

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Робототехнічна система віддаленого нагляду за тваринами з використанням мікроконтролера ESP32-CAM

Назва теми

КвРКІ.190198.01.24.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 КІ2с-19-1


Підпис

В.М. Семенишин
Ініціали, прізвище

Керівник


Підпис, дата

О.С. Засорнов
Ініціали, прізвище

Нормоконтролер


Підпис, дата

С.М. Лисенко
Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
Інженерії та системного
Програмування


Підпис

Т.О. Говоруценко
Ініціали, прізвище

« » червня 2022 р.

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 Г.О.Говоруценко

“ 11 ” 01 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Семенішиному Валерію Михайловичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Робототехнічна система віддаленого нагляду за тваринами з використанням мікроконтролера ESP32-CAM

Керівник проекту (роботи) Засорнов О.С к.т.н., доцент

Прізвище, ім'я, по батькові науковий ступінь, вчене звання

Затверджена наказом ректора університету від 06.01.2022 р. № 1

2. Строк подання студентом проекту (роботи) на кафедру 07.06.2022 р.

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

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Огляд існуючих систем для розв'язання завдання

Обґрунтування вибору компонентів та середовища реалізації

Реалізація робототехнічної системи віддаленого нагляду за тваринами з використанням мікроконтролера ESP32-CAM



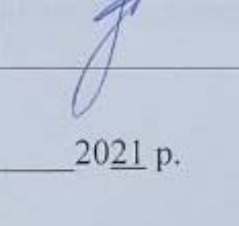
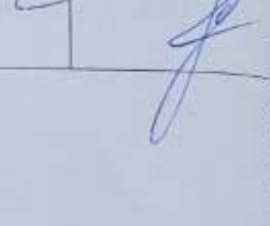
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Робототехнічна система годування та веб-сторінка перегляду

Компоненти робототехнічної системи віддаленого нагляду за тваринами з використанням мікроконтролера ESP32-CAM

Середовище розробки системи

6. Консультанти розділів дипломного проекту (роботи)

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

7. Дата видачі завдання « 06 » 09 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітки
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2022	виконав
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2022	виконав
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2022	виконав
4	Обґрунтування вибору компонентів та середовища реалізації	01.04.2022	виконав
5	Робота технічної системи віддаленого нагляду за тваринами з використанням мікроконтролера ESP32-CAM	30.04.2022	виконав
6	Оформлення пояснювальної записки згідно вимог	31.05.2022	виконав
7	Попередній захист ВКР	02.06.2022	виконав
8	Захист ВКР на засіданні ЕК	Червень 2022 року	

Студент

Керівник проекту (роботи)

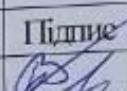


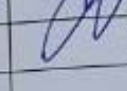

Підпис


Підпис

В.М. Семенишин
Ініціали, прізвище

О.С. Засорнов
Ініціали, прізвище

Ф о р м а т	Позначення	Найменування	К і л · л и с т і в	№ ек з	П р и м і т к а
		Текстові документи			
	КвРКІ 190198.03.24 ПЗ	Пояснювальна записка	56		
		Графічні матеріали			
	КвРКІ 190198.03.24 Е8	Робототехнічна система	1		
		годування та веб-сторінка			
		перегляду			
	КвРКІ 190198.03.24 Е8	Компоненти	1		
		робототехнічної системи			
		віддаленого нагляду за			
		тваринами з			
		використанням			
		мікроконтролера ESP32-			
		CAM			
	КвРКІ 190198.03.24 Е2	Середовище розробки	1		
		системи			

КвРКІ 190198.03.24 ВП				
м	Арк	№ докум	Підпис	Дата
розробив		Семеншин		
перевір.		Засорнов		
І. копир.		Лисенко		
Затв.		Говорушенко		
Відомість проекту			Літера	Аркуш
			У	1
			ХНУ, КІ2с-19-1	

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Роботехнічна система віддаленого нагляду за тваринами з використанням мікроконтролера ESP32-CAM».

Автор роботи: Семенишин Валерій Михайлович.

Керівник роботи: Засорнов Олександр Сергійович.

Пояснювальна записка: 56 с., 47 рис., 4 табл., 4 дод., 28 джерел.

Графічна частина: 10 презентаційних слайдів.

ПРОГРАМОВАНИЙ АПАРАТНИЙ ЗАСІБ МІКРОКОНТРОЛЕР КОМП'ЮТЕРНА СИСТЕМА ESP32-CAM

Метою роботи є розробка система віддаленого нагляду за тваринами.

У цій роботі розроблена система віддаленого нагляду за тваринами із використанням мікроконтролера ESP32-CAM. Розроблено апаратний засіб нагляду за тваринами на основі мікроконтролера ESP32-CAM. Розроблена система нагляду, дозволяє здійснювати віддалений нагляд за тваринами, а також система реалізує графічний людино-машинний інтерфейс керування процесом на основі графічної сенсорної панелі оператора.

Підпис студента



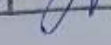


Дата

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ.....	4
ВСТУП.....	5
1. ОГЛЯД ІСНУЮЧИХ СИСТЕМ ДЛЯ РОЗВ'ЯЗАННЯ ЗАВДАННЯ.....	7
1.1 Furbo Dog Camera.....	7
1.2 Pet Camera.....	7
1.3 Робот Skymee Owl.....	9
1.4 MeasureON.....	9
1.5 PETLIBRO Automatic Cat Feeder.....	9
1.6 Sure Petcare -SureFlap – SureFeed – Годівниця Microchip Pet.....	11
1.7 Virginia Tech.....	12
1.8 Висновки та постановка задачі.....	12
2 ОБГРУНТУВАННЯ ВИБОРУ КОМПОНЕНТІВ ТА СЕРЕДОВИЩА РЕАЛІЗАЦІЇ.....	14
2.1 Апаратне середовище реалізації.....	14
2.2 Процес роботи ESP8266 з Arduino IDE.....	17
2.2.1 Установка за допомогою Boards Manager.....	18
2.2.3 Використання Arduino IDE.....	18
2.2.3 Компіляція та завантаження.....	20
2.5 Висновки.....	25
3 РЕАЛІЗАЦІЯ РОБОТОТЕХНІЧНОЇ СИСТЕМИ ВІДДАЛЕНОГО НАГЛЯДУ ЗА ТВАРИНАМИ З ВИКОРИСТАННЯМ МІКРОКОНТРОЛЕРА ESP32-CAM.....	26
3.1 Застосування ArduCAM-M-2MP ESP8266 для реалізації робототехнічної системи віддаленого нагляду за тваринами.....	26
3.2 Специфікація застосованих компонентів.....	26
3.3 Процес збірки робототехнічної системи віддаленого нагляду за тваринами.....	32

КВРКІ 190198.01.24 ПЗ

Зм.	Арк	№докум.	Підпис	Дата				
Виконав		Семетини В.			Робототехнічна система віддаленого нагляду за тваринами з використанням мікроконтролера ESP32-CAM	Літера	Аркуш	Аркушів
Перевір.		Засорин О.С.				2	81	
Н.контр.		Дісенко С.М.			ХНУ, КІ2с-19-1			
Затвер.		Споруджено						

3.4 Встановлення ПЗ засобами Arduino IDE	33
3.5 Встановлення ядра Arduino ESP32.....	35
3.6 Клонування та завантаження ядра	35
3.7 Встановлення інструментів Xtensa та ESP32	37
3.8 Установка драйверів FTDI	38
3.9 Налаштування роботи Arduino з WiFi	41
3.10 Встановлення бібліотеки Arduino	44
3.11 Застосування бібліотеки Blink.....	50
3.12 Програмування проєкту.....	51
3.13 Висновки	58
ВИСНОВКИ.....	60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	61
Додаток А.....	65
Копія креслення “ Робототехнічна система годування та веб-сторінко перегляду”	65
Додаток Б.....	66
Копія креслення “ Компоненти робототехнічної системи віддаленого нагляду за тваринами з викорис-танням мікроконтролера ESP32-CAM ”	66
Додаток В	67
Копія креслення “ Середовище розробки системи”	67
Додаток Г Код програмного забезпечення	68

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ

WLAN – Wireless Local Area Network
WI-FI – Wireless Fidelity
RSOS – Reconfigurable Smart Object System
MQTT – Message Queuing Telemetry Transport
CoAP – Constrained Application Protocol
HTTP – Hypertext Transfer Protocol
IoT – Internet of Things
M2M – Machine-to-Machine
REST – Representational State Transfer
TCP – Transmission Control Protocol
HTTPS – Hypertext Transfer Protocol Secure
MQTT – Message Queuing Telemetry Transport
UPnP – Universal Plug and Play
LLMNR – Link-Local Multicast Name Resolution
mDNS – multicast DNS
Zeroconf – Zero Configuration Networking
SSDP – Simple Service Discovery Protocol
SOAP – Simple Object Access Protocol
DDD – device descriptor document
АЦП – Аналого-цифровий перетворювач
LCD – Liquid-crystal display
SSID – Service Set Identifier
SSDP – Simple Service Discovery Protocol
UDN – Unique Device Name

					КВРКІ 190198.01.24 ПЗ	Анк
Зм.	Арк.	№докум.	Підпис	Дата		4

ВСТУП

Сьогодні стають популярними інформаційні технології віддаленого керування не тільки у промисловій галузі, а й у повсякденному побутовому житті.

У системі віддаленого нагляду за тваринами монітори можуть повідомити, якщо домашня тварина хворіє, нудьгує або поводить погано. Камера для домашніх тварин дозволить вийти з дому і все одно стежити за своєю хатньою твариною. Якщо присутня турбота поки немає, спостереження за домашніми тваринами може бути актуальним рішенням.

На ринку є багато різних систем моніторингу для догляду за тваринами. Деякі з них є простими камерами безпеки, які дозволяють лише спостерігати за своїм вихованцем. Інші системи мають функції, які дозволяють не тільки бачити, а й розмовляти зі своїм вихованцем або навіть ласувати його.

деякі системи дозволяють вашому вихованцеві бачити і «розмовляти» з вами. Такий тип віддаленої взаємодії робить розставання менш стресовим для багатьох домашніх тварин та їх власників. Частий електронний контакт з власником може зменшити тривогу та нудьгу вихованця при розлуці. Як приємно повернутися додому з роботи, щоб знайти непошкоджений, наприклад, диван.

Моніторинг домашніх тварин може бути дуже складним. Деякі пристрої моніторингу дозволяють вашому вихованцеві ініціювати виклик. Індикатор на датчику, керованому лапою, повідомляє вашому вихованцеві, коли ви можете поспілкуватися. Для цієї функції потрібен добре навчений вихованець і власник тварини!

Деякі підрозділи стежать за вихованцем і його оточенням. Доступні програми, які вимірюють температуру навколишнього середовища, щоб ви могли дистанційно регулювати термостат. Чи показує монітор сонце? Розумний будинок дозволить дистанційно закривати жалюзі.

А деякі пристрої мають додатковий хмарний сервіс для зберігання всіх фото та відео вашого вихованця. Існує навіть камера, яка кріпиться до нашійника або

					КВРКІ 190198.01.24 ПЗ	Анк
Зм.	Арк.	№докум.	Підпис	Дата		5

шлейки вашого вихованця, що дозволяє вам бачити світ з його точки зору. Це особливо цікаво з домашніми тваринами на вулиці, якщо розглядати камеру, яка працює без Wi-Fi.

Таким чином, завдання розроблення нових рішень нагляду за тваринами є дуже актуальним.

					КвРКІ 190198.01.24 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		6

1. ОГЛЯД ІСНУЮЧИХ СИСТЕМ ДЛЯ РОЗВ'ЯЗАННЯ ЗАВДАННЯ

1.1 Furbo Dog Camera

Сьогодні на ринку присутня велика кількість готових рішень нагляду за тваринами.

Furbo Dog Camera: Treat Tossing, Full HD Wifi Pet Camera і 2-стороннє аудіо, призначена для собак, сумісна з Alexa (Спостереження, камера для тварин).

Furbo Dog Camera була розроблена для собак та їхніх найкращих друзів. Він оснащений двостороннім звуком, нічним баченням і диспенсером для ласощів для домашніх тварин — усіма можна керувати з мобільного додатка. Ви можете спостерігати за своїм цуценям і грати з ним з роботи, під час відпустки або з коридору [1].

Функції:

1. Камера 1080p Full HD і нічне бачення: відео в прямому ефірі для спостереження за вашим домашнім улюбленцем на телефоні з ширококутним оглядом на 160 градусів вдень і вночі.

2. Двостороння аудіо та гавкіт: датчик гавкання Furbo визначає, коли ваша собака гавкає. Він надсилає push-сповіщення на ваш смартфон, коли виявляє гавкіт. Дізнайтеся, що відбувається вдома, і поговоріть, щоб заспокоїти їх за допомогою програми.

3. Забавне кидання ласощів: киньте ласощі своїм собакам за допомогою безкоштовного додатка Furbo для iOS/Android. Наповніть його улюбленими ласощами ваших собак і пограйте в гру.

4. Є можливість додавати голосове керування, поєднавши його з екранним пристроєм Alexa.

1.2 Pet Camera

					КвРКІ 190198.01.24 ПЗ	Анк
Зм.	Арк.	№докум.	Підпис	Дата		7

Камера для котів VStarcam з лазерною бездротовою камерою для собак 1080P Cat Toys, нічне бачення, звукові сповіщення про рух, програма дистанційного керування, домашня камера безпеки для домашніх тварин [2].

Функції:

1. Автоматична лазерна іграшка для котів і спільний доступ до облікового запису - камера для домашніх тварин VStarcam з лазерною іграшкою, щоб дражнити та тренувати домашніх тварин. Дистанційно керуйте обертанням лазера за допомогою мобільного телефону або встановлюйте круїз із фіксованою точкою, спілкуючись із домашніми тваринами, поки ви на вулиці. Камерою радіоняня для домашніх тварин можна поділитися з іншими, ви можете насолоджуватися солодкими спогадами зі своєю родиною.

2. Монітор камери тварин у приміщенні - камера Full HD 1080P Puppy, перевіряйте своїх домашніх тварин у будь-який час зі свого телефону. Kitty cam із ширококутним оглядом 107°, діапазоном повороту на 355° по горизонталі та 120° по вертикалі, забезпечує захист на 360° для вашого дому. Підтримка iOS / Android із 2,4 ГГц Wi-Fi .

3. Нічне бачення та камера для домашніх тварин із додатком для телефону, двостороннє аудіо - Інтелектуальні подвійні фільтри ГЧ-вирізання забезпечують високу якість зображення як вдень, так і вночі. Нічне бачення до 10 м в темряві. Завдяки вбудованому мікрофону та динаміку ви можете коли завгодно розмовляти зі своїм улюбленцем за допомогою цієї котячої камери. Цуценята/кошенята розважаються за допомогою камер, навіть коли вони вдома.

4. Виявлення звуку руху та просте налаштування - камера для собак із додатком для телефону надішле на ваш телефон сигнал тривоги, коли виявить гавкіт або слід руху. Завдяки цьому розумному монітору для котів і собак забезпечте безпеку вашого дому 24/7. Швидке налаштування Wi-Fi через додаток на смартфоні, забезпечує віддалений доступ.

5. Карта TF 128G / Хмарне сховище - Камера VStarcam Cat з підтримкою лазера 24 години на 24 години відеозйомки, максимум 128G TF-карта (НЕ ВХОДИТЬ) і хмарне сховище. Ви можете транслювати свої кадри в прямому ефірі або

					КВРКІ 190198.01.24 ПЗ	Анк
Зм.	Арк.	№докум.	Підпис	Дата		8

відтворювати їх за допомогою шкали часу. 12 місяців і цілодобове дружнє обслуговування клієнтів.

1.3 Робот Skymee Owl

Робот Skymee Owl — як домашня тварина для вашого вихованця. Частково це робот, частково дозатор ласощів, частково HD-камера, але все це весело для домашніх тварин та їх власників [3].

Функції:

1. Рухомий дистанційно керований робот.
2. Pro Bullet Дозатор для лікування.
3. Pro Bullet Нічне бачення.
4. Pro Bullet Двостороння розмова.

1.4 MeasureON

MeasureON забезпечує безперервний моніторинг здоров'я вашої тварини в режимі реального часу для цілодобового моніторингу. Шлейка дозволяє автоматично перевіряти стан здоров'я та життєво важливі функції вашої собаки в той час, коли вона почувається погано.

Розумна шлейка для собак дозволяє легко вимірювати життєві показники вашої собаки за допомогою Wi-Fi і програми VetMeasure. Це допоможе вам визначити, чи почувається ваша собака погано, порівнявши її життєво важливі показники зі звичайними показниками здоров'я.

Як тільки ви дізнаєтеся, що ваша собака погано себе почуває, ви можете зв'язатися зі своїм ветеринаром, щоб дізнатися найкращий план дій для вашого вихованця, щоб він знову почував себе найкращим чином.

1.5 PETLIBRO Automatic Cat Feeder.

					КВРКІ 190198.01.24 ПЗ	Анк
Зм.	Арк.	№докум.	Підпис	Дата		9

Автоматична годівниця для кішок PETLIBRO, тимчасова годівниця для котів з мішком для осушувачів для сухого корму для домашніх тварин, програмованим контролем порцій 1-4 прийоми їжі на день і диктофоном для кішок і собак [4].

Функції:

1. Простий у використанні та обслуговуванні: вбудований РК-екран PETLIBRO автоматичної годівниці для собак дозволяє швидко налаштувати, а кнопка FEED для додаткового годування; Правильний кут, щоб їжа не накопичувалася в розетці для їжі, а харчовий резервуар і лоток можна знімати для очищення. Конструкція надійного замка кришки, щоб домашні тварини не отримували їжу.

2. Гнучкий графік годування: більше не потрібно дзвінків пробудитися перед світанком або хвилюватися, коли ви переробите вночі! Проста в програмуванні автоматична годівниця PETLIBRO з таймером, щоб давати їжу в потрібний час, 1-4 прийоми їжі на день і до 9 порцій на один прийом їжі, призначений для здорового харчування ваших кішок і маленьких собак.

3. Автоматична годівниця для котів на 9 чашок стабільно забезпечує їжу для вашої кішки та маленької собаки безперервно протягом декількох днів, запевняючи вас, що вони залишаться ситими та щасливими, коли ви на короткому відпустці або працюєте довгі години, а також матимете з осушувачем сумка для збереження їжі свіжою.

4. Подвійне джерело живлення: використовуйте адаптер постійного струму на 5 В, підтримуючи живлення автоматичної годівниці для кішок за допомогою 3 лужних батарейок D-елементів (не входять у комплект), встановлених на випадок відключення електроенергії, щоб ваш вихованець отримував їжу стабільно. (Обладнаний пам'яттю налаштувань для запобігання раптового збою живлення)

5. Диктофон: залишайтеся на зв'язку зі своїм домашнім улюбленцем, за телефонувавши в автоматичну годівницю PETLIBRO перед їжею, за допомогою кліпу для запису голосу на 10 секунд, таким чином покращуючи зв'язок зі своїм домашнім улюбленцем, регулярно годуючи його, відчуваючи себе в безпеці та догляді.

					КВРКІ 190198.01.24 ПЗ	Арк
						10
Зм.	Арк.	№докум.	Підпис	Дата		

Автоматична годівниця PetSafe Healthy для кішок і собак – місткість 24 чашки. Якщо необхідно годувати кількох домашніх тварин або тварина, здається, вдихає їжу, ця годівниця допоможе вам. Забезпечення до дванадцяти прийомів їжі на день, а також функція повільного годування, як і коли їсть ваша тварина, залежить від вас, навіть коли вас немає вдома.

Ця модель вміщує до 24 чашок їжі та може використовуватися з сухими або напіввологими крупами.

Спокій про те, що вихованець залишиться нагодований, але не зможе наїстися, — це завжди чудово. Планування багаторазового годування для кількох тварин також є перевагою. А кому не подобається мати можливість зберігати кілька фунтів дрібної їжі, замість того, щоб наповнювати машину щодня?

