

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр  
Освітній рівень

РОБОТОТЕХНІЧНА СИСТЕМА ВИЗНАЧЕННЯ ЯКОСТІ БОРОШНА З  
ВИКОРИСТАННЯМ ESP32CAM  
Назва теми

КВРКІ 101052.21.01.02 ПЗ  
Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

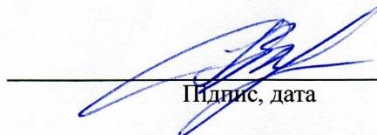
Назва

Виконав: студент III курсу, група KI2c-21-1

  
Підпис

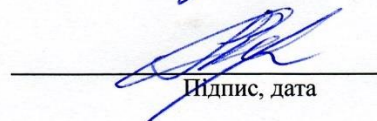
А.О. Гуцалюк  
Ініціали, прізвище

Керівник

  
Підпис, дата

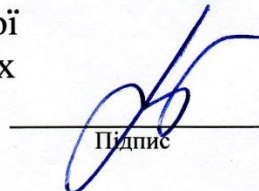
І.О. Засорнова  
Ініціали, прізвище

Нормоконтролер

  
Підпис, дата

І.О. Засорнова  
Ініціали, прізвище

До захисту допускаю:  
Зав. кафедри комп'ютерної  
інженерії та інформаційних  
систем

  
Підпис

Т.О. Говоруценко  
Ініціали, прізвище

«31» травня 2024 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 «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О. Говорущенко

“ 10 ” 01 2024 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Гуцалюку Артему Олексійовичу

Прізвище, ім'я, по батькові студента

1. Тема проєкту (роботи) Робототехнічна система визначення якості борошна з використанням ESP32CAM

Керівник проєкту (роботи) Засорнова І.О., к.т.н., доцент.

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

Затверджена наказом ректора університету від 15.02.2024 р. № 8

2. Строк подання студентом проєкту (роботи) на кафедру 01.06.2024 р.

3. Вихідні дані до проєкту (роботи) Завдання на кваліфікаційну роботу

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

Робототехнічна система визначення якості борошна з використанням ESP32CAM та ановка задачі щодо її удосконалення

Проектування робототехнічної системи визначення якості борошна

Програмно-апаратна реалізація та тестування робототехнічної системи


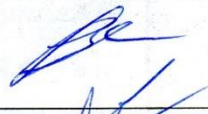


5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) \_\_\_\_\_

Спрощена електрично принципова схема

Алгоритм роботи системи

Інтерфейс програми

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Засорнова І.О., к.т.н., доцент		
Антиплагиат	Нічепорук А.О., доцент кафедри КПС		

7. Дата видачі завдання « 10 » 01 2024 р.

**КАЛЕНДАРНИЙ ПЛАН**

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2024	виконано
3	Робота над розділом 1 – Робототехнічна система визначення якості борошна з використанням ESP32CAM та постановка задачі щодо її удосконалення	01.03.2024	виконано
4	Робота над розділом 2 – Проектування робототехнічної системи визначання якості борошна	01.04.2024	виконано
5	Робота над розділом 3 – Програмно-апаратна реалізація та тестування робототехнічної системи	29.04.2024	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2024	виконано
7	Попередній захист ВКР	26.05.2024	виконано
8	Захист ВКР на засіданні ЕК	Червень 2024 року	

Студент

  
Підпис

А.О. Гуцалюк  
Ініціали, прізвище

Керівник роботи

  
Підпис

І.О. Засорнова  
Ініціали, прізвище



## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Робототехнічна система визначення якості борошна з використанням ESP32CAM».

Автор роботи: Гуцалюк Артем Олексійович.

Керівник роботи: Засорнова Ірина Олександрівна.

Пояснювальна записка: 66с., 29 рис., 4 табл., 5 дод., 60 джерел.

Графічна частина: 3 креслення.

**ESP32CAM, РОБОТОТЕХНІЧНА СИСТЕМА, ЯКІСТЬ БОРОШНА.**

Метою кваліфікаційної роботи є забезпечення обробки зображення платою ESP32CAM для робототехнічної системи визначення якості борошна. Поставлена мета досягається проєктуванням та розробкою підсистеми аналізу якості борошна за допомогою ESP32CAM, яка передбачає підвищення ефективності і точності процесів контролю якості борошна.

Об'єктом дослідження є процес обробки зображень у робототехнічній системі визначення якості борошна.

Предметом дослідження є підсистема керування обробки зображень у робототехнічній системі визначення якості борошна.

Практичне значення має спроектована та реалізована підсистема аналізу якості борошна на основі ESP32CAM

Під час проведення даного дослідження використаний метод систематичного огляду літератури для вивчення і аналізу предметної області даного дослідження з текстових джерел інформації.



Підпис студента



Дата

## ЗМІСТ

<b>ВСТУП</b> .....	4
<b>1 РОБОТОТЕХНІЧНА СИСТЕМА ВИЗНАЧЕННЯ ЯКОСТІ БОРОШНА З ВИКОРИСТАННЯМ ESP32CAM ТА ПОСТАНОВКА ЗАДАЧІ ЩОДО ЇЇ УДОСКОНАЛЕННЯ</b> .....	6
1.1 Аналіз структурних і функціональних особливостей робототехнічної системи.....	6
1.2 Аналіз наявного програмно-апаратного забезпечення предметної області .....	13
1.3 Визначення вимог до робототехнічної системи та розробка технічного завдання .....	19
1.4 Висновки. Постановка задачі .....	21
<b>2 ПРОЄКТУВАННЯ РОБОТОТЕХНІЧНОЇ СИСТЕМИ ВИЗНАЧАННЯ ЯКОСТІ БОРОШНА</b> .....	22
2.1 Вибір основних компонентів робототехнічної системи ESP32CAM.....	22
2.2 Обґрунтування вибору плати .....	23
2.3 Вибір програматора для ESP32CAM.....	27
2.4 Вибір компонентів для відображення інформації .....	31
2.5 Програмування робототехнічної системи .....	39
2.6 Висновки.....	42
<b>3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ РОБОТОТЕХНІЧНОЇ СИСТЕМИ</b> .....	43
3.1 Схема робототехнічної системи визначення якості борошна.....	43
3.2 Встановлення необхідних бібліотек Python.....	44
3.3 Встановлення необхідних ресурсів для ArduinoIDE.....	48
3.4 Основна програмна частина Arduino .....	53
3.5 Основна програмна частина Python .....	58
3.6 Розрахунок матеріальних витрат .....	62
3.7 Напрямки вдосконалення.....	63

КВРКІ 101052.21.01.02 ПЗ							
Зм.	Арк.	№ док.ум.	Підпис	Дата	Літера	Аркуш	Аркушів
Виконав	Гуцалюк А.О.			27.05			
Перевір.	Засорнова І.О.			31.05		2	66
Н.контр.	Засорнова І.О.			31.05	ХНУ КІ2с-21-1		
Затвер.	Говорущенко Т.О.			31.05			

Робототехнічна система визначення якості борошна з використанням ESP32CAM  
Пояснювальна записка

3.8 Висновки .....	63
<b>ВИСНОВОК</b> .....	65
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ</b> .....	67
<b>ДОДАТОК</b> Спрощена електрично принципова схема .....	74
<b>ДОДАТОК</b> Алгоритм роботи системи .....	75
<b>ДОДАТОК</b> Інтерфейс програми .....	76
<b>ДОДАТОК</b> Код програми на Python .....	77
<b>ДОДАТОК</b> Код програми для ESP32CAM.....	83

## ВСТУП

Одне з найбільш трансформаційних застосувань робототехніки — це робоче місце. У різних галузях промисловості роботи беруть на себе небезпечні або повторювані завдання, які часто зустрічаються у виробничих процесах [1]. Ця автоматизація дає змогу компаніям досягти більшої ефективності, підвищити продуктивність і, зрештою, стимулювати зростання бізнесу [2].

Хоча існує занепокоєння щодо переміщення робочих місць через роботів, дуже важливо розглядати автоматизацію як можливість для більш кваліфікованої робочої сили [3].

У сільськогосподарському секторі роботизовані системи революціонізують традиційні методи землеробства. Ці інноваційні рішення усувають нестачу робочої сили, підвищують ефективність і оптимізують врожайність [4].

Завдання, починаючи від посадки та збору врожаю до моніторингу стану ґрунту та внесення добрив і пестицидів, можуть бути автоматизовані за допомогою роботизованих систем [5].

Як вказується в статті про захист врожаю за допомогою робототехніки, сучасні сільськогосподарські роботи, оснащені набором датчиків і камер, перетворюються на потужні машини для збору даних. Ці роботи в режимі реального часу збирають інформацію про стан посівів, закономірності росту та умови навколишнього середовища на полі. Ці дані дають можливість фермерам приймати обґрунтовані рішення щодо зрошення, внесення добрив та боротьби зі шкідниками, що в кінцевому підсумку призводить до більш стійких та економічно ефективних сільськогосподарських практик [6].

Крім полів, робототехнічні системи з передовими датчиками та камерами знаходять застосування в перевірці та сортуванні продукції. Вони можуть класифікувати продукти на основі конкретних параметрів якості, таких як розмір, форма, колір і наявність дефектів [7]. Подібним чином у виробничих умовах роботи

					КВРКІ 101052.21.01.02 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

можуть забезпечувати якість продукту, виконуючи візуальні перевірки, точні вимірювання та різні тести [8].

Актуальність даної роботи полягає у зростанні попиту на якісні харчові продукти та необхідності ефективних процесів контролю якості в харчовій промисловості. Традиційні методи оцінки якості борошна часто є трудомісткими та суб'єктивними. Розробивши робототехнічну систему з використанням ESP32CAM, ми можемо оптимізувати процес контролю якості, зменшити людські помилки та забезпечити постійну якість продуктів на основі борошна.

Метою кваліфікаційної роботи є забезпечення обробки зображення платою ESP32CAM для робототехнічної системи визначення якості борошна.

Поставлена мета досягається проектуванням та розробкою підсистеми аналізу якості борошна за допомогою ESP32CAM, яка передбачає підвищення ефективності і точності процесів контролю якості борошна.

Об'єктом дослідження є процес обробки зображень у робототехнічній системі визначення якості борошна.

Предметом дослідження є підсистема керування обробки зображень у робототехнічній системі визначення якості борошна.

Для досягнення поставленої мети використовуються такі методи дослідження, як методи аналізу та моделювання процесів, принципи системного аналізу.

Практичне значення має спроектована та реалізована підсистема аналізу якості борошна на основі ESP32CAM.

# 1 РОБОТОТЕХНІЧНА СИСТЕМА ВИЗНАЧЕННЯ ЯКОСТІ БОРОШНА З ВИКОРИСТАННЯМ ESP32СAM ТА ПОСТАНОВКА ЗАДАЧІ ЩОДО ЇЇ УДОСКОНАЛЕННЯ

1.1 Аналіз структурних і функціональних особливостей робототехнічної системи

Сучасні робототехнічні системи зазнають швидкої трансформації, інтегруючи складні алгоритми, передові сенсорні технології та потужні обчислювальні можливості. Ці досягнення дають роботам змогу вирішувати дедалі ширший спектр завдань з більшою автономією та адаптивністю. Вони можуть взаємодіяти з навколишнім середовищем, в тому числі з людьми, а також обробляти та аналізувати інформацію в режимі реального часу.

Сучасні робототехнічні системи надають перевагу зручним інтерфейсам для полегшення безперешкодної співпраці між людиною і роботом [9].

Промислова революція для роботів почалася в 1954 році з представленням Unimate, розробленого Джорджем Деволом. Ця новаторська Робототехнічна рука, призначена для підняття важких речей на заводах, була прийнята General Motors як перший промисловий робот-користувач [10].

Протягом наступних двох десятиліть Unimation випустила вдосконалені версії Unimate, зміцнивши свою роль у різних галузях промисловості, зокрема в автомобілебудуванні [10].

Кінець 1960-х і 1970-ті, про що написано було Robotics Trends та Robotics Trends, роки стали свідками значного прогресу в робототехніці, з розробкою більш складних роботизованих рук, оснащених системами бачення (камерами) і вдосконаленими датчиками для підвищення їх спритності та функціональності [11]. Знаковим досягненням цієї епохи став Shakey, перший у світі мобільний робот, створений

					КВРКІ 101052.21.01.02 ПЗ	Арк. 6
Зм.	Арк.	№ докum.	Підпис	Дата		

Стенфордським науково-дослідним інститутом у 1966 році. Shakey, незважаючи на свої обмеження, проклав шлях мобільним роботам завдяки інноваційному використанню програмного та апаратного забезпечення для сприйняття та навігації [12].

Мелоді Чжу розповіла, що з прогресом робототехнічної зброї в 1950-х роках корпорація Barrett Electronics Corporation представила перший автономний керований транспортний засіб (AGV).

Цей електричний транспортний засіб з його здатністю самостійно переміщатися по заздалегідь визначених маршрутах поклав початок мобільним промисловим роботам [13].

Протягом наступних десятиліть AGV зазнали значних удосконалень, отримавши здатність демонструвати більш складну поведінку та орієнтуватися в динамічному середовищі [13].

У 1990-х роках відбулася інтеграція високоточних датчиків і лазерів у AGV, що ще більше розширило їхні можливості та проклало шлях до більш складних автономних мобільних роботів (AMR), які використовуються сьогодні [13].

У статті [14] автор обговорює повсюдну присутність робототехнічних систем у сучасному світі, висвітлюючи їхній вплив на різні галузі та повсякденну діяльність. Дані технологічні досягнення проникли не лише на заводи промислових лідерів, але й у невеликі простори наших будинків, трансформуючи спосіб, у який ми працюємо і живемо [15]. Також згадується про різноманітні сектори, які використовують робототехнічні системи, а саме:

– у промисловому виробництві роботи докорінно змінили виробничий ландшафт. У таких галузях, як автомобілебудування, електроніка та харчова промисловість, роботи стали інструментом автоматизації різних етапів виробництва [14].

Робототехнічні системи вирішують різноманітні завдання, починаючи від складських приміщень й точної обробки до ефективного пакування й ретельного сортування продукції [16];

– у галузі медицини сталася революція з появою роботів-хірургів. Ці технології дають можливість хірургам виконувати складні процедури з великою точністю, що сприяє швидшому одужанню пацієнтів [17]. Окрім хірургії, роботи також досягають успіхів у медичній діагностиці, реабілітаційній терапії та безлічі інших застосувань у сфері охорони здоров'я [18];

– логістика оптимізована завдяки інтеграції роботів. Ці робототехнічні системи відмінно справляються з такими завданнями, як виконання замовлень, сортування продуктів і навіть доставка, особливо на великих складах, вантажних терміналах і навіть в системах доставки «останньої милі» [19];

– сфера послуг використовує роботів, щоб покращити взаємодію з клієнтами та оптимізувати робочі процеси. Наприклад, у готелях, ресторанах і аеропортах все частіше розгортаються обслуговуючі роботи, які допомагають клієнтам виконувати такі завдання, як пошук інформації, обробка замовлень і навіть основні взаємодії [20];

– сільськогосподарський сектор також переживає революцію робототехніки. Роботи використовуються для автоматизації різноманітних завдань, включаючи зрошення, збирання врожаю та культивування землі, що призводить до підвищення ефективності та продуктивності на фермах у всьому світі [21].

Незважаючи на широкий спектр застосування робототехнічних систем, Самуель Карлссон вважають, що їх можна згрупувати в три основні категорії, виходячи з їхніх основних функцій [22]:

– роботи-маніпулятори у світі автоматизації чудово вміють точно переміщувати об'єкти та керувати ними [22]. В промислових умовах бездоганно виконуючи такі завдання, як складання продукту, пакування та сортування з непохитною точністю та ефективністю. У звіті Міжнародної федерації робототехніки (IFR) за 2022 рік зазначено, що більше половини (52%) усіх промислових роботів у світі класифікуються як маніпулятори [23];

– мобільні роботи, розроблені для переміщення навколо та взаємодії з навколишнім середовищем, пропонують широкий спектр можливостей [22]. Вони

					КВРКІ 101052.21.01.02 ПЗ	Арк. 8
Зм.	Арк.	№ докум.	Підпис	Дата		

можуть приймати форму корисних домашніх помічників, орієнтуватися в небезпечних середовищах з метою перевірки або навіть досліджувати глибини наших океанів для дослідницьких зусиль [24]. Прориви в сенсорних технологіях і одночасній локалізації та картографуванні (SLAM) постійно розширюють межі можливостей мобільних роботів [25];

– сенсорні роботи спеціально призначені для збору, обробки та аналізу даних [22]. Вони діють як очі та вуха різних операцій, прискіпливо відстежуючи умови навколишнього середовища, вимірюючи найважливіші параметри та навіть надаючи дані в реальному часі для керування та координації інших роботів у системі [26].

Стаття 2022 року, опублікована в журналі IEEE Robotics & Automation Magazine авторами Фенбей Лю, Ю Тянь та інші, досліджує цей прогрес, підкреслюючи, як ці роботи, оснащені різними датчиками, можуть значно оптимізувати виробничі лінії і гарантувати контроль [27].

Робототехнічні системи збору та управління даними стали повсюдними в багатьох галузях промисловості, відіграючи ключову роль у різних сценаріях [28], як зазначають дослідники Гаолян Пенг та Чжуцзюнь Чжан.

У сфері виробництва ці системи слідкують за виробничими лініями, автоматично збираючи важливі дані про процеси. Ці дані дають виробникам можливості:

– контролювати продуктивність: Інформація в режимі реального часу дозволяє ретельно відстежувати ефективність виробництва, визначати області для вдосконалення і забезпечувати стабільну якість продукції;

– відхилення від очікуваних параметрів можуть бути швидко зафіксовані, що дозволяє вчасно втрутитися і запобігти потенційним проблемам, які можуть порушити виробництво;

– оптимізувати виробничі потоки: Аналізуючи тенденції даних, можна виявити вузькі місця і відкоригувати виробничі процеси, щоб покращити загальний потік і максимізувати видобуток;

– збір даних здійснюється за допомогою різноманітних датчиків і пристроїв, пристосованих до конкретних потреб. Поширені приклади включають:

– датчики навколишнього середовища контролюють такі фактори, як температура, вологість і тиск, щоб забезпечити оптимальні умови для виробничих процесів [28];

– системи машинного зору оснащені сучасними можливостями обробки зображень, дозволяють проводити візуальний огляд продукції, полегшуючи контроль якості та виявлення дефектів [29];

– у певних ситуаціях аналіз звуку через мікрофони або відеозапис може надати цінну інформацію про роботу обладнання або потенційні проблеми [28].

Зібрані дані передаються до центральної системи для подальшої обробки та аналізу. Для цього необхідно:

– методи передачі даних: комунікаційні модулі, такі як Wi-Fi, Bluetooth або стільникові мережі, забезпечують безпечну і надійну передачу даних від точки збору до блоку обробки [30];

– системи обробки даних: мікроконтролери та мікропроцесори як вбудовані системи відіграють вирішальну роль у завданнях попередньої обробки даних, таких як фільтрація, агрегація та форматування інформації для подальшого аналізу [31].  
Комп'ютери та хмарні платформи: Для комплексного аналізу та зберігання даних часто використовують високопродуктивні комп'ютери або хмарні рішення [32].

Загальні функціональні особливості, які впливають на дизайн робототехнічних систем, включають наступні важливі параметри: потужність та швидкість роботи робототехнічної системи, точність виконання основного завдання системи, надійність системи за різних умов її експлуатації, гнучкість і маневреність, автономність і незалежність, сенсорні можливості та енергоефективність. Ці функції мають вирішальне значення для того, щоб продуктивність робота відповідала вимогам його передбачуваного застосування (таблиця 1.1) [33].

