

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

Кафедра комп'ютеризованих систем управління

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

_____Литвиненко О.Є.

«__» _____ 2021 р.

**ДИПЛОМНИЙ ПРОЄКТ
(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

**ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ
“БАКАЛАВР”**

Тема: _____ Апаратно-програмна система безпеки в *Smart House*

Виконавець: _____ Лузік М.Р.

Керівник: _____ к.т.н., доцент Росінська Г.П.

Нормоконтролер: _____ Тупота Є.В.

Київ 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет Кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютеризованих систем управління

Спеціальність 123 «Комп'ютерна інженерія»

(шифр, найменування)

Освітньо-професійна програма «Системне програмування»

Форма навчання денна

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Литвиненко О.Є.

«_____» _____ 2021 р.

ЗАВДАННЯ

на виконання дипломної роботи (проєкту)

Лузика Михайла Романовича

(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема дипломної роботи (проєкту) Апаратно-програмна система безпеки в *Smart House*

затверджена наказом ректора від « 04 » лютого 2021 р. № 135/ст.

2. Термін виконання роботи (проєкту): з 17.05.2021 по 20.06.2021

3. Вихідні дані до роботи (проєкту): мікроконтролерна система *Arduino Mega*, інфрачервоний датчик руху *HC-SR501*, GSM-модуль *SIM800L*, 12-кнопкова клавіатура, *lcd* дисплей, п'єзоелектричний випромінювач звуку, середовище *Arduino IDE*, мобільний пристрій сім-карта

4. Зміст пояснювальної записки: _____

1) Аналіз області дослідження;

2) Огляд апаратної та програмної частини проєкту;

3) Опис процесу розробки системи.

5. Перелік обов'язкового графічного (ілюстративного) матеріалу:

1) Позначення основних вузлів плати *Arduino*

2) Інтерфейс *Arduino IDE*

3) Модуль підключення пристроїв до *Arduino Mega*

4) Результат запису і зчитування даних з енергонезалежної пам'яті

5) Результат роботи системи при виявленні руху

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1	Провести аналіз літератури за темою дипломної роботи та дослідити існуючі системи та технології	17.05.2021 – 20.05.2021	
2	Написати перший розділ	20.05.2021 – 23.05.2021	
3	Провести огляд різновиду мікроконтролерів і датчиків до них	23.05.2021 – 25.05.2021	
4	Написати другий розділ	25.05.2021 – 26.05.2021	
5	Розробити апаратну та програмну частини системи	27.05.2021 – 30.05.2021	
6	Провести тестування роботи системи	30.05.2021 – 31.05.2021	
7	Написати третій розділ	31.05.2021 – 02.06.2021	
8	Оформити пояснювальну записку	02.06.2021 – 04.06.2021	
9	Підготувати графічний матеріал	04.06.2021 – 05.06.2021	

7. Дата видачі завдання: “ 17 ” травня 2021 р.

Керівник дипломного проекту _____ Росінська Г.П.
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання _____ Лузік М.Р.
(підпис випускника) (П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до дипломного проєкту «Апаратно-програмна система безпеки в *Smart House*»: 54 сторінки, 29 рисунків, 15 літературних джерел, 1 додаток.

Ключові слова: система безпеки, датчик руху, *gsm*, мікроконтролер, *arduino*.

Об'єкт дослідження – системи безпеки.

Предмет дослідження – апаратно-програмна система безпеки в *Smart House*.

Мета дипломного проєкту – створення системи безпеки, яка буде виявляти порушення безпеки об'єкта та інформування про це власника.

Методи дослідження – визначення основних правил налаштування мікроконтролерів.

Прогнозні припущення щодо розвитку об'єкта дослідження – створення робочого зразка системи та забезпечення безпеки житла.

Результати дипломної роботи рекомендується використовувати при проєктуванні нових пристроїв та систем, що мають схоже призначення та опираються на ті ж самі технології розробки.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	6
ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ОБЛАСТІ ДОСЛІДЖЕННЯ	9
1.1. Поняття системи безпеки розумного будинку	9
1.2. Дротові та бездротові технології	10
1.3. Огляд існуючих аналогів	12
1.4. Висновки до розділу	15
РОЗДІЛ 2 ОГЛЯД АПАРАТНОЇ ТА ПРОГРАМНОЇ ЧАСТИНА ПРОЄКТУ	17
2.1. Arduino – як платформа проєктування	17
2.2. Середовище розробки Arduino IDE	20
2.3. Програмування Arduino	22
2.4. Плати Arduino	24
2.5. Датчики та пристрої	29
2.6. Висновки до розділу	32
РОЗДІЛ 3 ОПИС ПРОЦЕСУ РОЗРОБКИ СИСТЕМИ	34
3.1. Проєктування та розробка апаратної частини системи	34
3.2. Тестування системи	44
3.3. Висновки до розділу	50
ВИСНОВКИ	51
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ	53
ДОДАТОК А	Ошибка! Закладка не определена.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

IDE – Integrated Development Environment

EEPROM – Electrically Erasable Programmable Read-Only Memory

UART – Універсальний асинхронний приймач-передавач (Universal Asynchronous Receiver-Transmitter)

SDA – (Serial Data) шина послідовних даних

SCL – (Serial Clock) шина тактування

GSM (Global System for Mobile Communications) – глобальний стандарт цифрового мобільного стільникового зв'язку з розділенням каналів за часом і частоті.

ВСТУП

Системи домашньої автоматизації в даний час мають велику популярність в світі. Їх будують з нуля, купують і інсталиують в процесі ремонту, а також збирають з доступних комплектуючих.

Пожежі, несправності в системах подачі води і недосконалі системи охорони стають причиною серйозних матеріальних втрат. Системи безпеки спрямовані на вирішення цих проблем.

«Розумний будинок» – це єдина система управління і контролю комфортом і безпекою будинку і його мешканців. Автоматичну роботу, контроль і регулювання інженерних систем будинку, контроль і охорону території в режимі реального часу і багато іншого забезпечить система безпеки розумного будинку. Питання впровадження в будинок такої системи не є абсолютно новим і революційним. На поточний час існують різні варіанти виконань систем безпеки.

Метою роботи є розробка системи безпеки розумного будинку, в основі якої буде здійснення охоронних функцій. Система буде здійснювати захист від проникнення сторонніх осіб у будинок.

Об'єкт дослідження – системи безпеки.

Предмет дослідження – апаратно-програмна система безпеки в Smart House.

Методи дослідження – визначення основних правил налаштування мікроконтролерів.

Метою дипломного проєкту є створення апаратно-програмної системи безпеки в *Smart House*. Для досягнення поставленої мети необхідно вирішити наступні задачі:

- провести аналіз існуючих систем безпеки;
- провести аналіз різних типів мікроконтролерів на базі *Arduino*;
- провести аналіз пристороїв, які необхідні для розробки системи;
- провести аналіз інструментів для проєктування програмної частини проєкту;
- написати програмний код для апаратно-програмної системи безпеки;

Кінцевий продукт повинен являти собою апаратно-програмну систему безпеки, яка буде реалізовувати такі функції:

- виявлення руху;
- активація звукової сигналізації;
- віддалене керування за допомогою мобільного пристрою;
- оповіщення власника про виявлений рух;

Виявлення руху буде здійснюватись за допомогою інфрачервоного датчику. Активація системи буде здійснюватись за допомогою введення коду. При появі руху система буде оповіщувати власника житла, відправляючи *SMS*-повідомлення на його номер телефону. Користувачу також доступні команди, для віддаленого керування системою, які необхідно відправити на номер сім-карти, що встановлюється в систему. Основною перевагою системи безпеки буде її низька ціна, швидке оповіщення та віддалене керування.

РОЗДІЛ 1

АНАЛІЗ ОБЛАСТІ ДОСЛІДЖЕННЯ

1.1. Поняття системи безпеки розумного будинку

Забезпечення безпеки і створення максимального комфорту – це основні завдання, які повинна вирішувати автоматизація будинку. Система безпеки розумного будинку – це система, яка здійснює захист майна, забезпечує безперебійну роботу обладнання, попереджає аварійні ситуації. Залежно від побажання господарів і індивідуальних характеристик пристроїв, система може забезпечити різний рівень захищеності, який, спільно з високим ступенем комфорту, стане запорукою спокою і благополуччя.

Інтелектуальна система безпеки розумних будинків передбачає забезпечення захисту за кількома основними напрямками, які умовно можна розділити на наступні групи:

- захист від загорянь і автоматична система пожежогасіння;
- захист від протікання;
- захист від проникнення сторонніх осіб на територію, що охороняється.

Контроль загорянь здійснюється за допомогою системи датчиків, що реагують на найменшу зміну складу повітря при появі в ньому чадного газу. У разі фіксування такої події, буде включена протипожежна система, припинено доступ свіжого повітря, який сприяє розгоранню вогню, перекритий газ і відключено електрику. Сигналізація у вигляді сирени і світлового сигналу сповістить присутніх в приміщеннях про пожежу, а якщо він почнеться, коли господарів немає вдома, система відправить *SMS* на мобільний телефон. Паралельно запускається функція видалення диму, а в службу безпеки надходить повідомлення про пожежу. Ці заходи зменшують вірогідність збільшення

Кафедра КСУ				НАУ 21 05 27 000 ПЗ			
Виконав	Лузік М.Р.			Аналіз області дослідження	Літера	Аркуш	Аркушів
Керівник	Росінська Г.П.					9	54
Консулт.					СП 436 123		
Норм. контр.	Тупота Є.В.						
Зав. Каф.	Литвиненко О.Є.						

вогнища загорання і перешкоджають поширенню вогню по інших приміщеннях. А це значить, ризик нанесення житлу серйозного збитку значно знижується.

Для запобігання сумних наслідків виходу з ладу елементів водопостачання створені системи контролю протікань. Основою таких систем є датчики витoku води, які встановлюються там, де вірогідність протікання найбільша, тобто в місцях з'єднання побутової техніки чи сантехніки з водопровідними трубами, а між ручними вентилями і фільтрами водоочищення ставляться магнітні клапани. У разі пошкодження гнучкого підведення і попадання вологи на підлогу датчики фіксують її наявність і подають сигнал до клапанів. Ті моментально перекривають подачу води, запобігаючи затопленню приміщення. Системи «розумного будинку» здатні подати сигнал на мобільний телефон, попередивши господарів про подію. Таким чином, повертаючись додому, власники будинку або квартири вже будуть знати про необхідність зв'язатися з ремонтними службами.

Захист від проникнення на територію сторонніх осіб включає в себе цілий комплекс систем і пристроїв. В першу чергу, це контроль цілісності периметра приміщення. При спробі увійти через двері або пробратися через вікно, заданий контур переривається, і спрацьовують системи звукового та світлового оповіщення. Тривожний сигнал у вигляді голосового або *SMS*-повідомлення надходить на мобільний телефон власника житла, крім того, паралельно відбувається оповіщення служб безпеки.

1.2. Дротові та бездротові технології

За способом доставки сигналу технології, що застосовуються в системі «Розумний будинок», діляться на дротову та бездротову. Перша використовує для зв'язку між окремими пристроями дроти, друга – радіохвилі.