Функції:

1. Сховище великої ємності
2. Повільне годування та годування за розкладом
3. Конструкція, яка захищає від домашніх тварин, робить його непроникним для сторонніх лап.

1.6 Sure Petcare -SureFlap – SureFeed – Годівниця Microchip Pet

Якщо коли-небудь доводилося шукати їжу за рецептом у ветеринара, відомо, наскільки це може бути дорого. Якщо треба годувати кількох домашніх тварин, і лише одному з них потрібен спеціальний корм, тримати кожну істоту окремо під час їжі може знадобитися багато часу. Тепер доступ до прописаної їжі матиме лише пацієнт. У будь-якому випадку мікрочіпування вашого вихованця має масу переваг, тому це ще один позитив [5].

Якщо є кілька домашніх тварин, ви використовуєте спеціальний корм або маєте інші обмеження, ця годівниця полегшить вам життя. Він зберігає до 32 унікальних ідентифікаторів, тому ви можете створити кілька налаштувань залежно від ситуації, що змінюється.

Функції:

					КвРКІ 190198.01.24 ПЗ	Анк
Зм.	Арк.	№докум.	Підпис	Дата		11

1. Активується мікрочіпом або включеною міткою
2. Неопренове ущільнення для збереження свіжості
3. Легке у використанні програмування однією кнопкою

1.7 Virginia Tech

Бездротові пристрої допомагають контролювати здоров'я худоби та місце розташування. Дослідники з Університету Карнегі-Меллона та Virginia Tech розробили універсальну, надійну та стійку до атак бездротову сенсорну мережу для інтелектуального моніторингу тварин [6].

Команда розробила 5-сантиметрову сенсорну вушну бирку, яка може контролювати біометричні дані, прискорення та місцезнаходження тварин. Працює від сонячної енергії, мітки не потрібно буде перезаряджати або замінювати. Бирка, прикріплена біля вуха, відповідає існуючим вушним биркам, що використовуються в промисловості, і не завдасть шкоди тварині.

Ця інноваційна сенсорна мережа використовуватиме малопотужну глобальну мережу (LP-WAN), щоб дозволити персоналу з догляду за тваринами дистанційно контролювати поведінку та здоров'я великої рогатої худоби. Пристрої LP-WAN мають на меті забезпечити бездротове підключення на надзвичайно низьких швидкостях на відстані в кілька миль.

Найбільшою проблемою в розвитку цих мереж є масштаб. Побудова мережі, в якій тисячі пристроїв спілкуються одночасно, призведе до перешкод і перешкоджатиме досягненню сигналів до базової станції за кілька кілометрів [7-8].

1.8 Висновки та постановка задачі

У даному розділі було проведено загальний аналіз відомих засобів віддаленого нагляду тварин.

Більшість засобів мають достатній функціонал для вирішення задачі віддаленого нагляду за тваринами.

					КВРКІ 190198.01.24 ПЗ	Анк
Зм.	Арк.	№докум.	Підпис	Дата		12

Головним недоліком розглянутих засобів є висока вартість рішень.

Для вирішення поставленого завдання можливим є застосування програмно-апаратні засобів, що базуються на системі ArduCAM-M-2MP ESP8266 Evaluation Kit.

					КвРКІ 190198.01.24 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		13

2 ОБГРУНТУВАННЯ ВИБОРУ КОМПОНЕНТІВ ТА СЕРЕДОВИЩА РЕАЛІЗАЦІЇ

2.1 Апаратне середовище реалізації

Для вирішення поставленого завдання було застосовано програмно-апаратні застоби, що базуються на системі ArduCAM-M-2MP ESP8266 Evaluation Kit.

Набір включає камеру та WIFI IoT на основі модулів ArduCAM-Mini-2MP-V2 і ArduCAM-ESP8266-Nano [9-10].

Користувач може реалізувати 2-мегапіксельну WIFI-камеру за допомогою протоколу HTTP або WebSocket на ESP8266, і камера може функціонувати як точка доступу.

Також мобільний телефон/ПК можуть підключатися до камери безпосередньо або діяти як станція, яка підключається до наявного домашнього маршрутизатора (рисунок 2.1).

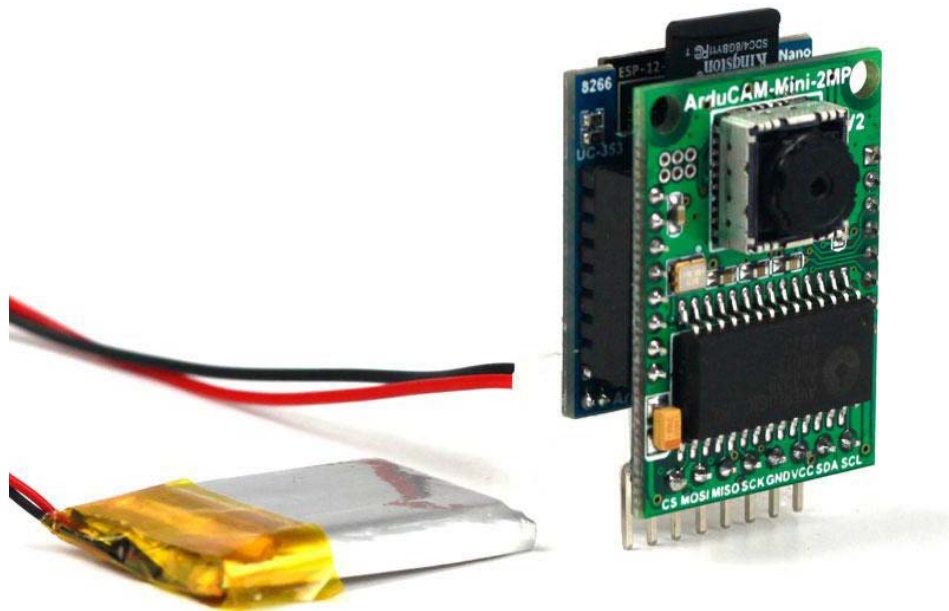


Рисунок 2.1 – програмно-апаратні застоби, що базуються на системі ArduCAM-M-2MP ESP8266 Evaluation Kit

					КВРКІ 190198.01.24 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		14

Набір може приймати нерухоме зображення JPEG з повною роздільною здатністю 2 МП, але потокове відео з низькою роздільною здатністю з низькою частотою кадрів через обмеження ESP8266.

Набір може живитися від USB або батареї за допомогою вбудованих схем зарядки (рисунок 2.2).

Комплект також можна використовувати окремо, він ідентичний камері ArduCAM-Mini-2MP і модулю ESP8266.

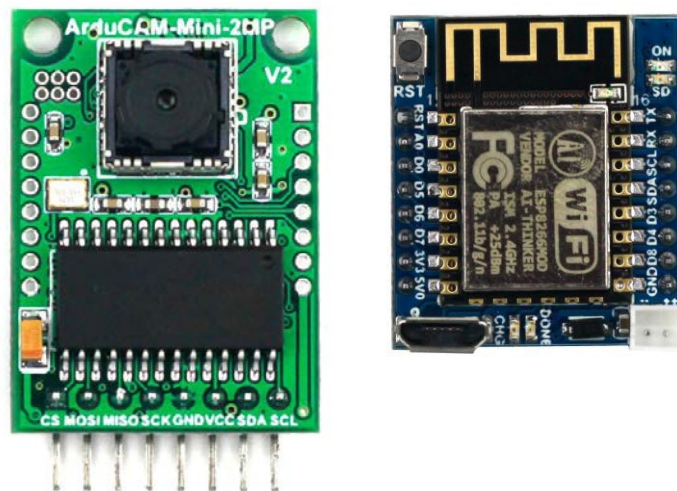


Рисунок 2.2 – ArduCAM-Mini-2MP-V2 і ESP8266-Nano

На модулі камери ArduCAM-Mini-2MP є два роз'єми, 8-контактний роз'єм на нижній стороні є стандартним роз'ємом ArduCAM.

Інший подвійний 16-контактний роз'єм - це спеціальний роз'єм для модуля ArduCAM-ESP8266-Nano, його можна добре поєднати з модулем ESP8266 Nano безпосередньо.

В таблиці 2.1 Визначено контактів стандартного роз'єму програмно-апаратних засобів ArduCAM-M-2MP.

На рисунку 2.3 показана схема підключення між ArduCAM mini 2MP і модулем ESP8266 Nano.

Таблиця 2.1 – Визначення контактів стандартного роз'єму ArduCAM-M-2MP

№ PIN	PIN-NAME	ТИП	ОПИС
1	CS	Вхідні дані	Вхід вибору підпорядкованої мікросхеми SPI
2	MOSI	Вхідні дані	Головний вихід SPI, підпорядкований вхід
3	MISO	Вихід	Головний вхід SPI підлеглий вихід
4	SCLK	Вхідні дані	Послідовний годинник SPI
5	GND	Земля	Заземлення
6	VCC	СИЛА	3,3 В~5 В Блок живлення
7	ПДР	Двонаправлений	Двопровідний послідовний інтерфейс даних I/O
8	SCL	Вхідні дані	Двопровідний послідовний інтерфейс clock

Визначення контактів ArduCAM-ESP8266-Nano детально представлено в таблиці 2.2.

Таблиця 2.3 – Визначення контактів ArduCAM-ESP8266-Nano

№ PIN	PIN-NAME	ТИП	ОПИС
1	RST	Вхідні дані	ESP8266скинути вхід
2	A0	Вхідні дані	Аналоговий вхід
3	D0/GPIO16	Вхідні дані	чїпвиберїть для камери
4	D5/SCK	Земля	Апаратне забезпечення SPI SCLK
5	D6/MISO	Вхідні дані	Апаратне забезпечення SPI MISO
6	D7/MOSI	Вихід	Апаратне забезпечення SPI MOSI
7	3,3 В	СИЛА	3,3 ВБлок живлення

Кінець таблиці 2.3 – Визначення контактів ArduCAM-ESP8266-Nano

8	5V	СИЛА	5VБлок живлення
9	GND	Земля	Заземлення
10	D8/GPIO15	IN OUT	GPIO
11	D4/GPIO2	IN OUT	GPIO
12	D3/GPIO0	IN OUT	Порт SD-карти
13	SDA/GPIO04	Двонаправлений	Двопровідний послідовний інтерфейс даних I/O
14	SCL/GPIO05	Вихід	Двопровідний послідовного інтерфейсу clock
15	Rx/GPIO03	Вхідні дані	Апаратне забезпечення UARTRX
16	Tx/GPIO01	Вихід	Апаратне забезпечення UARTTX

Схема підключення усіх наявних контактів подана на рисунку 2.3.

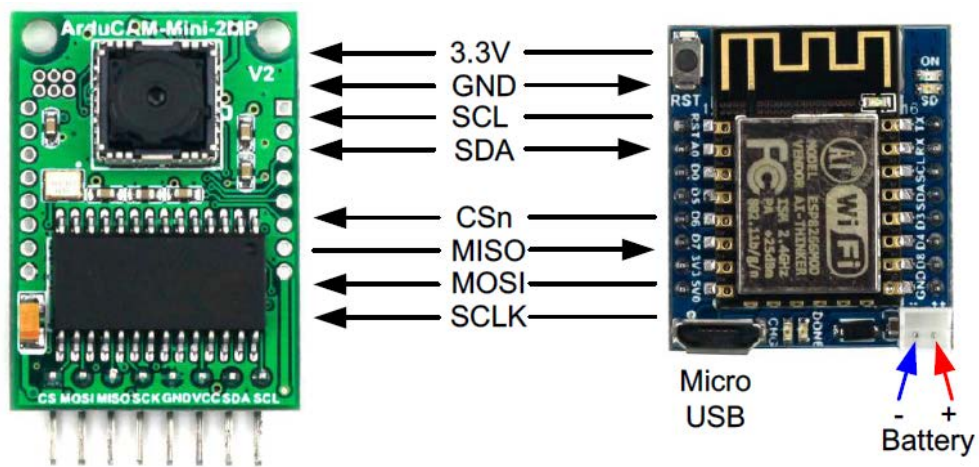


Рисунок 2.3 – Схема підключення

2.2 Процес роботи ESP8266 з Arduino IDE

Розглянемо процес роботи та розроблення програмного забезпечення для плати ArduCAM ESP8266 Nano за допомогою редактора коду Arduino IDE.

2.2.1 Установка за допомогою Boards Manager

Починаючи з версії 1.6.4, Arduino дозволяє встановлювати пакети сторонніх платформ за допомогою Boards Manager [11-12].

В загальному є можливість застосування рішень для операційних систем Windows, Mac OS і Linux (32 і 64 біт).

Для вирішення завдання необхідно встановити Arduino 1.6.8 з веб-сайту Arduino.

Після запуску редактора Arduino необхідно відкрити вікно «Параметри».

Наступним кроком є необхідність введення наступного посилання в полі Додаткові URL-адреси керівника ради.

Там можна додати кілька URL-адрес, розділивши їх комами, наприклад, http://www.arducam.com/downloads/ESP8266_UNO/package_ArduCAM_index.json.

Далі потрібно відкрити Диспетчер плат у меню Tool -> Board та встановити необхідні пакети.

В роботі було застосовано пакет доповнень ArduCAM_ESP8266_UNO.

Необхідно звернути увагу, що вихідний файл інсталюваного пакета знаходиться в C:\Users\Semeniuk\AppData\Local\Arduino\, C:\Users\Semeniuk\AppData\Local\Arduino\наприклад.

Вікно підключення пакету аддонів ArduCAM ESP8266 UNO представлено на рисунку 2.4.

2.2.3 Використання Arduino IDE

Після інсталяції пакета додаткових компонентів плати ArduCAM ESP8266 Nano, можна вибрати плату в меню Tool -> Board.

Існує кілька готових прикладів із File->Examples->ArduCAM.

Також можливим є використання базових прикладів безпосередньо або як відправну точку для розробки власного проєкту.

					КВРКІ 190198.01.24 ПЗ	Анк
Зм.	Арк.	№докум.	Підпис	Дата		18

З цією метою необхідно вибрати плату ArduCAM_ESP8266_UNO з меню Tool->Board (рисунок 3.5) [13-14].

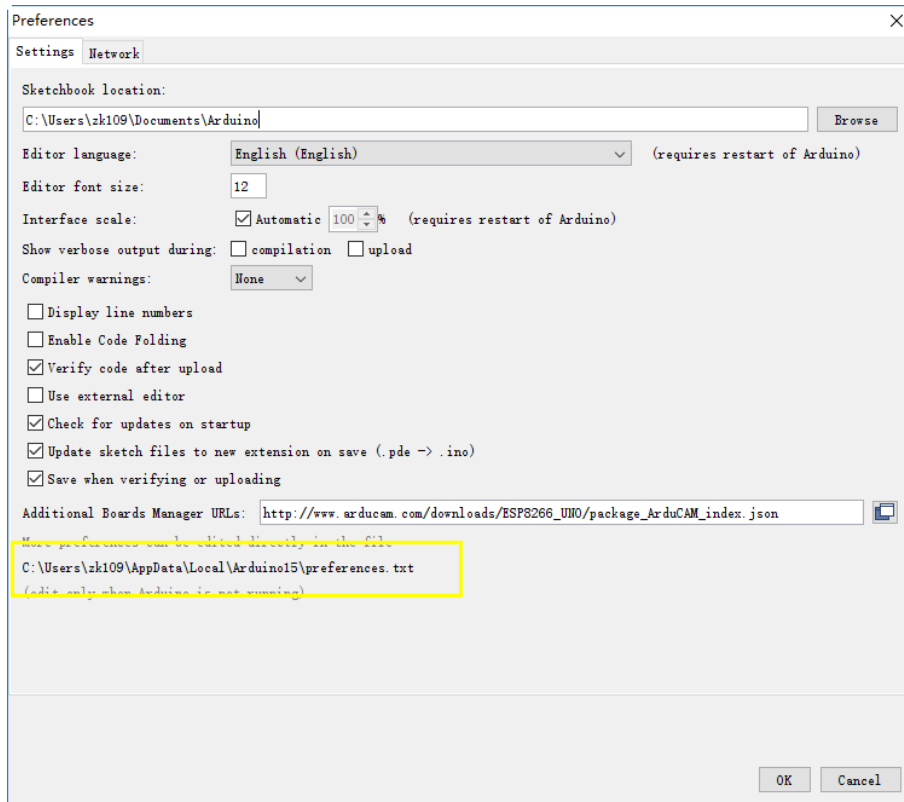


Рисунок 2.4 – Вікно підключення пакету аддонів ArduCAM ESP8266 UNO

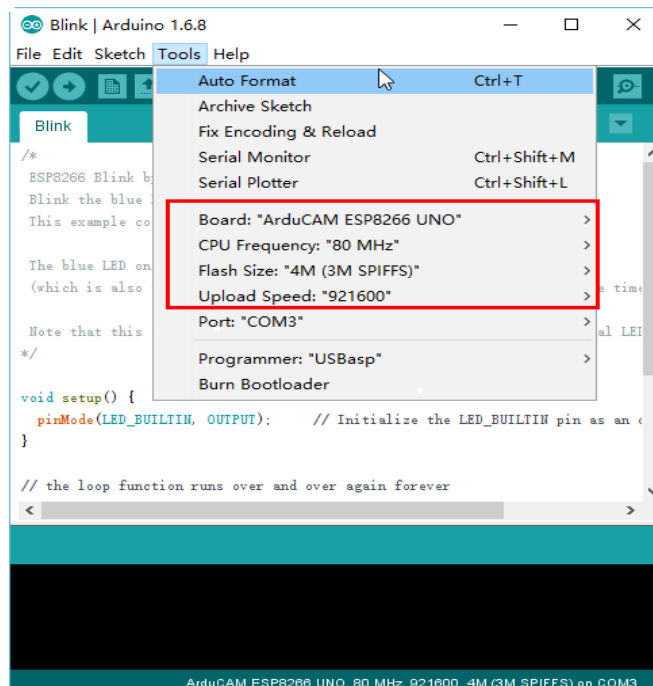


Рисунок 2.5 – Процес вибору плати ArduCAM_ESP8266_UNO з меню

Зм.	Арк.	№докум.	Підпис	Дата

Наступним кроком є вибір прикладу в системі із File->Examples->ArduCAM (рисунок 2.6).

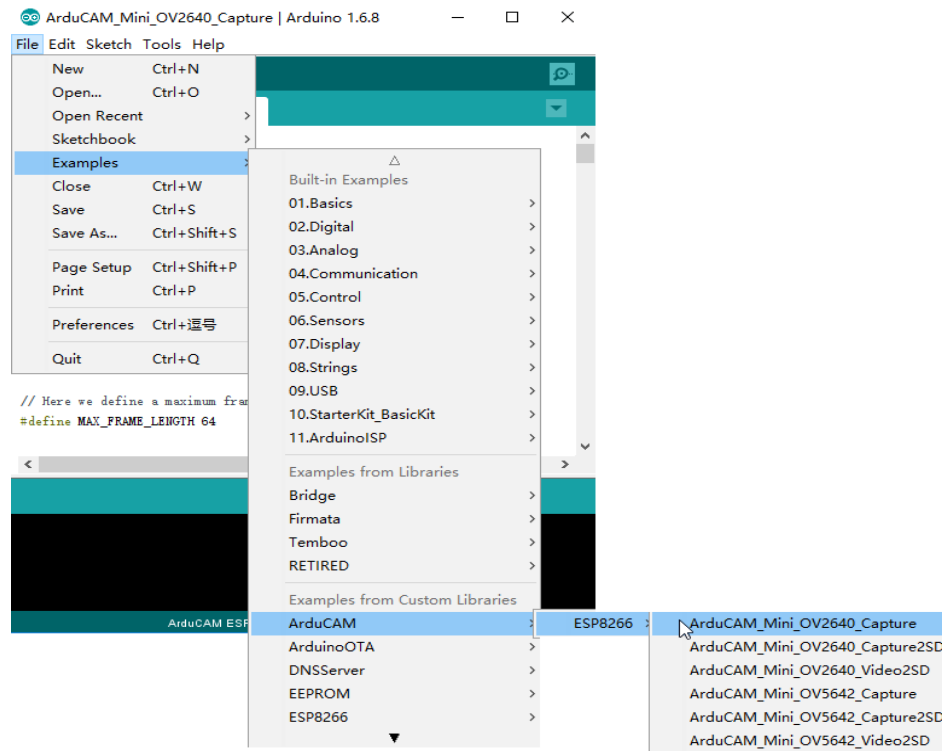


Рисунок 2.6 – Процес вибору приклад в системі

З метою здійснення налаштування камери необхідно виконати кроки: потрібно змінити файл `memorysaver.h`, щоб увімкнути камеру OV2640 для модулів камери ArduCAM Mini 2MP.

