

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ВИСОКИХ ТЕХНОЛОГІЙ

В.о. завідувача кафедри квантової радіофізики

доц. Сергій Олександрович КОЛІСНОВ

Протокол № ____ засідання кафедри

від “ ____ ” _____ 20 ____ р.

Модель розумного житла з адаптивним
енергоспоживанням

Випускна кваліфікаційна робота бакалавра

студента спеціальності

171 Електроніка

ОП «Електроніка (високі технології)»

Погорілець Евеліна Юріївна

Науковий керівник

к. ф. м. н., доцент кафедри

нанофізики конденсованих

середовищ

к.ф.-м.н. **Іванов Іван Іванович**

Оцінка захисту роботи

Київ 2024

АНОТАЦІЯ

Погорілець Е.Ю. Модель розумного житла з адаптивним енергоспоживанням – Випускна кваліфікаційна робота бакалавра за спеціальністю 171 Електроніка.

У роботі зроблено огляд на різні системи розумного будинку. Також розроблену нову модель системи розумного житла, яка є економічно вигідною. Ця система розроблена для однокімнатної квартири, при встановленні цієї системи ви зможете убезпечити квартиру від пожежі, протікань, зекономити кошти, також в системі є система охорони для вашого житла.

Ключові слова: розумний дім; модель розумного житла.

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. Огляд Літератури.....	6
1.1 Аналіз сучасного стану та тенденцій розвитку систем "розумний дім"	6
1.2 Функціональні можливості систем "розумний дім"	9
1.3 Класифікація систем "розумний дім"	11
1.4 Огляд апаратних та програмних засобів для реалізації систем "розумний дім"....	13
1.5 Програмовані плати, які були використані для розробки	19
РОЗДІЛ 2. Практична Частина.....	23
2.1 Функціональні можливості розробленої системи "розумний дім".....	23
2.2 Апаратне забезпечення системи "розумний дім".....	25
2.3 Програмне забезпечення системи "розумний дім".....	27
Охорона — режим для розумного будинку.....	27
Програма розумного дому	27
2.5 Тестування та налагодження системи "розумний дім".....	30
2.6 Розрахунок економічної вигоди від впровадження системи "розумний дім" для однокімнатної квартири.....	52
ВИСНОВОК.....	54
СПИСОК ЛІТЕРАТУРИ.....	56
ДОДАТОК А: Схеми електричних з'єднань.....	58
ДОДАТОК Б: План розумного дому розроблений на плані однокімнатної квартири. ...	60

ВСТУП

Актуальність теми та обґрунтування вибору теми дослідження

В дипломі розглядається розумний дім адаптований під трьохзоновий лічильник. Це варіант стандартних налаштувань розумного будинку, економічно вигідних. Зарахунок того, що прилади, які потребують багато електроенергії, тратять її в час, коли тариф дешевше, то оплата за комунальні послуги буде нижче. Я обрала цю тему для реалізації, бо ще не бачила жодних готових рішень економічно вигідних розумних будинків в Україні, адаптованих під українські тарифи.

Метою дипломної роботи є розробка макету економічного розумного дому з системою охорони для однокімнатної квартири.

Для досягнення мети дослідження поставлено такі завдання:

- Провести аналіз існуючих рішень для розумних будинків та визначити їхні переваги та недоліки.
- Розробити архітектуру економічного розумного дому з системою охорони.
- Вибрати та обґрунтувати компоненти для макета економічного розумного дому.
- Розробити програмне забезпечення для керування макетом.
- Зібрати та протестувати макет економічного розумного дому.
- Провести економічний аналіз розробленого макета.

Об'єктом дослідження є системи розумного дому.

Предметом дослідження є розробка макета економічного розумного дому з системою охорони для однокімнатної квартири.

Для вирішення поставлених завдань будуть використовуватись такі методи дослідження:

- Аналіз літературних джерел;

- Порівняльний аналіз;
- Моделювання;
- Програмування.

Наукова новизна дослідження полягає в розробці макета економічного розумного дому з системою охорони для однокімнатної квартири.

Практична значущість дослідження полягає в тому, що розроблений макет економічного розумного дому може бути використаний для:

- Демонстрації можливостей економічних розумних будинків;
- Розробка нових рішень для розумних будинків.

Результати дослідження будуть апробовані на кафедрі.

РОЗДІЛ 1.

Огляд Літератури

1.1 Аналіз сучасного стану та тенденцій розвитку систем "розумний дім"

Сучасний стан:

- Ринок систем "розумний дім" стрімко зростає. За прогнозами Statista, до 2025 року він сягне 151,4 мільярда доларів США. Цьому зростанню сприяють такі фактори, як:
 - Зниження вартості компонентів та обладнання
 - Зростання доступності високошвидкісного Інтернету
 - Зростаюча популярність економії енергії та екологічності
 - Зростаюча потреба в комфорті та безпеці
- Існує широкий спектр систем "розумний дім" на ринку. Ці системи можуть варіюватися за ціною, функціональністю та складністю. Деякі з найпопулярніших систем включають:
 - Amazon Alexa
 - Google Assistant
 - Apple HomeKit
 - Samsung SmartThings
- Системи "розумний дім" можуть використовуватися для автоматизації різних завдань у домі, таких як:
 - Вмикання та вимикання світла
 - Регулювання температури
 - Відкриття та закриття дверей та вікон
 - Управління побутовою технікою

- Моніторинг безпеки

Тенденції розвитку:

- Зростання популярності голосового керування. Все більше людей використовують голосові команди для керування своїми системами "розумний дім".
- Інтеграція з іншими системами. Системи "розумний дім" все частіше інтегруються з іншими системами, такими як системами безпеки, системами опалення та вентиляції, та системами моніторингу енергоспоживання.
- Зростання значення штучного інтелекту. Штучний інтелект використовується для персоналізації досвіду користувачів та для автоматизації завдань.
- Зростання уваги до кібербезпеки. Зростання популярності систем "розумний дім" також призводить до зростання ризиків кібербезпеки.

В Україні:

- Ринок систем "розумний дім" в Україні знаходиться на стадії розвитку.
- Проте, він стрімко зростає. За прогнозами, до 2025 року він сягне 1 мільярда доларів США.
- Цьому зростанню сприяють такі фактори:
 - Зниження вартості компонентів та обладнання
 - Зростання доступності високошвидкісного Інтернету
 - Зростаюча популярність економії енергії та екологічності
 - Зростаюча потреба в комфорті та безпеці
- Деякі з найпопулярніших систем "розумний дім" в Україні:

- Ajax
- Alexa
- Nedis SmartLife
- Xiaomi Mi Home
- Системи "розумний дім" можуть використовуватися для автоматизації різних завдань у домі, таких як:
 - Вмикання та вимикання світла
 - Регулювання температури
 - Відкриття та закриття дверей та вікон
 - Управління побутовою технікою
 - Моніторинг безпеки

Висновок:

Ринок систем "розумний дім" стрімко зростає як у світі, так і в Україні. Ці системи стають все більш доступними та функціональними.

1.2 Функціональні можливості систем "розумний дім"

Системи "розумний дім" пропонують широкий спектр функцій, які можна узагальнити за трьома основними напрямками:

1. Підвищення комфорту:

- Клімат-контроль: Автоматичне регулювання температури, вологості та вентиляції для створення сприятливого мікроклімату в кожній кімнаті.
- Освітлення: Управління освітленням за допомогою сценаріїв, датчиків руху та освітлення, голосового керування.
- Мультимедіа: Централізоване керування аудіо- та відеосистемами, створення фонові музики, інтеграція з онлайн-сервісами.
- Побутові прилади: Віддалене керування побутовою технікою, пральною машиною, посудомийною машиною, кондиціонером тощо.
- Система безпеки: Датчики диму, витоку газу, затоплення, охоронна сигналізація, відеоспостереження.

2. Економія ресурсів:

- Енергозбереження: Оптимізація використання електроенергії, контроль за витратами, автоматичне відключення освітлення та приладів при виході з кімнати.
- Водопостачання: Контроль витоку води, автоматичне регулювання поливу газону.
- Економія коштів: Регулювання роботи приладів для роботи їх у час дешевшого тарифу та економія за рахунок автоматичного

вимкнення, коли немає потреби. Можливо при користуванні системою, тільки при підключенні до загальних ресурсів міста.

3. Забезпечення безпеки:

- Система безпеки: Датчики диму, витоку газу, затоплення, охоронна сигналізація, відеоспостереження.
- Віддалений контроль: Можливість моніторити стан будинку та керувати системами з будь-якого місця за допомогою смартфона або планшета.
- Імітація присутності: Створення імітації присутності людей в будинку під час вашої відсутності, що може відлякувати зловмисників.

Важливо зазначити, що функціональні можливості систем "розумний дім" можуть значно відрізнятись залежно від конкретної реалізації та набору підключених пристроїв.

Ось деякі з додаткових можливостей, які можуть пропонувати системи "розумний дім":

- Голосове керування системою за допомогою розумних колонок.
- Створення сценаріїв для автоматизації рутинних дій (наприклад, "прихід додому", "нічний режим", "від'їзд з дому").
- Інтеграція з іншими системами розумного міста.
- Персоналізація налаштувань відповідно до потреб та вподобань користувачів.

1.3 Класифікація систем "розумний дім"

Існує декілька способів класифікації систем "розумний дім":

За рівнем складності:

- Прості системи: зазвичай складаються з декількох пристроїв, які можна управляти за допомогою смартфона або планшета.
- Складні системи: пропонують більш широкий спектр функцій та можливостей, а також можуть інтегруватися з іншими системами в будинку, такими як система безпеки або система опалення та кондиціонування.

За типом підключення:

- Дротові системи: використовують дротове з'єднання для зв'язку між пристроями.
- Бездротові системи: використовують бездротове з'єднання, таке як Wi-Fi або Zigbee, для зв'язку між пристроями.

За способом керування:

- Системи з централізованим керуванням: мають центральний контролер, який управляє всіма пристроями в системі.
- Децентралізовані системи: не мають центрального контролера, а кожен пристрій управляється самостійно.

За ціною:

- Бюджетні системи: доступні за ціною та пропонують базові функції.
- Системи середнього класу: пропонують більш широкий спектр функцій та можливостей.

- Преміальні системи: пропонують найширший спектр функцій та можливостей, а також найвищу якість.

Ось кілька прикладів популярних систем "розумний дім":

- Amazon Alexa
- Google Home
- Apple HomeKit
- Samsung SmartThings
- Fibaro

При виборі системи "розумний дім" важливо врахувати:

- Ваші потреби та бюджет
- Розмір вашого будинку
- Тип підключення, яке ви хочете використовувати
- Функції, які вам потрібні

Ви також можете проконсультуватися з фахівцем, щоб отримати допомогу з вибором системи "розумний дім", яка відповідає вашим потребам.

1.4 Огляд апаратних та програмних засобів для реалізації систем "розумний дім".

Аїах. Виробник: Україна. Відповідно, за замовчуванням підтримується український інтерфейс. Дана система автоматизації будинку повною мірою справляється відразу з двома важливими завданнями: забезпечує комфорт і зручність в управлінні життєзабезпеченням приміщення; гарантує безпеку житла в повній мірі, контролюючи межі об'єкту на предмет злону, а також електричну, пожежну, газову та інші можливі загрози для дому. Обладнання «Розумний будинок» Аїах працює на надійно зашифрованою і захищеною двостороннього радіозв'язку Jeweller власної розробки, має повну автономність від електромережі завдяки резервного джерела живлення - хабу, характеризується стильним дизайном всіх своїх пристроїв.

Переваги:

- простий монтаж;
- бездротовий канал зв'язку між системними елементами;
- велика зона дії сигналу (до 2000 м);
- наявність захисту від зняття будь-якого з датчиків (бампера);
- можливий доступ інших користувачів (повний або частковий);
- автономна робота хаба від акумулятора (до 16 годин);
- Wi-Fi і GSM-зв'язок;
- різноманітність способів інформування користувача (дзвінок, SMS, Push-повідомлення);

- розумна розетка показує витрата електроенергії (з урахуванням підключених приладів), автоматично відключається при перепадах напруги;
- установка за QR-кодом і управління за допомогою смартфона (iOS, Android);
- підключення до 100 пристроїв;
- наявність тривожної кнопки на пульті;
- невисока вартість комплекту (від 200\$).

Недоліки:

- функціонування тільки з роботою центрального контролера (Hub), тобто відсутність автономності датчиків;
- немає власної камери відеоспостереження (але є можливість підключення стороннього обладнання);
- управління тільки через телефон, хоча це знімає необхідність встановлювати будь-які додаткові програми на ПК.



Рис. 1.1 прилади системи охорони від компанії AJAX.

BroadLink. Виробник: Китай. За замовчуванням немає українського інтерфейсу, при необхідності його можна знайти. Обладнання «Розумний будинок» BroadLink являє собою комплект сучасних цифрових пристроїв, створених для раціонального управління побутовою технікою, а також освітлювальної, енергетичної, охоронної та іншими системами в будинку. Кожен елемент такого комплексу може працювати як самостійно, так і взаємодіяти один з одним.

Переваги:

- швидко встановлюється, підключається і налаштовується;
- має широкий асортимент датчиків (вологості, температури, освітлення, шуму, забруднення повітря);
- можна легко додавати і прибирати різні пристрої;
- функціонує без центрального хаба (автономна робота датчиків);
- бездротова взаємодія пристроїв між собою;
- є своя камера відеоспостереження;
- контролюється по Wi-Fi через Інтернет з будь-якої точки планети;
- демократична вартість обладнання (від 200\$).

Недоліки:

- невелика дальність дії сигналу (до 50 м);
- відсутність резервного живлення хаба;

- пульт працює тільки на прийом сигналів. [3]



Рис 1.2 головний блок управління пристроями розумного житла від компанії BroadLink.

Tuya Smart – це розумна екосистема для дому, вона включає широкий спектр пристроїв, таких як розумні розетки, вимикачі, датчики руху, розумні лампи, розумні замки і багато іншого. Однією з головних переваг Tuya Smart є її простота використання та універсальність. Ви можете легко керувати пристроями, використовуючи програму на вашому смартфоні, яка доступна як для iOS, так і для Android. Ви також можете інтегрувати Tuya Smart в інші екосистеми управління розумним будинком, такі як Amazon Alexa, Google Assistant та IFTTT. [4]

Tuya Inc — це китайська платформа штучного інтелекту, заснована в 2014 році. Компанія надає хмарну платформу для розробки та керування.

На міжнародному рівні Tuya співпрацює з такими компаніями, як Schneider Electric, Lenovo та Philips. Він є членом ради директорів Альянсу стандартів підключення і зобов'язався підтримувати стандарт підключення Matter. [5]

Переваги

- Широкий вибір пристроїв.
- Просте підключення.
- Універсальність. Всі пристрої з чіпом TuYa підходять для фірмового застосування, а також для деяких його аналогів.
- Працює через стандартні технології. Нема потреби в якомусь спеціалізованому шлюзі для взаємодії – більшість працює через Wi-Fi, а деякі пристрої працюють через ZigBee, GPRS та Bluetooth.
- Повна свобода доступу. Є можливість підключення до пристроїв із будь-якої точки світу. А також програма відправляє push-повідомлення на телефон. Швидкість відгуку досить висока, навіть якщо йдеться про бюджетне обладнання.
- Широкий вибір виробників. Велика кількість виробників, у тому числі іменитих брендів, пропонують продукцію, що підходить для підключення за технологією TuYa Smart.
- Може працювати з іншими системами розумного будинку.
- Голосове керування. Є можливість підключення до сервісів голосового управління, наприклад Google Home та Amazon Alexa.
- Всередині програми максимально просто і зручно створювати сценарії та алгоритми автоматизації – сценаріїв просто безмежна кількість.
- Сумісність. Абсолютно всі пристрої, які мають логотип Powered by TuYa, ідеально сумісні між собою.
- Доступність. Вартість обладнання досить низька, єдине, що потрібно заплатити за сховище для системи відеоспостереження.

Недоліки

- Працює на базі хмарного сервісу TuYa Cloud, що підтримує зв'язок між усіма пристроями. І якщо немає інтернету, тоді сценарії автоматизації не функціонуватимуть.
- Пристрої виробляються різними виробниками, зокрема, маловідомими. Це означає, що є велика кількість обладнання низької якості, яке може зіпсувати враження про систему. Потрібно бути уважним під час виборів.
- Локалізована продукція іноді надто складна для українського споживача.

Незручно, що не передбачено web-інтерфейс або версія клієнта для комп'ютера, що спростило б роботу з обладнанням. [6]



Рис. 1.3 представлено різноманітні пристрої компанії TuYaSmart для розумного дому.

Я обрала створювати програму та макет з пристроїв самостійно, адже таким чином я програмую власноруч свій будинок і отримую більше знань і навичок, хоча це займає більше часу і звісно це рішення для людей, які цікавляться інженерією і програмуванням.

1.5 Програмовані плати, які були використані для розробки

Програмовані плати - це плати, які можна програмувати за допомогою мов програмування, щоб виконувати різні функції та задачі. Існує багато видів програмованих плат, які відрізняються за розміром, формою, архітектурою, можливостями та призначенням.

ESP32/8266 - це програмована плата, яка має вбудований Wi-Fi та Bluetooth, а також мікроконтролер, який може виконувати різні завдання. Можна використовувати для створення бездротових мереж, смарт-пристроїв, сенсорів, контролерів тощо. Програмується за допомогою Arduino IDE, MicroPython, Lua, JavaScript та інших мов програмування. ESP32/ESP8266 є однією з найбільш економічних та енергоефективних програмованих плат [7].



Рис. 2.1 представлено програмовані плати серії ESP32. Зокрема праворуч мікроконтролер ESP32-C6, а ліворуч мікроконтролер ESP32 WROOM-32D.

Це лише деякі приклади програмованих плат, їх існує значно більше. Було обрано ESP8266. Це недорогий мікрочіп Wi-Fi з вбудованим програмним забезпеченням мережевого стеку TCP/IP та можливостями мікроконтролера, вироблений компанією Espressif Systems у Шанхаї, Китай. Чіп став популярним серед англomовної спільноти мейкерів у серпні 2014 року завдяки модулю ESP-01,

виготовленому стороннім виробником Ai-Thinker. Чіп має 32-бітний мікроконтролер CPU на основі стандарту Tensilica Diamond Standard 106Micro, який працює на частоті 80 або 160 МГц. Він має 17 GPIO-виводів, 32 КіБ RAM інструкцій, 32 КіБ RAM кешу інструкцій, 80 КіБ RAM користувацьких даних та 16 КіБ RAM системних даних ETS. [8] Чіп можна програмувати безпосередньо за допомогою набору розробника програмного забезпечення (SDK), випущеного компанією Espressif Systems у жовтні 2014 року. SDK доступний у двох версіях, одна з яких базується на FreeRTOS, а інша - на зворотному виклику. Альтернативою офіційного SDK Espressif є відкрите SDK ESP-Open-SDK, яке базується на засобах трансляції GCC (GNU Compiler Collection). [9]

1.6 Аналіз та вибір програмного середовища для розробки системи "розумний дім".

Arduino IDE:

- Простий у використанні.
- Широкий вибір бібліотек і підтримка спільноти.
- Зручно для початківців.

```

Blink | Arduino 1.8.5
Blink §
This example code is in the public domain.
http://www.arduino.cc/en/Tutorial/Blink
*/
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
32 Arduino/Genuino Uno on COM1

```

Рис. 3.1 скріншот з середовища програмування Arduino IDE, де написано програму мигання світлодіода.

Це середовище програмування в цій програмі я буду писати код. Програма складається з препроцесору і компілятора, які перевіряють і компілюють код. IDE написана на java. IDE не має файлової структури проекту, також не має рефакторинга, не має автоматичний доповнювач коду також код повільно компілюється. [10]

Аналоги Arduino IDE:

- Notepad++ +Arduino IDE
- Platform IO

- Programino IDE
- Maria Mole
- B4R – середовище дозволяє програмувати на мові Basic.
- Visual Studio Micro
- XOD
- Atmel Studio

Для розробки схеми макету розумного дому на основі Nanit та ESP8266 я скористалася програмою Fritzing, я обрала саме її, бо для неї є бібліотека для мого основного елемента Nanit в відмінності від інших подібних програм таких як Proteus та SimulIDE.

