Апаратно програмний комплекс дистанційного збору данних та керування об'єктами по протоколу Enhanced ShockBurst модуля NRF24L01+

Сергій Поліщук

17 січня 2020 р.

Зміст

1 Вступ

Наявність двох або більше плат Arduino, здатних спілкуватися між собою бездротово на відстані, відкриває безліч можливостей, таких як віддалений моніторинг даних датчиків, управління роботами, домашня автоматизація та інше. І коли справа зводиться до недорогих, але надійних двосторонніх радіочастотних рішень, є декілька варіантів:

- ZeegBee - ліцензований протокол, лідер ринку IoT комунікацій
- nRF24L01 - Nordic Semiconductor
- Bluetooth
- WiFi

Модуль трансивера nRF24L01 + (плюс) часто можна отримати в Інтернеті за менше одного долара США, що робить його одним з найдешевших варіантів радіо передачі даних, які є сьогодні на ринку. І найкраще, що ці модулі є надзвичайно крихітними, що дозволяє включати бездротовий інтерфейс майже в будь-який проект.

Усі приклади робочих проектів виконані за допомогою Visual Studio Code та PlatformIO Документаця написана за допомогою CLion (student license) разом з Markdown плагіном. Таблиці BOM конвертовані онлайн сервісом ConverCSV Усі схеми розроблено у EasyEDA під ліценцією MIT Друкована документація зібрана PanDoc

Внесення змін до документації або коду через систему контролю версій Git за адресою https://github.com/spolischook/ATP

Останню версію цієї друкованої документації у форматі pdf можна скачати за адресою https://github.com/spolischook/ATP/blob/master/doc/latex/main.pdf

2 Модуль NRF24L01 plus

2.1 Інтерфейси та характеристики

2.1.1 Радіочастота

Модуль приймача nRF24L01 + призначений для роботи в частотному діапазоні частот ISM (industrial, scientific and medical) 2,4 ГГц і використовує Гаусову модуляцію (GFSK) для передачі даних. Швидкість передачі даних конфігурується программою мікроконтроллера, та може бути 250kbps, 1Mbps або 2Mbps.

2.1.2 Споживання енергії

Робоча напруга модуля становить від 1,9 до 3,6 В, але гарна новина полягає в тому, що логічні входи мають 5-вольтову толерантність, тому ми можемо легко підключити його до Arduino або будь-якого логічного мікроконтролера 5В, не використовуючи жодного логічного перетворювача рівня.

Модуль підтримує програмовану вихідну потужність, а саме. 0 дБм, -6 дБм, -12 дБм або -18 дБм і споживає близько 12 мА під час передачі при 0 дБм, що навіть нижче, ніж у одного світлодіода. А найкраще, що він споживає 26 мкА в режимі очікування та 900 нА в режимі відключення живлення. Таким чином пристрої з радіоінтерфейсом nRF24L01, можуть працювати роками від батарейного живлення.

NORDIC SEMICONDUCTOR nRF24L01+ Preliminary Product Specification

5 Electrical specifications

Conditions: VDD = +3V, VSS = 0V, T_A = - 40°C to + 85°C

5.1 Power consumption

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
	Idle modes					
I_{VDD_PD}	Supply current in power down			900		nA
I_{VDD_ST1}	Supply current in standby-I mode	a		26		µA
I_{VDD_ST2}	Supply current in standby-II mode			320		µA
I_{VDD_SU}	Average current during 1.5ms crystal oscillator startup			400		µA
	Transmit					
I_{VDD_TX0}	Supply current @ 0dBm output power	b		11.3		mA
I_{VDD_TX6}	Supply current @ -6dBm output power	b		9.0		mA
I_{VDD_TX12}	Supply current @ -12dBm output power	b		7.5		mA
I_{VDD_TX18}	Supply current @ -18dBm output power	b		7.0		mA
I_{VDD_AVG}	Average Supply current @ -6dBm output power, ShockBurst™	c		0.12		mA
I_{VDD_TXS}	Average current during TX settling	d		8.0		mA
	Receive					
I_{VDD_2M}	Supply current 2Mbps			13.5		mA
I_{VDD_1M}	Supply current 1Mbps			13.1		mA
I_{VDD_250}	Supply current 250kbps			12.6		mA
I_{VDD_RXS}	Average current during RX settling	e		8.9		mA

a. This current is for a 12pF crystal. Current when using external clock is dependent on signal swing.
b. Antenna load impedance = 15Ω+j88Ω..
c. Antenna load impedance = 15Ω+j88Ω. Average data rate 10kbps and max. payload length packets.
d. Average current consumption during TX startup (130µs) and when changing mode from RX to TX (130µs).
e. Average current consumption during RX startup (130µs) and when changing mode from TX to RX (130µs).

Рис. 1: Струм у різних режимах

2.1.3 SPI інтерфейс

Модуль приймача nRF24L01+ підключається до мікроконтроллера через 4-контактний послідовний периферійний інтерфейс (SPI) з максимальною швидкістю передачі даних 10Мбіт/с. Всі параметри, такі як

- частотний канал (125 каналів, що вибираються),
- вихідна потужність (0 дБм, -6 дБм, -12 дБм або -18 дБм)
- швидкість передачі даних (250 кбіт / с, 1 Мбіт / с або 2 Мбіт / с),

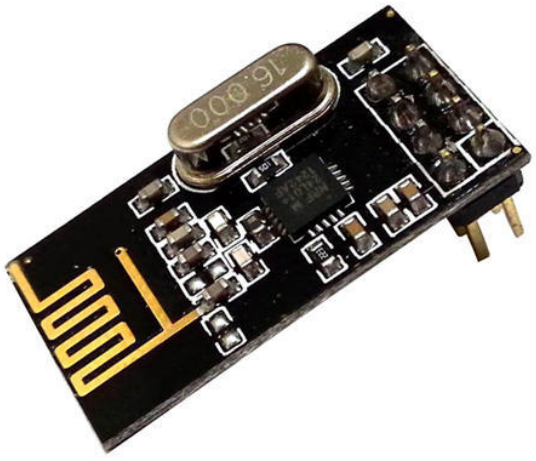
можуть бути налаштовані через інтерфейс SPI. Шина SPI використовує концепцію Master/Slave, в більшості поширених застосувань Arduino - це Master, а модуль приймача nRF24L01+ - Slave.

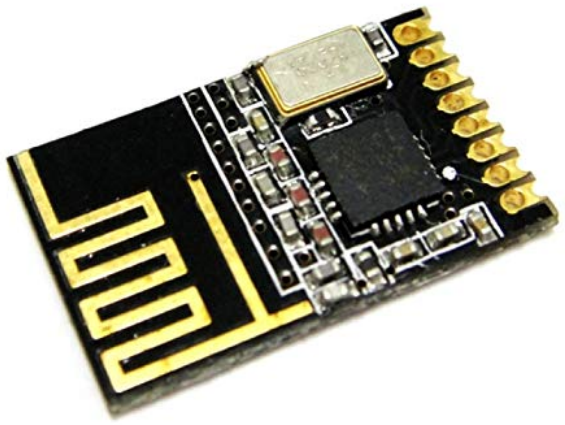
Таблиця характеристик модуля:

Характеристика	Значення
Діапазон частот	2.4 GHz, ISM діапазон
Максимальна швидкість передачі даних	2 Mb/s
Формат модуляції	Гаусова модуляція
Робоча напруга живлення	від 1.9 В до 3.6 В
Максимальний струм	13.5mA
Мінімальний струм (в режимі очікування)	26uA
Логічний вхід	3-5 В
Дальність передачі сигналу	до 800 м прямої видимості

2.1.4 Зовнішній вигляд модуля NRF24L01+

На основі чіпа nRF24L01 + доступні різноманітні модулі. Нижче представлені найпопулярніші версії.





Перші два варіанти з антеною на платі. Таким чином плата виходить більш компактною. Однак менша антена також означає меншу дальність передачі. Ця версія модуля здатна вести передачу до 100 метрів в зоні прямої видимості. У

приміщенні зі стінами, відстань передачі буде меншою.

Рис. 2: NRF24L01 з підсилювачем

Друга версія має коаксіальний роз'єм та duck-антену, але це не єдина відмінність. Справжня відмінність полягає в тому, що модуль поставляється зі спеціальним RFX2401C чіпом. Цей чіп разом з duck антеною допомагає модулю досягти значно більшого діапазону передачі, близько 1000 м.