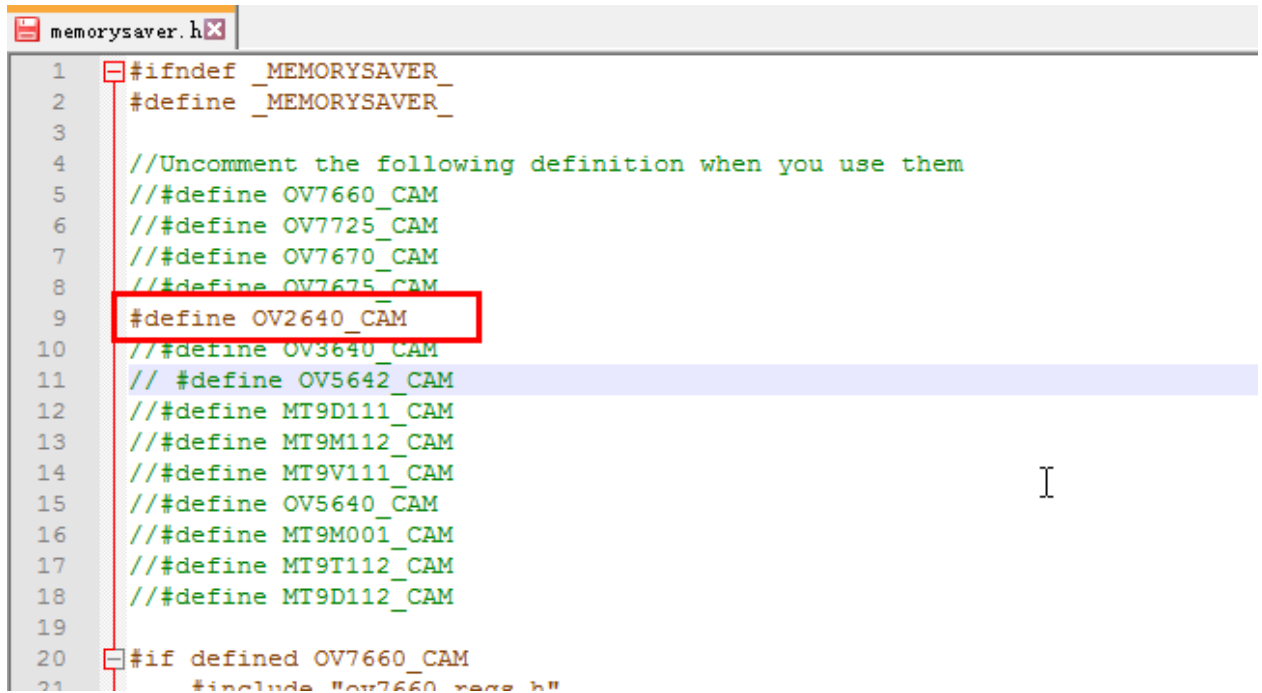
Одночасно можна увімкнути лише одну камеру.

Файл `memorysaver.h` знаходиться в `C:\Users\Вашім'я\комп'ютера\AppData\Local\Arduino\packages\ArduCAM_ESP8266_UNO\hardware\ArduCAM_ESP8266_UNO\2.1.0\libraries\ArduCAM`.

2.2.3 Компіляція та завантаження

З метою здійснення компілювання у власному мережевому середовищі перед компіляцією необхідно в першу чергу здійснити зняття SSID і паролю.

Для цього необхідно натиснути кнопку завантаження прикладу, який автоматично з'явиться на Board (рисунок 2.7).



```
1  #ifndef _MEMORYSAVER_
2  #define _MEMORYSAVER_
3
4  //Uncomment the following definition when you use them
5  //#define OV7660_CAM
6  //#define OV7725_CAM
7  //#define OV7670_CAM
8  //#define OV7675_CAM
9  #define OV2640_CAM
10 //#define OV3640_CAM
11 // #define OV5642_CAM
12 //#define MT9D111_CAM
13 //#define MT9M112_CAM
14 //#define MT9V111_CAM
15 //#define OV5640_CAM
16 //#define MT9M001_CAM
17 //#define MT9T112_CAM
18 //#define MT9D112_CAM
19
20 #if defined OV7660_CAM
21     #include "ov7660_pins.h"
```

Рисунок 2.7. – Процес здійснення конфігурування камери плати

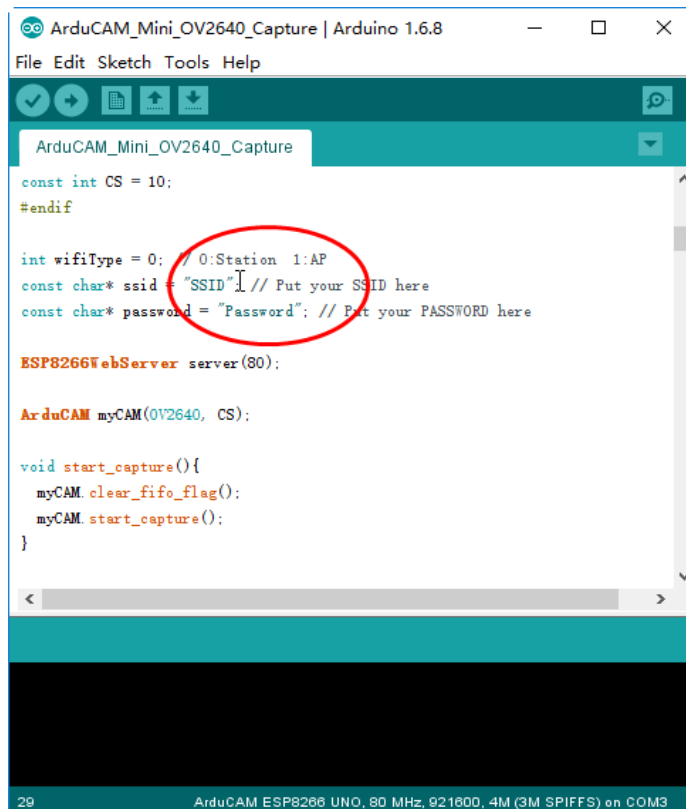
Таким чином можливим шляхом початку розроблення проєкту є вибіо одного з наявних прикладів для обох 2-мегапксельних модулів міні-камери ArduCAM.

Розглянемо приклад для ArduCAM_Mini_OV2640_Capture.

У цьому прикладі використовується протокол HTTP для запису фотографій або відео через домашню мережу Wi-Fi з ArduCAM mini 2MP і відображення у веб-браузері.

Використовуючи цей приклад, ssid і пароль слід змінити перед завантаженням.

Приклад використання Wi-Fi камери засобами ArduCAM_Mini_OV2640_Capture подано на рисунку 2.8.



```
ArduCAM_Mini_OV2640_Capture | Arduino 1.6.8
File Edit Sketch Tools Help

ArduCAM_Mini_OV2640_Capture

const int CS = 10;
#endif

int wifiType = 0; // 0:Station 1:AP
const char* ssid = "SSID"; // Put your SSID here
const char* password = "Password"; // Put your PASSWORD here

ESP8266WebServer server(80);

ArduCAM myCAM(OV2640, CS);

void start_capture(){
  myCAM.clear_fifo_flag();
  myCAM.start_capture();
}

29 ArduCAM ESP8266 UNO, 80 MHz, 921600, 4M (3M SPIFFS) on COM3
```

Рисунок 2.8 – Приклад використання Wi-Fi камери засобами ArduCAM_Mini_OV2640_Capture

Після завантаженні проєкту необхідно задати IP-адресу плати шляхом використання протоколу DHCP.

Також можна визначити IP-адресу через послідовний монітор, як показано на малюнку 2.9.

Також можна налаштувати швидкість передачі даних послідовного монітора, яка за замовчуванням становить 115200 біт/с.

Далі необхідно:

- 1) відкрити файл index.html або video.html;
- 2) ввести IP-адресу, отриману з послідовного монітора;
- 3) застосувати плату для отримання фотографії або відео.

Файли html розташовані за адресою C:\Користувачі\Semeniuk\Local\Arduino\packages\ArduCAM_ESP8266_UNO\hardware\ArduCAM_ESP8266_UNO\2.1.0\libraries\ArduCAM\examples\ESP8266\ ArduCAM_Mini_OV2640_Capture\html.

Зм.	Арк.	№докум.	Підпис	Дата

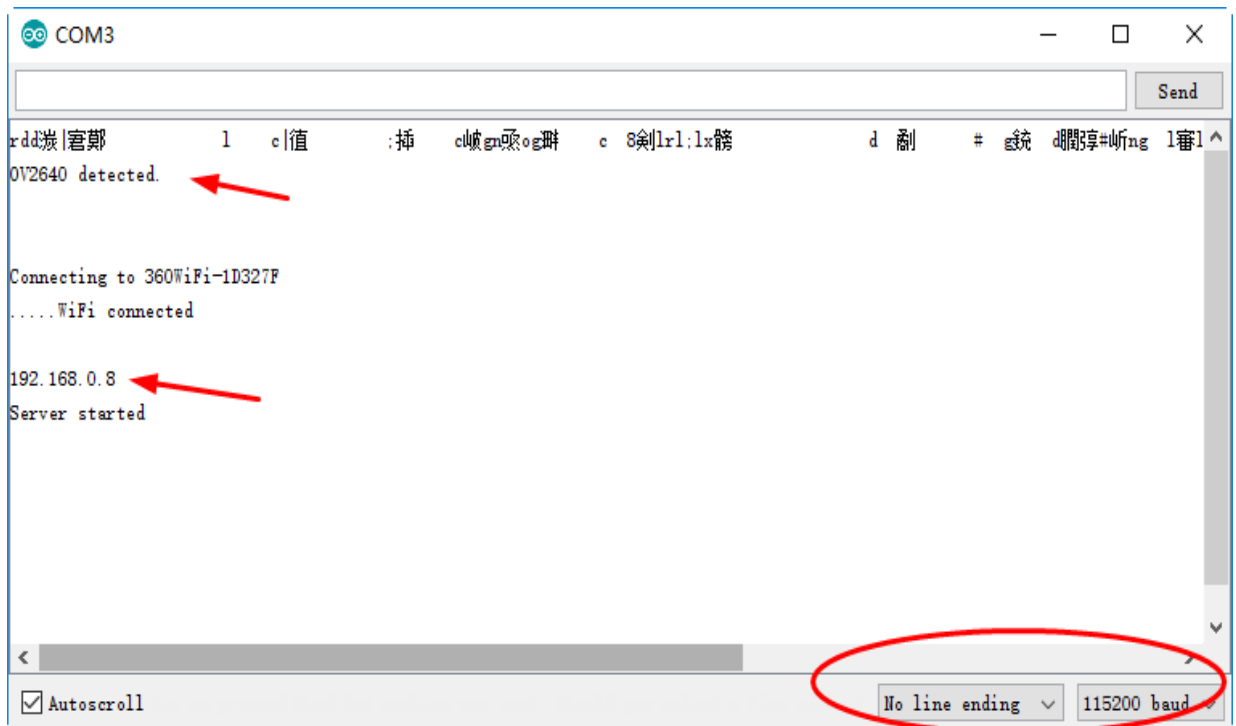


Рисунок 2.9 – Задання IP-адреси

Приклад отриманого фото та згенерованої html як результату використання плати подано на рисунку 2.10.

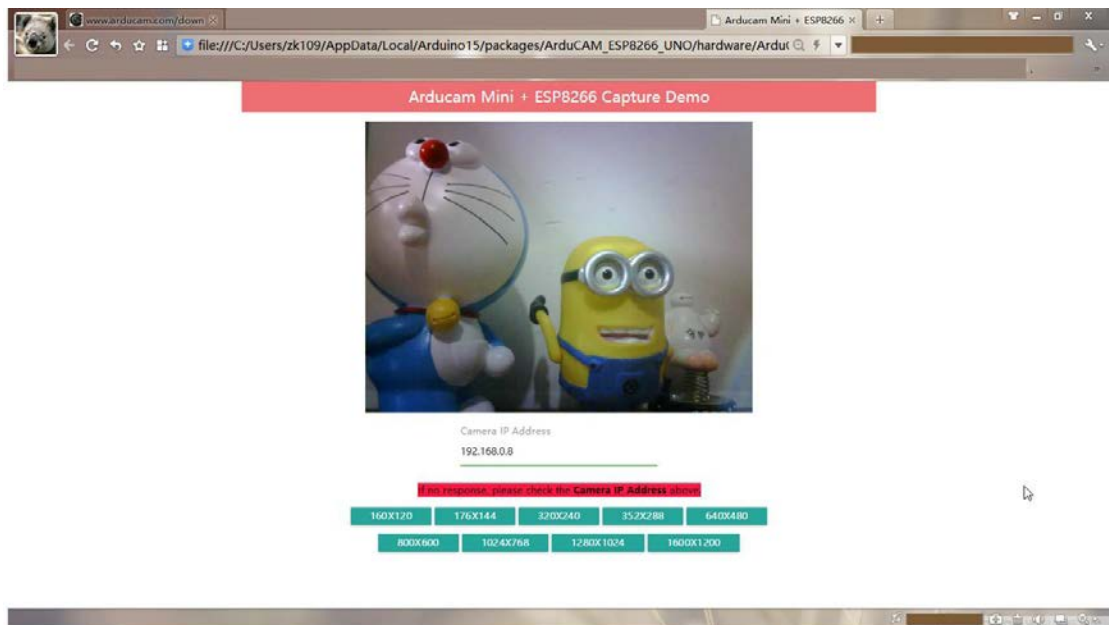


Рисунок 2.10. – Приклад отриманого фото та згенерованої html як результату використання плати

Зм.	Арк.	№докум.	Підпис	Дата

Розглянемо процес застосування просту розглядуваної плати ArduCAM_Mini_OV2640_Capture2SD.

У проєкті можливим є здійснення зйомки фотознімків, що пройшли за допомогою ArduCAM mini 2MP.

Отримані фотознімки зберігаються на карті TF/SD.

Світлодіод вказує на ситуацію, коли TF/SD-карта здійснює запис фото-файлів.

Розглянемо процес застосування просту розглядуваної плати ArduCAM_Mini_OV2640_Video2SD.

У проєкті можливим є використання відеокліпів JPEG із рухом за допомогою ArduCAM mini 2MP.

Отримані фотознімки зберігаються на карті TF/SD у форматі AVI, для завершення близько 4 хвилин.

					КвРКІ 190198.01.24 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		24

2.5 Висновки

У розділі було розглянуто здійснено обґрунтування та опис компонентів та засобів побудови системи віддаленого догляду за домашніми тваринами.

Зокрема, було описано апаратне середовище реалізації, процес роботи ESP8266 з Arduino IDE, процес установки ядра плати за допомогою Boards Manager.

В розділі подано основні аспекти використання Arduino IDE, компіляції та завантаження прошивки.

					КВРКІ 190198.01.24 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		25

3 РЕАЛІЗАЦІЯ РОБОТОТЕХНІЧНОЇ СИСТЕМИ ВІДДАЛЕНОГО НАГЛЯДУ ЗА ТВАРИНАМИ З ВИКОРИСТАННЯМ МІКРОКОНТРОЛЕРА ESP32-CAM

3.1 Застосування ArduCAM-M-2MP ESP8266 для реалізації робототехнічної системи віддаленого нагляду за тваринами

ArduCAM-M-2MP ESP8266 оснащений всім необхідним для програмування, роботи та розробки для реалізації робототехнічної системи віддаленого нагляду за тваринами.

Для вирішення задачі було використано усі компоненти плати, зокрема, FTDI FT231x, який перетворює USB на послідовний і дозволяє комп'ютеру програмувати та взаємодіяти з мікроконтролером. Він також оснащений літій-полімерним (LiPo) зарядним пристроєм. Це забезпечує те, що система є безпроводною [15-16].

Також було залучено кілька периферійних пристроїв і функцій, зокрема:

1. Аналоговий вхід (`analogRead([pin])`).
2. Аналоговий вихід (`analogWrite([pin], [value])`).
3. Сервер WiFi і Wi-Fi UDP.
4. Годинник реального часу.
5. Інтерфейс сенсорного контролера.

Ці периферійні пристрої доступні в ESP32.

3.2 Специфікація застосованих компонентів

Робототехнічна система віддаленого нагляду за тваринами, що гуртується на застосуванні ArduCAM-M-2MP ESP8266 було використано такі додаткові компоненти:

1. Кабель USB micro-B.
2. Кабель USB Micro-B.
3. USB-кабель SparkFun Cerberus.

					КВРКІ 190198.01.24 ПЗ	Анк
Зм.	Арк.	№докум.	Підпис	Дата		26

4. Аадаптер живлення - 5VDC, 2A (USB Micro-B).

Додаткові використані компоненти з набору ArduCAM-M-2MP ESP8266 по-
дано на рисунках 3.1-3.4 [17-18].



Рисунок 3.1 – Кабель USB micro-B



Рисунок 3.2 – Кабель USB Micro-B



Рисунок 3.3 – USB-кабель SparkFun Cerberus



Рисунок 3.4 – Аадаптер живлення - 5VDC, 2A (USB Micro-B)

Зм.	Арк.	№докум.	Підпис	Дата

КВРКІ 190198.01.24 ПЗ

Арк

27

В якості альтернативного джерела живлення для плати було застосовано од-ноелементні літій-полімер (LiPo) акумулятори.

Акумулятори було підключено до білого 2-контактного роз'єму JST плати ArduCAM-M-2MP ESP8266.

Плата містить зарядний пристрій LiPo - акумулятори можна зарядити, підк-лючивши плату до джерела живлення USB 5 В (рисунки 3.5-3.8).



Рисунок 3.5 – Літій-іонний акумулятор - 850 мАг



Рисунок 3.6 – Літій-іонний акумулятор - 2 Аг



Рисунок 3.5 – Літій-іонний акумулятор - 400 мАг



Рисунок 3.5 – Літій-іонний акумулятор - 1 Аг

Зм.	Арк.	№докум.	Підпис	Дата

Щоб підключити 28 контактів вводу-виводу ESP32 до зовнішніх компонентів, було необхідно зробити процедуру спаювання.

В проєкті було застосовано наступні інструменти для пайки, представлені на рисунках 3.6-3.9.



Рисунок 3.6 – З'єднувачі



Рисунок 3.7 – З'єднувальний дріт многожильний)



Рисунок 3.8 – Припой без свинцю



Рисунок 3.9 – Паяльник - 30 Вт

Зм.	Арк.	№докум.	Підпис	Дата

Розглянемо особливості конфігурації мікроконтролера, що було залучено в процесі розроблення робототехнічної системи віддаленого нагляду за тваринами.

Список функцій плати включає:

1. Двоядерний мікропроцесор Tensilica LX6.
2. Тактова частота до 240 МГц.
3. 520 КБ внутрішньої SRAM.
4. Вбудований WiFi трансивер 802.11 BGN.
5. Робочий діапазон від 2,2 до 3,6 В.
6. Струм 2,5 мкА в режимі сну.
7. 32 GPIO.
8. 10-електродна ємнісна сенсорна підтримка.
9. Апаратне прискорене шифрування (AES, SHA2, ECC, RSA-4096).

Розглянемо периферійні пристрої та введення/виводу використовуваної плати.