На сьогоднішній день дротова технологія є найбільш надійною і швидкісною. Вона має практично необмежені можливості по розширенню системи і хорошу захищеність мережі. До того ж провідна технологія не залежить від роботи автономних джерел живлення, тому що не використовує їх. Вона дає змогу відслідковувати статуси всіх пристроїв, адже всі пристрої в системі

взаємодіють один з одним в режимі онлайн, і цей зв'язок не залежить ні від джерела живлення, ні від відстані між ними. Сенсори і контролери для дротових систем коштують значно менше аналогів для бездротових систем.

Основним недоліком такої технології є труднощі монтажу. Для установки дротової системи необхідно провести дроти до кожного елемента. Зазвичай це прихована підводка, для чого в підлогах і стінах робляться штроби під кабель. Оптимально виконувати такий монтаж в процесі капітального ремонту або при будівництві будинку. Зате згодом дротова система не потребує будь-яких обслуговуванні і без проблем служить довгі роки.

Для забезпечення зв'язку між елементами за допомогою бездротової технології використовується радіосигнал. Існує кілька поширених протоколів, які можуть бути задіяні в таких системах: *Bluetooth, Wi-Fi, Z-Wave, ZigBee*.

Z-Wave – це протокол бездротового зв'язку з малою витратою енергії, розроблений для управління на відстані, контролю і автоматизації пристроїв в придатних для життя приміщеннях. Технологія використовує малопотужні і мініатюрні радіочастотні модулі, які вбудовуються в побутову електроніку і різні пристрої, такі як освітлювальні прилади, прилади опалення, пристрої контролю доступу, розважальні системи і побутову техніку. *Z-Wave* працює в діапазоні частот до 1 ГГц і оптимізована для передачі простих керуючих команд з малими затримками.

Zigbee – стандарт для набору високорівневих протоколів зв'язку, що використовують невеликі малопотужні цифрові трансивери. Заснований на стандарті *IEEE 802.15.4-2006* для бездротових персональних мереж. Передача даних здійснюється по радіоканалу. Частота залежить від регіону. Стандарт створювався з метою бути легше в експлуатації і нижче в ціні. Характерний низький період відповіді обладнання. Період активації займає не більше 15 мілісекунд. Завдяки переходу в сплячий режим значно падає потреба в електроенергії.

Головні переваги бездротових протоколів – мобільність, швидке розгортання мережі і простота налаштування. У цьому вони виграють у провідних аналогів, але недоліків набагато більше.

Бездротові технології можуть зазнавати впливу від інших хвильових мереж чи зловмисників. Радіосигнал не забезпечує зворотний зв'язок. Таким чином, користувач не має можливості контролювати стан обладнання. Ще один недолік – залежність від електроенергії. Прилади можуть одержувати живлення від мережі, але найчастіше це незручно, тому використовуються батарейки або акумулятори, які потрібно замінювати вчасно.

1.3. Огляд існуючих аналогів

Система безпеки розумного будинку *Mimi Smart* забезпечує наступні заходи безпеки: охоронні функції, відеоспостереження, контроль закриття вікон і дверей, а також захист від протікання води і побутового газу.

Якщо стороння людина влізе в приміщення – спрацюють датчики відкриття дверей і датчики руху всередині будинку. Сигнал про проникнення у приміщення або на територію об'єкту, що охороняється, буде миттєво переданий в службу охорони і на мобільні засоби зв'язку господарів будинку.

Камери, які встановлюються в будинку, можуть працювати весь день і фіксувати все, що відбувається в будинку. Сигнали, які передають датчики, а також записи з камер відеоспостереження, вносяться в пам'ять системи. Це робиться для того, щоб господар міг переглянути, коли і який датчик відправив сигнал, а також, що при цьому відбувалося на камерах відеоспостереження.

Датчики протікання встановлюються в місцях, де ймовірність поява протікання найбільш висока, наприклад під ванною або пральною машиною. У разі появи вологи на підлозі, датчики подадуть сигнал на клапани, які перекриють подачу води в систему, запобігши тим самим затоплення приміщення. Подібні дії будуть зроблені системою безпеки і в разі витоку газу. Одночасно з цим, система відправить на мобільний телефон власника повідомлення про те, що сталося.

Дана система використовує датчики та контролери компанії *Bron*:

– Центр управління (*Tablet*) – пристрій, який управляє усією системою, має sim модуль, за допомогою радіозв'язку підключається до інших пристроїв, коштує 295\$. Обмін даними між планшетом і компонентами системи здійснюється на частотах в діапазоні від 868,0 до 868,6 МГц.

– Інфрачервоний датчик руху (*MOTION S*) здатен виявити рух на відстані до 14 метрів, має налаштування чутливості, коштує 40\$. Один раз на хвилину датчик здійснює передачу сповіщення про поточний стан до центра управління та повинен прийняти від нього підтвердження. В проміжках часу між передаванням сповіщень датчик перебуває в енергозберігаючому режимі.

– Датчик відкриття (*DOOR S*) коштує 30\$, зручний у встановленні, є можливість прихованого монтажу. Якщо магнітокерований датчик знаходиться у зоні дії магніта то датчик передає на планшет сповіщення «закрито», у протилежному стані – сповіщення «відкрито».

– Датчик газу (*GAS S*) призначений для виявлення природного газу (метану), може бути інтегрований з клапаном перекриття газу, коштує 83\$.

– Датчик протікання (*LEAK S*) призначений для виявлення протікання води, має вологостійкий захист, коштує 35\$. У нормальному стані пристрій передає на планшет сповіщення «сухо», у разі попадання води на електроди невідкладно передає сповіщення «затоплено».

Система безпеки розумного будинку від компанії *Home Improvement Technologies* включає в себе роботу охоронної системи, яка захищає будинок від небажаного проникнення і допомагає уникнути аварійних ситуацій, вживаючи заходів для запобігання небажаних наслідків від прориву водопроводу, витоку газу.

Частиною системи безпеки "Розумний будинок" є різні датчики, які можуть визначити виникнення проблеми, передати повідомлення на пульт управління, який або відправить повідомлення власнику про виникнення неполадок, або прийме екстрених заходів – перекриє водяний або газовий вентиль, відключить електрику.

Контроль доступу запобіжить проникненню в будинок сторонніх, він може бути здійснений двома способами: кодовий доступ, при якому встановлюється спеціальний пристрій на двері або воротах; доступ за спеціальними картками або чіпам.

Коли система сигналізації зафіксує небажаний рух або відкриття вікон, включиться освітлення і звуковий сигнал сигналізації. Датчик руху, інтегрований з системою відеоспостереження, буде стежити за будинком і майном вдень і вночі. Якщо він зафіксує рух, власник отримає повідомлення із записом, на якому можна перевірити, чи слід турбуватися.

Крім замків на дверях, встановлюються також спеціальні ролети і механічні віконниці для захисту вікон. Крім цього, охоронний комплекс включає в себе установку камер відеоспостереження по периметру ділянки і в приміщеннях. Ви зможете спостерігати за тим, що відбувається всередині або зовні приміщення, перебуваючи в будь-якому місці, за допомогою смартфона або комп'ютера.

Стартовий комплект системи безпеки коштує 200\$ і складається з хаба, інфрачервоного датчика руху *MotionProtect*, датчика відчинення *DoorProtect* та брелка, який керує режимами охорони та надсилає сигнал тривоги на пульт після натиснення тривожної кнопки.

Хаб *Ajax* – головний пристрій сигналізації. Об'єднує всі датчики та контролює їхню роботу за допомогою дротового інтерфейсу. У разі небезпеки хаб миттєво відправляє *SMS* або телефонує власнику. Декілька каналів зв'язку для відправлення тривог – *Ethernet*, *GSM* (850/900/1800/1900 МГц).

Система безпеки розумного від компанії *Smart Home Company*. При несанкціонованому проникненні на об'єкт, що охороняється, спрацює система охоронної сигналізації, також забезпечується пожежна безпека, контроль витоку газу та запобігання затоплення.

Датчики руху зреагують на відкривання дверей. Відразу ж включиться сигналізація, яка оголосить тривогу. У цей момент система сигналізації відправить сигнал в охоронну службу. Одночасно з цим сигнал про проникнення в будинок отримує власник у вигляді *SMS* на телефон або на електронну пошту.

Як тільки завіє сирена, буде вестися відеоспостереження якщо встановлена система повного відеоспостереження за будинком. Запис допоможе розшукати грабіжників, якщо їм вдасться втекти.

При виникненні загрози пожежної безпеки (спрацьовування датчика диму), система Розумний будинок оповіщає власника приміщення на мобільний телефон, а також, в разі необхідності, пожежну службу.

У разі виявлення витoku газу відбувається автоматичне відключення його подачі, повідомлення на мобільний телефон власника за допомогою SMS, а також повідомлення відповідним органам.

Повний контроль споживаних енергоресурсів приміщення дозволяє своєчасно виявити витoku води. Щоб запобігти затопленню, система Розумний будинок відключить подачу води, повідомить у відповідні органи і надішле власнику SMS.

Система безпеки використовує контролери та датчики *Fibaro*. До складу стартового комплекту входить центральний контролер *Fibaro Home Center Lite*, датчик затоплення *Fibaro Flood Sensor FGFS-101*, датчик руху *Fibaro FGMS-001*, датчик диму *Fibaro Smoke Sensor FGSS-101* і датчик відкриття *Fibaro Door / Window Sensor FGK-101*. Вартість комплекту складає 420\$. Контролер *Fibaro Home Center Lite* є центром управління розумним будинком і взаємодіє з усіма датчиками за допомогою технології *Z-Wave*.

1.4. Висновки до розділу

У першому розділі було розглянуто поняття системи безпеки розумного будинку, технології які використовуються, а також розглянуті існуючі системи безпеки, їх функціонал і вартість.

Система безпеки розумного будинку покликана забезпечити цілісність інженерних комунікацій і захистити від проникнення зловмисників. У таких системах використовується дротове або бездротове з'єднання компонентів. Системи, які використовують дротову технологію, є набагато дешевшими і

надійнішими, але мають труднощі встановлення. Системи з бездротовими технологіями, навпаки, дуже легко встановлюються, але мають велику ціну.

Розглянувши існуючі аналоги систем безпеки розумного будинку, можна зробити висновок, що більш вигідним буде реалізація системи з використанням бездротової технології зв'язку між компонентами. Всі системи мають схожий функціонал, тому можна виділити основні компоненти та функції. Стандартом для всіх систем є реалізація охоронної сигналізації, у яку входить датчик руху, датчик відчинення і керуючий пристрій. Керуючий пристрій оброблює сигнали датчиків і при виявленні вторгнення до житла надсилає повідомлення на мобільний пристрій власника та вмикає звуковий сигнал тривоги.

РОЗДІЛ 2

ОГЛЯД АПАРАТНОЇ ТА ПРОГРАМНОЇ ЧАСТИНА ПРОЄКТУ

2.1. *Arduino* – як платформа проєктування

Arduino – це інструмент для проєктування електронних пристроїв (електронний конструктор), який більш щільно взаємодіє з навколишнім фізичним середовищем, ніж стандартні персональні комп'ютери, які фактично не виходять за рамки віртуальності. Пристрої на базі цієї платформи мають можливість приймати сигнали від різних цифрових та аналогових датчиків, а також можуть управляти різними виконавчими пристроями. Ардуіно являє собою готову апаратно-програмну платформу, головними компонентами якої є невелика плата-контролер введення / виведення і середовище розробки на основі *Processing* / *Wiring*. Проєкти Ардуіно можуть працювати самостійно, а також взаємодіяти з програмним забезпеченням, що працює на персональному комп'ютері (наприклад, додатками *Adobe Flash*, *Processing*, *MaxMSP*).