2.1.5 Що таке PA LNA

PA означає посилювач потужності (Power Amplifier). Це збільшує потужність сигналу, що передається з чіпа nRF24L01+. Тоді як LNA розшифровується як підсилювач з низьким рівнем шуму (Low-Noise Amplifier). Функція LNA полягає в тому, щоб приймати надзвичайно слабкий і невизначений сигнал з антени (як правило, на порядок мікровольт або менше -100 дБм) і посилити його на більш корисний рівень (зазвичай приблизно від 0,5 до 1 В)

LNA та PA підключаються до антени через дуплекспер, який розділяє два сигнали і запобігає відносно потужному виходу PA від перевантаження чутливого входу LNA.

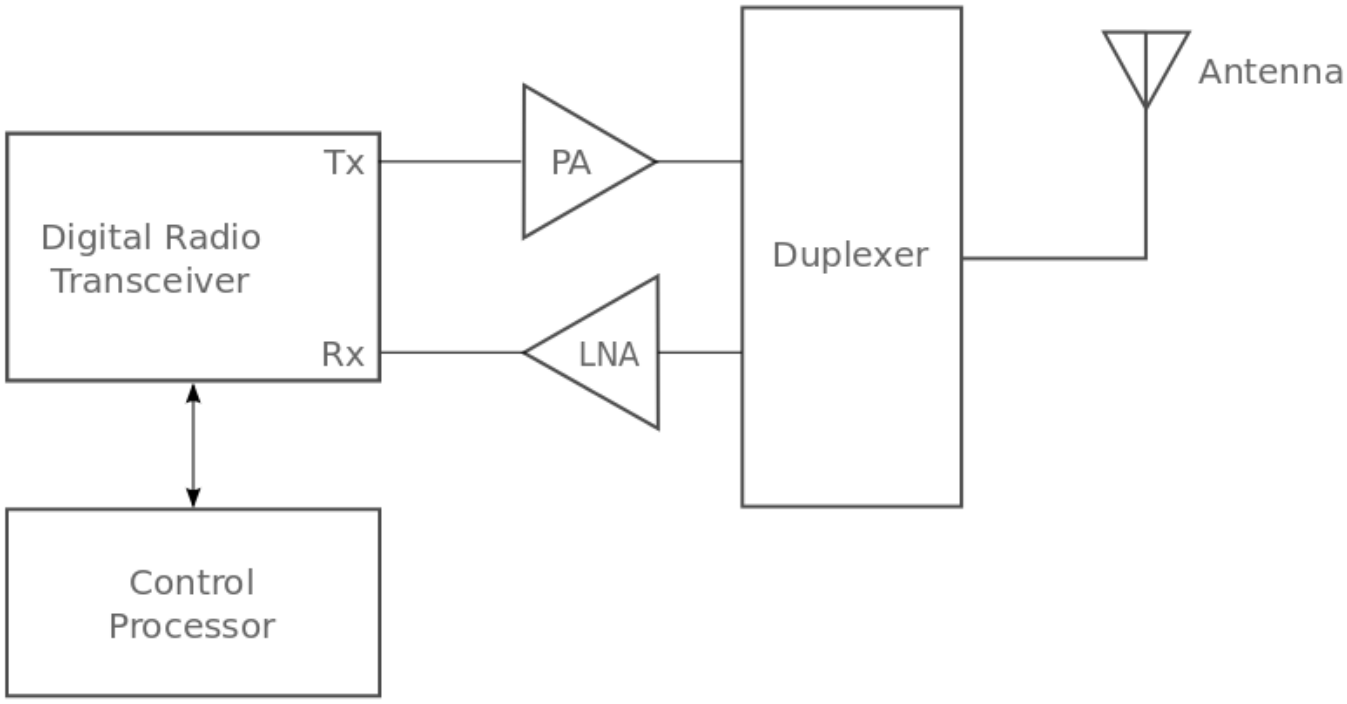


Рис. 3: Схема модуля NRF24L01 з PA LNA

Читайте про те, що таке PA LNA на StackOverflow - `https://electronics.stackexchange.com/questions/237267/what-is-a-pa-lna`

2.2 Принцип роботи nRF24L01+

2.2.1 Канал передачі данних (Pipes)

Модуль приймача nRF24L01 передає та приймає дані на певній частоті (Канал або Channel). Також для того, щоб два або більше модулів приймача могли спілкуватися один з одним, вони повинні знаходитися на одному каналі.
Цей канал може бути будь-якої частоти в діапазоні ISM 2,4 ГГц або, якщо бути точнішим, він може бути від 2,400 до 2,525 ГГц (2400 до 2525 МГц).

Кожен канал займає смугу пропускання менше 1 МГц.
Це дає нам 125 можливих каналів з інтервалом 1 МГц. Отже, модуль може використовувати 125 різних каналів, що дає можливість мати мережу з 125 незалежно працюючих мереж (6 Data Pipes +1 передавач) в одному місці.

2.2.2 nRF24L01+ MultiCeiver (TM) Network (Data Pipes)

NRF24L01 + надає функцію під назвою Multiceiver. Це абревіатура для декількох передавачів одного приймача. У якому кожен радіочастотний канал логічно розділений на 6 паралельних каналів даних, що називаються Data Pipes. Іншими словами, канал даних - це логічний канал у фізичному радіочастотному каналі.

Кожна передача даних має свою фізичну адресу (Data Pipe Address) і може бути налаштована. Це можна проілюструвати, як показано нижче.

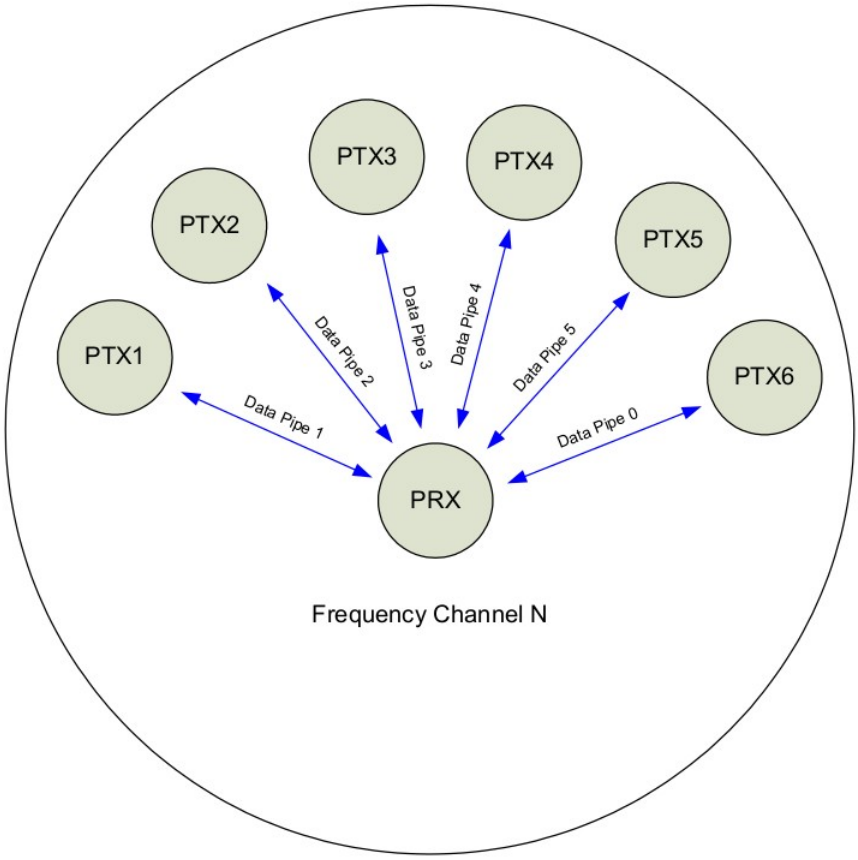


Рис. 4: Multiceiver data pipes

PRX нода, настроєна на пройом з 6 різних нод передавачів. Приймач може перестати слухати будь-який час і виконувати функцію передавача, але в такому разі він зможе передавати лише в один data pipe одночасно.

Слід зазаначити, що канали нумеруються від 0 до 5. Канал з 1 по 5 використовуються тільки для зчитування данних. Тоді як 0-й канал, буде використано як для передачі так і для прийому.

Піля завершення передачі по 0-му каналу і початку прийому (і навпаки), слід виставити відповідну адресу:

```
#define DIR_FIRST_SECOND 00001
#define DIR_SECOND_FIRST 00101

RF24 radio(9, 10); // CE, CSN
radio.openReadingPipe(0, DIR_SECOND_FIRST);
radio.startListening();
```

```
if (radio.available()) {
    char message[8];
    // зчитуємо 8 байт повідомлення у змінну message
    radio.read(message, sizeof(message));
}

radio.stopListening();
radio.openWritingPipe(DIR_FIRST_SECOND);
radio.write("Hello AKIT", 10);
```

Якщо немає необхідності читати данні з 6 каналів одночасно, рекомендуємо відкривати канали для зчитування данних починаючи з 1, а нульовий канал лише для передачі.