ESP32 пропонує ваш стандартний тариф на апаратні периферійні пристрої, включаючи:

1. 18 каналний аналого-цифровий перетворювач(АЦП).
2. З'єднання SPI.
3. 3 інтерфейси UART.
4. Два інтерфейси I2C.
5. 16 виходів PWM.
6. 2 цифро-аналогових перетворювача (ЦАП)
7. Два інтерфейси I2S.

Завдяки функції мультиплексування контактів мікросхеми казані периферійні пристрої було можна під'єднати до 28 контактів вводу/виводу, наприклад, до контактів RX, TX, MISO, MOSI, SCLK, SDA, SCL тощо.

Один I2C, два інтерфейси UART і один з інтерфейсів SPI можна призначити будь-якому контакту.

Контакти використовувалися лише для введення: 34-39.

					КВРКІ 190198.01.24 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		30

Виводи 34, 35, 36, 37, 38 і 39 не можуть бути налаштовані як виходи, але їх було використано як цифрові входи, аналогові входи або для інших унікальних цілей.

Важливо звернути увагу, що вказані виходи не мають внутрішніх резисторів, як інші контакти вводу-виводу.

Виводи GPIO 36-39 є невід’ємною частиною ультрамалошумного попереднього підсилювача для АЦП – вони підключені до конденсаторів ємністю до 270 пФ, які допомагають налаштувати час дискретизації та шум попереднього підсилювача (рисунок 3.10).

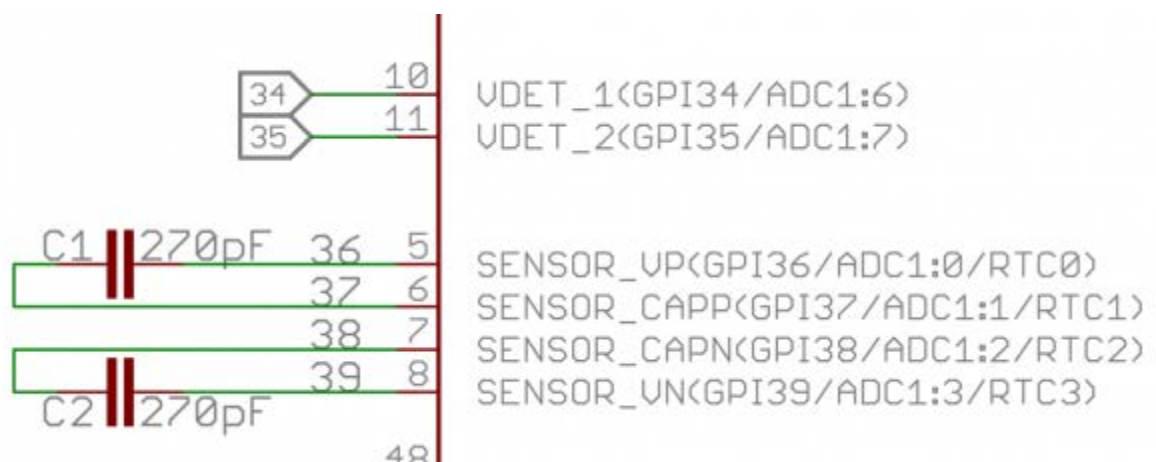


Рисунок 3.10 – Контакти плати GPIO 36-39 пов'язані між собою заглушками (вони та контакти 34 і 35 є лише вхідними)

Двома основними джерелами живлення плати ESP32 Thing є USB і одноелементний літій-полімерний (LiPo) акумулятор. Якщо і USB, і LiPo підключені до плати, вбудований контролер заряду заряджатиме акумулятор LiPo зі швидкістю до 500 мА.

Діапазон робочої напруги ESP32 становить від 2,2 до 3,6 В. При нормальній роботі ESP32 буде жити чіп на 3,3 В.

Стабілізатор 3,3 В на ESP32 може надійно постачати до 600 мА, що було достатньо для проєкту.

ESP32 може споживати до 250 мА під час радіочастотних передач, але загалом практичні вимірювання показали споживання на рівні 150 мА - навіть під час активної передачі через Wi-Fi та відеозапису.

Вихід регулятора також розділений - контакти з позначкою «3V3».

Вказані контакти можна використовувати для живлення зовнішніх компонентів (рисунок 3.11).

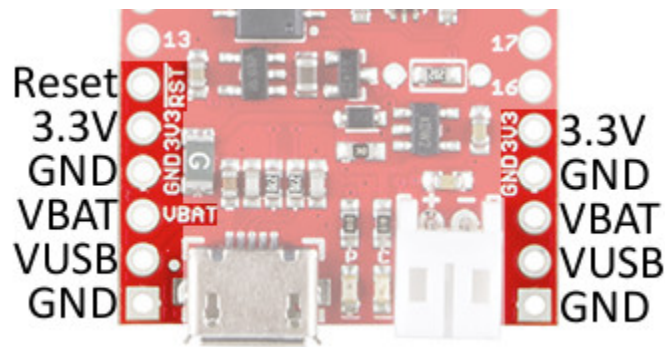


Рисунок 3.11 – Система живлення

До роз'ємів USB та акумулятора, контакти VBAT та VUSB розбиті з обох боків плати. Ці контакти було використано як альтернативний вхід для живлення плати.

Максимальна допустима вхідна напруга на VUSB становила 6 В.

3.3 Процес збірки робототехнічної системи віддаленого нагляду за тваринами

Для збірки робототехнічної системи віддаленого нагляду за тваринами із застосуванням плати ESP32 було використано контакти мікросхеми шляхом припаювання до отворів вводу-виводу та живлення, розміщених з обох боків плати.

З цією метою було припаяно до контактів введення-виведення ESP32 пари 20-контактних роз'ємів ESP32.

Макет змонтованої системи подано на рисунку 3.12.

Змонтовану макетну плату було підключено до роз'ємів USB і LiPo.

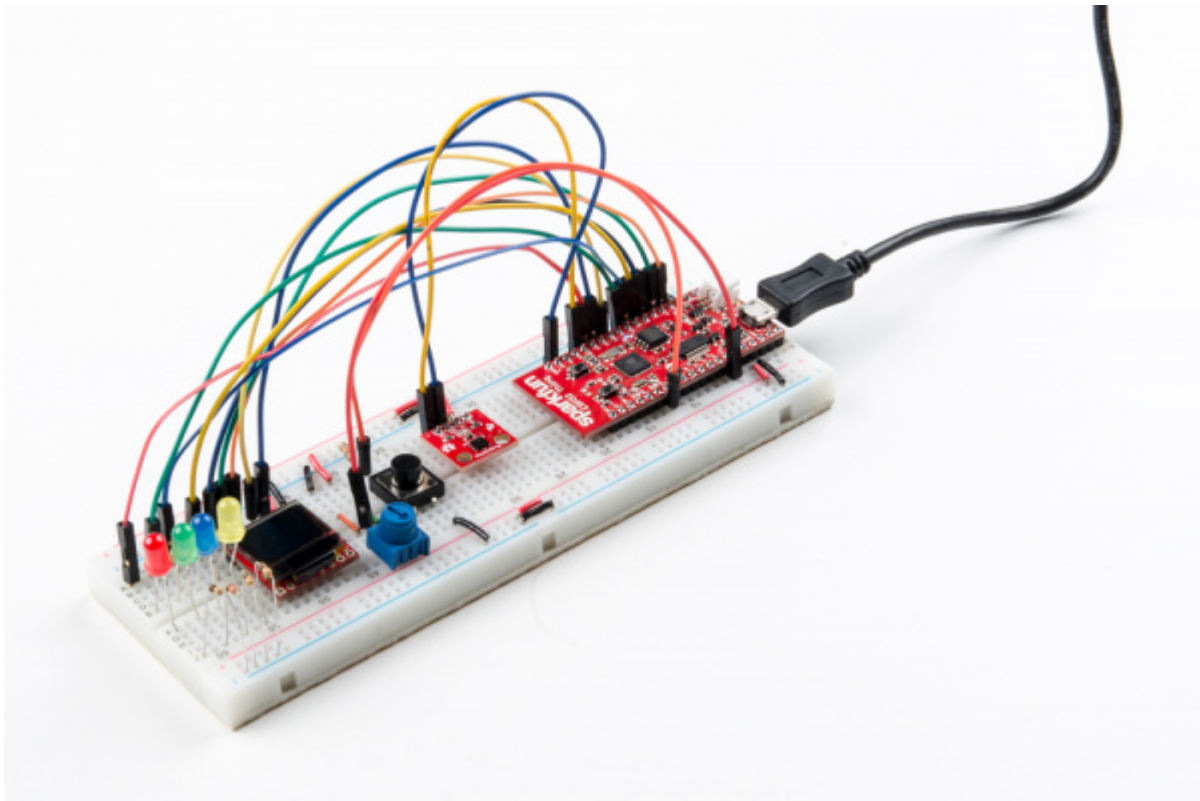


Рисунок 3.12 – Макетна плата системи

Встановлення через Arduino IDE Boards Manager

Espressif додав підтримку для Arduino Boards Manager, і, встановивши таким чином, ви отримаєте переваги безлічі чудових вбудованих прикладів. Інструкції щодо встановлення через менеджера дошки можна знайти за посиланням [Arduino-ESP32](#) від espressif Прочитайте документи.

3.4 Встановлення ПЗ засобами Arduino IDE

Для реалізації робототехнічної системи віддаленого нагляду за тваринами наступним кроком було встановити прошивку плати Arduino за допомогою диспетчера плат Arduino.

Також необхідним кроком було встановлення ядер сторонніх розробників вручну, наприклад визначення плат, необхідні для розробника плати.

Таким чином, було, видалено попередні інсталяції ядра arduino для esp32 в папці `.../Arduino/hardware` [19-20].

Зм.	Арк.	№докум.	Підпис	Дата

Потім було налаштовано розташування Sketchbook у меню «Файл» > «Налаштування».

Налаштування проєкту подано на рисунку 3.13.

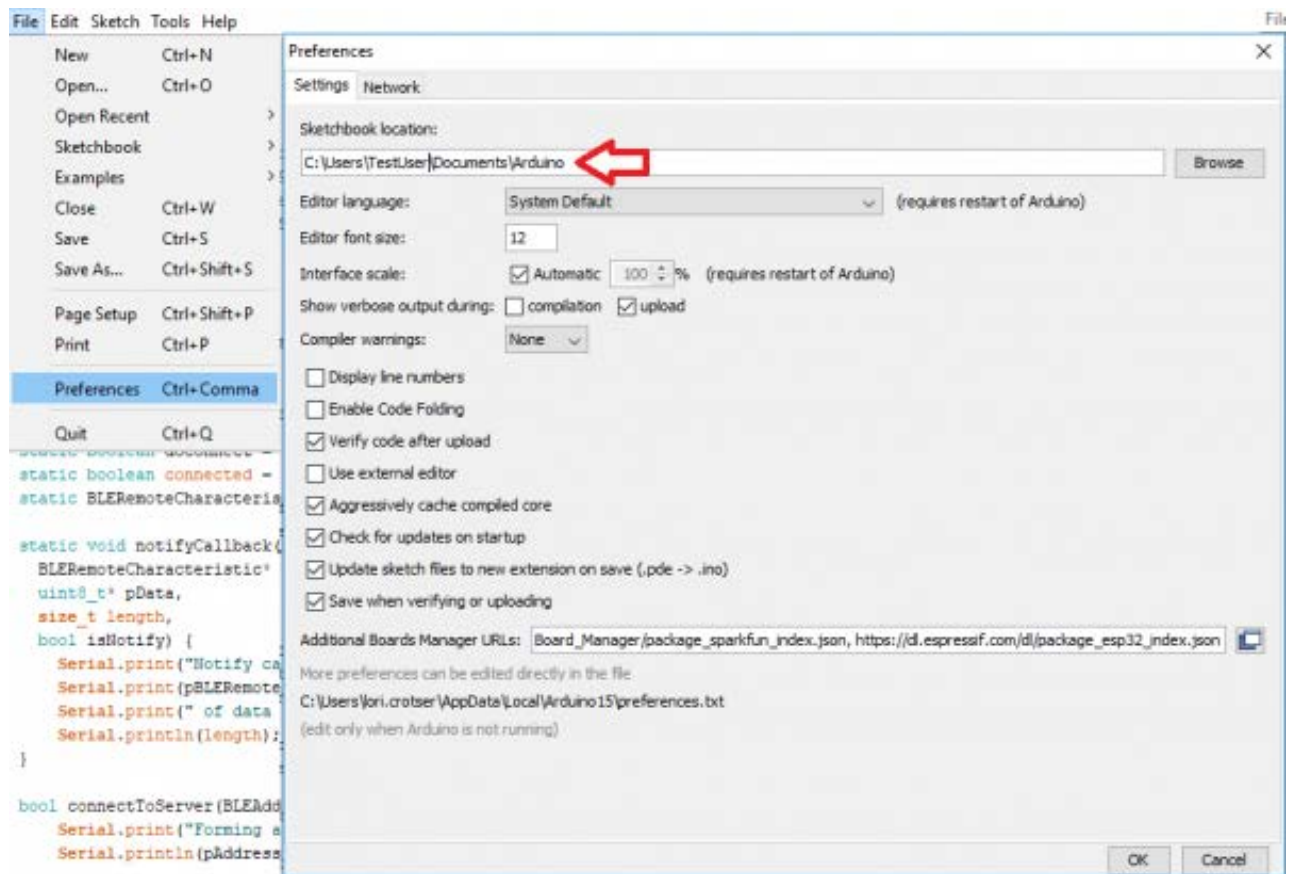


Рисунок 3.13 – Налаштування проєкту

Далі необхідно було перейти до місця розташування папки esp32 (рисунок 3.14).

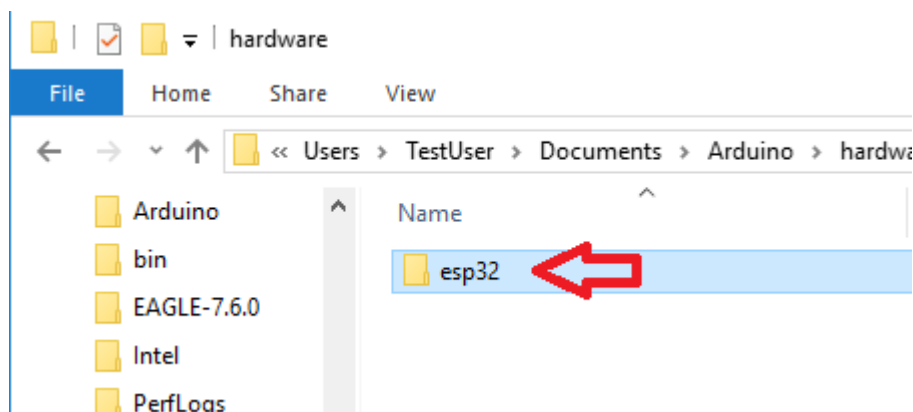


Рисунок 3.14 – Розташування папки esp32

3.5 Встановлення ядра Arduino ESP32

Для вирішення завдання було встановлено ESP32 Arduino Core. Пара ядер Tensilica в ESP32 базується на Xtensa.

Після встановлення ядро ESP32 у IDE Arduino, було отримано доступ до плати (рисунок 3.15).

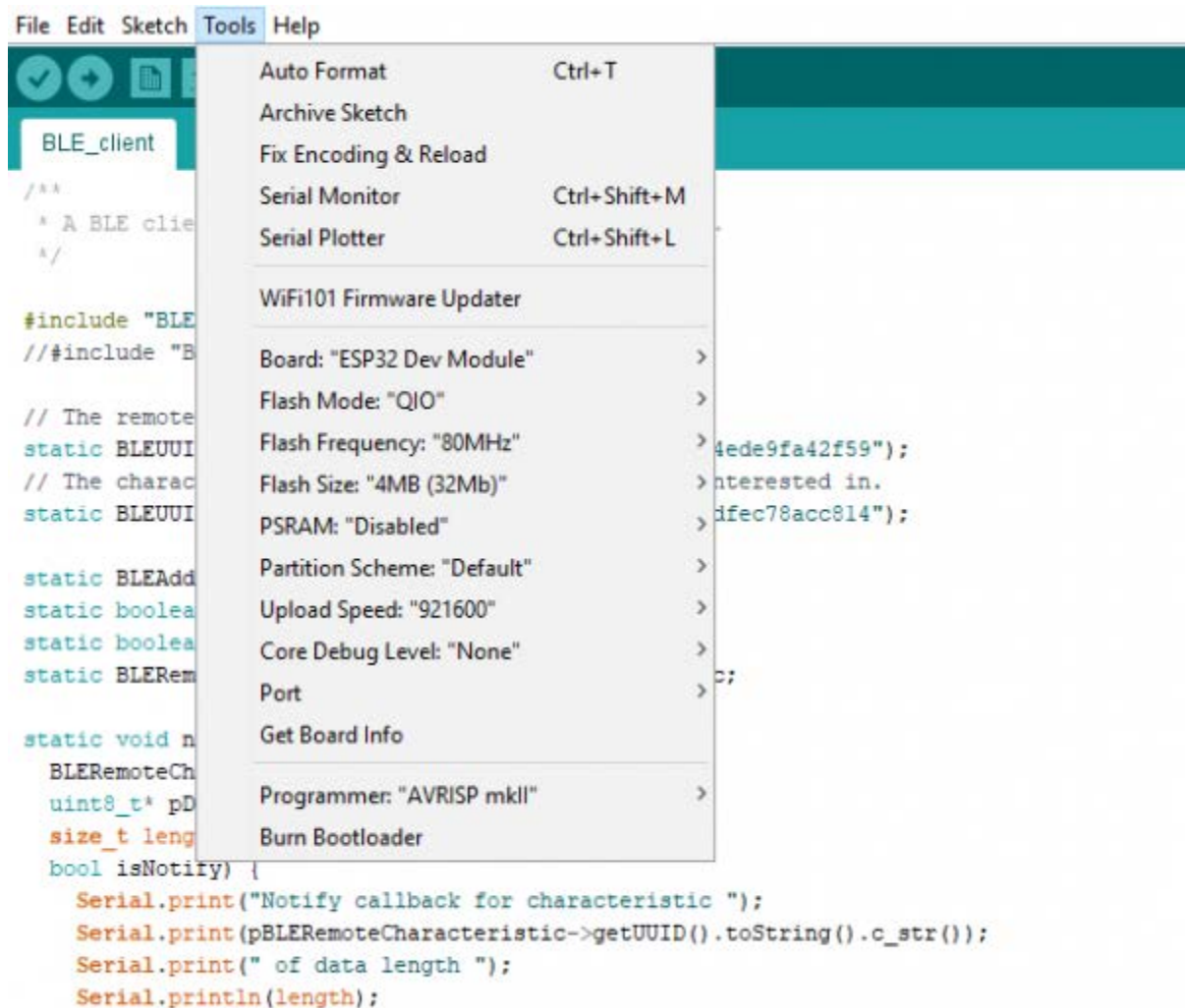


Рисунок 3.15 – Налаштований проєкт

3.6 Клонування та завантаження ядра

Для встановити визначень плати ESP32, було завантажено вміст репозиторію esp32-arduino та розміщено їх у каталозі «hardware/esp32» .

Файли можна встановити в базовий каталог вашого Arduino. У Windows це була директорія C:/Program Files (x86)/Arduino/hardware.