Випускаються різні моделі *Arduino*. Кожна з них призначена для різних завдань. Деякі плати принципово відрізняються від наведеної на малюнку нижче. Але більшість з них мають такі однакові вузли (рис 2.1):

Роз'єм живлення (*USB* / роз'єм для адаптера). Кожна плата *Arduino* повинна приєднуватися до джерела живлення. *Arduino Uno* може житися від *USB* кабелю від вашого персонального комп'ютера Або від окремого адаптера, який підключається до передбаченого на платі роз'єму. Рекомендована напруга живлення для *Arduino* – від 6 до 12 вольт. *USB* також використовується для завантаження вашої програми (скетчу) на плату. На малюнку з'єднання через *USB* відзначено (1), а роз'єм для зовнішнього джерела живлення – (2).

Кафедра КСУ				НАУ 21 05 27 000 ПЗ			
Виконав	Лузік М.Р.			Огляд апаратної та програмної частини проєкту	Літера	Аркуш	Аркушів
Керівник	Росінська Г.П.					17	54
Консульт.					СП 436 123		
Норм. контр.	Тупота Є.В.						
Зав. Каф.	Литвиненко О.Є.						

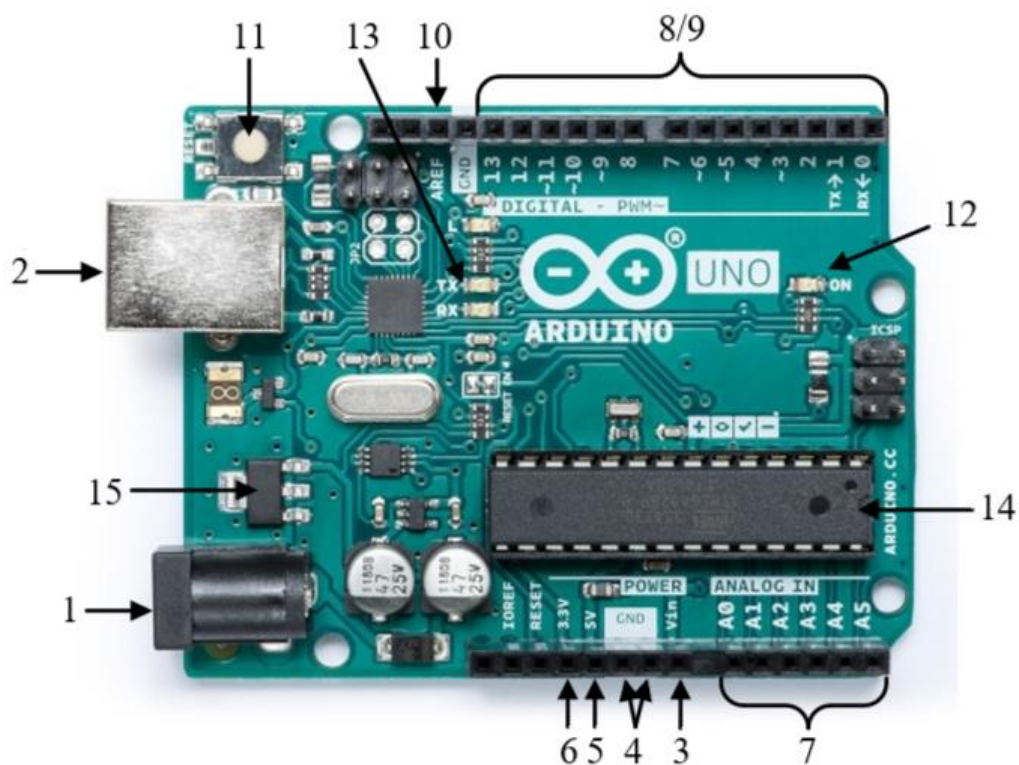


Рис. 2.1. Плата Arduino Uno

Піни (*Vin*, *5V*, *3.3V*, *GND*, *Analog*, *Digital*, *PWM*, *AREF*). Піни на платі Arduino – це роз'єми, до яких підключаються дроти від периферійних пристроїв (дуже часто для прототипів використовують монтажні плати (макетні плати) і дроти з конекторами на кінцях). На Arduino кілька типів пінів, кожен з яких підписаний відповідно до виконуваної функцією:

- *Vin* (3) – вхід живлення, використовується для отримання живлення від зовнішнього джерела. На вхід *Vin* рекомендується подавати напругу в діапазоні від 7В до 20В, в уникненні перегріву і згоряння вбудованого стабілізатора.

- *GND* (4) – контакт для підключення землі. Необхідний для створення замкненого кола при підключенні до контактів *Vin*, *5V* або *3.3V*. На платах кілька пінів *GND*, кожен з яких може використовуватися для заземлення електричного кола.

- *5V* (5) – пін, який на виході забезпечує живлення 5 вольт для зовнішніх пристроїв. При отриманні живлення платою з будь-яких інших джерел (*USB*, роз'єм живлення або *Vin*) на цьому контакті завжди можна отримати стабільну

напругу 5 вольт. Його можна вивести на макетну плату або подати безпосередньо на пристрій.

– *3.3V* (6) – джерело 3.3 вольта напруги для живлення зовнішніх пристроїв.

Працює за таким-же принципом, що і контакт *5V*.

– *Analog* (7) – на ділянці, яка підписана «*Analog In*» розташовані аналогові входи. Ці піни дозволяють зчитувати сигнали від аналогових датчиків (наприклад, датчик температури) і перетворювати їх в цифрові значення, з якими далі можна працювати.

– *Digital* (8) – цифрові піни, які використовуються для цифрових вхідних сигналів (наприклад, натискання кнопки) і для генерації цифрових вихідних сигналів (наприклад, живлення світлодіода).

– *PWM* (9) – піни, які працюю як в звичайному цифровому режимі, так і в режимі Широтно-імпульсної модуляції (ШИМ). Вони можуть імітувати аналоговий вихідний сигнал (наприклад, для поступового згасання світлодіода). Якщо *PWM* піни не мають окремої ділянки на платі, то позначаються символом «~» поруч із номером піна.

– *AREF* (10) – даний контакт відповідає за визначення опорної напруги аналогових входів платформи.

Кнопка скидання (11). При натисканні на неї контакт скидання замикається з землею і код, завантажений на *Arduino* починає відпрацьовувати заново. Корисна опція, якщо ваш код відпрацьовує без повторів, але ви хочете протестувати його роботу.

Індикатор живлення (12). Цей світлодіод загорається при підключенні *Arduino* до джерела живлення. Якщо цей індикатор не вмикається слід перевірити схему.

Світлодіоди *TX* і *RX* (13). *TX* – скорочення від *transmit* (передача), *RX* – від *receive* (прийом). Ці світлодіоди дозволяють візуально відслідковувати, передає або приймає дані *Arduino* (наприклад, при завантаженні програми на плату).

Головна інтегральна мікросхема (14). Чорна деталь з металевими конекторами з двох сторін це інтегральна мікросхема, мікропроцесор (*Integrated*

Circuit). Цей чіп різний в різних моделях *Arduino*, але зазвичай він відноситься до лінійки мікропроцесорів *ATmega* від компанії *ATMEL*.

Регулятор напруги (15) – контролює напругу, яка надходить на плату *Arduino*. Він не пропускає занадто велику напругу на плату, щоб уникнути її пошкодження.

2.2. Середовище розробки *Arduino IDE*

Arduino IDE – інтегроване середовище розробки для *Windows*, *MacOS* і *Linux*, яке розроблена на *Java* і призначене для створення і завантаження програм на *Arduino*-сумісні плати, а також на плати інших виробників.

Інтерфейс середовища розробки Ардуіно містить наступні основні елементи: текстовий редактор для написання коду, область для виведення повідомлень, панель інструментів і головне меню. Дане програмне забезпечення дозволяє комп'ютеру взаємодіяти з Ардуіно як для передачі даних, так і для прошивки коду в контролер.

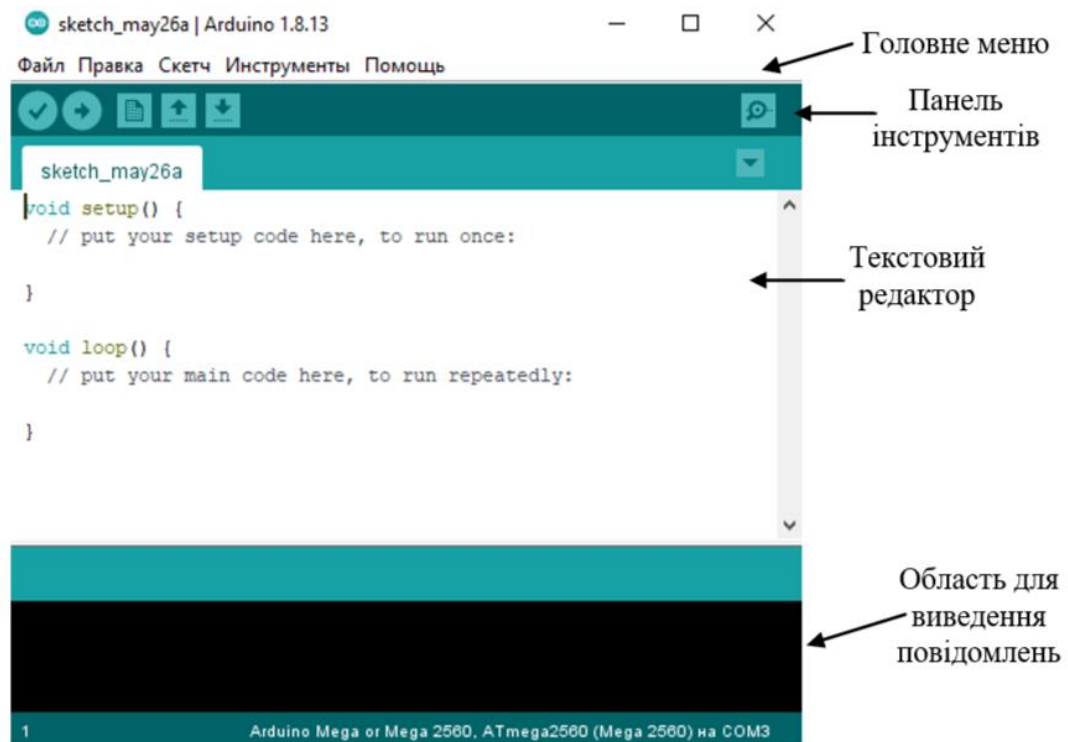


Рис. 2.2. Інтерфейс *Arduino IDE*

Програми, що створюються в середовищі розробки Ардуіно, ще називають скетчами. Скетчі пишуться в текстовому редакторі і зберігаються в файлах з розширенням .ino. Вбудований текстовий редактор має стандартні інструменти копіювання, вставки, пошуку і заміни тексту. Область повідомлень у вікні програми є, свого роду, зворотним зв'язком для користувача і інформує його про події (в тому числі і про помилки), що виникають в процесі запису або експорту написаного коду. У нижньому правому куті вікна програми вказується модель поточної плати і послідовний порт, до якого вона підключена.