Щоб вести передачу по декількох каналах одночасно, слід відкрити відповідний канал перед передачою:

```
#define DIR_FIRST_SECOND 00001
#define DIR_FIRST_THIRD 00002
#define DIR_SECOND_FIRST 00101
#define DIR_THIRD_FIRST 00102
#define READ_FIRST_PIPE 1
#define READ_SECOND_PIPE 2
uint8_t READ_CURRENT_PIPE;

RF24 radio(9, 10); // CE, CSN
radio.openReadingPipe(READ_FIRST_PIPE, DIR_SECOND_FIRST);
radio.openReadingPipe(READ_SECOND_PIPE, DIR_THIRD_FIRST);
radio.startListening();

// При прослуховуванні декількох каналів, має значення номер каналу
// за яким доступні данні
if (radio.available(&READ_CURRENT_PIPE)) {
    if (READ_CURRENT_PIPE == READ_FIRST_PIPE) handleFirsPipe();
    if (READ_CURRENT_PIPE == READ_SECOND_PIPE) handleSecondPipe();
}

radio.stopListening();
radio.openWritingPipe(DIR_FIRST_SECOND);
radio.write("Hello AKIT", 10);
```

```
radio.openWritingPipe(DIR_FIRST_THIRD);
radio.write("AKIT is the best", 16);
```

2.2.3 Enhanced ShockBurst протокол link

Модуль приймача nRF24L01+ використовує структуру пакетів, відому як Enhanced ShockBurst. Ця проста структура пакетів розбита на 5 різних полів, що проілюстровано нижче.

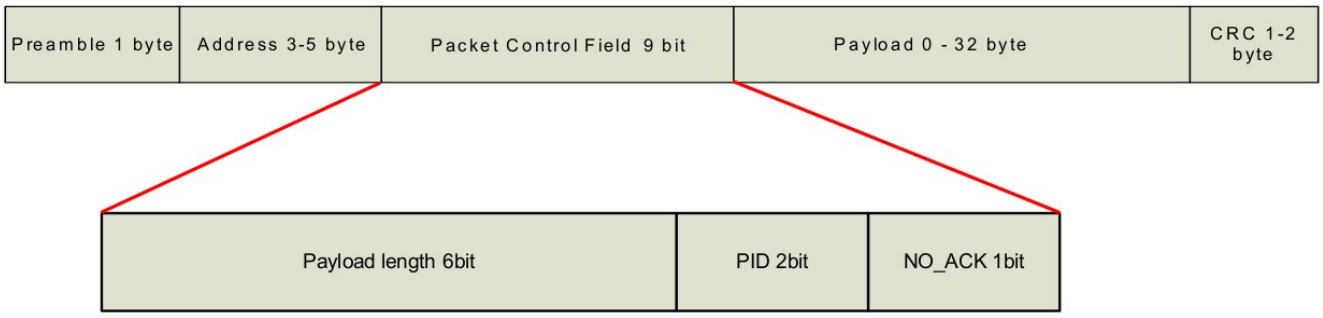


Рис. 5: Enhanced ShockBurst Protocol

Початкова структура ShockBurst складалася лише з полів преамбули, адреси, корисного навантаження та контрольної суми CRC. Enhanced ShockBurst забезпечив більшу функціональність для покращеної комунікації за допомогою нещодавно введеного Packet Control Field (PCF).

Ця нова структура покращила якість передачі одразу по кількох пунктах. По-перше, вона дозволяє змінювати довжину корисних навантажень Playload length, тобто в Playload можна записати від 1 до 32 байт.

По-друге, вона надає кожному відправленому пакету ідентифікатор пакета, що дозволяє пристрою, що приймає, визначати, чи є повідомлення новим, чи воно було повторно передано (і, таким чином, його можна ігнорувати).

Нарешті, і найголовніше, кожне повідомлення може вимагати надсилання підтвердження, коли воно отримане іншим пристроєм.

2.2.4 Транзакції (Automatic Packet Handling)

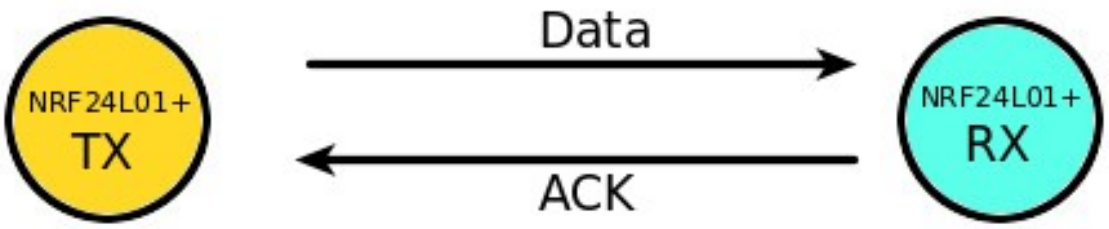


Рис. 6: Успішна транзакція

Успішна транзакція Першим є приклад позитивного сценарію.
Тут передавач починає зв'язок, надсилаючи пакет даних до приймача. Після передачі всього пакету він чекає (приблизно 130 мкс) для отримання пакета підтвердження (пакета ACK). Коли приймач отримує пакет, він надсилає ACK пакет передавачу. При отриманні пакету ACK передавач подає сигнал переривання (IRQ), щоб вказати, що дані успішно передані.

Рис. 7: Трансакція з втраченим пакетом даних

Трансакція з втраченим пакетом даних. Це негативний сценарій, коли необхідна повторна передача через втрату переданого пакету. Після передачі па-

кета передавач чекає отримання пакета ACK. Якщо передавач не отримує його протягом автоматичного затримки-затримки (ARD), пакет передається повторно. Коли повторно переданий пакет приймає приймач, передається пакет ACK, який, в свою чергу, генерує переривання у передавача.

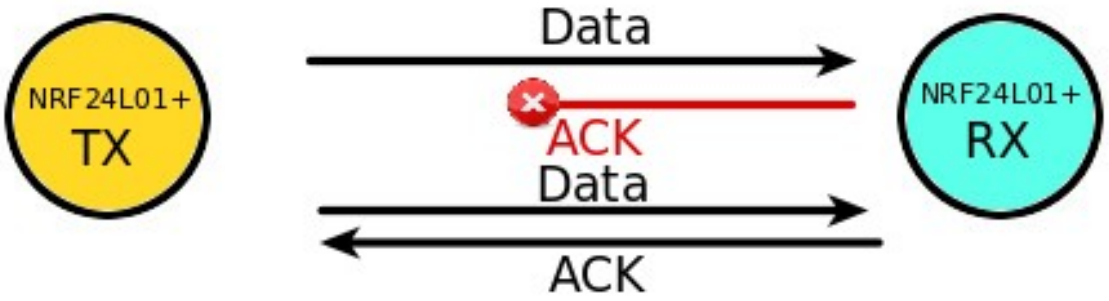


Рис. 8: Трансакція з втраченим підтвердженням

Трансакція з втраченим підтвердженням. Це знову негативний сценарій, коли необхідна повторна передача через втрату пакета ACK. Тут навіть якщо приймач отримує пакет з першої спроби, через втрату пакета ACK, передавач вважає, що одержувач взагалі не отримав пакет. Отже, після закінчення часу автоматичного повернення-затримки він повторно передає пакет. Тепер, коли одержувач отримує пакет, що містить той самий ідентифікатор, що і попередній, він відкидає його і знову надсилає пакет ACK.

Вся ця обробка здійснюється автоматично чіпом nRF24L01+ без участі мікроконтролера.