Fritzing – відкритою ініціативою зі створення програмного забезпечення для любителів електроніки. Воно дозволяє розробляти схеми, а потім створювати більш стабільні пристрої на основі цих прототипів. Fritzing є безкоштовним програмним забезпеченням під ліцензією GPL 3.0 або пізнішою. З відкритим вихідним кодом можна ознайомитися на GitHub, а бінарні файли доступні за певну плату, що дозволено ліцензією GPL [11, В оригіналі Fritzing is an open-source initiative to develop amateur or hobby CAD software for the design of electronics hardware, intended to allow designers and artists to build more permanent circuits from prototypes. It was developed at the University of Applied Sciences Potsdam. Fritzing is free software under the GPL 3.0 or later license, with the source code available on GitHub and the binaries at a monetary cost, which is allowed by the GPL.]

РОЗДІЛ 2.

Практична Частина

2.1 Функціональні можливості розробленої системи "розумний дім".

Система розрахована на однокімнатну квартиру. Вона має охорону, систему запобігання протокам та економить кошти. Основний тариф для оплати електроенергії — трьохзонний.

1. Охорона

- При поставленні квартири на охорону вимикаються всі розетки окрім, розетки для пральної машини, холодильник, посудомийна машина, сушильна машина, охорона (камера, WI-FI, сигналізація) .

- Коли квартира знімається з охорони вмикаються розетки.

- Сигналізація спрацьовує при присутності в кімнаті, але не знято з охорони.

2. Посудомийка, пралка, сушилка. В час, коли тариф найдорожчий (піковий), потрібно щоб прилади вимикалися. До прикладу, ми вмикаємо прилад і він працює якщо людина присутня, або у відповідний час, коли не найдорожчий тариф (нічний, напівпіковий). Коли тариф найдорожчий (піковий) і нікого поруч немає, то вони вимикаються.

3. Бойлер, 50 літрів. Вимкнений у пікові години. Включений, коли тариф не найдешевший (нічний), напівпіковий і приміщення знято з охорони, о 6 бойлер вмикається (незалежно від усього).

4. Водопостачання.

- Посудомийка, пралка кран зав'язаний на розетку.
- Кухня, санвузол вмикається вода при присутності людини.

- Якщо вологість > 80 санвузлі вода в санвузлі вимикається.

5. Система обігріву.

- Кондиціонер працює при зовнішній температурі від 5 до 15, і при більше 27, і людина в кімнаті.

- Обігрівач працює якщо температура зовнішня менше 5 і людина присутня в кімнаті.

- Провітрювач вимикається у пікові години.

6. Тільки за присутності людини поруч з плитою та праскою вони працюють.

7. Світло.

- Коридор по руху.
- Кухня по присутності і якщо темно.
- Санвузол, кладовка по присутності.
- Кімната має два вимикача, один для самовимкнення/ввімкнення, другий автоматичний за присутності і темноти.

2.2 Апаратне забезпечення системи "розумний дім".

1. Давач світла
2. Давач температури
3. Наніт
4. Давач присутності для санвузла
5. Лампочка санвузол
6. Мотор санвузол
7. Бойлер — реле
8. Давач руху для коридору
9. Лампочка для коридору
10. Вай фай коридор — есп 8266
11. Сигналізація — п'єзоелемент в кімнаті
12. Дзвінок — п'єзоелемент + кнопка
13. Вимикач світла для кімнати — кнопка
14. Лампочка для кімнати
15. Провітрювач — реле
16. Обігрівач — реле
17. Кондиціонер — реле
18. Давач присутності для кімнати
19. Лампочка для кухні
20. Плита — реле
21. Кран на кухні — мотор
22. Кран для посудомийки — мотор
23. Давач присутності для кухні
24. Посудомийка — реле
25. Холодильник — реле
26. Лампочка для кладовки
27. Праска — реле

28.Кран для пралки — мотор

29.Давач присутності для кладовки

30.Пралка, сушилка— реле

2.3 Програмне забезпечення системи "розумний дім".

Охорона — режим для розумного будинку.

Програма розумного дому

1. Якщо натиснули кнопку «охорона» вимикаються всі елементи окрім реле «пралка-сушилка», реле «холодильник», реле «посудомийка», esp8266, кнопка «дзвінок», кнопка «охорона», п'єзоелемент «дзвінок», п'єзоелемент «сигналізація». Код вказаний у додатку В.
2. Якщо натиснули кнопку «охорона» другий раз, то всі вимкнені реле вмикаються. Код вказаний у додатку В.
3. Якщо давач присутності в кімнаті подав сигнал присутності людини і якщо не натиснута кнопка «охорона» другий раз, п'єзоелемент «сигналізація» подає сигнал.
4. Якщо давач присутності в кладовці подав сигнал присутності людини або час нічний, напівпіковий реле «пралка-сушилка» вмикаємо.
5. Якщо давач присутності на кухні подав сигнал присутності людини або час нічний, напівпіковий реле «мийка» вмикаємо.
6. Якщо 6 година вечора або кнопку «охорона» натиснули другий раз, вмикається світлодіод бойлер.
7. Кран «пралка-сушилка» відповідає реле «пралка-сушилка».
8. Кран «посудомийка» відповідає реле «помудомийка».
9. Якщо давач присутності в санвузлі подав сигнал присутності людини і давач вологості показує менше 80 відсотків, вмикається кран «санвузол».

10. Якщо давач присутності на кухні подав сигнал присутності людини, вмикається кран «кухня».

11. Якщо давач зовнішньої температури показує від 5 до 15 градусів або більше 27 і давач присутності в кімнаті подав сигнал присутності людини, вмикається реле «кондиціонер».

12. Якщо давач зовнішньої температури показує менше 5 градусів і давач присутності в кімнаті подав сигнал присутності людини, вмикається реле «обігрівач».

13. Якщо час нічний та напівпіковий, вмикається провітрювач.

14. Якщо давач присутності в кладовці подає сигнал присутності людини, вмикається реле «праска».

15. Якщо давач присутності на кухні подає сигнал присутності людини, вмикається реле «плита».

16. Якщо давач руху в коридорі подає сигнал руху, лампочка для коридору вмикається.

17. Якщо давач присутності на кухні подає сигнал присутності людини і давач світла не подає сигнал освітленості, лампочка для кухні вмикається.

18. Якщо давач присутності в санвузлі подає сигнал присутності людини, лампочка в санвузлі вмикається.

19. Якщо давач присутності в кладовці подає сигнал присутності людини, лампочка в кладовці вмикається.

20. Якщо давач присутності в кімнаті подає сигнал присутності людини і давач світла не подає сигнал освітленості, лампочка в кімнаті вмикається.

21. При натисканні на кнопку в кімнаті лампочка в кімнаті вмикається/вимикається незалежно.

2.4 Розробка інтерфейсу користувача системи "розумний дім".

На сайті нашого розумного дому ми можемо побачити температуру всередині квартири та назовні, час та обрати режим для нашого розумного дому, економії коштів, охороні та запобіганням протоків.

21:37:10

My smart house

Оберіть режим

Standart

S

t вулиця

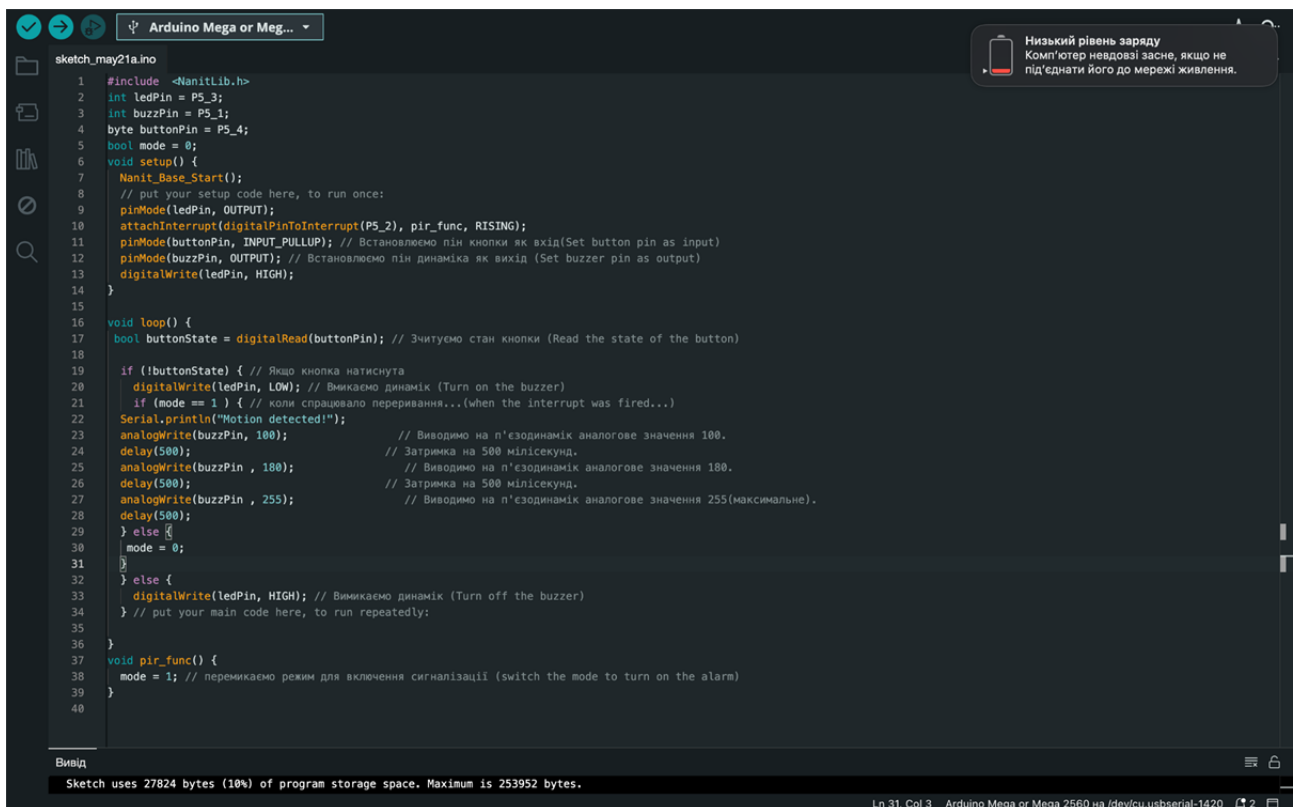
t дім

Рис 4.1 приклад сторінки сайту для системи розумного дому, яку розроблено. Зверху у нас час далі назва, під низом вибір режиму при натисканні на кнопку стандарт, якраз і обирається розроблений режим. Далі показується температура на вулиці і внизу показується температура в квартирі.

2.5 Тестування та налагодження системи "розумний дім"

У пункті один, два, три попереднього розділу у нас зазначено, якщо натиснули кнопку «охорона» вимикаються всі елементи окрім світлодіод «пралка-сушилка», світлодіод «холодильник», світлодіод «посудомийка», esp8266, кнопка «дзвінок», кнопка «охорона», п'єзоелемент «дзвінок», п'єзоелемент «сигналізація». Якщо натиснули кнопку «охорона» другий раз, то всі вимкнені прилади вмикаються.

Якщо давач присутності в кімнаті подав сигнал присутності людини і якщо не натиснута кнопка «охорона» другий раз, п'єзоелемент «сигналізація» подає сигнал. Отже, у нас мають вимкнутися лампочки при першому натисканні на кнопку, а також спрацювати сигналізація при фіксації руху в режимі охорона.



```

sketch_may21a.ino
1  #include <NanItLib.h>
2  int ledPin = P5_3;
3  int buzzPin = P5_1;
4  byte buttonPin = P5_4;
5  bool mode = 0;
6  void setup() {
7      NanIt_Base_Start();
8      // put your setup code here, to run once:
9      pinMode(ledPin, OUTPUT);
10     attachInterrupt(digitalPinToInterrupt(P5_2), pir_func, RISING);
11     pinMode(buttonPin, INPUT_PULLUP); // Встановлюємо пін кнопки як вход (Set button pin as input)
12     pinMode(buzzPin, OUTPUT); // Встановлюємо пін динаміка як вихід (Set buzzer pin as output)
13     digitalWrite(ledPin, HIGH);
14 }
15
16 void loop() {
17     bool buttonState = digitalRead(buttonPin); // Зчитуємо стан кнопки (Read the state of the button)
18
19     if (!buttonState) { // Якщо кнопка натиснута
20         digitalWrite(ledPin, LOW); // Вмикаємо динамік (Turn on the buzzer)
21         if (mode == 1) { // коли спрацювало переривання...(when the interrupt was fired...)
22             Serial.println("Motion detected!");
23             analogWrite(buzzPin, 100); // Виводимо на п'єзодинамік аналогове значення 100.
24             delay(500); // Затримка на 500 мілісекунд.
25             analogWrite(buzzPin, 180); // Виводимо на п'єзодинамік аналогове значення 180.
26             delay(500); // Затримка на 500 мілісекунд.
27             analogWrite(buzzPin, 255); // Виводимо на п'єзодинамік аналогове значення 255(максимальне).
28             delay(500);
29         } else {
30             mode = 0;
31         }
32     } else {
33         digitalWrite(ledPin, HIGH); // Вмикаємо динамік (Turn off the buzzer)
34     } // put your main code here, to run repeatedly:
35 }
36
37 void pir_func() {
38     mode = 1; // перемикаємо режим для включення сигналізації (switch the mode to turn on the alarm)
39 }
40
Вивід
Sketch uses 27824 bytes (10%) of program storage space. Maximum is 253952 bytes.
Ln 31, Col 3 Arduino Mega or Mega 2560 на /dev/cu.usbserial-1420

```

Рис. 5.1 скріншот виконання першої, другої, третьої частин програми у середовищі програмування Arduino IDE, тобто просто код, адже в Serial port у нас нічого не виводиться.

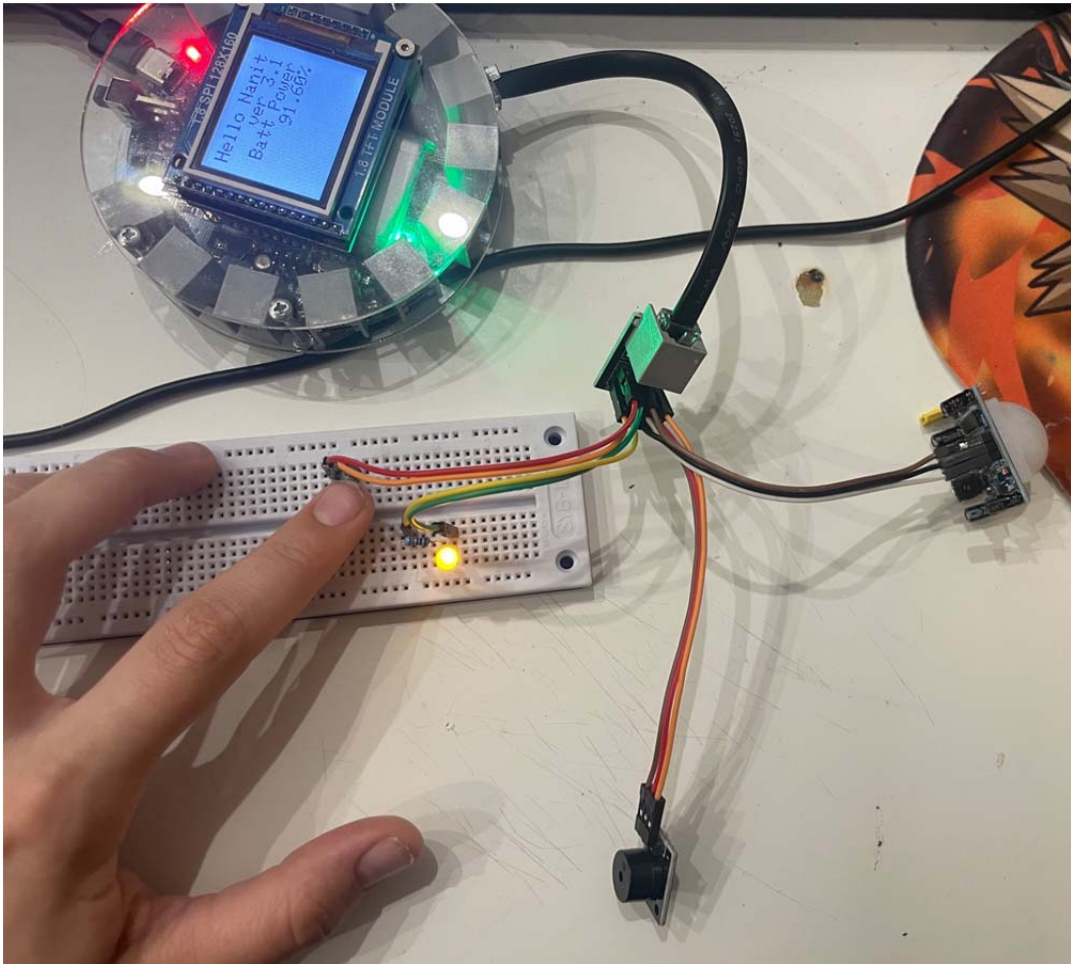
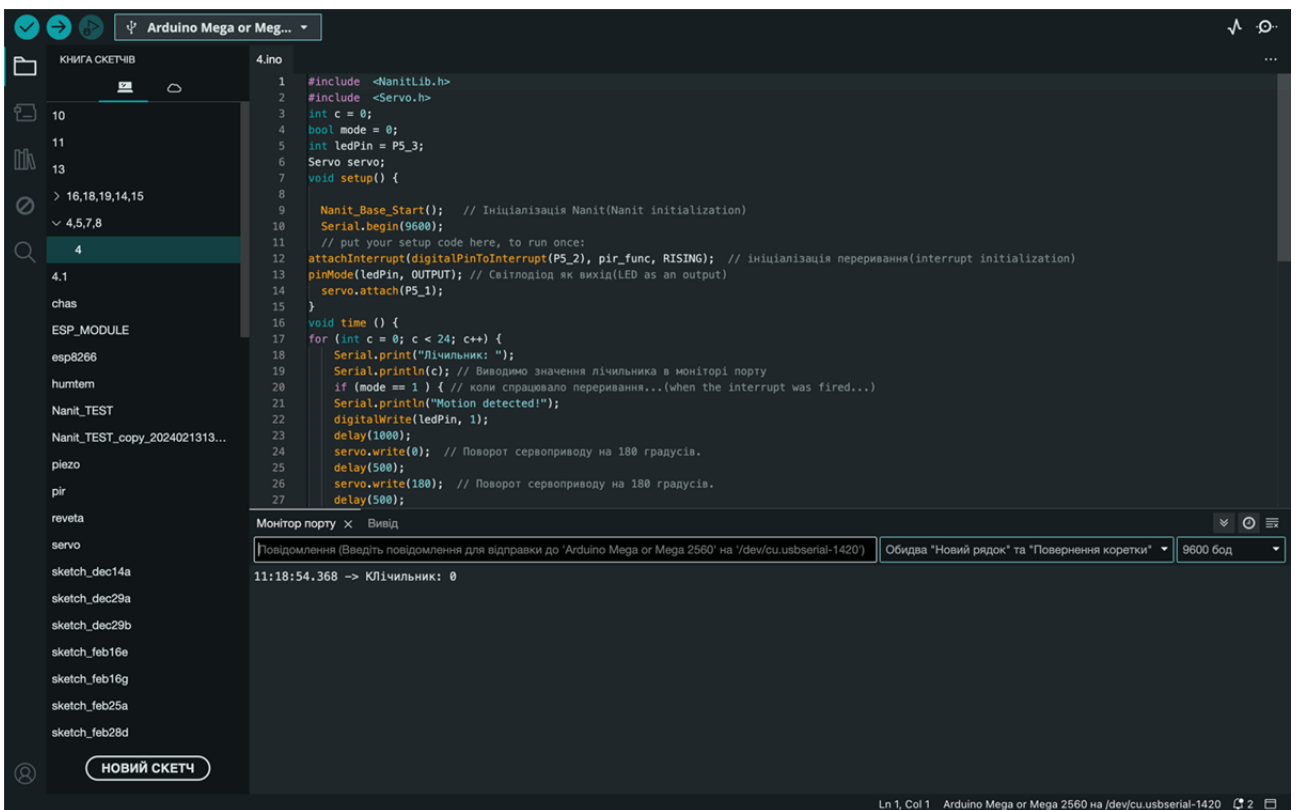


Рис. 5.2 робота частини моделі під пункти один, два, три. Тобто загорання лампочки – роботу пристрою, при вимкненому режимі охорона.