Таблиця 1.1 – Загальні функціональні особливості робототехнічних систем

№	Функціональна особливість	Характеристики
1	2	3
1	Потужність та швидкість	<ul style="list-style-type: none"> <li>– забезпечення достатньої сили для виконання фізичних завдань;</li> <li>– швидке реагування на зміни у виробничому середовищі для оптимізації робочих процесів</li> </ul>
2	Точність і надійність	<ul style="list-style-type: none"> <li>– виконання рухів з максимальною точністю для уникнення помилок та пошкоджень;</li> <li>– надійна робота в різних умовах, включаючи екстремальні та непередбачені ситуації;</li> <li>– забезпечення стабільної та неперервної роботи без виникнення збоїв чи аварій</li> </ul>
3	Гнучкість та маневреність	<ul style="list-style-type: none"> <li>– адаптація до різних конфігурацій та умов роботи для максимальної універсальності;</li> <li>– здатність до виконання різноманітних завдань за різних обставин;</li> <li>– можливість виконання складних маневрів та рухів для досягнення поставленої мети</li> </ul>
4	Автономність і самостійність	<ul style="list-style-type: none"> <li>– здатність до самостійного прийняття рішень та виконання дій без прямого керування оператором;</li> <li>– автоматизоване виконання повсякденних рутинних завдань без постійної участі людини;</li> <li>– забезпечення безперервної роботи без необхідності частого перезавантаження чи підзаряджання</li> </ul>

Кінець таблиці 1.1

1	2	3
5	Сенсорна та здатність до взаємодії з оточуючим середовищем	<ul style="list-style-type: none"> <li>– сприйняття широкого спектру інформації з оточення для прийняття оптимальних рішень;</li> <li>– здатність до взаємодії з різними об'єктами та предметами навколо;</li> <li>– використання різноманітних сенсорів для отримання точної та повної інформації про навколишнє середовище</li> </ul>
6	Енергоефективність	<ul style="list-style-type: none"> <li>– ефективне використання енергії для максимізації часу роботи та збільшення продуктивності;</li> <li>– мінімізація споживання енергії для забезпечення стабільної та довготривалої роботи;</li> <li>– використання енергозберігаючих технологій для оптимізації витрат електроенергії</li> </ul>

Ці функціональні особливості можуть змінюватися в залежності від конкретного типу робототехнічної системи, яка розробляється. Наприклад:

- для промислових роботів, які використовуються в автоматизованих виробничих лініях, потужність і швидкість можуть мати більше значення, ніж точність, оскільки головною метою є швидке та ефективне виробництво;

- у роботах для медичних або хірургічних застосувань надійність та точність виконання рухів можуть бути критичними, оскільки вони безпосередньо впливають на безпеку пацієнтів;

- для роботів, які використовуються в дослідницьких місіях у космосі, важливими можуть бути автономність та енергоефективність, оскільки вони повинні функціонувати без зовнішнього впливу та ефективно використовувати обмежені ресурси.

У підсумку, зазначений аналіз предметної області робототехнічних систем та їх структурних та функціональних особливостей відображає велику

різноманітність вимог та потреб у цьому сегменті технологічного простору. Розглянуті аспекти, такі як потужність, точність, автономність, та інші, є ключовими для розробки ефективних робототехнічних систем, спроможних відповідати вимогам сучасного світу. Зрозуміння цих особливостей визначає не лише успішність технічного проєкту, а й його можливий внесок у вирішення різноманітних завдань у сферах виробництва, медицини, досліджень та інших галузях.

## 1.2 Аналіз наявного програмно-апаратного забезпечення предметної області

Компанія Intel у своїй статті [34] розказує про те, що у світі робототехніки існує різноманіття підходів та технологій для вирішення подібних завдань, що включають в себе використання камер, сенсорів та інших компонентів. Аналіз наявного програмно-апаратного забезпечення в цій предметній області охоплює різні варіанти рішень, що вже застосовуються або досліджуються в інших проєктах. Цей аналіз дасть змогу виявити переваги та обмеження різних підходів, порівняти їх ефективність та потенціал в контексті нашої конкретної задачі. Враховуючи різноманітність доступних технологій та програмних рішень [34].

Аналізуючи існуюче програмне та апаратне забезпечення в предметній області робототехнічної системи визначення якості борошна з використанням ESP32CAM, ми можемо провести паралелі та ідеї з іншого реального проєкту, який демонструє застосування роботизованих систем у вирощуванні пшениці. Одним із таких проєктів, який пропонує цінні знання та ідеї, є ініціатива «AgBot II», розроблена дослідниками Техаського університету A&M.

Проєкт AgBot II [35], започаткований у співпраці з галузевими партнерами, був спрямований на розробку автономної робототехнічної системи, призначеної для точного землеробства, з особливим акцентом на вирощуванні пшениці. Незважаючи на те, що основною метою було вдосконалення моніторингу врожаю, управління та збору врожаю, проєкт також охоплював елементи оцінки якості, які

					КВРКІ 101052.21.01.02 ПЗ	Арк. 13
Зм.	Арк.	№ докум.	Підпис	Дата		

мають відношення до контексту робототехнічної системи визначення якості борошна [35].

Система AgBot II складалася з ряду роботизованих апаратних компонентів, призначених для виконання різноманітних завдань протягом усього процесу вирощування пшениці. До них належать автономні наземні транспортні засоби, оснащені системами навігації GPS, роботами для точного посіву та збирання врожаю, а також датчиками для моніторингу стану ґрунту, здоров'я врожаю та параметрів навколишнього середовища. Крім того, системи візуалізації, такі як мультиспектральні камери та гіперспектральні датчики, були інтегровані в робототехнічну платформу для отримання детальних даних про рослини пшениці та їх якісні характеристики.

Центральним елементом функціональності системи AgBot II були складні програмні рішення для автономної роботи, аналізу даних і прийняття рішень. Це включало програмне забезпечення для моніторингу та контролю в режимі реального часу для нагляду за ходом польових операцій і коригування параметрів за потреби. Алгоритми машинного навчання були використані для аналізу даних, зібраних із датчиків і систем обробки зображень, що дозволило виявити такі показники якості, як розмір зерна, колір і вміст білка. Крім того, були розроблені спеціальні модулі програмного забезпечення для взаємодії з системою визначення якості борошна на основі ESP32CAM, що забезпечує безперебійну інтеграцію робототехнічних процесів збору та аналізу даних.

Проєкт AgBot II є прикладом потенціалу робототехнічних систем для революції у вирощуванні пшениці та пов'язаних процесах, включаючи визначення якості борошна. Завдяки інтеграції передових апаратних і програмних технологій проєкт продемонстрував, як автономні робототехнічні системи можуть покращити моніторинг урожаю, управління та оцінку якості, що призводить до покращення врожайності, ефективності та стійкості. Надалі знання, отримані від таких проєктів, як AgBot II, продовжуватимуть інформувати про розробку інноваційних рішень для точного землеробства та виробництва продуктів харчування [36].

Проект «Виявлення об'єктів за допомогою камер ESP32 для контролю якості сталевих компонентів у виробничих конструкціях» надає цінні відомості, які можна застосувати до робототехнічної системи визначення якості борошна з використанням ESP32CAM. Обидва проекти використовують камери ESP32CAM для захоплення та обробки зображень, підкреслюючи універсальність цього обладнання в різних галузях промисловості.

Однією з паралелей між двома проектами є використання камер ESP32CAM для захоплення зображень у реальному часі. У проекті контролю якості сталевих компонентів камера ESP32 використовується для захоплення зображень болтових сталевих елементів, тоді як у робототехнічній системі визначення якості борошна ESP32CAM використовується для захоплення зображень зразків борошна. Це демонструє гнучкість камер ESP32 для зйомки зображень у різних програмах [37].

Обидва проекти також використовують методи обробки зображень для аналізу. У проекті контролю якості сталевих компонентів, C++ і алгоритм Faster R-CNN використовуються для обробки зображень для виявлення відсутніх болтів. Подібним чином у робототехнічній системі визначення якості борошна алгоритми обробки зображень можуть використовуватися для аналізу якості зразків борошна на основі візуальних характеристик, таких як колір, текстура та зернистість. Це підкреслює важливість обробки зображень в обох проектах для цілей контролю якості [38].

Загалом, проект «Виявлення об'єктів за допомогою камер ESP32 для контролю якості сталевих компонентів у виробничих конструкціях» є цінним довідником для розробки робототехнічної системи визначення якості борошна з використанням ESP32CAM, демонструючи потенціал камер ESP32 у додатках автоматизації та контролю якості. в різних галузях.

Системи машинного зору з використанням промислових камер і Raspberry Pi пропонують значний прогрес в оцінці якості борошна. Ці системи забезпечують покращену якість зображення, потужність обробки та гнучкість порівняно з більш простими рішеннями, такими як ESP32CAM. Використовуючи промислові камери

та Raspberry Pi, ці системи можуть аналізувати зразки борошна з вищою точністю та ефективністю, що робить їх для застосувань, які потребують точного контролю та аналізу якості.

Компоненти які використовувалися в даному проєкті:

– промислові камери, які використовуються в цих системах, як правило, пропонують високу роздільну здатність (наприклад, 5 МП або вище) із чудовою точністю кольору та продуктивністю в умовах слабкого освітлення. Ці камери призначені для промислового застосування, забезпечуючи надійну та стабільну якість зображення для точного аналізу;

– Raspberry Pi служить процесором для системи машинного зору. Завдяки потужним можливостям обробки Raspberry Pi може виконувати складні завдання аналізу зображень, включаючи алгоритми машинного навчання для класифікації якості борошна;

– налаштування освітлення: правильне освітлення має вирішальне значення для точного аналізу зображення. Система машинного зору оснащена стабільними налаштуваннями освітлення, щоб забезпечити послідовне освітлення зразків борошна, зменшуючи мінливість отриманих зображень;

– стрічковий конвеєр (додатково): для автоматизованого аналізу зразків борошна в систему можна інтегрувати стрічковий конвеєр. Це дозволяє безперервно надходити зразки, підвищуючи продуктивність і ефективність.

Особливості проєкту на основі Raspberry Pi:

– висока якість зображення промислової камери та потужна обробка Raspberry Pi дозволяють точніше оцінювати якість борошна порівняно з споживчими камерами та мікроконтролерами, такими як ESP32CAM;

– можливості обробки Raspberry Pi дозволяють реалізувати передові алгоритми машинного навчання. Ці алгоритми можуть аналізувати різні візуальні характеристики зразків борошна, такі як колір, текстура та зернистість, що забезпечує більш точну класифікацію;

– систему можна легко масштабувати, додавши додаткові камери та модулі Raspberry Pi. Ця масштабованість дозволяє збільшити пропускну здатність і здатність обробляти більшу кількість зразків борошна;

– платформа Raspberry Pi пропонує гнучкість у налаштуванні та інтеграції з іншими системами. Ця гнучкість дозволяє адаптувати систему до конкретних вимог і інтегрувати її в існуючі виробничі процеси.

Недоліки даного проєкту:

– промислові камери та пристрої Raspberry Pi дорожчі, ніж споживчі камери та мікроконтролери. Початкова вартість інвестицій для цих компонентів може бути вищою, особливо якщо врахувати потребу в кількох блоках для масштабованості;

– налаштування та підтримка системи машинного зору з промисловими камерами та Raspberry Pi вимагає технічних знань. Належне калібрування та конфігурація системи є важливими для оптимальної продуктивності;

– система споживає більше енергії порівняно з більш простими рішеннями, такими як ESP32CAM. Це більш високе енергоспоживання слід брати до уваги при плануванні енергопостачання системи та експлуатаційних витрат.

Системи машинного бачення з промисловими камерами та Raspberry Pi добре підходять для програм, що вимагають високої точності, великомасштабної обробки та інтеграції з іншими системами. Ці системи підходять для галузей промисловості, де якість борошна має вирішальне значення, таких як пекарні, млини та харчові підприємства.

Системи машинного зору з промисловими камерами та Raspberry Pi пропонують значні переваги в оцінці якості борошна, включаючи вищу точність, масштабованість і гнучкість. Хоча ці системи можуть мати вищу початкову вартість і вимагати технічного досвіду для налаштування та обслуговування, вони забезпечують виняткову продуктивність і добре підходять для вимогливих застосувань, де якість борошна має вирішальне значення. Ретельний аналіз вимог системи, бюджету та технічних можливостей має важливе значення при виборі між більш простими.

Один із методів використання системи на базі ESP32CAM полягає у використанні її як датчика руху різноманітних галузях. ESP32CAM – це мікроконтролер, що має вбудовану камеру та можливість взаємодії з іншими пристроями через Wi-Fi або Bluetooth зв'язок. Його функціональні можливості включають в себе виявлення руху, розпізнавання обличчя та об'єктів, зйомку відео різної якості та роздільної здатності, а також передачу даних через мережу.

Один із прикладів використання ESP32CAM як датчика руху – це система відеоспостереження для безпеки. Коли датчик руху виявляє рух у певній зоні, ESP32CAM автоматично запускає запис відео або знімає фотографії, а потім відправляє отримані дані на сервер або на мобільний пристрій користувача через Інтернет. Це дозволяє відстежувати події в реальному часі та забезпечує додатковий рівень безпеки для будинків, офісів або інших об'єктів.

Крім того, ESP32CAM може бути використаний як система контролю доступу. Наприклад, вона може реєструвати обличчя працівників та дозволяти їм доступ до певних приміщень або областей на основі розпізнавання обличчя. Це забезпечує зручний та безпечний спосіб контролю доступу до приміщень, зменшуючи ризик несанкціонованого доступу.

Крім того, ESP32CAM може бути використаний у виробництві для відстеження та контролю за робочими процесами. Він може зафіксувати рух об'єктів на конвеєрі або виробничому приміщенні, дозволяючи вчасно виявляти проблеми та здійснювати контроль за якістю продукції.

Аналіз існуючих програмних і апаратних рішень у цій галузі показує, що, незважаючи на те, що існують різні варіанти, доступні для подібних проєктів, вони часто суттєво відрізняються з точки зору основної спрямованості та складності. Існуючі рішення з використанням модуля ESP32CAM, який є центральним у проєкті «Робототехнічна система визначення якості борошна з використанням ESP32CAM», є дещо обмеженими. Незважаючи на те, що ESP32CAM пропонує економічно ефективну та доступну платформу для додатків обробки зображень, більшість існуючих проєктів, пов'язаних з ESP32CAM, орієнтовані на інші цілі,

такі як спостереження, моніторинг або DIY проєкти, а не на конкретні промислові програми, такі як оцінка якості борошна.

З іншого боку, існують рішення в галузі промислового машинного зору, які використовують апаратне забезпечення вищого класу, наприклад промислові камери та системи Raspberry Pi. Ці рішення забезпечують розширені можливості обробки зображень, вищу точність і потенціал інтеграції з іншими системами. Однак вони також мають вищу ціну, підвищену складність у налаштуванні та обслуговуванні та більш високе енергоспоживання порівняно з рішеннями на основі ESP32CAM.

Хоча рішення на основі ESP32CAM може не запропонувати такий самий рівень складності, як промислові камери та налаштування Raspberry Pi, воно надає більш доступний і доступний варіант для проєктів із простішими вимогами чи бюджетними обмеженнями. Модуль ESP32CAM у поєднанні з відповідним програмним забезпеченням і алгоритмами все ще може досягати надійних результатів для оцінки якості борошна.

### 1.3 Визначення вимог до робототехнічної системи та розробка технічного завдання

Приступаючи до розробки проєкту під назвою «Робототехнічна система для визначення якості борошна з використанням ESP32CAM», необхідне розуміння існуючої предметної області. Незважаючи на наявність аналогічних проєктів, важливо розпізнавати їх відмінності, особливо з точки зору вартості, складності та доступності.

Економічна ефективність: враховуючи різницю у вартості існуючих рішень, основною метою є розробка системи, яка є економічно доцільною без шкоди для якості. Це передбачає вибір апаратних компонентів, таких як ESP32CAM, відомий своєю доступністю, і водночас гарантує, що вони належним чином відповідають вимогам проєкту. Крім того, використання інструментів і алгоритмів програмного

забезпечення з відкритим кодом може ще більше мінімізувати витрати, пов'язані з розробкою програмного забезпечення;

Простота та відтворюваність: простота є ключем до сприяння більш широкому застосуванню та відтворюваності системи. Віддаючи пріоритет простоті як апаратного, так і програмного забезпечення, проєкт має на меті оптимізувати процес розробки та сприяти легкому тиражуванню іншими. Це включає використання легкодоступних компонентів, чітку документацію та стандартизовані процедури складання та експлуатації;

Зручний інтерфейс: інтерфейс користувача системи повинен бути інтуїтивно зрозумілим і доступним навіть для осіб з обмеженим технічним досвідом. Це забезпечує легкість експлуатації та заохочує до ширшого використання різними групами користувачів, включаючи дрібних виробників борошна, дослідників та любителів. Інтерфейс має надавати чіткі інструкції, візуальний зворотний зв'язок і прості елементи керування для ініціювання та моніторингу процесу оцінювання;

Масштабованість і гнучкість: хоча початкова увага зосереджена на створенні базової системи, важливо при проєктуванні пам'ятати про масштабованість і гнучкість. Це передбачає застосування модульного підходу до розробки апаратного та програмного забезпечення, що дозволяє легко розширювати та налаштовувати в майбутньому. Наприклад, система повинна підтримувати інтеграцію додаткових датчиків або модулів для розширення функціональних можливостей, таких як автоматизована обробка зразків або реєстрація даних;

Надійність і точність: незважаючи на наголос на простоті та доступності, система повинна забезпечувати надійні та точні результати оцінки якості борошна. Це вимагає ретельного тестування та перевірки на кожному етапі розробки, щоб забезпечити послідовність і точність вимірювань. Необхідно включити процедури калібрування для врахування коливань умов навколишнього середовища та продуктивності обладнання.

					КВРКІ 101052.21.01.02 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

#### 1.4 Висновки. Постановка задачі

У розділі було розглянуто історію створення та етапи розвитку робототехнічних систем. Також були виконані дослідження основних понять про робототехнічні системи, їх класифікації і види та основні складові компоненти даних систем.

Проаналізовано декілька робототехнічних систем. Першим є проект спостереження та контролю якості урожаю AgBot II. Також був проект з використанням ESP32 який мав на меті забезпечити надійний контроль якості сталевих компонентів у виробничих конструкціях. Іншим прикладом є робототехнічна система машинного зору на основі Raspberry Pi.

Визначено вимоги до робототехнічної системи: економічна ефективність, простота та відтворюваність, зручний інтерфейс, масштабованість і гнучкість, надійність і точність.

Для досягнення мети роботи необхідно виконати наступні задачі:

- здійснити вибір основних компонентів робототехнічної системи;
- обґрунтувати вибір плати;
- обрати програматор для ESP32CAM;
- обрати компоненти для відображення інформації;
- встановити необхідні бібліотеки для Arduino IDE та Phyton;
- створити програмну частину для Arduino IDE та Phyton;
- розрахувати матеріальні витрати;
- визначити напрямки вдосконалення.