2.3 Arduino і nRF24L01+

2.3.1 nRF24L01+ Pinout

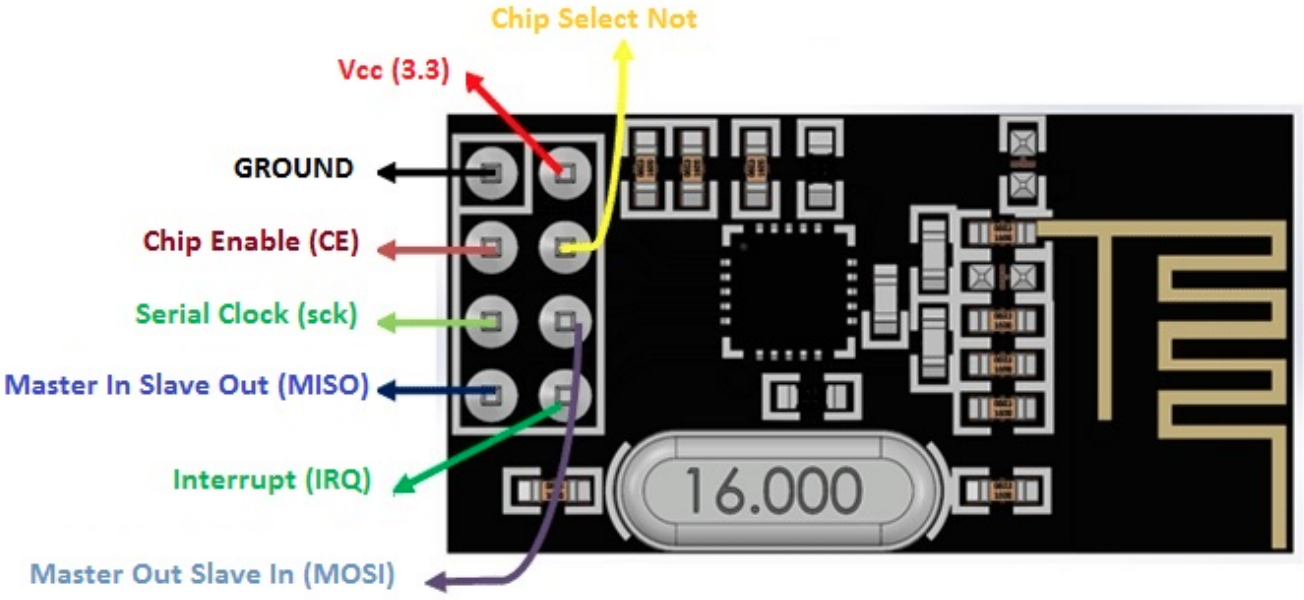


Рис. 9: nRF24L01+ Pinout

Окрім пінів живлення та стандартних пінів SPI інтерфейсу, наступні піни мають наступне призначення:

- Chip Enable (CE) - логічна одиниця на цей вхід зробить модуль активним. Використовується для контролю standby mode
- CSN (Chip Select Not) - LOW активний. Коли цей пін LOW, nRF24L01 починає прослуховувати дані на своєму SPI-порту для обробки даних.

2.3.2 Підключення NRF24L01 до Arduino Nano

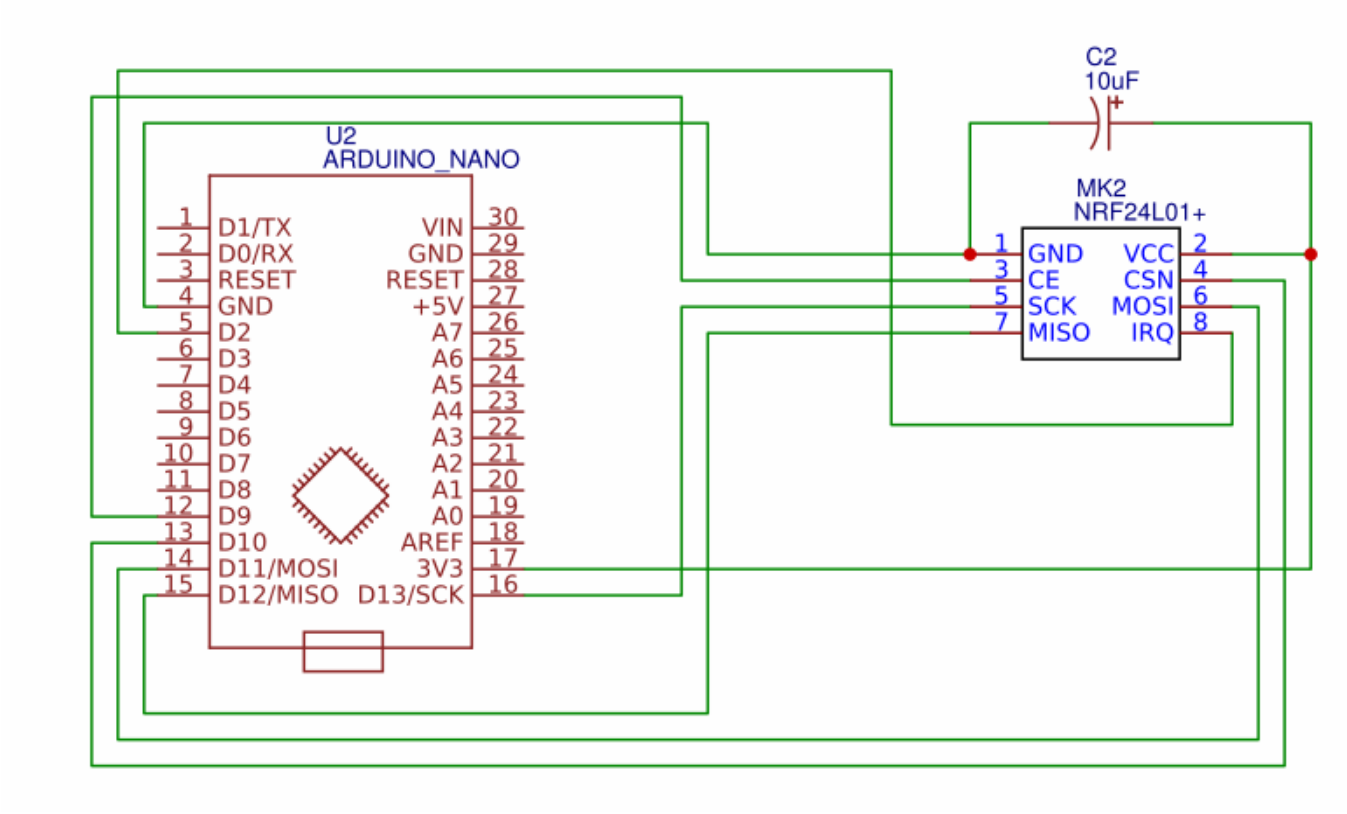


Рис. 10: NRF24L01 Arduino wiring

Піни SPI, а саме MOSI, MISO, SCK підкючаються до відповідних пінів мікроконтроллера.
Піни CE та CSN підключені до D9 і D10 відповідно
10 мкФ-конденсатор з'єднаний з VCC та GND має бути розміщений (фізично) якомога ближче до модуля. Конденсатор стабілізує живлення модуля.
Пін IRQ модуля підєднаний до D2, адже саме цей вхід може бути задіяний для преривань в Arduino Nano

2.3.3 Бібліотеки для роботи з nRF24L01+ Module

Взаємодія з модулем приймача nRF24L01+ напряму через SPI - непроста задача, але, на щастя, є ряд бібліотек які вже реалізували всю рутинну роботу.

https://github.com/nRF24/RF24 - реалізує базовий функціонал. Містить інтерфейси для роботи з Arduino, Raspberry Pi, ATtiny. Проста у використанні для новачків, але все ж пропонує багато для досвідчених користувачів.

https://github.com/nRF24/RF24Network - альтернатива ZigBee

https://github.com/nRF24/RF24Mesh - ця бібліотека має намір реалізувати mesh мережу для сенсорів, що дозволяє автоматичну та динамічну конфігурацію, яка може бути налаштована відповідно до багатьох сценаріїв.

2.3.4 Програмний код для роботи з nRF24L01+

На початку программи необхідно підключити бібліотеки для роботи з модулем nRF24L01+:

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

Після чого стане доступний класс RF24 Створення об'єкту, вимагає вказання Chip Enable (CE) та Chip Select (CS) пінів у якості аргументів. У всіх модулів це 9 та 10 цифровий пін:

```
RF24 radio(9, 10); // CE, CSN
```

Після створення об'єкту, необхідно встановити налаштування модуля:

```
// Включаємо модуль
radio.begin();
// Підсилення сигналу, може приймати 4 стани
// RF24_PA_MIN, RF24_PA_LOW, RF24_PA_HIGH and RF24_PA_MAX
radio.setPALevel(RF24_PA_MIN);
// Швидкість передачі данних, може бути:
// RF24_250KBPS for 250kbs, RF24_1MBPS for 1Mbps,
// RF24_2MBPS for 2Mbps
radio.setDataRate(RF24_250KBPS);
// Відкриваємо Pipe для передачі данних
radio.openWritingPipe(addresses[1]); // 00001
// Відкриваємо 1 Pipe для отримання данних
// Можна одночасно відкрити до 6 таких Pipes
radio.openReadingPipe(1, addresses[0]); // 00002
// Початок прийому данних з інших модулів
// Модуль може працювати лише в режимі прийому або передачі
radio.startListening();
// Встановлено максимальну к-сть спроб з максимайльним інтервалом
radio.setRetries(15, 15);
// Вибираємо канал передачі данних
radio.setChannel(120);
```

Таким чином модуль готовий для роботи. Для того щоб перевірити чи доступні данні для читання з модуля користуємось функцією available:

```
uint8_t dataPipe;
if (radio.available(&dataPipe)) {
radio.read(data, sizeof(data));
}
```

Функція приймає номер data pipe для читання, якщо в программі використовується лише один data Pipe, аргумент може бути опущений.