Лампочка показує роботу приладів і не роботу їх при режимі охорона, також при фіксації руху у режимі охорона п'єзоелемент подає сигнал.

У пункті чотири попереднього розділу у нас зазначено, якщо давач присутності в кладовці подав сигнал присутності людини або час нічний,

напівпіковий лампочка «пралка-сушилка» вмикаємо. Також у пункті п'ять маємо такі ж самі умови тільки для лампочки «посудомийка», якщо давач присутності на кухні подав сигнал присутності людини або час нічний, напівпіковий реле «посудомийка» вмикаємо. Також у пунктах сім і вісім зазначено, що крани підв'язані під ці прилади, тобто вмикаються і вимикаються разом з ними. Отже, у нас має вмикатися лампочка і кран з 0 до 8, потім вимикатися і знову вмикатися з 11 до 20, вимикатися, і вмикатися з 22 до 0. Також лампочка вмикається при фіксації руху.



```

1 #include <NanitLib.h>
2 #include <Servo.h>
3 int c = 0;
4 bool mode = 0;
5 int ledPin = P5_3;
6 Servo servo;
7 void setup() {
8
9     Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
10    Serial.begin(9600);
11    // put your setup code here, to run once:
12    attachInterrupt(digitalPinToInterrupt(P5_2), pir_func, RISING); // ініціалізація переривання(interrupt initialization)
13    pinMode(ledPin, OUTPUT); // Світлодіод як вихід(LED as an output)
14    servo.attach(P5_1);
15 }
16 void time () {
17     for (int c = 0; c < 24; c++) {
18         Serial.println("Лічильник: ");
19         Serial.println(c); // Виводимо значення лічильника в моніторі порту
20         if (mode == 1) { // коли спрацювало переривання...(when the interrupt was fired...)
21             Serial.println("Motion detected!");
22             digitalWrite(ledPin, 1);
23             delay(1000);
24             servo.write(0); // Поворот сервоприводу на 180 градусів.
25             delay(500);
26             servo.write(180); // Поворот сервоприводу на 180 градусів.
27             delay(500);
28         }
29     }
30 }

```

Монітор порту x Вивід

Повідомлення (Введіть повідомлення для відправки до 'Arduino Mega or Mega 2560' на '/dev/cu.usbserial-1420') Обидва "Новий рядок" та "Повернення каретки" 9600 бод

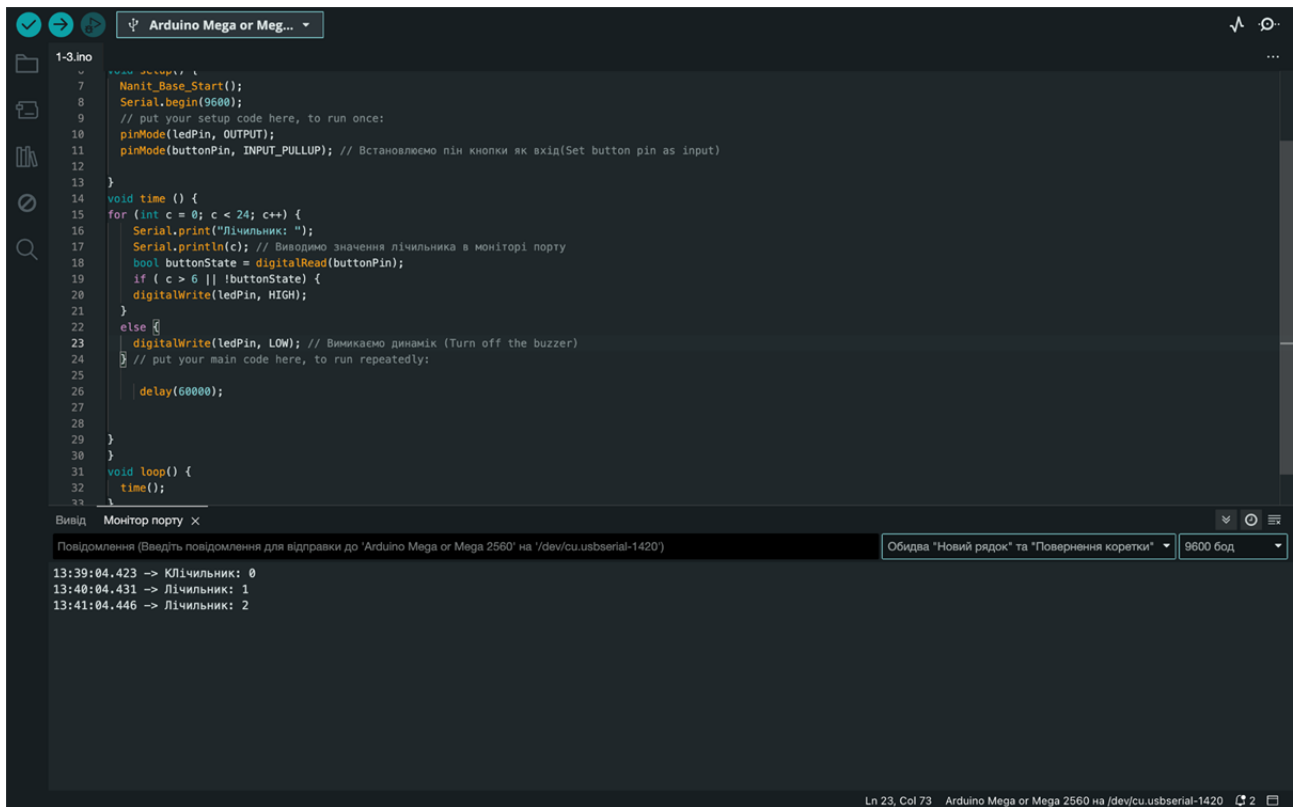
11:18:54.368 -> Лічильник: 0

Рис. 5.3 скріншот виконання четвертої, п'ятої, сьомої та восьмої частин програми у середовищі програмування Arduino IDE, тобто код та виведення в Serial Port лічильника хвилин (було використано хвилини замість годин, аби перевірка не зайняла багато часу).



На рис. 5.4 робота частини моделі під пункти чотири, п'ять, сім, вісім.
Тобто загорання лампочки – роботу пристрою, при присутності.

У пункті шість попереднього розділу у нас зазначено, якщо 6 година вечора або кнопку «охорона» натиснули другий раз, вмикається реле бойлер. Отже, має загоратися лампочка, що показує роботу бойлера, після 6 годин та якщо зняли житло з охорони.



The screenshot shows the Arduino IDE interface. The main window displays the code for a program named '1-3.ino'. The code includes a setup function and a time function that prints the current time and button state to the serial port. The serial monitor at the bottom shows the output of the program, which is the time and the state of the button (0, 1, or 2).

```
7  #include <Arduino.h>
8  #include <LedControl.h>
9  // put your setup code here, to run once:
10 pinMode(ledPin, OUTPUT);
11 pinMode(buttonPin, INPUT_PULLUP); // Встановлюємо пін кнопки як вхід (Set button pin as input)
12
13 }
14
15 void time () {
16   for (int c = 0; c < 24; c++) {
17     Serial.print("Лічильник: ");
18     Serial.println(c); // Виводимо значення лічильника в моніторі порту
19     bool buttonState = digitalRead(buttonPin);
20     if ( c > 6 || !buttonState) {
21       digitalWrite(ledPin, HIGH);
22     }
23     else {
24       digitalWrite(ledPin, LOW); // Вимикаємо динанік (Turn off the buzzer)
25     }
26     // put your main code here, to run repeatedly:
27     delay(60000);
28   }
29 }
30
31 void loop() {
32   time();
33 }
```

Вивід Монитор порту x

Повідомлення (Введіть повідомлення для відправки до 'Arduino Mega or Mega 2560' на '/dev/cu.usbserial-1420')

Обидва "Новий рядок" та "Повернення каретки" 9600 бод

13:39:04.423 -> Лічильник: 0
13:40:04.431 -> Лічильник: 1
13:41:04.446 -> Лічильник: 2

Ln 23, Col 73 Arduino Mega or Mega 2560 на /dev/cu.usbserial-1420

Рис. 5.5 скріншот виконання шостої частини програми у середовищі програмування Arduino IDE, тобто код та виведення в Serial Port лічильника хвилин (було використано хвилини замість годин, аби перевірка не зайняла багато часу).

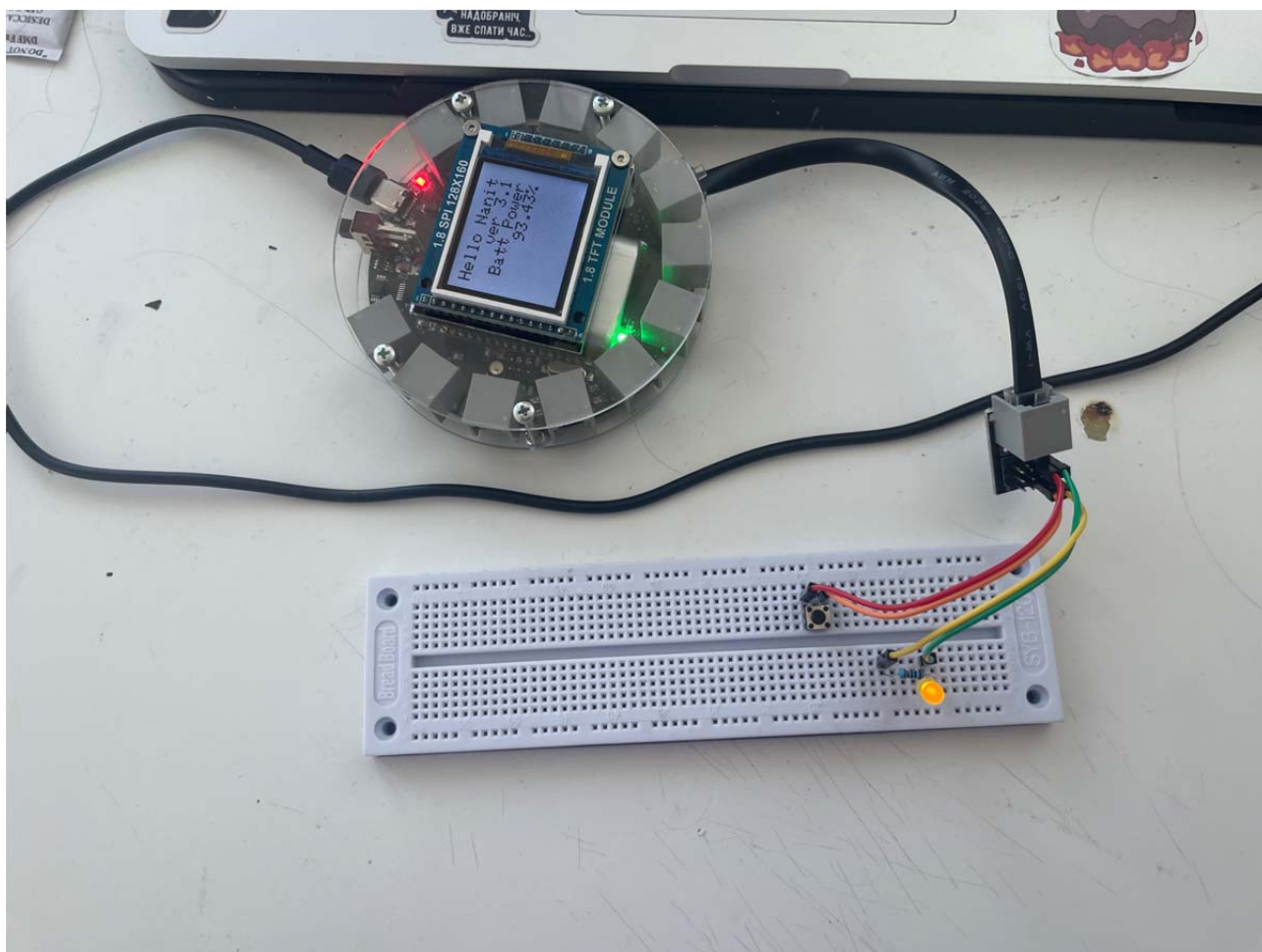
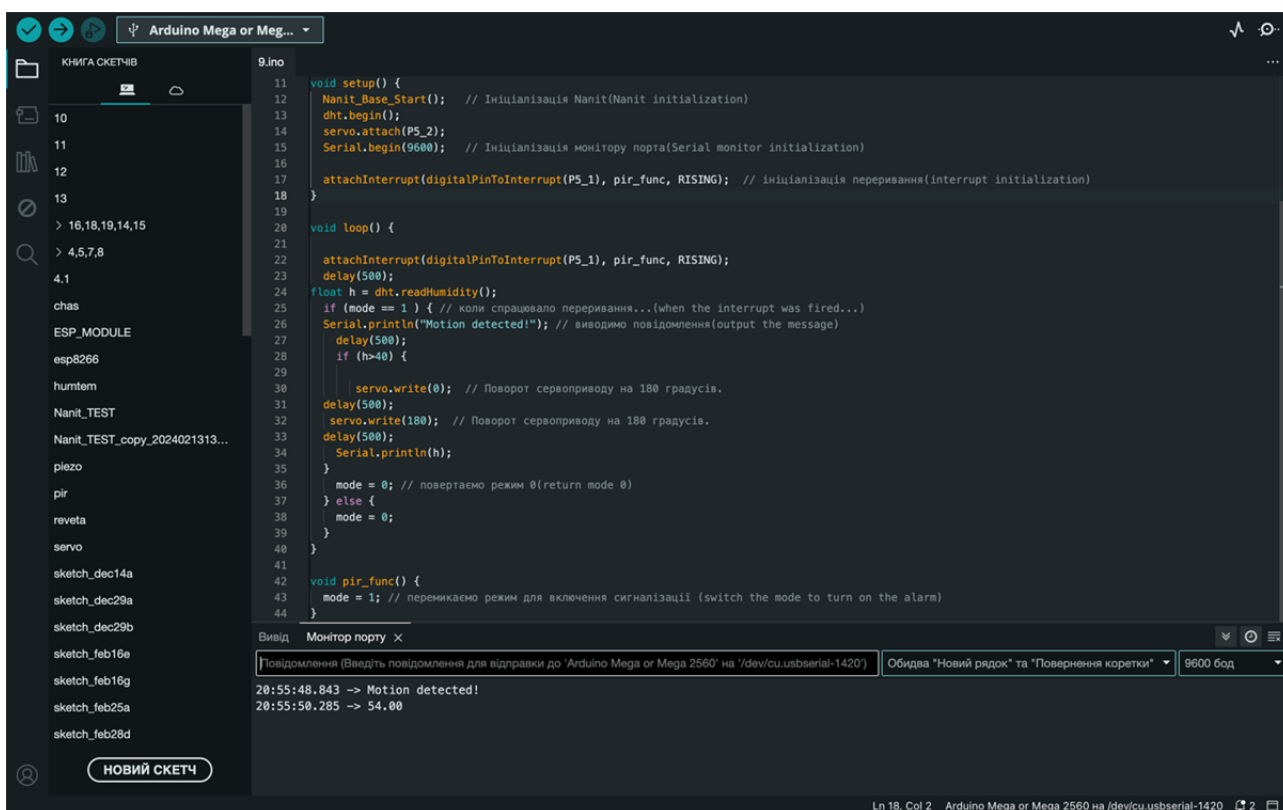


Рис. 5.7 робота частини моделі під пункт шість. Тобто загорання лампочки – роботу пристрою, після 6 годин (хвилин).

Світлодіод загорається в потрібний час і при натисканні на кнопку. Лампочка і кран працюють, тобто вмикаються і вмикаються у нічний та напівпіковий час і лампочка вмикається при фіксації руху.

У пункті дев'ять у нас зазначено, якщо давач присутності в санвузлі подав сигнал присутності людини і давач вологості показує менше 80 відсотків, вмикається кран «санвузол». Тобто тільки якщо вологість не перевищує 80 і зафіксовано рух, то вмикається кран для санвузла.



```
9.ino
11 void setup() {
12   Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
13   dht.begin();
14   servo.attach(P5_2);
15   Serial.begin(9600); // Ініціалізація монітору порта(Serial monitor initialization)
16
17   attachInterrupt(digitalPinToInterrupt(P5_1), pir_func, RISING); // Ініціалізація переривання(interrupt initialization)
18 }
19
20 void loop() {
21
22   attachInterrupt(digitalPinToInterrupt(P5_1), pir_func, RISING);
23   delay(500);
24   float h = dht.readHumidity();
25   if (mode == 1) { // коли спрацювало переривання...(when the interrupt was fired...)
26     Serial.println("Motion detected!"); // виводимо повідомлення(output the message)
27     delay(500);
28     if (h>40) {
29       servo.write(0); // Поворот сервоприводу на 180 градусів.
30     }
31     delay(500);
32     servo.write(180); // Поворот сервоприводу на 180 градусів.
33     delay(500);
34     Serial.println(h);
35   }
36   mode = 0; // повертаємо режим 0(return mode 0)
37 } else {
38   mode = 0;
39 }
40 }
41
42 void pir_func() {
43   mode = 1; // перемикаємо режим для вклення сигналізації (switch the mode to turn on the alarm)
44 }
```

Вивід Монитор порту X

Повідомлення (Введіть повідомлення для відправки до 'Arduino Mega or Mega 2560' на '/dev/cu.usbserial-1420') Обидва "Новий рядок" та "Повернення коретки" 9600 бод

20:55:48.843 -> Motion detected!
20:55:50.285 -> 54.00

Ln 18, Col 2 Arduino Mega or Mega 2560 на /dev/cu.usbserial-1420

Рис. 5.8 скріншот виконання дев'ятої частини програми у середовищі програмування Arduino IDE, тобто код та виведення в Serial Port фіксації руху та відсотку вологості.