## 2 ПРОЄКТУВАННЯ РОБОТОТЕХНІЧНОЇ СИСТЕМИ ВИЗНАЧАННЯ ЯКОСТІ БОРОШНА

### 2.1 Вибір основних компонентів робототехнічної системи ESP32CAM

При розробці робототехнічної системи визначення якості борошна необхідно враховувати компактність і ефективність. Було запропоновано обрати плату ESP32CAM в якості основного компонента через її компактний розмір і вражаючі можливості, включаючи модуль камери з високою роздільною здатністю, вбудований Wi-Fi і потужний мікропроцесор. Ця плата буде слугувати основою для системи, забезпечуючи бездоганну інтеграцію з іншими компонентами. Для відображення інформації можна використати світлодіоди, як дешеве та просте рішення, або обрати дисплей який буде більш гнучким у варіантах виводу інформації. Крім того, система повинна мати ефективне джерело живлення, що має забезпечує оптимальну продуктивність і подовжений час безвідмовної роботи для додатків, що живляться від батареї.

Однією з ключових функцій робототехнічної системи є визначення якості борошна, яке ґрунтується на аналізі візуальних даних, знятих камерою. Отже, вибір відповідного рішення обробки зображень має вирішальне значення. Обраний рішення має бути не тільки сумісним з мікропроцесором вибраного пристрою, але й достатньо ефективним для обробки зображень у режимі реального часу, гарантуючи, що система може забезпечити точну оцінку якості борошна.

Іншим важливим аспектом, який слід враховувати, є енергоспоживання вибраних компонентів. Оскільки основний пристрій живиться від батареї, важливо використовувати енергоефективні компоненти, щоб збільшити тривалість роботи. Необхідно ретельно розглянути вимоги до потужності кожного компонента та їх вплив на загальне енергоспоживання системи. Це може включати вибір малопотужних альтернатив або впровадження стратегій керування живленням для оптимізації споживання енергії.

					КВРКІ 101052.21.01.02 ПЗ	Арк. 22
Зм.	Арк.	№ докум.	Підпис	Дата		

Крім того, міцність і надійність вибраних компонентів є важливими для забезпечення довговічності та сталості роботи робототехнічної системи. Компоненти, схильні до виходу з ладу з часом, можуть поставити під загрозу точність і надійність процесу оцінки якості борошна. Таким чином, дуже важливо вибирати компоненти з підтверженою репутацією надійності та довговічності навіть у складних умовах експлуатації.

Нарешті, слід також враховувати вартість компонентів, оскільки вона безпосередньо впливає на загальний бюджет проєкту. При виборі компонентів важливо знайти баланс між продуктивністю та доступністю. Це може включати вивчення альтернативних компонентів або розробку інноваційних рішень, які можуть досягти бажаних функцій за менших витрат.

## 2.2 Обґрунтування вибору плати

В основі робототехнічної системи якості борошна лежить ESP32CAM, універсальна плата розробки від AI-Thinker. На рисунку 2.1 зображено зовнішній вигляд плати.



Рисунок 2.1 – Зовнішній вигляд ESP32CAM

Цей компактний модуль пропонує переконливу комбінацію функцій, які роблять його підходящим для нашого проекту. ESP32CAM об'єднує камеру високої роздільної здатності, можливості Wi-Fi для бездротового зв'язку та двоядерний процесор для ефективної обробки зображень [39].

Одним із переваг ESP32CAM є його економічна ефективність. Порівняно з іншими рішеннями на ринку, ESP32CAM пропонує широкий набір функцій за значно нижчою ціною. Така доступність робить його вдалим вибором для проекту, дозволяючи розробити економічно ефективну робототехнічну систему без шкоди для продуктивності чи функціональності.

Роздільна здатність камери може відрізнятися залежно від конкретної моделі ESP32CAM, але найчастіше це сенсор OV2640 із роздільною здатністю 640×480 пікселів [40]. Цієї роздільної здатності достатньо для отримання детальних зображень зразків борошна для аналізу якості. Зовнішній вигляд камери ESP32CAM зображено на рисунку 2.2.



Рисунок 2.2 – Зовнішній вигляд камери ESP32CAM

ESP32CAM побудований на основі потужного двоядерного мікропроцесора, який дозволяє йому з легкістю справлятися великою кількістю різноманітних задач. Ця особливість дозволяє ESP32CAM виконувати алгоритми обробки зображень, необхідні для оцінки якості борошна в режимі реального часу. Крім того,

можливості Wi-Fi ESP32CAM забезпечують надійний зв'язок і передачу даних, забезпечуючи віддалений моніторинг і контроль системи.

На додаток до переваг у продуктивності та вартості, ESP32CAM також має високий ступінь універсальності. Модуль сумісний із широким спектром датчиків, приводів та інших компонентів, що дозволяє легко інтегрувати та розширювати можливості системи. Ця гнучкість гарантує, що робототехнічна система може бути адаптована та вдосконалена за потреби, враховуючи нові вимоги чи технології, коли вони виникають.

ESP32CAM підтримується спільнотою розробників, які роблять внесок у його постійний розвиток та вдосконалення. Ця спільнота надає велику кількість ресурсів, зокрема бібліотеки, навчальні посібники та приклади проєктів, які можна використовувати для оптимізації процесу розробки та подолання потенційних проблем.

Підсумовуючи, ESP32CAM пропонує основу робототехнічної системи для визначення якості борошна. Його доступність, високопродуктивний процесор і універсальність роблять його переконливим вибором для проєкту, а потужна підтримка спільноти забезпечує доступ до цінних ресурсів і досвіду. Використовуючи можливості ESP32CAM, ми можемо розробити ефективне рішення для оцінки якості борошна.

Характеристики ESP32CAM описані в таблиці 2.1.

Таблиця 2.1 – Характеристики ESP32CAM

Характеристика	Опис
1	2
Мікропроцесор	Двоядерний мікропроцесор Xtensa LX6, що працює на частоті до 240 МГц
Пам'ять	4 МБ флеш-пам'яті та 520 КБ SRAM
Камера	Модуль камери OV2640 з роздільною здатністю до 2 МП

Кінець таблиці 2.1

1	2
Зв'язок	Wi-Fi (802.11 b/g/n) и Bluetooth
Інтерфейси	UART, SPI, I2C, PWM
Джерело живлення	Діапазон робочої напруги від 3,0 В до 3,6 В, рекомендована вхідна напруга 5 В
Розміри	Компактний форм-фактор, розміри 27 мм х 40,5 мм (без антени)
Підтримка розробки	Підтримується Arduino IDE та ESP-IDF, з великими ресурсами спільноти

ESP32CAM має загалом 14 штифтів, розташованих у два ряди вздовж країв модуля. Ці виводи можна розділити на виводи живлення, зв'язку та універсальні виводи вводу/виводу (GPIO). Нижче наведено короткий огляд розводки виводів ESP32CAM та їх призначення:

1. Виводи живлення:

- 3V3: вихідний вивід, що забезпечує живлення 3,3 В;
- GND: вивід заземлення, використовується як опорний рівень напруги.

2. Комунікаційні виводи:

- U0TXD і U0RXD це виводи UART для послідовного зв'язку. U0TXD – це вивід для передавання, а U0RXD – для приймання;
- GPIO15 і GPIO2: Комунікаційні виводи SPI. GPIO15 слугує виводом тактового генератора (SCK), а GPIO2 – виводом даних (MOSI);
- GPIO13 і GPIO14: комунікаційні виводи I2C. GPIO13 – це вивід послідовних даних (SDA), а GPIO14 – вивід послідовного тактового генератора (SCL).

3. Виводи GPIO: ESP32CAM має кілька виводів GPIO, які можна використовувати для різних цілей, наприклад, для вводу, виводу або взаємодії з датчиками та виконавчими пристроями. Ці виводи включають GPIO0, GPIO1,

GPIO3, GPIO4, GPIO5, GPIO9, GPIO10, GPIO12 та GPIO16. Зверніть увагу, що деякі виводи GPIO мають специфічні функції або обмеження, наприклад, GPIO0 і GPIO2, які беруть участь у процесі завантаження, і з ними слід поводитися обережно.

На схемі (рисунок 2.3) показано розташування виводів, включаючи виводи GPIO, виводи живлення та комунікаційні інтерфейси. Таке розташування дозволяє легко інтегруватися з іншими пристроями і датчиками.



Рисунок 2.3 – Розташування контактів плати ESP32CAM

### 2.3 Вибір програматора для ESP32CAM

У розробці робототехнічної системи модуль ESP32CAM відіграє вирішальну роль як основний пристрій, відповідальний за збір візуальних даних і їх обробку для оцінки якості борошна. Щоб спростити процес програмування та покращити

загальний досвід розробки, ми можемо використати ESP32CAM-MB (рисунок 2.4), зручну плату програмування, розроблену спеціально для ESP32CAM.

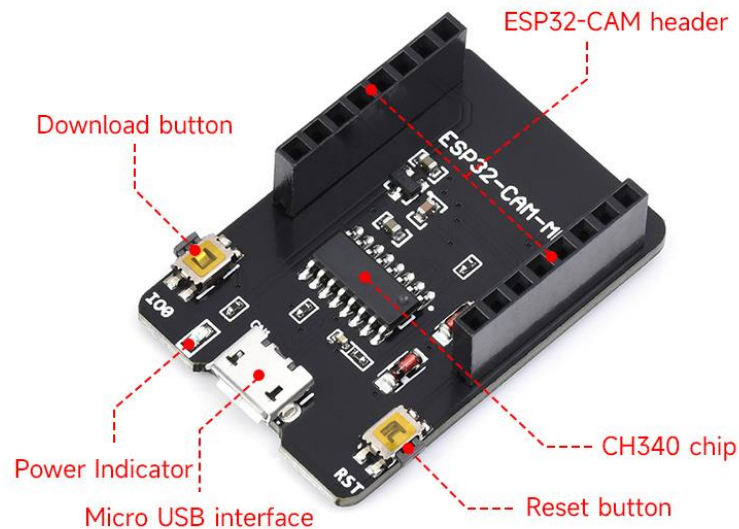


Рисунок 2.4 – Зовнішній вигляд ESP32CAM-MB

ESP32CAM-MB – це компактна і проста у використанні плата програмування, яка спрощує процес програмування модуля ESP32CAM. Він оснащений мостом USB-UART, який дозволяє користувачам підключати ESP32CAM безпосередньо до комп'ютера через кабель USB.

За допомогою ESP32CAM-MB користувачі можуть легко програмувати ESP32CAM за допомогою популярних платформ розробки, таких як Arduino IDE або ESP-IDF. Плата сумісна з обома платформами, що дозволяє розробникам використовувати для проєкту свої бажані інструменти та бібліотеки. Ця гнучкість робить ESP32CAM-MB вдалим вибором для програмування ESP32CAM у робототехнічній системі визначення якості борошна.

Хоча ESP32CAM-MB забезпечує зручне і просте рішення для програмування модуля ESP32CAM в робототехнічній системі для визначення якості борошна, існує альтернативний варіант: програматор FTDI.

FTDI-програмактор – це універсальний перетворювач USB в послідовний інтерфейс, який можна використовувати для програмування різних плат мікроконтролерів, включаючи ESP32CAM (рисунок 2.5).

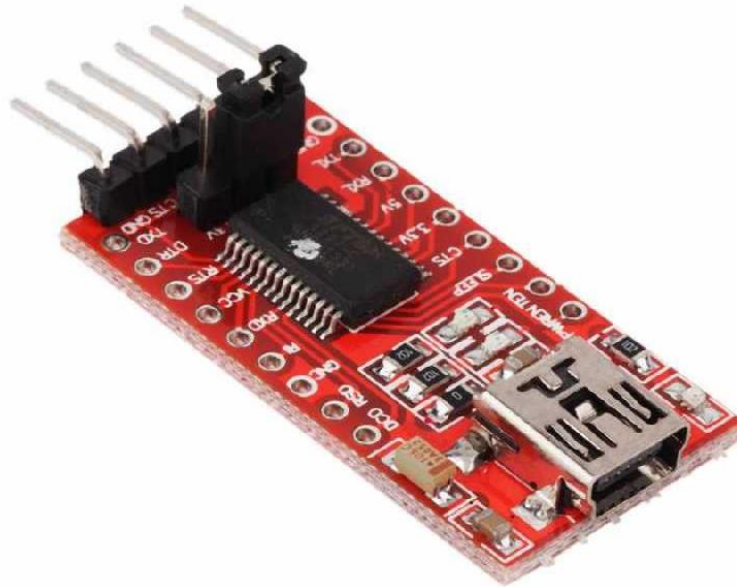


Рисунок 2.5 – Зовнішній вигляд FTDI-програмактора

FTDI-програмактор підключається до модуля ESP32CAM через його контакти UART, забезпечуючи зв'язок між модулем і комп'ютером. Використовуючи USB-кабель для підключення FTDI-програмактора до комп'ютера, користувачі можуть легко завантажувати код і налагоджувати ESP32CAM. Це робить програмактор FTDI життєздатною альтернативою ESP32CAM-MB для програмування ESP32CAM в робототехнічній системі.

Однією з переваг використання програмактора FTDI є його універсальність. На відміну від ESP32CAM-MB, який розроблений спеціально для ESP32CAM, програмактор FTDI можна використовувати з широким спектром плат мікроконтролерів.

Однак важливо зазначити, що використання FTDI-програмактора з ESP32CAM вимагає додаткового підключення та налаштування порівняно з ESP32CAM-MB. Контакти FTDI-програмактора мають бути підключенні до відповідних контактів

UART на модулі ESP32CAM, забезпечивши правильну полярність і рівні напруги. Це може зайняти більше часу і призвести до помилок, ніж використання ESP32CAM-MB, який пропонує більш спрощений процес програмування за принципом "підключи і працюй" (рисунок 2.6).

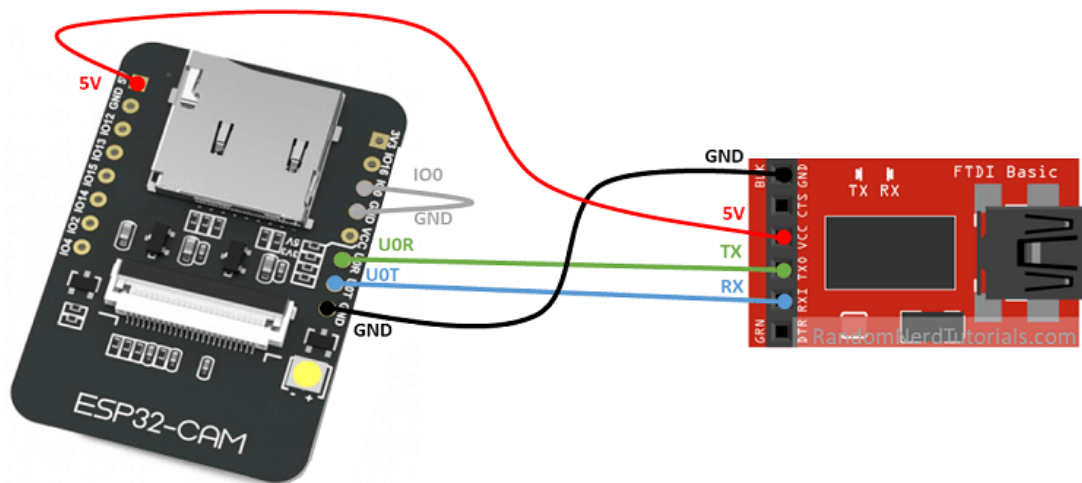


Рисунок 2.6 – Підключення ESP32CAM та FTDI-програмактора

На додаток до ESP32CAM-MB і програмактора FTDI, для програмування модуля ESP32CAM доступний ще один варіант: CP2102. CP2102 – це мікросхема моста USB-UART, яка служить економічно вигідною альтернативою раніше згаданим рішенням для програмування (рисунок 2.7).

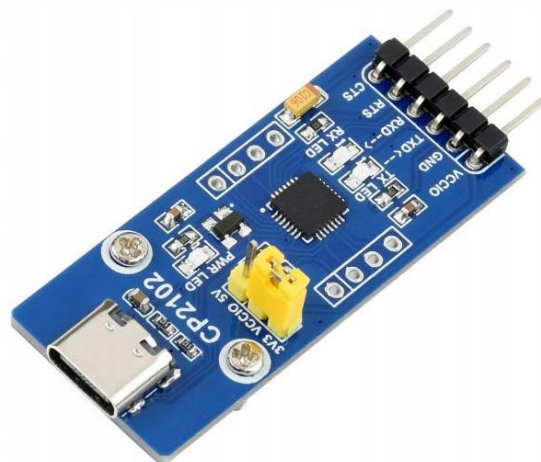


Рисунок 2.7 – Зовнішній вигляд CP2102 -програмактора

Подібно до програматора FTDI, CP2102 підключається до модуля ESP32CAM через контакти UART, забезпечуючи зв'язок між модулем і комп'ютером. Користувачі можуть завантажувати код і налагоджувати ESP32CAM, підключивши CP2102 до комп'ютера за допомогою USB-кабелю.

Основною перевагою CP2102 є його доступність. Він зазвичай доступний за нижчою ціною в порівнянні з ESP32CAM-MB і програматором FTDI, що робить його привабливим варіантом для розробників з обмеженим бюджетом.

Однак CP2102 зазвичай вважається менш зручним у використанні, ніж інші рішення для програмування. Він вимагає додаткового підключення та налаштування, подібно до програматора FTDI, і може не забезпечувати такий же рівень сумісності або надійності. Крім того, CP2102 може не підтримувати деякі функції та розширення, такі як автоматичне скидання, які можуть спростити процес програмування. На рисунку 2.8 зображено підключення ESP32CAM до CP2102 для програмування плати.

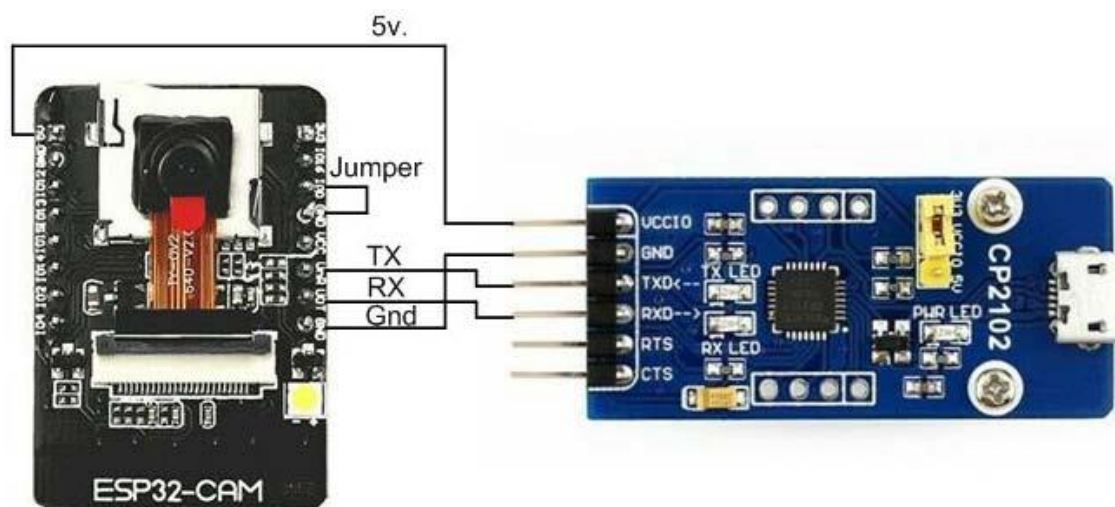


Рисунок 2.8 – Підключення ESP32CAM та CP2102-програматора

#### 2.4 Вибір компонентів для відображення інформації

При розробці робототехнічної системи важливо обробляти та аналізувати дані, зібрані модулем ESP32CAM. Хоча дані можуть бути оброблені на комп'ютері,

відображення результатів на самому пристрої є більш зручним і зрозумілим для користувача. Для вирішення даного питання є два варіанти виведення інформації: за допомогою світлодіодів і за допомогою світлодіодного дисплея.