Щоб передати данні, спочатку переключаємось в режим передачі, і не забуваємо після write повернути режим читання:

```
radio.stopListening();
radio.write("Hello AKIT", 10);
radio.startListening();
```

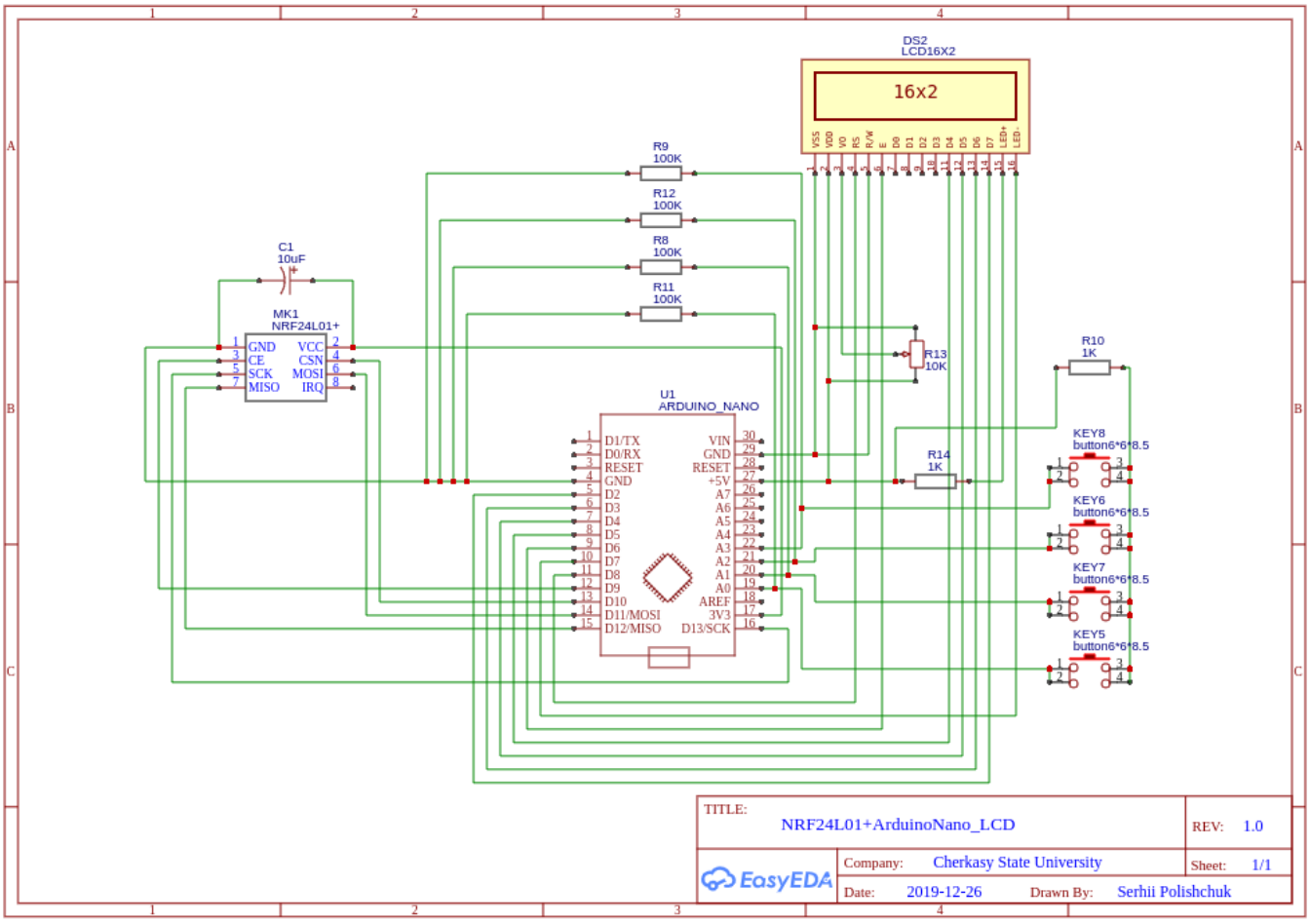
Т.я. писати модуль може тільки в один Pipe, то данні будуть передані по раніше вибраній функцією openWritingPipe адресі.

3 Модулі

3.1 ArduinoNano NRF24L01+ LCD

EasyEDA project - https://easyeda.com/spolischook/nrf24l01-arduinonano-lcd

3.1.1 Схема електрична принципова

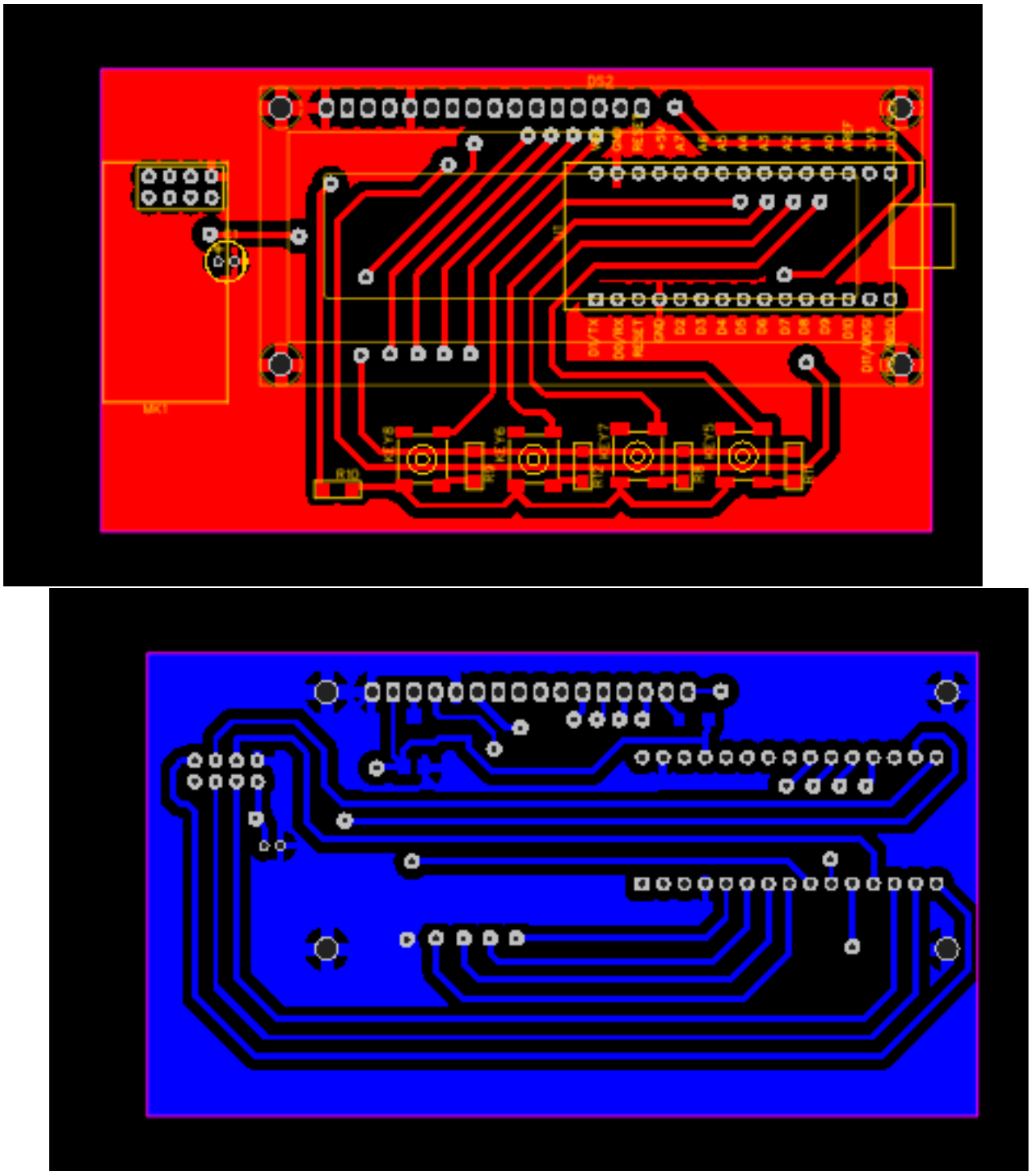


3.1.2 Список елементів модуля (BOM, bill of materials)

