Хмельницький національний університет
Факультет програмування
та комп'ютерних і телекомунікаційних систем
Кафедра комп'ютерної інженерії та системного програмування

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Програмно-технічний засіб охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi
Назва теми

КвРКІ 180238.18.02.15 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»
Шифр, назва

Освітня програма «Комп'ютерна інженерія»
Назва

Виконав: студент IV курсу, група КІ–18–2 _____ А.І. Мандрик
Підпис, Ініціали, прізвище

Керівник _____ К.Ю. Бобровнікова
Підпис, дата, Ініціали, прізвище

Нормоконтролер _____ С.М. Лисенко
Підпис, дата, Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної інженерії та інформаційних систем _____ Т.О. Говорущенко
Підпис, Ініціали, прізвище

«_3_» _червня_ 2022 р.

Хмельницький 2022

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЯ ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

" 11 " 01 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Мандрику Андрію Ігоровичу
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-технічний засіб охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi.
Керівник проекту (роботи) Бобровнікова К.Ю., к.т.н.
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.05.2022 р. № 18

2. Строк подання студентом проекту (роботи) на кафедру 03.06.2022 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Дослідження функцій охоронної сигналізації в кіберфізичної системи «розумний будинок» та постановка задачі на
Проскування системи охоронної сигналізації та нагляду в кіберфізичній системі «розумний будинок» на платформі Raspberry Pi
Програмно-апаратна реалізація та тестування засобу охоронної сигналізації та нагляду в кіберфізичній системі "розумний будинок" на платформі Raspberry Pi

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)
Схеми апаратних з'єднань
Інтерфейси програмно-апаратного засобу на платформі Raspberry Pi
Блок-схема роботи охоронної системи

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата завдання видав	Підпис, дата завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КІСП		
Антиплагіат	Нічепорук А.О., доцент кафедри КІСП		

7. Дата видачі завдання «_11_» 01 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2022
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	25.01.2022
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	27.02.2022
4	Робота над розділом 2 – проєктування підсистеми	13.03.2022
5	Робота над розділом 3 – програмно-апаратна реалізація підсистеми	07.04.2022
6	Оформлення пояснювальної записки згідно вимог	20.05.2022
7	Попередній захист ВКР	24.05.2022
8	Захист ВКР на засіданні ЕК	Червень 2022 року

Студент Підпис А.І. Мандрик (Ініціали, прізвище)

Керівник проекту (роботи) Підпис К.Ю. Бобровнікова (Ініціали, прізвище)

№ рядка	формат	Позначення	Найменування	Кіл. листів	№ екз	Примітка
			Текстові документи			
1		КвРКІ 180238.18.02.15 ПЗ	Пояснювальна записка	82		
			Графічні матеріали			
2		КвРКІ 180238.18.02.15 Е8	Схеми апаратних з'єднань	1		
3		КвРКІ 180238.18.02.15 Е8	Інтерфейси програмно-	1		
			апаратного засобу			
			на платформі Raspberry Pi			
4		КвРКІ 180238.18.02.15 Е8	Блок-схеми роботи	1		
			Охоронної системи			

Зм	Арк	№ докум	Підпис	Дата	КвРКІ 180238.18.02.15 ВП		
Розробив		Мандрик			Відомість проекту	Літера	Аркуш 1 / Аркушів 1
Перевір.		Бобровнікова				У	
Н. контр.		Лисенко				ХНУ, КІ-18-2	
Затв.		Говорущенко					

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-технічний засіб охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi».

Автор роботи: Мандрик Андрій Ігорович.

Керівник роботи: Бобровнікова Кіра Юліївна.

Пояснювальна записка: 65 с., 47 рис., 1 табл., 4 дод., 45 джерел.

Графічна частина: 8 презентаційних слайдів.

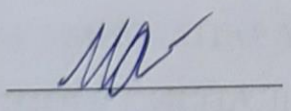
ОХОРОННА СИГНАЛІЗАЦІЯ, КІБЕРФІЗИЧНА СИСТЕМА "РОЗУМНИЙ БУДИНОК", СИСТЕМА НАГЛЯДУ.

Метою роботи є проектування та реалізація програмно-технічного засобу охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi, для автоматизації охоронної сигналізації у кіберфізичній системі "Розумний будинок".

Об'єктом дослідження є процес автоматизації охоронної сигналізації в кіберфізичній системі "Розумний будинок".

Предметом дослідження є програмно-технічний засіб охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi.

Практична цінність роботи полягає в спроектованому та створеному програмно-технічному засобі охоронної сигналізації та нагляду, який може стати складовою частиною в кіберфізичній системі «Розумний будинок», та надає автоматизувати охоронну сигналізацію.



Підпис студента

02.06.2022

Дата

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

PoE – Power over Ethernet

SoC – System on a Chip

GPU – Graphic Processing Unit

RAM – Random Access memory

SD – Storage Device

LPDDR – Low-Power Double Data Rate

BLE - Bluetooth Low Energy

GPIO – General Purpose Input/Output

USB – Universal Serial Bus

CSI – Camera Serial Interface

DSI – Display Serial Interface

HDMI – High-Definition Multimedia Interface

OS – Operating System

IoT – Internet of Things

NoIR – No Infrared filter

RISC – Reduced Instruction Set Computer

MQTT – Message Queue Telemetry Transport

EBCDIC – Extended Binary-Coded Decimal Interchange Code

IMG – Image

ISO – ISO-9660/ Optical Disc Image

ASCII – American Standard Code For Information Interchange

LED – Light Emitting Diode

I2C – Inter-Integrated Circuit

UART – Universal Asynchronous Receiver-Transmitter

SSH – Secure Shell Protocol

DIY – Do It Yourself

ВСТУП

Сьогодні багато людей цікавляться системами домашньої безпеки, навіть ті, хто ніколи раніше не замислювався про їхню покупку. Дослідження, проведене компанією Parks Associates [1], що спеціалізується на аналізі ринку споживчих технологій, показало, новий сегмент домогосподарств почав встановлювати власні системи безпеки. Серед основних причин: зниження витрат, простота установки та бажання краще зрозуміти, що саме роблять ці системи.

Самостійно контрольовані системи дозволяють домовласнику або орендареві віддалено переглядати відеозаписи та отримувати повідомлення при спрацюванні давача руху, дверей або вікна, але при цьому не відбувається автоматичного оповіщення поліції. З іншого боку, контрольовані системи домашньої безпеки DIY передають інформацію до компанії, якій доручено зв'язатися із правоохоронними та іншими органами під час надзвичайної ситуації. Споживачі можуть спати, працювати та подорожувати, знаючи, що система автоматично зателефонує до служби порятунку у разі злому, пожежі чи іншої події.

Більшість цих систем домашньої безпеки, як контрольованих, так і неконтрольованих, є продовженням ринку бездротових технологій "розумного будинку", що швидко зростає. З початку 2000-х років стільникові контрольні панелі, бездротові давачі та інші інновації разом дали споживачам альтернативу традиційним системам безпеки з жорсткою проводкою. Те, що раніше було трудомістким і тривалим процесом, який виконувався майже виключно професіоналами і включав свердління гіпсокартону прокладання проводів, стало набагато простіше зі зростанням доступності бездротових технологій.

Майже половина нових систем домашньої безпеки, придбаних за останні два роки - це системи для персонального використання. Це зрушення викликане побоюваннями споживачів щодо ризиків "професійної" установки, таких як:

– чи не зашкодять підрядники мій будинок;

– чи знають ці монтажники, що роблять;

– чи доведеться мені залишатися вдома і чекати кілька годин, поки установник приїде та виконає роботу;

– чи почуватиметься замовник, впустивши у свій будинок незнайому людину;

Приблизно один із семи осіб, опитаних компанією Parks Associates, повідомив, що вирішив встановити власну систему частково тому, що не хоче пускати монтажників у свій будинок. Тому є безліч причин: грабіжники видавали себе за продавців, які працюють із сигналізацією. Установників звинувачували у крадіжках із будинків, у яких вони працювали.

Саме через ці причини було вирішено створити систему для охорони будинку засобами Raspberry Pi Model B+.

Метою роботи є проектування та реалізація програмно-технічного засобу охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi, для автоматизації охоронної сигналізації у кіберфізичній системі "Розумний будинок".

Об’єктом дослідження є процес автоматизації охоронної сигналізації в кіберфізичній системі "Розумний будинок".

Предметом дослідження є програмно-технічний засіб охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi.

Практична цінність роботи полягає в спроектованому та створеному програмно-технічному засобі охоронної сигналізації та нагляду, який може стати складовою частиною в кіберфізичній системі «Розумний будинок», та надає автоматизувати охоронну сигналізацію.

1 ДОСЛІДЖЕННЯ ФУНКЦІЙ ОХОРОННОЇ СИГНАЛІЗАЦІЇ В КІБЕРФІЗИЧНОЇ СИСТЕМИ «РОЗУМНИЙ БУДИНОК» ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Концепція охоронної системи

Компанії, від незалежних роздрібних торговців до великих монополістів індустрії безпеки пропонують різні варіанти того, що включає поняття DIY. Є навіть люди які будуть увесь дім на основі концепції "розумного будинку", які ретельно підбирають кожне обладнання та ретельно інтегрують його в індивідуальну систему домашньої безпеки [1-2]. Але більшість споживачів шукають простіший, але дуже ефективний проект домашньої безпеки DIY. Досить легко встановити новий маршрутизатор або розумну лампу – чому б не бути таким же простим у встановленні камери та давачів безпеки?

Продавці сигналізацій ставлять простоту використання основою сучасних систем домашньої безпеки. Оскільки багато пристроїв, що використовуються, є бездротовими (тобто вони не залежать від стаціонарних телефонів, жорстких проводів живлення або і того, і іншого), ці рішення є модульними і мобільними. Споживачі можуть вибрати ті пристрої, які їм потрібні, і відмовитися від тих, які не потрібні.

Бездротові камери, давачі та інше обладнання можна від'єднувати, переставляти та переносити з одного місця в інше. Таке обладнання, як камери відео, можна закріпити лише кількома гвинтами. А такі клеї, як на давачі розбиття скла, дозволяють встановлювати обладнання без будь-яких інструментів.

При всій цій простоті роздрібні продавці часто залишають користувачам одне небажане завдання: підключення пристроїв до центрального концентратора. Багато пристроїв заявляють, що вони "підключаються та працюють".

Але давачі і камери, що використовуються в домашніх системах безпеки, часто не є такими, і встановлення першого з'єднання може включати в себе гнітючу

послідовність підключень і відключень, метушню з маршрутизатором і читання посібників [3].

Запрограмоване обладнання працює прямо із коробки. Кожен пристрій ретельно налаштовується для роботи з певним концентратором чи контрольною панеллю. Frontpoint - одна з небагатьох компаній, що надають дійсно запрограмовані пакети DIY, в яких кожен пристрій одразу підключається до концентратора, а також пропонує попереднє програмування будь-яких нових пристроїв, доданих до системи пізніше.

Системи безпеки працюють найкраще у поєднанні з іншими перевіреними заходами щодо підвищення безпеки будинку. Хороший ландшафтний дизайн може зменшити кількість місць, де можуть сховатися грабіжники, та забезпечити меншу кількість точок проникнення. Зміцнення вікон, дверей та замків допоможе запобігти спробам злому. Але хоча такі рішення, як і раніше, важливі, варто підкреслити, що встановлення електронного обладнання для спостереження та моніторингу надзвичайно ефективне і швидко стає одним із найпростіших проектів із забезпечення безпеки будинку в стилі DIY [4].

Крім того, є й інші причини прийняти технологічну революцію у системах безпеки, а саме:

1) Традиційні системи можуть бути дорогими. Статистика, зібрана на цифровому ринку HomeAdvisor, показує, що встановлена підрядником система домашньої сигналізації коштує в середньому близько 700 доларів, а може сягати 1850 доларів. Системи DIY коштують у середньому вдвічі дешевше і починаються від $69. Купівля обладнання заздалегідь також дає змогу продавцям систем сигналізації DIY знизити вартість моніторингу. Середня щомісячна плата на 20 доларів нижча у постачальників рішень DIY у порівнянні з традиційними системами безпеки.

2) Бездротові системи DIY відрізняються надзвичайною стійкістю. Дверні давачі, панелі керування та навіть зовнішні камери можуть працювати від акумулятора або стандартного джерела живлення з резервним акумулятором під

час відключення електроенергії (для роботи камер WiFi маршрутизатору також потрібний резервний акумулятор). Frontpoint Hub використовує як стільникове з'єднання, так і резервне WiFi, гарантуючи продовження моніторингу в порівнянні з системами, що виходять з ладу при збоях стаціонарних та дротових з'єднань [5].

3) Ремонт відбувається швидше, простіше та дешевше. Скарги на високу плату - зокрема один випадок, пов'язаний із заміною батареї за 600 доларів - тільки підігріли зростаючий інтерес до домашніх систем безпеки DIY. Бездротові системи усувають занепокоєння з приводу надійності, кваліфікації та суджень установників. Завдяки докладній документації, відео, мобільним посібникам та підтримці клієнтів на вимогу, авторитетний дилер сигналізації може провести споживача через процес ремонту обладнання протягом кількох хвилин, а не днів.

4) Дистанційне управління, повідомлення та контроль - це лише один із багатьох варіантів використання системи домашньої безпеки DIY. Повідомлення - як текстові, так і електронною поштою - можуть повідомити будь-якій кількості адресатів, коли відкриваються двері, розбивається вікно або не закриваються двері гаража. Такі повідомлення полегшують реагування на злом. Але це далеко не єдине їхнє застосування. Творче використання сенсорних повідомлень може допомогти захистити від зловживання рецептурними препаратами, захистити дітей від небезпечних місць у будинку та забезпечити належний догляд за членами сім'ї з когнітивними порушеннями. Камери можуть контролювати роботу нянь, хатніх робітниць і навіть дітей, які не бажають працювати вдома.

5) Домашні системи безпеки DIY ще не повністю захопили ринок. Але є всі підстави вважати, що це станеться. На додаток до зручності та економічності, вони мають унікальну адаптованість, а все більша частина американців орендує, а не володіє своїм житлом. А більшість людей переїжджають до нового будинку близько 11 разів за своє життя.

6) Наявність портативної системи, що швидко встановлюється і легко адаптується до нового будинку, вкрай важливо - саме це пропонують кращі з сучасних технологій безпеки "розумного будинку".

1.2 Основні поняття концепції Інтернету речей

1.2.1 Що таке Інтернет речей

З моменту появи терміну в 1999 році Інтернет речей (IoT), він перетворився з простого бачення на відчутну реальність. Це можна пояснити широким використанням протоколу Інтернету (IP), зростанням усюдисущих обчислень та постійним удосконаленням аналітики даних, а також іншими факторами розвитку. За оцінками [6], до 2023 року до IoT буде підключено 25,4 мільярда пристроїв.

Однак, незважаючи на постійне розширення, IoT залишається певною мірою незрозумілою концепцією, про яку часто говорять в абстрактних термінах, навіть якщо вона забезпечує очевидні переваги.

IoT можна описати як розширення інтернету та інших мережних підключень до різних давачів і пристроїв - або "мовленням" - що дозволяє навіть простим об'єктам, таким як лампочки, замки та вентиляційні отвори, отримати більш високий ступінь обчислювальних та аналітичних можливостей.

Інтероперабельність – один із ключових аспектів IoT, що сприяють зростанню його популярності.

Підключені або "розумні" пристрої - так часто називають "речі" в IoT - мають можливість збирати дані з навколишнього середовища та обмінюватися ними з іншими пристроями та мережами.

Завдяки аналізу та обробці даних устрою можуть виконувати свої функції практично без участі людини.

Враховуючи кількість підключених пристроїв, що постійно зростає [7], IoT продовжує свій шлях еволюції, додаючи різні шари до даних, які вже передаються і обробляються, і породжуючи складні алгоритми, які призводять до підвищення рівня автоматизації.

А завдяки різноманітності "речей", які можуть бути до нього підключені, IoT забезпечує різноманітні програми як для окремих користувачів, так і для цілих галузей.

1.2.2 Як працює Інтернет речей

"Речі", складові IoT, можуть бути будь-якими - від фітнес-трекера до автономного транспортного засобу. Незалежно від того, яку функцію вони виконують для користувачів, ці пристрої повинні мати наступні компоненти для правильної роботи в якості частин відповідних систем IoT.

Для того, щоб система IoT почала обробляти дані, вони спочатку збираються з навколишнього середовища. Вони збираються давачами в пристроях, які можуть вимірювати явища або зміни в навколишньому середовищі.

Тип даних, що вимірюються пристроєм, залежить від його функції: Це може бути пульс людини у разі фітнес-трекера або відстань до найближчого об'єкта у разі автономного транспортного засобу.

Дані повинні бути передані від пристрою до іншої частини системи IoT, будь то комп'ютер або інший пристрій. А щоб цей зв'язок мав хоч якесь значення, пристрій повинен мати унікальну ідентифіковану присутність в Інтернеті, що досягається за допомогою власної IP-адреси.

Більшість пристроїв IoT здатні виконувати свої основні функції без фізичної взаємодії зі своїми користувачами.

IoT-пристрої повинні бути здатні робити дії на основі даних від своїх давачів та подальшого зворотного зв'язку з мережею. Наприклад, розумна лампочка може увімкнутися за командою свого користувача, навіть якщо він знаходиться за багато миль від неї.

Так само клапан на "розумному" зафводі може автоматично відкриватися або закриватися відповідно до даних, зібраних його давачами на виробничій лінії.

Незважаючи на те, що пристрої зазвичай створюються з урахуванням вимог автоматизації, для систем IoT необхідна наявність інших технологій. Завершальними ланками того, як системи IoT обробляють дані є наступні компоненти.

IoT-шлюз діє як міст для передачі даних різних пристроїв у хмару. Він також допомагає перевести різні протоколи різних IoT пристроїв в один стандартний протокол і відфільтрувати непотрібні дані, зібрані пристроями.

Хмара - це місце, де збираються всі дані різних пристроїв і де програмне забезпечення може отримати ці дані для обробки. Оскільки більшість обробки даних відбувається у хмарі, це знижує навантаження на окремі пристрої.

Інтерфейс користувача передає користувачам дані, зібрані пристроями, і дозволяє користувачам віддавати необхідні команди для виконання пристроями.

Рада з архітектури Інтернету випустила керівний документ, в якому описано чотири канали зв'язку, що використовуються в IoT. Ці чотири моделі також демонструють, як зв'язок між пристроями IoT допомагає розширити цінність кожного пристрою і підвищити якість загального досвіду користувача:

Device-to-Device модель, рисунок 1.1, представляє, як два або більше пристроїв підключаються та обмінюються даними безпосередньо один з одним. Зв'язок між пристроями зазвичай здійснюється за допомогою таких протоколів, як Bluetooth, Z-Wave та Zigbee. Ця модель часто зустрічається в пристроях і пристроях домашньої автоматизації, що носяться, де невеликі пакети даних передаються від одного пристрою до іншого, як, наприклад, від дверного замка до лампочки.

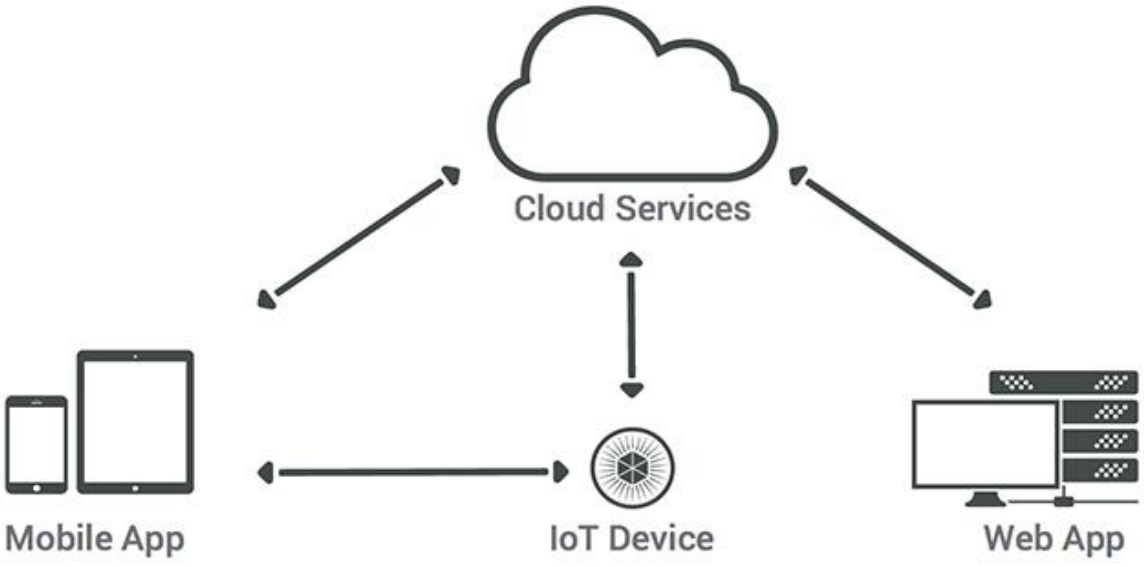


Рисунок 1.1 – Схема роботи Device-to-Device моделі [8]

Багато пристроїв IoT підключаються до хмари, часто за допомогою проводового Ethernet або Wi-Fi. Підключення до хмари дозволяє користувачам та відповідним програмам отримати доступ до пристроїв, що уможливлює віддалене виконання команд, а також оновлення програмного забезпечення пристроїв. Через це з'єднання пристрою також можуть збирати дані про користувачів для покращення роботи своїх постачальників послуг [9].

Перш ніж підключитися до хмари, IoT-пристрої можуть спочатку встановити зв'язок із проміжним пристроєм-шлюзом. Шлюз може перекладати протоколи та додавати додатковий рівень безпеки для системи IoT. Наприклад, у випадку розумного будинку, всі розумні пристрої можуть бути підключені до концентратора (шлюзу), який допомагає різним пристроям працювати разом, незважаючи на різні протоколи підключення.

Обмін даними між пристроями. Як продовження моделі "пристрій - хмара", ця модель дозволяє користувачам отримувати доступ до колекції даних з різних інтелектуальних пристроїв і аналізувати їх. Наприклад, компанія може використовувати цю модель для доступу до інформації з усіх пристроїв, що працюють у будівлі компанії та об'єднаних у хмару. Ця модель також допомагає зменшити проблеми із переносністю даних.

1.2.3 Сфера застосування Інтернету речей

Як інтернет загалом впливає широкий спектр користувачів, і IoT. Залежно від масштабу підключення та кількості задіяних пристроїв, IoT може мати значні та специфічні програми як для окремого користувача, так і для цілого міста. До загальних областей застосування IoT належать такі.

Люди безпосередньо використовують пристрої IoT через технології, які можна носити, наприклад, смарт-годинник та фітнес-трекери, а також пристрої, які допомагають отримувати та збирати інформацію в режимі реального часу.

У домашніх господарствах IoT можна використовувати для створення більш підключеного, енергоефективного та зручного будинку. Різні аспекти підключеного будинку також можуть бути віддалено доступні та контролювати власники будинку за допомогою комп'ютера або портативного смарт-пристрою.

Давачі в автомобілі, що рухається, дозволяють збирати в реальному часі дані про нього і його оточення.

Автономні транспортні засоби використовують різні давачі у поєднанні з передовими системами керування для оцінки навколишнього оточення та, відповідно, самостійного керування.

Завдяки застосуванню IoT на заводах виробники можуть автоматизувати завдання, що повторюються, а також отримати доступ до інформації про будь-яку частину всього виробничого процесу.