На панелі інструментів знаходиться 6 кнопок (рис. 2.3):



Рис. 2.3. Панель інструментів

- Перевірити – використовується для перевірки синтаксису написаного коду. У разі знаходження помилок, вони будуть виведені в області повідомлень.
- Завантажити – використовується для компіляції програми і завантаження її в мікроконтролер Ардуіно.
- Створити – використовується для створення нового файлу.
- Відкрити – використовується для відкриття файлу зі скетчем.
- Зберегти – використовується для збереження поточного коду.
- Монітор послідовного порту – використовується для відкриття програми «Serial Monitor». Ця програма відображає дані, що надходять від Ардуіно на комп'ютер по послідовному інтерфейсу (підтримується робота як з *USB-*, так і зі звичайними версіями Ардуіно). Для надсилання даних зовнішнього пристрою, необхідно ввести текст у вікні програми і клацнути по кнопці "Відправити" (або натиснути *Enter*). У списку необхідно вибрати тільки швидкість передачі даних,

що відповідає тій швидкості, яку ви вказали в функції *Serial.begin()* в вашому скетчі. За допомогою функції *Serial.print()* можна виводити текст у вікно програми.

Бібліотеки розширюють функціональність програм і несуть в собі додаткові функції, наприклад, для роботи з апаратними засобами, функції по обробці даних і т.д. Для підключення бібліотеки необхідно вибрати її з меню Скетч → Підключити бібліотеку. Після цього бібліотека буде скомпільована, а в початок програми буде додано один або кілька операторів `#include`. Бібліотеки завантажуються в контролер разом зі скетчем, тому кожна підключена бібліотека потребує додаткового місця в пам'яті мікроконтролера.

Для того, щоб прошити програму в контролер, необхідно правильно вибрати плату в меню *Tools* → *Board* і послідовний порт в меню *Tools* → *Serial Port*. Після вибору використовуваного порту і плати, необхідно натиснути кнопку Завантажити на панелі інструментів або вибрати пункт Завантажити з меню Файл. Після цього відбудеться скидання Ардуіно і почнеться процес завантаження програми в пам'ять контролера. У процесі завантаження на більшості моделей Ардуіно будуть блимати світлодіоди *RX* і *TX*. По завершенню процесу прошивки, програма виведе відповідне повідомлення або помилку.

Завантаження програми в Ардуіно здійснюється за допомогою завантажувача – невеликої програми, що прошита в пам'яті мікроконтролера, яка дозволяє завантажувати в нього код без зовнішніх апаратних засобів. Завантажувач активізується на кілька секунд після перезавантаження пристрою, після чого він запускає на виконання останній завантажений в контролер скетч. При запуску завантажувача буде блимати вбудований світлодіод, підключений до 13 ніжки контролера.

2.3. Програмування Arduino

Для програмування *Arduino* використовується спрощена версія мови C++ з обумовленими функціями. Всі прошивки для *Arduino* повинні містити мінімум 2 функції: *setup ()* і *loop ()*.

Функція *setup* () викликається, коли стартує скетч. Використовується для ініціалізації змінних, визначення режимів роботи виходів, встановлення з'єднання з додатковими модулями і налаштування підключених бібліотек. Функція *setup* запускає тільки один раз, після кожної подачі живлення або скидання плати *Arduino*.

Функція *loop*() виконується після функції *setup*(). Як випливає з назви, весь код між фігурними дужками в *loop*() крутиться в циклі, дозволяючи вашій програмі здійснювати обчислення і реагувати на них. Наприклад мікроконтролер *ATmega328*, який встановлений в більшості плат *Arduino*, буде виконувати функцію *loop* близько 10000 раз в секунду (якщо не використовуються затримки і складні обчислення).

Основними типами функцій є функції цифрового, аналогового та розширеного введення / виведення, а також функції часу.

До цифрових належать функції:

– *pinMode*() – налаштовує режим роботи вказаного виводу: як вхід або як вихід;

– *digitalWrite*() – відправляє на цифровий вивід значення *HIGH* або *LOW*. Якщо функцією *pinMode*() вивід налаштований як вихід (*OUTPUT*), то при виконанні функції *digitalWrite*() його напругу буде змінено на 5 В (або 3.3 В для плат, що працюють від 3.3В) при відправці *HIGH*, або на 0 В – при відправці *LOW*. Якщо вивід налаштований як вхід *INPUT*, то відправка функцією *digitalWrite*() значення *HIGH* призведе до підключення внутрішнього підтягуючого резистора номіналом 20 КОм. Запис значення *LOW* призведе до відключення підтяжки.

– *digitalRead*() – зчитує рівень сигналу *HIGH* або *LOW* з зазначеного цифрового виводу.

До аналогових належать функції:

– *analogWrite*() – Формує задану напруга на виводі у вигляді ШІМ-сигналу. Після виклику *analogWrite*, на виводі буде безперервно генеруватися ШІМ-сигнал з заданим коефіцієнтом заповнення до наступного виклику функції *analogWrite*

(або до моменту виклику *digitalRead* або *digitalWrite*, взаємодіючих з цим же виводом). Частота ШІМ становить приблизно 490 Гц.

– *analogRead()* – зчитує величину напруги з зазначеного аналогового виводу. За допомогою функції *analogReference* можна змінювати вхідний діапазон і роздільну здатність.

– *analogReference()* – встановлює джерело опорної напруги, що використовується при зчитуванні аналогового сигналу, тобто задає максимальне значення вхідного діапазону.

До розширеного введення / виведення належать функції:

– *tone()* – генерує на виводі прямокутний сигнал заданої частоти. Функція також дозволяє задавати тривалість сигналу. Якщо тривалість сигналу не вказана, він буде генеруватися доти, поки не буде викликана функція *noTone*. Одночасно може генеруватися тільки один сигнал заданої частоти. Якщо сигнал вже генерується на будь-якому виводі, то використання функції *tone* для цього виводу просто змінить частоту сигналу, а виклик функції для іншого виводу не матиме ніякого ефекту.

– *noTone()* – припиняє генерування прямокутного сигналу після використання функції *tone ()*. Якщо сигнал не генерується, функція ні до чого не призводить.

– До часових належать наступні функції:

– *millis()* – повертає кількість мілісекунд, що пройшли з моменту старту програми Ардуіно. Число, що повертається, переповниться (скинеться в 0) приблизно через 50 днів.

– *delay()* – призупиняє виконання програми на вказаний проміжок часу (в мілісекундах).

2.4. Плати *Arduino*

Існує велика кількість різних оригінальних плат *Arduino*, і ще більше *Arduino*-сумісних, створених великою кількістю сторонніх компаній. Вони відрізняються характеристиками встановлених мікроконтролерів, розмірами, кількістю портів введення-виведення і іншими параметрами, завдяки чому є

можливість вибрати плату, максимально відповідну поставленим вимогам. Нижче представлені найбільш розповсюджені плати *Arduino*.

Arduino Leonardo – це плата, яка побудована на базі мікроконтролера *ATmega32U4*. Дана плата має роз'єм мікро-*USB*, роз'єм живлення, роз'єм для внутрішньосхемного програмування *ICSP (In-Circuit Serial Programming)* і кнопка скидання. Кількість цифрових входів і виходів 20, а аналогових - 12. Вбудована flash-пам'ять має об'єм 32 КБ, з яких чотири виділяються для завантажувача. ОЗУ пристрою 2,5 КБ, а тактова частота дорівнює 16 МГц.

Відмінність *Leonardo* від всіх попередніх плат полягає в тому, що його *USB*-контролер вбудований безпосередньо в мікроконтролер *ATmega32U4*, що виключає необхідність в додатковому процесорі. Завдяки цьому при приєднанні до комп'ютера *Leonardo* може визначатися не тільки як віртуальний *COM*-порт, але і як звичайна миша або клавіатура.



Рис. 2.4. *Arduino Leonardo*

Arduino Uno – це пристрій на основі мікроконтролера *ATmega328*. У його склад входить все необхідне для зручної роботи з мікро контролером: 14 цифрових входів / виходів (з них 6 можуть використовуватися в якості ШІМ-виходів), 6 аналогових входів, роз'єм *USB*, роз'єм живлення, роз'єм для внутрішньосхемного програмування і кнопка скидання. Характеристики плати: тактова частота – 16 МГц; робоча напруга – 5 В; флеш-пам'ять – 32 Кб; оперативна пам'ять – 2 Кб.

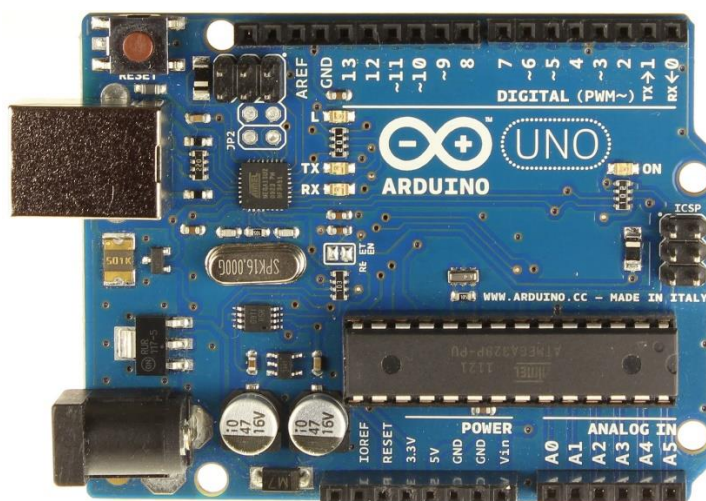


Рис. 2.5. *Arduino Uno*

Arduino Due – це потужна плата, заснована на 32-бітному ARM-процесорі AT91SAM3X8E від Atmel. Він володіє тактовою частотою 84 МГц, а його 32-бітна архітектура дозволяє виконувати більшість операцій над цілими числами в 4 байта за один такт. Плата має 54 цифрових виводи (з яких 12 можуть працювати в якості ШІМ-виходів), 12 аналогових входів, 4 *UART* (апаратних приймача, що здійснюють послідовну передачу даних), *USB* з підтримкою технології *OTG*, 2 ЦАП (цифро-аналогових перетворювача), 2 *TWI*, роз'єм живлення, роз'єм *SPI*, роз'єм *JTAG*, кнопка скидання і кнопка очищення пам'яті. Особливість інтерфейсу *SPI* в тому, що його не можна використовувати для внутрисхемного програмування, а виключно для зв'язку з іншими пристроями. На відміну від інших плат, робоча напруга *Arduino Due* становить 3.3 В. Дана плата має на борту 512 КБ флеш-пам'яті і 96 КБ оперативної пам'яті.