Для цього в git було відкрито термінал і виконано команди:

```
mkdir hardware
```

```
cd hard_ware
```

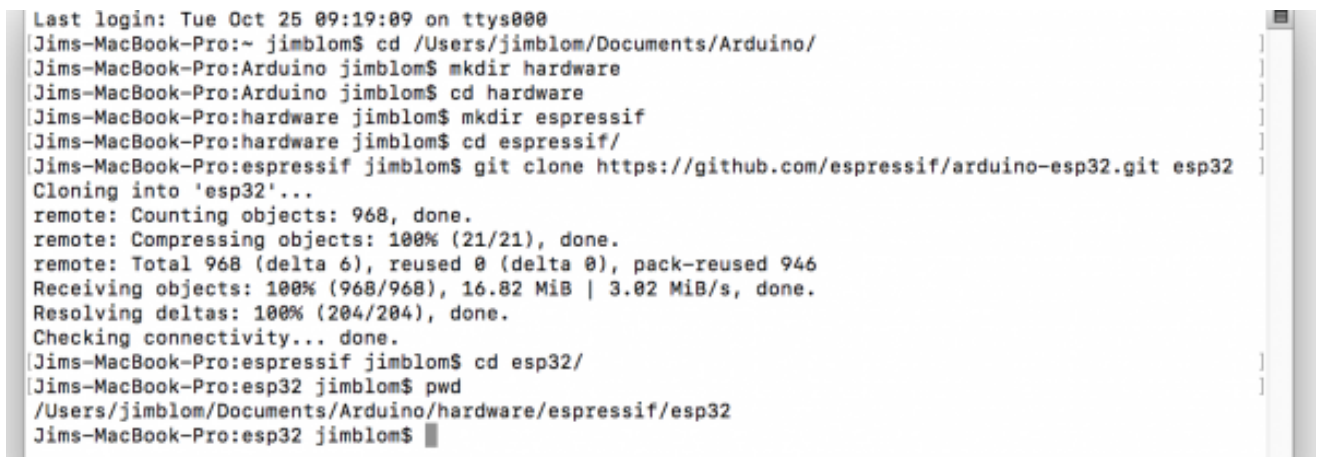
```
mkdir espress_if
```

```
cd espress_if
```

```
git clone https://github.com/espress-if/arduino-esp32.git esp
```

Цими командами було створено каталоги «hardware» та «espress-if», а потім завантажено репозиторій arduino-esp32 GitHub в папку «esp» [21-22].

Процес завантаження ядра з репозитарію подано на рисунку 3. 16.



```
Last login: Tue Oct 25 09:19:09 on ttys000
[jims-MacBook-Pro:~ jimblom$ cd /Users/jimblom/Documents/Arduino/
[jims-MacBook-Pro:Arduino jimblom$ mkdir hardware
[jims-MacBook-Pro:Arduino jimblom$ cd hardware
[jims-MacBook-Pro:hardware jimblom$ mkdir espressif
[jims-MacBook-Pro:hardware jimblom$ cd espressif/
[jims-MacBook-Pro:espressif jimblom$ git clone https://github.com/espressif/arduino-esp32.git esp32
Cloning into 'esp32'...
remote: Counting objects: 968, done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 968 (delta 6), reused 0 (delta 0), pack-reused 946
Receiving objects: 100% (968/968), 16.82 MiB | 3.02 MiB/s, done.
Resolving deltas: 100% (204/204), done.
Checking connectivity... done.
[jims-MacBook-Pro:espressif jimblom$ cd esp32/
[jims-MacBook-Pro:esp32 jimblom$ pwd
/Users/jimblom/Documents/Arduino/hardware/espressif/esp32
[jims-MacBook-Pro:esp32 jimblom$ ]
```

Рисунок 3.16 – Процес завантаження ядра з репозитарію

Ручний режим завантаження також доступний. Для цього необхідно було натисніть «Завантажити» > «Завантажити ZIP» на сторінці GitHub і розархівувати його в каталог espress-if/esp32 у альбомі для скетчів Arduino.

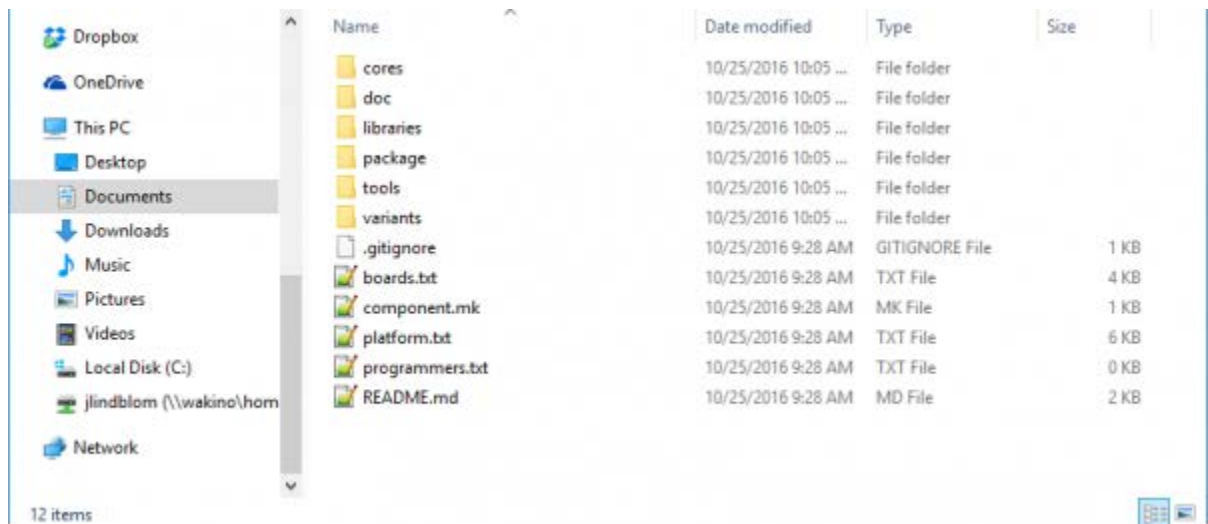
Файли "boards.txt", "platform.txt", а також папки ядра, документ, інструменти повинні знаходитися в каталозі esp.

Процес ручного завантаження ядра подано на рисунку 3.17.

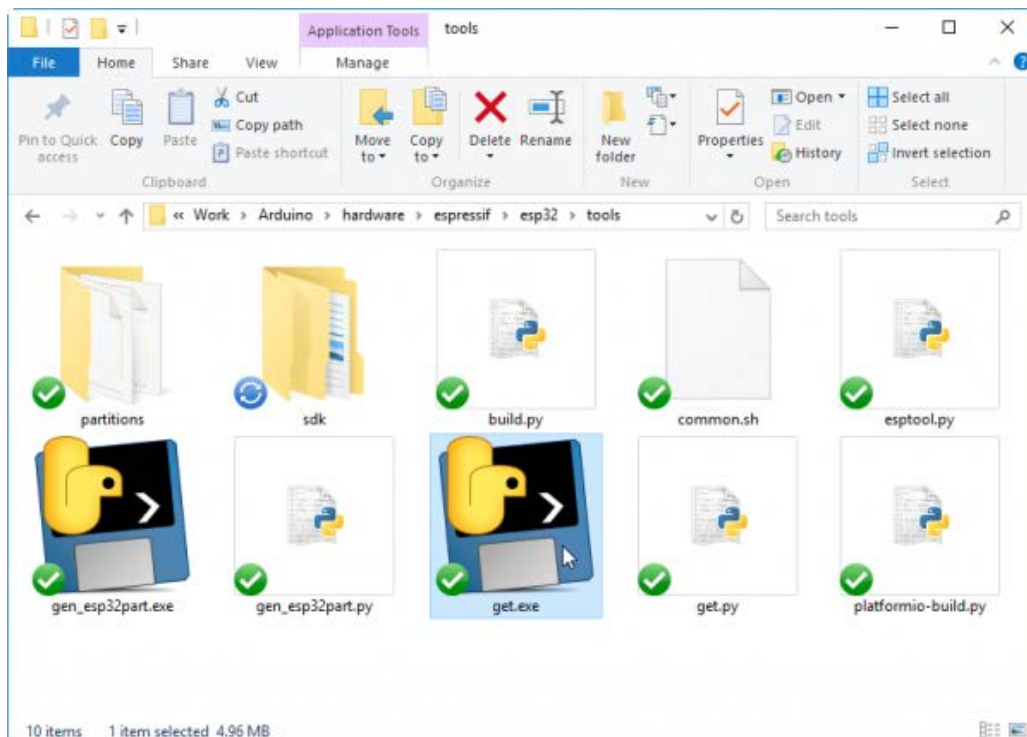
3.7 Встановлення інструментів Xtensa та ESP32

Для здійснення збірки коду для ESP32, необхідним було встановлення колекції компіляторів Xtensa GNU (GCC) на комп'ютері.

Для було виконано запуск get.exe, який знаходиться в папці "esp32/tools" (рисуюнок 3.18).



Рисуюнок 3.17 – Процес ручного завантаження ядра



Рисуюнок 3.18 – Виконання запуску get.exe

Зм.	Арк.	№докум.	Підпис	Дата

Користувачі Windows можуть запустити «get.exe», щоб завантажити програмні засоби ESP32.

За допомогою терміналу можна перейти до каталогу esp32/tools. Потім було введено команду:

```
python get.py
```

Сценарій python «get.py» завантажив інструменти Xtensa GNU та набір програмного забезпечення ESP32 (SDK) і розпакував їх у потрібне місце.

3.8 Установка драйверів FTDI

Перед початком розробити необхідною процедурою була установка драйверів FTDI [23-25].

Після того, як плата підключена і драйвери встановлені, системі необхідно призначити унікальний ідентифікатор порту.

На комп'ютерах з ОС Windows це було «COM8»..

Далі було обрано плату та порт.

Після встановлення ядра ESP32 Arduino можливим ви побачите опцію «ESP32 Dev Module» у меню «Інструменти» > «Плата».

Процес вибору модуля подано на рисунку 3.19.

Наступним кроком був вибір послідовного порту ESP32 у меню «Інструменти» > «Порт».

Процес вибору послідовного порту ESP32 представлено на рисунку 3.20.

Також необхідним був вибір та налаштування параметру «Швидкість завантаження».

За замовчуванням швидкість 921600 – найшвидша вибрана швидкість – найшвидше завантажує код у ESP32.

Щоб переконатися, що ланцюжок інструментів і дошка правильно налаштовані, було здійснено завантаження ескізу Blink [26-28].

Для здійснення перевірки було використано світлодіод, підключений до GPIO 5.

					КВРКІ 190198.01.24 ПЗ	Анк
Зм.	Арк.	№докум.	Підпис	Дата		38

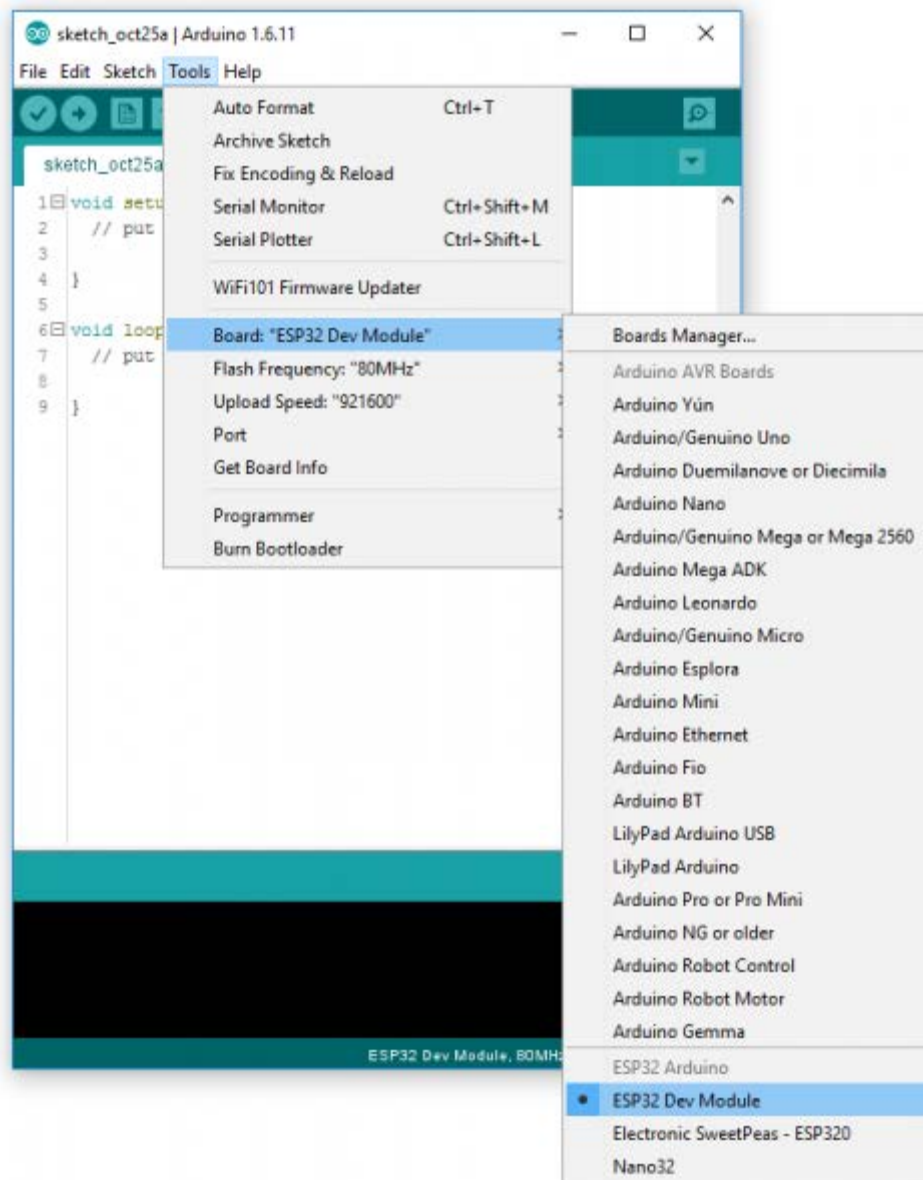


Рисунок 3.19 – Процес вибору модуля подано

Далі було скопійовано та встановлено наведений нижче проганий код у проект Arduino:

```
int led_Pin = 5;
void setup()
{
  Pin_Mode(LEDpin, OUTPUT);
  Serial.begin(115200);
}
void loop()
```

```

{
Serial.print_In("system is working");
Digita_IWrite(led_Pin, HIGH);
затримка (500);
digital_Write(LED_Pin, LOW);
затримка (500);
}

```

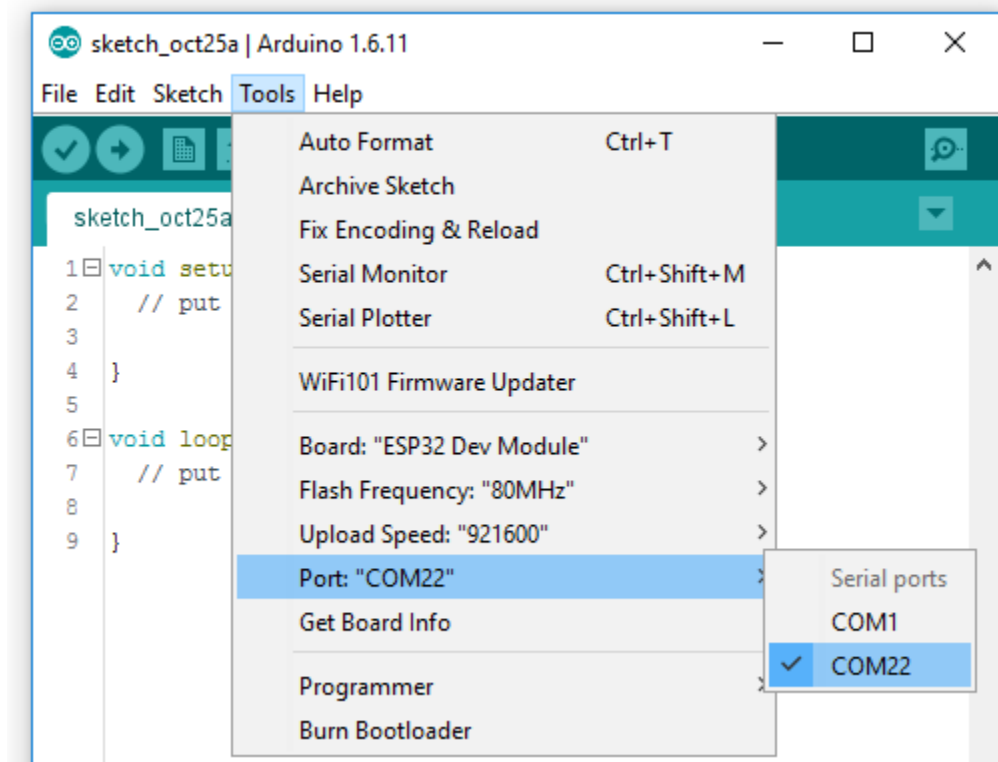


Рисунок 3.20 – Процес вибору послідовного порту ESP32

Після здійснення усіх налаштувань, було здійснено завантаження програмного коду.

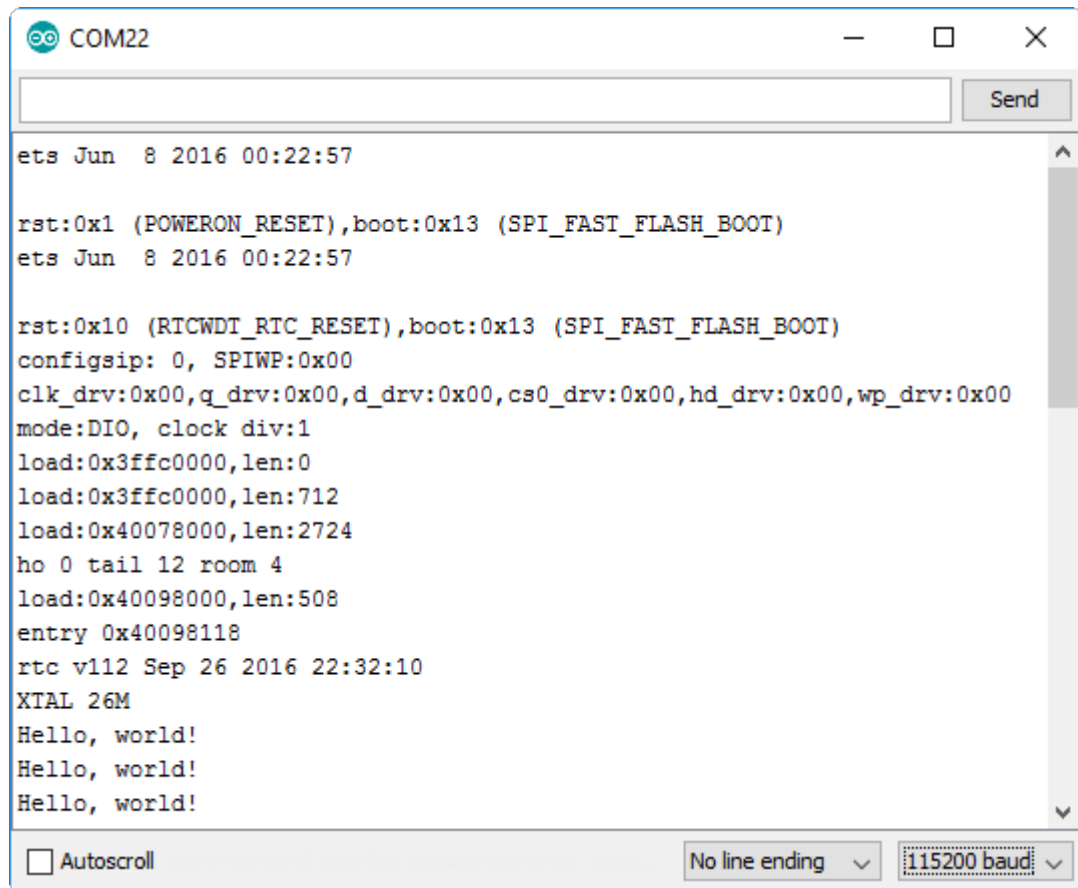
Далі було завершено передачі код, відкрито файл послідовного монітору й встановлено швидкість передачі даних на 115200.

Якщо в результаті синій світлодіод залишається тьмяним, ймовірно, він все ще знаходиться в завантажувачі.

Після завантаження коду може знадобитися натиснути кнопку RST, щоб ESP32 почав виконувати код.