Інформація, що надається давачами на заводських верстатах, може допомогти у розробці способів зробити всю виробничу лінію ефективнішою та менш аварійною.

У більш широкому масштабі з впровадженням технологій IoT підприємства можуть стати більш економічними, ефективнішими та продуктивнішими.

Наприклад, офісні будівлі можуть бути оснащені давачами, що дозволяють контролювати рух ліфтів або споживання енергії. Різні галузі промисловості, природно, мають різні сфери застосування IoT:

У охороні здоров'я IoT-пристрою можуть використовуватися для отримання миттєвих та точних даних про стан пацієнтів, а в роздрібній торгівлі IoT-пристрою можуть застосовуватися для допомоги покупцям у пошуку товарів та контролю товарних запасів.

Спільне використання різних пристроїв IoT може охопити міські та громадські райони. IoT-пристрої можуть збирати дані з навколишнього середовища та впливати на нього, допомагаючи керувати різними аспектами міського управління, такими як контроль дорожнього руху, управління ресурсами та громадська безпека.

1.3 Проблеми концепції Інтернету речей у розрізі охоронної системи

IoT - це відносно нова технологія, що розвивається. Як така, вона схильна до деяких істотних проблем, особливо з урахуванням того, що в найближчі роки очікується поява ще більшої кількості пристроїв у мережі. Нижче наведено деякі аспекти, в яких IoT продовжує стикатися з певними проблемами.

Незважаючи на розширення сфери застосування, зростаюча кількість підключених пристроїв робить стандартизацію та регулювання IoT складною та стомлюючою справою. Проблеми стандартизації та регулювання можуть змінюватись від технічних проблем до юридичних питань. Фрагментація, наприклад, є технічною проблемою, з якою стикаються користувачі через відсутність стандартів IoT. Різні інтелектуальні пристрої можуть використовувати різні протоколи бездротового зв'язку, такі як Blutooth, Wi-Fi, Zigbee [10] та 5G, що ускладнює зв'язок усередині систем IoT. З іншого боку, відсутність регулювання наголошує на існуючих проблемах, пов'язаних з інтернетом, а також додає ще один рівень складності до цих проблем. Одним із прикладів є визначення відповідальності:

У разі виникнення дефектів та порушень, пов'язаних з використанням пристроїв IoT, відсутність регулювання ускладнює визначення відповідальності. Стандарти та правила впливають на загальну якість послуг, що надаються технологіями IoT, і тому стосуються всіх зацікавлених сторін IoT, будь то індивідуальні користувачі, виробники пристроїв або організації, що інтегрують технології у свої процеси.

Зі збільшенням різноманітності особистої інформації, що передається через Інтернет, зростає і розуміння конфіденційності. IoT ще більше ускладнює це питання, оскільки розширює типи даних, що записуються та передаються через Інтернет. Оскільки IoT працює краще, отримуючи якомога детальніше уявлення про навколишнє середовище, це створює компроміс між конфіденційністю користувача та якістю обслуговування. Визначити точки, у яких збирання даних

має бути обмежене, або навіть повністю припинити збирання даних з міркувань конфіденційності, також складно, особливо з урахуванням автоматизованого характеру більшості систем IoT.

Проблеми безпеки завжди будуть присутні під час роботи з даними та інформацією. IoT додає свої власні проблеми безпеки, оскільки має доступ до широкого спектру особистої інформації та тісно інтегрована в індивідуальну та організаційну діяльність.

Ці характеристики IoT роблять технологію привабливою для кіберзлочинців [11-12]. Крім того, будь-яке порушення, атака або вразливість одного пристрою або системи IoT послаблює загальну безпеку відповідних мереж.

До інших загроз безпеки, пов'язаних з технологіями IoT, належать такі:

– однорідність "розумних" пристроїв, що серійно випускаються, означає поширення одних і тих же можливих вразливостей;

– автоматизація систем iot ускладнює виявлення вразливостей та порушень через зниження необхідності втручання людини;

– середовище, в якому розгорнуто пристрої iot, робить ці пристрої вразливими до непередбачених фізичних загроз, коли зловмисники можуть безпосередньо розкривати пристрої;

– взаємопов'язаність систем iot робить кожну частину системи можливістю витоку даних і кібератак, які можуть поширитися інші зачеплені мережі;

Для різних типів пристроїв та систем IoT можуть застосовуватись різні методи забезпечення безпеки.

Однак забезпечення безпеки IoT за збереження його актуальності є загальною відповідальністю ключових гравців - від виробників IoT до кінцевих користувачів.

Сильні засоби захисту можуть бути інтегровані виробниками ще на етапі проектування, а постачальники послуг можуть забезпечити підтримку безпеки шляхом розповсюдження оновлень та виправлень, коли це необхідно.

Користувачі, наприклад, організації, що застосовують інтелектуальні пристрої у своєму бізнесі, можуть постійно контролювати всі свої пристрої, не залежачи повністю від автоматизації IoT.

Адекватні рішення щодо кібербезпеки можуть додати кілька рівнів захисту від непередбачених ризиків для всіх зацікавлених сторін.

Відповідальність за безпеку кожного з учасників IoT не існує у вакуумі. Спільний підхід до забезпечення безпеки IoT дозволяє не лише захистити особисті та корпоративні активи, але й зробити підключений світ більш захищеним.

1.4 Порівняння відомих рішень систем охоронних сигналізацій

Компанія Eufy дотримується свого девізу "розумний будинок спрощений", рисунок 1.2, створюючи прості у використанні пристрої та прилади для розумного будинку.

Рисунок 1.2 – Модулі розумної системи безпеки Eufy

Щодо безпеки, бренд продає широкий асортимент дверних відео дзвінків, розумних замків, камер безпеки, прожекторних камер, камер для приміщень та систем сигналізації.

Ціни зазвичай починаються від $40 і сягають $1 250 [13].

В залежності від продукту деякі функції, доступні в системах безпеки Eufy, включають розпізнавання осіб, пряму трансляцію, двостороннє аудіо, нічне бачення, миттєве сповіщення з мобільного телефону, а також давачі руху з регульованою чутливістю та функцією відстеження руху.

Компанія Eufy отримала п'ятизіркові відгуки за функції та функціональність, продуктивність та надійність, простоту установки, можливості підключення, зручність використання, дизайн та зовнішній вигляд, співвідношення ціни та якості та загальну задоволеність.

Google Nest - це лінійка продуктів для розумного будинку, в яку входить все, починаючи від розумних колонок і потокових пристроїв і до систем домашньої безпеки, включаючи камери з розумними дверними дзвінками на батарейках. Вони зазвичай продаються в роздріб за ціною 329 доларів США кожна, доступні варіанти комплектації. Google Nest, рисунок 1.3, отримав п'ять зірок за продуктивність та надійність і чотири зірки за функції та функціональність, можливості підключення, співвідношення ціни та якості, загальну задоволеність та скрізь.

Рисунок 1.3 – Логотип компанії Google Nest

Камера Nest Cam може використовуватися як у приміщенні, так і на вулиці, і може визначити, чи вловлює пристрій руху, виробленого людиною, твариною або транспортним засобом.

Що стосується дверних відео дзвінків, Nest Doorbell дозволяє відповідати в режимі реального часу [14] або залишати заздалегідь задані повідомлення. Він також працює від акумулятора, що робить його сумісним з усіма будинками та позбавляє необхідності возитися з довгими кабелями.

Компанія Swann, рисунок 1.4, пропонує нові та відремонтовані рішення щодо забезпечення безпеки для дому та бізнесу, включаючи дверні відео дзвінки, Wi-Fi системи безпеки, камери безпеки, охоронне освітлення та системи безпеки. Асортимент систем безпеки компанії, як правило, коштує від $279,95 до $2199,95.

Swann увійшов у наші останні рейтинги з п'ятьма зірками за характеристики та функціональність, плюс чотири зірки у більшості інших категорій, таких як продуктивність та надійність, співвідношення ціни та якості та загальна задоволеність.

Рисунок 1.4 – Набір камер та док станція охоронної системи Swann

Загальна вартість установки системи домашньої безпеки залежить від багатьох факторів , таких [15] як тип і кількість обладнання, яке необхідно підключити (і де), потужність, розташування, кількість вікон у будинку і так далі.

Більшість виробників у наші дні пропонують знижки та плани оплати, тому обов'язково пройдіться по магазинах, перш ніж ухвалити остаточне рішення про покупку.

Наявність системи домашньої безпеки не є 100% надійним варіантом [16] захисту оселі від злому, але вона значно знижує ризик, а деякі дослідження навіть показують, що до 60% зломщиків будуть залякані системою сигналізації. Існують десятки інших переваг системи домашнього відеоспостереження, включаючи віддалений доступ до вашої власності, зниження страхового внеску та, звичайно, душевний спокій.

При цьому система домашньої безпеки коштує недешево, тому краще спочатку провести серйозне дослідження, щоб знайти систему, оптимальну для будинку і бюджету [17].

1.5 Висновки. Постановка задачі

У цьому розділі було розглянуто основні поняття концепції «системи безпеки» а також основні складові інтернету речей. Було сформульовано основні ідеї які лягли в основу цих понять.

Було складено список з найбільш поширених та популярних готових рішень щодо обладнання будинку охоронною сигналізацією, проаналізовано ринок та оцінено рентабельність створення такої охоронної системи власноруч.

Задачею бакалаврської роботи є:

– дослідити ринок та порівняти відомі рішення охоронної сигналізації в кіберфізичній системі «Розумний будинок»;

– здійснити підбір елементної бази для проектування програмно-технічного засобу охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi;

– спроектувати електричні схеми підключення давачів та плат розширення до головної плати;

– виконати тестування кожної складової програмно-технічного засобу охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi;

– виконати з’єднання елементів за допомогою паяльника на друкованій платі;

– створити скрипт, який буде використаний як прошивка плати Raspberry Pi;

– протестувати роботу програмно-технічного засобу охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi.

2 ПРОЄКТУВАННЯ СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ ТА НАГЛЯДУ В КІБЕРФІЗИЧНІЙ СИСТЕМІ «РОЗУМНИЙ БУДИНОК» НА ПЛАТФОРМІ RASPBERRY PI

2.1 Опис вибраних апаратних рішень

Для створення ядра охоронної системи було вибрано програмований мікроконтролер сімейства Raspberry Pi B+ який зображено на рисунку 2.1 [2]. Цей програмований пристрій має значні переваги в лінійці схожих за функціоналом пристроїв.

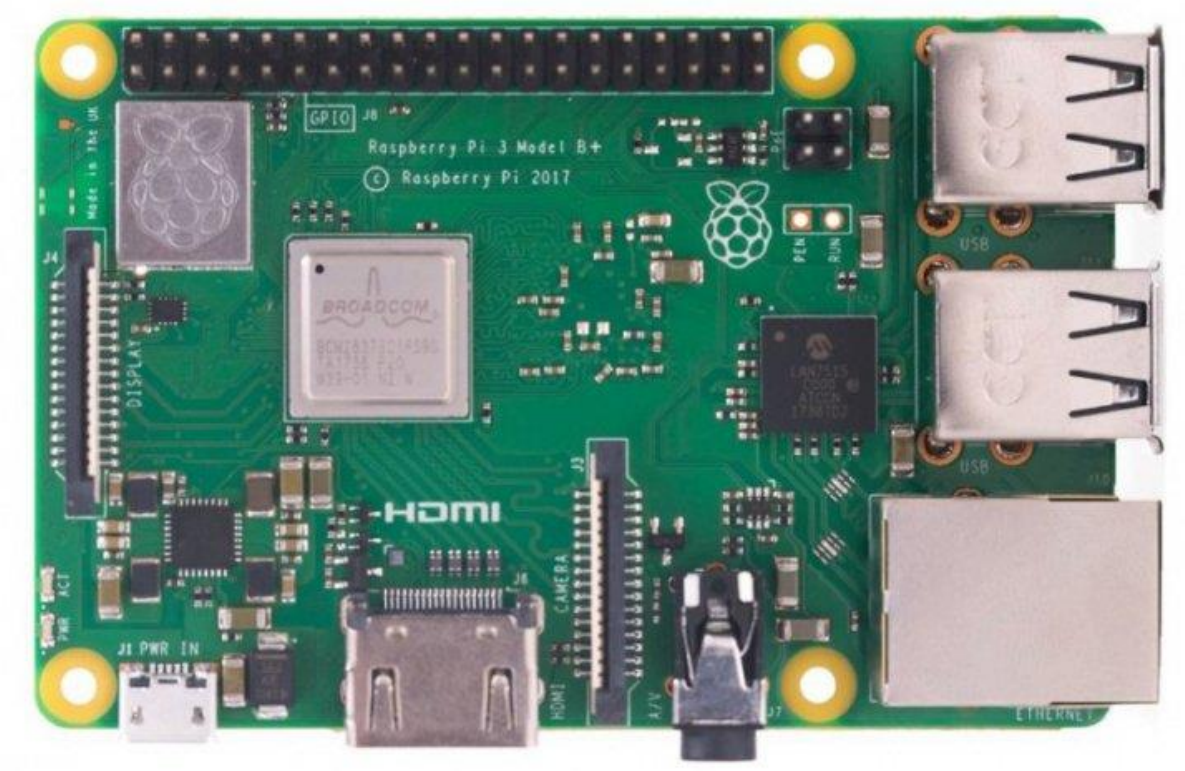


Рисунок 2.1 – Зображення плати з мікроконтролером Raspberry Pi Model B+

Pi 3 Model B+ має швидкий чотириядерний процесор із частотою 1,4 ГГц, також містить модуль Wi-Fi 802.11ac та порт підключення Gigabit Ethernet з можливістю живлення по Ethernet (PoE) [18]. Також він продається окремо, та є бюджетним, але його легко знайти у складі комплектів, що включають кабелі живлення, корпуси та інші доповнення.

Ця модель володіє Soc на основі 64 бітної системи запрограмованої на чіпі BCM2837B0, також на платі є 4х ядерний процесор ARM Cortex-A53 з тактовою частотою 1.44 ГГц. Ця плата має розпаяний відеочіп Broadcom VideoCore IV для виведення зображення по HDMI порту.

Одною з ключових особливостей плати через яку її було обрано є присутність антени Wi-Fi 802.11b/g/n/ac що працює на частотах 2,4 ГГц та 5 ГГц, а також вбудодваний Bluetooth 4.2, BLE [19].

Також є чотири, а не два монтажні отвори, також розробники збільшили кількість контактів на роз'ємі вводу/виводу загального призначення з 26 до 40, що дає більшу гнучкість у використанні Pi для отримання вхідних сигналів та керування іншими пристроями.

Процес установки B+ є простим та ідентичним до встановлення плат цього сімейства. Також потрібна картка microSD об'ємом не менше 4 Гб замість повнорозмірної SD, більшість таких накопичувачів поставляються з адаптером, для можливості використовувати їх у звичайних пристроях читання карток SD. Існує кілька способів встановлення кількох операційних систем однією SD-карту.

Raspberry Pi Foundation поставляє Raspberry Pi OS (раніше називану Raspbian), засновану на Debian – дистрибутиві Linux, Windows 10 IoT Core, RISC OS [20].

Основними мови програмування використовуються Python і Scratch. Стандартна прошивка має закритий код, але доступні неофіційні з відкрити вихідним кодом. Багато інших операційних систем також можуть працювати на Raspberry Pi. Підтримується також мікроядро seL4.

Raspberry Pi OS має високу швидкодію, з мінімальною затримкою відкриває меню та програми, переміщує і змінює розмір вікон без помітних затримок.

Модуль камери Raspberry Pi, зображений на рисунку 2.2 - це офіційний аксесуар Raspberry Pi [21], який працює з усіма моделями Pi і може бути використаний для зйомки зображень фото та відеозображень у високій якості. Він підключається безпосередньо до послідовного порту камери плати Pi інтерфейсу

камери (CSI), який спеціально призначений для цих модулів, щоб забезпечити високошвидкісну роботу. Сама камера є 5-мегапіксельним сенсором з фіксованим фокусом, що підтримує режими 1080p, 720p, і VGA відео режими і фотозйомку.

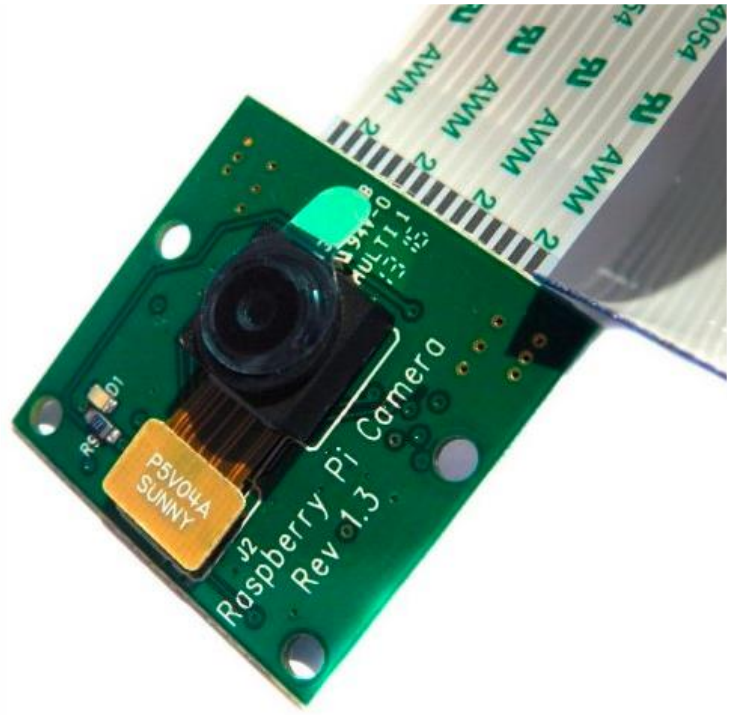


Рисунок 2.2 – Офіційний модуль камери Raspberry Pi, третя ревізія

Дарлінгтон TIP120, рисунок 2.3, що дозволить вмикати та вимикати навантаження до 80 В та 5 А від GPIO порту. У системі буде використано порт B мікроконтролеру MCP23017 для управління виходами, такий принцип роботи підходить для будь-якого [23] з GPIO виходів.

Для того, щоб увімкнути світло або світлодіодний масив з Raspberry Pi GPIO чи схеми розширювача портів, необхідно використати пристрій [22], який дозволить керувати вищим струмом та напругою, ніж можуть забезпечити порти GPIO. Буде використано транзистор

TIP122 – це NPN-транзистор з парою Дарлінгтона. Він функціонує як звичайний NPN транзистор, але оскільки всередині нього знаходиться пара Дарлінгтона, він має хороший номінальний струм колектору близько 5А та коефіцієнт посилення близько 1000.

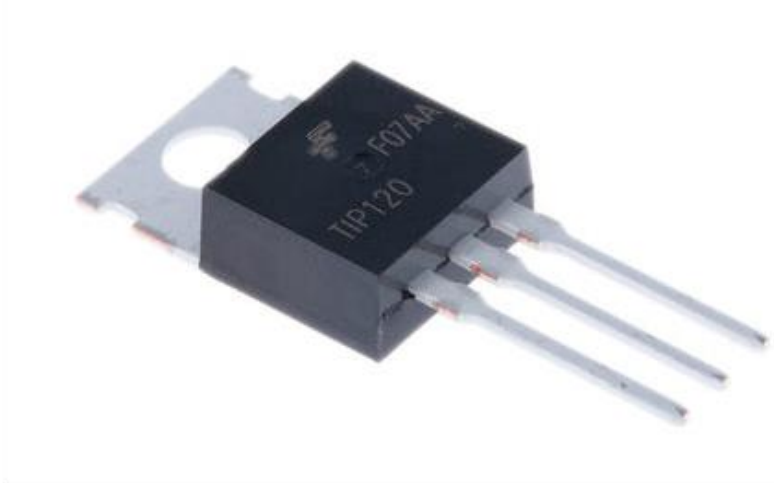


Рисунок 2.3 – Зображення транзистора TIP120 з парою Дарлінгтона

Також може витримувати напругу близько 80 В через колектор-емітер, тому використовується для керування великими навантаженнями. Пара транзисторів Дарлінгтона що були використанні для розроблюваної схеми, рисунок 2.4 [24].

Рисунок 2.4 – Пара транзисторів Дарлінгтона TIP122, що ули використані у пристрої

У проекті використано модуль інфрачервоного освітлення BL0307-50-63 у поєднанні із сумісною камерою. Стандартний модуль камери Raspberry Pi не буде

працювати з інфрачервоним освітленням, тому що він містить інфрачервоний фільтр, але ми можемо використовувати версію NoIR модуль камери [25-26].

Модуль камери Raspberry Pi NoIR такий самий, як і стандартний за винятком того, що в нього не вбудований інфрачервоний фільтр, що означає, що він буде бачити в темряві за допомогою інфрачервоного освітлення. Це робить його придатним для спостереження за борсуками вночі, а також для використання у нашій домашній системі безпеки.

Інфрачервоний світлодіодний масив BL0307-50-63 зображений на рисунку 2.5 або кластер буде використаний, щоб непомітно висвітлити область, яку буде відображено за допомогою камери.

Рисунок 2.5 – Інфрачервоний кластер BL0307-50-63

Стандартна камера Raspberry Pi [27] відмінно підходить для денної зйомки, але для нічних знімків вона не підходить. Для цієї задачі є два рішення: перший – висвітлити область зйомки яскравим світлом при спрацьовуванні PIR-детектора, а другий – використовувати модуль камери Raspberry Pi NoIR, рисунок 2.6, та інфрачервону світлодіодну матрицю, щоб камера бачила у темряві [28].

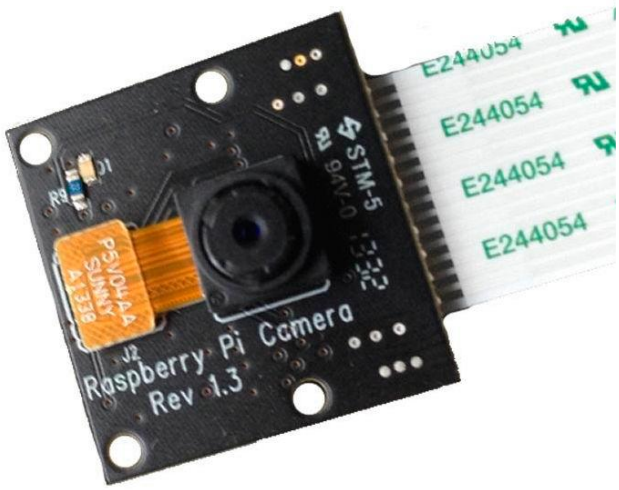


Рисунок 2.6 – Модуль камери Raspberry Pi NoIR

Pi NoIR v2 оснащений 8-мегапіксельним сенсором Sony IMX219, він є звичайним модулем камери, з однією відмінністю: не використовується інфрачервоний фільтр.

Це означає, що фотографії, зроблені при денному світлі, будуть мати погану якість, але це дає можливість бачити в темряві за допомогою інфрачервоного освітлення.

Він підключається до плати завдяки роз'єму на верхній частині, та використовує CSI інтерфейс, спроектований спеціально для роботи з камерами. Давач має невеликі розміри, 25мм x 23мм x 9мм.

Його вага становить близько 3 грам. Для підключення до плати Raspberry Pi використовується короткий стрічковий кабель [29-30].

Невеликі PIR-давачі, такі як Parallax є доволі доступні, давач який використовується, показаний на рисунку 2.7, можна встановити на корпус Raspberry Pi разом із модулем камери, що утворює автономний пристрій.

Потрібний лише один порт для Raspberry Pi, щоб визначити, коли спрацьовує давач руху PIR. Такий давач руху можна підключити за допомогою вбудованого порт GPIO, а не використовувати розширювач портів Raspberry Pi.