ID	Name	Designator	Quantity
1	ARDUINO_NANO	U1	1
2	NRF24L01+	MK1	1
3	10K	R13	1
4	10uF	C1	1
5	LCD16X2	DS2	1
6	button6*6*8.5	KEY5,KEY6,KEY7,KEY8	4

ID	Name	Designator	Quantity
7	100K	R8,R9,R11,R12	4
8	1K	R10,R14	2

3.1.3 Друкована плата (PCB)



3.1.4 Програмне забезпечення

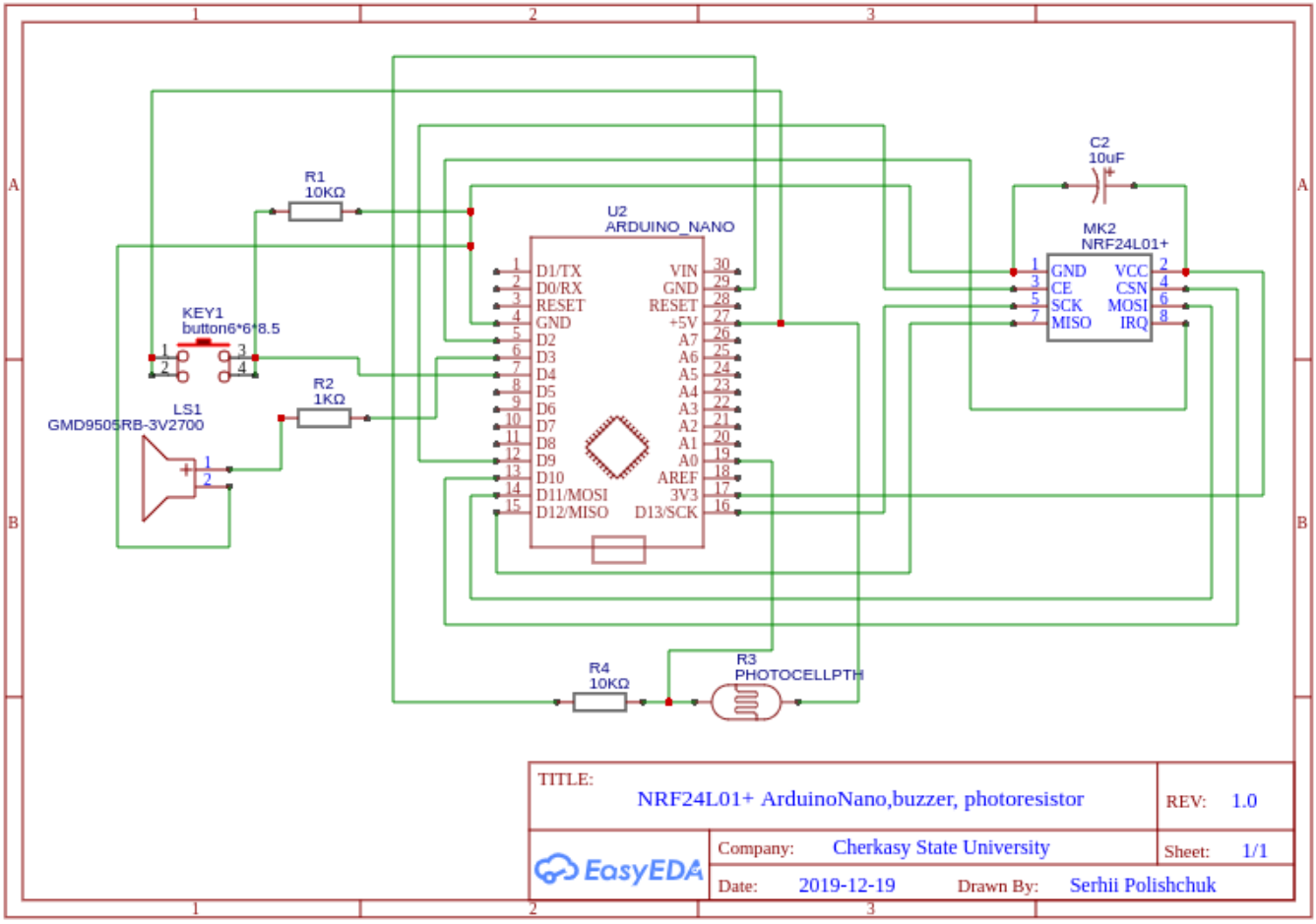
3.1.5 Дефекти

В схемі, а точніше в її фізичній реалізації, є один суттєвий недолік. Функція `radio.write()` повертає 0, незалежно від фактичної доставки повідомлення до приймача. Таким чином пакет `Ack` ігнорується.
Дефект зберігається при зміні модуля `NRF24L01+`

3.2 Модуль контролю освітлення + сирена

EasyEDA project - https://easyeda.com/spolischook/nrf24l01-arduino-nano-zoom-photoresis

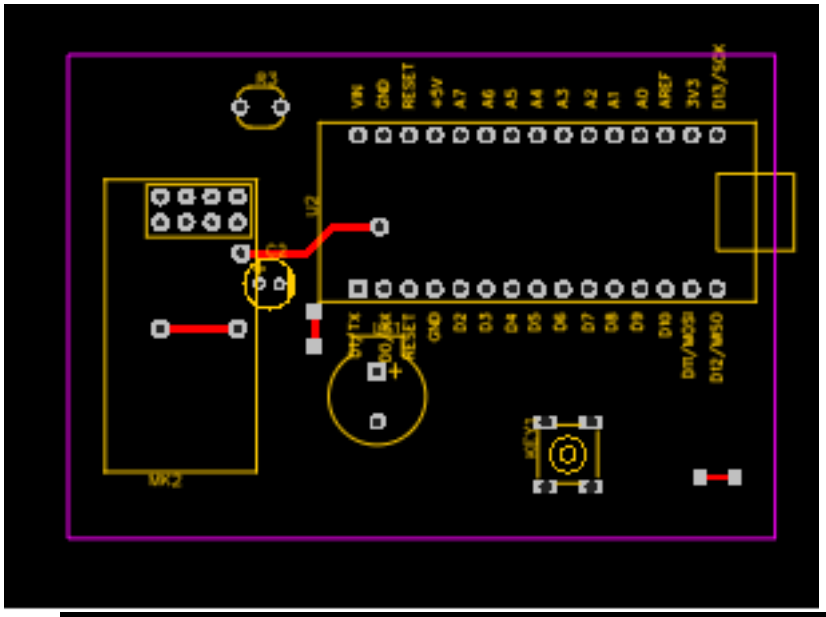
3.2.1 Схема електрична принципова

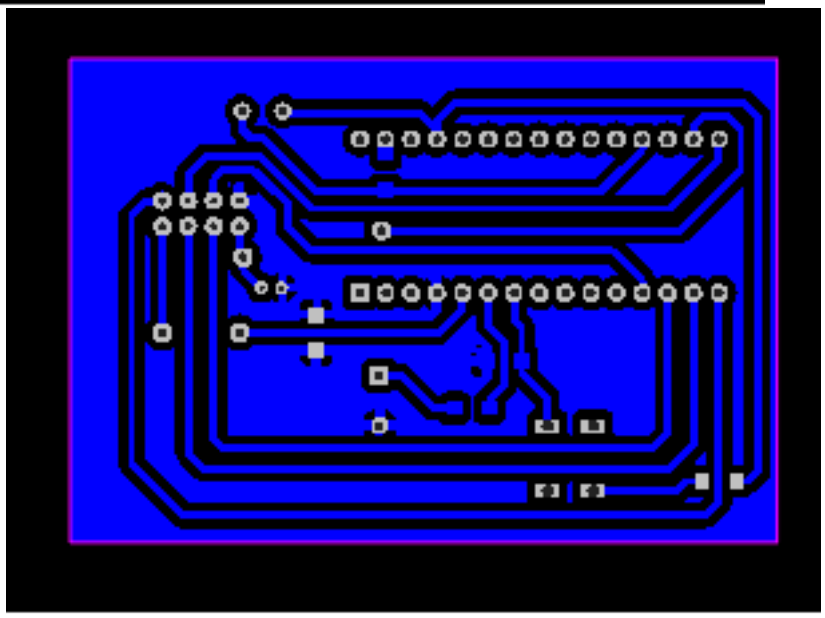


3.2.2 Список елементів модуля (BOM, bill of materials)