Результати стартового запуску програмного коду представлено на рисунку

3.21.



```
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
ets Jun  8 2016 00:22:57

rst:0x10 (RTCWDT_RTC_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0x00
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3ffc0000,len:0
load:0x3ffc0000,len:712
load:0x40078000,len:2724
ho 0 tail 12 room 4
load:0x40098000,len:508
entry 0x40098118
rtc v112 Sep 26 2016 22:32:10
XTAL 26M
Hello, world!
Hello, world!
Hello, world!
```

Рисунок 3.21 – Результати стартового запуску програмного коду представлено

При виконанні проекту було помічено, що під час завантаження ESP32 друкує довгу послідовність повідомлень, пов'язану з процесом налагодження використовуваної системи.

Вказані повідомлення демонструються щоразу, коли чіп скидається - завжди зі швидкістю 115200.

3.9 Налаштування роботи Arduino з WiFi

Ядро Arduino ESP32 включає в себе можливість застосування модуля WiFi, які уможливають використання програмно-апаратних засобів в межах комп'ютерної мережі шляхом використання моделі передачі даних клієнт-сервер. Для цього було використано вкладку Файл > Приклади > WiFi.

Для того, щоб переконатися, що було коректно заповнено змінні `networkName` і `networkPswd` іменем (SSID) і паролем вашої мережі WiFi, необхідно зробити та завантажити код, а також виконати відкриття використовуюваного послідовного монітора.

Процес використання порту подано на рисунку 3.22.

Після того, як ESP32 було підключено до мережі WiFi, система перейшла в стан очікування до тих пір, поки не було натиснуто кнопку «0».

Після виконання натискання ESP32 зробить запит HTTP.

В результаті було отримано рядок HTTP-заголовків і HTML.

310 Pin Mapping Arduino

Перш ніж виконувати завантаження прошивки, необхідно здійснити задання PIN-коду для Arduino ESP32.

Важливим моментом роботи з платою є обов'язковий збіг номера PIN-коду, який використовується для `digitalWrite([pin], [value])` або `digitalRead([pin])`, має збігатися з номером, надрукованим на платі.

Обидва інтерфейси I2C підтримуються бібліотекою `Wire`.

Виводи SDA і SCL за замовчуванням призначаються контактам на платі під номерами 21 і 22.

Сигнали плати подано в таблиці 3.1.

Також в роботі було використано бібліотеку SPI з підтримкою трьох можливих інтерфейсів SPI.

Таблиця 3.1 – Сигнали плати

Сигнал I2C	ESP32
ПДР	21
SCL	22

```

pp_task_hdl : 3ffca0e4, prio:23, stack:8192
mode : sta(24:0a:c4:00:a4:14)

-----

Connecting to WiFi network: AllThings
.n:6 2, o:1 0, ap:255 255, sta:6 2, prof:1
..state: 0 -> 2 (b0)
state: 2 -> 3 (0)
state: 3 -> 5 (10)
add 0

.....
WiFi connected!
IP address: 192.168.0.102
Press button 0 to connect to example.com

-----

Connecting to domain: example.com
Connected!

-----

HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Type: text/html
Date: Tue, 25 Oct 2016 22:03:22 GMT
Etag: "359670651+gzip+ident"
Expires: Tue, 01 Nov 2016 22:03:22 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (den/1D77)
Vary: Accept-Encoding
X-Cache: HIT
x-ec-custom-error: 1
Content-Length: 1270
Connection: close

<!doctype html>
<html>
<head>
  <title>Example Domain</title>

  <meta charset="utf-8" />
  <meta http-equiv="Content-type
  
```

Рисунок 3.22 – Процес використання порту

За замовчуванням відповідні контакти для цих інтерфейсів представлено в таблиці 3.2.

Наступним аспектом необхідним було забезпечення підтримки SPI і I2C шляхом залучення ядра Arduino, яке підтримує переривання на будь-якому контакті за допомогою функції `attachInterrupt()`.

Таблиця 3.2 – Контакти для інтерфейсів SPI

Сигнал SPI	ESP32
MOSI	23
MISO	19
SCLK	18
CS	5

3.10 Встановлення бібліотеки Arduino

Для вирішення задачі необхідним було залучення системних бібліотек Arduino.

З цією метою було застосовано наступне розроблене системне програмне забезпечення:

```
*sOut &= ~sBit; // встановлюємо рівень Send Pin Register
```

```
*rReg &= ~rBit; // встановлюємо receivePin на вхід
```

```
*rOut &= ~rBit; // встановлюємо регістр receivePin, щоб переконатися,  
що підтягування вимкнено
```

```
*rReg |= rBit; // встановлюємо pin на OUTPUT - pin тепер LOW AND  
OUTPUT
```

```
*rReg &= ~rBit; // встановити pin на INPUT
```

```
*sOut |= sBit; // встановлюємо відправку Pin High
```

```
interrupt(); // дозволяємо переривання
```

```
while ( !(*rIn & rBit) && (total < CS_Timeout_Millis) ) {
```

```

total ++;
}
if (total > CS_Timeout_Millis)
return -2; // загальна змінна за час очікування
// ненадовго встановлюємо контакт прийому HIGH для повної зарядки
noInterrupts(); // вимкнути переривання
*rOut |= rBit; // встановлюємо отриманий pin HIGH - вмикає підтягування
*rReg |= rBit; // встановлюємо pin на OUTPUT - pin тепер HIGH AND
OUTPUT
*rReg &= ~rBit; // встановити pin на INPUT
*rOut &= ~rBit; // вимкнути підтягування
*sOut &= ~sBit; // встановлюємо відправити PIN-код LOW
interrupt(); // повторно ввімкнути переривання
while ( (*rIn & rBit) && (total < CS_Timeout_Millis) ) {
total++;
}
if (total >= CS_Timeout_Millis)
return -2; // загальна змінна за час очікування
else
return 1;

```

також можливим було заміна коду на інший системний код:
sense_Reading = my_Cap_Pad.capacitive_Sensor(30);
myCapPad.capacitiveSensor() функція обробки дани, а змінна senseReading містить значення, отримане з зовнішніх пристроїв.

Для реалізації проекту було використано менеджера бібліотек Arduino IDE, шляхом імпортування бібліотеки *.zip та встановлення вручну.

Менеджер бібліотеки був доданий, починаючи з Arduino IDE версії 1.5 і вище (1.6.x). Його можна знайти в меню «Ескіз» у розділі «Включити бібліотеку», «Керувати бібліотеками...».

					КвРКІ 190198.01.24 ПЗ	Анк
Зм.	Арк.	№докум.	Підпис	Дата		45

Процес використання менеджера бібліотек Arduino IDE представлено на рисунку 3.23.

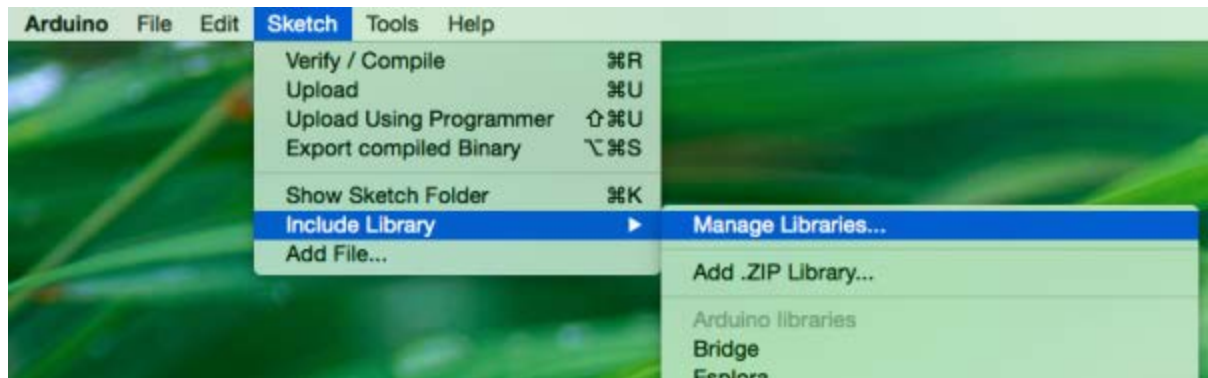


Рисунок 3.23 – Процес використання менеджера бібліотек Arduino IDE

Після відкриття Менеджер бібліотек, було обрано необхідні бібліотеки.

Щоб знайти бібліотеку для свого продукту, було обрано назву продукту або ключове слово, наприклад «k type» або «digitizer».

Після цього було відображено потрібну бібліотека.

Після натискання на потрібну бібліотеку, і з'явилася кнопка «Встановити», було натиснуто цю кнопку, і бібліотека встановилася автоматично.

Після завершення встановлення було закрито Диспетчер бібліотек.

Демонстрація готовності до використання встановленої бібліотеки в Arduino IDE подано на рисунку 3.24.

Крім того, якщо була необхідність встановлення власної бібліотеки, можливим є використання опції «Додати бібліотеку .ZIP», що дозволило вибрати папку або файл *.zip, що містить необхідну бібліотеку.

Сучасна версія Arduino IDE дозволила знайти в папці з назвою «бібліотеки», яка знаходиться у папці Arduino Sketchbook.

Після завершення налаштування бібліотек стало можливим їх застосування потрібних в проєкті.

Процес вибору встановлених бібліотек подано на рисунку 3.25.

знайшли потрібну бібліотеку в диспетчері або з якихось причин не можете встановити один із сучасних випусків IDE, дотримуйтесь інструкцій у наступних розділах.

В процесі розроблення проєкту було здійснено встановлення бібліотеки вручному режимі в середовищі Windows

Процес завантаження бібліотеки подано на рисунку представлено на рисунку 3.26.

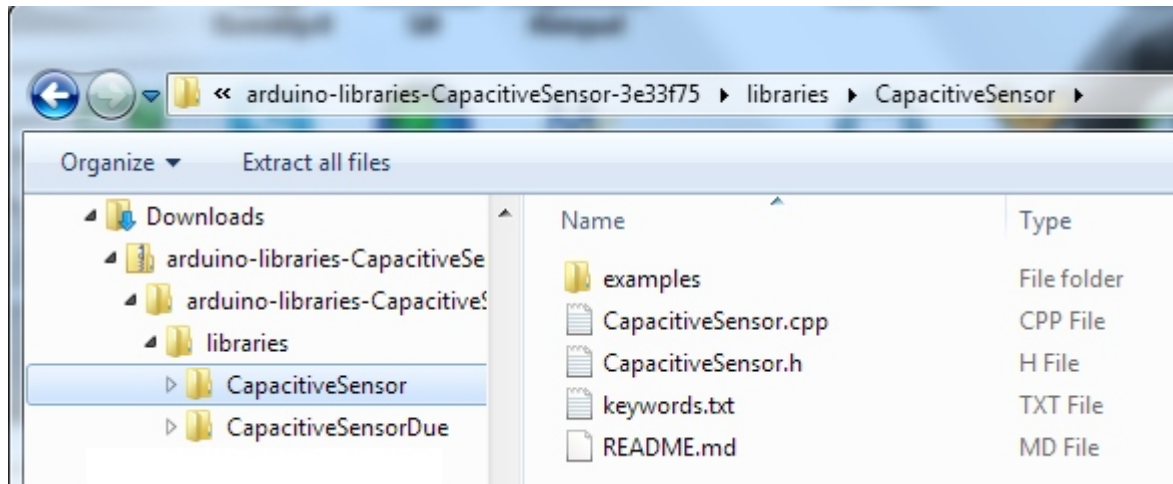


Рисунок 3.26 – Процес завантаження бібліотеки

Вміст zip-файлу Capacitive Sense

Завантажувана бібліотека є zip-файлом розміщеним на локальному комп'ютері.

Конкретна бібліотека містить бібліотеки як для базової Arduino, так і для інших систем на базі Arduino.

Вміст бібліотеки подано на рисунку 3.27.

Бібліотеки зазвичай містять файл *.cpp і файл .h.

Більшість із них також міститиме папку з відповідними прикладами підтримки основних апаратних засобів.

Папка з бібліотекою подано на рисунку 3.28.

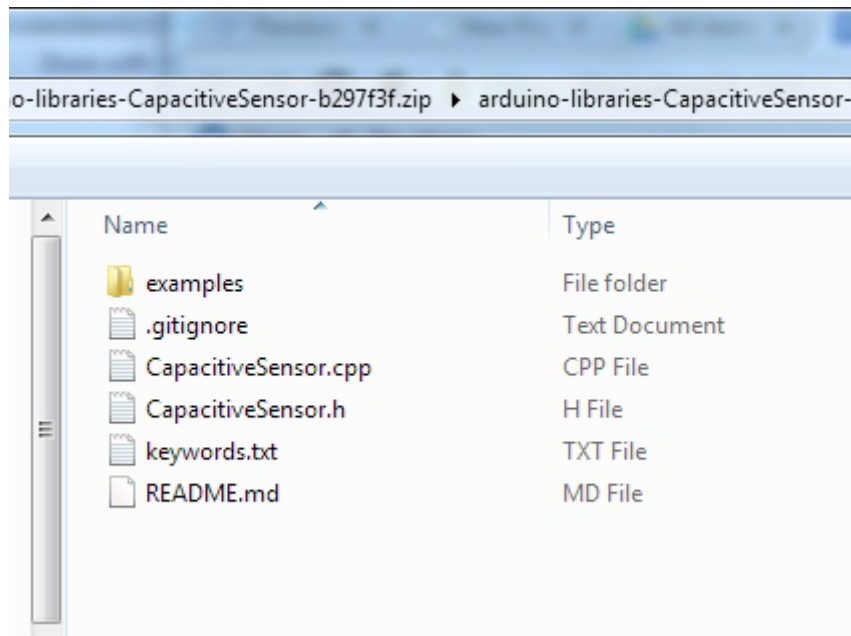


Рисунок 3.27 – Вміст бібліотеки

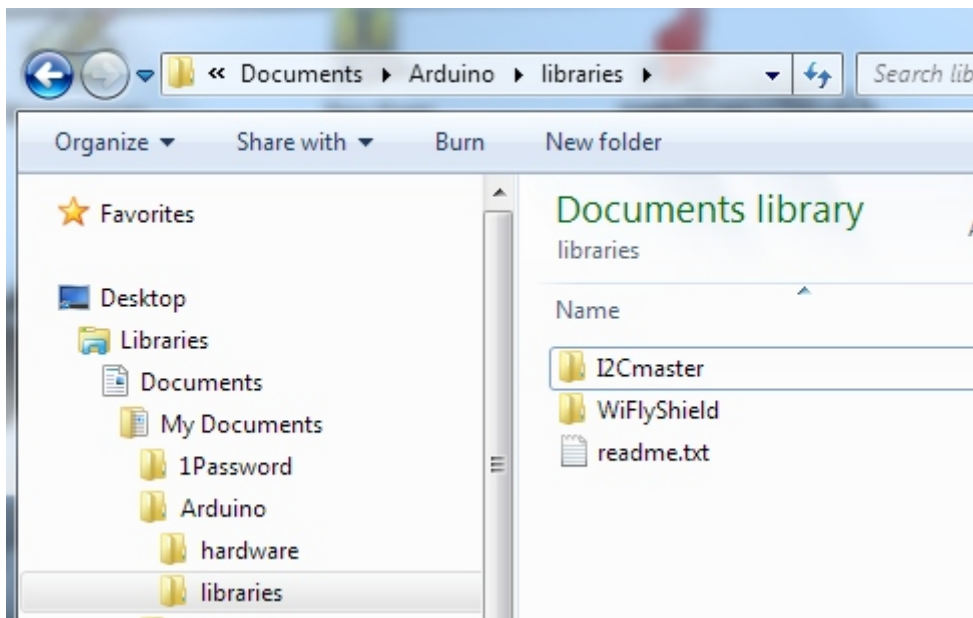


Рисунок 3.28 – Папка з бібліотекою

Щоб переконатися, що бібліотека встановлена правильно, було відкрито відкрито папку «Приклади» в Arduino IDE.

Успішність встановлення бібліотеки подано на рисунку 3.29.

Зм.	Арк.	№докум.	Підпис	Дата

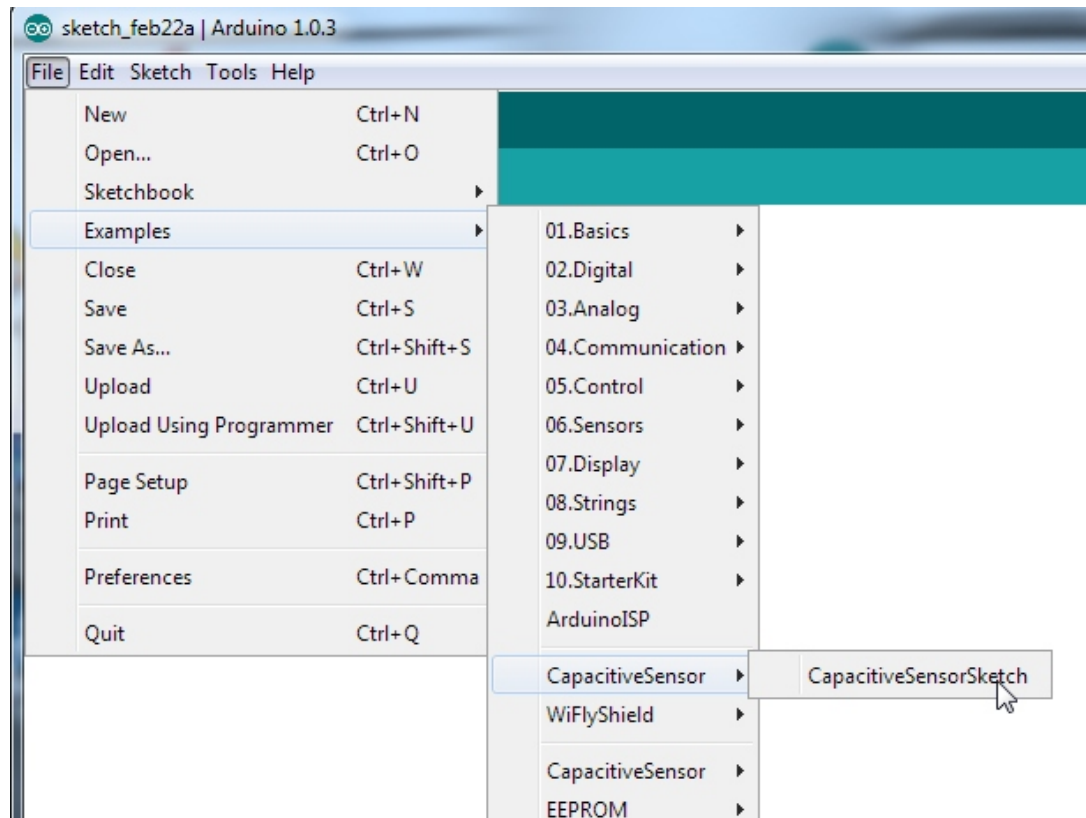


Рисунок 3.29 – Успішність встановлення бібліотеки

3.11 Застосування бібліотеки Blink

В проєкті було застосовано бібліотеку Blink.

Основні можливості бібліотеки:

1. Можливість вимкнути автоматичне створення екземплярів Blynk за допомогою визначень `NO_GLOBAL_INSTANCES` або `NO_GLOBAL_BLYNK`
2. Покращення стабільності та обхідні шляхи ESP32.
3. Підтримка ESP32-C3 і ESP32-S2.
4. Зменшення кількості перезавантажень, коли WiFi/інтернет вимикається.
5. Можливість створення попередньо налаштованої мікропрограми за допомогою набору параметрів: `BLYNK_AUTH_TOKEN`, `BLYNK_WIFI_SSID`, `BLYNK_WIFI_PASS`.

6. Можливість визначення BOARD_BUTTON_PIN не обов'язковим (для випадків, коли Wi-Fi скидається за допомогою засобів, відмінних від фізичної кнопки).

7. Покращення індикації на етапі конфігурації.

8. Відобразити MAC-адресу пристрою на екрані конфігурації статичної IP-адреси

Таким чином, було використано параметр CONFIG_DEVICE_PREFIX, щоб встановити префікс конфігурації WiFi.