Рисунок 2.7 – Давач PIR sensor 28027

Щоб вмикати та вимикати охорону систему було обрано перемикач Lorlin WRL-5-E-S-2-B (рисунок 2.8), та контрольну панель CDVI ECO 100 (рисунок 2.9) ці модулі сертифіковані як водостійкі, що дозволить використовувати їх на вулиці.

Використовуючи автономну охоронну клавіатуру, реалізована можливість дозволити кожному користувачеві мати свій власний код для встановлення та зняття системи з охорони.

Наприклад, CDVI ECO 100 – це недорога клавіатура [31-32], яка може одночасно зберігати данні 100 унікальних паролів. Коли авторизований користувач вводить правильний код, вона ставить систему на охорону, замикаючи внутрішній перемикач. Коли код знову вводиться, клавіатура знімає систему з охорони, розмикаючи перемикач.

Охоронний ключ та панель з кнопками для введення пароля зображені на рисунках 2.8 та 2.9 відповідно. Ці засоби є універсальними, за допомогою них можна як входити в будинок так і вводити пароль чи використовувати ключ для встановлення чи знімання систему не охорону

Рисунок 2.8 – Електронний перемикач з блокуванням WRL-5-E-S-2-B

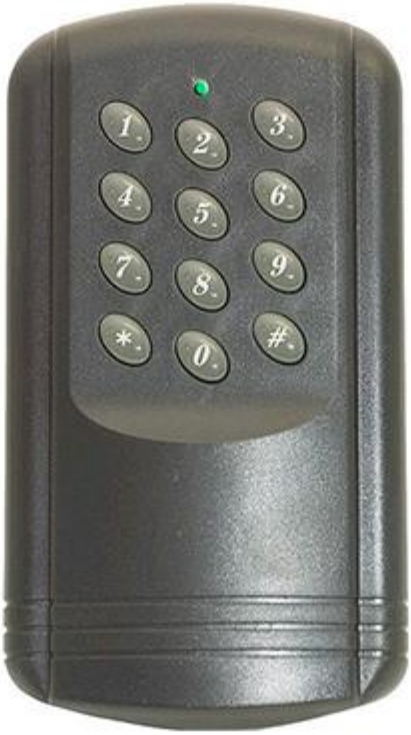


Рисунок 2.9 – Охоронна клавіатура CDVI ECO 100

Для передачі сигналів було використано два варіанта приймачів.

Перший приймач сигналів що підключений до плати на частоті 433 МГц, був розроблений на основі модуля XY-MV-5V, він зображений на рисунку 2.10, разом

з бібліотекою 433-Util. Вона була розроблена для Arduino, але портованана на Raspberry Pi.

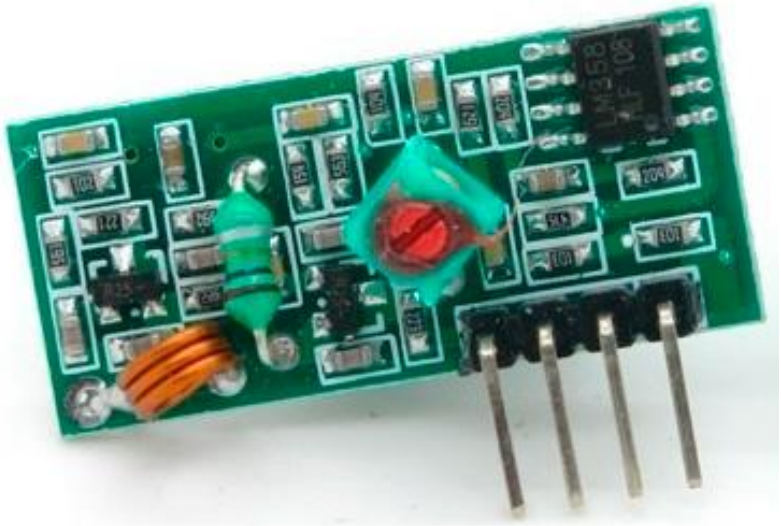


Рисунок 2.10 – XY-MV-5V, модуль приймач сигналів на частоті 433 МГц

Для передачі команд був використаний брелок [33] який буде виконувати передачу сигналів на частоті 433 МГц (рисунок 2.11).

Рисунок 2.11 – Брелок-передавач сигналів на частоті 433 МГц

Модуль приймача може приймати сигнали від брелка дистанційного керування, такого, як показано на рисунку 2.11, який отримує вихідний сигнал у вигляді серії кодованих повідомлень. Ці повідомлення потім декодуються бібліотекою 433- Util [34].

2.2 Опис вибраних програмних рішень

PuTTY - це програма з відкритим вихідним кодом, що використовує мережеві протоколи Telnet та rlogin [35] на платформах Windows та UNIX у поєднанні з емулятором терміналу xterm. По мережі PuTTY використовує всі вищезгадані протоколи для забезпечення віддаленого сеансу роботи на комп'ютері. Це популярний інструмент для текстового спілкування, а також популярна утиліта для підключення Linux-серверів з комп'ютерів на базі операційної системи Microsoft [36]. У головному вікні (рисунок 2.12) PuTTY знаходиться сесія, яка запускається на віддаленому комп'ютері та через яку можна надсилати команди безпосередньо на віддалений комп'ютер.

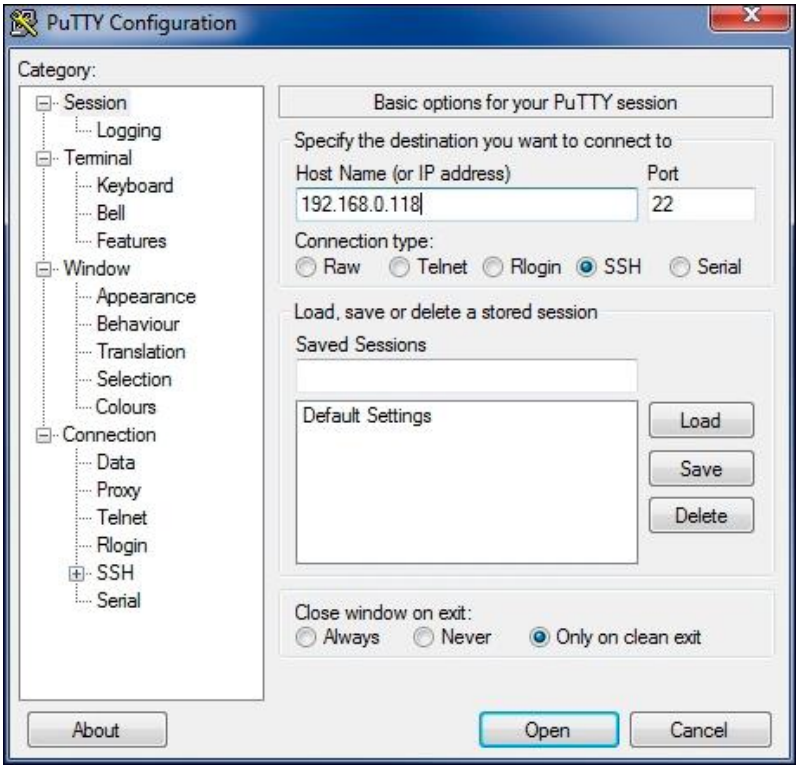


Рисунок 2.12 – Головне вікно програми PuTTY

GParted (GNOME Partition Editor) – одна з найпопулярніших програм для створення розділів. Вона входить до складу більшості сучасних дистрибутивів Linux. Вона також поставляється у великій кількості спеціалізованих дистрибутивів для відновлення даних.

GParted – це графічна програма, тому вона добре підходить для сучасного використання, включаючи менш обізнаних користувачів. Інтерфейс програми GParted зображений на рисунку 2.14. Редактор розділів GNOME Partition Editor [37] дозволяє керувати розділами ваших жорстких дисків, флеш-карток та інших пристроїв для зберігання інформації.

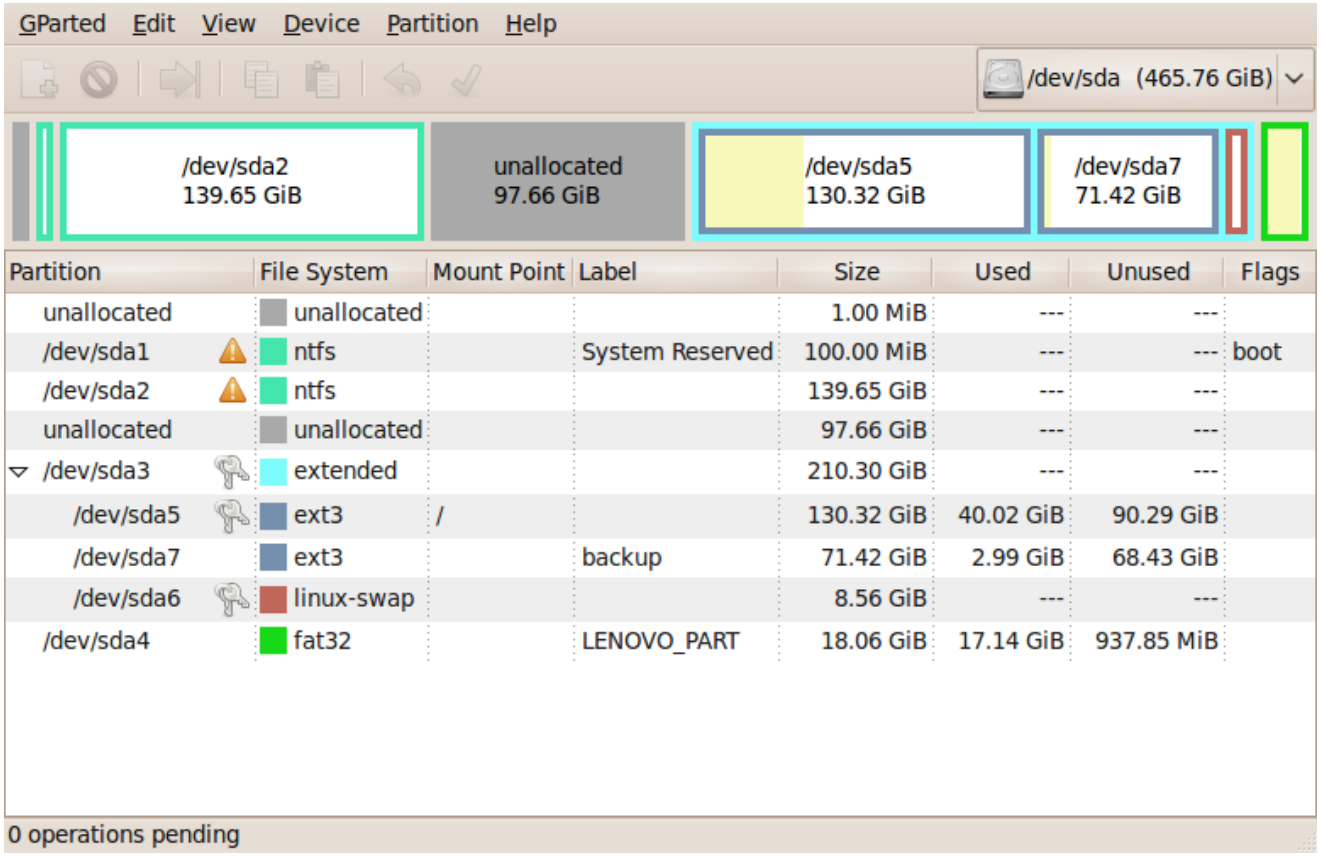


Рисунок 2.14 – Інтерфейс утиліти GParted

У цьому проекті можливості GParted будуть використані для того щоб записати завантажувальний диск на SD карту [38], після чого, прошити мікроконтролер Raspberry Pi Model B+ під операційну систему Linux raspbian-jessie OS.

GParted постачається у більшості дистрибутивів, але її також можна завантажити з офіційного сайту у вигляді архіву.

DD - це утиліта командного рядка для Unix, Unix-подібних операційних систем [39], основною метою якої є перетворення та копіювання файлів. DD використовується для запису та резервного копіювання IMG або ISO файлів операційної системи на картку пам'яті або диск.

У Unix драйвери пристроїв для апаратного забезпечення (наприклад, жорстких дисків) та спеціальні файли пристроїв (такі як /dev/zero та /dev/random) відображаються у файловій системі як звичайні файли; dd також може читати та/або писати з/в ці файли, якщо ця функція реалізована у відповідному драйвері.

Головне вікно утиліти яке відображається коли користувач відкриває програму зображено на рисунку 2.15. В результаті dd можна використовувати для завдань, як резервне копіювання завантажувального сектора жорсткого диска і отримання фіксованої кількості випадкових даних.

Програма dd також може виконувати перетворення даних при їхньому копіюванні, включаючи зміну порядку байтів і перетворення в текстові кодування ASCII і EBCDIC і назад [40-41].

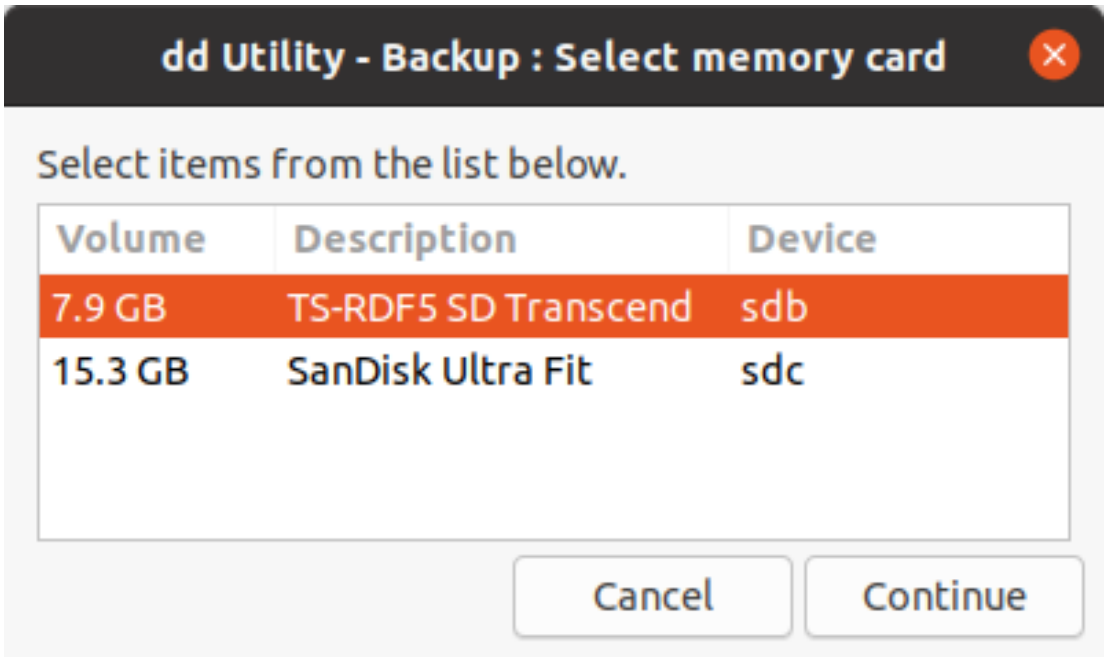


Рисунок 2.15 – Утиліта DD Unix

2.3 Вимоги до апаратної складової програмно-технічного засобу охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi

На основі усіх складових проекту, які були описані в попередньому розділі та деяких допоміжних можна зіставити список вимог до апаратної частини системи охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок":

- raspberry pi model b plus;
- бредборд;
- led маячки;
- резистори різних номіналів;
- кнопки та перемикачі;
- кабелі;
- регулятор напруги ld1117v33;
- керамічні конденсатори номіналом 100 нф, 16в;
- електролітичні конденсатори номіналом 10 пф, 16в;
- плата розширювач 16bit gertboard;
- магнітний давач розімкнення;
- сигнальні кабелі;
- пасивний інфрачервоний давач;
- оптоізолятор 4n25;
- набір діодів різних номіналів;
- 433 мгц приймач та брелок;
- блок живлення на 12в постійного струму;
- модуль камери raspberry pi;
- модуль камери raspberry pi noir;
- інфра червоний led кластер;
- usb веб камера.

Також знадобиться USB-клавіатура та миша, телефонний зарядний пристрій Micro USB для живлення та HDMI [42] або композитний дисплей для підключення Raspberry PI. Окрім цього необхідно стабільне інтернет з’єднання.

2.4 Вимоги до програмної складової проекту

Вибір найкращої мови програмування для Raspberry Pi залежить від мети проекту, його складових та сфери застосування.

Якщо новачок розробляє невеликий проект та лише вивчає програмування в цілому або просто хочете інтегрувати свій Raspberry Pi в будь-який продукт, краще використовувати python. Тому, що існує безліч існуючих бібліотек, книг і підручників, написаних на python, і python є офіційною мовою pi, і насправді назва була натхненна python "py" thon та "pi".

Мета проекту – побудувати архітектуру охоронної системи тому як основні мови будуть застосовуватись C і асемблер на Pi.

Програмувати на C простіше, ніж на асемблері, а результати за швидкістю майже однакові. Мова C можна назвати машинно-незалежним мовою асемблера, і тому, вивчаючи його, користувач глибше занурюєтеся в систему, ніж при використанні інших мов. Це робить його хорошим способом покращити своє розуміння комп'ютерів та обчислень загалом.

У цьому проекті не необхідна велика швидкість обчислень, хоча проект є IoT, але не має модулів які повинні мати миттєвий відклик, тому мова C є ідеальним вибором. Потрібно просто блимати декількома світлодіодами та зчитувати показання давачів, ви можете писати практично будь-якою мовою. Тому для безпосередньої взаємодії з іншими системами та керування підключеннями до зовнішнього обладнанням на повній швидкості, C – найкращий вибір.

Також будуть потрібні наступні утиліти для реалізації проекту:

– dd;

– gnome console;

– gnome partition editor;

– win32 disk imager;

– putty;

– visual studio code.

Для розробки доцільно використовувати операційну систему Windows 8, або новішу, Linux Debian або Ubuntu [43].

Окрім цього для реалізації візуальної складової проекту було використано PHP Web Server він працює за схемою вказаною на рисунку 2.16.

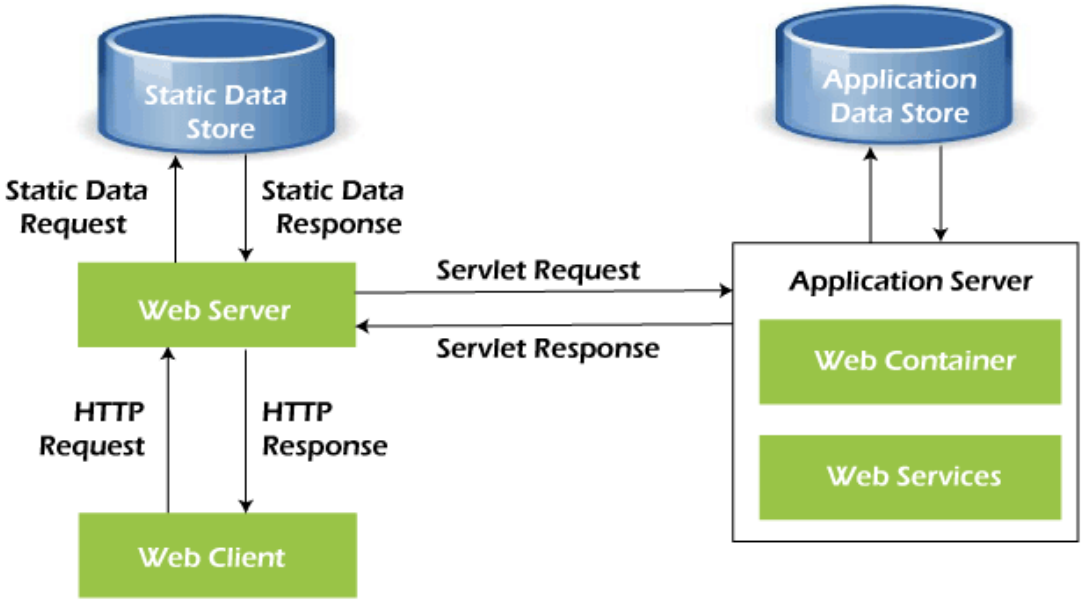


Рисунок 2.16 – Схема функціонування веб сервера [44]

2.5 План розроблення програмно-технічного засобу охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi

Проект можна розбити на декілька головних пунктів, які стануть ключовими точками в розробці.

Налаштування Raspberry Pi Model B Plus [45]. На цьому етапі необхідно приготувати прошивку, зробити завантажуваний SD накопичувач, після чого завантажити мікросхему уже в новому середовищі.

Налаштування середовища розробки всередині Raspberian OS яку було завантажено у пам'ять плати на попередньому кроці.

Приєднання та тестування головних давачів, підключення I2C та UART сумісних пристроїв. Підключення живлення для плати. Проведення тестових запусків проекту у зменшеному масштабі, щоб переконатись що всі компоненти працюють штатно.

Після цього необхідно виконати наступні пукти:

1) Приєднання розширювача I2C та розширювача портів;
2) Підключення магнітного контактного давача;
3) Підключення пасивного інфрачервоного давача;
4) Підключення камер до охоронної системи;
5) Побудова панелі керування на веб сервері;
6) Проведення необхідних підготовок та підключення деяких незначних електронних компонентів для покращення стабільності роботи;
7) Поєднання усіх складових разом, що включає в себе підключення джерел живлення, на 12В та 3,3В, підключення опто-ізолятора;
8) Підключення усіх розширювачів портів до вже готової збірки;
9) Розробка контрольних скриптів;
10) Налаштування усіх портів GPIO, ініціалізація проекту;
11) Перевірка системного циклу;
12) Налаштування автоматичного запуску усіх систем одним скриптом;
13) Створення файлової системи що базується на RAM, щоб уникнути швидкого відпрацювання SD карти;

2.6 Висновки

В цьому розділі було проведено опис та аналіз основних апаратних та програмних рішень шо були використані по ходу створення охоронної системи на базі мікроконтролера Raspberry Pi Model B Plus.

Було складено список усіх необхідних для відтворення проекту вимог, в нього включені як апаратні пристрої, давачі, мікросхеми, електронні пристрої а також утиліти, які покращають процес розробки прошивки та створення середовища для розробки.

В кінці було описано план виконання робіт за яким буде виконана збірка усіх складових у фінальний пристрій.

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ЗАСОБУ ОХОРОННОЇ СИГНАЛІЗАЦІЇ ТА НАГЛЯДУ В КІБЕРФІЗИЧНІЙ СИСТЕМІ "РОЗУМНИЙ БУДИНОК" НА ПЛАТФОРМІ RASPBERRY PI

3.1 Підготовка sd карти до прошивання апаратного пристрою raspberry pi та завантаження ос

Потрібно взяти останню версію образу ОС Raspbian із сайту Raspberry Pi та завантажити ZIP-файл із образом Raspbian OS на комп'ютер. Для підготовки карти було використано утиліту Win32 Disk Imager. Тепер усе готово до завантаження Raspberry Pi.

Потрібно вставитиу SD карту і увімкнути живлення. При першому запуску системи, необхідіно налаштувати та розширити файлову систему, щоб використовувати весь простір на SD-карті. На рисунку 3.1 показано зображення робочого стола.

Тепер потрібно створити образ SD-карти, вставити картку SD у комп'ютер і запустити програму Win32 Disk Imager.