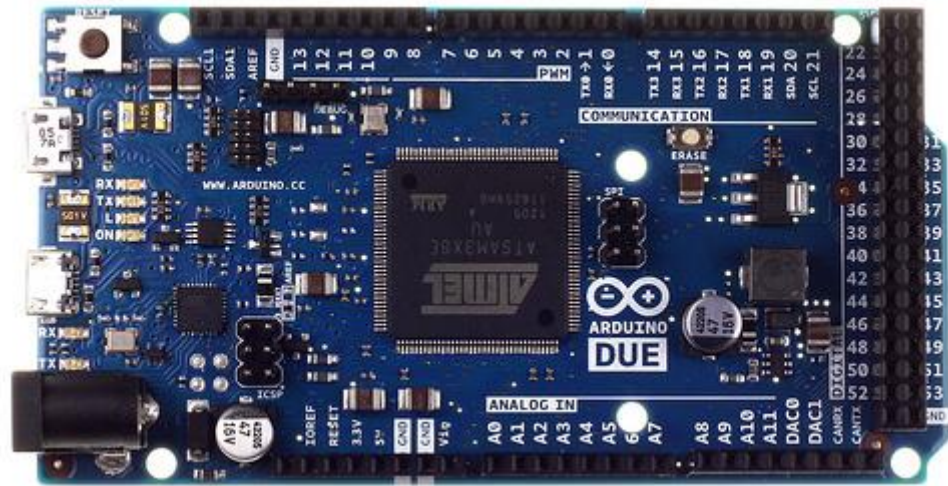


Рис. 2.6. *Arduino Due*

Arduino Mega 2560 – це пристрій на основі мікроконтролера *ATmega2560*. У його склад входить все необхідне для зручної роботи з мікроконтролером: 54 цифрових входу / виходу (з яких 15 можуть використовуватися в якості ШІМ-виходів), 16 аналогових входів, 4 *UART* (апаратних приймача для реалізації послідовних інтерфейсів), кварцовий резонатор на 16 МГц, роз'єм *USB*, роз'єм живлення, роз'єм *ICSP* для внутрішньосхемного програмування і кнопка скидання. Робоча напруга даної плати складає 5 В, об'єм флеш-пам'яті – 256 Кб, об'єм оперативної пам'яті – 8 Кб. Плати *Arduino Mega* використовуються там, де необхідна велика кількість входів і виходів. Мною була обрана саме ця плата для проектування системи безпеки.

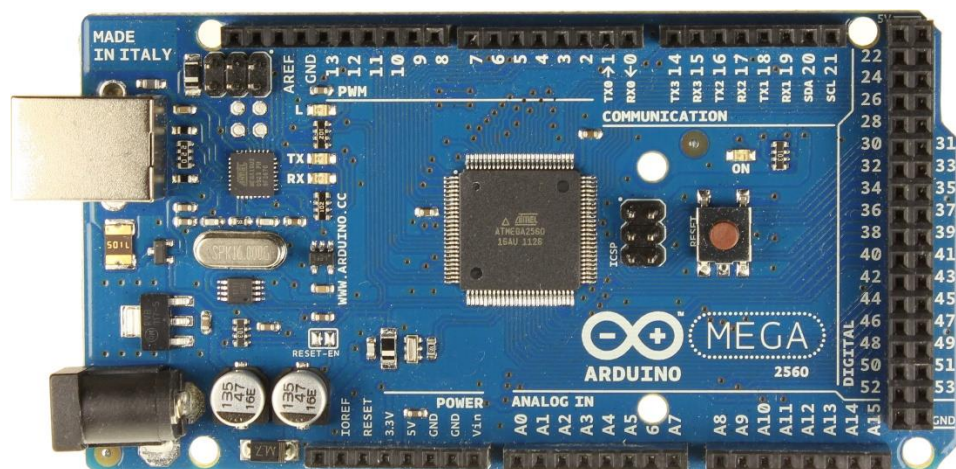


Рис. 2.7. *Arduino Mega 2560*

Arduino Micro – контролер, який повністю відповідає своїй назві; це одна з найменших плат з лінійки *Arduino*. Незважаючи на свій невеликий розмір, вона все ж має велику кількість вхідних і вихідних виводів: 20 цифрових каналів вводу / виводу, з яких 7 можуть бути використані як виходи ШІМ, а також має 12 аналогових входів. Характеристики даної плати: робоча напруга – 5 В, тактова частота – 16 МГц, флеш-пам'ять – 32 Кб; оперативна пам'ять – 2.5 Кб.

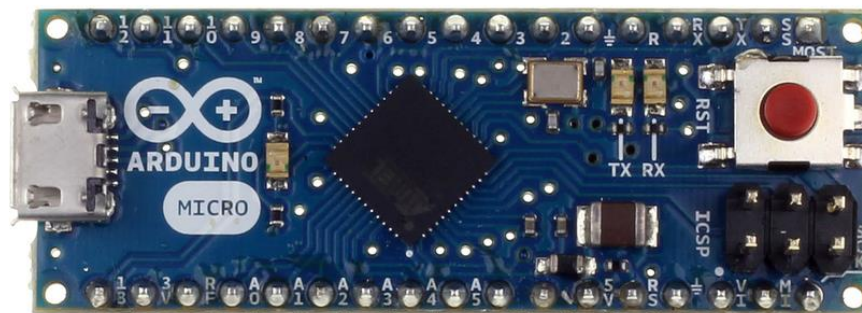


Рис. 2.8. *Arduino Micro*

Arduino Ethernet – це пристрій на основі мікроконтролера *ATmega328*. У його склад входить 14 цифрових виходів, 6 аналогових входів, кварцовий резонатор на 16 МГц, роз'єм *RJ45*, роз'єм живлення, роз'єм *ICSP*, а також кнопка скидання. Характеристики даної плати: тактова частота – 16 МГц; флеш-пам'ять – 32 Кб; оперативна пам'ять – 2 Кб; робоча напруга – 5 В. *Arduino Ethernet* має додатковий модуль *POE (Power Over Ethernet)*. Ця опція дозволяє даній платі житися безпосередньо від мережі *Ethernet*, без необхідності використання зовнішнього джерела живлення за умови, що харчування *POE* підключено на іншому кінці кабелю *Ethernet*.

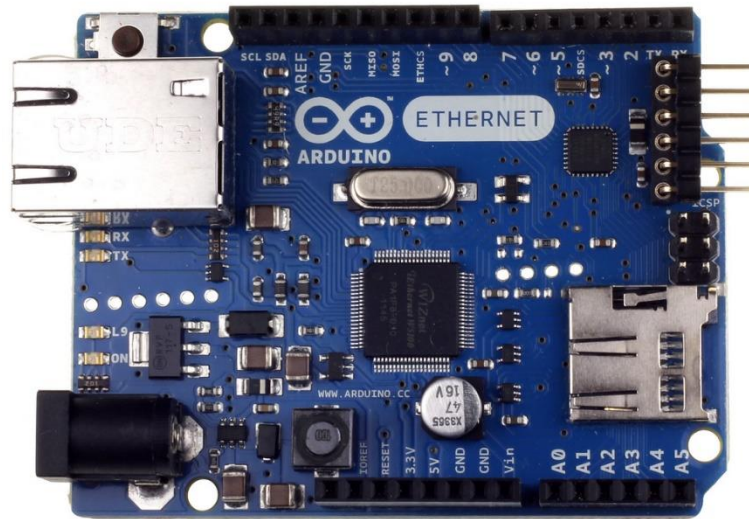


Рис. 2.9. *Arduino Ethernet*

2.5. Датчики та пристрої

Основним датчиком в системі безпеки є інфрачервоний датчик руху *HC-SR501*. Інфрачервоні датчики дозволяють виявляти рух людей в контрольованій зоні, так як тіло людини випромінює тепло. Конструкція ППР датчика руху не дуже складна – він складається з піроелектричного елемента, що відрізняється високою чутливістю (деталь циліндричної форми, в центрі якої розташований кристал) до наявності в зоні дії певного рівня інфрачервоного випромінювання. Чим вище температура об'єкта, тим більше випромінювання. Зверху *PIR*-датчика встановлюється півсфера, розділена на кілька ділянок (лінз), кожна з яких фокусує випромінювання теплової енергії на різні сегменти датчика руху. Найчастіше в якості лінзи застосовують лінзу Френеля, яка за рахунок концентрації теплового випромінювання дозволяє розширити діапазон чутливості інфрачервоного датчика руху Ардуіно.

Основними технічними характеристиками датчика руху Ардуіно: зона виявлення рухомих об'єктів становить від 0 до 7 метрів; діапазон кута спостереження – 110° ; напруга живлення – 4.5-6 В; робочий струм – до 0.05 мА; температурний режим – від -20° до $+50^\circ$ С; регульований час затримки від 0.3 до 300 с.



Рис. 2.10. Зовнішній вигляд датчик руху *HC-SR501*

Для взаємодії користувача з системою необхідні пристрої вводу та виводу. У якості пристрою вводу була обрана 12-кнопочка мембранна клавіатура. Для використання мембранної клавіатури потрібно підключити її до пристрою, яким потрібно керувати, за потребою створити програму управління і підключити живлення. Принцип роботи – при натисканні будь-якої клавіші замикається між собою певна пара проводів. Характеристики даної клавіатури: час спрацювання 1 мс; надійність не менше 1 млн натискань; максимальна постійна робоча напруга 35В.



Рис. 2.11. 12-кнопочка мембранна клавіатура

У якості пристрою виводу інформації був обраний символний дисплей *LCD 1602*. Він має на борту перехідник інтерфейсної шини *IIC / I2C*. Характеристики

дисплею: колір фону – зелений, колір символів – чорний, має два рядки по 16 символів в кожній, напруга живлення 5 В.

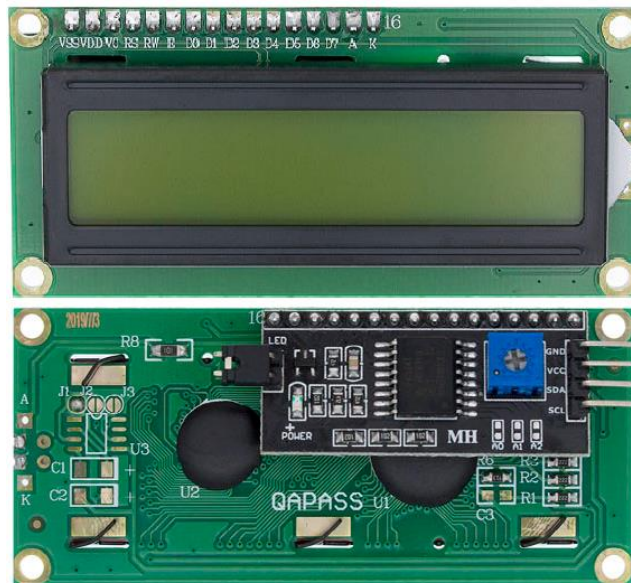


Рис. 2.12. Вигляд дисплею спереду та ззаду

Для віддаленого керування а також сповіщення користувача був обраний модуль *SIM800L*. *GSM*-модуль (*Global System for Mobile Communications*) використовує мережу телефонного зв'язку оператора, для отримання і передачі сигналу на віддалений об'єкт управління. Основою модуля є чіп *SIM800L* від *SimCom*. Робоча напруга чіпа складає від 3.4 В до 4.4 В. Основні технічні параметри модуля:

- формат *SIM* карти: *microSIM*;
- чотири діапазони частот 850/900/1800/1900 МГц;
- підтримка *2G* мережі;
- струм в режимі очікування – 0,7 мА, граничний струм – 2 А.



Рис. 2.13. Модуль *GSM SIM800L*

Модуль *SIM800L* має нестандартну напругу живлення 3.4 – 4.4 В, а плата ардуіно забезпечує живлення 3.3 В і 5 В. Для забезпечення необхідної напруги використовується окреме спільне джерело живлення на 7 – 12 В і *DC-DC* понижуючий перетворювач *MP1584*, за допомогою якого встановлюється напруга для *GSM*-модуля. Основні характеристики: вхідна напруга 4.5 – 28 В, вихідна 0.8 – 20 В (налаштовується спеціальним резистором на платі), максимальний вихідний струм – 3 А (залежить від різниці між вхідною і вихідною напругами).