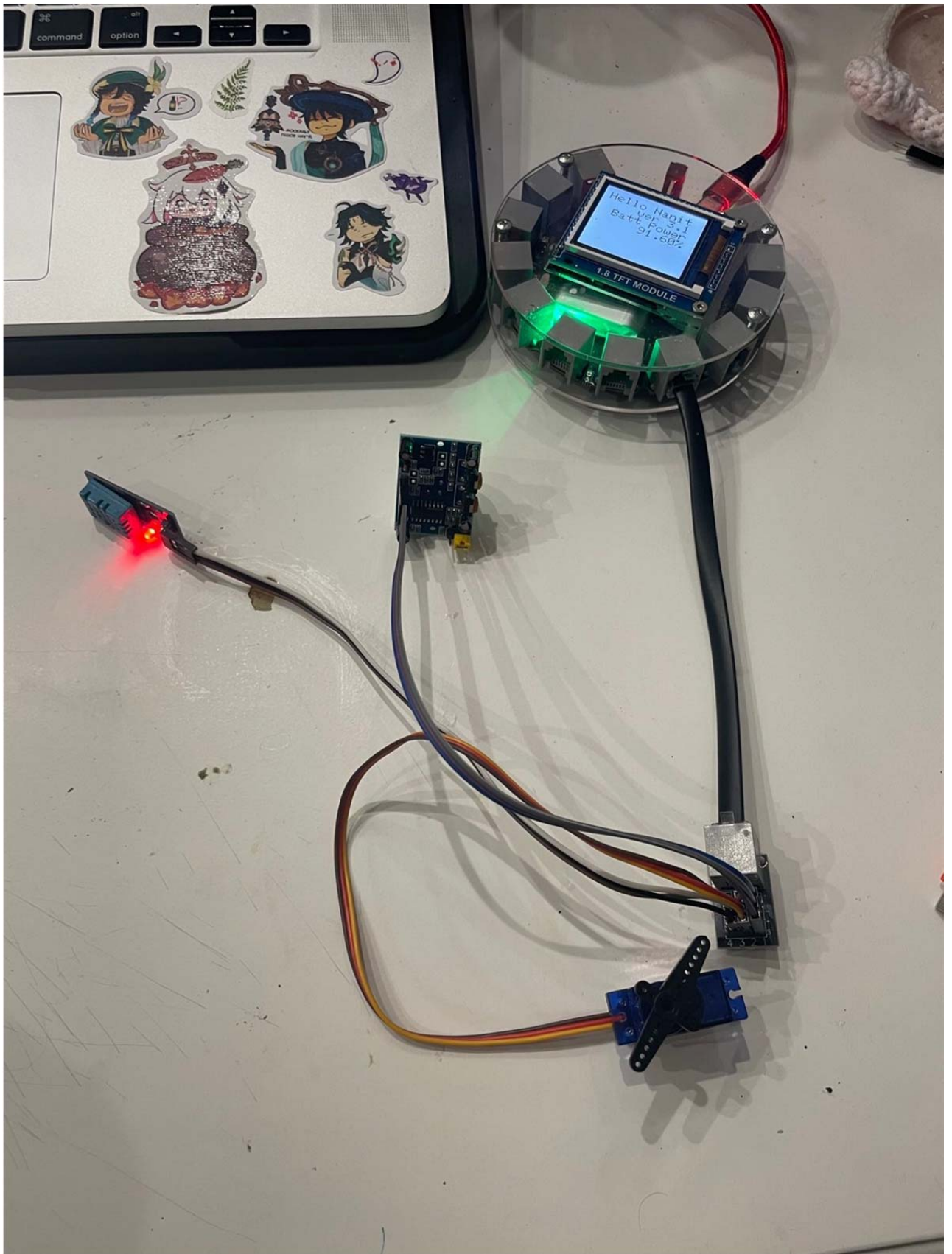
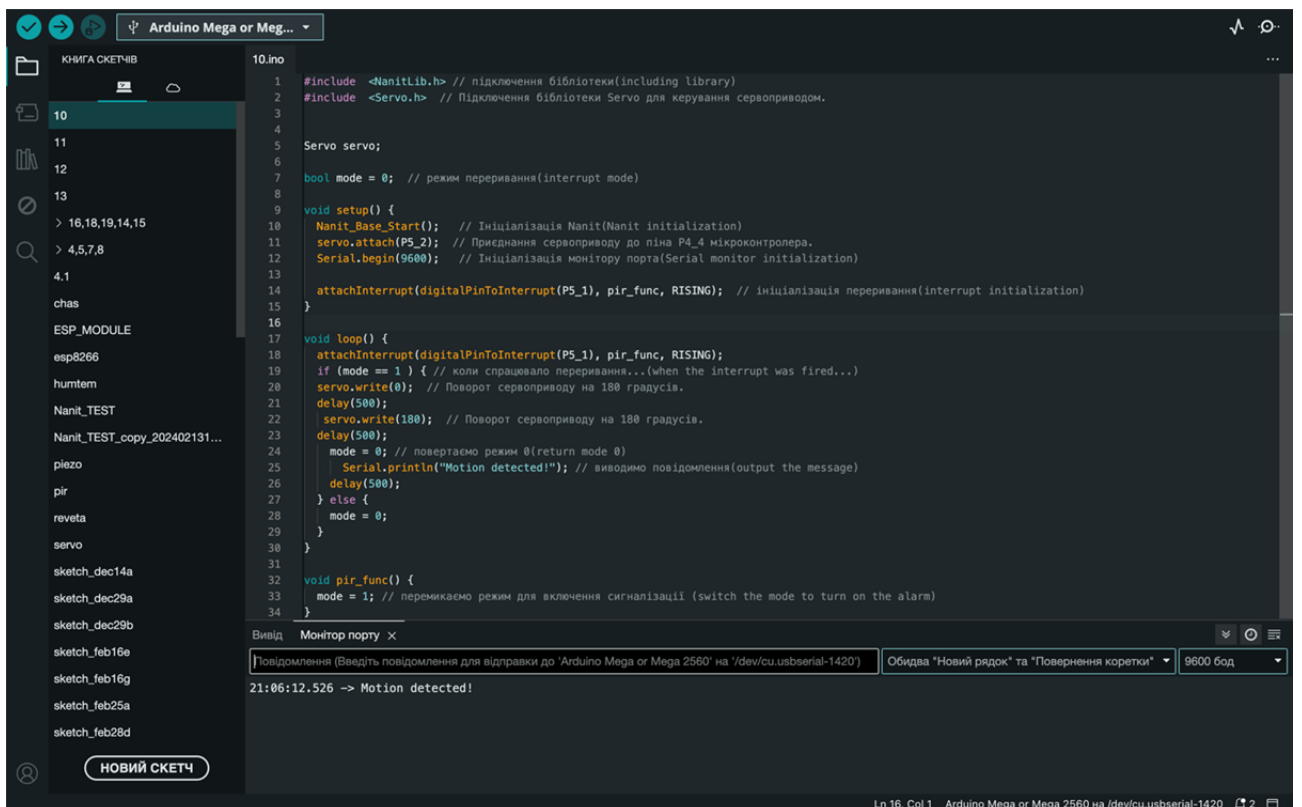


Рис. 5.9 робота частини моделі під пункт дев'ять.

Кран працює, тобто вмикається при фіксації руху і якщо вологість менше 80%.

У пункті десять у нас зазначено, якщо давач присутності на кухні подав сигнал присутності людини, вмикається кран «кухня». Тобто тільки при присутності людини кран працює.



```

1 #include <NanitLib.h> // підключення бібліотеки(including library)
2 #include <Servo.h> // Підключення бібліотеки Servo для керування сервоприводом.
3
4
5 Servo servo;
6
7 bool mode = 0; // режим переривання(interrupt mode)
8
9 void setup() {
10   Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
11   servo.attach(P5_2); // Приєднання сервоприводу до пін P4_4 мікроконтролера.
12   Serial.begin(9600); // Ініціалізація монітору порта(Serial monitor initialization)
13
14   attachInterrupt(digitalPinToInterrupt(P5_1), pir_func, RISING); // ініціалізація переривання(interrupt initialization)
15 }
16
17 void loop() {
18   attachInterrupt(digitalPinToInterrupt(P5_1), pir_func, RISING);
19   if (mode == 1) { // коли спрацювало переривання...(when the interrupt was fired...)
20     servo.write(0); // Поворот сервоприводу на 180 градусів.
21     delay(500);
22     servo.write(180); // Поворот сервоприводу на 180 градусів.
23     delay(500);
24     mode = 0; // повертаємо режим 0(return mode 0)
25     Serial.println("Motion detected!"); // виводимо повідомлення(output the message)
26     delay(500);
27   } else {
28     mode = 0;
29   }
30 }
31
32 void pir_func() {
33   mode = 1; // перемикаємо режим для включення сигналізації (switch the mode to turn on the alarm)
34 }

```

Вивід: Монітор порту x

Повідомлення (Введіть повідомлення для відправки до 'Arduino Mega or Mega 2560' на '/dev/cu.usbserial-1420') Обидва "Новий рядок" та "Повернення каретки" 9600 бод

21:06:12.526 -> Motion detected!

Ln 16, Col 1 Arduino Mega or Mega 2560 на /dev/cu.usbserial-1420 2

Рис. 5.10 скріншот виконання десятої частини програми у середовищі програмування Arduino IDE, тобто код та виведення в Serial Port фіксації руху.

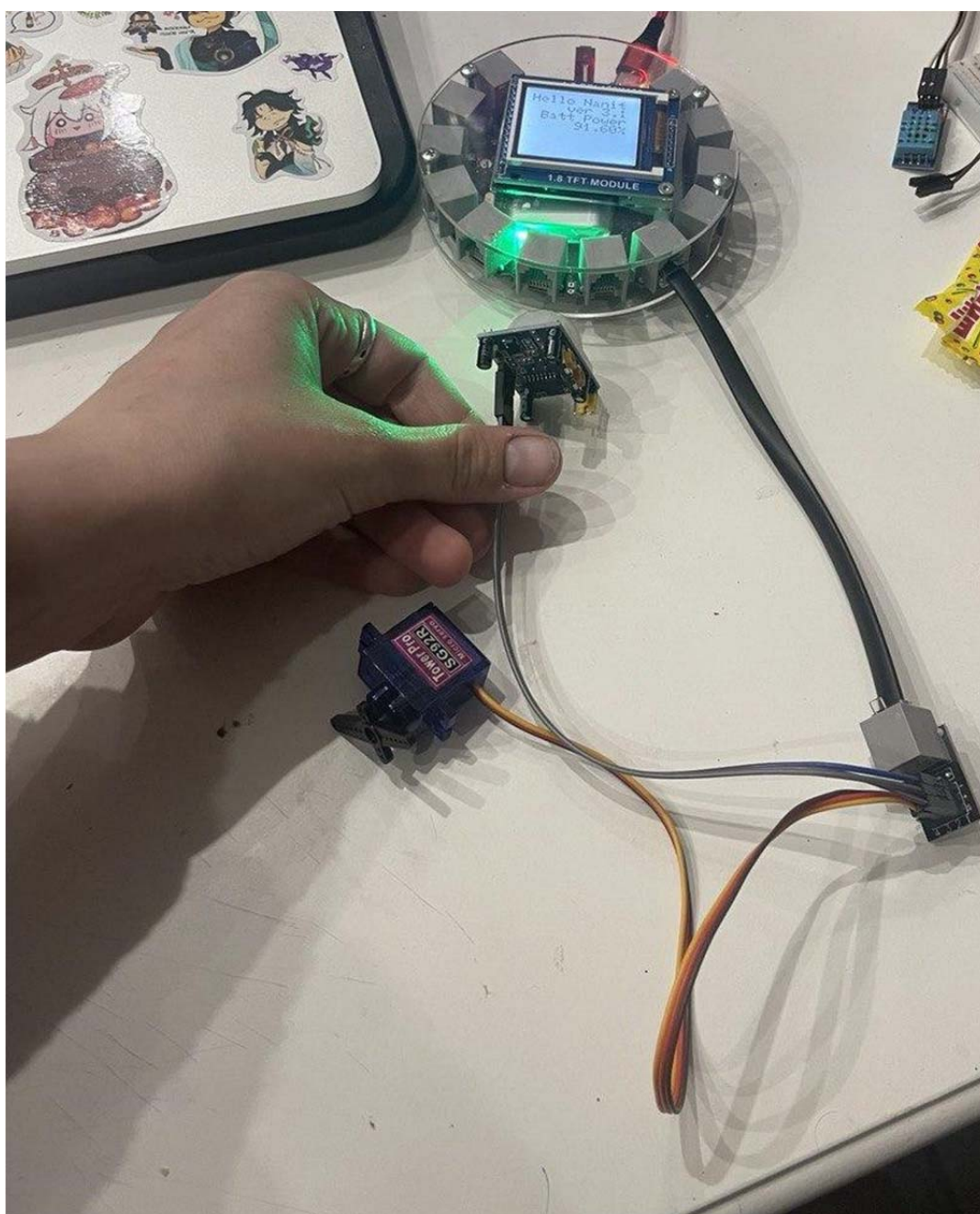


Рис. 5.11 робота частини моделі під пункт десять.

Кран працює, тобто вмикається при фіксації людини поруч.

У пункті одинадцять у нас зазначено, якщо датчик зовнішньої температури показує від 5 до 15 градусів або більше 27 і датчик присутності в кімнаті подав сигнал присутності людини, вмикається реле «кондиціонер». Отже, у нас має вмикатися лампочка при температурі від 5 до 15 і більше 27 градусів і при присутності людини.

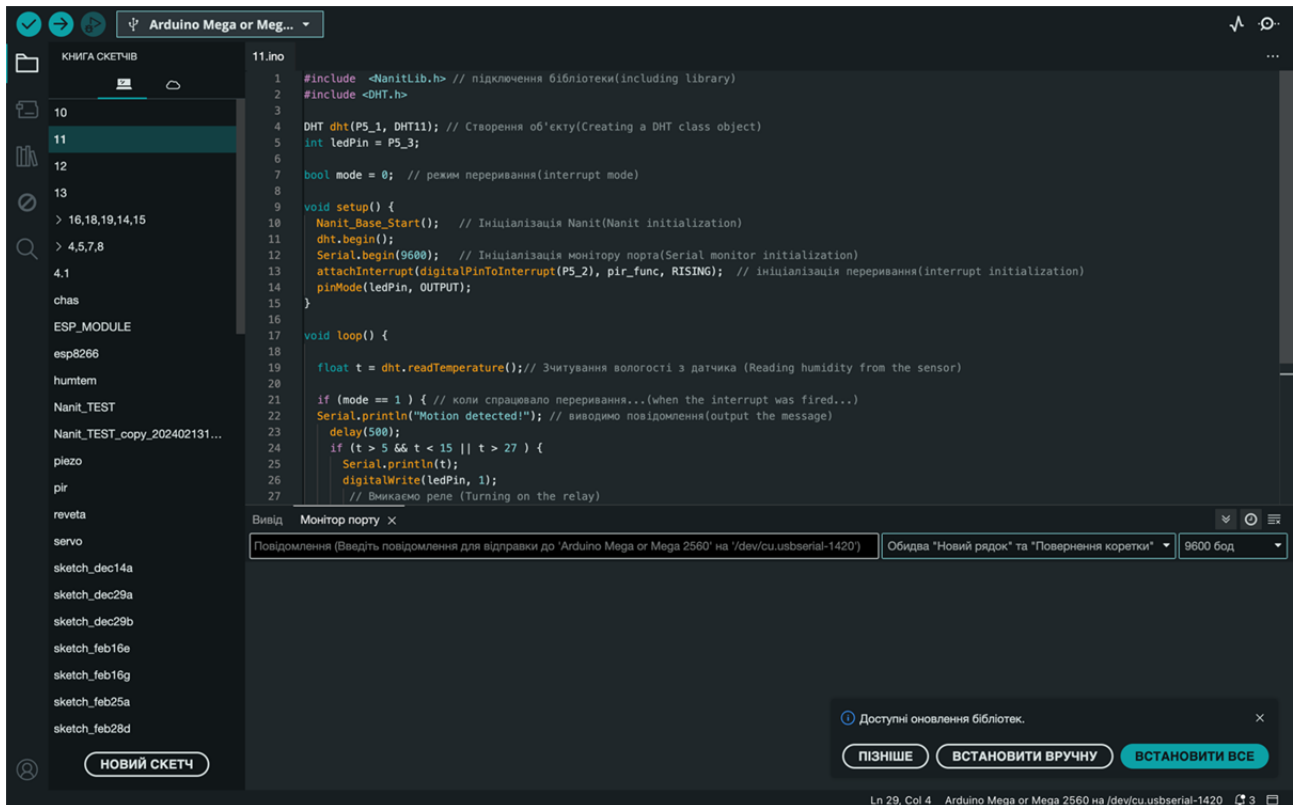


Рис. 5.12 скріншот виконання одинадцятої частини програми у середовищі програмування Arduino IDE.

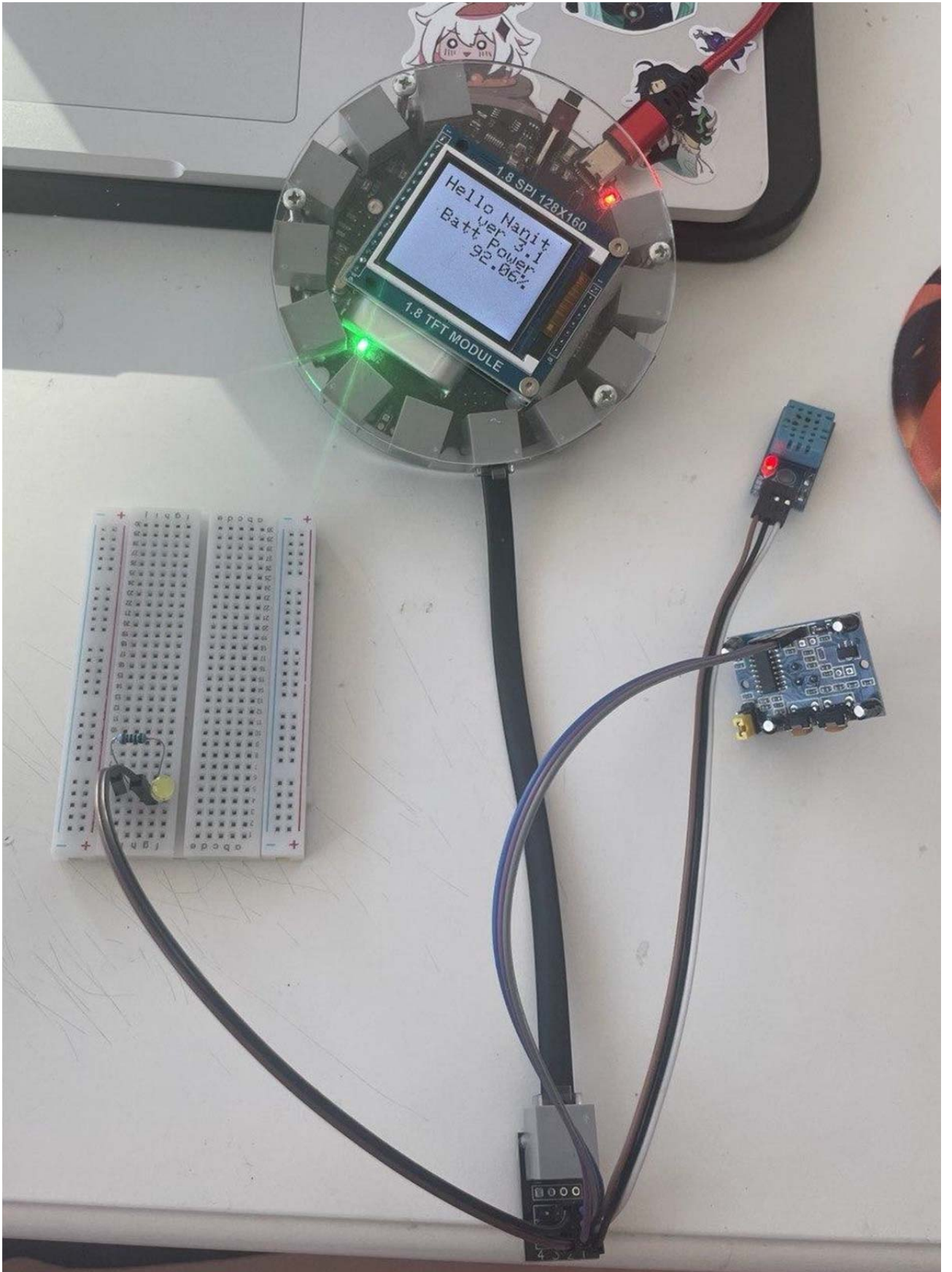
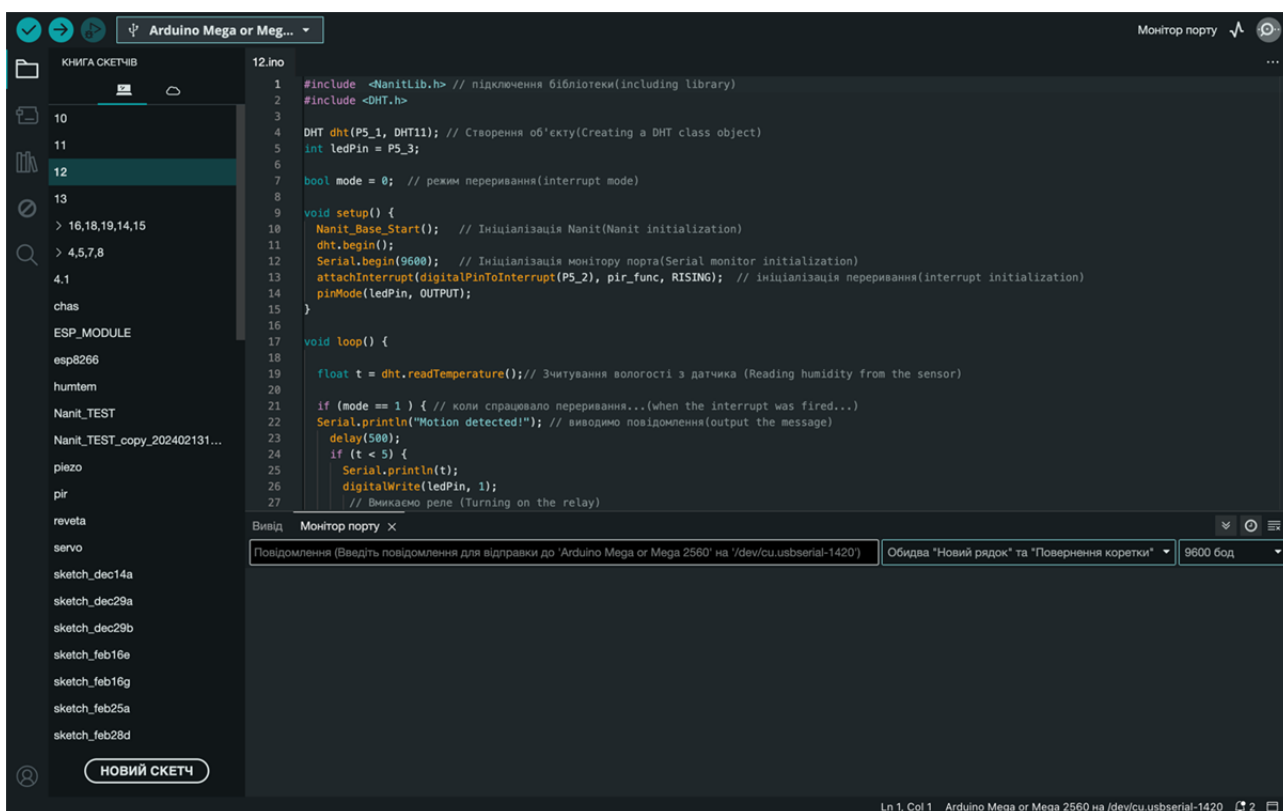


Рис. 5.13 робота частини моделі під пункт одинадцять. Тобто лампочка не горить – кондиціонер не працює.