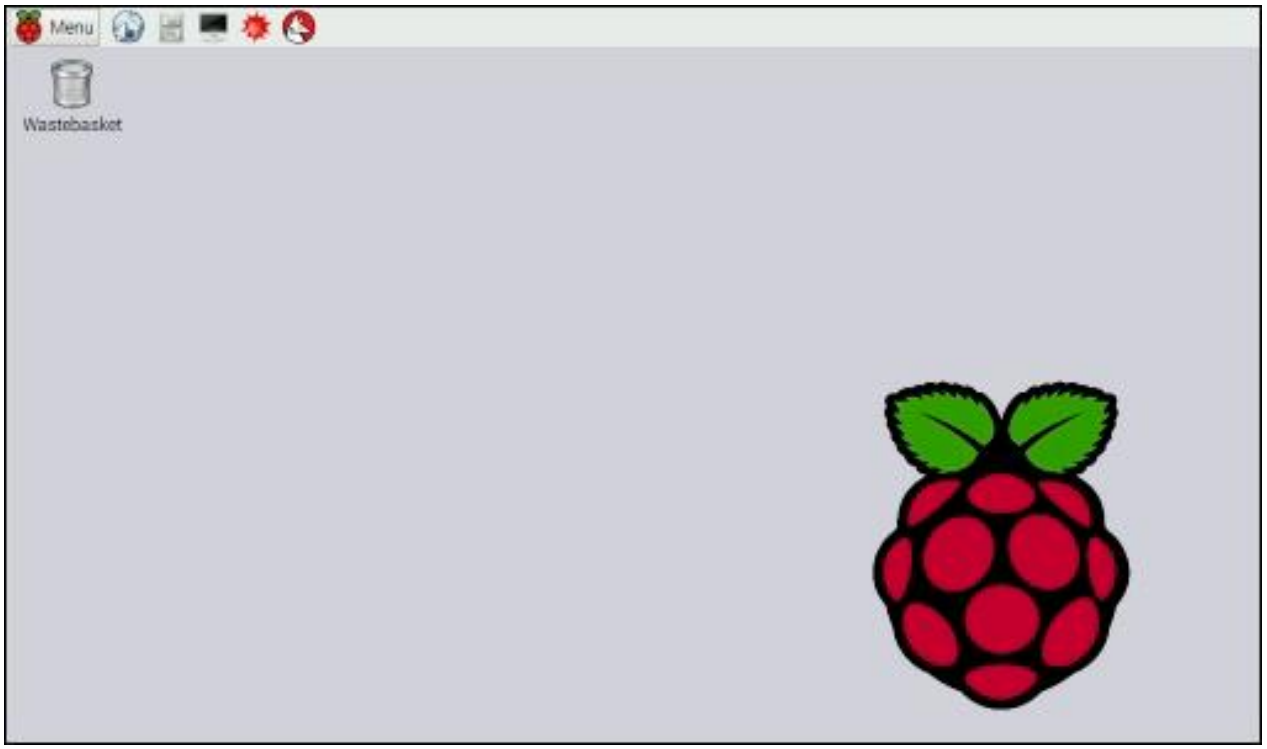


Рисунок 3.1 – Графічний інтерфейс Raspberry Pi Linux

Після цього потрібно вибирати літеру пристрою SD-карти та переконатись, що вона правильна.

На моніторі, що підключений до Pi, можна побачити, що система завантажується. Рекомендовано підключити до плати монітор, принаймні, щоб переконатися, що переконатися, що все працює правильно.

У новій версії Raspbian Jessie завантаження відбувається прямо на робочий стіл з графічним інтерфейсом, що є значною зміною в порівнянні з попередніми версіями, де користувач потрапляв в утиліту raspi-config.

Для того щоб розширити кількість використовуваної пам'яті необхідно відкрити утиліту Raspberry Pi Configuration в меню Menu натиснувши кнопку Preferences, рисунок 3.2.

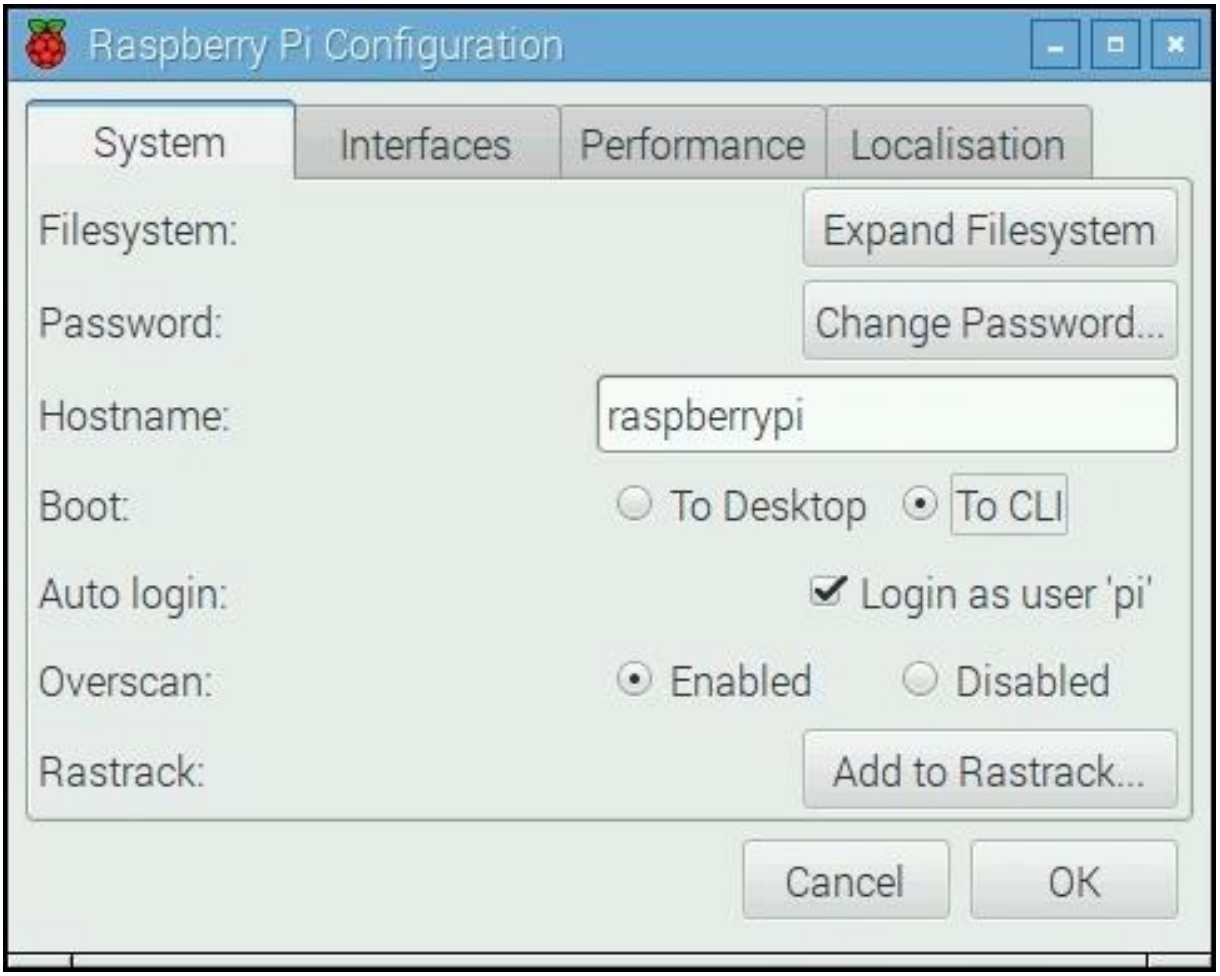


Рисунок 3.2 – Вікно програми Raspberry Pi Configuration Tool

3.2 Підготовка Raspberry Pi до подальшого підключення апаратних пристроїв.

Після завантаження в оболонку і підключення Ethernet домашньої мережі, Raspberry Pi отримає IP-адресу. Pi підключиться мережі та отримає IP-адресу від вашого маршрутизатора автоматично.

Після цього можна побачити IP-адресу, яка була видана безпосередньо перед запрошенням до входу в систему, як показано на рисунку 3.3.

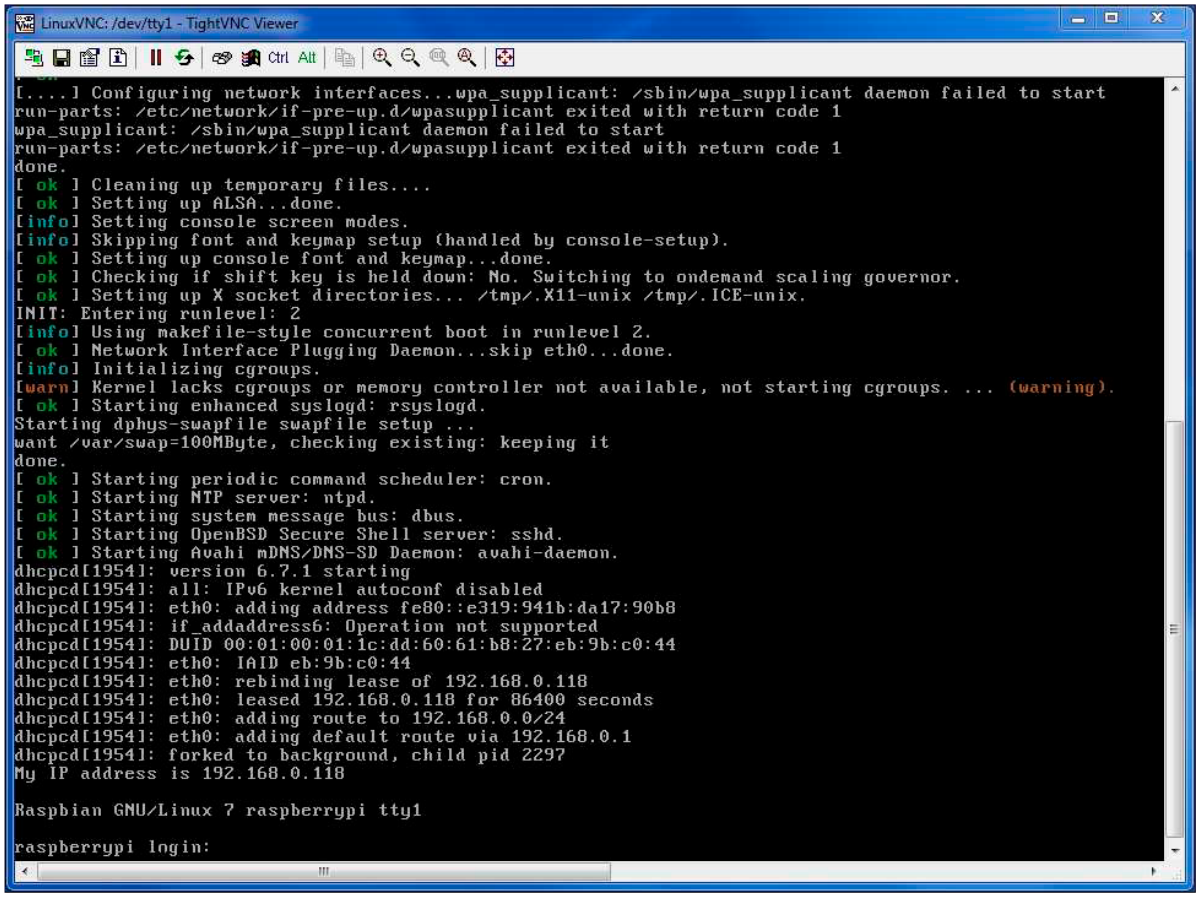


Рисунок 3.3 – Вікно програми TighVNC viewer з логом підключення до мережі

Як видно з скріншота 3.3, роутер видав IP-адресу, 192.168.0.118. Тепер можна отримати віддалений доступ до Pi за допомогою клієнта безпечної оболонки (SSH), щоб підключитися до нього, за допомогою комп'ютера.

Тепер необхідно встановити програму PuTTY, для підключення до плати за протоколом SSH, рисунок 3.4.

Після цього ввести IP-адресу Raspberry Pi у полі Ім'я хоста та натиснути кнопку Відкрити. Після цього відбудеться підключення до Pi у вікні віддаленого терміналу.

Після того було введено пароль для входу в систему, за допомогою Putty можна отримати повний доступ до файлової системи та командного рядку Raspberry Pi.

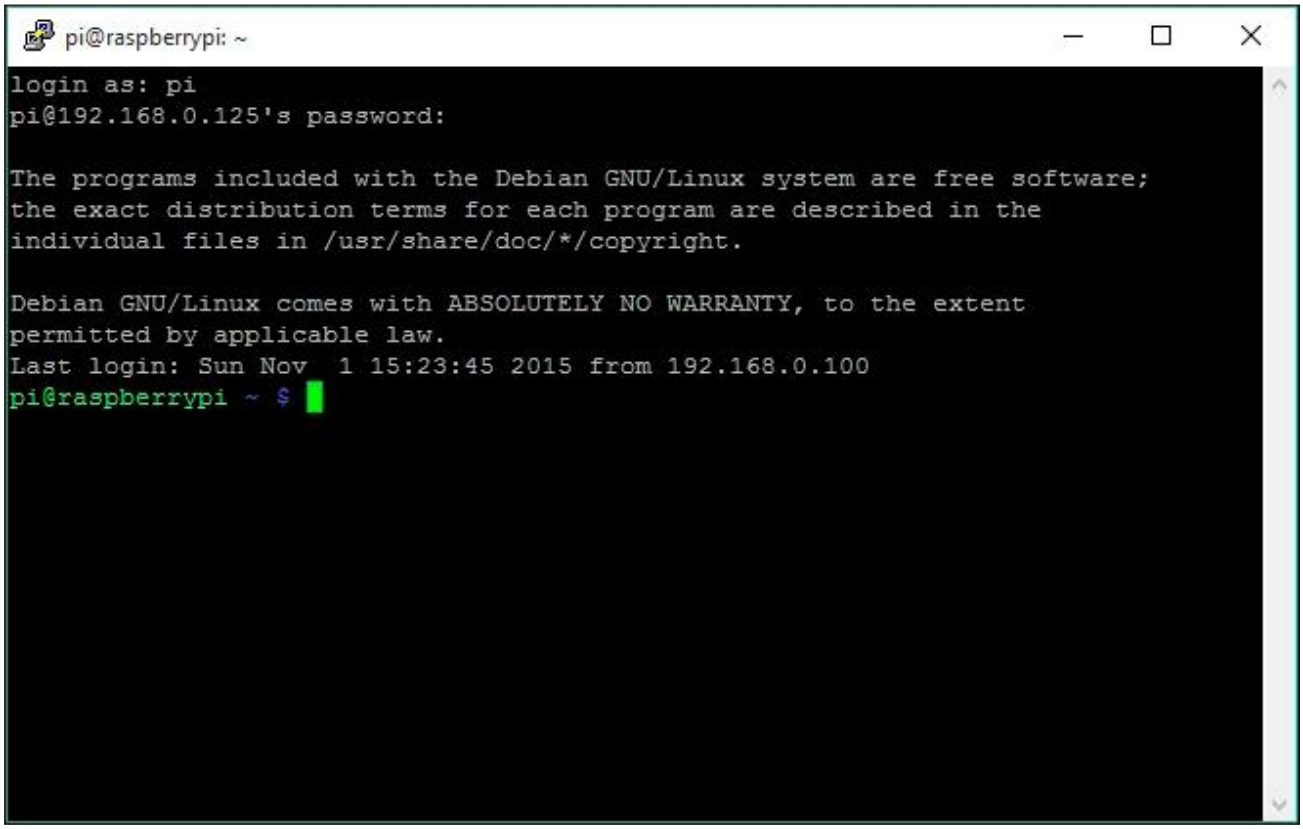


Рисунок 3.4 – Консоль Raspberry Pi у PuTTY

Щоб використати командний рядок для запуску віддаленої оболонки Raspberry Pi - наприклад, з іншої системи Linux – необхідно вписати наступну команду у вікні терміналу:

ssh

pi@192.168.0.125

Після цього буде запропоновано ввести пароль Pi та перейти в сеанс оболонки.

3.3 Підключення пристроїв до Raspberry Pi.

3.3.1 Підключення та тестування магнітного контактного давача

Тепер, коли було налаштовано Raspberry Pi, потрібно запрограмувати його так, щоб Raspberry Pi міг виявляти давачі, які підключені до нього як частину системи безпеки.

Спочатку підключаються перемикачі до нашої системи у вигляді магнітних давачів - найпоширенішого компонента, що використовується в домашніх системах безпеки для виявлення вторгнень через двері та вікна.

Магнітний давач встановлюється на дверній рамі (щоб він міг підключитися до дротів ланцюга сигналізації), а відповідний магніт прикріплюється до дверей, досить близько до краю, щоб контакти давача з'єднувалися (або розривалися залежно від типу), коли магніт знаходиться прямо навпроти нього.

На схемі, що зображена на рисунку 3.5, було підключено резистори, що підтягують землю, але їх також можна підключити безпосередньо на платі.

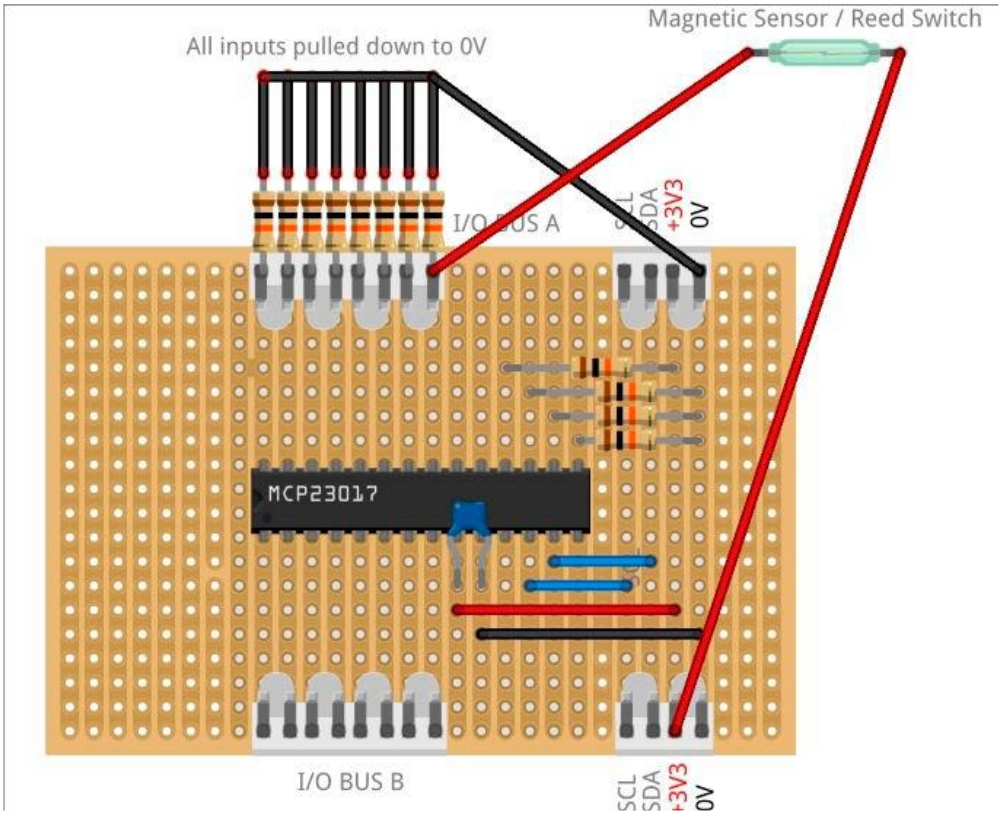


Рисунок 3.5 – Схема підключення манітного давача

Щоб перевірити вхідне значення порту, необхідно ввести команду команду i2cget:

$ sudo i2cget -y 1 0x20 0x12

Вона має повернути 0x00, що означає, що всі входи вимкнені (двійковий %000000).

Але коли магніт переміщується, що супроводжує, поруч із перемикачем давача (наприклад, якщо двері закриті), перемикач замикається, підтягуючи вхідний сигнал високого рівня до лінії +3,3 В.

Якщо можна прочитати у значенні вхідного сигналу порту, виконавши ту саму команду, тепер вона повертає 0x01, що вказує на те що перший біт є високий (двійковий %00000001).

Тепер, коли все готово і магнітний давач визначає чи зачинені двері, з'явилась можливість контролювати цей давач за допомогою простого сценарію Bash який використовує команди інструменту I2C, встановленого нами раніше.

Лістинг коду для poll-magnetic-switch.sh виглядає так:

```
sudo i2cset –y 1 0x20 0x00 0xFF
while true doS  WITCH=$(sudo i2cget –y 1 0x20 0x12)
if [ $SWITCH == "0x01" ]
i2cget -y 1 0x20 0x12
then  echo "The door is closed!"
sleep 1
else echo "The door is open!"
fi
```

Після запуску сценарію і натискання на кнопку, в консолі з'являться повідомлення "Двері відчинені!", що прокручується на екрані консолі до тих пір, поки кнопка натиснення.

Пізніше, коли буде підключено усі сенсори необхідно бути дописати скрипт щоб він відрізняв який саме давач спрацював, ця Bash функція, буде аналізувати шістнадцяткове значення що повертається після виконання команди.

3.3.2 Підключення пасивного інфрачервоного давача руху

Тепер необхідно додати в систему пасивний інфрачервоний давач руху (PIR). У принципі, всі ці давачі працюють однаково, виявляючи присутність тепла тіла у певному діапазоні; тому вони зазвичай використовуються для увімкнення систем сигналізації, коли хтось (або щось, наприклад, домашня кішка) входить до кімнати. Рисунок 3.6 зображення промислового пристрою для виявлення руху.

Рисунок 3.6 – PIR-давач руху GardScan QX-PIR

Пристрої PIR-давачів бувають різних форматів, включаючи різні матеріали в мікросхемах давачів та лінзу перед оглядовим вікном давача, які можуть впливати на дальність, поле зору і чутливість пристрою.

Розташування пристрою також залежить від того, яку область необхідно захистити. PIR-давачі зазвичай мають фіксоване поле зору (наприклад, 90 або 110 градусів), але їхній радіус дії варіюється в залежності від кута, під яким вони спрямовані вниз, і висоти, на якій вони розташовані.

Наявні у продажу системи сигналізації підключаються за допомогою 4 або 6-жильного кабелю сигналізації. На наступній схемі (рисунок 3.7) показані з'єднання проводів для давача GardScan PIR, це типова схема для більшості готових охоронних систем.

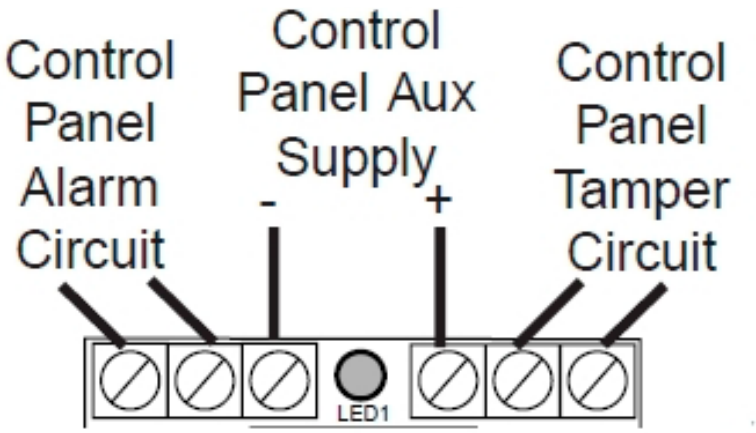


Рисунок 3.7 – Типові з'єднання для сенсорних пристроїв охоронної системи

Цей конкретний пристрій має нормально замкнутий вихід, що означає, що ланцюг сигналізації буде розірваний, коли давач спрацьовує.