Рис. 2.14. *DC-DC* понижуючий перетворювач *MP1584*

П'єзоелектричний випромінювач звуку – п'ятивольтовий активний динамік безперервного сигналу з вихідною частотою 2.3 КГц. Даний модуль використовується для звукового сповіщення системи.



Рис. 2.15. Зовнішній вигляд п'єзоелектричного випромінювача звуку

2.6. Висновки до розділу

В даному розділі були розглянуті основні пристрої, які необхідні для розробки апаратно-програмної системи безпеки в *Smart House*.

Плату *Arduino* необхідно вибирати в залежності від функцій майбутньої системи і модулів, які будуть підключатися до неї. На даний час доступна велика кількість, які відрізняються розміром.

Інфрачервоними датчики дозволяють вловлювати рух. Дуже часто використовуються в системах сигналізації. Ці датчики малі за габаритами, недорогі, споживають мало енергії, легкі в експлуатації, практично не схильні до зносу.

GSM модуль в проєктах Ардуіно дозволяє підключатися до віддалених автономним пристроїв через стільниковий зв'язок. Використовуючи такий модуль можна відправляти команди на пристрої і приймати інформацію від нього за допомогою *SMS*-команд.

Для проєктування та розробки апаратної частини системи було обрані такі компоненти: за основу пристрою було обрано плату *Arduino Mega*, для виявлення руху обрано інфрачервоний датчик руху *HC-SR501*, для передачі повідомлень і віддаленого управління використано *GSM*-модуль *SIM800L*, у якості інтерфейсу керування системою обрано 12-кнопову клавіатуру і дисплей з *lcd* екраном.

РОЗДІЛ 3

ОПИС ПРОЦЕСУ РОЗРОБКИ СИСТЕМИ

3.1. Проєктування та розробка апаратної частини системи

Для проєктування та розробки системи, необхідно спочатку обрати елементи, що будуть використовуватися, та створити схему роботи пристрою. За основу пристрою було обрано платформу *Arduino Mega*. У якості датчику виявлення руху був обраний інфрачервоний датчик руху *HC-SR501*. Для керування системою використано *lcd* дисплей з двома рядами по 16 символів і 12-кнопкова мембранна клавіатура. Для надсилання повідомлень власнику обраний *GSM*-модуль *SIM800L*. Для реалізації звукової сирени використано п'єзоелектричний випромінювач звуку.

Першим етапом проєктування системи є побудова схеми з'єднань усіх елементів системи та самого мікроконтролера. Схема з'єднань пристроїв наведена на рис. 3.1.

Червоними лініями на цій схемі позначено з'єднання, по яким йде живлення, чорними лініями позначено заземлення. Зеленою та синьою лініями позначене з'єднання *GSM*-модуля із платою *Arduino*. Жовтими лініями *Arduino* з'єднуються з світлодіодами. Помаранчевими лініями поєднані клавіатура та плата. Білою і сірою лініями зображено з'єднання *Arduino* із інтерфейсною шиною *IIC / I2C*, яка підключена до дисплея. Коричневою лінією з платою з'єднується цифровий вихід датчика руху. Після того, як було побудовано схему з'єднань, слід зібрати реальний пристрій з'єднуючи усе як зображено на цій схемі.

Кафедра КСУ				НАУ 21 05 27 000 ПЗ			
Виконав	Лузік М.Р.			Опис процесу розробки системи	Літера	Аркуш	Аркушів
Керівник	Росінська Г.П.					34	54
Консульт.					СП 436 123		
Норм. контр.	Тупота Є.В.						
Зав. Каф.	Литвиненко О.Є.						

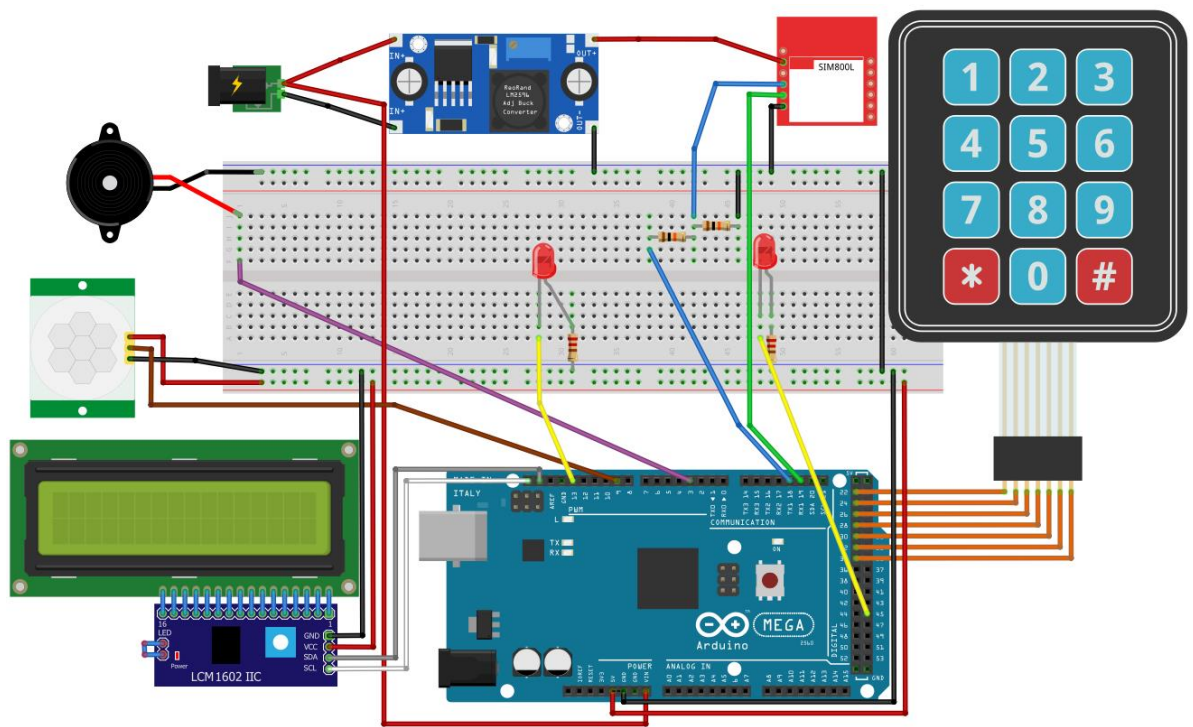


Рис. 3.1. Модель підключення пристроїв

Програмування функціоналу системи відбувається шляхом підключення мікроконтролера до комп'ютера та завантаження на нього скетчу з виконуваним кодом.

На початку скетчу оголошуються підключені бібліотеки, макроси, константи та глобальні змінні:

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <Keypad.h>
#include <TinyGsmClient.h>
#include <EEPROM.h>
#define LedPin 45
#define SoundPin 3
#define SerialMon Serial
#define SerialAT Serial1
TinyGsm modem(SerialAT);
String _response;
String phone = "+38", _phone = "";
```

```

String retrievedString;
String pin = "", _pin = "";
String buttons;
const byte ROWS = 4; // число строк клавиатуры
const byte COLS = 3; // число столбцов клавиатуры
char hexaKeys[ROWS][COLS] = {
    {'1', '2', '3'},
    {'4', '5', '6'},
    {'7', '8', '9'},
    {'*', '0', '#'}
};
byte rowPins[ROWS] = {22, 24, 26, 28};
byte colPins[COLS] = {30, 32, 34};
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins,
ROWS, COLS);
LiquidCrystal_I2C lcd(0x27, 16, 2);
bool access = false, alarm = false, waiting = false, smsSend = false, signalization
= false;
int count= 0;

```

У перших чотирьох рядках цього фрагмента коду підключаються бібліотеки, які забезпечують доступ до необхідних нам методів, що прописані в них.

Бібліотека *LiquidCrystal_I2C* – це бібліотека для роботи з символьними графічними дисплеями, які підключаються до *Arduino* за допомогою апаратної шини *I2C*. *I2C* – послідовна асиметрична шина для зв'язку між інтегральними схемами всередині електронних приладів. Використовує дві двонаправлені лінії зв'язку (*SDA* і *SCL*), які застосовується для з'єднання низькошвидкісних периферійних компонентів з процесорами і мікроконтролерами.

Бібліотека *Wire* – це бібліотека, яка використовується для зв'язку мікроконтролера з пристроями і модулями через інтерфейс *I2C*. Для зв'язку по *I2C*

використовується всього два контакти: лінія даних (*SDA*) і лінія тактового сигналу (*SCL*). До відповідних роз'ємів *Arduino* можна підключити до 120 пристроїв, що підтримують інтерфейс *I2C*. Для обміну даними з такими пристроями і потрібна *Arduino* бібліотека *Wire*.

Бібліотека *Keypad* – це бібліотека, яка необхідна для використання спільно з *Arduino* клавіатур матричного типу.

Бібліотека *TinyGsmClient* – це бібліотека *Arduino* для модулів *GSM/GPRS*.

Бібліотека *EEPROM* – бібліотека для роботи з енергонезалежною пам'яттю *EEPROM*. *EEPROM* (*Electrically Erasable Programmable Read-Only Memory*) – пам'ять, в якій інформація зберігається навіть після вимкнення пристрою (подібно маленькому жорсткому диску). Дана бібліотека дозволяє записувати і зчитувати інформацію з цієї пам'яті.

Далі за допомогою директиви препроцесора *#define* створюємо макроси заміни коду, для того щоб було зручно звертатися до виходів наших датчиків, а також для відправлення *AT* команд *GSM*-модулю. Далі йдуть глобальні змінні типу *String* у які записується інформація, яка надходить від *GSM*-модуля, змінні для збереження номеру телефону, пін-коду і нажатих кнопок клавіатури. Константні змінні визначають кількість стовпчиків та рядків клавіатури, масив типу *char* містить значення кнопок клавіатури. Масиви типу *byte* містять номери входів плати мікроконтролера, до яких підключені дроти, які відповідають рядкам та стовпчикам клавіатури. Далі створюється змінна типу *Keypad*, за допомогою якої буде визначатися, яка кнопка була натиснена. Після чого визначається об'єкт дисплея. Вкінці створено змінні, що знадобляться нам під час роботи програми.

Далі йде блок коду, що задає основні налаштування:

```
pinMode(SoundPin, OUTPUT);
```

```
pinMode(LedPin, OUTPUT);
```

```
lcd.begin();
```

```
lcd.backlight();
```

У двох перших рядках коду функція *pinMode* задає режими роботи цифрових виходів на *Arduino*. Останні два рядки ініціалізують *lcd* екран і вмикають підсвітку фону.

Збереження даних в *EEPROM*. Енергонезалежна пам'ять використовується для збереження даних, таких як номер телефона, пін-код і статус сигналізації. Збереження цих даних в *EEPROM* необхідне на випадок перебоїв енергопостачання. Таким чином система завжди буде знати номер телефона та пін-код, а статус сигналізації дозволить системі увімкнути сигналізацію, якщо вона була активна до виключення світла.

При кожному включенні системи йде зчитування енергонезалежної пам'яті та перевірка на наявність в ній номера телефона, пін-коду а також статусу сигналізації, який записується в *EEPROM* при кожній активації/деактивації сигналізації.