Одним з простих і економічно ефективних методів виведення інформації є використання декількох світлодіодів для відображення різних рівнів якості або індикаторів стану (рисунок 2.9). Запрограмувавши ESP32CAM на керування світлодіодами, система може забезпечити візуальне представлення результатів оцінки якості борошна.

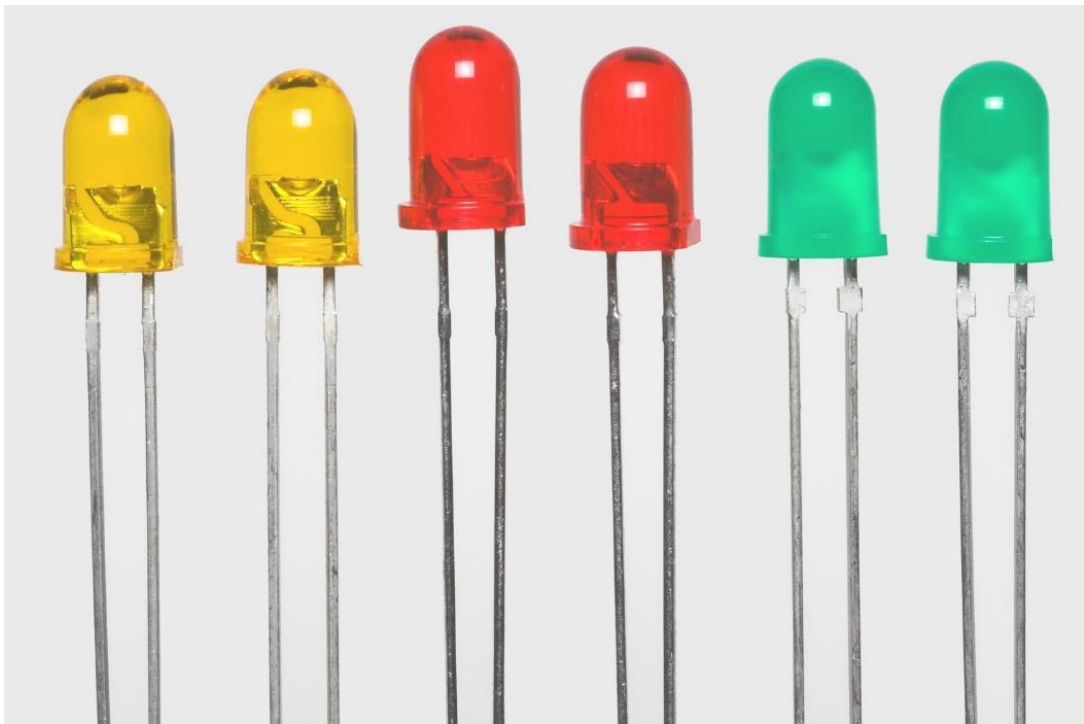


Рисунок 2.9 – Зовнішній вигляд світлодіодів

Світлодіоди недорогі і широко доступні, що робить їх вдалим варіантом для бюджетних проєктів.

Керування світлодіодами за допомогою ESP32CAM є відносно простим і вимагає мінімальної кількості додаткових компонентів. Ця простота дозволяє швидше створювати прототипи і полегшує пошук і усунення несправностей в процесі розробки.

Однак, використання світлодіодів забезпечує лише базове представлення результатів і може не передавати детальну або специфічну інформацію.

Користувачам може знадобитися зрозуміти значення кольору або стану кожного світлодіода для точної інтерпретації результатів, що може призвести до плутанини або неправильної інтерпретації.

На рисунку 2.10 зображення приклад базового підключення плати ESP32 та світлодіодів системи світлофор.

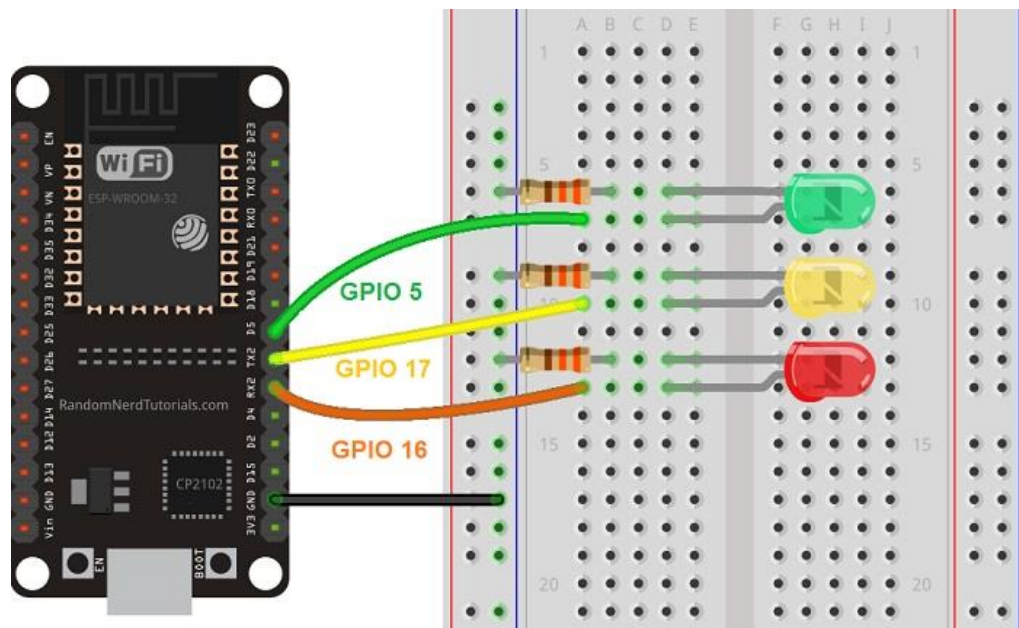


Рисунок 2.10 – Приклад використання світлодіодів та ESP32

Альтернативним методом виведення інформації є використання світлодіодного дисплея, наприклад, 7-сегментного дисплея або OLED-дисплея. Ці дисплеї можуть відображати більш детальну інформацію, включаючи числові значення і текст, забезпечуючи більш чітке представлення результатів оцінки якості борошна.

Світлодіодні дисплеї можуть відображати числові значення, текст та інші відповідні дані, пропонуючи більш повне розуміння результатів.

Користувачі можуть швидко і легко інтерпретувати інформацію, що відображається на світлодіодному дисплеї, без необхідності розбиратися в

конкретних колірних кодах або станах світлодіодів. Така простота інтерпретації може призвести до більш точного та ефективного прийняття рішень в процесі оцінки якості борошна.

Однак світлодіодні дисплеї, як правило, дорожчі, ніж окремі світлодіоди, і деякі з них можуть потребувати додаткових компонентів, таких як драйвери дисплея або перетворювачі.

Для робототехнічної системи є велика кількість можливих дисплеїв. Розглянемо декілька з них (таблиця 2.2).

Таблиця 2.2 – Характеристики дисплеїв

Тип дисплея	Роздільна здатність	Глибина кольору	Інтерфейс	Розмір	Енергоспоживання
16×2 символний РК-дисплей	16×2 символів	Монохромний	Паралельний/I2C	2.7"	Низький
Графічний OLED-дисплей (SSD1306)	128×64 пікселів	Монохромний	I2C/SPI	0.96" – 2.42"	Від низького до середнього
TFT РК-дисплей (ILI9341)	320×240 пікселів	16-бітний колір	SPI	2.4" – 3.5"	Від середнього до високого
ST7789 TFT РК-дисплей	240×320 пікселів	16-бітний колір	SPI	1.14" – 3.5"	Від середнього до високого

Символьний РК-дисплей з роздільною здатністю 16×2 – це монохромний дисплей, здатний відобразити 16 символів на рядок у двох рядках. Він має низьке

енергоспоживання і доступний з паралельним або I2C інтерфейсом. Цей дисплей простий у використанні та економічно вигідний, що робить його придатним для виведення базової інформації.

Однак його обмежена роздільна здатність і монохромне відображення можуть не забезпечити рівень деталізації, необхідний для більш складного представлення даних (рисунок 2.11).

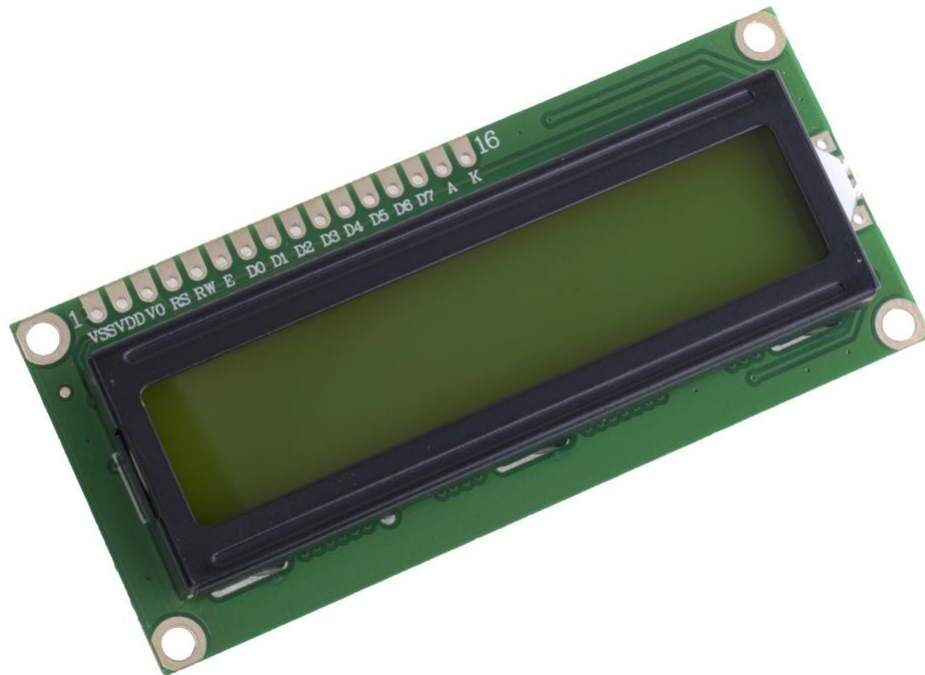


Рисунок 2.11 – Зовнішній вигляд 16×2 символьного РК-дисплей

Графічний OLED-дисплей на базі контролера SSD1306 має кілька переваг, які роблять його чудовим вибором для нашого проєкту. З роздільною здатністю 128×64 пікселів він забезпечує достатньо місця для відображення відповідної інформації про якість борошна. Доступність розмірів від 0,96" до 2,42" дозволяє вибрати дисплей, який найкраще відповідає фізичним обмеженням Робототехнічної системи.

Однією з ключових особливостей цього дисплея є використання технології OLED, яка має ряд переваг над традиційними РК-дисплеями. OLED-дисплеї

забезпечують кращу контрастність і ширші кути огляду, гарантуючи, що відображувана інформація залишається чіткою і розбірливою навіть за різних умов освітлення. Це особливо важливо для нашого проєкту, оскільки Робототехнічна система може використовуватися в різних середовищах з різною освітленістю.

Крім того, OLED-дисплей має інтерфейс I2C або SPI, що спрощує процес підключення та комунікації між дисплеєм і модулем ESP32CAM. Така простота інтеграції дозволяє нам більше зосередитися на розробці алгоритмів оцінки якості борошна і менше – на тонкощах комунікації з дисплеєм (рисунок 2.12).

Низьке та середнє енергоспоживання графічного OLED-дисплея є вирішальним фактором для робототехнічної системи з живленням від батареї. Мінімізація енергоспоживання має важливе значення для збільшення часу роботи пристрою, дозволяючи йому виконувати більше оцінок якості борошна на одному заряді.

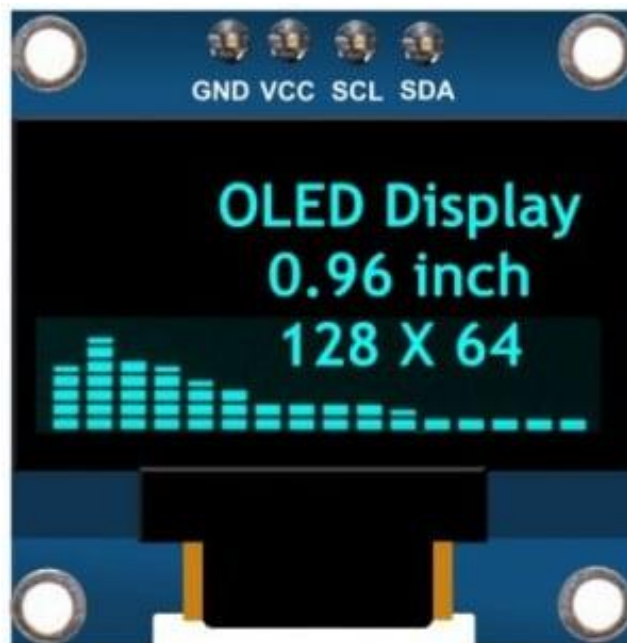


Рисунок 2.12 – Графічний OLED-дисплей (SSD1306)

TFT РК-дисплей на базі контролера ILI9341 має вражаючу роздільну здатність 320×240 пікселів і підтримує 16-бітну глибину кольору. Дисплей з високою роздільною здатністю забезпечує достатньо місця для відображення

детальної інформації про якість борошна, наприклад, графіків, діаграм та інших візуальних зображень, які можуть допомогти зрозуміти результати оцінки.

Однією з ключових особливостей цього дисплея є його здатність забезпечувати високоякісну передачу кольору. Ця можливість може бути особливо корисною в нашому проєкті, якщо ми вирішимо включити кольорові індикатори або візуалізації для представлення різних аспектів якості борошна. Використання кольору може зробити відображувану інформацію більш інтуїтивно зрозумілою для користувачів.

TFT РК-дисплей ILI9341 (рисунок 2.13) використовує інтерфейс SPI для зв'язку, що забезпечує надійну та ефективну передачу даних між дисплеєм і модулем ESP32CAM. Однак важливо відзначити, що цей дисплей має середнє і високе енергоспоживання і більш високу вартість в порівнянні з іншими варіантами дисплеїв. Ці фактори можуть бути недоліками, особливо для додатків з живленням від батареї, таких як наша робототехнічна система, де мінімізація енергоспоживання і контроль витрат є вирішальними факторами.



Рисунок 2.13 – TFT РК-дисплей (ILI9341)

TFT РК-дисплей ST7789 (рисунок 2.14) має роздільну здатність 240×320 пікселів і підтримує 16-бітну глибину кольору. Цей дисплей з високою роздільною здатністю забезпечує достатньо місця для відображення детальної інформації про якість борошна, наприклад, графіків, діаграм та інших візуальних зображень, які можуть допомогти зрозуміти результати оцінки.



Рисунок 2.14 – ST7789 TFT РК-дисплей

Однією з ключових особливостей цього дисплея є його здатність забезпечувати високоякісну передачу кольору. Ця можливість може бути особливо корисною в нашому проєкті, якщо ми вирішимо включити кольорові індикатори або візуалізації для представлення різних аспектів якості борошна. Використання кольору може зробити відображувану інформацію більш інтуїтивно зрозумілою для користувачів.

Враховуючи специфічні вимоги та обмеження нашої робототехнічної системи для визначення якості борошна, графічний OLED-дисплей SSD1306 є найбільш підходящим варіантом для наших потреб. Цей дисплей пропонує збалансоване поєднання роздільної здатності, енергоспоживання та вартості.

## 2.5 Програмування робототехнічної системи

Програмне забезпечення для робототехнічної системи складається з двох основних частин: код мікроконтролера, написаний за допомогою Arduino IDE, і код обробки зображень, написаний на Python на комп'ютері.

Arduino IDE — це популярна платформа з відкритим кодом, яка спрощує процес написання та завантаження коду на плати мікроконтролерів. Він пропонує зручний інтерфейс і широкий спектр бібліотек, є вибором для програмування головної плати нашої робототехнічної системи якості борошна.

Код Arduino буде виконувати різні завдання, такі як:

- ініціалізація та налаштування модуля камери та інших периферійних пристроїв;
- зйомка зображень зразків борошна через рівні проміжки часу;
- управління виконавчими механізмами та датчиками для маніпулювання пробами борошна;
- зв'язок з комп'ютером для передачі даних і отримання результатів обробки зображень.

Щоб проаналізувати зроблені зображення та визначити якість борошна, ми будемо використовувати Python, потужну та універсальну мову програмування. Python пропонує багато бібліотек для обробки зображень, машинного навчання та аналізу даних, тому він добре підходить для цього завдання.

Код Python відповідатиме за:

- отримання зображень від мікроконтролера через інтерфейс зв'язку (наприклад, послідовний, Wi-Fi або Bluetooth);
- попередня обробка зображень для покращення їх якості та видалення шумів;
- застосування алгоритмів обробки зображень для виділення релевантних ознак, пов'язаних із якістю борошна.
- аналіз виділених ознак для визначення якості борошна на основі попередньо визначених критеріїв.
- надсилання результатів аналізу назад до мікроконтролера для подальшої обробки або відображення.

Об'єднавши можливості Arduino IDE для програмування мікроконтролерів і Python для обробки зображень, можна створити ефективне та надійне програмне рішення для робототехнічної системи якості борошна. Цей підхід дозволяє використовувати сильні сторони обох платформ, забезпечуючи безперебійний зв'язок між платою та комп'ютером, одночасно досягаючи точної та надійної оцінки якості борошна.

Щоб забезпечити безперебійну передачу даних між мікроконтролером і комп'ютером та відповідно взаємодію плати та комп'ютера, ми будемо використовувати Wi-Fi як інтерфейс зв'язку. Плата ESP32CAM, яка використовується, має вбудовані можливості Wi-Fi.

Wi-Fi зв'язок відповідатиме за:

- встановлення безпечного з'єднання між мікроконтролером і комп'ютером;
- передача отриманих зображень з мікроконтролера на комп'ютер для обробки зображень;
- отримання результатів обробки зображень з комп'ютера та відображення їх на дисплеї дошки або збереження для подальшого аналізу.

На рисунку 2.15 представлено простий алгоритм за яким буде працювати робототехнічна система.