Тепер є можливість додати цей сенсорний пристрій у ланцюг сигналізації. На рисунку 3.8 показано схему всіх давачів, підключених до однієї зони

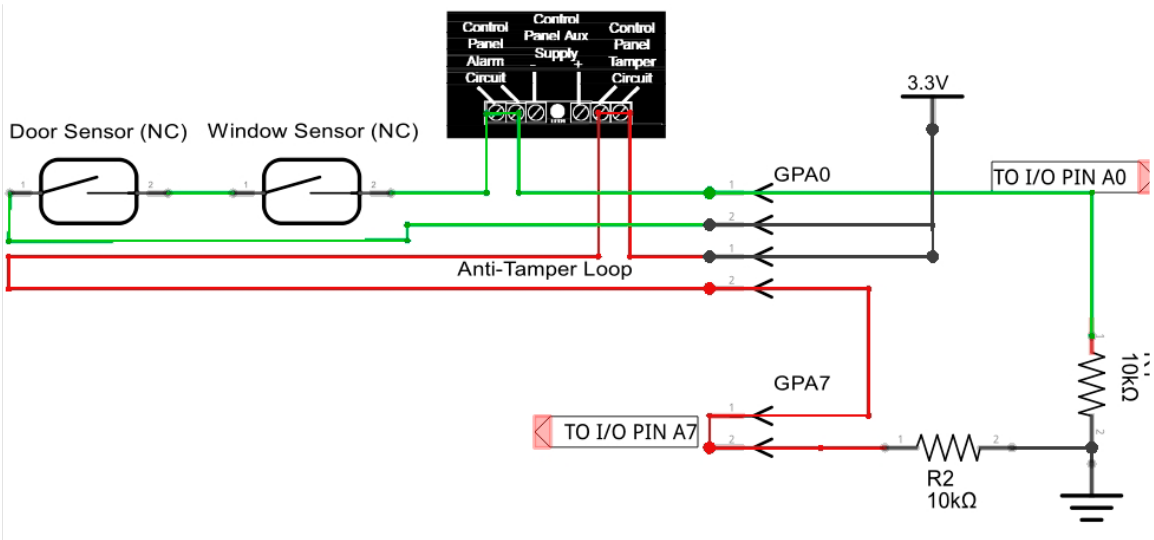


Рисунок 3.8 – Схема для зон з усіма давачами та темперним шлейфом на одній платі

Це краща конфігурація для наших сенсорних пристроїв, оскільки це означає, що вони можуть бути послідовно підключені в кожній з зон.

До цих пір використовувалось напруга +3,3 для живлення давача перемикачі і ланцюг сигналізації. Насправді це не дуже хороша ідея, і ми робили це тільки для зручності, щоб перевірити наші входи GPIO.

В кінцевій системі, необхідно буде використовувати живлення 12 В для проходження через давач і схеми захисту від несанкціонованого доступу. Це пов'язано з тим, що більш висока напруга краще проходить через систему і менш сприйнятлива до шумів, які можуть завадити спрацьовуванню або спричинити помилкове спрацьовування. Це також робить його сумісність із наявними у продажу системами та аксесуарами.

Змусити схеми зон використовувати 12 В замість 3,3 В так просто, як змінити джерела живлення, і насправді всі давачі, які використовували досі, можуть працювати з 12В, що проходять через їх перемикачі.

Однак, якщо подати напругу 12 В на входи GPIO Raspberry Pi або на розширювач портів, можна спалити схему. Тому нам необхідно додати додаткову схему, яка дозволить використовувати 12-вольтові сигнальні ланцюги, а також захистити входи плати керування.

3.3.3 Підключення та тестування мікроконтролеру захисту плати від перепадів напруги

Ефективним способом захисту входів зон від сигнальних входів 12 В є використання невеликого недорогого пристрою, що називається опт ізолятор або оптопара. Він ізолює ланцюг сигналізації від цифрових входів плати керування за допомогою світла.

Усередині опт ізолятор знаходиться інфрачервоний світлодіод, що передає світло на фото транзистор, коли через нього проходить струм. і цим включає його. Ланцюги електрично ізольовані, оскільки вони керуються лише світлом. У проекті було використано плату 4N25, рисунок 3.9.

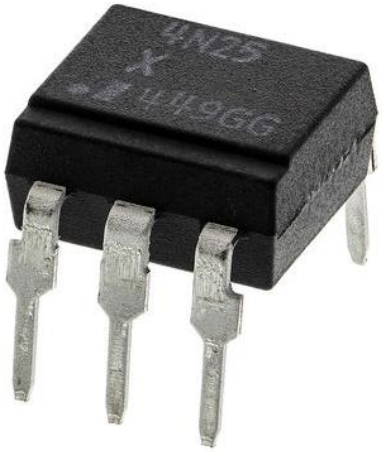


Рисунок 3.9 – Мікросхема оптопари 4N25

Тепер, коли визначена мікросхема та план, як ми з'єднати ланцюг сигналізації 12 В з входами на контрольній панелі, було побу весь ланцюг живлення, який буде використовуватись для кожної з зон системи безпеки.

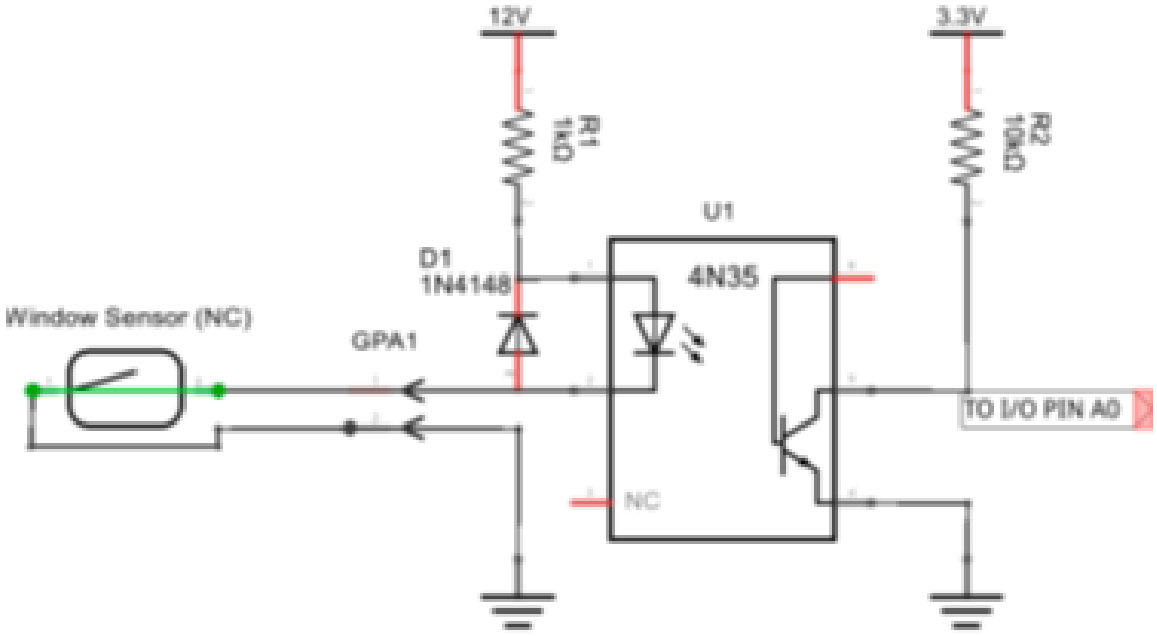


Рисунок 3.10 – Універсальна схема живлення зон

Живлення 12 проходить через світлодіод опт ізолятора, а струм обмежується резистором 1 КОм. Діод 1N4148 у зворотному напрямку служить для захисту

оптопари від напруги зворотної полярності. Резистор 1 КОм розраховується з того факту, що у схемі є джерело живлення 12В і пряме падіння напруги (Vf) 1,2 на світлодіоді при струмі (If) близько 10 мА.

Поки ланцюг сигналізації замкнутий, струм тече і світлодіод горить – цей транзистор залишається увімкненим, а на вході порту GPIO підтримується низький рівень. Якщо ланцюг сигналізації розривається, світлодіод оптопари вимикається, а це своєю чергою. вимикає транзистор. Після цього вхід GPIO підтягується до високого рівня резистором 10-ком резистора.

Іншою перевагою цієї схеми є те, що вона повинна бути інверсивною - тобто якщо оптопара вийде з ладу з якоїсь причини, тривожний вхід на GPIO повинен бути підтягнутий до високого рівня, що приведе до його спрацьовування, а не до відмови.

3.3.4 Підключення та тестування радіомаяка 433 МГц

Коли модуль приймача увімкне реле, це замкне ланцюг 12 В через світлодіод оптопари, увімкнувши його. Це змусить транзистор підтягнути вихід GPIO до землі, подавши на нього низький вхідний сигнал.

Цей тип схеми може бути використаний для будь-якого парного приймача бездротових охоронних пристроїв.

Схема сполучення модуля приймача з входом GPIO Коли модуль приймача увімкне реле, це завершить ланцюг 12 В через світлодіод оптопари, увімкнувши його. Це змусить транзистор підтягне виведення GPIO до землі, подавши на нього низький вхідний сигнал.

У будь-якій системі корисно мати можливість реєструвати дані, коли щось відбувається. Логуванння можна зробити таким чином щоб, записувати у файл журнал дані щоразу, коли детектор у зоні спрацьовує. Таким чином, можна вести журнал, коли хтось входить до кімнати робити про це помітку в файлі, яку можна переглянути пізніше, навіть якщо навіть якщо система не поставлена на охорону.

На рисунку 3.11 показано підключення приймача через GPIO

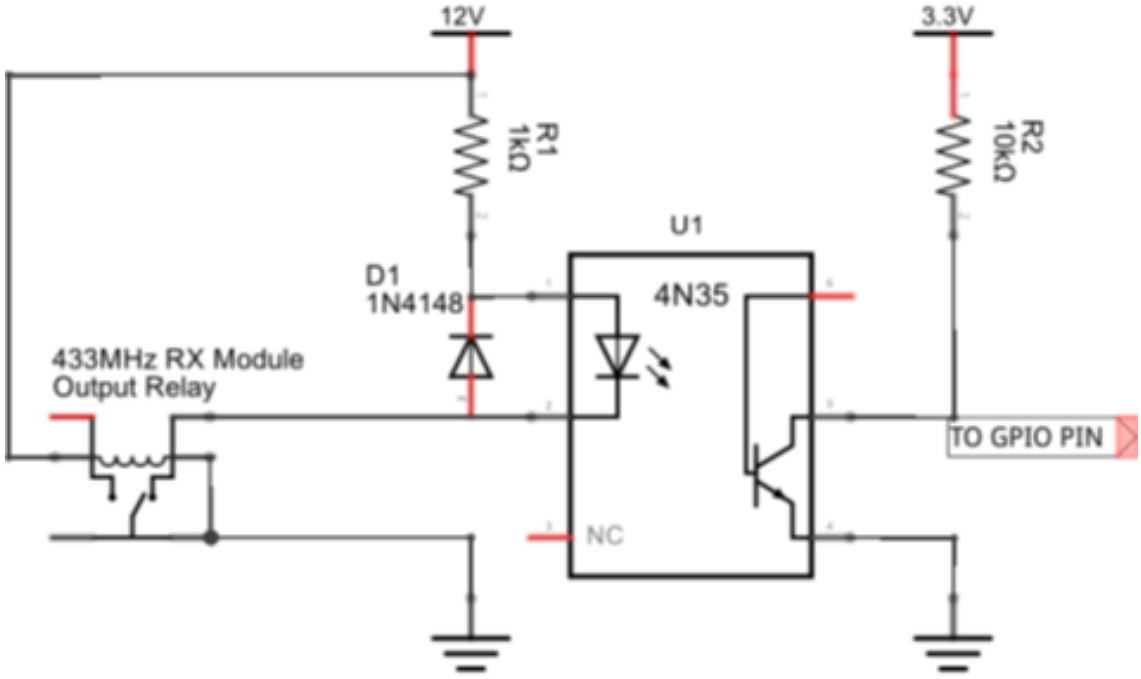


Рисунок 3.11 – Схема сполучення модуля приймача з входом GPIO

Нижче наведений лістинг функці, який записує данні коли у зонах, підключених до входів GPIO спрацьовує давач:

sudo i2cset –y 1 0x20 0x00 0xFF

CURR_STATE="0x00" LAST_STATE="0x00"

LOG_FILE="/etc/pi-alarm/zones.log"

while true do CURR_STATE=$(sudo i2cget –y 1 0x20 0x12)

if ["$CURR_STATE" != "$LAST_STATE"]

then TIMESTAMP=`date "+%Y-%m-%d %H:%M:%S"`

i2cset –y 1 0x20 0x00 0xFF

Fi $LAST_STATE = $CURR_STATE

sleep 1 done

Попередній приклад показує, що фактично можна виписати зону або зони, які змінюються, розшифрувавши шістнадцяткове значення, що повертається командою i2cget у складових цих зон.

Таким чином буде вестись логування при передачі даних по радіоканалу 433 МГц

3.3.5 Підключення камер до системи безпеки

Як згадувалося раніше, модуль підключається безпосередньо до плати Raspberry Pi через спеціальні інтерфейси камери, як показано на рисунку 3.12.

При підключенні камери контактна сторона стрічкового кабелю спрямована у бік HDMI, а синя сторона кабелю - до розетки. роз'єм.

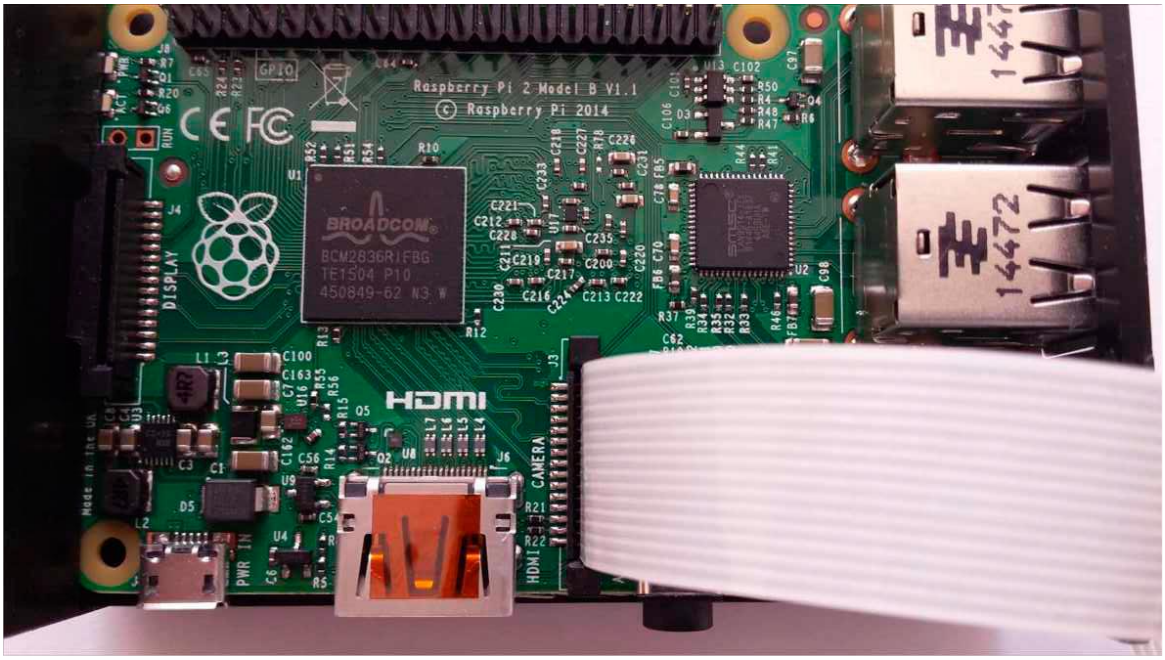


Рисунок 3.12 – Зображення конектора камери підключеного до роз’єма плати Raspberry Pi

Перш ніж використовувати модуль камери, необхідно включити підтримку камери на Raspberry Pi.

Для цього необхідно використати інструмент raspi-config, як це було виконано раніше з шиною I2C шиною.

Спочатку було приєднано камеру та підключено до плати за допогою SSH. Після чого було введено команду sudo raspi-config.

Після чого було обрано пункт 5 у списку, що має назву Enable Camera, так як показано на рисунку 3.13. Та у меню що з’явиться після цього обрати пункт Enable, рисунок 3.14.

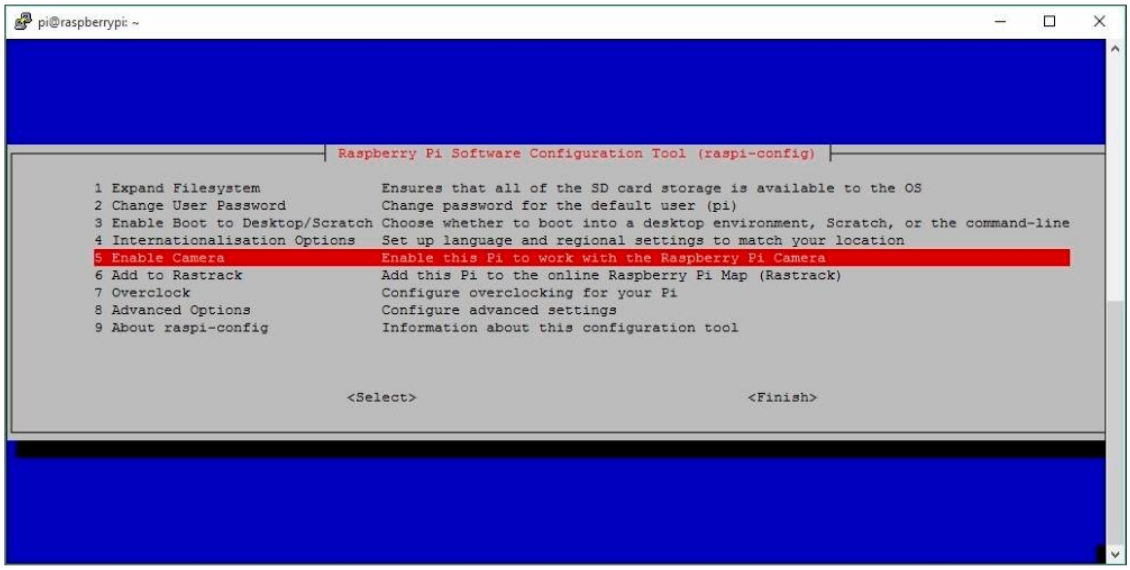


Рисунок 3.13 – Меню raspi-config

Рисунок 3.14 – Екран підтвердження увімкнення камери

Отже, після цього, у системі є метод захоплення нерухомих зображень та відео, які можна використовувати у системі безпеки.

Для того, щоб це працювало постійно, було написано сценарій для постійної відеозйомки, але в цьому випадку це швидко заповнить нашу карту пам'яті і не буде особливо ефективним. Отже, було об’єднано систему камер із детекторами руху, який було підлючено раніше. У попередньому розділі було ствоено зону тривоги, де було кілька давачів і детектор руху, підключений до системи на вхід GPA0.

Далі наведений сценарій, який зніматиме відеокліп щоразу, коли спрацьовує детектор руху спрацьовує:

while true do

GPA=$(sudo i2cget –y 1 0x20 0x12)

if [$GPA == "0x01"] then sleep 0.5

else sDate='date +%d%m%y'

sTime='date +%T'

echo "Zone 1 Activate at $sDate $sTime"

raspivid –o $sDate$sTime.h264 –t 20000

MP4Box -fps 30 -add $sDate$sTime.h264 $sDate$sTime.mp4 fi

done

3.3.6 Підключення та тестування e-mail сервіса

Зберігати зображення на Raspberry Pi не дуже ефективно, набагато зручніше - надсилалися зображення відразу після зйомки, щоб користувач міг переглянути їх на своєму смартфоні. Простий, швидкий і надійний спосіб зробити це - просто надіслати їх електронною поштою.

Тому в систему домашньої безпеки було додано функцію надсилання електронною поштою, щоб знімки прикріплювалися до повідомлення і відразу ж відправлялися на адресу електронної пошти, яку користувач зможе переглянути зі свого смартфона.

Потім зображення можна видалити з Raspberry Pi, щоб не засмічувати місце на SD-карті цими досить великими файлами.

Для цієї задачі було використано безкоштовний модуль з відкритим кодом, який називається ssmtp. Для його встановлення необхідно було ввести наступну послідовність команд:

sudo apt-get update

sudo apt-get install ssmtp

Після цього було налаштовано клієнта для надсилання листів через обліковий запис електронної пошти обліковий запис. У наступному конфігураційному файлі було використано обліковий запис Gmail, як провайдер електронних листів.

Після його встановлення можна використовувати команду mail для зручнішого надсилання електронних листів.

Для тестування було використано команду для відправки тестового листа через обліковий запис (G)mail, який було налаштовано раніше, використовуючи наступну команду, щоб переконатися, що ваші налаштування працюють:

echo "Test Email" | mail-s "Test Pi-Mail" me@mydomain.com

3.4 Побудова веб-інтерфейсу

Після того як було зібрано усі апаратні елементи разом, щоб створити повноцінну систему домашньої безпеки, що включає контактні вимикачі для дверей та вікон, давачі руху та камери, щоб зробити щасливі знімки потенційних зловмисників, системи безпеки вимагають наявності контрольної панелі, яка дозволяє ставити та знімати систему з охорони та контролювати стан зон у системі.

Користувач також може поставити на охорону лише певні зони або автоматично ставити та знімати систему з охорони у певний час доби. Необхідне для цього обладнання, таке як перемикачі, світлодіоди та РК-дисплеї може бути досить дорогим і забирати багато часу, крім того, воно може зробити систему менш конфігурованою та гнучкою. Тому в конкретній системі було створено панель керування на основі веб-технологій, доступ до якої можна отримати за допомогою браузера мобільного телефону. Це також означає, що користувач може керувати системою віддалено.

3.4.1 Встановлення веб-серверу

Існує кілька доступних веб-серверів, які рекомендується встановлювати на Raspberry Pi, і всі вони підійдуть для системи безпеки. Але було обрано веб-сервер

lighttpd, тому що він простий у використанні та легкий. lighttpd часто називається "Lighty". Крім самого веб-сервера, також було встановлено підтримку PHP, що дозволяє писати динамічні веб-сторінки для взаємодії із системою Linux системою.

За допомогою програми PuTTY було виконано наступні дії для того щоб встановити веб-сервер lighttpd:

- оновлено встановник веб-пакетів: sudo apt-get update;
- встановлено пакет lighttpd: sudo apt-get install lighttpd;
- встановлено пакет підтримки PHP5: sudo apt-get install php5-cgi;
- перезапущено веб-сервер: sudo /etc/init.d/lighttpd.

Тепер на накопичувачі Raspberry Pi встановлений веб-сервер PHP. За промовчанням файли веб-вмісту встановлюються в папку /var/www, а Lighty встановлює тестову сторінку в цьому місці, до якої можна отримати доступ з браузера, просто ввівши IP-адресу Raspberry Pi, як показано на рисунку 3.15.