Лампочка працює, тобто вмикається при температурі від 5 до 15 градусів і більше 27 та фіксації руху.

У пункті дванадцять у нас зазначено, якщо давач зовнішньої температури показує менше 5 градусів і давач присутності в кімнаті подав сигнал присутності людини, вмикається реле «обігрівач». Отже, у нас має вмикатися лампочка при температурі менше 5 градусів і при присутності людини.



```
1 #include <NanitLib.h> // підключення бібліотеки(including library)
2 #include <DHT.h>
3
4 DHT dht(P5_1, DHT11); // Створення об'єкту(creating a DHT class object)
5 int ledPin = P5_3;
6
7 bool mode = 0; // режим переривання(interrupt mode)
8
9 void setup() {
10   Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
11   dht.begin();
12   Serial.begin(9600); // Ініціалізація монітору порта(Serial monitor initialization)
13   attachInterrupt(digitalPinToInterrupt(P5_2), pir_func, RISING); // ініціалізація переривання(interrupt initialization)
14   pinMode(ledPin, OUTPUT);
15 }
16
17 void loop() {
18
19   float t = dht.readTemperature(); // Зчитування вологості з датчика (Reading humidity from the sensor)
20
21   if (mode == 1) { // коли спрацювало переривання...(when the interrupt was fired...)
22     Serial.println("Motion detected!"); // виводимо повідомлення(output the message)
23     delay(500);
24     if (t < 5) {
25       Serial.println(t);
26       digitalWrite(ledPin, 1);
27       // Вмикаємо реле (Turning on the relay)
```

Рис. 5.14 скріншот виконання дванадцятої частини програми у середовищі програмування Arduino IDE, тобто просто код.

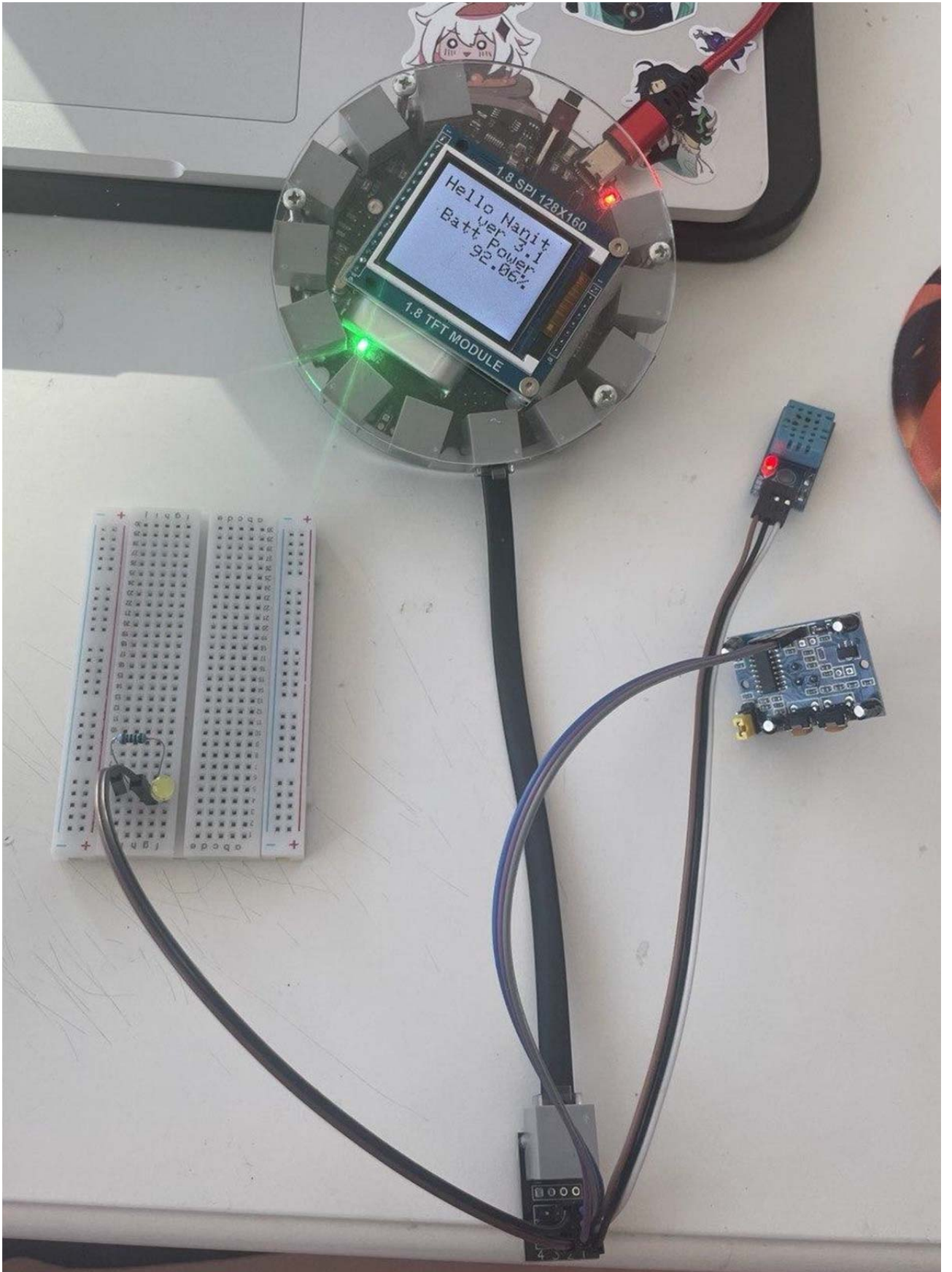


Рис. 5.15 робота частини моделі під пункт одинадцять. Тобто лампочка не горить – обігрівач не працює.

Лампочка працює, тобто вмикається при температурі менше 5 градусів та фіксації руху.

У пункті тринадцять у нас зазначено, якщо час нічний та напівпіковий, вмикається провітрювач.

Отже, у нас має вмикатися лампочка з 0 до 8, потім вимикатися і знову вмикатися з 11 до 20, вимикатися, і вмикатися з 22 до 0.

```

13.ino
2  int c = 0;
3  int ledPin = P5_2;
4
5  void setup() {
6
7      Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
8      Serial.begin(9600);
9      // put your setup code here, to run once:// ініціалізація переривання(interrupt initialization)
10     pinMode(ledPin, OUTPUT); // Світлодіод як вихід(LED as an output)
11 }
12 void time () {
13     for (int c = 0; c < 24; c++) {
14         Serial.println("Лічильник: ");
15         Serial.println(c); // Виводимо значення лічильника в моніторі порту
16         delay(60000); // Затримка для уникнення дрібних помилок
17     }
18 }
19 void loop() {
20     Serial.println(c);
21     if ( c < 8 || c > 11 || c > 22 ) {
22         if ( c < 20 ){
23             digitalWrite(ledPin, 1); // вмикаємо напругу для вклучення світлодіода(turn on the voltage to turn on the LED)
24             delay(60000); // чекаємо 1 сек(wait 1 sec)
25         }
26     }
27 }
28
Вивід Монитор порту x
Повідомлення (Введіть повідомлення для відправки до 'Arduino Mega or Mega 2560' на '/dev/cu.usbserial-1420') Обидва "Новий рядок" та "Повернення коретки" 9600 бод
17:26:58.193 -> K0
Ln 22, Col 21 Arduino Mega or Mega 2560 на /dev/cu.usbserial-1420

```

Рис. 5.16 скріншот виконання тринадцятої частини програми у середовищі програмування Arduino IDE, тобто код та виведення значень лічильника часу.

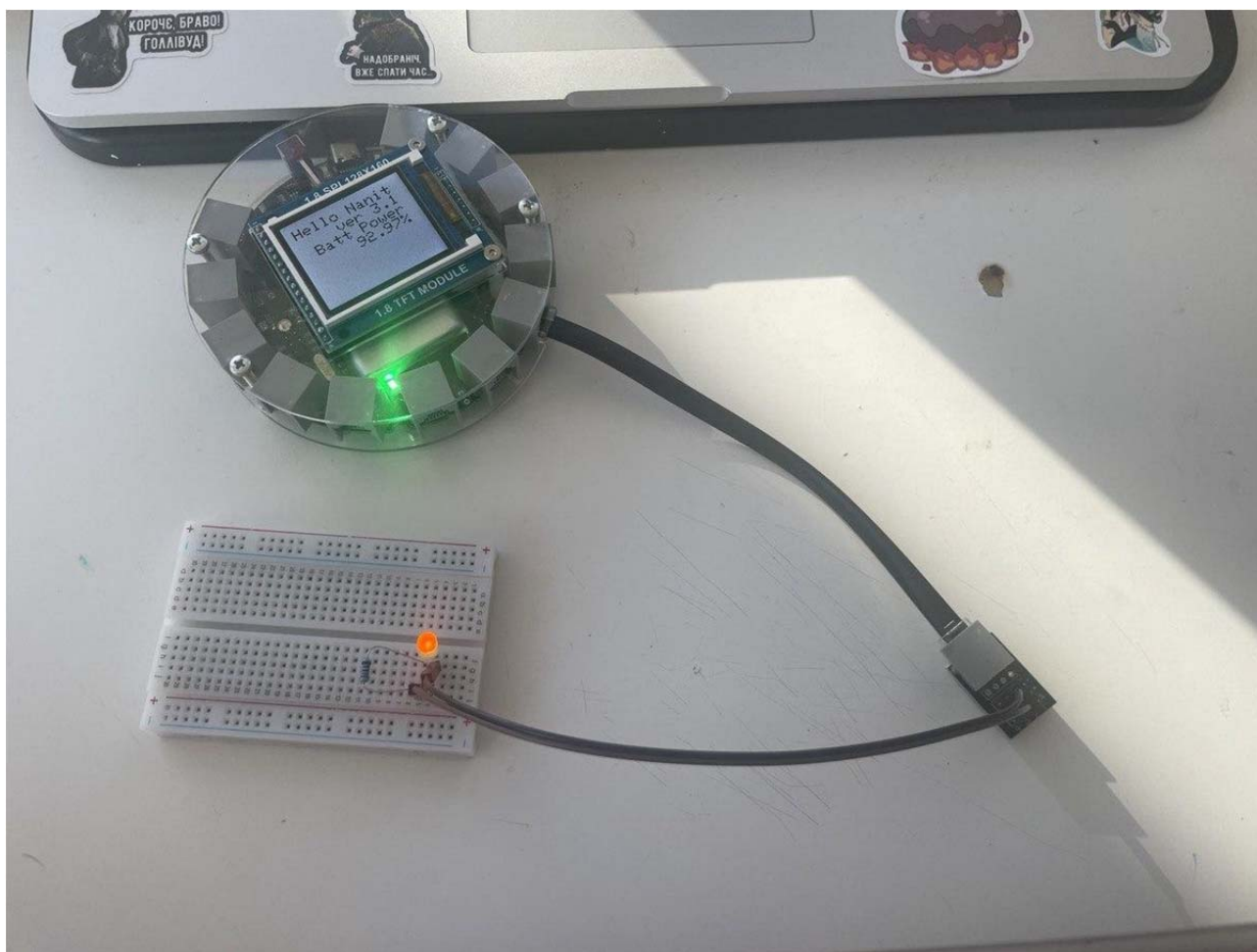
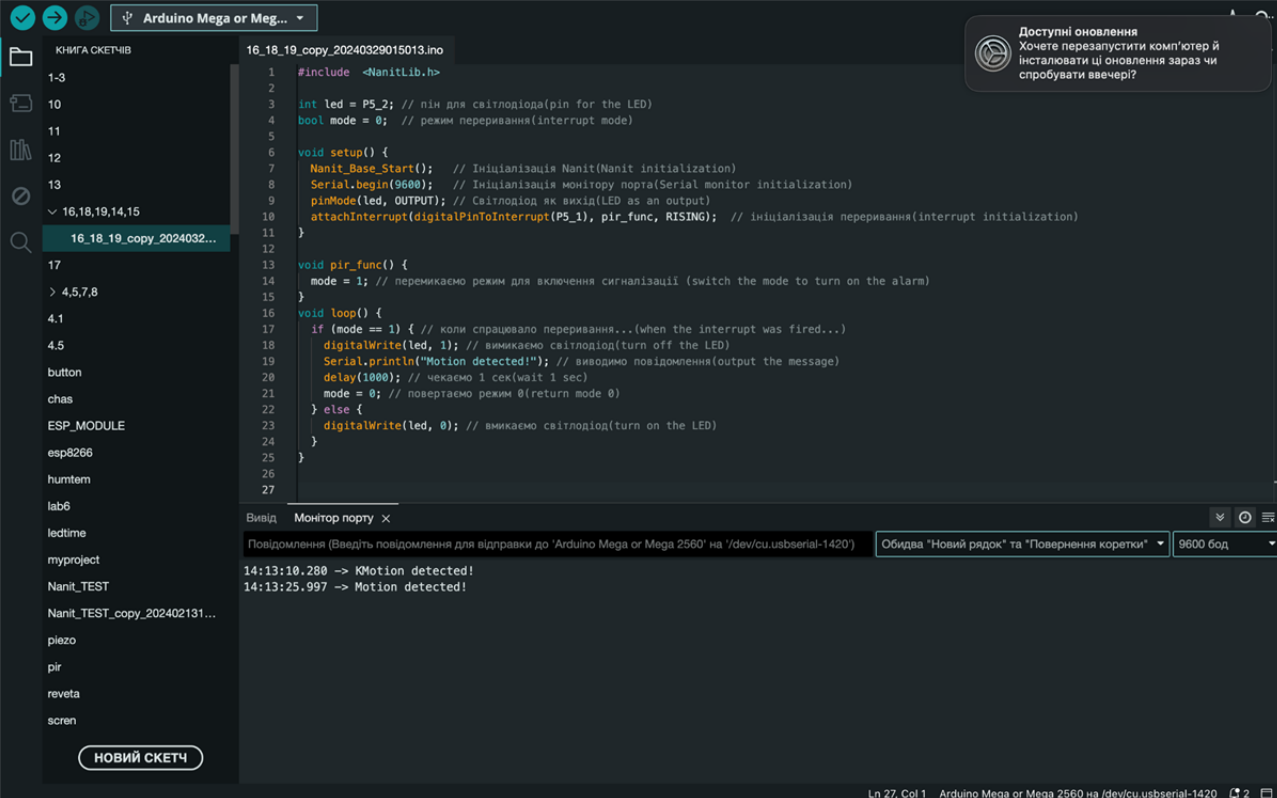


Рис. 5.17 робота частини моделі під пункт тринадцять. Тобто лампочка горить – провітрювач працює.

Лампочка працює, тобто вмикається і вмикається у нічний та напівпіковий час.

У пункті чотирнадцять у нас зазначено, якщо давач присутності в кладовці подає сигнал присутності людини, вмикається реле «праска». Також схожа умова у пункті п'ятнадцять тільки для плити. Якщо давач присутності на кухні подає сигнал присутності людини, вмикається реле «плита». Також так само працює лампочки у пунктах шістнадцять, вісімнадцять і дев'ятьнадцять. Якщо давач руху в коридорі подає сигнал руху, лампочка для коридору вмикається. Якщо давач руху в коридорі подає сигнал руху, лампочка для коридору вмикається. Якщо давач

присутності в кладовці подає сигнал присутності людини, лампочка в кладовці вмикається. Отже, лампочка має вмикатися при присутності.



The screenshot shows the Arduino IDE interface. The main editor displays the following code:

```
1 #include <NanitLib.h>
2
3 int led = P5_2; // пін для світлодіода(pin for the LED)
4 bool mode = 0; // режим переривання(interrupt mode)
5
6 void setup() {
7   Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
8   Serial.begin(9600); // Ініціалізація монітору порта(Serial monitor initialization)
9   pinMode(led, OUTPUT); // Світлодіод як вихід(LED as an output)
10  attachInterrupt(digitalPinToInterrupt(P5_1), pir_func, RISING); // ініціалізація переривання(interrupt initialization)
11 }
12
13 void pir_func() {
14   mode = 1; // перемикаємо режим для включення сигналізації (switch the mode to turn on the alarm)
15 }
16
17 void loop() {
18   if (mode == 1) { // коли спрацювало переривання...(when the Interrupt was fired...)
19     digitalWrite(led, 1); // вмикаємо світлодіод(turn off the LED)
20     Serial.println("Motion detected!"); // виводимо повідомлення(output the message)
21     delay(1000); // чекаємо 1 сек(wait 1 sec)
22     mode = 0; // повертаємо режим 0(return mode 0)
23   } else {
24     digitalWrite(led, 0); // вмикаємо світлодіод(turn on the LED)
25   }
26 }
27
```

The Serial Monitor at the bottom shows the following output:

```
Повідомлення (Введіть повідомлення для відправки до 'Arduino Mega or Mega 2560' на '/dev/cu.usbserial-1420') Обидва "Новий рядок" та "Повернення каретки" 9600 бод
14:13:10.280 -> Motion detected!
14:13:25.997 -> Motion detected!
```

Рис. 5.18 скріншот виконання чотирнадцятої, п'ятнадцятої, шістнадцятої, вісімнадцятої та дев'ятнадцятої частин програми у середовищі програмування Arduino IDE, тобто код та виведення повідомлення про зафіксований рух.