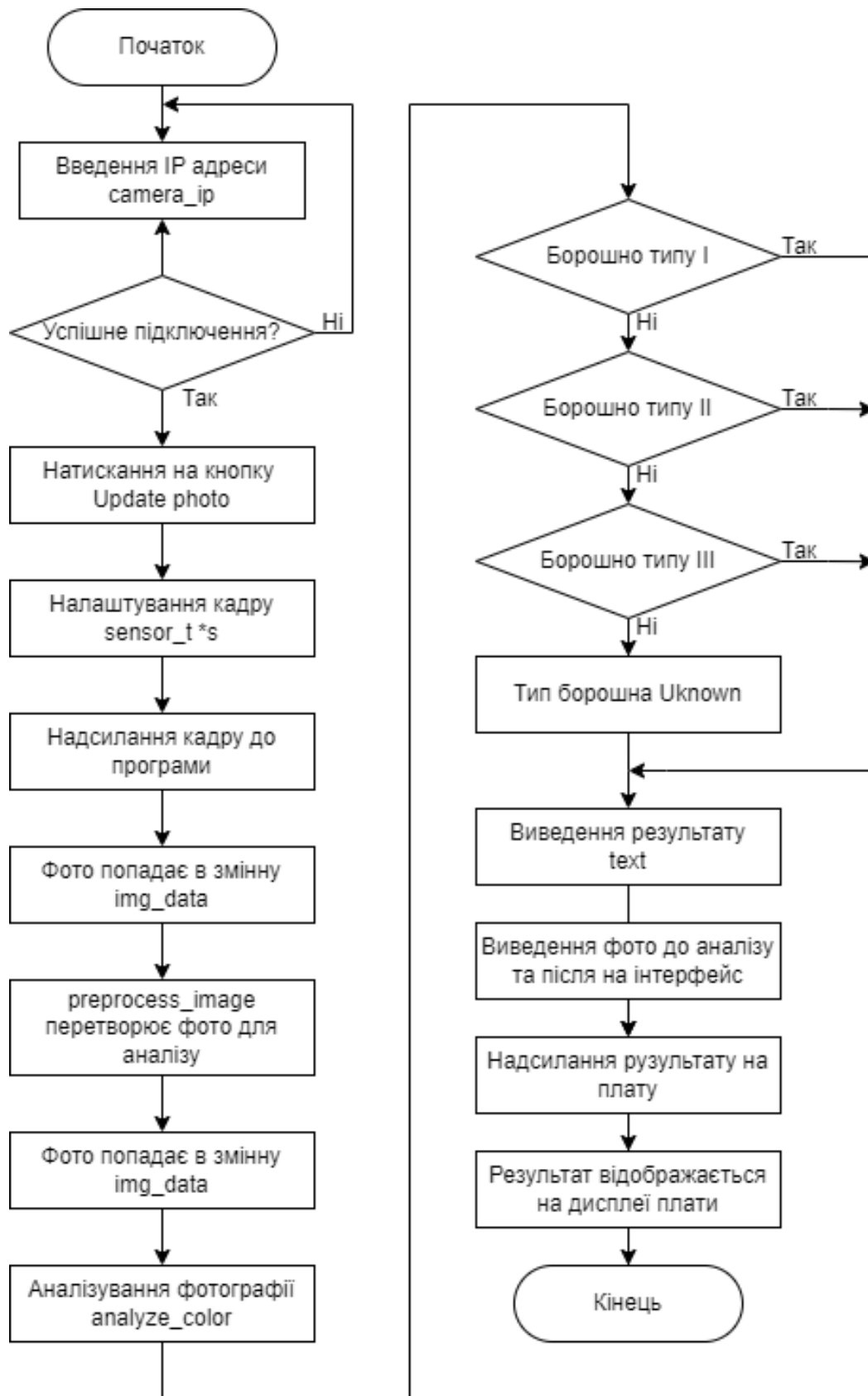


Рисунок 2.15 – Алгоритм роботи робототехнічної системи ESP32CAM

## 2.6 Висновки

У розділі обрано компоненти робототехнічної системи якості борошна та їх інтеграції. Ретельний відбір сумісних компонентів, включаючи плату ESP32CAM з модулем камери високої роздільної здатності, вбудованим Wi-Fi і потужним мікропроцесором, гарантує, що система може ефективно визначати якість борошна на основі візуального аналізу даних.

У розділі також наголошувалося на важливості врахування таких факторів, як енергоспоживання, надійність і вартість під час вибору компонентів. Беручи до уваги ці аспекти, система розроблена для ефективної, надійної роботи та в межах бюджетних обмежень.

Крім того, у розділі представлена робота програмного забезпечення та роль Arduino IDE для програмування мікроконтролерів і Python для обробки зображень на комп'ютері. Використання зв'язку Wi-Fi забезпечує безперебійну передачу даних і віддалений доступ до системи, підвищуючи її загальну функціональність і зручність для користувача.

У розділі було представлено алгоритм роботи робототехнічної системи, що може точно й ефективно оцінювати якість борошна, одночасно забезпечуючи економічно ефективне та універсальне рішення для різних застосувань.

### 3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ РОБОТОТЕХНІЧНОЇ СИСТЕМИ

#### 3.1 Схема робототехнічної системи визначення якості борошна

Модулі робототехнічної системи якості борошна, які були обрані в попередньому розділі, можна використовувати для створення макета. Плата ESP32CAM була обрана як основний компонент для системи, що забезпечує камеру високої роздільної здатності, можливості Wi-Fi і двоядерний мікропроцесор в одному компактному корпусі.

Компонування робототехнічної системи якості борошна включає [41]: плату ESP32CAM, OLED-дисплей (SSD1306, 128×64 пікселів) та інші необхідні компоненти. Макет робототехнічної системи якості борошна представлена на рисунку 3.1.

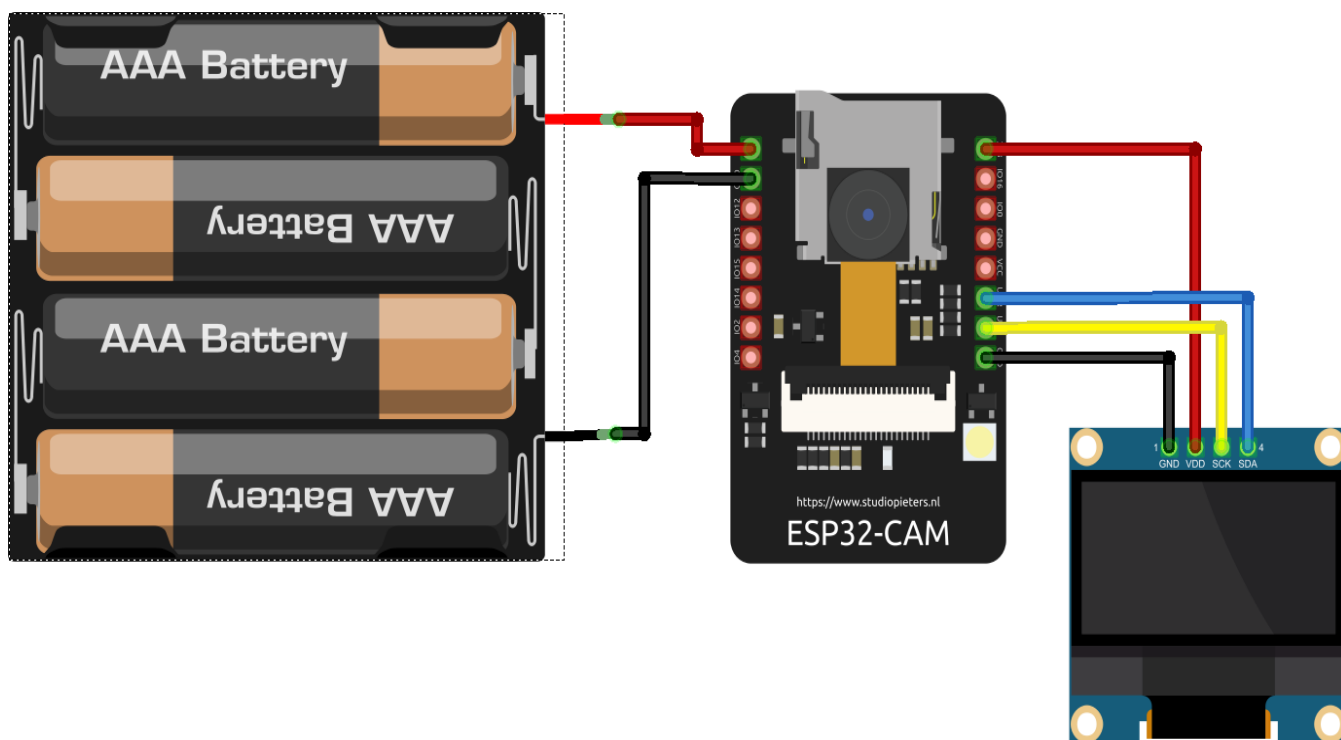


Рисунок 3.1 – Макет робототехнічної системи визначення якості борошна

Зм.	Арк.	№ докum.	Підпис	Дата

Плата ESP32CAM служить основним процесором і виконує такі завдання, як захоплення зображень, обробка даних і зв'язок з іншими пристроями чи мережами.

OLED-дисплей (SSD1306, 128×64 пікселів) використовується для відображення інформації про аналіз якості борошна, надаючи операторам зручний інтерфейс для моніторингу продуктивності системи.

Поєднуючи ці модулі, система може отримувати, обробляти та відображати відповідні дані, забезпечуючи безперебійний зв'язок з іншими пристроями чи мережами.

### 3.2 Встановлення необхідних бібліотек Python

Код Python для робототехнічної системи якості борошна відіграє вирішальну роль в обробці зображень та аналізі даних.

Щоб успішно реалізувати код, потрібно встановити різні бібліотеки, які надають певні функції. Тому необхідно розглянути деякі основні бібліотеки, їх призначення та те, як запускати код Python, який складається з кількох фрагментів [42].

OpenCV (бібліотека комп'ютерного зору з відкритим кодом) — це популярна бібліотека для комп'ютерного зору в реальному часі, машинного навчання та обробки зображень.

Бібліотека пропонує широкий спектр функцій для обробки зображень і відео, виявлення функцій і розпізнавання об'єктів.

У контексті робототехнічної системи якості борошна OpenCV використовується для обробки зображень, отриманих платою ESP32CAM, і виділення відповідних функцій для аналізу якості борошна [43].

На рисунку 3.2 зображено процес встановлення бібліотеки OpenCV для аналізу фото за допомогою комп'ютера.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18363.693]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Dhanush>pip install opencv-python
Collecting opencv-python
  Using cached opencv_python-4.2.0.32-cp37-cp37m-win_amd64.whl (33.0 MB)
Requirement already satisfied: numpy>=1.14.5 in c:\users\dhanush\appdata\local\pro
Installing collected packages: opencv-python
Successfully installed opencv-python-4.2.0.32

C:\Users\Dhanush>
```

Рисунок 3.2 – Встановлення OpenCV

NumPy (числовий Python) — це бібліотека для мови програмування Python, яка забезпечує підтримку великих багатовимірних масивів і матриць разом із набором математичних функцій для роботи з цими структурами. Це важливий інструмент для чисельних обчислень у Python і широко використовується в робототехнічній системі якості борошна для обробки зображень та аналізу даних. На рисунку 3.3 зображено процес встановлення бібліотеки NumPy [44].

```
Command Prompt
Microsoft Windows [Version 10.0.19044.1826]
(c) Microsoft Corporation. All rights reserved.

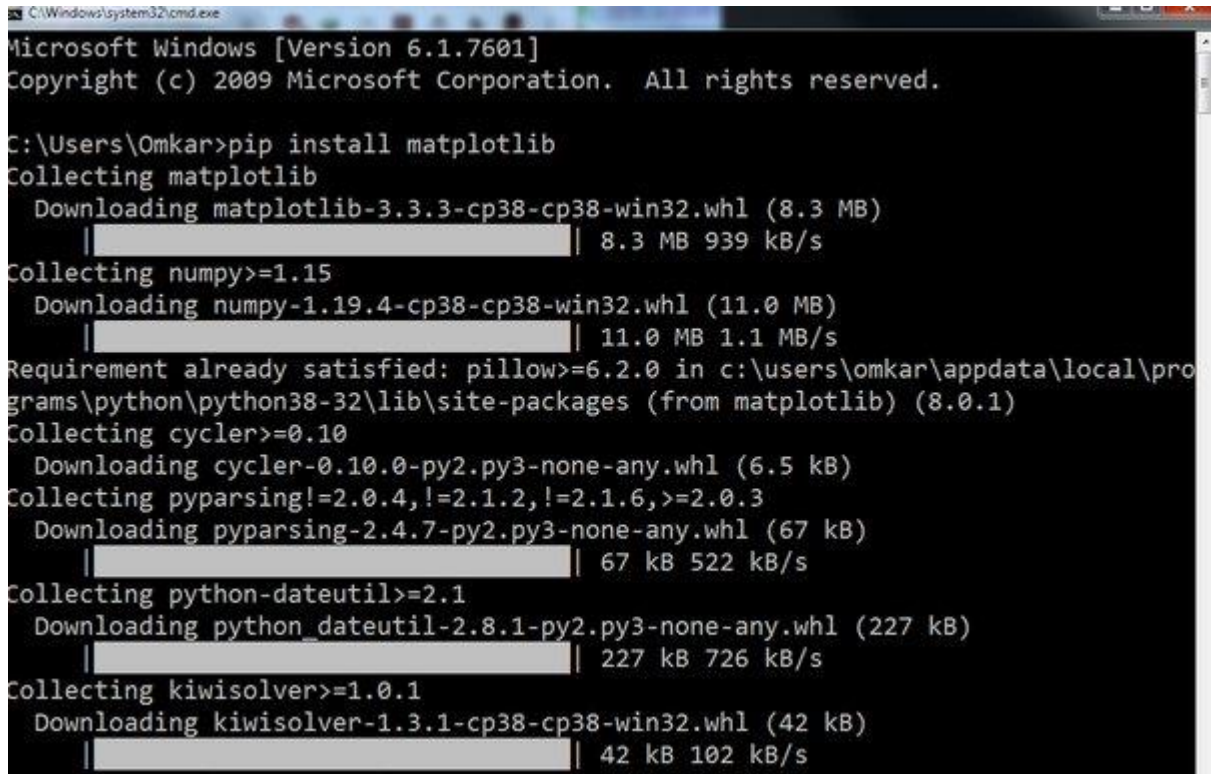
C:\Users\RADIK>py -m pip list
Package      Version
-----
pip          22.2.1
setuptools  63.2.0

C:\Users\RADIK>py -m pip install numpy
Collecting numpy
  Using cached numpy-1.23.1-cp310-cp310-win_amd64.whl (14.6 MB)
Installing collected packages: numpy
Successfully installed numpy-1.23.1

C:\Users\RADIK>
```

Рисунок 3.3 – Встановлення NumPy

Matplotlib — це бібліотека побудови графіків для мови програмування Python, яка надає об'єктно-орієнтований API для вбудовування графіків у програми. Він використовується в робототехнічній системі якості борошна для візуалізації результатів обробки зображень і аналізу даних, що дозволяє операторам краще зрозуміти продуктивність системи. На рисунку 3.4 зображено процес встановлення бібліотеки Matplotlib [45].



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Omkar>pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.3.3-cp38-cp38-win32.whl (8.3 MB)
    |-----| 8.3 MB 939 kB/s
Collecting numpy>=1.15
  Downloading numpy-1.19.4-cp38-cp38-win32.whl (11.0 MB)
    |-----| 11.0 MB 1.1 MB/s
Requirement already satisfied: pillow>=6.2.0 in c:\users\omkar\appdata\local\programs\python\python38-32\lib\site-packages (from matplotlib) (8.0.1)
Collecting cycler>=0.10
  Downloading cycler-0.10.0-py2.py3-none-any.whl (6.5 kB)
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3
  Downloading pyparsing-2.4.7-py2.py3-none-any.whl (67 kB)
    |-----| 67 kB 522 kB/s
Collecting python-dateutil>=2.1
  Downloading python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
    |-----| 227 kB 726 kB/s
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.3.1-cp38-cp38-win32.whl (42 kB)
    |-----| 42 kB 102 kB/s
```

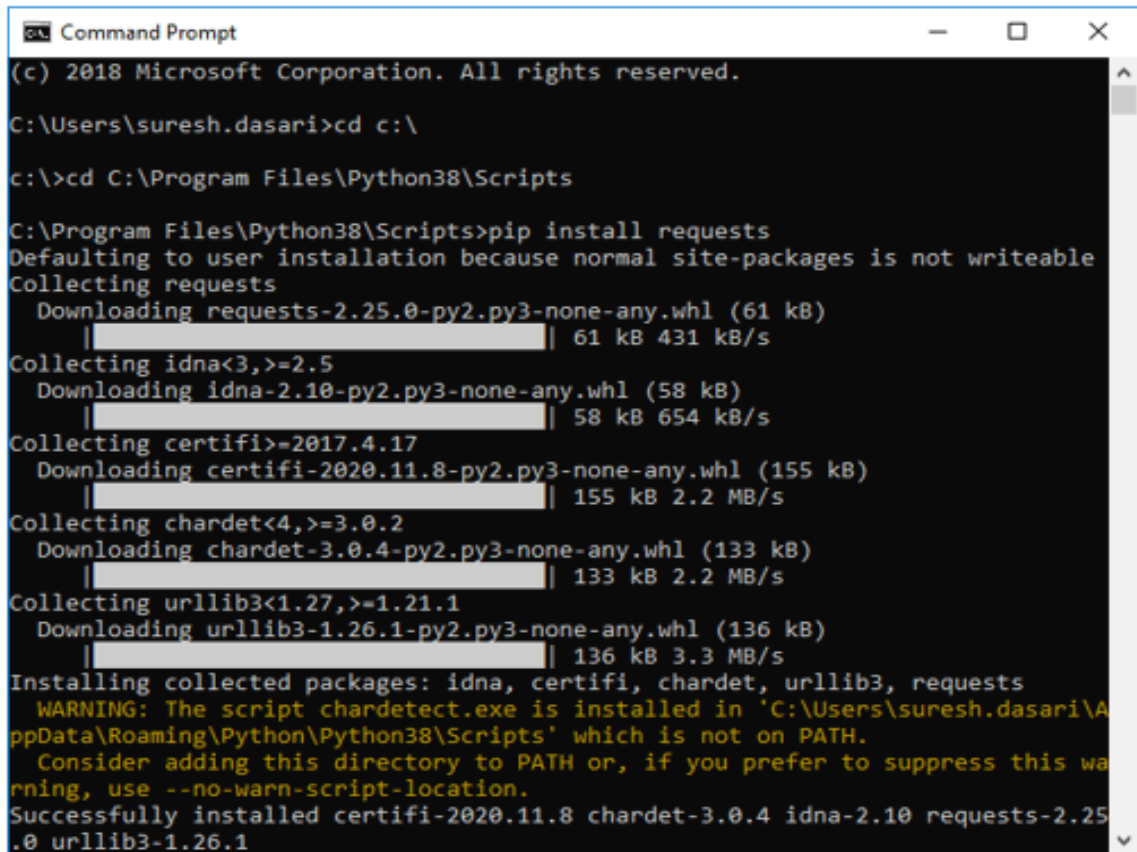
Рисунок 3.4 – Встановлення Matplotlib

Додатковою важливою бібліотекою для робототехнічної системи якості борошна є запити. Бібліотека запитів — це популярний модуль Python, який використовується для створення HTTP-запитів. Він спрощує надсилання запитів та є необхідною для ефективного взаємодії в робототехнічній системі. Цей модуль Python спрощує відправку різних типів HTTP-запитів, таких як GET, POST, PUT, DELETE, та обробку отриманих відповідей [46].

Завдяки бібліотеці запитів, можна легко надсилати зображення, отримані з ESP32CAM, або оброблені дані на комп'ютер для подальшого аналізу та зберігання.

Це робить бібліотеку запитів необхідним інструментом для робототехнічної системи, які вимагають обміну даними з веб-службами та АРІ для виконання різних завдань, у тому числі для контролю якості борошна [47].