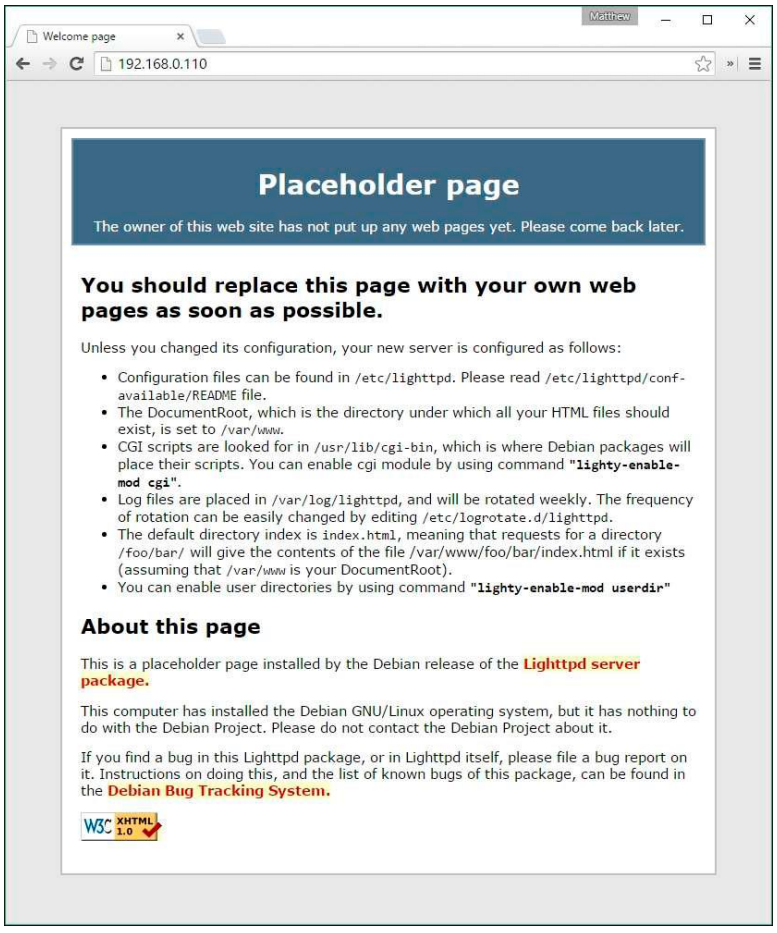


Рисунок 3.15 – Заголовна сторінка веб-сервера lighttpd

Тепер у браузері можна ввести IP-адресу Raspberry Pi, а потім /phpinfo.php, наприклад, http://192.168.0.110/phpinfo.php, і перед користувачем відкриється сторінка показана на рисунку 3.16, де написані усі конфігурації веб-сервера та Raspberry Pi.4

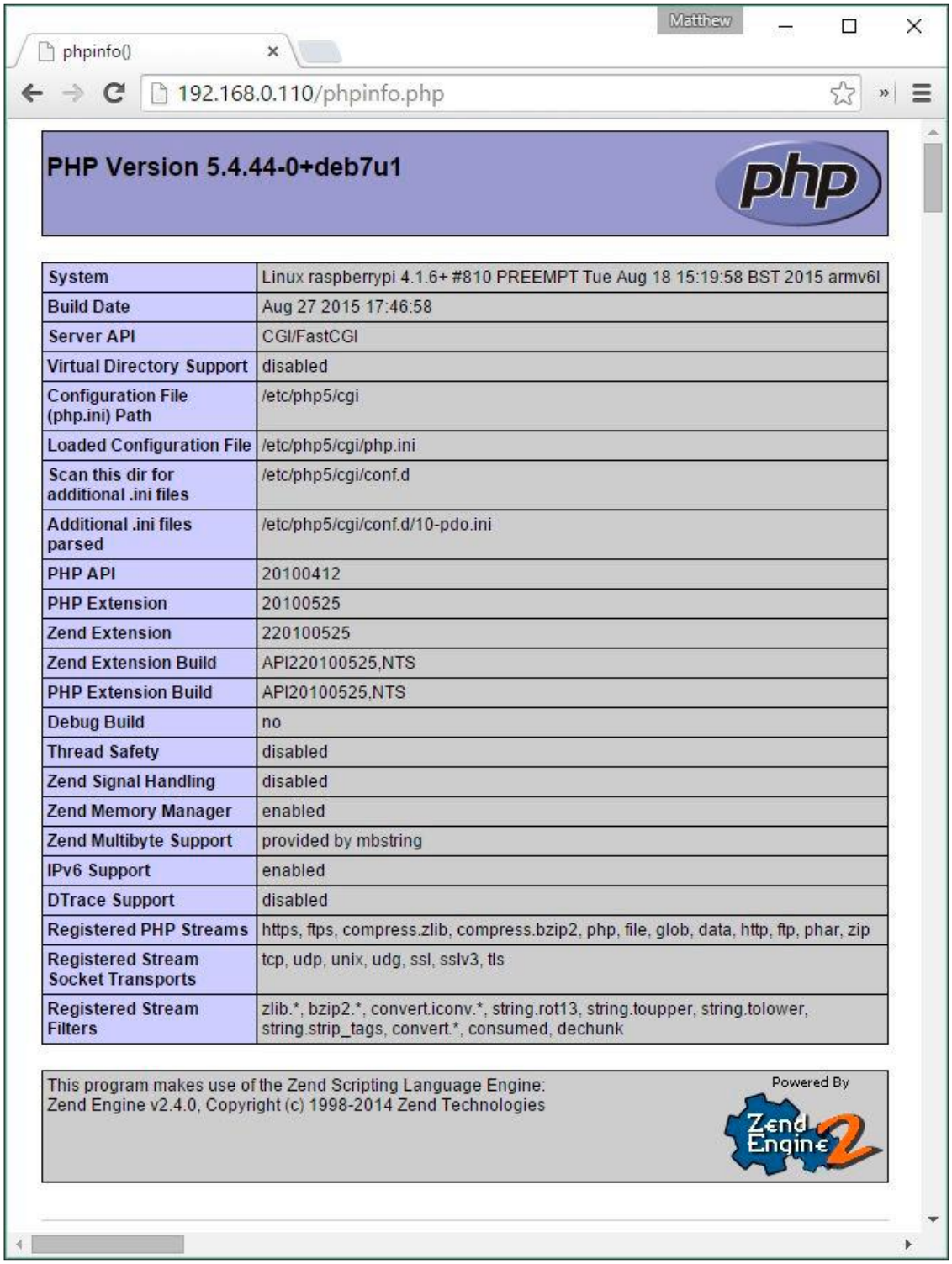


Рисунок 3.16 –Веб-сторінка з інформацією про веб-сервер

3.4.2 Контроль давачів

Для того щоб знати, які елементи управління користувач буде мати на панелі управління сигналізацією, необхідно скласти схему системи із зазначенням кількості входів зон і входів та виходів управління входів зон і входів та виходів управління.

Усі підключення портів було представлено у вигляді таблиці 3.1. У проекті було створено 8-зонну систему для сигнальних входів, використовуючи порт А на платі розширення вводу/виводу, використовуючи власні виходи GPIO для таких речей, як кнопки та сигнальні виходи.

Таблиця 3.1 – Підключення портів до розширювача та плати Raspberry Pi

Порт	I/O пін	Назва/Призначення
Розширювач А	0 (A0)	Зона 1 Вхід (Вхідний/Вихідний канал)
	1 (A1)	Зона 2 Вхід
	2 (A2)	Зона 3 Вхід
	3 (A3)	Зона 4 Вхід
	4 (A4)	Зона 5 Вхід
	5 (A5)	Зона 6 Вхід
	6 (A6)	Зона 7 Вхід
	7 (A7)	Зона 8 Вхід захищений від несанкціонованого доступу
Розширювач Б	0 (B0)	
	1 (B1)	
	2 (B2)	
	3 (B3)	
	4 (B4)	
	5 (B5)	
	6 (B 6)	

Кінець таблиці 3.1 – Підключення портів до розширювача та плати Raspberry Pi

	7 (B 7)	
Raspberry Pi GPIO	0 (GP0)	Перемикач постановки на охорону/зняття з охорони (вхід)
	1 (GP1)	
	2 (GP2)	
	3 (GP3)	
	4 (GP4)	Світлодіод постановки на охорону (вихід)
	5 (GP5)	Зумер (звуковий індикатор) постановки на охорону/зняття з охорони (вихід)
	6 (GP6)	Світлодіод тривоги (вихід)
	7 (GP7)	Сигнал тривоги (вихід)

Терміни "постановка на охорону" та "зняття з охорони" означають включення (постановка на охорону) та вимкнення (зняття з охорони) системи охоронної сигналізації.

Зона 1 системи буде пов'язана з частиною постановки на охорону та зняття з охорони. так як вона буде підключена до давачів на двері, через які людина входить або це буде спеціальна зона для входу чи виходу. Коли користувач встановлює сигналізацію, потрібно небагато часу, щоб вийти з дому. На сайті система дізнається, що користувач покинув будинок по моніторингу зони виходу.

Користувач буде ставити та знімати систему з охорони через веб-панель управління або за допомогою перемикача на вході GP0.

3.4.3 Головний конфігураційний файл

Система використовує головний конфігураційний файл, який розповість їй, як все налаштовано та підключено. Цей файл конфігурації використовуватиметься як веб-панель керування, так і основний скрипт керування сигналізацією, щоб дві

підсистеми могли "розмовляти" один з одним. Файл налаштувань зберігається в тому ж місці, де було створювати наші сценарії керування в, тобто в папці. /etc/pi-alarm.

Файл конфігурацій має зміст наведений далі:

ZONE_LABEL_1="Zone 1 - Entry/Exit"

ZONE_ENABLE_1=1 ZONE_ENABLE_2=1

SYSTEM_ARMED=0 ZONE_STATUS_1=0

3.4.4 Створення веб-сторінки

Перше, що було зроблено, це створено HTML-файл, який було використано для тестування, перш ніж помістити HTML у файл PHP, щоб змусити його взаємодіяти з системою. Це полегшує налаштування того, як він буде виглядати заздалегідь, не заважаючи PHP-скриптам.

Хоча розмітка містить посилання на файл CSS, цей файл ще не створено, тому на рисунку 3.17 показано головну сторінку для керування, без стилів.

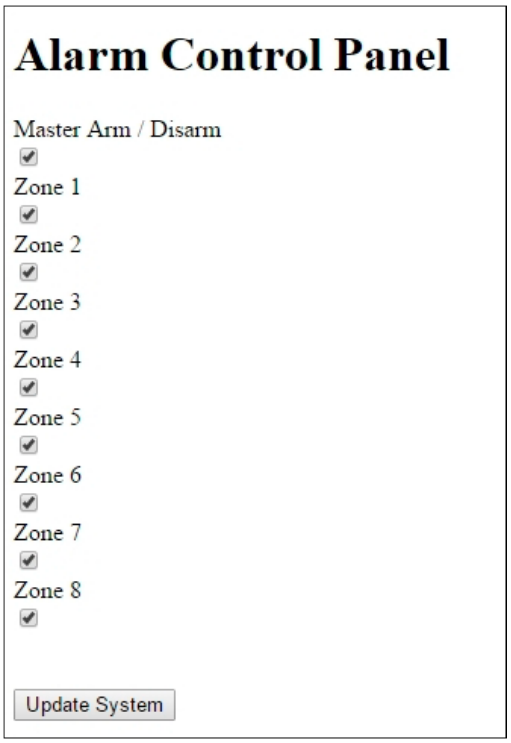


Рисунок 3.17 – Сторінка керування зонами без стилів

Розмітка CSS3 розроблена спеціально для панелі управління, і дозволяє їй виглядати досить красиво, а також робить її зручною для використання на сенсорних мобільних пристроїв. Вигляд сторінки з застосованими стилями наведено на рисунку 3.18.

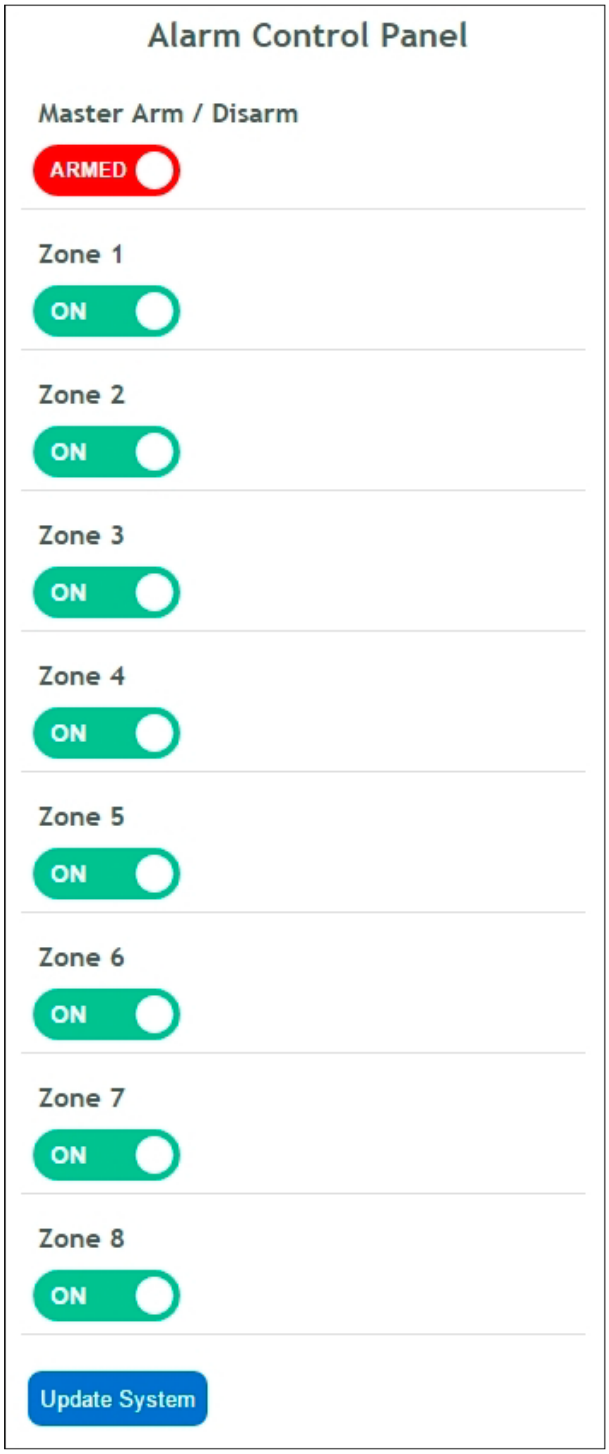


Рисунок 3.18 – Сторінка керування зонами з використанням стилів

Після визначення коду макета для сторінки панелі керування, було вставлено його в PHP-сторінку, щоб PHP-скрипт на веб-сервері динамічно змінював його залежно від стану системи безпеки. PHP-скрипт допоможе виконати такі основні функції, як оновлення конфігураційного файлу із зазначенням положення перемикачів увімкнення/вимкнення для зон, постановка на охорону та зняття з охорони системи та повідомлення, яка зона спрацювала при виявленні вторгнення.

На рисунку 3.19 показано написа зверху сторінки що сигналізує про успішне збереження файла конфігурацій.

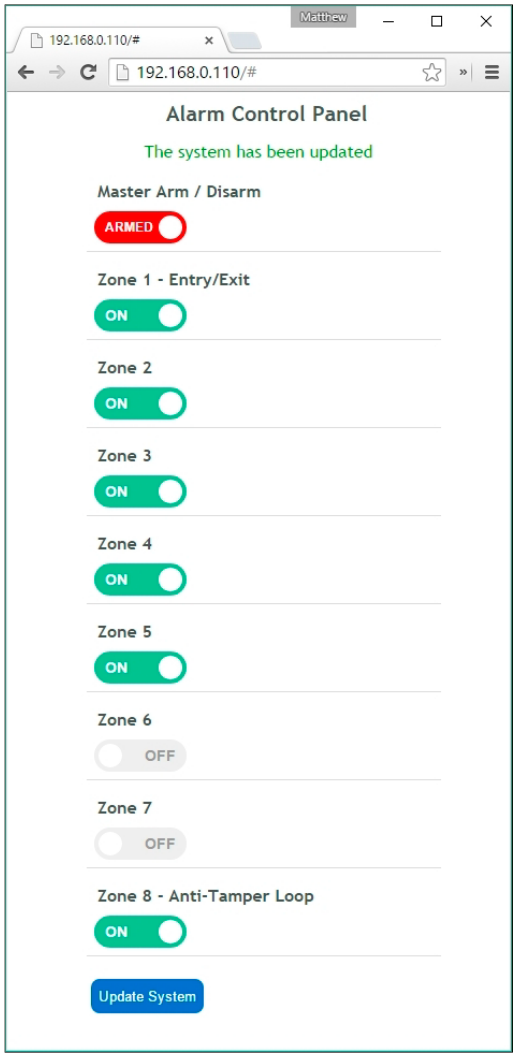


Рисунок 3.19 – Сторінка налаштування зон після додавання у проект PHP коду

index.php налаштований як сторінка за замовчуванням у конфігурації Lighty, тому вам потрібно додавати його до кінця URL, досить просто IP-адреси. Змінивши

положення перемикачів, а потім натиснувши кнопку Обновити систему, користувач повинний виявити, що значення налаштувань оновлюються відповідним чином у файлі alarm.cfg.

Останнє, що потрібно зробити, це створити невеликий сценарій Bash, який виконуватиме завдання оновлення налаштувань у файлі alarm.cfg. Причина, через яку потрібно це зробити, полягає в тому, що необхідно використовувати команду Linux sed для оновлення файлу. Те, як система викликає команду sed, означає, що їй потрібно створити тимчасовий файл. Якщо не зробити невелике налаштування веб-серверу через його контекст розташування файлів, це не спрацює. Тому простіше створити скрипт-заглушку Bash, який викликається PHP-скриптом. Таким чином, середовище Bash працює з контекстом тимчасового файлу.

3.4.5 Налаштування веб-адміна

Webmin - це досить хороший веб-інтерфейс для адміністрування Unix/Linux систем, що добре зарекомендував себе.

Є кілька способів встановити Webmin: або вручну завантаживши та розпакувавши його, або оновивши вихідні файли репозиторію, щоб можна було використовувати aptget.

Необхідо ввести декілька команд для того щоб встановити веб-адмін:

cd ~

sudo wget http://www.webmin.com/jcameron-key.asc

sudo apt-key add jcameron-key.asc

sudo apt-get update

sudo apt-get install webmin

Після цього можна буде побачити повідомлення показане на рисунку 3.20, у вікні командного вікна.

Webmin, за замовчуванням, працює на порту 10000 та використовує безпечний протокол HTTPS, таким чином, щоб отримати доступ до нього, потрібно ввести в браузері наступний URL: hpps://192.168.0.99:10000.

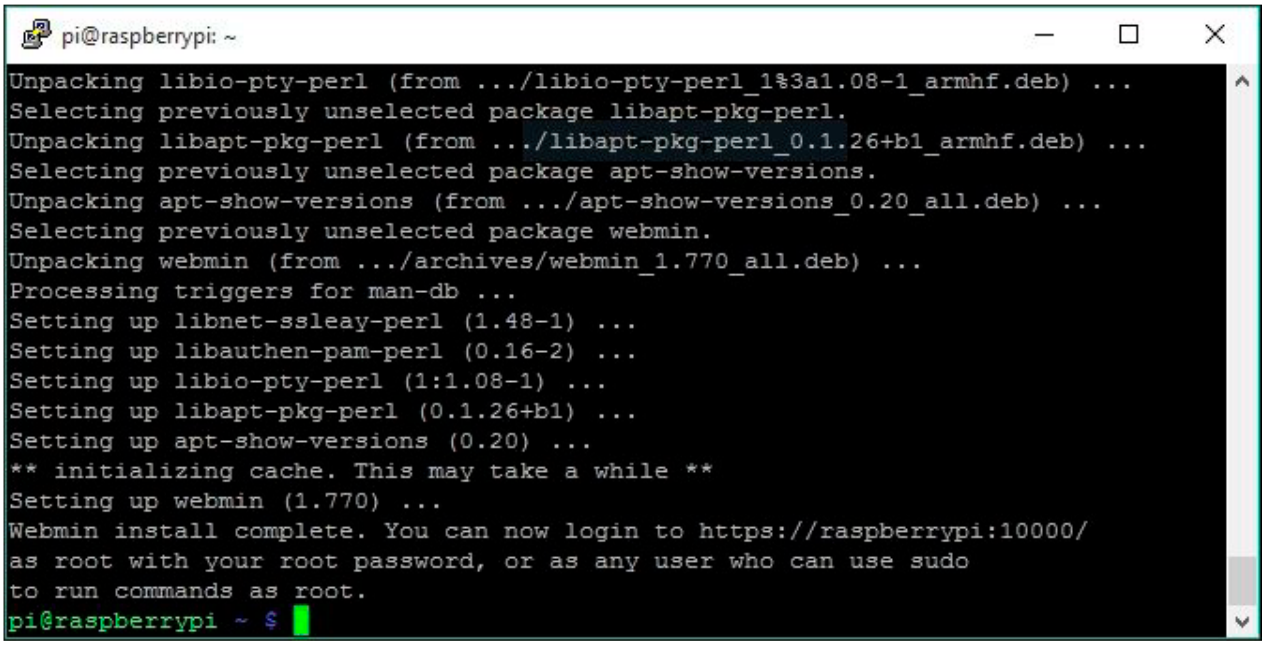


Рисунок 3.20 – Вікно командного рядка після успішного встановлення веб-адміна

Після цього у вікні авторизації, рисунок 3.21 необхідно ввести логін і пароль, такі самі що і коли вводиться команда sudo.

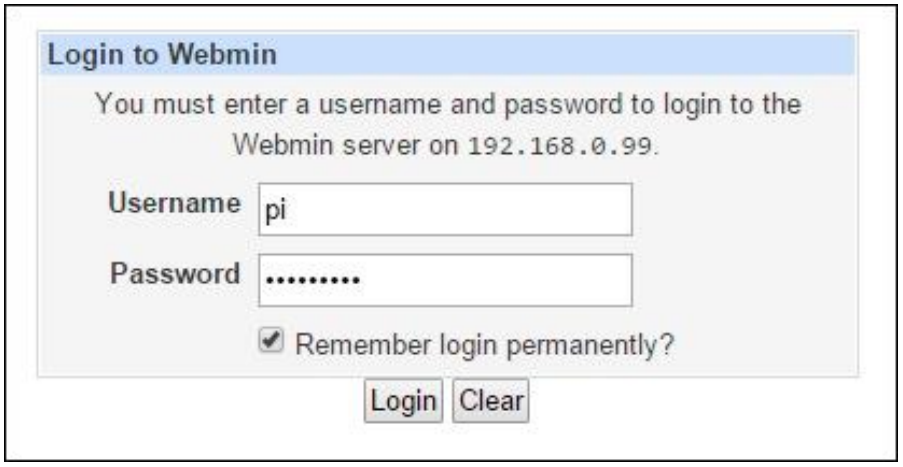


Рисунок 3.21 – Вікно авторизації веб-адміна

Після входу в систему, перед користувачем відкриється головна сторінка з інформацією про систему. Вигляд системної інформації Webmin Webmin ,рисунок 3.22, поставляється з великою кількістю модулів і не всі з них встановлені, тому можна перейти у розділ панелі "Невикористовувані модулі", щоб дізнатися, чи є щось, що ще можна додати до Webmin.