```
String newPhone, newPin, newSig;
phoneAddrOffset = readStringFromEEPROM(0, &newPhone);
pinAddrOffset = readStringFromEEPROM(phoneAddrOffset, &newPin);
readStringFromEEPROM(pinAddrOffset, &newSig);
delay(1000);
if (newPhone.startsWith("+380"))
{
    phone = newPhone;
}
if (newPin.startsWith("pin"))
{
    pin = newPin.substring(3);
}
```

Функція *int readStringFromEEPROM (int addrOffset, String *strToRead)* – зчитує з пам'яті рядок за вказаною адресою зсуву та повертає його.

Якщо номер телефону та пін-код ще не записані, то запускаються функції для їх введення та запам'ятовування.

```

if (phone == "+38" )
{
    lcd.setCursor(0, 0);
    lcd.print("Enter phone number");
    while (phone == "+38")
    {
        EnterPhone();
    }
}
if (pin == "")
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Create a pin");
    while (pin == "")
    {
        CreatePin();
    }
}

```

Функції *EnterPhone()* і *CreatePin()* – надають користувачу можливість ввести номер телефону та пін-код. Після введення даних запускається функція *writeStringToEEPROM*.

Функція *int writeStringToEEPROM(int addrOffset, String &strToWrite)* записує рядок за вказаною адресою.

Налаштування GSM модуля

```

SerialAT.begin(9600);
lcd.setCursor(0,0);
lcd.print("Activating");
lcd.setCursor(0,1);
lcd.print("GSM module");

```

```

do {
  _response = sendATCommand("AT", true);
  _response.trim();
} while (_response != "OK");
do {
  _response = sendATCommand("AT+CMGF=1;&W", true);
  _response.trim();
} while (_response != "OK");
Serial.println("SIM800L enabled");
lcd.clear();

```

Рядок *SerialAT.begin(9600)*; вказує швидкість обміну даних з *GSM* модулем.

У рядку *_response = sendATCommand("AT", true)* функція *sendATCommand* надсилає команду «AT» *GSM* модулю для налаштування швидкості обміну даними і записує відповідь модуля в змінну *_response*. Команда буде надсилатись до тих пір, поки модуль не відповість «OK».

У рядку *_response = sendATCommand("AT+CMGF=1;&W", true)* надсилається команда «AT+CMGF=1;&W» для ввімкнення текстового режиму *SMS*. Після того, як модуль відповість «OK», налаштування буде закінчено.

Активація та деактивація сигналізації виконується за допомогою введення пін-коду на матричній клавіатурі. Якщо пін-код введено правильно, то сигналізація активується або деактивується в залежності від поточного стану системи.

```

char customKey = customKeypad.getKey();
if (customKey)
{
  buttons += customKey;
  lcd.setCursor(k, 1);
  lcd.print('*');
  k = k + 1;
  if (k == 4)

```



```

    {
        lcd.setCursor(0, 1);
        currentMillis = millis();
        previousMillis = currentMillis;
        while (currentMillis - previousMillis < 5000)
        {
            currentMillis = millis();
            customKey = customKeypad.getKey();
            Serial.println(customKey);
            delay(100);
            if (customKey == '#')
            {
                access = true;
                break;
            }
            else
            {
                access = false;
            }
        }
    }
}

```

Система завжди очікує введення пін-кода: `char customKey = customKeypad.getKey()`. В змінну `customKey` записується значення нажатої кнопки. Якщо була введена цифра, то у рядку `buttons += customKey` вона записується до змінної `buttons`.

Після того як було введено 4 цифри система очікує натиснення кнопки «#» упродовж 5 секунд для перевірки пін-кода.

```

if (buttons == pin)
{

```

```

if (signalization == true)
{
    waiting = false;
    lcd.print("Alarm diactivated");
    SignalizationOff();
}
else
{
    lcd.print("Alarm activated");
    waiting = true;
    waitingOn = millis();
    waitingOff = millis();
    SignalizationOn();
}
delay(1000);
lcd.clear();
k = 0;
buttons = "";
}
else
{
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print("Access denied!");
    delay(1000);
    lcd.clear();
    k = 0;
    buttons = "";
}
}

```

Якщо введено правильний пін-код і сигналізація була ввімкнена запускається функція *SignalizationOff()* яка вимикає сигналізацію і записує її стан в *EEPROM*. Змінна *waiting* вказує чи потрібно вмикати попереджувальний звуковий сигнал і надавати час, щоб покинути приміщення.

Якщо введено правильний пін-код і сигналізація була вимкнена запускається функція *SignalizationOn()* яка вмикає сигналізацію і записує її стан в *EEPROM*.

Взаємодія з GSM-модулем.

```
if (SerialAT.available()) {  
    if (_response.startsWith("+CMTI:")) {  
        int index = _response.lastIndexOf(",");  
        String result = _response.substring(index+1, _response.length());  
        result.trim();  
        _response=sendATCommand("AT+CMGR="+result, true);  
        parseSMS(_response);  
        sendATCommand("AT+CMGDA=\"DEL ALL\"", true);  
    }  
}
```

У рядку *if (_response.startsWith("+CMTI:"))* перевіряємо чи прийшло *SMS*, після чого отримуємо вміст *SMS*. Далі запускаємо функцію *parseSMS()*, у яку передаємо вміст повідомлення, для того щоб виділити з нього номер телефону та команду. Якщо номер телефону є номером власника, то виконується надіслана команда.

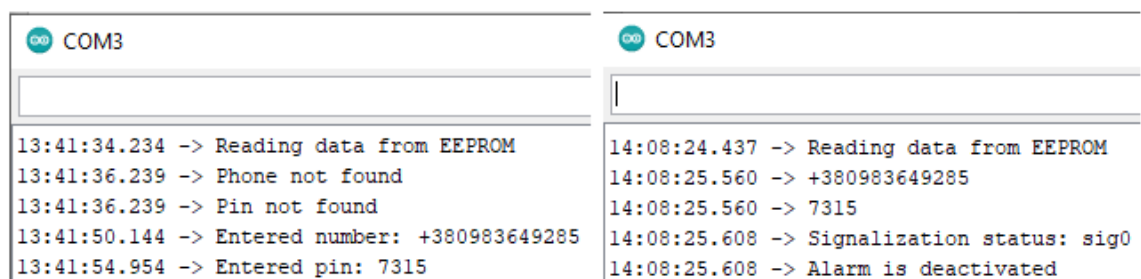
Після цього посилаємо команду модуля для видалення повідомлення *sendATCommand("AT+CMGDA=\"DEL ALL\"", true)*. Повідомлення необхідно видаляти, оскільки вони зберігаються в пам'яті сім-карти, яка є доволі невеликою.

Весь програмний код скетчу для систем безпеки знаходиться в додатку.

3.2. Тестування системи

Тестування буде проводитися за допомогою вбудованих інструментів середовища розробки *Arduino IDE*. Для тестування роботи системи було розроблено спеціальний скетч, який буде виводити інформацію в послідовний порт.

Тест №1. Перевірка запису і зчитування даних з енергонезалежної пам'яті. В пам'яті нічого не записано. З клавіатури вводяться номер телефону та пін-код, після чого вимикається живлення.



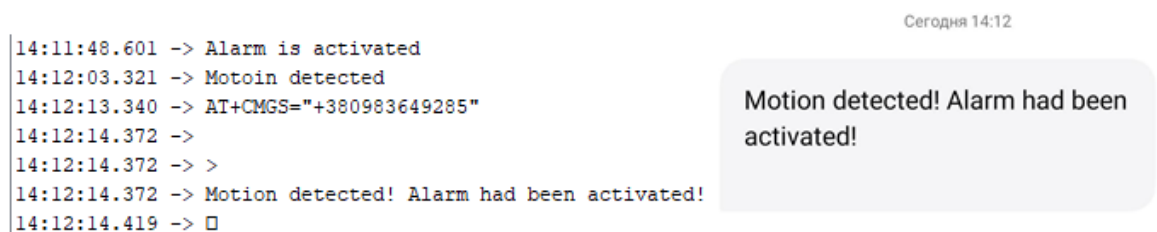
```
COM3
13:41:34.234 -> Reading data from EEPROM
13:41:36.239 -> Phone not found
13:41:36.239 -> Pin not found
13:41:50.144 -> Entered number: +380983649285
13:41:54.954 -> Entered pin: 7315

COM3
14:08:24.437 -> Reading data from EEPROM
14:08:25.560 -> +380983649285
14:08:25.560 -> 7315
14:08:25.608 -> Signalization status: sig0
14:08:25.608 -> Alarm is deactivated
```

Рис. 3.2. Результат тесту №1

З результату тесту, який зображений на рис. 3.2, можна побачити, що після відновлення подачі живлення система зчитала з пам'яті введені раніше дані.

Тест №2. Перевірка роботи системи при виявленні руху.



```
Сегодня 14:12
14:11:48.601 -> Alarm is activated
14:12:03.321 -> Motion detected
14:12:13.340 -> AT+CMGS="+380983649285"
14:12:14.372 ->
14:12:14.372 -> >
14:12:14.372 -> Motion detected! Alarm had been activated!
14:12:14.419 -> □

Motion detected! Alarm had been activated!
```

Рис. 3.3. Результат тесту №2

Як видно з рис. 3.3, було виявлення руху система надала 10 секунд для вимкнення сигналізації і після чого відправила повідомлення на мобільний номер власника.

Тест №3. Перевірка роботи команди «*Turn off*».

```
14:14:47.398 -> +CMTI: "SM",1
14:14:47.398 -> AT+CMGR=1
14:14:48.519 ->
14:14:48.519 -> +CMGR: "REC UNREAD","+380983649285","", "21/05/29,14:14:38+12"
14:14:48.613 -> Turn off
14:14:48.613 ->
14:14:48.613 -> OK
14:14:48.613 ->
14:14:48.613 -> Phone: +380983649285
14:14:48.660 -> Message: Turn off
14:14:48.660 -> Alarm deactivation
14:14:48.660 -> Alarm is deactivated
14:14:48.707 -> AT+CMGDA="DEL ALL"
14:14:49.735 ->
14:14:49.735 -> OK
```

Рис. 3.4. Результат тесту №3

З результату тесту, який зображений на рис. 3.4, видно, що система отримала і обробила повідомлення з номера власника, після чого сигналізація була вимкнена.

Тест №4. Перевірка роботи команди «*Turn on*».

```
14:15:29.724 -> +CMTI: "SM",1
14:15:29.771 -> AT+CMGR=1
14:15:30.852 ->
14:15:30.852 -> +CMGR: "REC UNREAD","+380983649285","", "21/05/29,14:15:20+12"
14:15:30.944 -> Turn on
14:15:30.944 ->
14:15:30.944 -> OK
14:15:30.944 ->
14:15:30.944 -> Phone: +380983649285
14:15:30.991 -> Message: Turn on
14:15:30.991 -> Alarm activation
14:15:30.991 -> Alarm is activated
14:15:31.037 -> AT+CMGDA="DEL ALL"
14:15:32.060 ->
14:15:32.060 -> OK
```

Рис. 3.5. Результат тесту №4

З результату тесту, який зображений на рис. 3.5, видно, що система отримала і обробила повідомлення з номера власника, після чого сигналізація була ввімкнена.

Тест №5. Перевірка роботи команди «Status».