На рисунку 3.5 зображено процес встановлення бібліотеки запитів для подальшого її використання.



```
Command Prompt
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\suresh.dasari>cd c:\
c:\>cd C:\Program Files\Python38\Scripts
C:\Program Files\Python38\Scripts>pip install requests
Defaulting to user installation because normal site-packages is not writeable
Collecting requests
  Downloading requests-2.25.0-py2.py3-none-any.whl (61 kB)
  |-----| 61 kB 431 kB/s
Collecting idna<3,>=2.5
  Downloading idna-2.10-py2.py3-none-any.whl (58 kB)
  |-----| 58 kB 654 kB/s
Collecting certifi>=2017.4.17
  Downloading certifi-2020.11.8-py2.py3-none-any.whl (155 kB)
  |-----| 155 kB 2.2 MB/s
Collecting chardet<4,>=3.0.2
  Downloading chardet-3.0.4-py2.py3-none-any.whl (133 kB)
  |-----| 133 kB 2.2 MB/s
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.1-py2.py3-none-any.whl (136 kB)
  |-----| 136 kB 3.3 MB/s
Installing collected packages: idna, certifi, chardet, urllib3, requests
  WARNING: The script chardetect.exe is installed in 'C:\Users\suresh.dasari\AppData\Roaming\Python\Python38\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed certifi-2020.11.8 chardet-3.0.4 idna-2.10 requests-2.25.0 urllib3-1.26.1
```

Рисунок 3.5 – Встановлення бібліотеки запитів

Щоб запустити код Python, необхідно виконати наступні дії:

- завантажити останню версію Python з офіційного сайту [48], (рисунок 3.6);
- встановити необхідні бібліотеки за допомогою команд;
- зберегти код Python у файлі з розширенням .py (наприклад, flour\\_quality\\_system.py);
- відкрити термінал або командний рядок і перейти до каталогу, де збережено код Python;

- запусити код Python за допомогою такої команди: `python flour_quality_system.py`

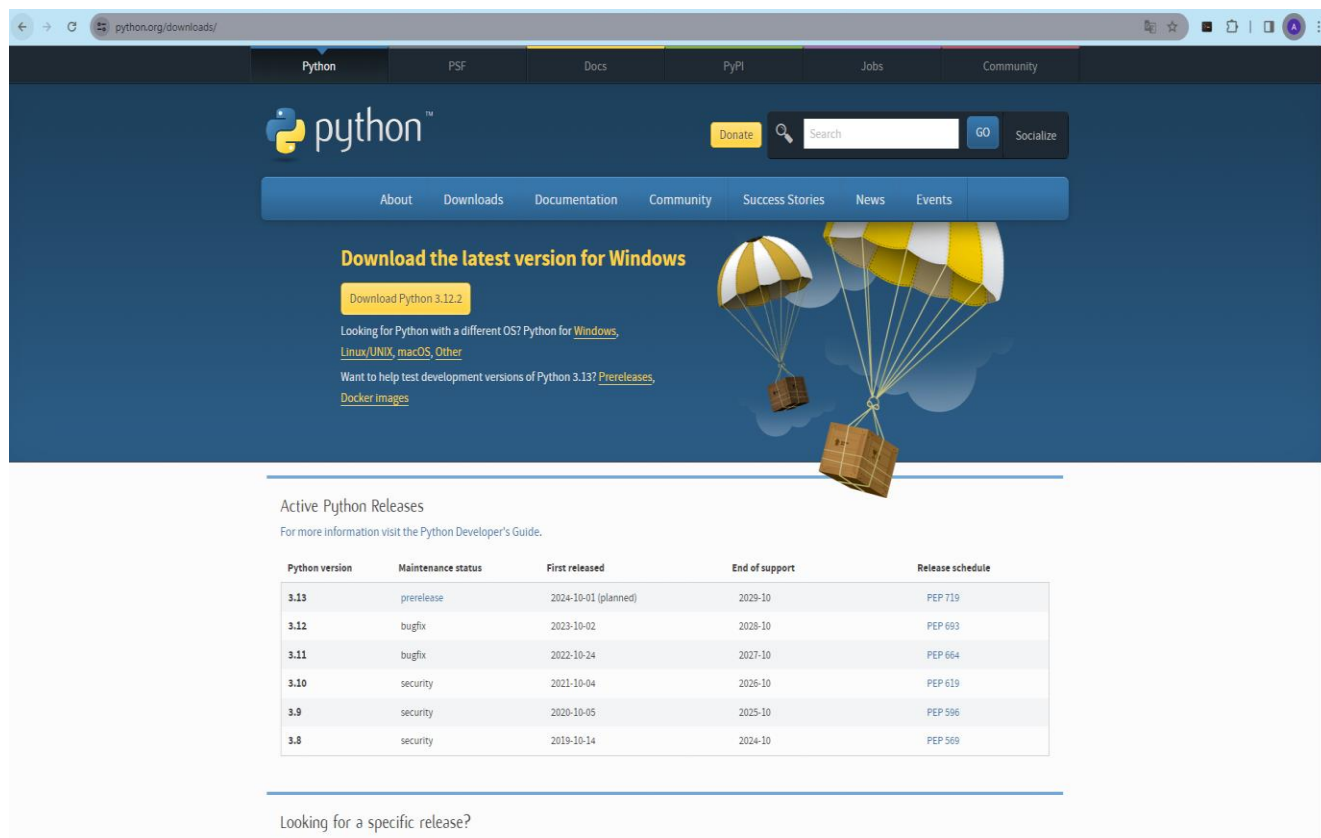


Рисунок 3.6 – Офіційний сайт Python

### 3.3 Встановлення необхідних ресурсів для ArduinoIDE

Щоб запрограмувати плату ESP32CAM і максимально використати її функції, важливо встановити необхідні ресурси для інтегрованого середовища розробки Arduino (IDE) [49].

Перш ніж запрограмувати плату ESP32CAM за допомогою Arduino IDE, потрібно встановити визначення плати ESP32. Необхідно виконати наступні кроки, щоб додати визначення плати до Arduino IDE (рисунок 3.7):

– відкрити Arduino IDE і перейти до вікна «Параметри», натиснувши «Файл» і далі «Параметри»;

- у вікні «Параметри» знайти поле «Additional Boards Manager URLs» і натиснути кнопку поруч із ним;
- ввести таку URL-адресу в нове поле, що з'явиться:  
[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_dev\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json);
- натиснути «Ок», щоб закрити вікно «Параметри» та зберегти зміни;

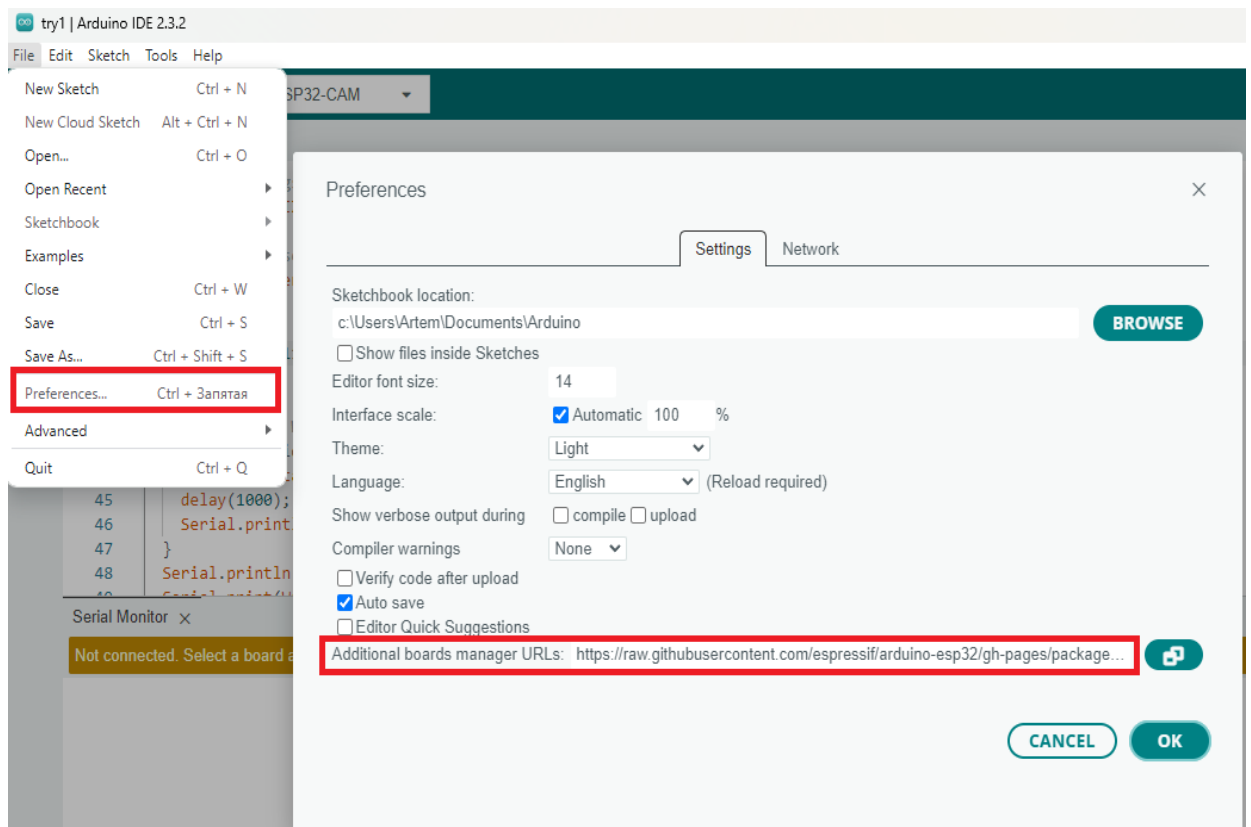


Рисунок 3.7 – Визначення плати

- перейти до «Інструменти» > «Дошка» > «Менеджер дошок»;
- у вікні «Boards Manager» знайти «ESP32» і встановити пакет «ESP32 by Espressif Systems» (рисунок 3.8).
- після завершення встановлення необхідно обрати «AI-Thinker ESP32-CAM» як плату, перейшовши до «Інструменти» > «Плата» > «AI-Thinker ESP32-CAM» (рисунок 3.9).

Щоб використовувати різні функції та функції плати ESP32CAM, необхідно встановити спеціальні бібліотеки [50].

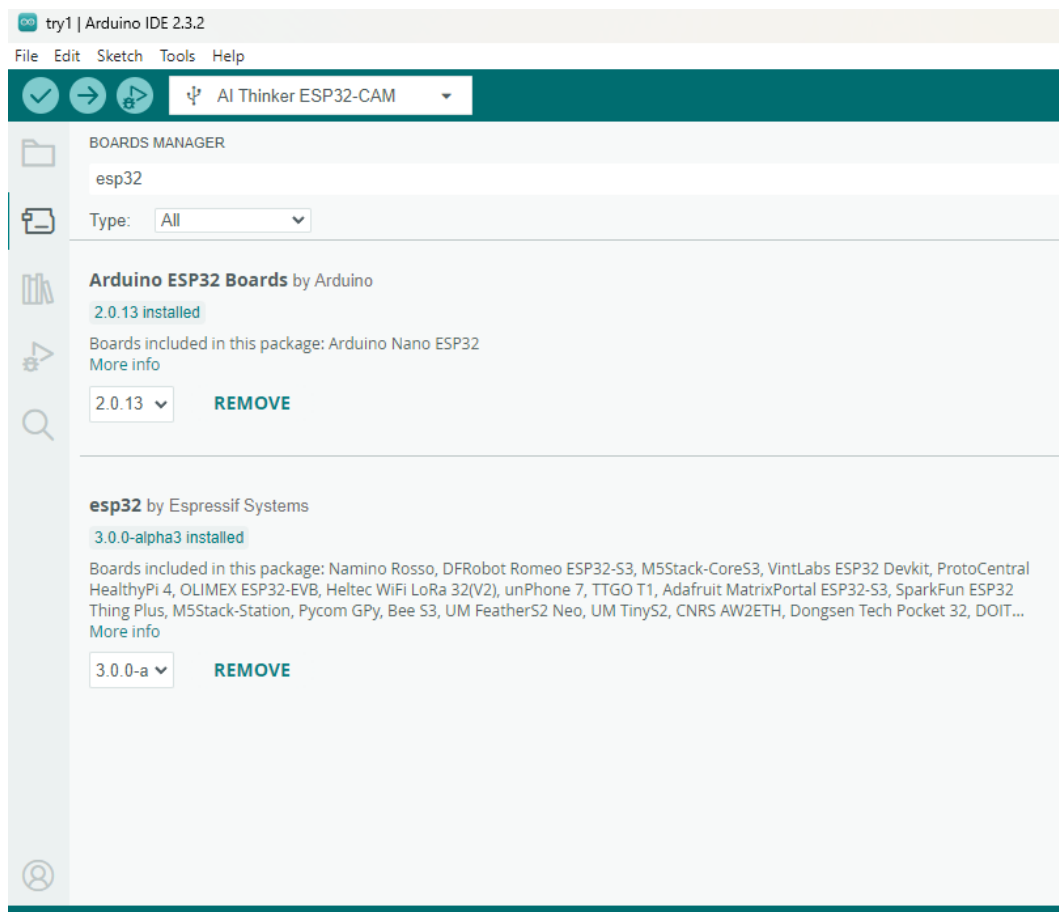


Рисунок 3.8 – Встановлення ESP32 by Espressif Systems

ArduinoJson — це популярна легка бібліотека JSON, спеціально розроблена для Arduino та інших вбудованих платформ. Це важливий інструмент для проєктів, які вимагають ефективного синтаксичного аналізу та генерації даних JSON, що є поширеним форматом даних, що використовується в програмах IoT, API та зв'язку між пристроями.

Бібліотека пропонує простий та інтуїтивно зрозумілий API, що полегшує роботу зі структурами даних JSON у вашому коді.

Бібліотека ESP32CAM — це потужний і універсальний інструмент, спеціально розроблений для плати ESP32-CAM. Він надає широкий спектр пов'язаних із камерою функцій і можливостей, що дозволяє легко отримувати

доступ до модуля камери та керувати ним, знімати зображення та транслювати відео [51].

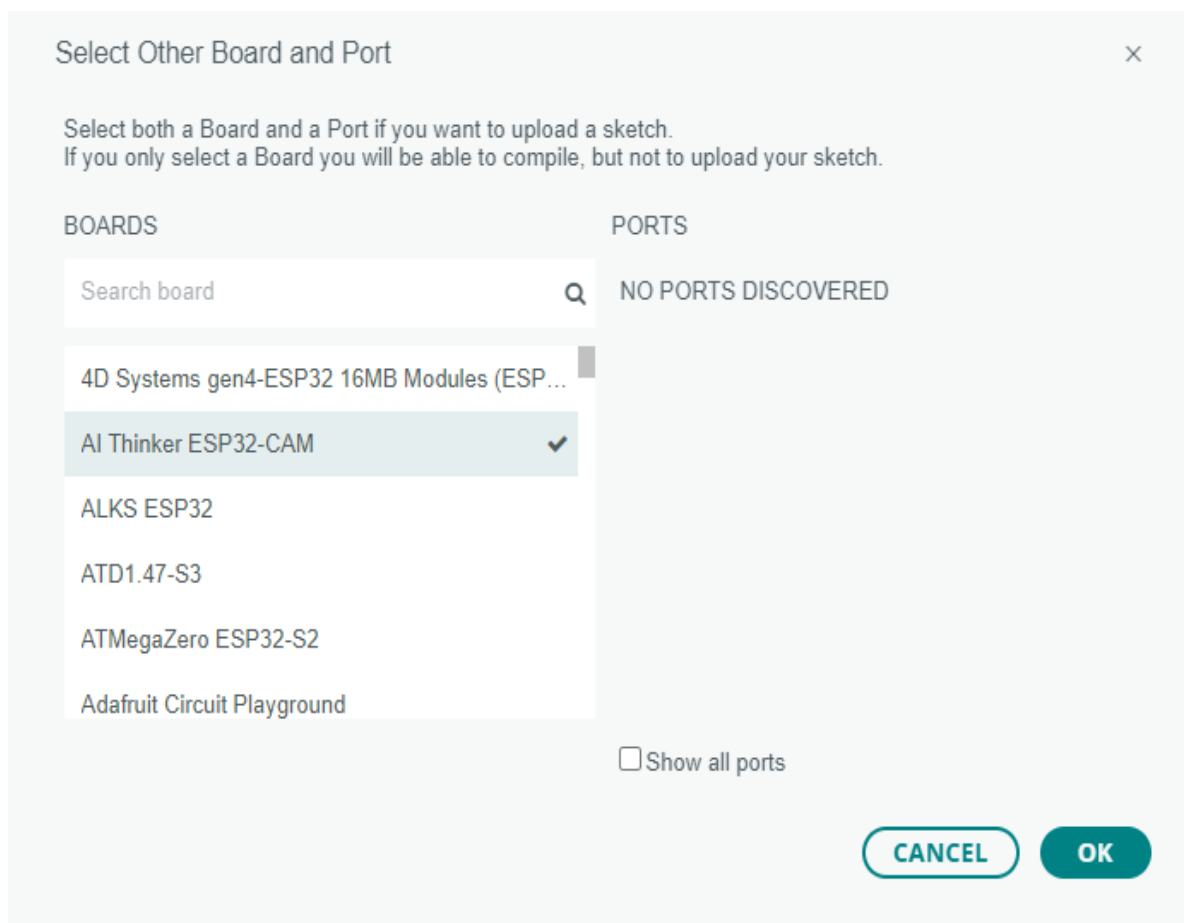


Рисунок 3.9 – Вибір плати

Бібліотека WiFi — це вбудована бібліотека для Arduino IDE, яка забезпечує підключення до Wi-Fi для плат на основі ESP32.

Бібліотека Wi-Fi необхідна для проєктів, які вимагають мережевого зв'язку, наприклад надсилання даних на сервер, підключення до хмарних служб або дистанційне керування пристроями.

Для коректної роботи необхідно установити ці бібліотеки, виконуючи наступні дії:

– в Arduino IDE перейдіть до «Sketch» > «Include Library» > «Manage Libraries»;

– у вікні «Менеджер бібліотек» необхідно знайти та встановити згадані вище бібліотеки та встановить їх одну за одною (рисунок 3.10).