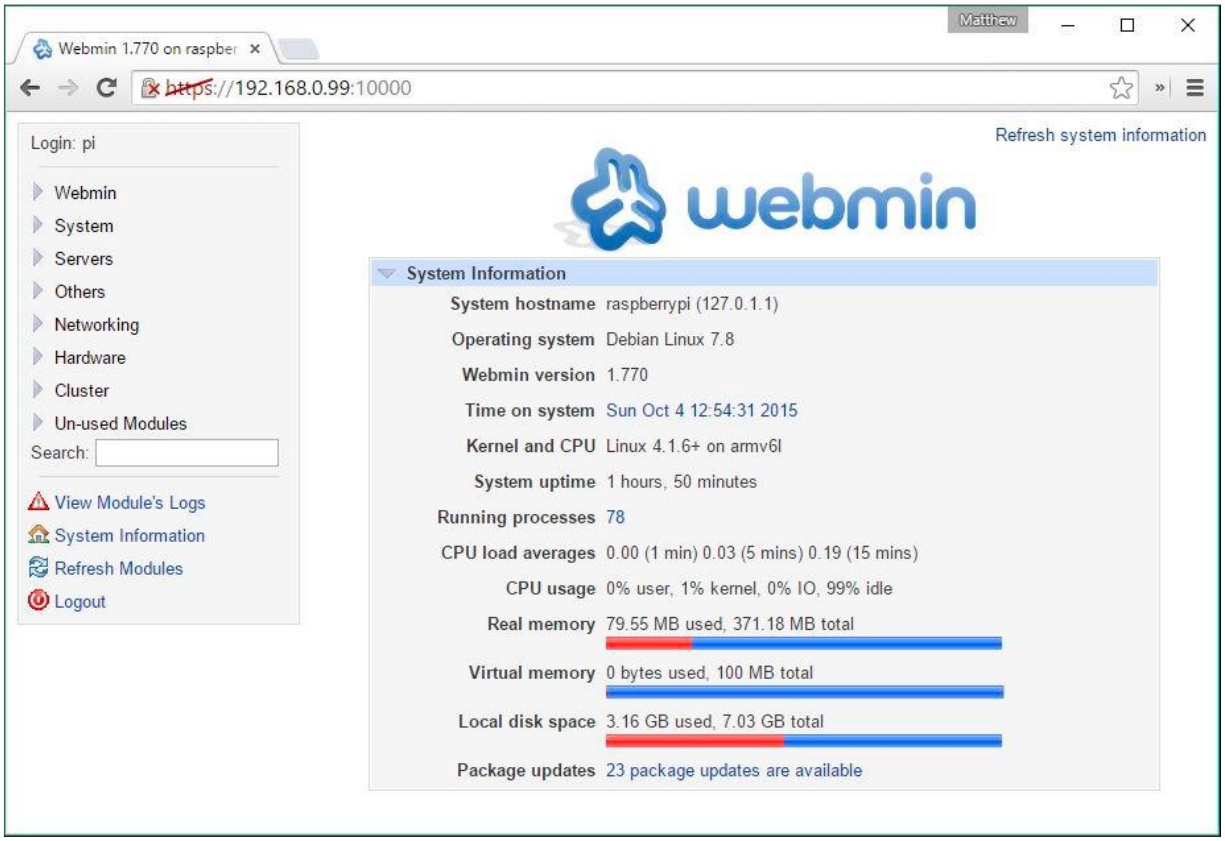


Рисунок 3.22 – Зображення веб-інтерфейса Webadmin

Так само, як раніше був налаштований віддалений доступ для панелі керування сигналізацією можна зробити це з Webadmin - просто потрібно налаштувати переадресацію портів на маршрутизаторі для порту 10000. Після цього користувач отримає доступ до Webmin з будь-якого місця, використовуючи https://<Ip-address>:10000.

3.5 Опис сценарію керування програмно-технічнім засобом охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi

Блок-схема, зображена на рисунку 3.23 дозволяє представити, як має працювати система, та різні логічні рішення, які виконує скрипт.

Система буде очікувати, доки не буде поставлена на охорону або апаратним ключем, або апаратною клавішею, або програмним перемикачем веб-панелі.

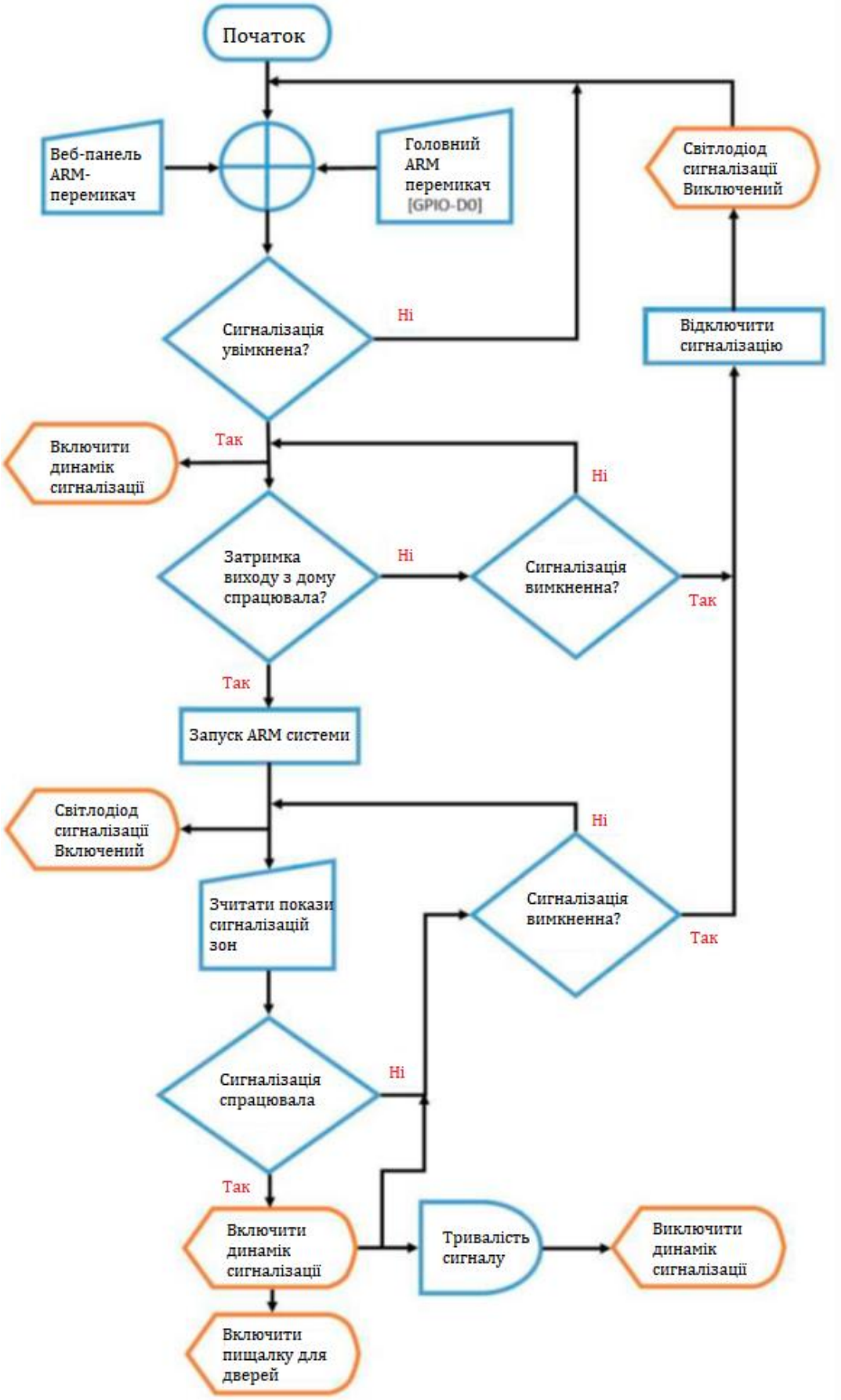


Рисунок 3.23 – Блок-схема роботи охоронної системи

Коли система вперше ставиться на охорону, вона подає звуковий сигнал на вихід протягом попередньо визначеного перед постановкою системи на охорону періоду. Це дає можливість залишити територію або зняти систему з охорони, перш ніж вона почне контролювати входи.

Заздалегідь було спланована карту давачів, що зображена на рисунку 3.24, кожна зона відповідає певній кімнаті, так користувач може точно дізнатись в які зоні спрацював давач.

Рисунок 3.24 – План будинку з поміченими зонами

Після встановлення системи на охорону включиться світлодіод "Охорона", і система чекатиме, чи не з'явиться якийсь із входів. система чекатиме, чи не спрацює якийсь із входів зони тривоги. Вона також чекатиме, чи не буде знято сигналізацію з охорони після вашого повернення до будинку. у дім. За бажанням,

можна встановити таймер входу в зону входу. для затримки перед увімкненням сигналізації.

Якщо сигналізація спрацює, то буде увімкнено основний дзвінок тривоги, а також звуковий індикатор на виході. Головний дзвінок повинен звучати лише звучати лише деякий час, залежно від екологічних обмежень у вашому районі. Залежно від екологічних обмежень у вашому районі він буде вимкнений через заздалегідь визначений період часу. але внутрішній зумер залишиться увімкненим.

При спрацьовуванні система чекатиме, поки користувач зніме її з охорони, а потім скине її.

На цьому етапі система вже зібрана, дальше буде описано сценарій її виконання:

1. Користувач умикає систему охорони для зони 1, рисунок 3.25, після чого виходить з будинку.

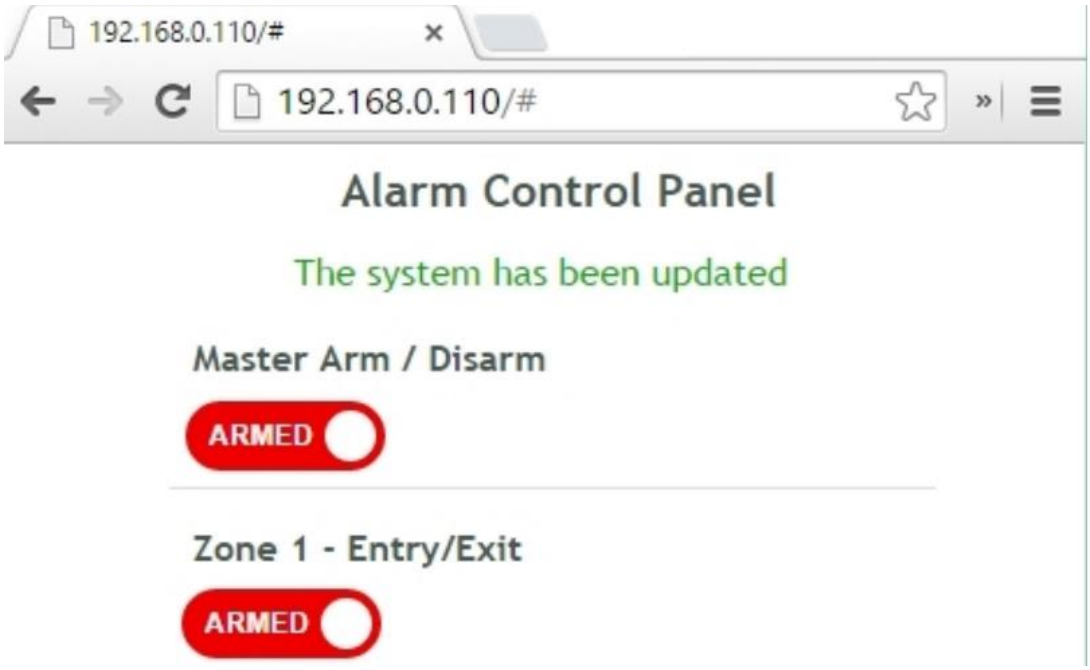


Рисунок 3.25 – Інтерфейс панелі керування з увімкненим захистом зони 1

2. Користувач покидає будинок, через декілька хвилин після цього вмикається охоронна система.

3. Давач монооксиду вуглецю, що встановлений в зоні 1 видає значення показані на рисунку 3.26.

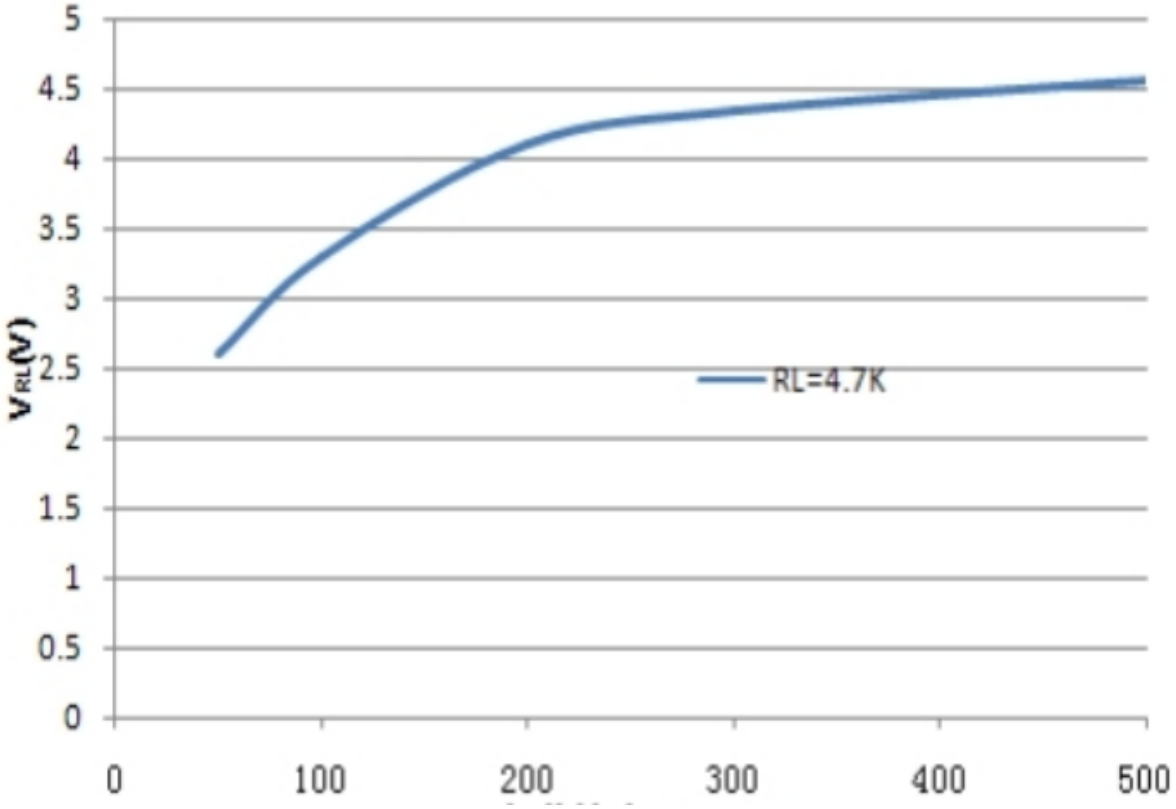


Рисунок 3.26 – Покази давача монооксиду вуглецю, що свідчать про пожежу

4. Після цього користувачу завдяки інтерфейсу веб-адміністратора що був налаштований раніше приходить повідомлення на електрону адресу, про цей інцидент, що дозволяє користувачу зрозуміти де відбулось запалювання та відповідним чином реагувати.

3.6 Висновки

В цьому розділі було описано кроки, та процес збірки системи, процес налаштування, та інші процеси.

Спочатку було підключено усі давачі до плати Raspberry Pi, а також до плати розширення, після цього наступним кроком стало створення веб інтерфейсу, та тестування сумісності.

Також було виконано розбиття на зони будинку, з можливістю увімкнення охоронної сигналізації лише в певних зонах.

Останнім кроком у цьому розділі стало створення та підключення веб-адміна, після чого було створено візуальне представлення скрипта який вшитий в накопичувач на програмованій платі.

ВИСНОВКИ

Програмно-технічний засіб що був створений у ході виконання бакалаврської роботи це охоронна система яка слугує для захисту будинку чи підприємства на якому вона встановлена, це важлива складова кіберфізичної системи «розумний будинок»

Метою роботи було проектування та реалізація програмно-технічного засобу охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi, для автоматизації охоронної сигналізації у кіберфізичній системі "Розумний будинок".

Об’єктом дослідження був процес автоматизації охоронної сигналізації в кіберфізичній системі "Розумний будинок".

Предметом дослідження є програмно-технічний засіб охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi.

Практична цінність роботи полягає в спроектованому та створеному програмно-технічному засобі охоронної сигналізації та нагляду, який може стати складовою частиною в кіберфізичній системі «Розумний будинок», та надає автоматизувати охоронну сигналізацію.

В першому розділі було розглянуто основні поняття концепції «системи безпеки» а також основні складові інтернету речей. Було сформульовано основні ідеї які лягли в основу цих понять. Було складено список з найбільш поширених та популярних готових рішень щодо обладнання будинку охоронною сигналізацією, проаналізовано ринок та оцінено рентабельність створення такої охоронної системи власноруч.

В другому розділі було проведено опис та аналіз основних апаратних та програмних рішень шо були використані по ходу створення охоронної системи на базі мікроконтролера Raspberry Pi Model B Plus. Було складено список усіх необхідних для відтворення проекту вимог, в нього включені як апаратні пристрої, давачі, мікросхеми, електронні пристрої а також утиліти, які покращають процес розробки прошивки та створення середовища для розробки.

В третьому розділі було описано кроки, та процес збірки системи, процес налаштування, та інші процеси. Спочатку було підключено усі давачі до плати Raspberry Pi, а також до плати розширення, після цього наступним кроком стало створення веб інтерфейсу, та тестування сумісності.

Останнім кроком у цьому розділі стало створення та підключення веб-адміна, після чого було створено візуальне представлення скрипта який вшитий в накопичувач на програмованій платі.

Основна практична цінність бакалаврської роботи полягає в реалізованому програмно-технічному засобі охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi є те що він може наглядати за будинком в автоматичному режимі, керувати певними пристроями в будинку а також на основі даних з давачів приймати відповідні рішення, цей апаратно-програмний пристрій є невід’ємною складовою кіберфізичної системи «розумний будинок»

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Paputungan, Irving Vitra, Mahbub Ramadhan Al Fitri, Unan Yusmaniar Oktiawati. Motion and Movement Detection for DIY Home Security System. 2019. 125 p.

2. Najib, A.A., Munadi, R., Karna, N.B.A. Security system with RFID control using E-KTP and internet of things. *Bulletin of Electrical Engineering and Informatics*. 2021. 1436-1445 pp.

3. Desnitsky, V., Levshun, D., Chechulin, A., Kotenko, I.V. Design Technique for Secure Embedded Devices: *Application for Creation of Integrated Cyber-Physical Security System*. 2016. 60-80 pp.

4. Bhatkule, A.V., Shinde, U.B., Zanwar, S.R. Home based security control system using raspberry pi and GSM. 2016. 162-169 pp.

5. Ding, F., Li, Z., Ai, C., Su, R., Zhang, D., Zhu, H. July. Design of an IoT-Based Efficient Security Scheme in Home Wireless System. *International Conference on Artificial Intelligence and Security*. 2018. 287-296 pp.

6. Ahmad, S., Saha, A., Chek, L.W., Mekhilef, S., Azam, T., Ahmed, M., Orabi, M., Ghoneim, S., Alharthi, M., Alamri, B. Smart home automation and security system design based on iot applications. *Asean engineering journal*. 2019. 57-71 pp.

7. Pangaribowo, E.H., Keban, Y.T., Darwin, M. Elderly care: A study on community care services in Sleman, DIY. *Indonesia. Journal of Aging Research*. 2020. 259-317 pp.

8. Coşkun, S. Developing a Face Recognition System for Indoor Security. 2019. 37-121 pp.

9. Best Home Security Systems of. 2022. URL: https://www.cnet.com/home/security/best-home-security-system/ (дата звернення: 25.04.2022).

10. Yaldaie, A. Home automation and security system with the Raspberry Pi. 2017. 141-147 pp.

11. Sandeep, V., Guruprasad Hegde, C.N., Girish, P.P. Face Detection based Locker Security System using Raspberry Pi. 2017. 73-78 pp.

12. Rajasinghe, N. An Intelligent Network Security System. 2018. 47-49 pp.

13. Hong, S. Technology trends and policies for IoT security. 2020. 1-6 pp.

14. Park, E., Del Pobil, A.P., Kwon, S.J. The role of Internet of Things (IoT) in smart cities: *Technology roadmap-oriented approaches*. 2018. 1388 p.

15. Raspberry Pi Model B+. URL: https://raspberry.com.ua/p/raspberry-pi-model-b-plus/ (дата звернення: 29.01.2022).

16. Tiruvayipati, S., Yellasiri, R., Viability of an Uncomplicated IoT SaaS Development for Deployment of DIY Applications Over HTTP with Zero Investment. In Advances in Decision Sciences, Image Processing, Security and Computer Vision. 2020. 206-213 pp.

17. TIP120 – Darlington NPN Transistor. URL: https://components101.com/transistors/tip120-pinout-datasheet-equivalent (дата звернення: 29.01.2022).

18. Complete Guide to Home Security Systems https://www.security.org/home-security-systems/ (дата звернення: 25.04.2022).

19. Ande, R., Adebisi, B., Hammoudeh, M., Saleem, J. Internet of Things: Evolution and technologies from a security perspective. *Sustainable Cities and Society*. 2020. 101-128 pp.

20. Kesavan, G., Sanjeevi, P., Viswanathan, P. August. A 24 hour IoT framework for monitoring and managing home automation. 2016. 1-5 pp.

21. What Is A Home Security System and How Does It Work https://www.security.org/home-security-systems/what-is-a-home-security-system/ (дата звернення: 25.04.2022).

22. Malić, M., Dobrilović, D., Petrov, I. Example of IoT platform usage for wireless video surveillance with support of NoSQL and cloud systems. 2016. 27-34 pp.

23. What is the difference between Model B+ v1.2 and Model 2B. URL: https://raspberrypi.stackexchange.com/questions/39583/what-is-the-difference-between-model-b-v1-2-and-model-2b-v1-1 (дата звернення: 25.04.2022).

24. Raspberry Pi модель B+ 512Мб URL: https://arduino.ua/prod800-raspberry-pi-model-b+-512mb (дата звернення: 25.04.2022).

25. Abraham, S., Vurkaç, M., Miguel, A., Nguyen, N.K., Ong, O.J.S. June. Teaching Embedded Systems in the Context of Internet of Things. 2019.

26. Fortino, G., Guerrieri, A., Pace, P., Savaglio, C., Spezzano, G. IoT Platforms and Security: An Analysis of the Leading Industrial/Commercial Solutions. Sensors. 2022. 296 p.

27. Smart Home Security Systems. URL: https://cpisecurity.com (дата звернення: 25.04.2022).

28. Varghese, L., Deepak, G., Santhanavijayan, A. An IoT analytics approach for weather forecasting using raspberry Pi 3 Model B+. 2019. 1-5 pp.

29. Sałuch, M., Tokarski, D., Grudniewski, T., Chodyka, M., Nitychoruk, J., Woliński, P., Jaworska, B., Adamczewski, G. Raspberry PI 3B+ microcomputer as a central control unit in intelligent building automation management systems. 2018. 196 p.

30. How to Use Raspberry Pi 3 Model B+ V1.2 URL: https://www.instructables.com/How-to-Use-Raspberry-Pi-3-Model-B-V12-for-the-Firs/ (дата звернення: 25.04.2022).

31. Raja, A.A., Naveedha, R., Niranjanadevi, G., Roobini, V. An internet of things (IoT) based security alert system using raspberry pi. 2016. 37-41 pp.

32. Flurry, G. Raspberry Pi 3 Model B+ Setup. In Java on the Raspberry Pi. 2021. 21-48 pp.

33. Raspberry Pi Home Security. URL: https://brinkshome.com/smartcenter/raspberry-pi-home-security (дата звернення: 25.04.2022).

34. Raspberry Pi High Quality security camera. URL: https://www.raspberrypi.com/news/raspberry-pi-high-quality-security-camera/ (дата звернення: 25.04.2022).

35. Raspberry Pi Home Security. URL: https://brinkshome.com/smartcenter/raspberry-pi-home-security (дата звернення: 27.04.2022).