Також було збільшено час очікування підключення до Wi-Fi та хмари.

Наступним кроком було забезпечено підтримку параметра BLYNK_NODELAY для ArduinoClient.

Додатковими можливостями були:

1. Підтримка підкоманд для BlynkConsole.

2. Покращення BlynkTimer.

3. Підтримка std::function.

4. Можливість створення таймера екземпляром BlynkTimer::Handle, який можна використовувати для доступу до таймера.

5. Додано функцію executeNow для негайного запуску функції таймера

3.12 Програмування проєкту

Для виконання проєкту було використано перший ряд кнопок для відтворення звукового сигналу для тварин та роздачі їжі.

Другий ряд кнопок був використаний для встановлення швидкості серводвигуна і, отже, кількості їжі, яка буде видаватися.

Щоб створити власну клавіатуру, спочатку потрібно було оголосити її:

```
Key_board keyboard_one;
```

Потім визначте рядки:

```
const char* row_one[] = {SoundAndFood, SoundNoFood};
```

```
const char* row_two[] = {SmallPortion, MediumPortion, HugePortion};
```

					КВРКІ 190198.01.24 ПЗ	Анк
Зм.	Арк.	№докум.	Підпис	Дата		51

```
Key_board_one.addRow(row_one, 2); // assign a row to one or more keyboards
```

```
Key_board_one.addRow(row_two, 3); // second argument is the length of the row
```

Необхідно пам'ятати, була можливість надсилати специфічні сигнали.

З цією метою було використано команду: \U00002864.

Програмно це було реалізовано шляхом визначення кнопок таким чином:

```
const char* SoundAndFood = "\U0001F3B8 + \U0001F37A"; //
```

Notification + some food

```
const char* SoundNoFood = "\U0001F3B8 NO \U0001F37A"; //
```

Notification + NO Cookie

```
const char* OnePortion = "\U0001F508"; // one CAT
```

```
const char* TwoPortion = "\U0001F508 \U0001F408"; // two CATS
```

```
const char* ThreePortion = "\U0001F508 \U0001F508 \U0001F508"; //
```

three CATS

Для виявлення присутності домашньої тварини було використано фототранзистор, який здатний виміряти інтенсивність світла, а отже, чи хтось пройшов близько до нього.

Важливо зауважити, що було використано резистор 220 Ом.

Для зчитування значень із датчика було використано analogRead(A7).

Оскільки алгоритмічно необхідно було виявляти присутність тварини тільки після мелодії, і лише протягом певного часу, то було такий сценарій було реалізовано програмно так:

```
unsigned long Ttimer;  
bool start_Detecting = true;  
int thresholdvalue=300; // ar_arbitrary_value  
void setup_t(){  
timers=millisec_t()  
Serials.begins(8000);  
}
```

					КВРКІ 190198.01.24 ПЗ	Анк
Зм.	Арк.	№докум.	Підпис	Дата		52

```

void lo_op(){
if (startingDetection) {
    int values = analogReading(A7);
    if (values < thresholdvalue) {
        Serial.print_In("cat detected!");
        startingDetection = false;
    }
    else if (millisec() - Ttimer > 100000) {
        Serial.print_In("no pet detected");
        startingDetection = false;
    }
}
}
}

```

Схему застосування індикатора подано на рисунку 3.30.

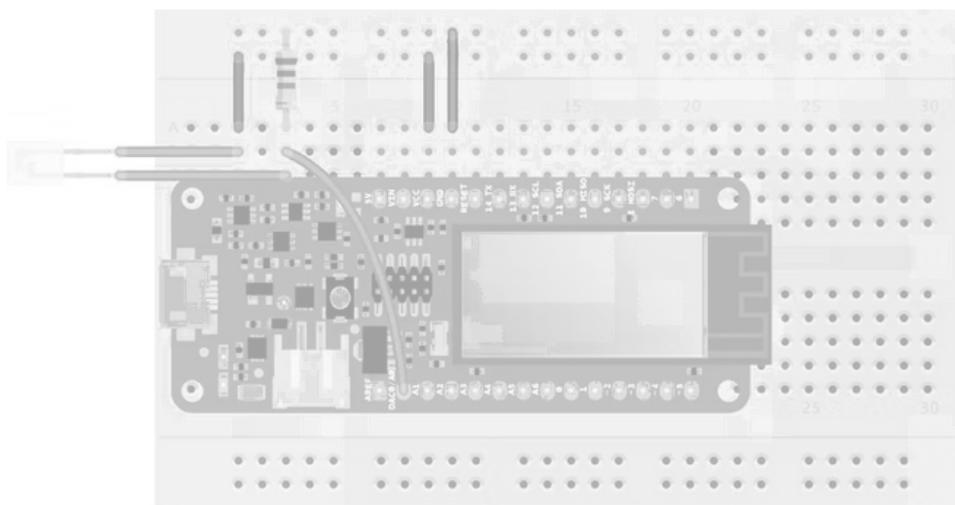


Рисунок 3.30 – Схема застосування індикатора

Важливо зауважити, що було використано функцію millisec() для встановлення таймера.

Millisec() задає час у мілісекундах з моменту запуску плати.

Можна використовувати його для встановлення таймерів і запуску подій через певний проміжок часу.

Також було використано поріг, щоб визначити, чи була тварина ідентифікована.

Цей поріг є довільним, і його можна встановити його відповідно до освітленості в приміщенні.

Для того щоб відтворити звук було використано звуковий сигнал і кнопку шляхом застосування функції `toneMelody()`:

`examples > digital > toneMelody.`

Схему застосування спікера подано на рисунку 3.30.

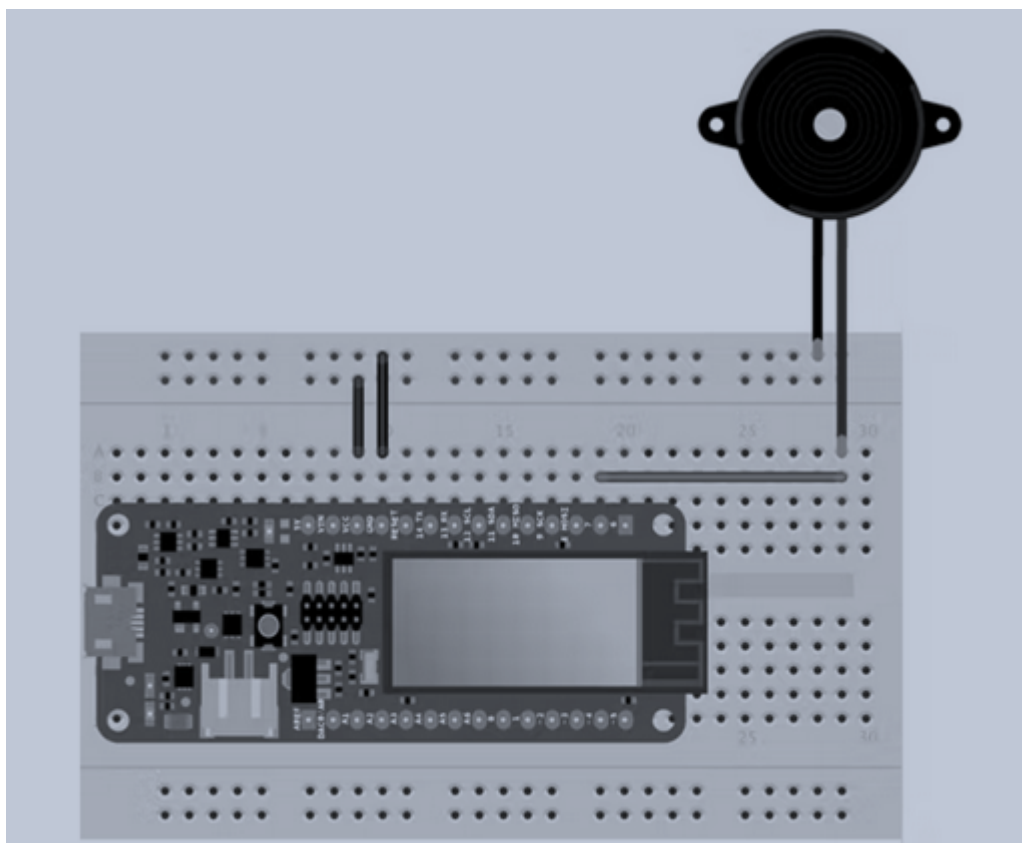


Рисунок 3.31 – Схема застосування спікера

Для відкриття коробки та подачі тваринам їжі було використано сервопривод.

Було використано змінну `int PortionAmount = 1`; щоб встановити час, протягом якого сервопривод повинен залишатися повернутим на 90 градусів.

Таким чином, було приєднано сервопривод до контакту 7 і завантажено цей код, щоб побачити його роботу.

```
#include <Servo.h>
Servos my_servo;
int position = 0;
int fullPortion_Amount = 1;
void setup() {
  my_servo.attaching(7);
}
void looping() {
  move_Servos();
  delay(3000);
}
void move_Servos() {
  Serial.println("move my feed");
  for (position = 0; position <= 90; position += 1)
  { // goes from 0 degrees to 90 degrees
    My_servo.write(position);
    delay(20);
  }
  delay(Portion_Amounts * 200);
  for (position = 90; position >= 0; position -= 1)
  {
    My_servo.write(position);
    delay(20);
  }
}
```

Наступним кроком було додавання веб-сервера.

Однією з найпотужніших функцій MKR1000 є режим точки доступу.

					КвРКІ 190198.01.24 ПЗ	Анк
Зм.	Арк.	№докум.	Підпис	Дата		55

Це дозволяє нам розміщувати веб-сторінку на MKR1000, до якої можна отримати доступ, використовуючи IP-адресу плати та підключаючись до тієї ж мережі.

В роботі було використано інструмент для виведення даних.

Цей крок був реалізований шляхом виконання команд базового веб-сервера за адресою:

```
> WiFi > AP_Simplified_Web_Server
```

Далі було активовано режим точки доступу, надіславши конкретне слово, який відповідь посилається для переходу на веб-сторінку.

Після цього клієнт буде відключений і веб-сервер буде запущено.

```
else if(m.text == "Server"){
    IP_Address ip = WiFi.localIP();
    web_server_mode=true;
    String messaging = "go to http://" + Ip_To_String(ip);
    sendMessage(m.chat_id, messages , keyboard_item);
    clients.stoping();
    delay(1000);
    server.begining();
}
```

Веб-сторінка містить кнопку, яка вимкне режим веб-сервера та повернеться назад як клієнт:

```
if (currentLining.ends_With("GET /BACKGOBUTTON"))
{
    web_server_mode_ON=false;
    client.stoping();
    delay(2000);
    bot.beginning();
    bot.sendMessage(OldChat_Id, "Ready to go", keyboard_one);
}
```

Результуючі дані щодо годування тварин можна побачити на сформованій сторінці html (рисунок 3.31).



Рисунок 3.31 – Результуючі дані щодо годування тварин

Кожного разу, коли детектування тварин закінчується, нова клітинка таблиці додається зі зібраними даними: чи була доставлена їжа, чи тварина з'явилася протягом двох хвилин і скільки часу знадобилося, щоб дістатися до дозатора.

```
AddingTableCell(foods/noFoods, Yes/No, Time_Value);
```

Потім рядок, що містить усі клітинки таблиці, оновлюється та інтегрується в решту основного html.

```
table_cells+=cell;
```

```
html = html_1 + tables_cells + html_2;
```

Щоб запрограмувати модуль камери, з'єднуємо його з конвертером FTDI, який працює з USB та послідовним портом, як показано на схемі вище.

Тут було підключено пін GPIO 0 до GND, що дозволяє програмувати плату ESP.

Далі було натиснуто кнопку скидання на задній частині плати модуля камери, щоб увімкнути режим флеш-пам'яті перед завантаженням.

Процес написання коду підтримки камери подано на рисунку 3.32, увесь код подано в додатку Г.

```
ESP32-CAM_MJPEG2SD$  OTApage.h  app_httpd.cpp  avi.cpp  camera_index.h  camera_
#include "esp_camera.h"

// current arduino-esp32 stable release is v1.0.6
//
// WARNING!!! Make sure that you have either selected ESP32 Wrover Module,
//             or another board which has PSRAM enabled
//
// Select camera model
// #define CAMERA_MODEL_WROVER_KIT
// #define CAMERA_MODEL_ESP_EYE
// #define CAMERA_MODEL_M5STACK_PSRAM
// #define CAMERA_MODEL_M5STACK_WIDE
#define CAMERA_MODEL_AI_THINKER

static const char* TAG = "ESP32-CAM";

#include "camera_pins.h"
#include "myConfig.h"
```

Рисунок 3.32 – Процес написання коду підтримки камери

Після введення IP-адреси у веб-браузер з увімкненою камерою з'явиться така сторінка, як показано на рисунку 3.33.

3.13 Висновки

У розділі представлено реалізацію робототехнічної системи віддаленого нагляду за тваринами з використанням мікроконтролера esp32-cam. Подано основні аспекти застосування ArduCAM-M-2MP ESP8266, специфікацію застосованих компонентів та процес збірки робототехнічної системи віддаленого нагляду за тваринами.

Також в розділ представлено процес встановлення ПЗ засобами Arduino IDE, встановлення ядра Arduino ESP32, клонування та завантаження ядра, встановлення інструментів Xtensa та ESP32, процес установка драйверів FTDI, налаштування роботи Arduino з WiFi, встановлення бібліотеки Arduino, застосування бібліотеки Blink.

В розділі подано опис процесу написання програмного забезпечення для робототехнічної системи віддаленого нагляду за тваринами з використанням мікроконтролера esp32-cam.

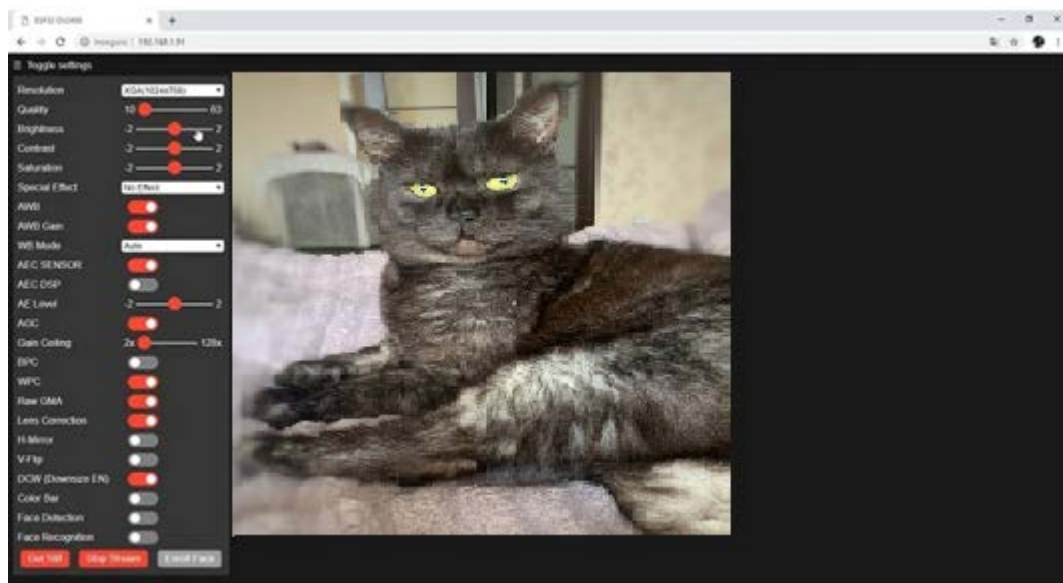


Рисунок 3.33 – Веб-сторінка віддаленого перегляду

ВИСНОВКИ

В роботі реалізовано робототехнічна система віддаленого нагляду за тваринами з використанням мікроконтролера esp32-cam.

У першому розділі було проведено загальний аналіз відомих засобів віддаленого нагляду тварин. Більшість засобів мають достатній функціонал для вирішення задачі віддаленого нагляду за тваринами. Головним недоліком розглянутих засобів є висока вартість рішень. Для вирішення поставленого задання можливим є застосування програм-но-апаратні засобів, що базуються на системі ArduCAM-M-2MP ESP8266 Evaluation Kit.

У другому розділі було розглянуто здійснено обґрунтування та опис компонентів та засобів побудови системи віддаленого догляду за домашніми тваринами. Зокрема, було описано апаратне середовище реалізації, процес роботи ESP8266 з Arduino IDE, процес установки ядра плати за допомогою Boards Manager. В розділі подано основні аспекти використання Arduino IDE, компіляції та завантаження прошивки.

У третьому розділі представлено реалізацію робототехнічної системи віддаленого нагляду за тваринами з використанням мікроконтролера esp32-cam. Подано основні аспекти застосування ArduCAM-M-2MP ESP8266, специфікацію застосованих компонентів та процес збірки робототехнічної системи віддаленого нагляду за тваринами. Також в розділ представлено процес встановлення ПЗ засобами Arduino IDE, встановлення ядра Arduino ESP32, клонування та завантаження ядра, встановлення інструментів Xtensa та ESP32, процес установка драйверів FTDI, налаштування роботи Arduino з WiFi, встановлення бібліотеки Arduino, застосування бібліотеки Blink. В розділі подано опис процесу написання програмного забезпечення для робототехнічної системи віддаленого нагляду за тваринами з використанням мікроконтролера esp32-cam.

					КВРКІ 190198.01.24 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		60

20&linkId=714a8de782c5ea2242ef3f7c1be89bff&language=en_US&ref_=as_li_ss_tl&_x_tr_sl=auto&_x_tr_tl=uk&_x_tr_hl=ru .

7. Sure Petcare -SureFlap – SureFeed – Годівниця Microchip Pet. URL: https://www-amazon-com.translate.googleusercontent.com/translate/dp/B00O0UIPTY?&linkCode=s11&tag=ide+in+g+07-20&linkId=ad1f1f1122800f44ad209876802d1656&language=en_US&ref_=as_li_ss_tl&_x_tr_sl=auto&_x_tr_tl=uk&_x_tr_hl=ru

8. Університет Карнегі-Меллона та Virginia Tech сенсорна вушна бирка. URL: <https://www.cmu.edu/news/stories/archives/2021/november/wireless-device-monitoring-livestock.html>.

9. R. Chauhan et al., Design of Robotic Snake with ESP 32 CAM and Arduino, *2021 19th OITS International Conference on Information Technology (OCIT)*, 2021, pp. 6-9, doi: 10.1109/OCIT53463.2021.00013.

10. B. Gadgay, D. C. Shubhangi and R. Abhijeet, VTU campus surveillance and student smart accessibility system using IOT, *2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, 2021, pp. 1-6, doi: 10.1109/CSITSS54238.2021.9683672.

11. T. Ahmed, A. T. Bin Nuruddin, A. B. Latif, S. S. Arnob and R. Rahman, A Real-Time Controlled Closed Loop IoT Based Home Surveillance System for Android using Firebase, *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, 2020, pp. 601-606, doi: 10.1109/ICCAR49639.2020.9108016.

12. K. Ak, A. Kassim, J. -H. Lim and J. Y. Tham, Attribute Manipulation Generative Adversarial Networks for Fashion Images, *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 10540-10549, doi: 10.1109/ICCV.2019.01064.

13. C. Banupriya and S. Chandrakala, A low power Built In Self Repair technique for word oriented memories, *2014 International Conference on Electronics and Communication Systems (ICECS)*, 2014, pp. 1-5, doi: 10.1109/ECS.2014.6892679.

14. P. Habiby and R. Niaraki asli, An improved BIRA for memories with optimal repair rate using a flipping analyzer, *20th Iranian Conference on Electrical Engineering (ICEE2012)*, 2012, pp. 188-193, doi: 10.1109/IranianCEE.2012.6292350.