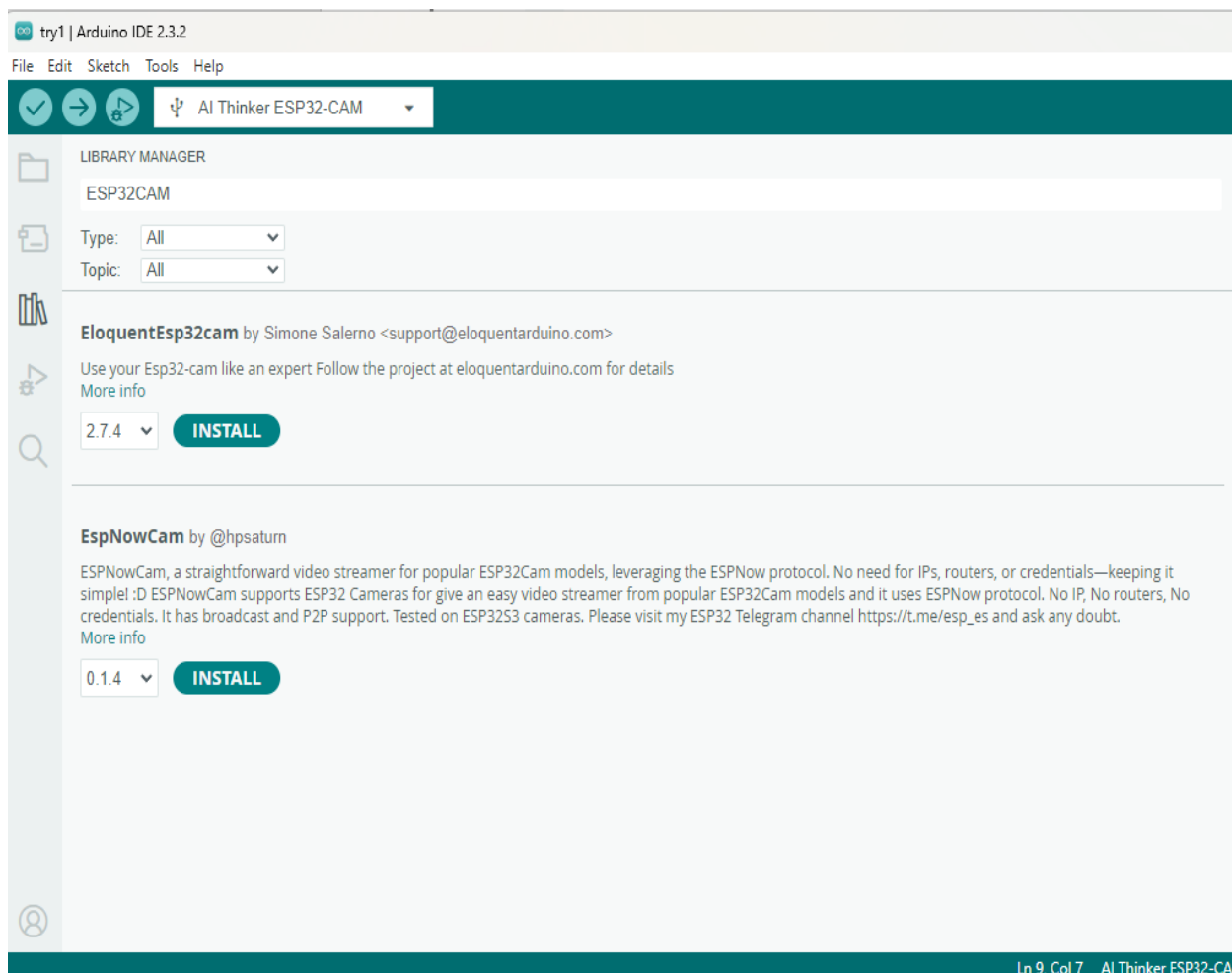


Рисунок 3.10 – Встановлення бібліотек Arduino IDE

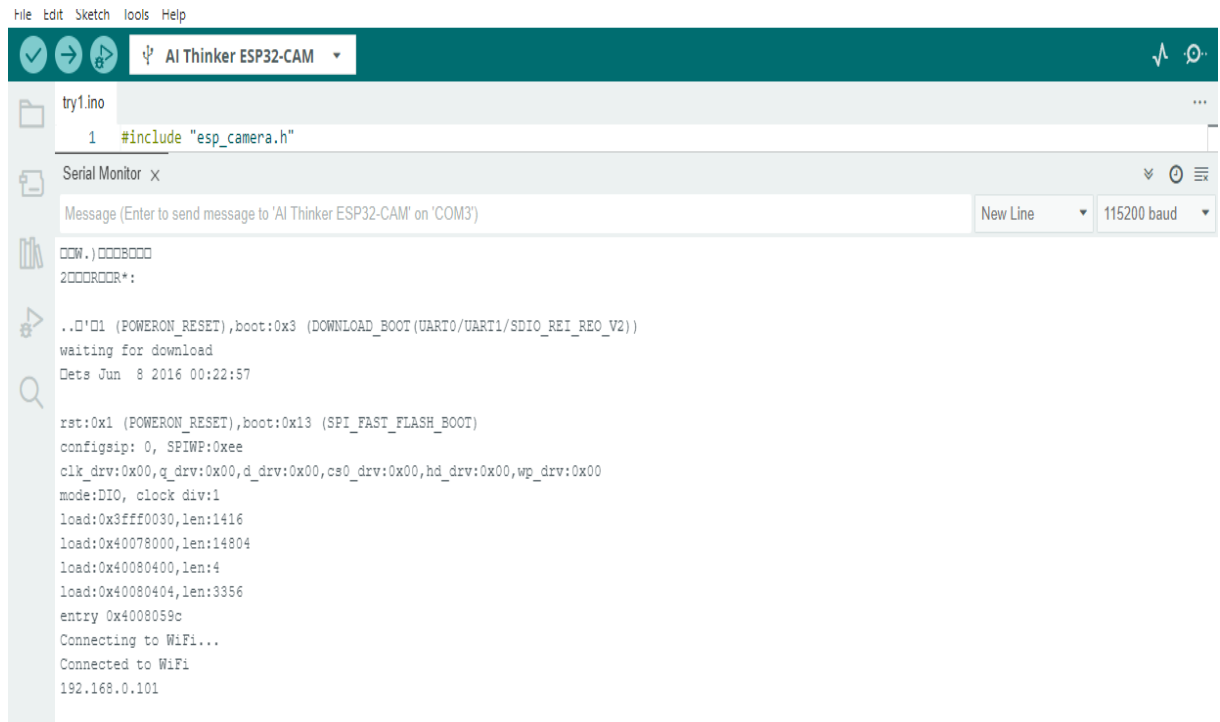
Таким чином, ці бібліотеки ArduinoJson, ESP32CAM і WiFi – є важливими інструментами для розробки програм для плати ESP32CAM. Вони забезпечують надійну основу для роботи з даними JSON, доступу до функцій камери та встановлення з’єднання Wi-Fi відповідно [52].

Після встановлення необхідних визначень плати та бібліотек необхідно обрати правильний порт для плати ESP2-CAM. Для цього потрібно виконати наступні дії:

– підключити плату ESP32CAM до комп’ютера за допомогою програматора FTDI або плати програмування ESP32CAM-MB;

– в Arduino IDE перейти до «Інструменти» > «Порт» і обрати порт, пов'язаний із платою ESP32-CAM.

Тепер можливо писати, компілювати та завантажувати код на плату ESP32CAM. Після всіх дій та завантаження коду на плату буде отримано код, який підтверджує успішне встановлення зв'язку (рисунок 3.11).



```
try1.ino
1 #include "esp_camera.h"

Serial Monitor x
Message (Enter to send message to 'AI Thinker ESP32-CAM' on 'COM3')
New Line 115200 baud

DOW.)000B0000
2000R00R+:

..D'01 (POWERON_RESET),boot:0x3 (DOWNLOAD_BOOT(UART0/UART1/SDIO_REI_REO_V2))
waiting for download
Dets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1416
load:0x40078000,len:14804
load:0x40080400,len:4
load:0x40080404,len:3356
entry 0x4008059c
Connecting to WiFi...
Connected to WiFi
192.168.0.101
```

Рисунок 3.11 – Встановлення зв'язку з платою

На рисунку 3.11 відображено, що код успішно спрацював та зміг підключитися до мережі Wi-Fi та отримав IP адрес, який відобразив в повідомленні, а саме 192.168.0.101.

### 3.4 Основна програмна частина Arduino

Розроблено код для робототехнічної системи якості борошна на основі ESP32CAM, який можна розбити на кілька сегментів, кожен з яких відповідає за окрему функцію [53].

В першу чергу, необхідно підключити бібліотеки, які були встановлені раніше. Для цього вказати що використано їх в коді за допомогою наступного коду:

```
#include "esp_camera.h"
#include "ArduinoJson.h"
#include <WiFi.h>
#include <Adafruit_SSD1306.h>
```

Так, як проєкт використовує Wi-Fi, то необхідно вказати код для підключення до мережі. WiFi SSID і пароль для підключення плати ESP32CAM до мережі WiFi, який визначається за допомогою наступного коду:

```
const char* ssid = "назва мережі";
const char* password = "пароль від мережі";
```

Обов'язковим є заповнення полів вірними фактичними обліковими даними WiFi.

Далі необхідно використати код, який визначає властивості та ініціалізацію OLED-дисплея за допомогою бібліотеки Adafruit\_SSD1306.

```
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
```

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);
```

Рядки, SCREEN\_WIDTH 128 і SCREEN\_HEIGHT 64, визначають ширину та висоту OLED-дисплея як 128 пікселів і 64 пікселів відповідно. Ці константи використовуються пізніше в коді для встановлення розмірів дисплея.

Третій рядок, визначає контакт скидання OLED як -1, вказуючи, що контакт скидання не підключено до жодного конкретного контакту GPIO на платі ESP32CAM, і бібліотека не повинна виконувати апаратне скидання на дисплеї.

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET) ініціалізує екземпляр класу Adafruit_SSD1306 під назвою «display»
```

із попередньо визначеними константами SCREEN\_WIDTH, SCREEN\_HEIGHT і OLED\_RESET і вказівником на інтерфейс I2C.

Для правильного визначення директиви препроцесора необхідно її правильно оголосити. Для того, щоб зробити це вказується наступний код:

```
#define CAMERA_MODEL_AI_THINKER
```

Це означає, що код налаштовано для модуля ESP32CAM від AI Thinker.

Визначення контактів GPIO для моделі камери AI-THINKER є важливим елементом у функціонуванні робототехнічної системи.

В умовному блоці, керованому #if defined(CAMERA\_MODEL\_AI\_THINKER), код визначає контакти GPIO, пов'язані з цією конкретною моделлю камери:

```
#if defined(CAMERA_MODEL_AI_THINKER)
```

```
#define PWDN_GPIO_NUM 32
```

```
#define RESET_GPIO_NUM -1
```

```
#define XCLK_GPIO_NUM 0
```

```
#define SIOD_GPIO_NUM 26
```

```
#define SIOC_GPIO_NUM 27
```

```
#define Y9_GPIO_NUM 35
```

```
#define Y8_GPIO_NUM 34
```

```
#define Y7_GPIO_NUM 39
```

```
#define Y6_GPIO_NUM 36
```

```
#define Y5_GPIO_NUM 21
```

```
#define Y4_GPIO_NUM 19
```

```
#define Y3_GPIO_NUM 18
```

```
#define Y2_GPIO_NUM 5
```

```
#define VSYNC_GPIO_NUM 25
```

```
#define HREF_GPIO_NUM 23
```

```
#define PCLK_GPIO_NUM 22
```

Ці контакти відповідають за керування різними функціями модуля камери, дозволяючи системі ефективно взаємодіяти з камерою та отримувати високоякісні зображення для аналізу якості борошна.

Визначені контакти керують основними функціями, такими як вимкнення живлення, скидання, годинник, дані та сигнали синхронізації, забезпечуючи плавний зв'язок і роботу між платою ESP32CAM і модулем камери. Завдяки точному визначенню цих штифтів код закладає основу для успішного захоплення та обробки зображень.

Код створює екземпляр сервера Wi-Fi на порту 80 та 81. Порт 80 використовуватиметься для надсилання захоплених зображень клієнтам, а порт 81 для отримання результатів від коду Python:

```
WiFiServer controlServer(81); // Use a different port for control, e.g., 81
WiFiServer server(80);
```

Для блоку, код визначає функцію, яка відповідає за керування вхідними з'єднаннями від керуючих клієнтів, читання команд і їх відображення на OLED-дисплеї.

```
void handleControlClient() {
    // Wait for a control client to connect
    WiFiClient controlClient = controlServer.available();
    if (!controlClient) {
        return;
    }
    // Read the command
    String command = controlClient.readStringUntil('\n');
    command.trim();
    // Display the flour type on the OLED display
    display.clearDisplay();
    display.setCursor(0, 0);
    display.println(command);
}
```

```
display.display();  
  
// Close the control client  
controlClient.stop(); }
```

Після здійснення всіх налаштувань та підключення основного коду необхідно здійснити компіляцію та завантаження коду на саму плату. Для цього необхідно натиснути на кнопку Upload, як вказано на рисунку 3.12.

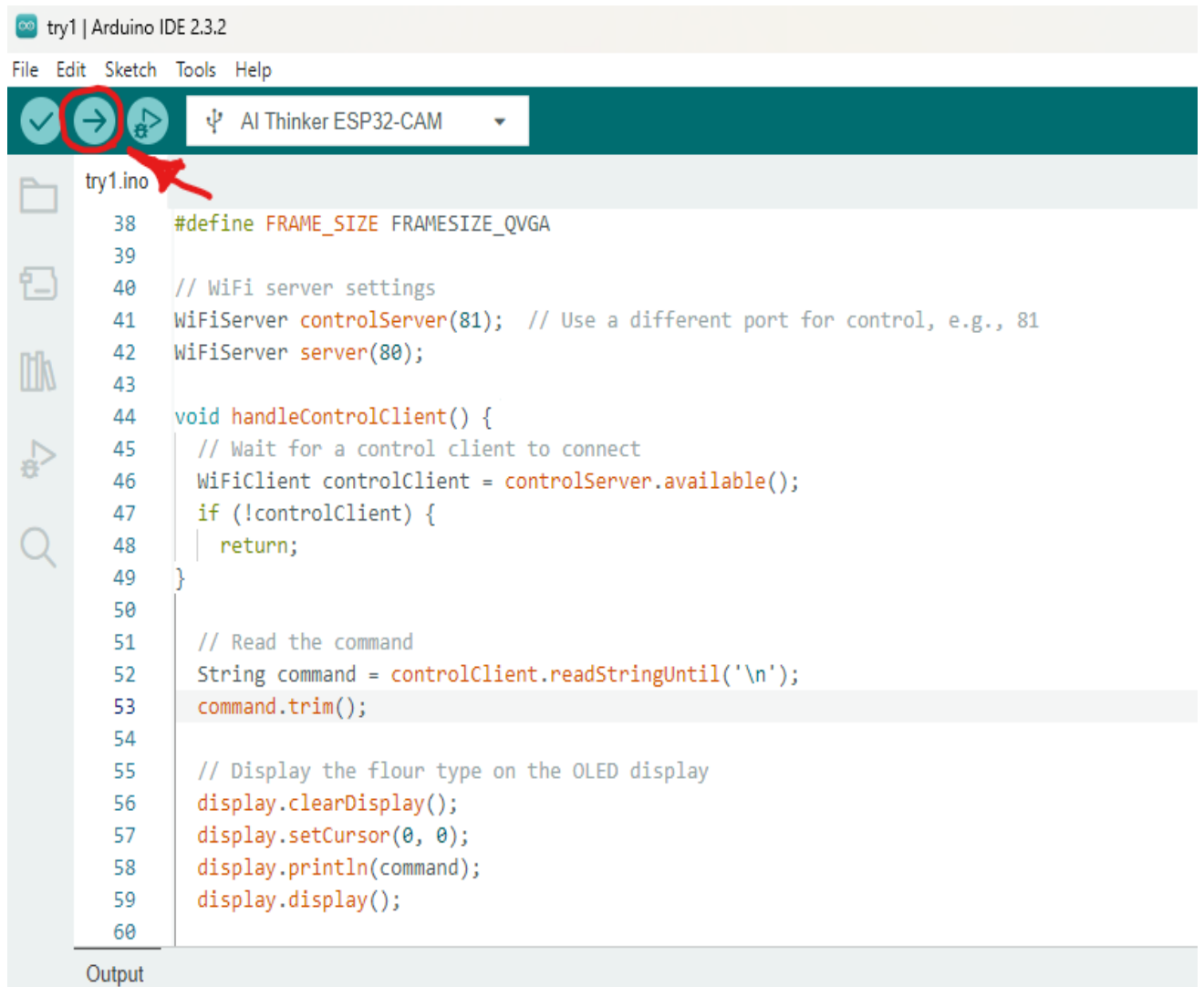
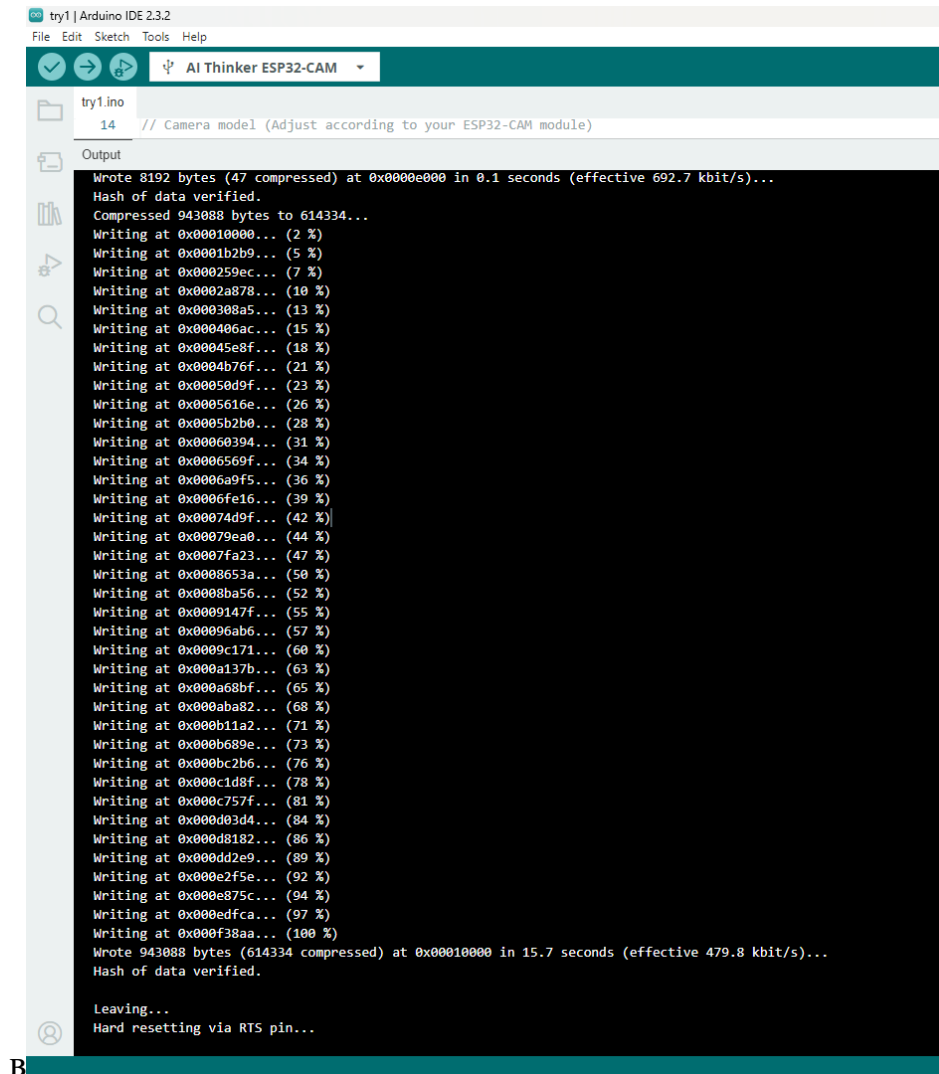


Рисунок 3.12 – Завантаження коду на плату

Після натискання кнопки вивантаження коду на плати запуститься процес компіляції коду з наступним його завантаження на плату ESP32CAM. З'явиться вікно яке буде повідомляти про хід процесу його завантаження на плату

ESP32CAM та про його успішність. Приклад вікна ходу завантаження зображено на рисунку 3.13.



```
try1 | Arduino IDE 2.3.2
File Edit Sketch Tools Help
AI Thinker ESP32-CAM
try1.ino
14 // Camera model (Adjust according to your ESP32-CAM module)
Output
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.1 seconds (effective 692.7 kbit/s)...
Hash of data verified.
Compressed 943088 bytes to 614334...
Writing at 0x00010000... (2 %)
Writing at 0x0001b2b9... (5 %)
Writing at 0x000259ec... (7 %)
Writing at 0x0002a878... (10 %)
Writing at 0x000308a5... (13 %)
Writing at 0x000406ac... (15 %)
Writing at 0x00045e8f... (18 %)
Writing at 0x0004b76f... (21 %)
Writing at 0x00050d9f... (23 %)
Writing at 0x0005616e... (26 %)
Writing at 0x0005b2b0... (28 %)
Writing at 0x00060394... (31 %)
Writing at 0x0006569f... (34 %)
Writing at 0x0006a9f5... (36 %)
Writing at 0x0006fe16... (39 %)
Writing at 0x00074d9f... (42 %)
Writing at 0x00079ea0... (44 %)
Writing at 0x0007fa23... (47 %)
Writing at 0x0008653a... (50 %)
Writing at 0x0008ba56... (52 %)
Writing at 0x0009147f... (55 %)
Writing at 0x00096ab6... (57 %)
Writing at 0x0009c171... (60 %)
Writing at 0x000a137b... (63 %)
Writing at 0x000a68bf... (65 %)
Writing at 0x000abab2... (68 %)
Writing at 0x000b11a2... (71 %)
Writing at 0x000b689e... (73 %)
Writing at 0x000bc2b6... (76 %)
Writing at 0x000c1d8f... (78 %)
Writing at 0x000c757f... (81 %)
Writing at 0x000d03d4... (84 %)
Writing at 0x000d8182... (86 %)
Writing at 0x000dd2e9... (89 %)
Writing at 0x000e2f5e... (92 %)
Writing at 0x000e875c... (94 %)
Writing at 0x000edfca... (97 %)
Writing at 0x000f38aa... (100 %)
Wrote 943088 bytes (614334 compressed) at 0x00010000 in 15.7 seconds (effective 479.8 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
```

Рисунок 3.13 – Успішне завантаження коду

Після завантаження коду, потрібно натиснути на кнопку RESET на платі і ESP32CAM буде готова до використання в проекті.