ID	Name	Designator	Quantity
1	GMD9505RB-3V2700	LS1	1
2	ARDUINO_NANO	U2	1
3	PHOTOCELLPTH	R3	1
4	NRF24L01+	MK2	1
5	10uF	C2	1
6	button6*6*8.5	KEY1	1
7	10KΩ	R1,R4	2
8	1KΩ	R2	1

3.2.3 Друкована плата (PCB)



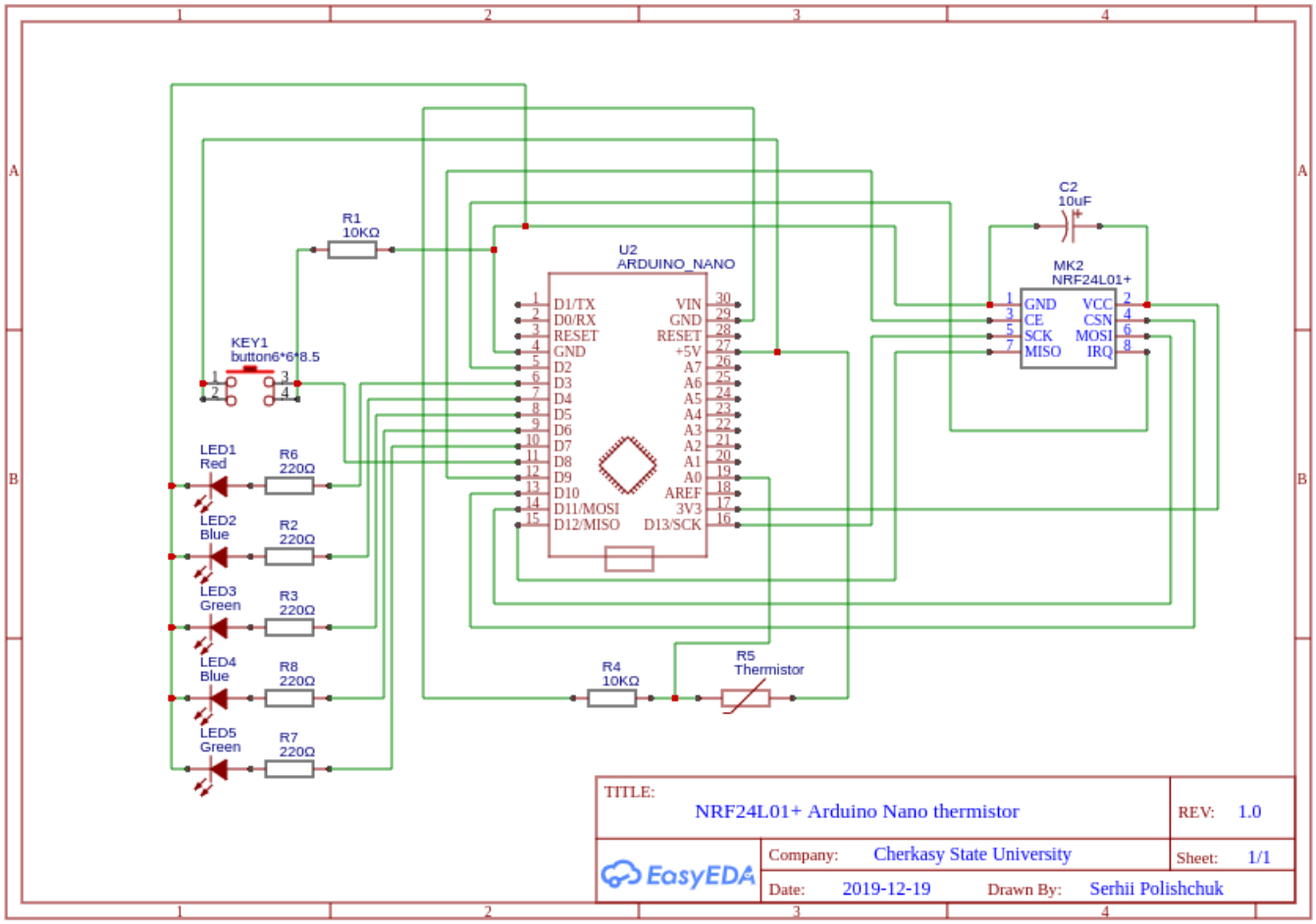


3.2.4 Програмне забезпечення

3.3 Модуль контролю температури

EasyEDA project - https://easyeda.com/spolischook/nrf24l01-arduinonano_termistor

3.3.1 Схема електрична принципова

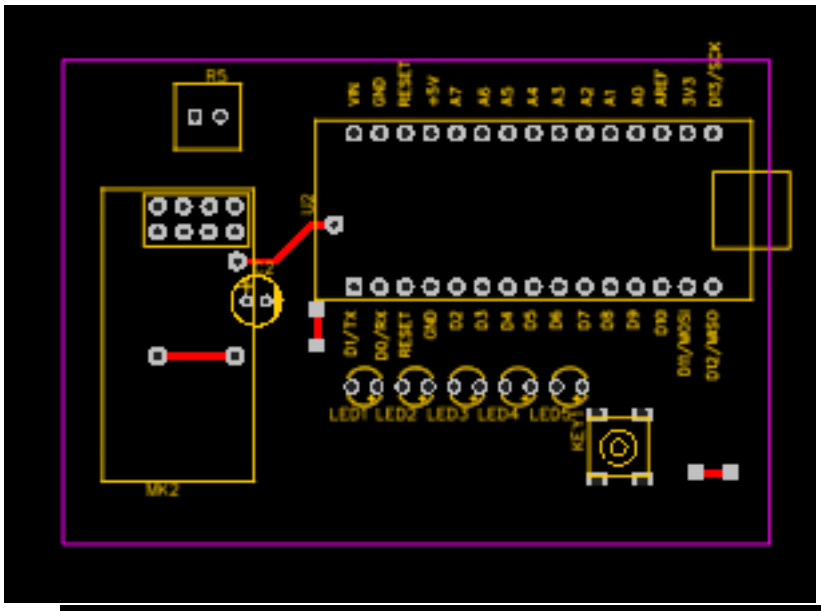


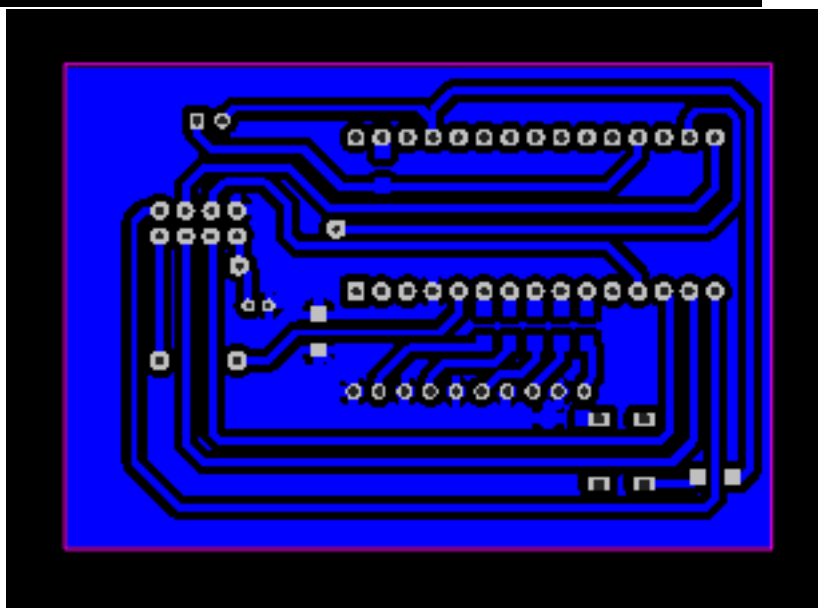
3.3.2 Список елементів модуля (BOM, bill of materials)

ID	Name	Designator	Quantity
1	Thermistor	R5	1
2	ARDUINO_NANO	U2	1
3	NRF24L01+	MK2	1
4	Green	LED3,LED5	2
5	Blue	LED2,LED4	2
6	Red	LED1	1
7	10uF	C2	1
8	button6*6*8.5	KEY1	1