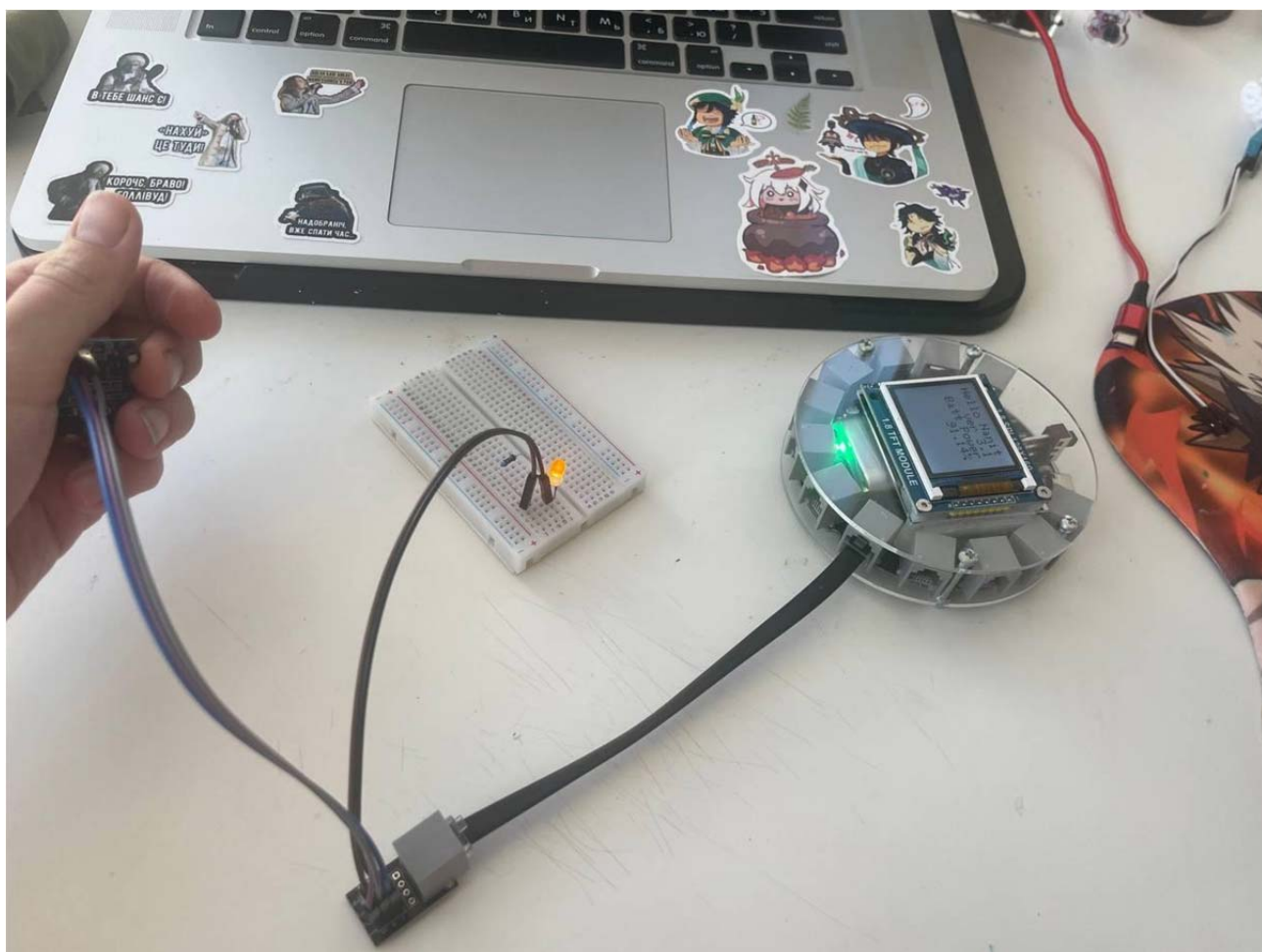


Рис. 5.19 робота частини моделі під пункти чотирнадцять, п'ятнадцять, шістнадцять, вісімнадцять, дев'ятнадцять. Тобто загорання лампочки – роботу пристрою, при присутності.

Лампочка працює, тобто вмикається при фіксації присутності.

У пункті сімнадцять у нас зазначено, якщо давач присутності на кухні подає сигнал присутності людини і давач світла не подає сигнал освітленості, лампочка для кухні вмикається. Також схожа умова у пункті двадцять тільки для лампочки в кімнаті. Якщо давач присутності в кімнаті подає сигнал присутності людини і давач світла не подає сигнал освітленості, лампочка в кімнаті вмикається.

Отже, у нас має вмикатися лампочка, коли немає світла і помічено рух.

```

1 #include <NanItLib.h>
2 int ledPin = P4_3;
3 bool mode = 0;
4 void setup() {
5   NanIt_Base_Start(); // Ініціалізація NanIt (NanIt initialization)
6   Serial.begin(9600); // Ініціалізація монітору порта (Serial monitor initialization)
7   pinMode(ledPin, OUTPUT);
8   attachInterrupt(digitalPinToInterrupt(P4_2), pir_func, RISING);
9 }
10
11 void loop() {
12
13   bool digitalPhresValue = digitalRead(P4_1); // Зчитування цифрового сигналу(Read digital signal)
14   bool l = digitalPhresValue;
15   Serial.print("D:");
16   Serial.println(l); // Вивід цифрового значення (Output digital value)
17   delay(1000); // Затримка 100 мілісекунд (Delay for 100 milliseconds)
18   if(l>0){
19     digitalWrite(ledPin, 1);
20     if (mode == 1 ){
21       Serial.println("Motion detected!");
22       digitalWrite(ledPin, 1); // вмикаємо напругу для включення світлодіода(turn on the voltage to turn on the LED)
23       delay(1000);
24       mode = 0;
25     }
26   }
27   else
  
```

Вивід Монітор порту X

Повідомлення (Введіть повідомлення для відправки до 'Arduino Mega or Mega 2560' на '/dev/cu.usbserial-1420') Обидва "Новий рядок" та "Повернення коретки" 9600 бод

```

12:26:52,374 -> D:0
12:26:53,366 -> D:0
12:26:54,357 -> D:0
12:26:55,349 -> D:0
12:26:56,374 -> D:0
12:26:57,367 -> D:0
12:26:58,358 -> D:0
12:26:59,349 -> D:1
12:27:00,372 -> D:1
12:27:01,364 -> D:1
12:27:02,356 -> D:1
12:27:03,376 -> D:1
12:27:04,355 -> D:0
  
```

Ln 19, Col 3 Arduino Mega or Mega 2560 на /dev/cu.usbserial-1420

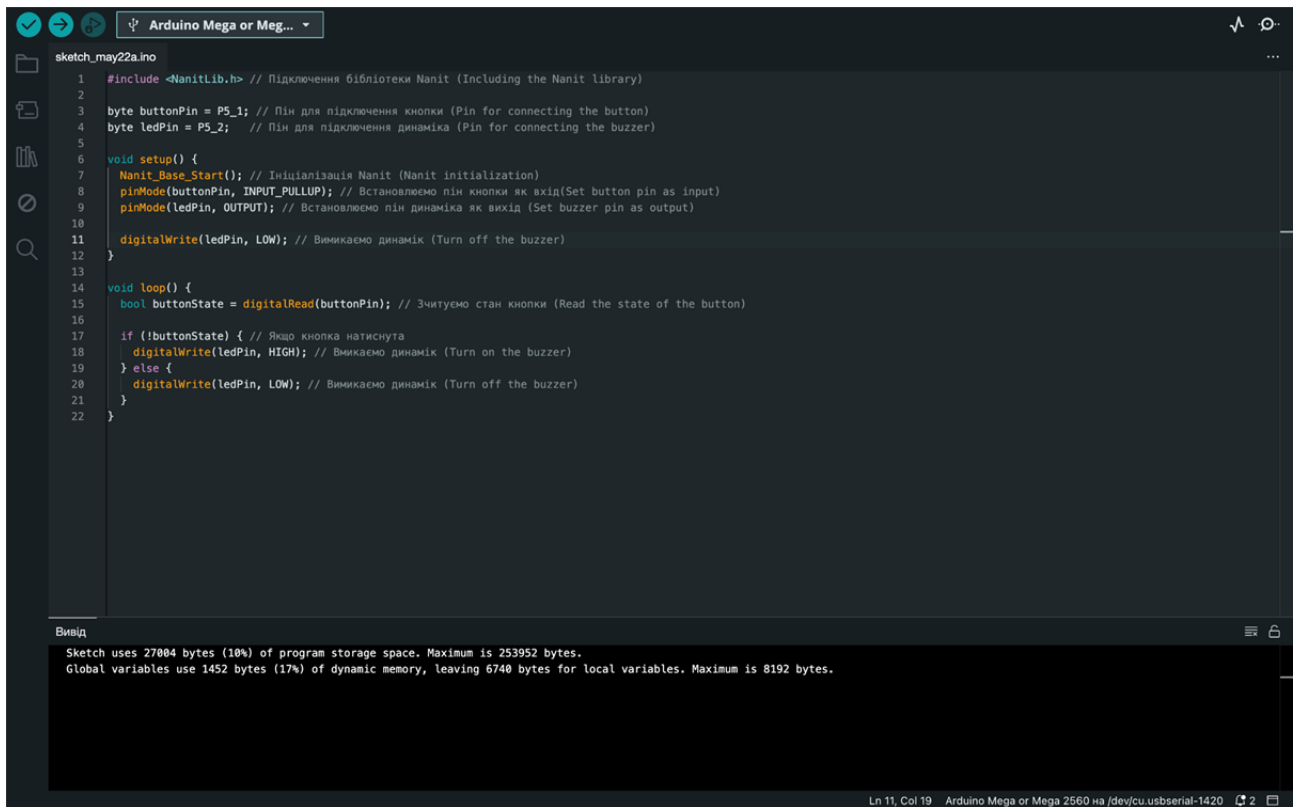
Рис. 5.20 скріншот виконання сімнадцятої та двадцятої частини програми у середовищі програмування Arduino IDE, тобто код та виведення чи світло чи ні, якщо світло, то 0, якщо темно 1.



Рис. 5.21 робота частини моделі під пункт сімнадцять та двадцять. Тобто загорання лампочки, при присутності та відсутності світла навколо.

Лампочка працює, тобто загорається при русі, якщо немає світла.

У пункті двадцять один зазначено при натисканні на кнопку в кімнаті лампочка в кімнаті вмикається/вимикається незалежно. Тобто лампочка вмикається при натисканні на кнопку.



```
sketch_may22a.ino
1 #include <NanitLib.h> // Підключення бібліотеки Nanit (Including the Nanit library)
2
3 byte buttonPin = P5_1; // Пін для підключення кнопки (Pin for connecting the button)
4 byte ledPin = P5_2; // Пін для підключення динаміка (Pin for connecting the buzzer)
5
6 void setup() {
7   Nanit_Base_Start(); // Ініціалізація Nanit (Nanit initialization)
8   pinMode(buttonPin, INPUT_PULLUP); // Встановлюємо пін кнопки як axid(Set button pin as input)
9   pinMode(ledPin, OUTPUT); // Встановлюємо пін динаміка як вихід (Set buzzer pin as output)
10
11   digitalWrite(ledPin, LOW); // Вимикаємо динамік (Turn off the buzzer)
12 }
13
14 void loop() {
15   bool buttonState = digitalRead(buttonPin); // Зчитуємо стан кнопки (Read the state of the button)
16
17   if (!buttonState) { // Якщо кнопка натиснута
18     digitalWrite(ledPin, HIGH); // Вмикаємо динамік (Turn on the buzzer)
19   } else {
20     digitalWrite(ledPin, LOW); // Вимикаємо динамік (Turn off the buzzer)
21   }
22 }
```

Вивід

Sketch uses 27804 bytes (18%) of program storage space. Maximum is 253952 bytes.
Global variables use 1452 bytes (17%) of dynamic memory, leaving 6748 bytes for local variables. Maximum is 8192 bytes.

Ln 11, Col 19 Arduino Mega or Mega 2560 на /dev/cu.usbserial-1420

Рис. 5.22 скріншот виконання двадцять першої частини програми у середовищі програмування Arduino IDE, тобто код, адже немає виведення в Serial Port.

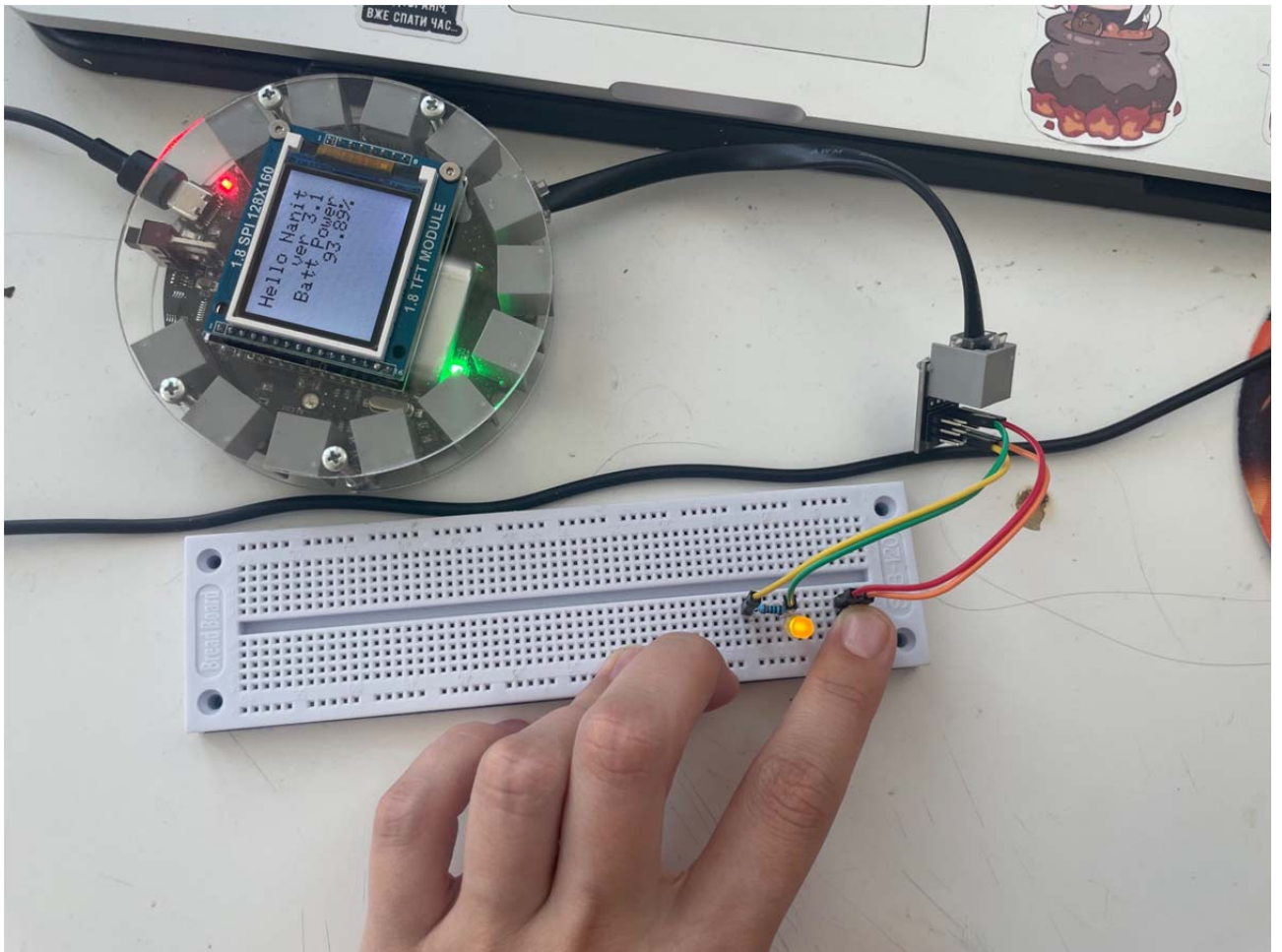


Рис. 5.23 робота частини моделі під пункт двадцятьодин. Тобто загорання лампочки, при натисканні на кнопку.

Лампочка загоряється при натисненні на кнопку.

2.6 Розрахунок економічної вигоди від впровадження системи "розумний дім" для однокімнатної квартири.

Детальний розрахунок економії та окупності системи "розумний дім"

1. Економія електроенергії:

- Посудомийка (4 кВт/год):

Пік: $4 \text{ кВт/год} * 3,96 \text{ грн/кВт.год} * 0,5 \text{ години} = 7,92 \text{ грн}$

Напівпік: $4 \text{ кВт/год} * 2,64 \text{ грн/кВт.год} * 0,5 \text{ години} = 5,28 \text{ грн}$

Економія: $7,92 \text{ грн} - 5,28 \text{ грн} = 2,64 \text{ грн} (33,33\%)$

- Пральна машина (2 кВт/год):

Пік: $2 \text{ кВт/год} * 3,96 \text{ грн/кВт.год} * 1 \text{ година} = 7,92 \text{ грн}$

Напівпік: $2 \text{ кВт/год} * 2,64 \text{ грн/кВт.год} * 1 \text{ година} = 5,28 \text{ грн}$

Економія: $7,92 \text{ грн} - 5,28 \text{ грн} = 2,64 \text{ грн} (33,33\%)$

- Сушильна машина (3 кВт/год):

Пік: $3 \text{ кВт/год} * 3,96 \text{ грн/кВт.год} * 1 \text{ година} = 11,88 \text{ грн}$

Напівпік: $3 \text{ кВт/год} * 2,64 \text{ грн/кВт.год} * 1 \text{ година} = 7,92 \text{ грн}$

Економія: $11,88 \text{ грн} - 7,92 \text{ грн} = 3,96 \text{ грн} (33,33\%)$

- Бойлер (50 л, 2 кВт):

Пік: $2 \text{ кВт/год} * 3,96 \text{ грн/кВт.год} * 4 \text{ години} = 31,68 \text{ грн}$

Напівпік: $2 \text{ кВт/год} * 2,64 \text{ грн/кВт.год} * 4 \text{ години} = 21,12 \text{ грн}$

Економія: $31,68 \text{ грн} - 21,12 \text{ грн} = 10,56 \text{ грн} (33,33\%)$

- Кондиціонер (1,5 кВт):

Пік: $1,5 \text{ кВт/год} * 3,96 \text{ грн/кВт.год} * 2 \text{ години} = 11,88 \text{ грн}$

Економія: $11,88 \text{ грн} * 0,2 = 2,38 \text{ грн} (20\%)$

- Обігрівач (1 кВт):
 Пік: $1 \text{ кВт/год} * 3,96 \text{ грн/кВт.год} * 2 \text{ години} = 7,92 \text{ грн}$
 Економія: $7,92 \text{ грн} * 0,2 = 1,58 \text{ грн} (20\%)$
- Плита (2 кВт):
 Пік: $2 \text{ кВт/год} * 3,96 \text{ грн/кВт.год} * 0,5 \text{ години} = 3,96 \text{ грн}$
 Економія: $3,96 \text{ грн} * 0,1 = 0,396 \text{ грн} (10\%)$
- Праска (1 кВт):
 Пік: $1 \text{ кВт/год} * 3,96 \text{ грн/кВт.год} * 0,5 \text{ години} = 1,98 \text{ грн}$
 Економія: $1,98 \text{ грн} * 0,1 = 0,198 \text{ грн} (10\%)$
- Світло (50 Вт):
 Пік: $0,05 \text{ кВт/год} * 3,96 \text{ грн/кВт.год} * 12 \text{ годин} = 2,376 \text{ грн}$
 Економія: $2,376 \text{ грн} * 0,5 = 1,188 \text{ грн} (50\%)$

2. Загальна економія за день:

$2,64 \text{ грн} + 2,64 \text{ грн} + 3,96 \text{ грн} + 10,56 \text{ грн} + 2,38 \text{ грн} + 1,58 \text{ грн} + 0,396 \text{ грн} + 0,198 \text{ грн} + 1,188 \text{ грн} = 26,12 \text{ грн}$

3. Економія за місяць:

$26,12 \text{ грн/день} * 30 \text{ днів} = 783,6 \text{ грн}$

4. Економія за рік:

$783,6 \text{ грн/місяць} * 12 \text{ місяців} = 9403,2 \text{ грн}$

5. Порівняння з витратами:

- Вартість системи: 10 000 грн
- Вартість обслуговування: 100 грн/рік

6. Окупність інвестицій:

$(10\,000 \text{ грн} + 100 \text{ грн}) / 9403,2 \text{ грн} = 1,06 \text{ років}$

ВИСНОВОК

В рамках цієї дипломної роботи було проаналізовано різні схеми реалізацій "розумного дому". Були проаналізовані різні моделі та платформи розумного дому, а також їхні ключові характеристики, переваги та недоліки.

В ході дослідження було визначено, що "розумний дім" пропонує ряд суттєвих переваг, таких як:

- Підвищення комфорту та зручності: автоматизація рутинних завдань та можливість керувати пристроями за допомогою смартфона або голосових команд значно економлять час та зусилля.
- Економія енергії та ресурсів: розумні системи можуть оптимізувати споживання енергії, води та інших ресурсів, що призводить до зниження витрат.
- Посилення безпеки: інтегровані системи безпеки та моніторингу можуть захистити ваш дім від вторгнень, пожеж та інших загроз.
- Доступність та інклюзивність: розумні технології можуть допомогти людям з обмеженими можливостями вести більш самостійне та комфортне життя.