```
14:20:21.436 -> +CMTI: "SM",2
14:20:21.436 -> AT+CMGR=2
14:20:22.557 ->
14:20:22.557 -> +CMGR: "REC UNREAD","+380983649285","", "21/05/29,14:20:12+12"
14:20:22.602 -> Status
14:20:22.602 ->
14:20:22.602 -> OK
14:20:22.649 ->
14:20:22.649 -> Phone: +380983649285
14:20:22.649 -> Message: Status
14:20:22.649 -> AT+CMGS="+380983649285"
14:20:23.672 ->
14:20:23.672 -> >
14:20:23.672 -> Signalization is inactive
14:20:23.719 -> □
14:20:24.757 -> Signalization is inactive
14:20:24.804 -> >
14:20:24.804 -> >
14:20:24.804 -> AT+CMGDA="DEL ALL"
14:20:27.515 ->
14:20:27.515 -> +CMGS: 47
14:20:27.515 ->
14:20:27.515 -> OK
```

Рис. 3.6. Результат тесту №5

З результату тесту, який зображений на рис. 3.6, видно, що система отримала і обробила команду з номера власника, після чого відправила інформацію про статус системи.

3.3. Керівництво користувача

Налаштування системи. При першому включенні, система запропонує ввести номер телефона, на це номер будуть надсилатися повідомлення, а також з цього номера можна буде відправляти команди для керування системою. Після введення номера ви можете підтвердити, що номер введено правильно, натиснувши на клавішу «#», або скасувати введення, натиснувши на «*», якщо номер введено неправильно.



Рис. 3.7. Введення номера телефону

Після введення номера телефону система запропонує створити пін-код для активації/деактивації сигналізації. Пін-код повинен складатися з 4-х цифр. Після введення пін-коду ви можете підтвердити, що пін-код введено правильно, натиснувши на клавішу «#», або скасувати введення, якщо бажаєте змінити пін-код, натиснувши на «*». Тепер система готова до роботи.



Рис. 3.8. Введення пін-коду

Активация сигналізації. Для активації сигналізації необхідно ввести пін-код, якщо він був введений правильно, то на екрані з'явиться напис, що сигналізація активована.



Рис. 3.9. Активація сигналізації

Після цього увімкнеться попереджувальний звуковий сигнал і буде надано 30 секунд для того, щоб покинути приміщення.

Деактивація сигналізації. Після повернення до дому датчик однозначно помітить рух і вам буде надано 30 секунд для того, щоб вимкнути сигналізацію. При правильному введенні пін-коду, на екрані з'явиться напис, що сигналізація деактивована.



Рис. 3.10. Деактивація сигналізації

Команди для керування системою. Для віддаленого керування системою можна використати команди, які необхідно відправити на номер системи. Доступні команди:

- «*Commands*» надасть список команд;
- «*Turn on*» увімкне сигналізацію;
- «*Turn off*» вимкне сигналізацію;

«*Status*» надасть інформацію про поточний стан сигналізації (активована або деактивована).

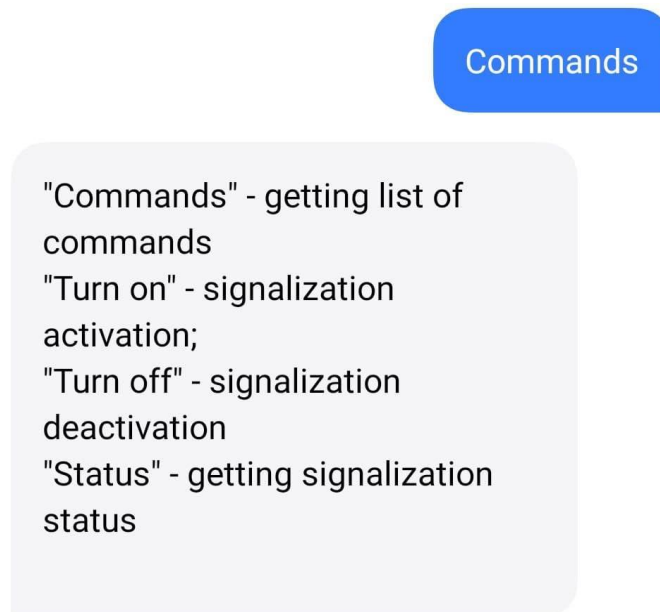


Рис. 3.11. Результат відправки команди «*Commands*»

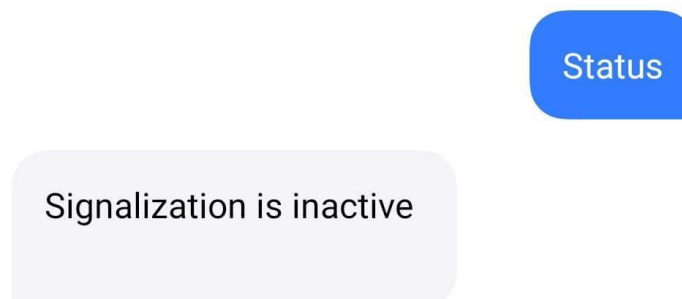


Рис. 3.12. Результат відправки команди «*Status*», якщо сигналізація деактивована

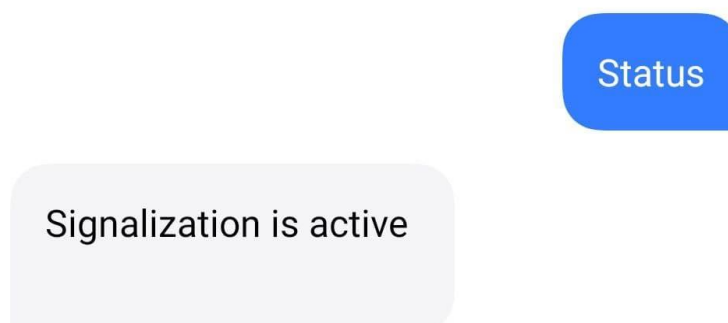
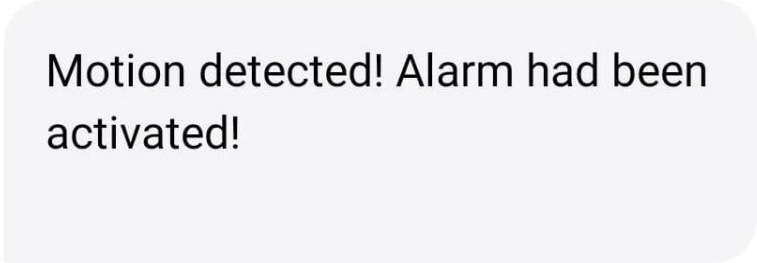


Рис. 3.13. Результат відправки команди «*Status*», якщо сигналізація активована

Виявлення руху. Якщо датчики помітять рух і сигналізація не буде вимкнена упродовж 30 секунд, то увімкнеться постійний звуковий сигнал, а на телефон власника надійде повідомлення про те, що було виявлено рух.



Motion detected! Alarm had been activated!

Рис. 3.14. Повідомлення про виявлення руху

3.3. Висновки до розділу

У третьому розділі було описано розробку системи безпеки в *Smart House*. Під час виконання було створено пристрій на базі *Arduino*, а також розроблено систему та проведено її тестування. Були проведені такі тести як, перевірка працездатності датчику руху, *GSM*-модуля, перевірка роботи системи після відновлення подачі живлення, а також перевірка віддаленого керування системою. Було розроблено керівництво користувача, яке включає в себе кроки користування системою.

ВИСНОВКИ

Метою дипломного проєкту було створення недорогої апартно-програмної системи безпеки в *Smart House*. Протягом виконання дипломного проєкту були вирішені наступні задачі:

- проведено аналіз існуючих систем безпеки, та виділенні основні функції, які варто реалізувати;
- проведено аналіз різних типів мікроконтролерів на базі *Arduino*;
- проведено аналіз пристороїв, які використовувались під час розробки системи;
- проведено аналіз інструментів для проєктування програмної частини проєкту, що використовувалися під-час розробки системи;
- написано програмний код апаратної частини системи безпеки у середовищі *Arduino IDE*;
- систему було успішно протестовано і створено керівництво користувача.

Під час написання дипломного проєкту було ознайомлено з платформою *Arduino*. Ця платформа є розповсюдженою та зручною у використанні.

Система безпеки в *Smart House* – це система, яка забезпечує безпеку будинку або квартири. Рівень безпеки залежить від набору датчиків, які встановлені. Кожна система безпеки передбачає установку керуючого пристрою, клавіатури для включення / вимкнення сигналізації, звукового сигналу тривоги і датчиків руху, по одному на кімнату. У професійних системах також можуть бути встановлені датчики газу (в приміщенні, де знаходиться газове обладнання), датчики диму (в приміщеннях, де можливі загоряння), датчики затоплення (в приміщеннях, де знаходяться комунікації), датчиків розбиття скла, датчиків відкриття дверей.

Під час виконання дипломного проєкту було розроблену систему безпеки на базі плати *Arduino Mega*, з використання інфрачервоного датчика руху, *GSM*-модуля, клавіатури для введення пін-коду і номера телефону, дисплей і п'єзоелектричного випромінювача звуку, який виконує роль звукового сигналу

тривоги. З'єднання всіх компонентів системи було виконано за допомогою дротів, таке рішення є дешевшим, ніж використання бездротової технології зв'язку. Система здатна виявляти рух та повідомляти про це власника житла, також є можливість віддаленого керування за допомогою *SMS* команд.

Створена система буде корисною для людей, які на довгий час залишають свої квартири або будинки, а також людей, які хочуть убезпечити своє житло.

СПИСОК БІБЛОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бойченко С.В., Іванченко О.В. Положення про дипломні роботи (проекти) випускників Національного авіаційного університету. – К.: НАУ, 2017. – 63 с.
2. ДСТУ 3008-95 Документація. Звіти у сфері науки і техніки. Структура і правила оформлення : видання офіційне, Київ : Держстандарт України, 1995. – 38 с.
3. <https://topguard.ua> [Електронний ресурс]. – Режим доступу:
<https://topguard.ua/poleznoye/421-sistemy-bezopasnosti-dlya-doma>
4. <https://helpersystems.com> [Електронний ресурс]. – Режим доступу:
<https://helpersystems.com/blog/chim-drotova-sistema-rozumnogo-domu-krasha-za-bezdrotovu/>
5. <https://en.wikipedia.org> [Електронний ресурс]. – Режим доступу:
https://en.wikipedia.org/wiki/Home_automation
6. <https://www.safewise.com> [Електронний ресурс]. – Режим доступу:
<https://www.safewise.com/home-security-faq/how-do-security-systems-work/>
7. <http://know.smartelements.ru> [Електронний ресурс]. – Режим доступу:
<http://know.smartelements.ru/main:sensors>
8. <https://doc.arduino.ua> [Електронний ресурс]. – Режим доступу:
<https://doc.arduino.ua/ru/prog/>
9. www.theengineeringprojects.com [Електронний ресурс]. – Режим доступу:
www.theengineeringprojects.com//2018/10/introduction-to-arduino-ide.html
10. <http://codius.ru> [Електронний ресурс]. – Режим доступу:
<http://codius.ru/articles/251>
11. <https://ravesli.com/> [Електронний ресурс]. – Режим доступу:
<https://ravesli.com/uroki-cpp/>

12. Петин В.А. Проекты с использованием контроллера Arduino. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2015. – 464 с.: ил. – (Электроника)
13. <https://alexgyver.ru> [Электронный ресурс]. – Режим доступа:
<https://alexgyver.ru/lessons/arduino-reference/>
14. Монк Саймон. Програмируем Arduino. Проффессиональная работа со скетчами. – СПб.: Питер, 2017. – 252 с.
15. Джереми Блум. Изучаем Arduino: инструменты и методы технического волшебства. – СПб.: БХВ-Петербург, 2015. – 336 с.