36. Saude, N., Vardhini, P.H. October. IoT based Smart Baby Cradle System using Raspberry Pi B+. 2020. 273-278 pp.

37. Yevsieiev, V., Skripkin, A. Development of Architecture for Mobile Robot Control Based on Raspberry Pi Model 3 B+. 2022.

38. How to Build a Motion-Triggered Raspberry Pi Security. URL: https://www.tomshardware.com/how-to/raspberry-pi-security-camera (дата звернення: 27.04.2022).

39. Keep Your Home Secure with Raspberry Pi. URL: https://www.nutsvolts.com/magazine/article/keep-your-home-secure-with-raspberry-pi (дата звернення: 27.04.2022).

40. Home Alarm System With Raspberry Pi. URL: https://www.researchgate.net/publication/336387526_Security_System_using_Raspberry_Pi (дата звернення: 27.04.2022).

41. Imam Rasyid, A. Optimalisasi Jaringan Dan Monitoring Di SMAN 4 Bandung Menggunakan. 2017. 157 p.

42. Westfall, J. Basic Linux Administration via GUI (Webmin). 2021. 77-110 pp.

43. An IoT based smart solution for leaf disease detection. 2017. 193-198 pp.

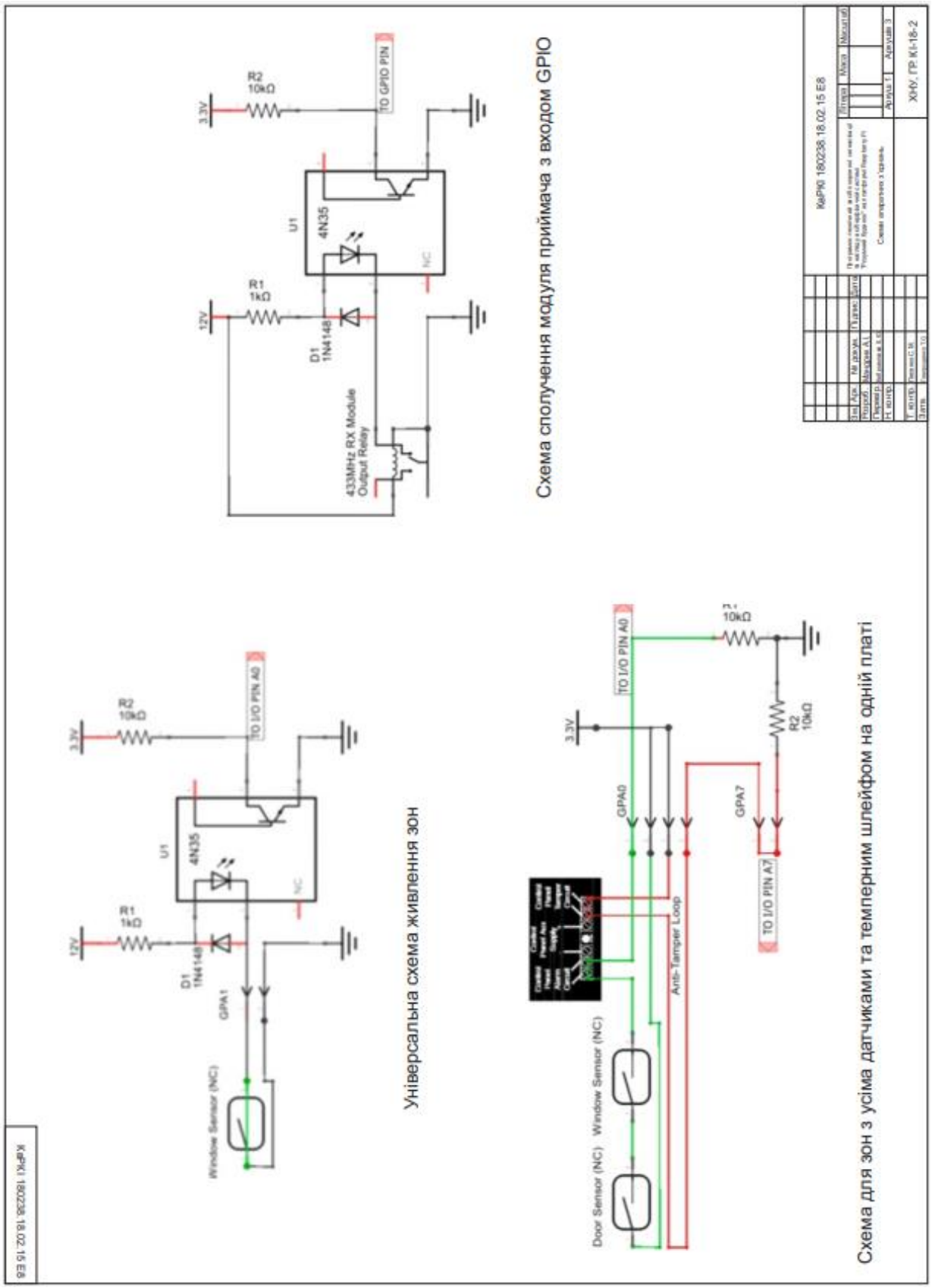
44. Web Servers. URL: https://www.javatpoint.com/web-servers (дата звернення: 29.01.2022).

45. Webmin – A Web Based System Administration Tool for Linux. URL: https://www.tecmint.com/install-webmin-in-linux/ (дата звернення: 27.04.2022).

Додаток А

(обов’язковий)

Копія креслення «Схеми апаратних з’єднань»



Додаток Б

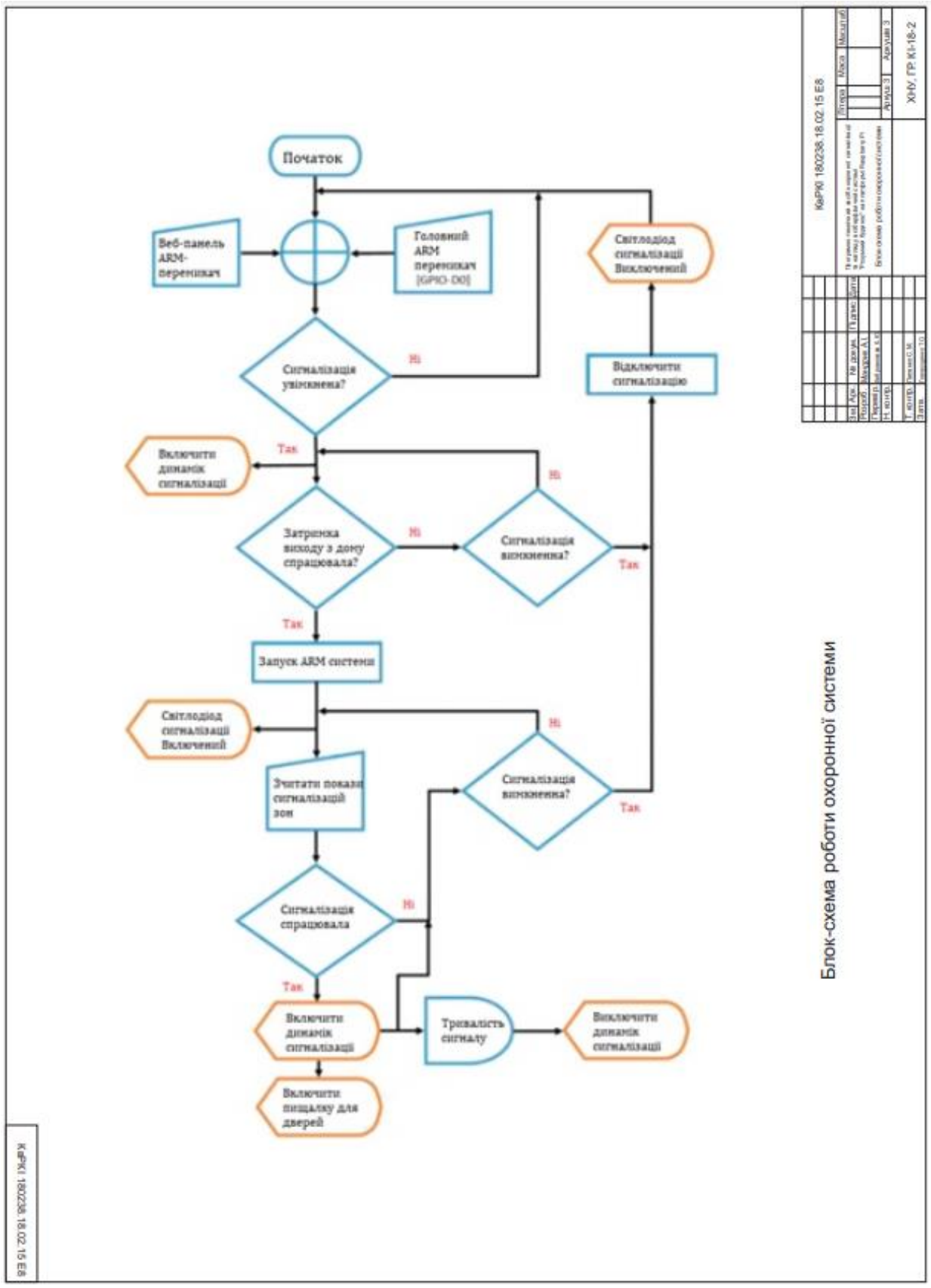
(обов’язковий)

Копія креслення «Інтерфейси програмно-технічного засобу»

Додаток В

(обов’язковий)

Копія креслення «Блок-схеми програм»



Додаток Г

Лістинг коду Bash- скриптів

```
#This function will read the port inputs and set the
#status of each zone
function almReadZoneInputs()
{
 #preserve previous zone status
 ALM_ZONE_INPUT_PREV=$ALM_ZONE_INPUT_STAT
 #read the 8-bit hex value of port a
 ALM_ZONE_INPUT_READ=$(sudo i2cget -y 1 0x20 0x12)
 if [[ $ALM_ZONE_INPUT_READ = *"Error"* ]]; then
 #An error occurred reading the I2C bus - set default value
 ALM_ZONE_INPUT_READ="0x00"
 fi
 #remove the 0x at the start of the value to get the hex value
 local L_HEX=${ALM_ZONE_INPUT_READ:2}
 #convert the hex value to binary
 local L_BIN=$(echo "obase=2; ibase=16; $L_HEX" | bc )
 #zero pad the binary to represent all 8 bits (b7-b0)
 ALM_ZONE_INPUT_STAT=$(printf "%08d" $L_BIN)
 echo "[ALM] Zone I/O Status: $ALM_ZONE_INPUT_STAT
($ALM_ZONE_INPUT_READ)"
 #check each zone input to see if it's in a triggered state
 #a triggered state may be either 1 or 0 depending on the
input's configuration
 #you'll need to set the logic here accordingly for each input
 #the ALM_ZONES_STAT array contains the definitive trigger
value for each input
 #zone 1 test (bit 0)
 local L_FLG=${ALM_ZONE_INPUT_STAT:7:1}
 if [ $L_FLG -eq 0 ]; then ALM_ZONES_STAT[0]=0; else
ALM_ZONES_STAT[0]=1; fi
 #zone 2 test (bit 1)
 local L_FLG=${ALM_ZONE_INPUT_STAT:6:1}
```

```
 if [ $L_FLG -eq 0 ]; then ALM_ZONES_STAT[1]=0; else
ALM_ZONES_STAT[1]=1; fi
 #zone 3 test (bit 2)
 local L_FLG=${ALM_ZONE_INPUT_STAT:5:1}
 if [ $L_FLG -eq 0 ]; then ALM_ZONES_STAT[2]=0; else
ALM_ZONES_STAT[2]=1; fi
 #zone 4 test (bit 3)
 local L_FLG=${ALM_ZONE_INPUT_STAT:4:1}
 if [ $L_FLG -eq 0 ]; then ALM_ZONES_STAT[3]=0; else
ALM_ZONES_STAT[3]=1; fi
 #zone 5 test (bit 4)
 local L_FLG=${ALM_ZONE_INPUT_STAT:3:1}
 if [ $L_FLG -eq 0 ]; then ALM_ZONES_STAT[4]=0; else
ALM_ZONES_STAT[4]=1; fi
 #zone 6 test (bit 5)
 local L_FLG=${ALM_ZONE_INPUT_STAT:2:1}
 if [ $L_FLG -eq 0 ]; then ALM_ZONES_STAT[5]=0; else
ALM_ZONES_STAT[5]=1; fi
 #zone 7 test (bit 6)
 local L_FLG=${ALM_ZONE_INPUT_STAT:1:1}
 if [ $L_FLG -eq 0 ]; then ALM_ZONES_STAT[6]=0; else
ALM_ZONES_STAT[6]=1; fi
 #zone 8 test (bit 7)
 local L_FLG=${ALM_ZONE_INPUT_STAT:0:1}
 if [ $L_FLG -eq 0 ]; then ALM_ZONES_STAT[7]=0; else
ALM_ZONES_STAT[7]=1; fi
 echo "[ALM] Zone Trigger Status: $ALM_ZONES_STAT[*]"
}

# perform exit delay ########################
 echo "[ALM] Alarm now in EXIT DELAY state"
 almSetExitBuzzer 1 #switch on exit buzzer
 COUNTER=$ALM_EXIT_DELAY
 while [[ $STAT_RET_VAL = "1" && $COUNTER -gt 0 ]]; do
 sleep 1
 #read the control panel status file
```

```
. /etc/pi-alarm/alarm.cfg
almGetArmedSwitchStatus #result is returned in STAT_RET_VAL
COUNTER-=1
echo -n "X$COUNTER " # indicate exit mode
done
almSetExitBuzzer 0 #switch off exit buzzer
##############################################
# system now armed - monitor inputs #########
ALM_SYS_ARMED=1
echo "[ALM] Alarm now in ARMED state"
almSetArmedLED 1 #switch on armed LED
#read the control panel status file
. /etc/pi-alarm/alarm.cfg
almReadZoneInputs # > ALM_ZONES_STAT[x]
#check each zone input to set if it's enable
#and has been triggered
#NUM_ZONES setting is stored in alarm.cfg
while [[ $ALM_SYS_ARMED -eq 1 ]]; do
echo -n "A" #indicate armed mode
ALM_ZONE_TRIGGER=0
for (( i=$NUM_ZONES; i>0; i-- )); do
if [[ $ALM_ZONES_STAT[$i-1] -eq 1 ]]; then
#zone has been triggered
echo "[ALM] Zone $i TRIGGERED"
E_VAR="ZONE_ENABLE_$i"
E_VAL=`echo "$E_VAR"` #get zone enabled status loaded
from alarm.cfg
if [[ $E_VAL -eq 1 ]]; then
#zone is enabled
ALM_ZONE_TRIGGER=1 #set alarm triggered flag
echo "[ALM] Zone $i ENABLED - alarm will be triggered"
almUpdateConfigSetting "ZONE_STATUS_$i" "1"
## YOU CAN INSERT CODE HERE TO TAKE CAMERA IMAGE IF
YOU WANT##
## REFER BACK TO CHAPTER 6 ##
fi
```

```
 fi
 done
 . /etc/pi-alarm/alarm.cfg
if [[ $ALM_ZONE_TRIGGER -eq 1 ]]; then
 # alarm has been triggered
 almSetAlarmLED 1
 echo "[ALM] A zone has been triggered"

 ######################################
 # ZONE 1 is the ENTRY zone - if that's triggered then
delay
 if [[ $ALM_ZONES_STAT[0] -eq 1 ]]; then
 # perform entry delay ###########
 echo "[ALM] Alarm now in ENTRY state"
 setExitBuzzer 1 #switch on entry/exit buzzer

 COUNTER=$ALM_EXIT_DELAY
 STAT_RET_VAL="0"
 while [[ $STAT_RET_VAL = "1" && $COUNTER -gt 0 ]]; do
 echo -n "E$COUNTER " #indicate entry mode
 sleep 1
 #read the control panel status file
 . /etc/pi-alarm/alarm.cfg
 almGetArmedSwitchStatus #result is returned in
STAT_RET_VAL
 COUNTER-=1
 done
 fi
 ######################################
 almGetArmedSwitchStatus #result is returned in STAT_RET_
```

UNICHECK

Ім'я користувача:
Кафедра КІ

ID перевірки:
1011399997

Дата перевірки:
31.05.2022 16:03:50 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
31.05.2022 16:04:09 EEST

ID користувача:
100005591

Назва документа: **Мандрик_Програмно-технічний засіб охоронної сигналізації та нагляду в кіберфізичній сис**

Кількість сторінок: **69** Кількість слів: **10449** Кількість символів: **74640** Розмір файлу: **7.94 MB** ID файлу: **1011282727**

2.39%
Схожість

Найбільша схожість: **0.97%** з Інтернет-джерелом (https://github.com/giusbyte/haus/blob/master/poll-magnetic-switch.s..

1.6% Джерела з Інтернету	61	Сторінка 71
1.06% Джерела з Бібліотеки	87	Сторінка 71

0% Цитат

Не знайдено жодних цитат

Не знайдено жодних посилань

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	202

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 0.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 9%**

	Документ		Суммарное совпадение по Базе Данных	
ID: 104276 Название: Програмно-технічний засіб охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi Добавлено в БД: 2022-05-31 Авторы: А.І. Мандрик Руководители: К.Ю. Бобровнікова Консультанты: Опоненты:	Символы	Лексемы	Символы	Лексемы
	68538	556	708 (1%)	9 (2%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Мандрик Андрій Ігорович

Тема: Програмно-технічний засіб охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 ; кількість сторінок записки 65

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є проектування та реалізація програмно-технічного засобу охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi, для автоматизації охоронної сигналізації у кіберфізичній системі "Розумний будинок".

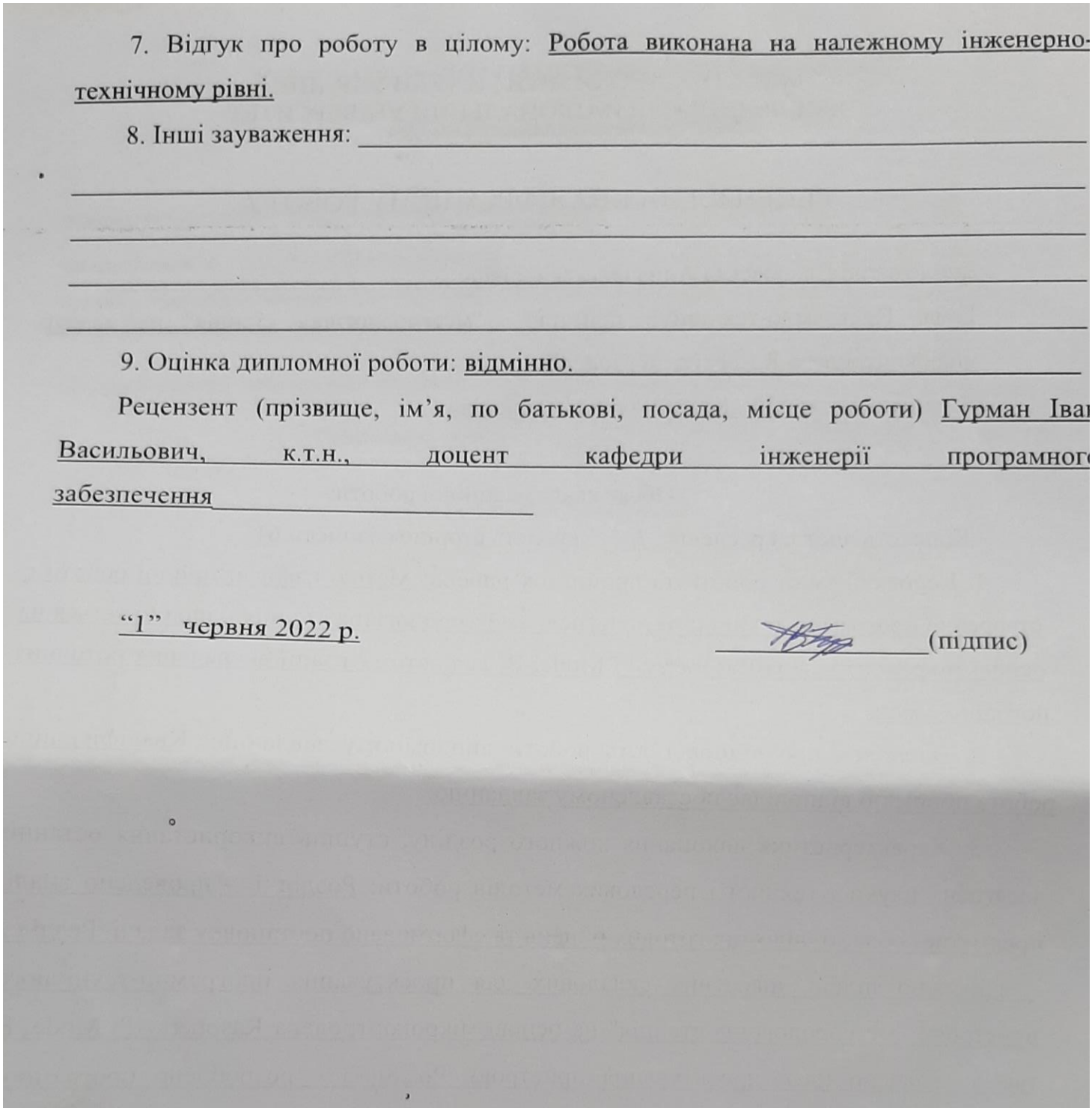
2. Висновок про відповідність роботи дипломному завданню: Кваліфікаційна робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: Розділ 1 – проведено аналіз предметної області, також розглянуто ринкові рішення, що вже існують, оцінено доцільність реалізації, здійснено постановку задачі . Розділ 2 – проведено аналіз використаних апаратних та програмних складових, обґрунтований вибір цих складових та розроблено план створення програмно-технічного засобу. Розділ 3 – розроблено та протестовано програмно-технічний засіб охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi. Усі розділи відповідають завданню.

4. Позитивні сторони роботи: застосування розробленого програмно-технічного засобу охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" надає можливість автоматизувати охоронну сигналізацію в кіберфізичній системі "Розумний будинок".

5. Негативні сторони роботи: недоліком розроблюваного пристрою є погана маштабованість в зв'язку з неможливістю додати велику кількість однотипних давачів.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Оформлення пояснювальної записки відповідає діючим стандартам оформлення документації.



7. Відгук про роботу в цілому: Робота виконана на належному інженерно-технічному рівні.

8. Інші зауваження: _____

_____

_____

_____

_____

9. Оцінка дипломної роботи: відмінно.

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Гурман Іван Васильович, к.т.н., доцент кафедри інженерії програмного забезпечення

"1" червня 2022 р. _____(підпис)

Завідувачу кафедри КПС
д-ру техн.наук, проф. Говорущенко Т. О.

Мандрика А. І.
ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ-18-2

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність плагіату ознайомлений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

23.05
дата

підпис

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА СИСТЕМНОГО ПРОГРАМУВАННЯ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Програмно-технічний засіб охоронної сигналізації та нагляду в кіберфізичній системі "Розумний будинок" на платформі Raspberry Pi

Автор: Мандрик Андрій Ігорович

Спеціальність: 123 – Компютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Бобровнікова Кіра Юліївна, к.т.н,

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення, які мають місце в розділах аналізу існуючих аналогів та прототипів, не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
2) усі запозичення є фрагментарними, або мають належним чином оформленні посилання;
3) в якості запозичень в окремих місцях системою зафіксовано зарезервовані ключові слова мови програмування, які використовуються для розв'язку великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення.
4) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів із україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 2.39% і адресується до 148 першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ К. Ю. Бобровнікова

Гарант ОП _____ С. М. Лисенко

Завідувач кафедри КІС _____ Т. О. Говорущенко