Розроблена власна оригінальна схема моделі "Розумний дім" на основі однокімнатної квартири з кухнею, кладовкою та санвузлом підлаштовану під використання в Україні. Це модель режиму певної системи розумного житла, яку може для себе обрати користувач, як стандартну, вона не потребує якогось доналаштування користувачем чи розробником, інтерфейс системи є максимально простим в розробці та в використанні. Перевагами моделі є економія енергії та ваших коштів,

захист від пожежі, протікань та охоронна система. Також ви можете побачити роботу вашого розумного житла через веб-сторінку, де все розписано українською мовою.

Економія коштів відбувається за рахунок роботи пристроїв, що багато споживають енергії у час дешевого тарифу. Все підлаштовано під тарифи в Україні.

Економія енергії відбувається за рахунок вимкнення світла під час відсутності людини в кімнаті.

Захист від пожежі реалізований, дякуючи тому, що легкозаймисті прилади вимикаються під час відсутності людини в кімнаті, це такі прилади як праска, плита.

Захист від протікань реалізований, дякуючи тому, що крани перекриваються при високій вологості також кран для санвузла та рукомийника на кухні перекритий при відсутності людини поруч. Та кран пральної та посудомийної машинки перекритий, коли вони не працюють.

Охоронна система реалізована, дякуючи, сигналізації. Тобто власник ставить квартиру на охорону при покиданні житла і в цьому режимі при фіксації присутності в кімнаті когось спрацьовує сигналізація, тобто вмикається сирена.

Створена принципова схема і розводка проєкту. Тобто місцезнаходження кожного приладу для нашого проєкту.

Даний проєкт було протестовано та намічено як можливо його покращити та реалізувати на реальній квартирі.

Отже, ця модель режиму є зручна для середньостатистичного власника однокімнатної квартири в Україні. Дякуючи тому, що вона розроблена для українського користувача з простим інтерфейсом та мінімум налаштування.

СПИСОК ЛІТЕРАТУРИ

1. Рейтинг систем "розумний дім": https://www.moyo.ua/ua/news/chto_takoe_sistema_umnyu_dom_9_fishek_dlya_komforta.html
2. Посібник з вибору системи "розумний дім": <https://ek.ua/ua/post/1990/618-chto-takoe-umnyu-dom-funkcii-vidy-sostavlyayuschie-i-ekosistemy/>
3. Кращі системи "Розумний будинок" по виробниках 2024 року. ТОП 5 надійних та якісних систем "Розумний будинок" рейтингу: <https://vencon.ua/ua/articles/rejting-sistem-umnyu-dom-proizvoditelyam>
4. <https://homesmart.com.ua/ekosistema-tuya-smart-smart-life/>
5. Що таке Tuya Smart?: <https://prolum.com.ua/rozumnyi-budynok-prostymy-slovamy-poyasniuiemo-shcho-take-tuya-smart/>
6. Нова ціна електроенергії для населення (червень 2023): <https://energozbut.ck.ua/news/728-nova-cina-elektroenergiyi-dlya-naselennya.html>
7. Patrascu M. Integrating Services and Agents for Control and Monitoring: Managing Emergencies in Smart Buildings. Service Orientation in Holonic and Multi- Agent Manufacturing and Robotics. / Patrascu., 2014.
8. Котунова, Д. Г. Огляд DIY елементів для систем «Smart Home» / Д. Г. Котунова, О. М. Павловський // XIII Науково-практична конференція студентів, аспірантів та молодих вчених «Погляд у майбутнє приладобудування», 13-14 травня 2020 р., м. Київ, Україна: збірник праць конференції. – Київ : КПІ ім. Ігоря Сікорського, 2020.
9. Моніт Я.В. Система «Розумний будинок» з відкритим програмним забезпеченням/ Я.В.Моніт // XIX науково-технічна конференція

- студентів та молодих учених «Гіротехнології, навігація, керування рухом та конструювання авіаційно-космічної техніки», 15-16 лютого 2016 р. – К.: «Політехніка», 2016.
10. Becker J., Kugeler M., and Rosemann M. Process Management: A Guide for the Design of Business Processes. Springer, 2003.
 11. Gardiner Tom. Automated Benchmarking of Description Logic Reasoners / Gardiner Tom, Ian Horrocks, Dmitry Tsarkov. - Description Logics Workshop 2006.
 12. Granzer W. P. Security in Building Automation Systems / Wolfgang Praus Granzer. Munich: Apress, 2018.
 13. Hähnle, R. Tableaux and Related Methods. Handbook of Automated Reasoning / Hähnle, R. – Volume I. Elsevier science, 2001.
 14. Hayes-Roth F. - Building Expert Systems / Hayes-Roth F., Waterman D., Lenat D.- Addison-Wesley, 1983.
 15. Kaczor K. Overview of Expert System Shells / Kaczor K., Szymon B., Grzegorz J. - Krakow, Poland: Institute of Automatics: AGH University of Science and Technology, Poland, 5 December 2010.

ДОДАТОК А: Схеми електричних з'єднань.

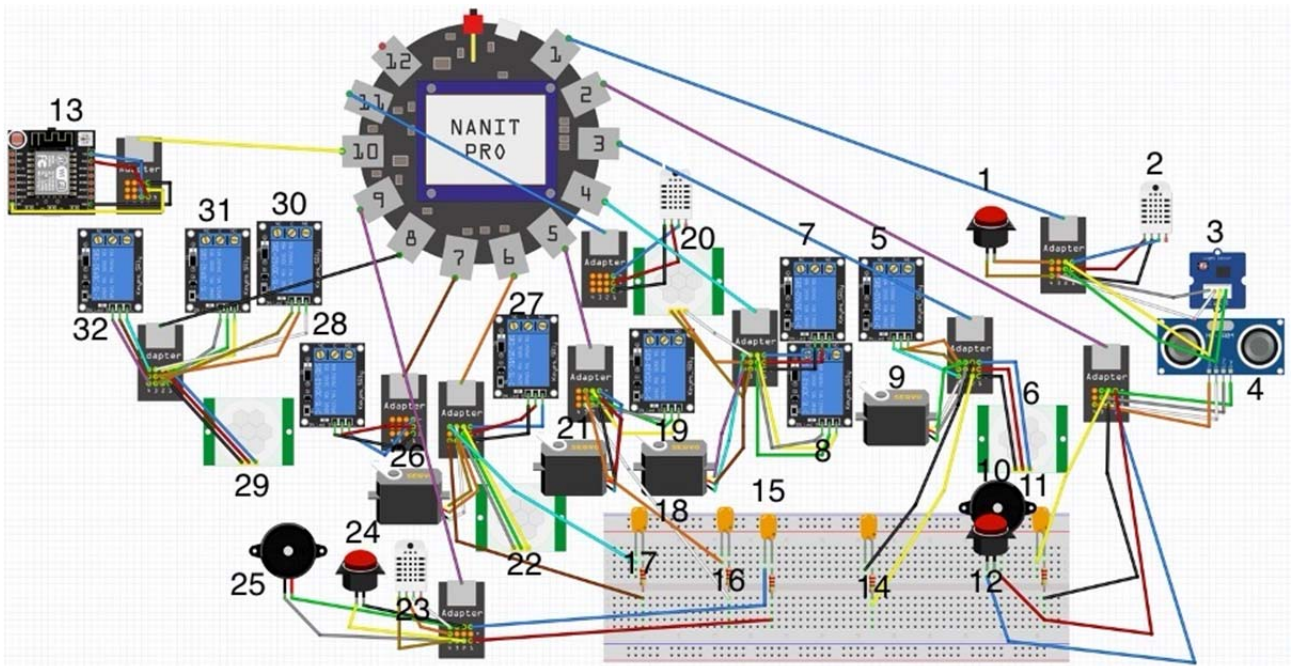


Рис. А1.1 схема розумного дому з усіма компонентами зроблена у симуляторі Fritzing. Легенда:

- 1 – кнопка-охорона,
- 2 – датчик температури на вулиці,
- 3 – датчик світла,
- 4 – датчик руху коридор,
- 5 – реле для ввімкнення/вимкнення бойлера,
- 6 – датчик присутності санвузол,
- 7 – реле для ввімкнення/вимкнення посудомийки,
- 8 – реле для ввімкнення/вимкнення плити,
- 9 – сервомотор для ввімкнення/вимкнення води для санвузла,
- 10 + 12 – п'єзоелемент + кнопка – дверний дзвінок,
- 11 – світлодіод коридор,
- 13 – мікроконтролер-інтернетточка для забезпечення комунікації пристроїв між собою та сайту, сервісу управління пристроями,
- 14 – світлодіод санвузол,

- 15 – світлодіод кімната,
- 16 – світлодіод кухня,
- 17 – світлодіод кладовка,
- 18 – сервомотор для ввімкнення/вимкнення води для посудомийки,
- 19 – реле для ввімкнення/вимкнення холодильника, 20 – давач присутності
кухня,
- 21 – сервомотор для ввімкнення/вимкнення води на кухні,
- 22 – давач присутності кладовка,
- 23 – давач вологості для санвузла,
- 24 – кнопка для вимкнення світла в спальні самостійно,
- 25 – п'єзоелемент-сигналізація,
- 26 – сервомотор для ввімкнення/вимкнення води в пральній машині,
- 27 – реле для ввімкнення/вимкнення праски,
- 28 – реле для ввімкнення/вимкнення пральна машинка,
- 29 – давач присутності кімната,
- 30 – реле для ввімкнення/вимкнення кондиціонера,
- 31 – реле для ввімкнення/вимкнення обігрівача,
- 32 – реле для ввімкнення/вимкнення провітрювача.

**ДОДАТОК Б: План розумного дому розроблений на плані
однокімнатної квартири.**



Рис. Б1.1 розміщення приладів у квартирі. У коридорі у нас встановлено сигналізацію в яку входить кнопка, п'єзоелемент, давач руху, який

знаходиться в кімнаті, wifi, тобто блок який зв'язує прилади між собою та з якого завантажується сайт для спілкування з користувачем, лампочка, давач руху. У санвузлі у нас встановлено сервомотор, який перекриває кран, лампочка, давач присутності, реле (розумна розетка) для управління бойлером. На кухні у нас встановлено реле (розумна розетка) для управління холодильником, реле (розумна розетка) для управління посудомийкою, давач присутності, сервомотор, який перекриває кран в умивальнику, сервомотор, який перекриває кран на посудомийці, реле (розумна розетка) для управління плитою, лампочка. У кладовці у нас розміщено реле (розумна розетка) для управління пралкою та сушилкою, давач присутності, сервомотор, який перекриває кран на пралці, реле (розумна розетка) для управління праскою, лампочка. У кімнаті у нас розміщено давач присутності, реле (розумна розетка) для управління кондиціонером, реле (розумна розетка) для управління обігрівачем, реле (розумна розетка) для управління провітрювачем, лампочка, кнопка для самостійного вмикання світла.

ДОДАТОК В. Код до програми.

Код для пункту 1:

```
#include <NanitLib.h>
int ledPin = P5_3;
int buzzPin = P5_1;
byte buttonPin = P5_4;
bool mode = 0;
void setup() {
  Nanit_Base_Start();
  // put your setup code here, to run once:
  pinMode(ledPin, OUTPUT);
  attachInterrupt(digitalPinToInterrupt(P5_2), pir_func, RISING);
  pinMode(buttonPin, INPUT_PULLUP); // Встановлюємо пін кнопки як
вхід(Set button pin as input)
  pinMode(buzzPin, OUTPUT); // Встановлюємо пін динаміка як вихід (Set
buzzer pin as output)
  digitalWrite(ledPin, HIGH); }
void loop() {
  bool buttonState = digitalRead(buttonPin); // Зчитуємо стан кнопки (Read the
state of the button)
  if (!buttonState) { // Якщо кнопка натиснута
    digitalWrite(ledPin, LOW); // Вмикаємо динамік (Turn on the buzzer)
    if (mode == 1 ) { // коли спрацювало переривання...(when the interrupt was
fired...)
      Serial.println("Motion detected!");
      analogWrite(buzzPin, 100); // Виводимо на п'єзодинамік
аналогове значення 100.
      delay(500); // Затримка на 500 мілісекунд.
```

```

    analogWrite(buzzPin , 180);           // Виводимо на п'єзодинамік
аналогове значення 180.
    delay(500);                          // Затримка на 500 мілісекунд.
    analogWrite(buzzPin , 255);          // Виводимо на п'єзодинамік
аналогове значення 255(максимальне).
    delay(500); } else {
    mode = 0; } } else {
    digitalWrite(ledPin, HIGH); // Вимикаємо динамік (Turn off the buzzer)
} // put your main code here, to run repeatedly: }
void pir_func() {
    mode = 1; // перемикаємо режим для включення сигналізації (switch the
mode to turn on the alarm) }

```

Код для пункту 2:

```

#include <NanitLib.h>
int ledPin = P5_3;
int buzzPin = P5_1;
byte buttonPin = P5_4;
bool mode = 0;
void setup() {
    Nanit_Base_Start();
    // put your setup code here, to run once:
    pinMode(ledPin, OUTPUT);
    attachInterrupt(digitalPinToInterrupt(P5_2), pir_func, RISING);
    pinMode(buttonPin, INPUT_PULLUP); // Встановлюємо пін кнопки як
вхід(Set button pin as input)
    pinMode(buzzPin, OUTPUT); // Встановлюємо пін динаміка як вихід (Set
buzzer pin as output)
    digitalWrite(ledPin, HIGH); }
void loop() {

```

```

bool buttonState = digitalRead(buttonPin); // Зчитуємо стан кнопки (Read the
state of the button)
if (!buttonState) { // Якщо кнопка натиснута
    digitalWrite(ledPin, LOW); // Вмикаємо динамік (Turn on the buzzer)
    if (mode == 1 ) { // коли спрацювало переривання...(when the interrupt was
fired...)
        Serial.println("Motion detected!");
        analogWrite(buzzPin, 100); // Виводимо на п'єзодинамік
аналогове значення 100.
        delay(500); // Затримка на 500 мілісекунд.
        analogWrite(buzzPin , 180); // Виводимо на п'єзодинамік
аналогове значення 180.
        delay(500); // Затримка на 500 мілісекунд.
        analogWrite(buzzPin , 255); // Виводимо на п'єзодинамік
аналогове значення 255(максимальне).
        delay(500); } else {
            mode = 0; } } else {
                digitalWrite(ledPin, HIGH); // Вимикаємо динамік (Turn off the buzzer) } //
put your main code here, to run repeatedly:}

void pir_func() {
    mode = 1; // перемикаємо режим для включення сигналізації (switch the
mode to turn on the alarm)}

    Код для пункту 3:
#include <NanitLib.h>

int ledPin = P5_3;
int buzzPin = P5_1;
byte buttonPin = P5_4;
bool mode = 0;
void setup() {

```

```

Nanit_Base_Start();
// put your setup code here, to run once:
pinMode(ledPin, OUTPUT);
attachInterrupt(digitalPinToInterrupt(P5_2), pir_func, RISING);
pinMode(buttonPin, INPUT_PULLUP); // Встановлюємо пін кнопки як
вхід(Set button pin as input)
pinMode(buzzPin, OUTPUT); // Встановлюємо пін динаміка як вихід (Set
buzzer pin as output)
digitalWrite(ledPin, HIGH); }
void loop() {
bool buttonState = digitalRead(buttonPin); // Зчитуємо стан кнопки (Read the
state of the button)
if (!buttonState) { // Якщо кнопка натиснута
digitalWrite(ledPin, LOW); // Вмикаємо динамік (Turn on the buzzer)
if (mode == 1 ) { // коли спрацювало переривання...(when the interrupt was
fired...)
Serial.println("Motion detected!");
analogWrite(buzzPin, 100); // Виводимо на п'єзодинамік
аналогове значення 100.
delay(500); // Затримка на 500 мілісекунд.
analogWrite(buzzPin , 180); // Виводимо на п'єзодинамік
аналогове значення 180.
delay(500); // Затримка на 500 мілісекунд.
analogWrite(buzzPin , 255); // Виводимо на п'єзодинамік
аналогове значення 255(максимальне).
delay(500); } else {
mode = 0; }} else {
digitalWrite(ledPin, HIGH); // Вимикаємо динамік (Turn off the buzzer)
} // put your main code here, to run repeatedly:}

```

```

void pir_func() {
    mode = 1; // перемикаємо режим для включення сигналізації (switch the
mode to turn on the alarm)}

    Код для пункту 4:
#include <NanitLib.h>
#include <Servo.h>
int c = 0;
bool mode = 0;
int ledPin = P5_3;
Servo servo;
void setup() {
    Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
    Serial.begin(9600);
    // put your setup code here, to run once:
    attachInterrupt(digitalPinToInterrupt(P5_2), pir_func, RISING); // ініціалізація
переривання(interrupt initialization)
    pinMode(ledPin, OUTPUT); // Світлодіод як вихід(LED as an output)
    servo.attach(P5_1); }
void time () {
for (int c = 0; c < 24; c++) {
    Serial.print("Лічильник: ");
    Serial.println(c); // Виводимо значення лічильника в моніторі порту
    if (mode == 1 ) { // коли спрацювало переривання...(when the interrupt was
fired...)
        Serial.println("Motion detected!");
        digitalWrite(ledPin, 1);
        delay(1000);
        servo.write(0); // Поворот сервоприводу на 180 градусів.
        delay(500);
    }
}
}

```

```

servo.write(180); // Поворот сервоприводу на 180 градусів.
delay(500); } else {
digitalWrite(ledPin, 0);
mode = 0; }
if ( c < 8 || c > 11 || c > 22) {
  if(c < 20) {
    digitalWrite(ledPin, 1); // вмикаємо напругу для включення
світлодіода(turn on the voltage to turn on the LED)
    // чекаємо 1 сек(wait 1 sec)
    servo.write(0); // Поворот сервоприводу на 180 градусів.
    delay(500);
    servo.write(180); // Поворот сервоприводу на 180 градусів.
    delay(500); } else {
      digitalWrite(ledPin, 0); }} else {
      digitalWrite(ledPin, 0);}
    delay(60000); } }
void pir_func() {
  mode = 1; // перемикаємо режим для включення сигналізації (switch the
mode to turn on the alarm) }
void loop() {
  time(); }

```

Код для пункту 5:

```

#include <NanitLib.h>
#include <Servo.h>
int c = 0;
bool mode = 0;
int ledPin = P5_3;
Servo servo;
void setup() {

```

```

Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
Serial.begin(9600);
// put your setup code here, to run once:
attachInterrupt(digitalPinToInterrupt(P5_2), pir_func, RISING); // ініціалізація
переривання(interrupt initialization)
pinMode(ledPin, OUTPUT); // Світлодіод як вихід(LED as an output)
servo.attach(P5_1); }
void time () {
for (int c = 0; c < 24; c++) {
    Serial.print("Лічильник: ");
    Serial.println(c); // Виводимо значення лічильника в моніторі порту
    if (mode == 1 ) { // коли спрацювало переривання...(when the interrupt was
fired...)
        Serial.println("Motion detected!");
        digitalWrite(ledPin, 1);
        delay(1000);
        servo.write(0); // Поворот сервоприводу на 180 градусів.
        delay(500);
        servo.write(180); // Поворот сервоприводу на 180 градусів.
        delay(500); }
    else {
        digitalWrite(ledPin, 0);
        mode = 0; }
    if ( c < 8 || c > 11 || c > 22) {
        if(c < 20){
            digitalWrite(ledPin, 1); // вмикаємо напругу для включення
світлодіода(turn on the voltage to turn on the LED)
            // чекаємо 1 сек(wait 1 sec)
            servo.write(0); // Поворот сервоприводу на 180 градусів.