					КВРКІ 190198.01.24 ПЗ	Анк
Зм.	Арк.	№докум.	Підпис	Дата		62

15. Van Huu Thinh, Designing & manufacturing a rotary feeder, *2017 International Conference on System Science and Engineering (ICSSE)*, 2017, pp. 400-403, doi: 10.1109/ICSSE.2017.8030905.
16. Vania, K. Karyono and I. H. T. Nugroho, Smart dog feeder design using wireless communication, MQTT and Android client, *2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, 2016, pp. 191-196, doi: 10.1109/IC3INA.2016.7863048.
17. A. Kumar, S. Thakur and P. Bhattacharjee, Real Time Monitoring of AMR Enabled Energy Meter for AMI in Smart City - An IoT Application, *2018 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, 2018, pp. 219-222, doi: 10.1109/iSES.2018.00055.
18. Hashim H., Salihudin S. F. B., Saad P. S. M. Development of IoT Based Healthcare Monitoring System. *In 2022 IEEE International Conference in Power Engineering Application (ICPEA)*. 2022. Pp. 1-5.
19. Xiaomi.UA. URL: <https://www.xiaomi.ua> (дата звернення 14.05.2022)
20. Zhao W., Yi L. Research on the evolution of the innovation ecosystem of the Internet of Things: A case study of Xiaomi (China). *Procedia Computer Science*, 2022. Pp. 56-62.
21. Song Y., Luximon Y., Leong B. D., Qin Z. The e-commerce performance of internet of things (IoT) in disruptive innovation: Case of Xiaomi. *In Proceedings of the 2019 3rd International Conference on Software and e-Business*. 2019. Pp. 188-192.
22. Ammar M., Russello G., Crispo B. Internet of Things: A survey on the security of IoT frameworks. *J. Inf. Secur.* 2018. Pp. 8-27.
23. Pamungkas H., Akbar S. R., Ichsan M. H. H. Implementasi Sistem Pengenalan Perangkat dan Layanan Sensor dan Aktuator pada Rumah Cerdas Berbasis Arsitektur Publish-Subscribe. *Jurnal Pengembangan Teknologi Informasi & Ilmu Komputer*. 2018. Pp. 2522-2531.
24. Patnaikuni D. R. P. A Comparative Study of Arduino, Raspberry Pi and ESP8266 as IoT Development. *International Journal of Advanced Research in Computer Science*. 2017. Pp. 2350-2352.

25. Jahary M.M, Ichsan M.H.H., Primananda R. Implementasi Protokol Universal Plug and Play (UPnP) Pada Sensor dan Aktuator Untuk Otomasi Lampu. *Universitas Brawijaya*. 2019. Pp. 101-114.

26. Pusparini D., Ichsan M.H.H. Implementasi Low Power System untuk Pengambilan dan Pengiriman Data Berdasarkan Kehadiran Manusia. *Universitas Brawijaya*. 2019. Pp. 32-35.

27. Pratama R. P. Implementation and testing of the ESP8266 module with the MQTT-Dash android application on the MQTT network. *Asian Scientific Journal of Information Technology*. 2018. Pp. 157-164.

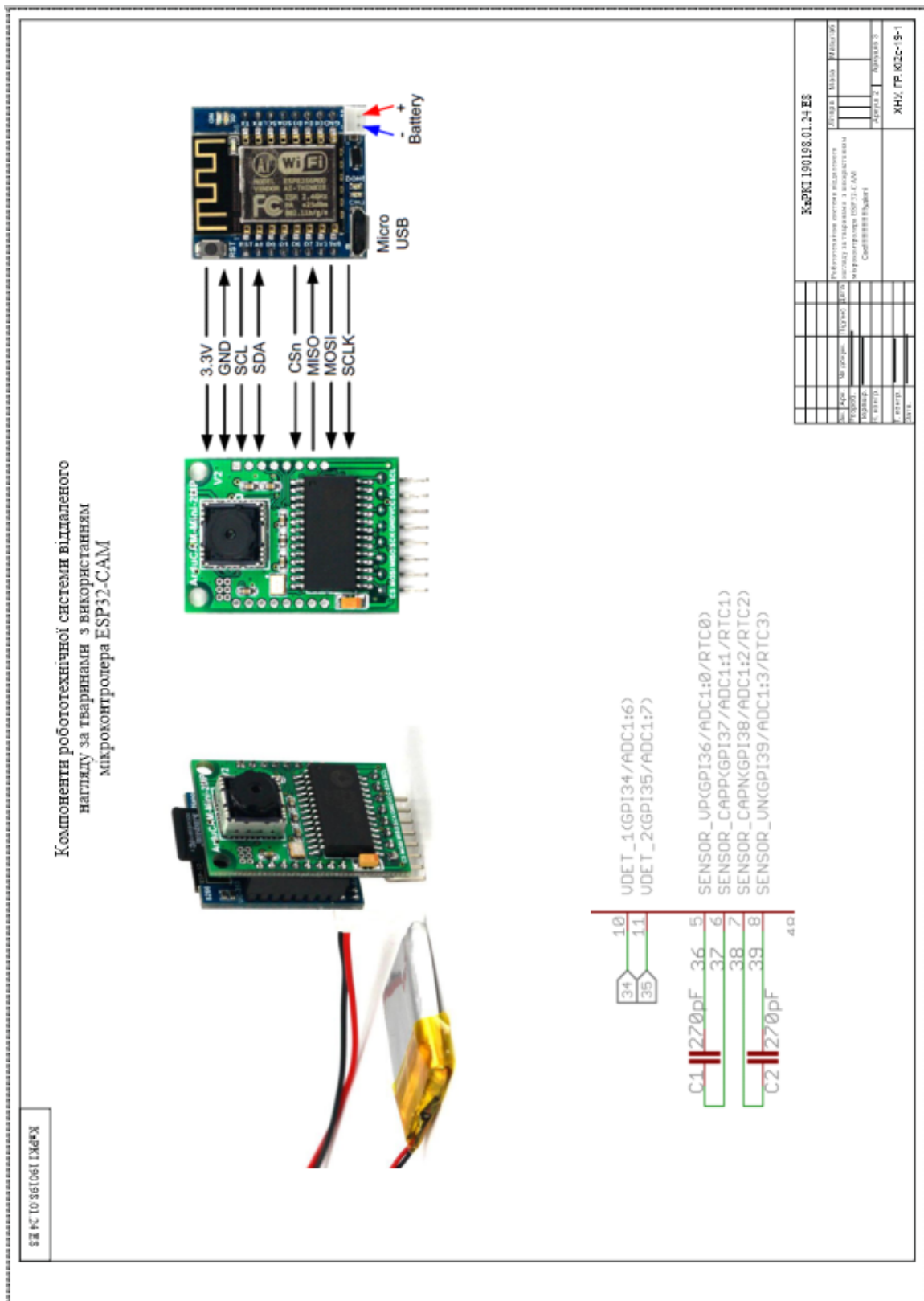
28. Jabbar, W. A., Shang H. K., Hamid S. N., Almohammed A. A., Ramli R. M., Ali M. A. IoT-BBMS: Internet of Things-based baby monitoring system for smart cradle. 2019. Pp. 791-793.

					КВРКІ 190198.01.24 ПЗ	Анк
Зм.	Арк.	№докум.	Підпис	Дата		64

Додаток Б

(обов'язково)

Копія креслення “Компоненти робототехнічної системи віддаленого нагляду за тваринами з використанням мікроконтролера ESP32-CAM”



Додаток Г

Код програмного забезпечення

```
/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <WiFiClient.h>
const char ssid[] = "Your WiFi SSID"; // WiFi name
const char password[] = "Your WiFi Password"; // WiFi password

#include <BlynkSimpleEsp32.h>
char auth[] = "Your Blynk Auth Token"; // Blynk Authentication Token
WiFiServer server(80);

#include <SPI.h>
#include <TFT_eSPI.h>
TFT_eSPI tft = TFT_eSPI(); // Resolution 135 x 240
#define FF17 &FreeSans9pt7b
#define FF21 &FreeSansBold9pt7b
#define ROW1 0,16
#define ROW2 0,38
#define ROW3 0,60
#define ROW4 0,82
#define ROW5 0,104
#define ROW6 0,126

#define BUTTON1 35
#define BUTTON2 0

#define STEPPER_IN1 27
#define STEPPER_IN2 26
#define STEPPER_IN3 25
#define STEPPER_IN4 33

#define REVOLUTION_STEP 2048 // 1 revolution

boolean stepperDirection = false;
int stepperStep = 0;
int stepperStepCount = 0;
boolean stepperMove = false;
long prevMillisStepper = 0;
int intervalStepper = 4; // Minimum is 2

boolean button1Pressed = false;
boolean button2Pressed = false;

BLYNK_WRITE(V0) // Button Widget is writing to pin V0
{
  stepperDirection = param.asInt();

  tft.fillRect(120, 65, 120, 25, TFT_BLACK);
```

```

tft.setCursor(120, 82);
if (stepperDirection) {
  tft.print("CCW");
}
else {
  tft.print("CW");
}
}

```

```

BLYNK_WRITE(V1) // Button Widget is writing to pin V1
{
  int stepperSpeed = param.asInt();

  tft.fillRect(120, 87, 120, 25, TFT_BLACK);
  tft.setCursor(120, 104);
  if (stepperSpeed == 1) {
    intervalStepper = 4;
    tft.print("Low");
  }
  else if (stepperSpeed == 2) {
    intervalStepper = 3;
    tft.print("Medium");
  }
  else if (stepperSpeed == 3) {
    intervalStepper = 2;
    tft.print("High");
  }
}

```

```

BLYNK_WRITE(V2) // Button Widget is writing to pin V2
{
  stepperMove = true;
  stepperStepCount = 0;
  stepperStep = 1;

  tft.fillRect(120, 109, 120, 25, TFT_BLACK);
  tft.setCursor(120, 126);
  tft.print("Run");
}

```

```

void setup()
{
  pinMode(BUTTON1, INPUT_PULLUP);
  pinMode(BUTTON2, INPUT_PULLUP);

  pinMode(STEPPER_IN1, OUTPUT);
  pinMode(STEPPER_IN2, OUTPUT);
  pinMode(STEPPER_IN3, OUTPUT);
  pinMode(STEPPER_IN4, OUTPUT);

  Serial.begin(115200);
  Serial.print("Initialize Blynk.");
}

```

```

tft.init();
tft.setRotation(3);
tft.fillScreen(TFT_BLACK);

tft.setFreeFont(FF21);
tft.setTextColor(TFT_BLUE);
tft.setCursor(ROW1);
tft.print("Blynk Status:");

tft.setFreeFont(FF17);
tft.setTextColor(TFT_WHITE);
tft.setCursor(120, 16);
tft.print("Initialize...");

WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected.");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());

Blynk.begin(auth, ssid, password);

tft.fillRect(120, 0, 120, 35, TFT_BLACK);
tft.setCursor(120, 16);
tft.print("Ready!");

tft.setFreeFont(FF21);
tft.setTextColor(TFT_RED);
tft.setCursor(ROW3);
tft.print("STEPPER");
tft.setTextColor(TFT_GREEN);
tft.setCursor(ROW4);
tft.print("Direction:");
tft.setCursor(ROW5);
tft.print("Speed:");
tft.setCursor(ROW6);
tft.print("Status:");

tft.setFreeFont(FF17);
tft.setTextColor(TFT_YELLOW);
tft.setCursor(120, 82);
tft.print("CW");
tft.setCursor(120, 104);
tft.print("Low");
tft.setCursor(120, 126);
tft.print("Stop");

Blynk.virtualWrite(0, 0);

```

```

Blynk.virtualWrite(1, 1);
Blynk.virtualWrite(2, 0);
}

void loop()
{
  Blynk.run();

  if (digitalRead(BUTTON1) == LOW &&
      button1Pressed == false) {
    button1Pressed = true;

    stepperDirection = false;
    stepperMove = true;
    stepperStepCount = 0;
    stepperStep = 1;
  }
  else if (digitalRead(BUTTON1) == HIGH &&
           button1Pressed == true) {
    button1Pressed = false;
  }

  if (digitalRead(BUTTON2) == LOW) {

    stepperDirection = true;
    stepperMove = true;
    stepperStepCount = 0;
    stepperStep = 1;
  }
  else if (digitalRead(BUTTON2) == HIGH &&
           button2Pressed == true) {
    button2Pressed = false;
  }

  if (millis() - prevMillisStepper > intervalStepper) {

    if (stepperMove == true) {
      if (stepperDirection) {
        if (stepperStep++ >= 3) {
          stepperStep = 0;
        }
      }
      else {
        if (stepperStep-- == 0) {
          stepperStep = 3;
        }
      }
    }

    if (stepperStepCount++ == REVOLUTION_STEP) {
      stepperMove = false;
      stepperStep = 4;

      Blynk.virtualWrite(2, 0);
    }
  }
}

```

```

tft.fillRect(120, 109, 120, 25, TFT_BLACK);
tft.setCursor(120, 126);
tft.print("Stop");
}

switch (stepperStep) {
case 0:
    digitalWrite(STEPPER_IN1, HIGH);
    digitalWrite(STEPPER_IN2, LOW);
    digitalWrite(STEPPER_IN3, LOW);
    digitalWrite(STEPPER_IN4, LOW);
    break;

case 1:
    digitalWrite(STEPPER_IN1, LOW);
    digitalWrite(STEPPER_IN2, HIGH);
    digitalWrite(STEPPER_IN3, LOW);
    digitalWrite(STEPPER_IN4, LOW);
    break;

case 2:
    digitalWrite(STEPPER_IN1, LOW);
    digitalWrite(STEPPER_IN2, LOW);
    digitalWrite(STEPPER_IN3, HIGH);
    digitalWrite(STEPPER_IN4, LOW);
    break;

case 3:
    digitalWrite(STEPPER_IN1, LOW);
    digitalWrite(STEPPER_IN2, LOW);
    digitalWrite(STEPPER_IN3, LOW);
    digitalWrite(STEPPER_IN4, HIGH);
    break;

default:
    digitalWrite(STEPPER_IN1, LOW);
    digitalWrite(STEPPER_IN2, LOW);
    digitalWrite(STEPPER_IN3, LOW);
    digitalWrite(STEPPER_IN4, LOW);
    break;
}
}

prevMillisStepper = millis();
}
}

```

Ось повний ескіз:

```
#include <WiFi101.h>
#include <SPI.h>
#include <TelegramBot.h>
const char* ssid = SECRET_SSID; // your network SSID (name)
const char* password = SECRET_PSWD; // your network password
const char BotToken[] = SECRET_BOT_TOKEN;
WiFiSSLClient client;
TelegramBot bot (BotToken, client);
TelegramKeyboard keyboard_one;
String OldChatId = "";
void setup() {
  Serial.begin(115200);
  delay(3000);
  // attempt to connect to Wifi network:
  Serial.print("Connecting Wifi: ");
  Serial.println(ssid);
  while (WiFi.begin(ssid, password) != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println("");
  Serial.println("WiFi connected");
  // choose the emoji you like using UNICODE
  // here's the list https://unicode.org/emoji/charts/full-emoji-list.html
  const char* SoundAndFood = "\U0001F3B6 + \U0001F36A"; // Note + Cookie
  const char* SoundNoFood = "\U0001F3B6 NO \U0001F36A"; // Note NO Cookie
  const char* OnePortion = "\U0001F408"; // CAT
  const char* TwoPortion = "\U0001F408 \U0001F408"; // 2 CATS
  const char* ThreePortion = "\U0001F408 \U0001F408 \U0001F408"; // 3 CATS
  // define your row's
  const char* row_one[] = {SoundAndFood, SoundNoFood};
  const char* row_two[] = {OnePortion, TwoPortion, ThreePortion};
  keyboard_one.addRow(row_one, 2); // assing a row to one or more keyboards
  keyboard_one.addRow(row_two, 3); // second argument is the length of the row
  bot.begin();
}
void loop() {
  message m = bot.getUpdates(); // Read new messages
  if ( m.chat_id != 0 ) { // Checks if there are some updates
    OldChatId = m.chat_id;
    Serial.println(m.text);
    bot.sendMessage(m.chat_id, "Hello !", keyboard_one);
  }
  delay(1000);
}
```

Sat May 28 20:14:14 EEST 2022, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 1.0%

 Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 14%**

ID: 104135 Название: Робототехнічна система віддаленого нагляду за тваринами з використанням мікроконтролера ESP32-CAM Добавлено в БД: 2022-05-28 Авторы: В.М. Семенишин Руководители: О.С. Засорнов Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	41731	421	409 (1%)	6 (1%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

Ім'я користувача:
Кафедра КІ

ID перевірки:
1011356892

Дата перевірки:
28.05.2022 21:02:08 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
28.05.2022 21:02:25 EEST

ID користувача:
100005591

Назва документа: Семенишен_Робототехнічна система віддаленого нагляду за тваринами з використанням м...
Кількість сторінок: 57 Кількість слів: 6951 Кількість символів: 49456 Розмір файлу: 4.26 MB ID файлу: 1011242507

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

3.11% Схожість

Найбільша схожість: 0.95% з джерелом з Бібліотеки (ID файлу: 1011224357)

2.03% Джерела з Інтернету 22 Сторінка 59

1.24% Джерела з Бібліотеки 66 Сторінка 59

0.35% Цитат

Цитати 1 Сторінка 60

Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи 31

Підозріле форматування 27 сторінок

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Семенишин Валерій Михайлович

Тема: Робототехнічна система віддаленого нагляду за тваринами з використанням мікроконтролера ESP32-CAM

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 63

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розроблення робототехнічної системи віддаленого нагляду за тваринами з використанням мікроконтролера ESP32-CAM.

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

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі було проведено загальний аналіз відомих засобів віддаленого нагляду тварин. Більшість засобів мають достатній функціонал для вирішення задачі віддаленого нагляду за тваринами. Головним недоліком розглянутих засобів є висока вартість рішень. Для вирішення поставленого завдання можливим є застосування програм-но-апаратні засобів, що базуються на системі ArduCAM-M-2MP ESP8266 Evaluation Kit.

У другому розділі було розглянуто здійснено обґрунтування та опис компонентів та засобів побудови системи віддаленого догляду за домашніми тваринами. Зокрема, було описано апаратне середовище реалізації, процес роботи ESP8266 з Arduino IDE, процес установки ядра плати за допомогою Boards Manager. В розділі подано основні аспекти використання Arduino IDE, компіляції та завантаження прошивки.

У третьому розділі представлено реалізацію робототехнічної системи віддаленого нагляду за тваринами з використанням мікроконтролера esp32-cam. Подано основні аспекти застосування ArduCAM-M-2MP ESP8266, специфікацію застосованих компонентів та процес збірки робототехнічної системи віддаленого нагляду за

тваринами. Також в розділ представлено процес встановлення ПЗ засобами Arduino IDE, встановлення ядра Arduino ESP32, клонування та завантаження ядра, встановлення інструментів Xtensa та ESP32, процес установка драйверів FTDI, налаштування роботи Arduino з WiFi, встановлення бібліотеки Arduino, застосування бібліотеки Blink. В розділі подано опис процесу написання програмного забезпечення для робототехнічної системи віддаленого нагляду за тваринами з використанням мікроконтролера esp32-cam.

5. Негативні сторони роботи: недостатня увага опису функцій роботи з контролером.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

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

8. Інші зауваження: _____

9. Оцінка дипломної роботи: задовільно (3.00/E)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Джулій В.М., к.т.н., доцент, кафедри кібербезпеки Хмельницького національного університету

“ ___ ” _____ 2022 р.

 (підпис)

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

Семенишин В.М.

ПІБ здобувача вищої освіти

ФПКТС, 4 курсу, групи КІ2с-19-1

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.

дата



підпис

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

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

Назва: Роботехнічна система віддаленого нагляду за тваринами з використанням мікроконтролера ESP32-CAM

Автор: Семенишин Валерій Михайлович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Засорнов Олександр Сергійович, к.т.н, доцент

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 укріплення 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 розміщені в розділі аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів щодо використаних програмних скриптів, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 8,1% і адресується до 104 періоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру практичної роботи і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІСН

 О.С. Засорнов

С. М. Лисенко

Т. О. Говорунченко