ID	Name	Designator	Quantity
9	10KΩ	R4,R1	2
10	220Ω	R6,R2,R3,R7,R8	5

3.3.3 Друкована плата (PCB)





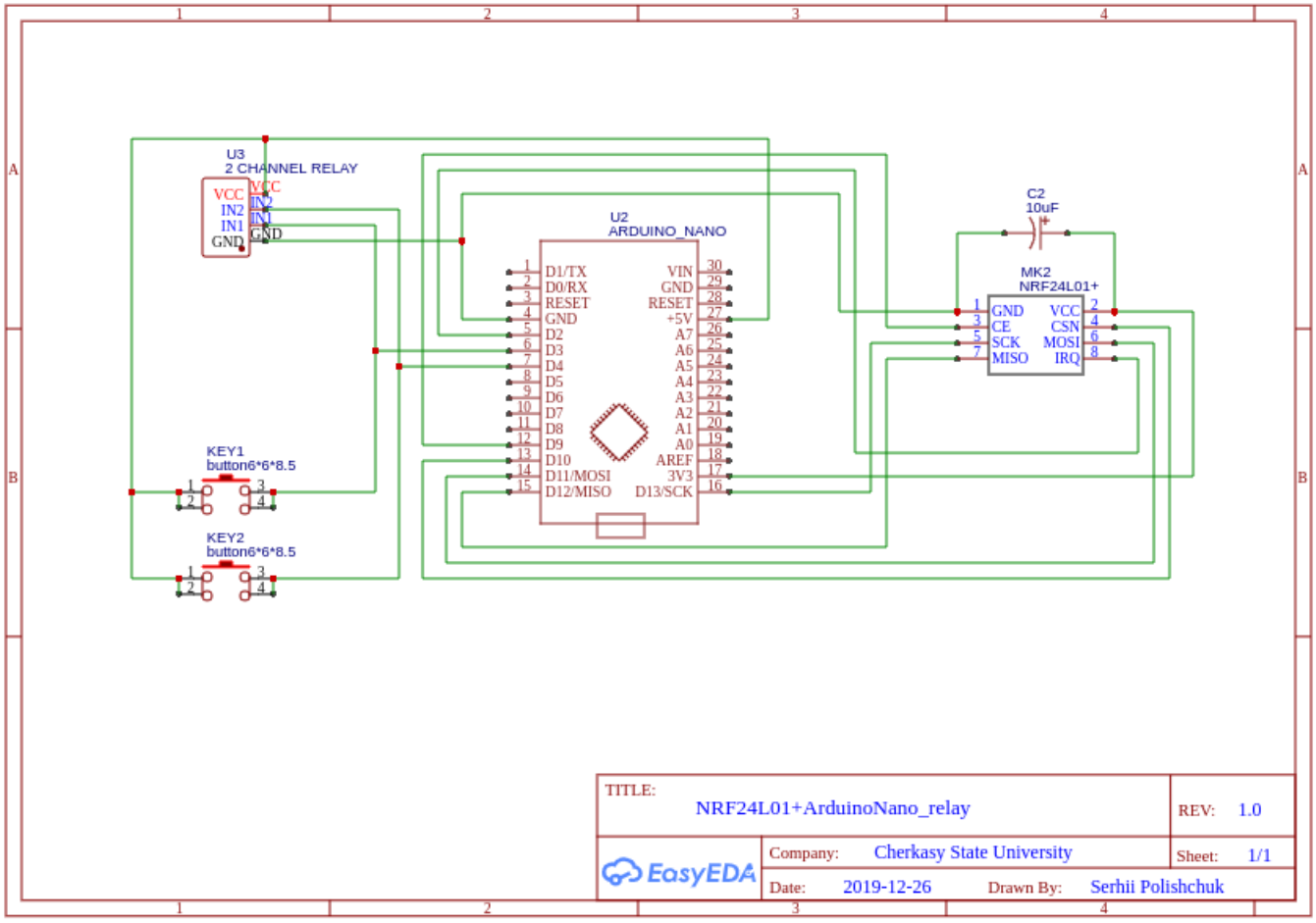
3.3.4 Програмне забезпечення

Программа для перевірки модуля.

3.4 Виконавчий модуль з двома релле

EasyEDA project - https://easyeda.com/spolischook/nrf24l01-arduinonano_termistor

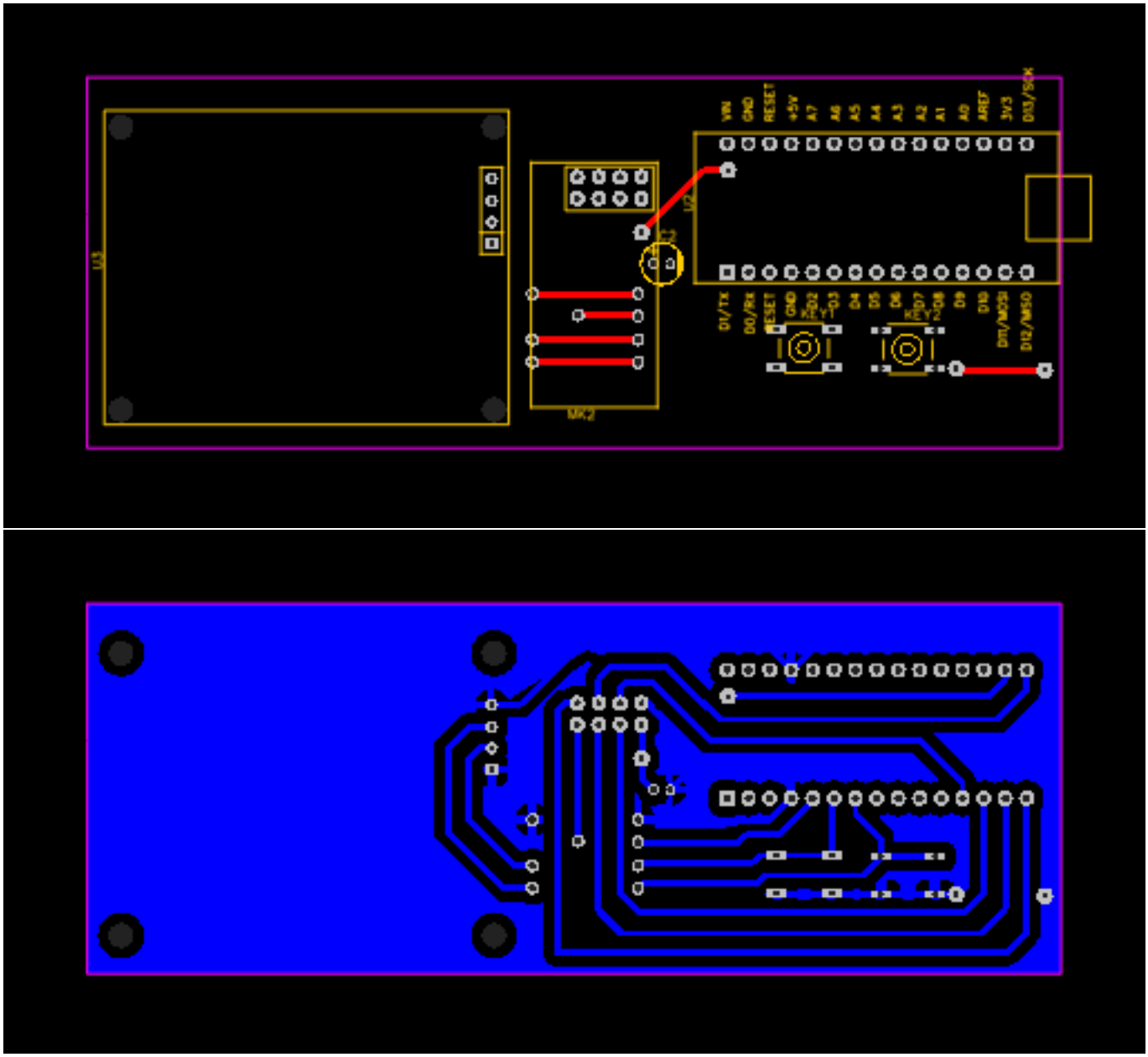
3.4.1 Схема електрична принципова



3.4.2 Список елементів модуля (BOM, bill of materials)

ID	Name	Designator	Quantity
1	2 CHANNEL RELAY	U3	1
2	ARDUINO_NANO	U2	1
3	NRF24L01+	MK2	1
4	10uF	C2	1
5	button6*6*8.5	KEY1,KEY2	2

3.4.3 Друкована плата (PCB)



3.4.4 Програмне забезпечення

Программа для перевірки модуля.

4 Проект комунікації модулів