```

```

    delay(500);
    servo.write(180); // Поворот сервоприводу на 180 градусів.
    delay(500); } else {
    digitalWrite(ledPin, 0); } } else {
    digitalWrite(ledPin, 0); }
    delay(60000);    } }
void pir_func() {
    mode = 1; // перемикаємо режим для включення сигналізації (switch the
mode to turn on the alarm) }
void loop() {
    time(); }

```

Код для пункту 6:

```

#include <NanitLib.h>
int ledPin = P5_1;
byte buttonPin = P5_2;
int c = 0;
void setup() {
    Nanit_Base_Start();
    Serial.begin(9600);
    // put your setup code here, to run once:
    pinMode(ledPin, OUTPUT);
    pinMode(buttonPin, INPUT_PULLUP); // Встановлюємо пін кнопки як
вхід(Set button pin as input) }
void time () {
for (int c = 0; c < 24; c++) {
    Serial.print("Лічильник: ");
    Serial.println(c); // Виводимо значення лічильника в моніторі порту
    bool buttonState = digitalRead(buttonPin);
    if ( c > 6 || !buttonState) {

```

```

digitalWrite(ledPin, HIGH); } else {
digitalWrite(ledPin, LOW); // Вимикаємо динамік (Turn off the buzzer) } //
put your main code here, to run repeatedly:
delay(60000); } }
void loop() {
time(); }

```

Код для пункту 7:

```

#include <NanitLib.h>
#include <Servo.h>
int c = 0;
bool mode = 0;
int ledPin = P5_3;
Servo servo;
void setup() {
Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
Serial.begin(9600);
// put your setup code here, to run once:
attachInterrupt(digitalPinToInterrupt(P5_2), pir_func, RISING); // ініціалізація
переривання(interrupt initialization)
pinMode(ledPin, OUTPUT); // Світлодіод як вихід(LED as an output)
servo.attach(P5_1); }
void time () {
for (int c = 0; c < 24; c++) {
Serial.print("Лічильник: ");
Serial.println(c); // Виводимо значення лічильника в моніторі порту
if (mode == 1 ) { // коли спрацювало переривання...(when the interrupt was
fired...)
Serial.println("Motion detected!");
digitalWrite(ledPin, 1);

```

```

delay(1000);
servo.write(0); // Поворот сервоприводу на 180 градусів.
delay(500);
servo.write(180); // Поворот сервоприводу на 180 градусів.
delay(500); } else {
digitalWrite(ledPin, 0);
mode = 0; }
if ( c < 8 || c > 11 || c > 22) {
  if(c < 20) {
    digitalWrite(ledPin, 1); // вмикаємо напругу для включення
світлодіода(turn on the voltage to turn on the LED)
    // чекаємо 1 сек(wait 1 sec)
    servo.write(0); // Поворот сервоприводу на 180 градусів.
    delay(500);
    servo.write(180); // Поворот сервоприводу на 180 градусів.
    delay(500); } else {
      digitalWrite(ledPin, 0); } } else {
    digitalWrite(ledPin, 0); }
    delay(60000); } }
void pir_func() {
  mode = 1; // перемикаємо режим для включення сигналізації (switch the
mode to turn on the alarm) }
void loop() {
  time(); }
    Код для пункту 8:
#include <NanitLib.h>
#include <Servo.h>
int c = 0;
bool mode = 0;

```

```

int ledPin = P5_3;
Servo servo;
void setup() {
  Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
  Serial.begin(9600);
  // put your setup code here, to run once:
  attachInterrupt(digitalPinToInterrupt(P5_2), pir_func, RISING); // ініціалізація
  переривання(interrupt initialization)
  pinMode(ledPin, OUTPUT); // Світлодіод як вихід(LED as an output)
  servo.attach(P5_1); }
void time () {
for (int c = 0; c < 24; c++) {
  Serial.print("Лічильник: ");
  Serial.println(c); // Виводимо значення лічильника в моніторі порту
  if (mode == 1 ) { // коли спрацювало переривання...(when the interrupt was
  fired...)
    Serial.println("Motion detected!");
    digitalWrite(ledPin, 1);
    delay(1000);
    servo.write(0); // Поворот сервоприводу на 180 градусів.
    delay(500);
    servo.write(180); // Поворот сервоприводу на 180 градусів.
    delay(500); } else {
    digitalWrite(ledPin, 0);
    mode = 0; }
  if ( c < 8 || c > 11 || c > 22) {
    if(c < 20) {
      digitalWrite(ledPin, 1); // вмикаємо напругу для включення
світлодіода(turn on the voltage to turn on the LED)

```

```

    // чекаємо 1 сек(wait 1 sec)
    servo.write(0); // Поворот сервоприводу на 180 градусів.
    delay(500);
    servo.write(180); // Поворот сервоприводу на 180 градусів.
    delay(500); } else {
    digitalWrite(ledPin, 0); } } else {
    digitalWrite(ledPin, 0); }
    delay(60000); } }
void pir_func() {
    mode = 1; // перемикаємо режим для включення сигналізації (switch the
mode to turn on the alarm) }
void loop() {
    time(); }

```

Код для пункту 9:

```

#include <NanitLib.h> // підключення бібліотеки(including library)
#include <Servo.h> // Підключення бібліотеки Servo для керування
сервоприводом.
#include <DHT.h>
DHT dht(P5_3, DHT11); // Створення об'єкту(Creating a DHT class object)
Servo servo;
bool mode = 0; // режим переривання(interrupt mode)
void setup() {
    Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
    dht.begin();
    servo.attach(P5_2);
    Serial.begin(9600); // Ініціалізація монітору порта(Serial monitor
initialization)
    attachInterrupt(digitalPinToInterrupt(P5_1), pir_func, RISING); //
ініціалізація переривання(interrupt initialization) }

```

```

void loop() {
  attachInterrupt(digitalPinToInterrupt(P5_1), pir_func, RISING);
  delay(500);
  float h = dht.readHumidity();
  if (mode == 1 ) { // коли спрацювало переривання...(when the interrupt was
  fired...)
    Serial.println("Motion detected!"); // виводимо повідомлення(output the
  message)
    delay(500);
    if (h>80) {
      servo.write(0); // Поворот сервоприводу на 180 градусів.
    delay(500);
    servo.write(180); // Поворот сервоприводу на 180 градусів.
    delay(500);
    Serial.println(h); }
    mode = 0; // повертаємо режим 0(return mode 0) } else {
    mode = 0; } }
void pir_func() {
  mode = 1; // перемикаємо режим для включення сигналізації (switch the
  mode to turn on the alarm) }
  Код для пункту 10:
#include <NanitLib.h> // підключення бібліотеки(including library)
#include <Servo.h> // Підключення бібліотеки Servo для керування
сервоприводом.
#include <DHT.h>
DHT dht(P5_3, DHT11); // Створення об'єкту(Creating a DHT class object)
Servo servo;
bool mode = 0; // режим переривання(interrupt mode)
void setup() {

```

```

Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
dht.begin();
servo.attach(P5_2);
Serial.begin(9600); // Ініціалізація монітору порта(Serial monitor
initialization)
attachInterrupt(digitalPinToInterrupt(P5_1), pir_func, RISING); //
ініціалізація переривання(interrupt initialization) }
void loop() {
attachInterrupt(digitalPinToInterrupt(P5_1), pir_func, RISING);
delay(500);
float h = dht.readHumidity();
if (mode == 1 ) { // коли спрацювало переривання...(when the interrupt was
fired...)
Serial.println("Motion detected!"); // виводимо повідомлення(output the
message)
delay(500);
if (h>80) {
servo.write(0); // Поворот сервоприводу на 180 градусів.
delay(500);
servo.write(180); // Поворот сервоприводу на 180 градусів.
delay(500);
Serial.println(h); }
mode = 0; // повертаємо режим 0(return mode 0) } else {
mode = 0; } }
void pir_func() {
mode = 1; // перемикаємо режим для включення сигналізації (switch the
mode to turn on the alarm) }

```

Код для пункту 11:

```
#include <NanitLib.h> // підключення бібліотеки(including library)
```

```

#include <DHT.h>
DHT dht(P5_1, DHT11); // Створення об'єкту(Creating a DHT class object)
int ledPin = P5_3;
bool mode = 0; // режим переривання(interrupt mode)
void setup() {
  Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
  dht.begin();
  Serial.begin(9600); // Ініціалізація монітору порта(Serial monitor
initialization)
  attachInterrupt(digitalPinToInterrupt(P5_2), pir_func, RISING); //
ініціалізація переривання(interrupt initialization)
  pinMode(ledPin, OUTPUT); }
void loop() {
  float t = dht.readTemperature();// Зчитування вологості з датчика (Reading
humidity from the sensor)
  if (mode == 1 ) { // коли спрацювало переривання...(when the interrupt was
fired...)
    Serial.println("Motion detected!"); // виводимо повідомлення(output the
message)
    delay(500);
    if (t > 5 && t < 15 || t > 27 ) {
      Serial.println(t);
      digitalWrite(ledPin, 1);
      // Вмикаємо реле (Turning on the relay)
      delay(500); // Затримка 3 секунди (Delay for 3 seconds) }
    mode = 0; // повертаємо режим 0(return mode 0) } else {
    mode = 0; } // Вимикаємо реле (Turning off the relay)
    delay(500); // Затримка 3 секунди (Delay for 3 seconds) }
void pir_func() {

```

```
mode = 1; // перемикаємо режим для включення сигналізації (switch the
mode to turn on the alarm) }
```

Код для пункту 12:

```
#include <NanitLib.h> // підключення бібліотеки(including library)
#include <DHT.h>
DHT dht(P5_1, DHT11); // Створення об'єкту(Creating a DHT class object)
int ledPin = P5_3;
bool mode = 0; // режим переривання(interrupt mode)
void setup() {
  Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
  dht.begin();
  Serial.begin(9600); // Ініціалізація монітору порта(Serial monitor
initialization)
  attachInterrupt(digitalPinToInterrupt(P5_2), pir_func, RISING); //
ініціалізація переривання(interrupt initialization)
  pinMode(ledPin, OUTPUT); }
void loop() {
  float t = dht.readTemperature();// Зчитування вологості з датчика (Reading
humidity from the sensor)
  if (mode == 1 ) { // коли спрацювало переривання...(when the interrupt was
fired...)
    Serial.println("Motion detected!"); // виводимо повідомлення(output the
message)
    delay(500);
    if (t < 5) {
      Serial.println(t);
      digitalWrite(ledPin, 1);
      // Вмикаємо реле (Turning on the relay)
      delay(500); // Затримка 3 секунди (Delay for 3 seconds) }
```

```

mode = 0; // повертаємо режим 0(return mode 0) } else {
mode = 0; }
// Вимикаємо реле (Turning off the relay)
delay(500); // Затримка 3 секунди (Delay for 3 seconds) }
void pir_func() {
mode = 1; // перемикаємо режим для включення сигналізації (switch the
mode to turn on the alarm) }
Код для пункту 13:
#include <NanitLib.h>
int c = 0;
int ledPin = P5_2;
void setup() {
Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
Serial.begin(9600);
// put your setup code here, to run once:// ініціалізація переривання(interrupt
initialization)
pinMode(ledPin, OUTPUT); // Світлодіод як вихід(LED as an output) }
void time () {
for (int c = 0; c < 24; c++) {
Serial.print("Лічильник: ");
Serial.println(c); // Виводимо значення лічильника в моніторі порту
delay(60000); // Затримка для уникнення дрібних помилок } }
void loop() {
Serial.println(c);
if ( c < 8 || c > 11 || c > 22 ) {
if ( c < 20 ){
digitalWrite(ledPin, 1); // вмикаємо напругу для включення
світлодіода(turn on the voltage to turn on the LED)
delay(60000); // чекаємо 1 сек(wait 1 sec) } } }

```

Код для пункту 14:

```
#include <NanitLib.h>
int led = P5_2; // пін для світлодіода(pin for the LED)
bool mode = 0; // режим переривання(interrupt mode)
void setup() {
  Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
  Serial.begin(9600); // Ініціалізація монітору порта(Serial monitor
initialization)
  pinMode(led, OUTPUT); // Світлодіод як вихід(LED as an output)
  attachInterrupt(digitalPinToInterrupt(P5_1), pir_func, RISING); //
ініціалізація переривання(interrupt initialization) }
void pir_func() {
  mode = 1; // перемикаємо режим для включення сигналізації (switch the
mode to turn on the alarm) }
void loop() {
  if (mode == 1) { // коли спрацювало переривання...(when the interrupt was
fired...)
    digitalWrite(led, 1); // вимикаємо світлодіод(turn off the LED)
    Serial.println("Motion detected!"); // виводимо повідомлення(output the
message)
    delay(1000); // чекаємо 1 сек(wait 1 sec)
    mode = 0; // повертаємо режим 0(return mode 0) } else {
    digitalWrite(led, 0); // вмикаємо світлодіод(turn on the LED)} }
```

Код для пункту 15:

```
#include <NanitLib.h>
int led = P5_2; // пін для світлодіода(pin for the LED)
bool mode = 0; // режим переривання(interrupt mode)
void setup() {
  Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
```

```

Serial.begin(9600);      // Ініціалізація монітору порта(Serial monitor
initialization)
pinMode(led, OUTPUT); // Світлодіод як вихід(LED as an output)
attachInterrupt(digitalPinToInterrupt(P5_1),  pir_func,  RISING);      //
ініціалізація переривання(interrupt initialization) }
void pir_func() {
  mode = 1; // перемикаємо режим для включення сигналізації (switch the
mode to turn on the alarm)}
void loop() {
  if (mode == 1) { // коли спрацювало переривання...(when the interrupt was
fired...)
    digitalWrite(led, 1); // вимикаємо світлодіод(turn off the LED)
    Serial.println("Motion detected!"); // виводимо повідомлення(output the
message)
    delay(1000); // чекаємо 1 сек(wait 1 sec)
    mode = 0; // повертаємо режим 0(return mode 0) } else {
    digitalWrite(led, 0); // вмикаємо світлодіод(turn on the LED) } }

```

Код для пункту 16:

```

#include <NanitLib.h>
int led = P5_2; // пін для світлодіода(pin for the LED)
bool mode = 0; // режим переривання(interrupt mode)
void setup() {
  Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
  Serial.begin(9600);      // Ініціалізація монітору порта(Serial monitor
initialization)
  pinMode(led, OUTPUT); // Світлодіод як вихід(LED as an output)
  attachInterrupt(digitalPinToInterrupt(P5_1),  pir_func,  RISING);      //
ініціалізація переривання(interrupt initialization) }
void pir_func() {

```

```

    mode = 1; // перемикаємо режим для включення сигналізації (switch the
mode to turn on the alarm)
}
void loop() {
    if (mode == 1) { // коли спрацювало переривання...(when the interrupt was
fired...)
        digitalWrite(led, 1); // вимикаємо світлодіод(turn off the LED)
        Serial.println("Motion detected!"); // виводимо повідомлення(output the
message)
        delay(1000); // чекаємо 1 сек(wait 1 sec)
        mode = 0; // повертаємо режим 0(return mode 0) } else {
        digitalWrite(led, 0); // вмикаємо світлодіод(turn on the LED) }}

```

Код для пункту 17:

```

#include <NanitLib.h>
int ledPin = P4_3;
bool mode = 0;
void setup() {
    Nanit_Base_Start(); // Ініціалізація Nanit (Nanit initialization)
    Serial.begin(9600); // Ініціалізація монітору порта (Serial monitor
initialization)
    pinMode(ledPin, OUTPUT);
    attachInterrupt(digitalPinToInterrupt(P4_2), pir_func, RISING);}
void loop() {
    bool digitalPhresValue = digitalRead(P4_1); // Зчитування цифрового
сигналу(Read digital signal)
    bool l = digitalPhresValue;
    Serial.print("D:");
    Serial.println(l); // Вивід цифрового значення (Output digital value)
    delay(1000); // Затримка 100 мілісекунд (Delay for 100 milliseconds)

```

```

if(l>0){
digitalWrite(ledPin, 1);
if (mode == 1 ){
Serial.println("Motion detected!");
digitalWrite(ledPin, 1); // вмикаємо напругу для включення
світлодіода(turn on the voltage to turn on the LED)
delay(1000);
mode = 0; }} else {
digitalWrite(ledPin, 0);
mode = 0; }}
void pir_func() {
mode = 1; // перемикаємо режим для включення сигналізації (switch the
mode to turn on the alarm)}

```

Код для пункту 18:

```

#include <NanitLib.h>
int led = P5_2; // пін для світлодіода(pin for the LED)
bool mode = 0; // режим переривання(interrupt mode)
void setup() {
Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
Serial.begin(9600); // Ініціалізація монітору порта(Serial monitor
initialization)
pinMode(led, OUTPUT); // Світлодіод як вихід(LED as an output)
attachInterrupt(digitalPinToInterrupt(P5_1), pir_func, RISING); //
ініціалізація переривання(interrupt initialization) }
void pir_func() {
mode = 1; // перемикаємо режим для включення сигналізації (switch the
mode to turn on the alarm) }
void loop() {

```

```

if (mode == 1) { // коли спрацювало переривання...(when the interrupt was
fired...)
    digitalWrite(led, 1); // вимикаємо світлодіод(turn off the LED)
    Serial.println("Motion detected!"); // виводимо повідомлення(output the
message)
    delay(1000); // чекаємо 1 сек(wait 1 sec)
    mode = 0; // повертаємо режим 0(return mode 0) } else {
    digitalWrite(led, 0); // вмикаємо світлодіод(turn on the LED) }}

```

Код для пункту 19:

```

#include <NanitLib.h>
int led = P5_2; // пін для світлодіода(pin for the LED)
bool mode = 0; // режим переривання(interrupt mode)
void setup() {
    Nanit_Base_Start(); // Ініціалізація Nanit(Nanit initialization)
    Serial.begin(9600); // Ініціалізація монітору порта(Serial monitor
initialization)
    pinMode(led, OUTPUT); // Світлодіод як вихід(LED as an output)
    attachInterrupt(digitalPinToInterrupt(P5_1), pir_func, RISING); //
ініціалізація переривання(interrupt initialization)}
void pir_func() {
    mode = 1; // перемикаємо режим для включення сигналізації (switch the
mode to turn on the alarm)}
void loop() {
    if (mode == 1) { // коли спрацювало переривання...(when the interrupt was
fired...)
        digitalWrite(led, 1); // вимикаємо світлодіод(turn off the LED)
        Serial.println("Motion detected!"); // виводимо повідомлення(output the
message)
        delay(1000); // чекаємо 1 сек(wait 1 sec)
    }
}

```

```

mode = 0; // повертаємо режим 0(return mode 0) } else {
digitalWrite(led, 0); // вмикаємо світлодіод(turn on the LED) } }

```

Код для пункту 20:

```

#include <NanitLib.h>
int ledPin = P4_3;
bool mode = 0;
void setup() {
  Nanit_Base_Start(); // Ініціалізація Nanit (Nanit initialization)
  Serial.begin(9600); // Ініціалізація монітору порта (Serial monitor
initialization)
  pinMode(ledPin, OUTPUT);
  attachInterrupt(digitalPinToInterrupt(P4_2), pir_func, RISING); }
void loop() {
  bool digitalPhresValue = digitalRead(P4_1); // Зчитування цифрового
сигналу(Read digital signal)
  bool l = digitalPhresValue;
  Serial.print("D:");
  Serial.println(l); // Вивід цифрового значення (Output digital value)
  delay(1000); // Затримка 100 мілісекунд (Delay for 100 milliseconds)
  if(l>0){
    digitalWrite(ledPin, 1);
    if (mode == 1 ){
      Serial.println("Motion detected!");
      digitalWrite(ledPin, 1); // вмикаємо напругу для включення
світлодіода(turn on the voltage to turn on the LED)
      delay(1000);
      mode = 0; }} else{
    digitalWrite(ledPin, 0);
    mode = 0; }}

```

```
void pir_func() {  
    mode = 1; // перемикаємо режим для включення сигналізації (switch the  
mode to turn on the alarm) }  
  
    Код для пункту 21:  
#include <NanitLib.h> // Підключення бібліотеки Nanit (Including the Nanit  
library)  
byte buttonPin = P5_1; // Пін для підключення кнопки (Pin for connecting the  
button)  
byte ledPin = P5_2;  
void setup() {  
    Nanit_Base_Start(); // Ініціалізація Nanit (Nanit initialization)  
    pinMode(buttonPin, INPUT_PULLUP); // Встановлюємо пін кнопки як  
вхід(Set button pin as input)  
    pinMode(ledPin, OUTPUT);  
    digitalWrite(ledPin, LOW); }  
void loop() {  
    bool buttonState = digitalRead(buttonPin); // Зчитуємо стан кнопки (Read the  
state of the button)  
    if (!buttonState) { // Якщо кнопка натиснута  
        digitalWrite(ledPin, HIGH); } else {  
        digitalWrite(ledPin, LOW); } }
```