

## گزارش کار ششم آزمایشگاه ریزپردازنده

تهیه و تنظیم: مبین خیبری

شماره دانشجویی: 994421017

استاد راهنما: دکتر نیکزاد

### چکیده:

در این جلسه، ابتدا اقدامات انجام شده در جلسات گذشته مورد بررسی قرار گرفتند و سپس توجه گروه‌های مختلف دانشجویان روی طراحی و پیاده‌سازی خواسته‌های آزمایش بعدی معطوف گردید. به طور خلاصه در این گزارش قصد داریم مراحل مختلف لازم جهت طراحی یک شبکه‌ی ارتباطی بی‌سیم را قدم به قدم شرح دهیم.

### طراحی شبکه‌ی ارتباطی بی‌سیم توسط NRF 2401

1. هر گره (Node) بتواند با گره‌های دیگر در ارتباط باشد.
2. هر گره تشخیص دهد که از کدام گره پیام دریافت کرده است.
3. در صورت عدم موفقیت در ارسال پیام، دوباره ارسال شود.

طراحی چنین سیستمی در واقع تفاوت‌چندانی با پروژه‌ی قبلی ندارد و تنها کفایت چند قسمت کوتاه را به آن کد اضافه کنیم.

پیام ما باید به گونه‌ای به مخاطب ارسال شود که 5 بایت اولیه‌ی آن حاوی آدرس ما به عنوان فرستنده باشد. علاوه بر این لازم است که آدرس فرستنده‌ی پیام با یک جداکننده‌ی مورد توافق – که در اینجا ":" – در نظر گرفته شده – از متن پیام جدا شود.

در کلاس درس، فرض را بر این گذاشتیم که سه برد موجود در کلاس، به ترتیب آدرس‌های زیر را اختیار کرده‌اند:

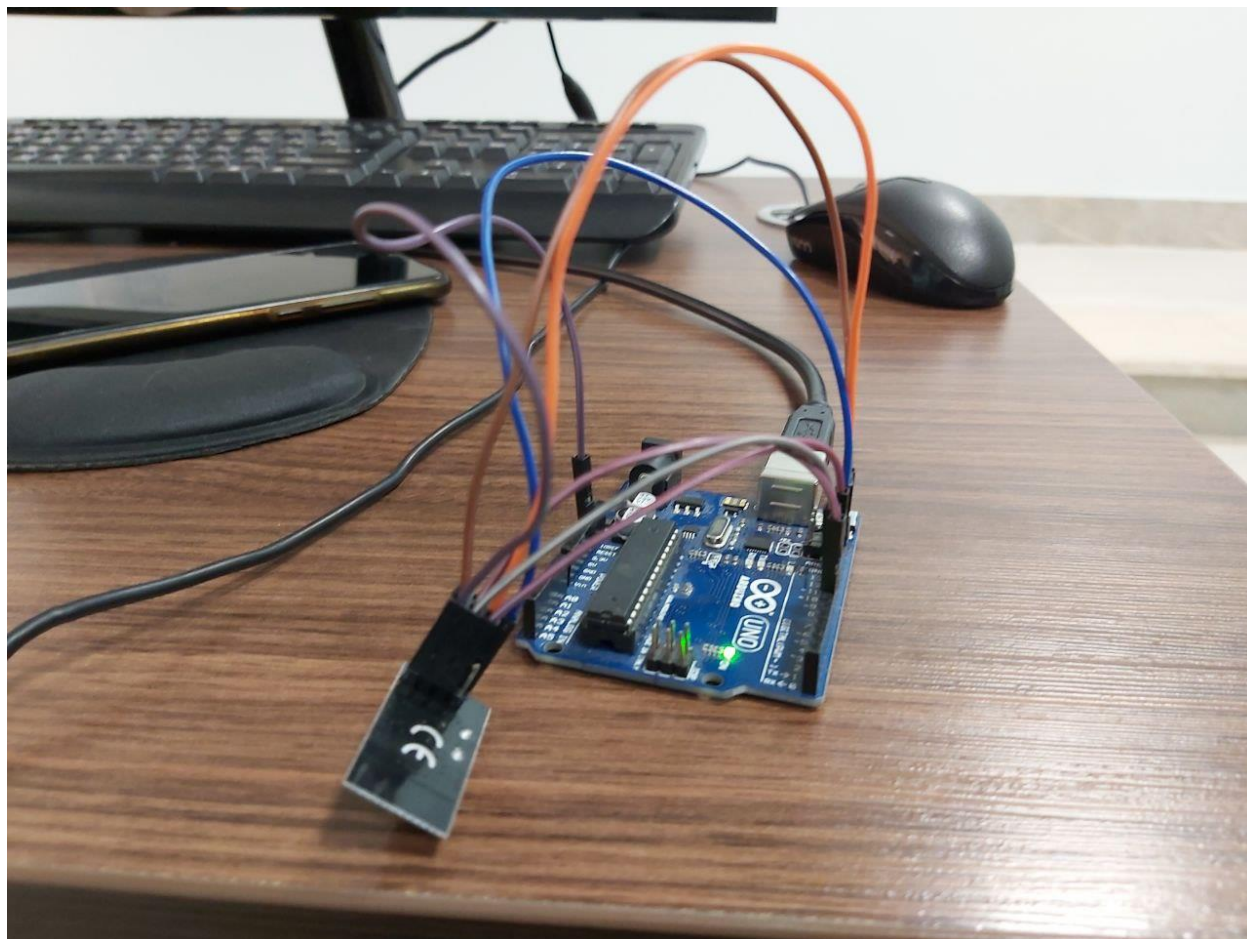
1Node

2Node

3Node

لازم به ذکر است که این آدرس‌ها می‌توانند هر عبارت دیگری را نیز شامل شوند و لزومی به استفاده‌ی دقیق از اسامی اعلام‌شده نیست و این آدرس‌ها با توافق جمع انتخاب شده‌اند.

تصویر زیر نحوه‌ی اتصال میان برد و ماژول گیرنده/فرستنده‌ی بی‌سیم را نشان می‌دهد:



در ادامه، قطعه‌کد لازم برای پیاده‌سازی این سیستم ارتباطی بی‌سیم را مشاهده می‌کنید:

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

```
#include "printf.h"
```

```
#include "RF24.h"
```

```
RF24 radio(7, 8); // using pin 7 for the CE pin, and pin 8 for the CSN pin
```

```
uint8_t address[6] = {"2Node" };
```

```
bool radioNumber = 1; // 0 uses address[0] to transmit, 1 uses address[1] to transmit
```

```
bool role = false; // true = TX role, false = RX role
```

```
void setup() {
```

```
    // initialize both serial ports:
```

```
    Serial.begin(9600);
```

```
    if (!radio.begin()) {
```

```
        Serial.println(F("radio hardware is not responding!!"));
```

```
        while (1) {} // hold in infinite loop
```

```
    }
```

```
    //int a = Serial.read();
```

```
    // set the TX address of the RX node into the TX pipe
```

```
    radio.openWritingPipe(address[radioNumber]); // always uses pipe 0
```

```
    // set the RX address of the TX node into a RX pipe
```

```
    radio.openReadingPipe(1, address[!radioNumber]); // using pipe 1
```

```
    radio.startListening();
```

```
}
```

```
String str;
```

```
char msg[32];
```

```
byte addr[5];
```

```
void loop() {
```

```
    // read from port 1, send to port 0:
```

```
    if (Serial.available()) {
```

```
        str = Serial.readString();
```

```
        Serial.println(str);
```

```
for(int i=0;i<5;i++)
{ addr[i]=str[i];}

for(int i=0;i<5;i++)
{ msg[i]=address[i];}

for(int i=5;i<str.length();i++)
msg[i]=str[i];
msg[str.length()]='\0';
Serial.println(msg);
radio.stopListening();
radio.openWritingPipe(addr);
bool report = radio.write(msg, sizeof(msg)); // transmit & save the report
if (report) Serial.print(F("Transmission successful! ")); // payload was delivered
else Serial.println(F("Transmission failed or timed out")); // payload was not
delivered
radio.startListening();
}
if (radio.available()) {
radio.read(msg,32);
Serial.println(msg);
}
}
```