0]); // 00002
  radio.setPALevel(RF24_PA_MIN);
}

void loop() {
  delay(5);

  radio.stopListening();
  int potValue = analogRead(A0);
```

Receiver_Example_2

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


#define button 4

RF24 radio(7, 8); // CNS, CE
const byte addresses[][6] = {"00001", "00002"};
Servo myServo;
boolean buttonState = 0;

void setup() {
  pinMode(button, INPUT);
  myServo.attach(5);
  radio.begin();
  radio.openWritingPipe(addresses[0]); // 00002
  radio.openReadingPipe(1, addresses[1]); // 00001
  radio.setPALevel(RF24_PA_MIN);
}

void loop() {
  delay(5);
```

```
  radio.begin();
  radio.openWritingPipe(addresses[1]); // 00001
  radio.openReadingPipe(1, addresses[0]); // 00002
  radio.setPALevel(RF24_PA_MIN);
}

void loop() {
  delay(5);

  radio.stopListening();
  int potValue = analogRead(A0);
  int angleValue = map(potValue, 0, 1023, 0, 180);
  radio.write(&angleValue, sizeof(angleValue));

  delay(5);
  radio.startListening();
  while (!radio.available());
  radio.read(&buttonState, sizeof(buttonState));
  if (buttonState == HIGH) {
    digitalWrite(led, HIGH);
  }
  else {
    digitalWrite(led, LOW);
```

```
  pinMode(button, INPUT);
  myServo.attach(5);
  radio.begin();
  radio.openWritingPipe(addresses[0]); // 00002
  radio.openReadingPipe(1, addresses[1]); // 00001
  radio.setPALevel(RF24_PA_MIN);
}

void loop() {
  delay(5);
  radio.startListening();         I
  if ( radio.available()) {
    while (radio.available()) {
      int angleV = 0;
      radio.read(&angleV, sizeof(angleV));
      myServo.write(angleV);
    }
    delay(5);
    radio.stopListening();
    buttonState = digitalRead(button);
    radio.write(&buttonState, sizeof(buttonState));
  }
}
```

Transmitter_Example_2

```
  radio.openReadingPipe(1, addresses[0]); // 00002
  radio.setPALevel(RF24_PA_MIN);
}

void loop() {
  delay(5);

  radio.stopListening();
  int potValue = analogRead(A0);
  int angleValue = map(potValue, 0, 1023, 0, 180);
  radio.write(&angleValue, sizeof(angleValue));

  delay(5);
  radio.startListening();
  while (!radio.available());
  radio.read(&buttonState, sizeof(buttonState));
  if (buttonState == HIGH) {
    digitalWrite(led, HIGH);
  }
  else {
    digitalWrite(led, LOW);
  }
}
```

Receiver_Example_2

```
  pinMode(button, INPUT);
  myServo.attach(5);
  radio.begin();
  radio.openWritingPipe(addresses[0]); // 00002
  radio.openReadingPipe(1, addresses[1]); // 00001
  radio.setPALevel(RF24_PA_MIN);
}

void loop() {
  delay(5);
  radio.startListening();
  if ( radio.available()) {
    while (radio.available()) {
      int angleV = 0;
      radio.read(&angleV, sizeof(angleV));
      myServo.write(angleV);
    }
    delay(5);
    radio.stopListening();
    buttonState = digitalRead(button);
    radio.write(&buttonState, sizeof(buttonState));
  }
}
```