### 3.5 Основна програмна частина Python

Програмна частина на Python аналізує колір борошна з фотографії, яку надсилає плата ESP32CAM і відображає результати в графічному інтерфейсі

користувача та відправляє результат на плату. Програма розділена на кілька функцій і блоків [54].

В першу чергу необхідно імпортувати бібліотеки, таких як OpenCV (cv2), NumPy, socket, PIL (бібліотека зображень Python), tkinter і потоки. Для цього необхідно вказати наступний код [55]:

```
import cv2
import numpy as np
import socket
from PIL import Image, ImageTk
import tkinter as tk
from threading import Thread
from PIL.Image import Resampling # Import Resampling from PIL.Image
```

Ці бібліотеки використовуються для обробки зображень, роботи в мережі, створення графічного інтерфейсу користувача та багатопоточності.

Далі йде функція `def preprocess_image(img)`. Вона приймає зображення як вхідні дані, далі змінює його розмір до 300×300 пікселів та перетворює його на колірний простір HSV і гарантує, що на виході буде масив NumPy. Колірний простір HSV полегшує ідентифікацію конкретних кольорів для системи, наприклад білий (борошняний) [56].

Для самого аналізу фото на якість борошна використовується функція `def analyze_color(hsv)`. Данна функція аналізує попередньо оброблене зображення HSV, щоб визначити тип борошна.

Функція встановлює пороги кольору для білого борошна та створює маску, яка необхідна для розпізнавання білого кольору за допомогою наступного коду:

```
# Визначте порогові значення кольору для білого борошна
lower_white = np.array([0, 0, 200])
upper_white = np.array([180, 30, 255])
# Створення маски для білого кольору (борошна)
mask = cv2.inRange(hsv, lower_white, upper_white)
```

```

# Обчислення середніх значень HSV у замаскованій області
avg_hsv = cv2.mean(hsv, mask=mask)
avg_h, avg_s, avg_v, _ = avg_hsv
# Визначте тип борошна на основі середнього значення яскравості.
flour_type = "
if avg_v >= 245:
    flour_type = 'Type 1 – Very White'
elif avg_v >= 230:
    flour_type = 'Type 2 – White'
elif avg_v >= 215:
    flour_type = 'Type 3 – Off-White'
else:
    flour_type = 'Unknown'

```

Виконується обчислення середнього значення HSV у області маски та визначає тип борошна на основі середнього значення яскравості. В кінці функція надсилає команду на плату ESP32CAM про тип борошна за допомогою клієнта TCP і повертає тип борошна та середнє значення яскравості [57].

Для вірного функціонування графічної оболонки та можливості оновлення фотографії для аналізу при натисканні кнопки використовується функція `def update_image()`. Данна функція відповідає за захоплення зображення з плати ESP32CAM, його попередню обробку, аналіз кольору за допомогою попередніх функцій та оновлення GUI новим зображенням і результатами [58].

Функція підключається до ESP32CAM за допомогою TCP-клієнта, отримує необроблені дані зображення JPEG, декодує їх і попередньо обробляє. Потім викликає функцію `analyze_color` і оновлює графічний інтерфейс новим зображенням і типом муки [59].

Ця функція створює графічний інтерфейс `tkinter`, який містить кнопку «Оновити фото», текстове поле для введення IP-адреси, мітку результату та дві мітки зображення (одну для оригінального зображення та одну для

проаналізованого зображення). Він також налаштовує макет за допомогою рамок. Графічний інтерфейс користувача оновлюється початковим зображенням і запускає цикл для періодичного оновлення відображених зображень [60].

Сценарій виконується шляхом виклику функції `main` під час запуску як основного модуля (`if \_\_name\_\_ == "\_\_main\_\_":`). Це створює GUI, підключається до плати ESP32-CAM і запускає процес аналізу та відображення зображення.

Після налаштування коду виконується його запуск. Вигляд графічного інтерфейсу програми зображено на рисунку 3.14.

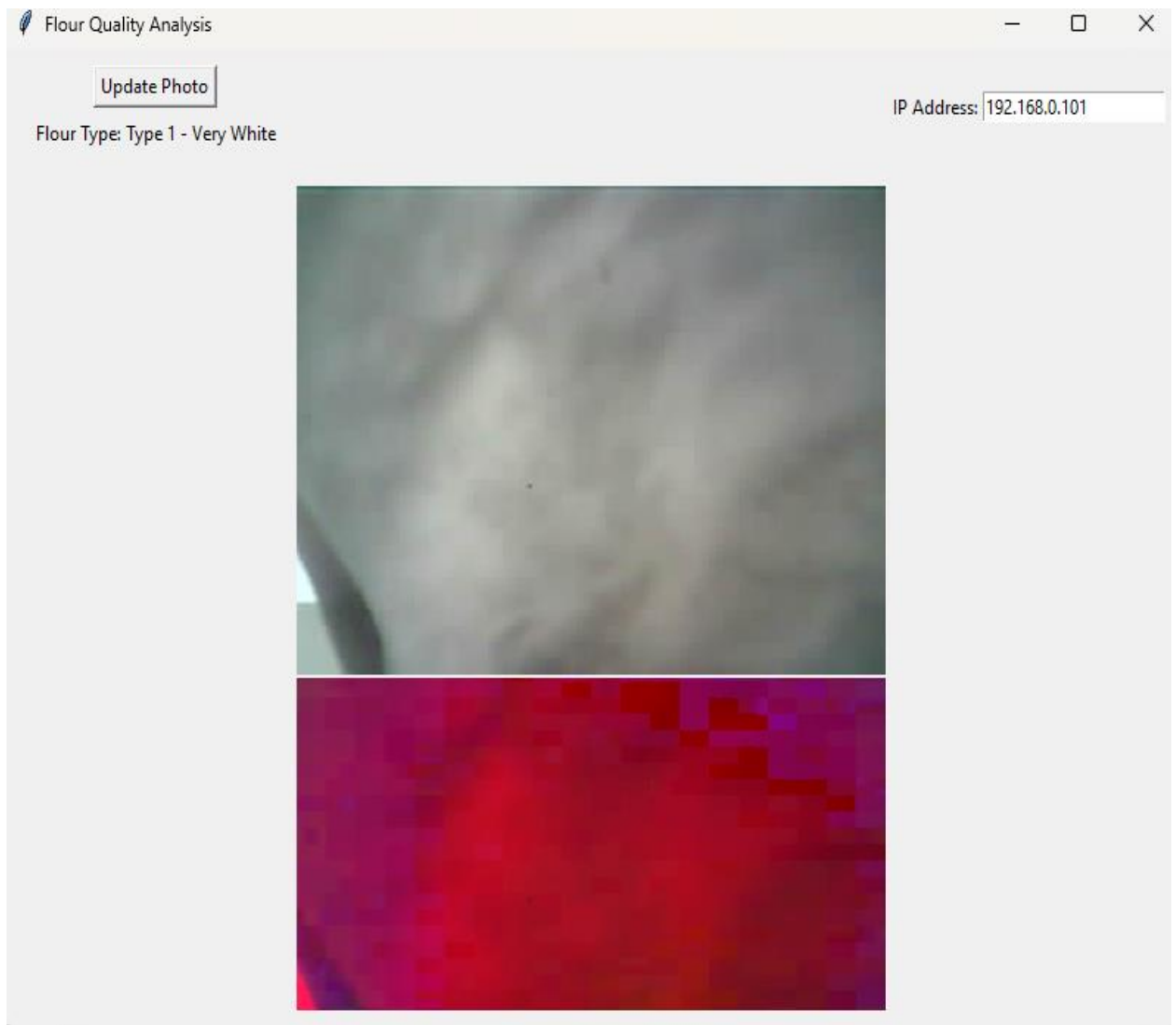


Рисунок 3.14 – Приклад інтерфейсу програми Phytom

На графічному інтерфейсі відображається кнопка для оновлення фотографій, поле для введення адреси плати, яку вона отримала після підключення, та самі зображення. Верхнє зображення являється оригінальним, а нижнє показує його вигляд після здійснення всіх маніпуляцій з метою аналізу.

### 3.6 Розрахунок матеріальних витрат

Для детального розрахунку матеріальних витрат, пов'язаних зі створенням робототехнічної системи якості борошна на основі плати ESP32CAM, необхідно визначити деталі, які були використані для системи:

- плата з мікроконтролером ESP32CAM;
- камера OV2640 ;
- батарейний відсік із вимикачем 3xAA;
- лужні батареї Duracell AA (LR06) MN1500;
- дисплей OLED S0.96" I2C 128x64 (синій);
- програматор ESP32CAM-MB;
- пластик для корпусу.

Вартість кожного компонента може відрізнитися в залежності від постачальника і конкретної обраної моделі.

Вартість компонентів даної системи представлена в таблиці 3.1.

Таблиця 3.1 – Вартість компонентів робототехнічної системи якості борошна на основі плати ESP32CAM

№	Найменування компонентів	Ціна, грн
1	2	3
1	плата з мікроконтролером ESP32CAM	180
2	камера OV2640	120
3	батарейний відсік із вимикачем 3xAA	25
4	лужні батареї Duracell AA (LR06) MN1500	189

Кінець таблиці 3.1

1	2	3
5	дисплей OLED S0.96" I2C 128x64 (синій)	112
6	програматор ESP32CAM-MB	130
7	пластик для корпусу	до 100
Всього: 856		

У таблиці 3.1 вказана ціна реалізації 856 грн. Якщо виключити програматор з переліку, ціна якого становить 130 грн, то ціна опуститься до 736 грн. З урахування різниці в ціні, вартість можливо заокруглити до 750 грн.

### 3.7 Напрямки вдосконалення

Робототехнічна система, призначена для оцінки якості борошна, може застосовуватися як у домашніх, так і в промислових умовах для завдань, пов'язаних з оцінкою якості.

Одним із потенційних удосконалень є модернізація камери для отримання зображень вищої якості, що, у свою чергу, дасть кращі результати у визначенні якості борошна.

Крім того, можна вдосконалити інтерфейс користувача та код, щоб полегшити взаємодію з користувачем. Іншим можливим напрямком розвитку є інтеграція алгоритмів машинного навчання для автоматизації та вдосконалення процесу аналізу, що зрештою призведе до більш точної та ефективної оцінки якості борошна.

### 3.8 Висновки

Після детального опису встановлення необхідних компонентів для Arduino IDE та Python було представлено програмну частину ESP32CAM. Робототехнічна

система визначення якості борошна на основі ESP32CAM зібрана з наступних компонентів: плата мікроконтролера ESP32CAM, камера OV2640, батарейний відсік з перемикачем 3×AA, лужні батарейки Duracell AA (LR06) MN1500, OLED дисплей S0.96" I2C 128×64 (синій), і пластик на корпус.

Процес складання почався з кріплення всіх компонентів до основної плати. Потім компоненти були з'єднані згідно з попередньо розробленою схемою. Програмування проводилося в середовищі Arduino IDE. Для програмування модуля ESP32CAM його підключили до порту USB за допомогою спеціального конвертера USB до UART. Цей конвертер дуже універсальний і широко використовується для програмування різних пристроїв.

Були розраховані витрати на створення системи визначення якості борошна. Система включає в себе модуль ESP32CAM, конвертер USB в UART, набір перемичок для Arduino і пластиковий корпус. Після розрахунку матеріальні витрати на систему склали біля 750 грн.

Визначено можливі шляхи вдосконалення системи. Одним із потенційних удосконалень є модернізація камери для підвищення якості зображення. Також є можливість покращити інтерфейс і код для більш зручної взаємодії з користувачем.

					КВРКІ 101052.21.01.02 ПЗ	Арк. 64
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВОК

У першому розділі було розглянуто основні поняття про робототехнічні системи, їх класифікації та основні компоненти. Також було проаналізовано системи які мали мету визначення якості такі, як AgBot II та проект з використанням ESP32 для контролю якості сталевих компонентів у виробничих конструкціях. В кінці розділу було визначено основні задачі для розробки робототехнічної системи визначення якості борошна

У другому розділі було розглянуто різні варіації компонентів для розробки системи. В результаті було обрано компоненти робототехнічної системи визначення якості борошна та їх інтеграція. Основою робототехнічної системи визначення якості борошна являється плата ESP32CAM з модулем камери високої роздільної здатності, вбудованим Wi-Fi і потужним мікропроцесором. Було розглянуті такі компоненти, як екран для відображення інформації, програматор для плати. В розділі також подано основні особливості використання Arduino IDE та Python.

Також, у розділі було представлено спрощений алгоритм роботи програмного забезпечення системи. Для досягнення роботи програмної частини було обрано Arduino IDE для програмування мікроконтролерів і Python як засіб для обробки зображень на комп'ютері та створення користувацького інтерфейсу.

Також проаналізовано, що використання зв'язку Wi-Fi забезпечить безперебійну передачу даних і віддалений доступ до системи, підвищуючи її загальну функціональність і зручність для користувача.

У третьому розділі була реалізована апаратна та програмна частина пристрою. Для системи було створено спрощену електричну принципову схему. Плата ESP32CAM слугує основою для виконання коду так визначення якості борошна. Для виведення інформації використовується OLED дисплей OV2640 та користувацький інтерфейс створений за допомогою Python.

					КВРКІ 101052.21.01.02 ПЗ	Арк. 65
Зм.	Арк.	№ докум.	Підпис	Дата		

Процес складання системи почався з кріплення всіх компонентів до основної плати. Далі компоненти були з'єднані згідно з попередньо розробленою схемою. Програмування плати ESP32CAM проводилося в середовищі Arduino IDE. Для програмування модуля ESP32CAM його підключили до порту USB за допомогою спеціального конвертера USB до UART.

Були розраховані витрати на створення системи визначення якості борошна. Система включає в себе модуль ESP32CAM, конвертер USB в UART, набір переминок для Arduino і пластиковий корпус. Після розрахунку матеріальні витрати на систему склали біля 750 грн.

Визначено можливі шляхи вдосконалення системи. Одним із потенційних удосконалень є модернізація камери для підвищення якості зображення. Також є можливість покращити інтерфейс і код для більш зручної взаємодії з користувачем.

Мета кваліфікаційної роботи забезпечення обробки зображення платою ESP32CAM для робототехнічної системи визначення якості борошна досягнена.

					КВРКІ 101052.21.01.02 ПЗ	Арк. 66
Зм.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. International federation of robotics (IFR). URL: <https://ifr.org/> (дата звернення: 16.02.2024).
2. The robotics revolution: the next great leap in manufacturing. URL: <https://www.bcg.com/publications/2015/lean-manufacturing-innovation-robotics-revolution-next-great-leap-manufacturing>. (дата звернення: 16.02.2024).
3. Lund S., Chui M., Bughin J., Manyik J. Jobs lost, jobs gained: workforce transitions in a time of automation. URL: <https://www.mckinsey.com/~media/mckinsey/industries/public%20and%20social%20sector/our%20insights/what%20the%20future%20of%20work%20will%20mean%20for%20jobs%20skills%20and%20wages/mgi-jobs-lost-jobs-gained-executive-summary-december-6-2017.pdf>. (дата звернення: 17.02.2024).
4. Adaptive swarm robotics could revolutionize smart agriculture. URL: <https://www.nifa.usda.gov/about-nifa/impacts/adaptive-swarm-robotics-could-revolutionize-smart-agriculture>. (дата звернення: 17.02.2024).
5. Mesbahi M. The Future of Agriculture: adopting more advanced robots for next-gen farming. URL: <https://www.wevolver.com/article/the-future-of-agriculture-adopting-more-advanced-robots-for-next-gen-farming>. (дата звернення: 18.02.2024).
6. Balaska V, Adamidou Z, Vryzas Z, Gasteratos A. Sustainable crop protection via robotics and artificial intelligence solutions. *Machines*. 2023. Vol. 11 No. 8. P. 3-8. DOI: <https://doi.org/10.3390/machines11080774>
7. Hailin W., Liu G., Wu J., Zhang Z., Yang Y. Design of intelligent inspection robot system for electrical equipment based on visual recognition. *2021 IEEE 4th International Conference on Automation, Electronics and Electrical*

					КВРКІ 101052.21.01.02 ПЗ	Арк. 67
Зм.	Арк.	№ докум.	Підпис	Дата		

- Engineering (AUTEEE)*. Shenyang, China, November 19-21. 2021. Shenyang, 2021. P. 409-413. DOI: 10.1109/AUTEEE52864.2021.9668826.
8. Povlsen K. New generation of robots can help small manufacturers. URL: <https://hbr.org/2023/11/a-new-generation-of-robots-can-help-small-manufacturers>. (дата звернення: 20.02.2024).
  9. Wang W., Siau K. Artificial intelligence, machine learning, automation, robotics, future of work and future of humanity: a review and research agenda. *Journal of Database Management*. 2019. Vol. 30, No 1. P. 61-79. DOI: <https://doi.org/10.4018/JDM.2019010104> (дата звернення: 20.02.2024).
  10. Gasparetto A., Scaleria L. A brief history of industrial robotics in the 20th century. *Advances in Historical Studies*. 2019. No. 8. P. 24-35. DOI: <https://doi.org/10.4236/ahs.2019.81002>. (дата звернення: 20.02.2024).
  11. The future. URL: <https://www.britannica.com/technology/robot-technology/The-future>. (дата звернення: 21.02.2024).
  12. Vinoth Shivan. Whats happening in robotics? Five trends to watch. URL: <https://www.linkedin.com/pulse/whats-happening-robotics-five-trends-watch-vinoth-k>. (дата звернення: 23.02.2024).
  13. The history of automated guided vehicles. URL: [https://www.solving.com/agv-dictionary/the-history-of-automated-guided-vehicles/#:~:text=The%20history%20of%20Automated%20Guided%20Vehicles%20\(AGVs\)%20traces%20back%20to,Barrett%20Electronics%20of%20Northbrook%2C%20Illinois](https://www.solving.com/agv-dictionary/the-history-of-automated-guided-vehicles/#:~:text=The%20history%20of%20Automated%20Guided%20Vehicles%20(AGVs)%20traces%20back%20to,Barrett%20Electronics%20of%20Northbrook%2C%20Illinois). (дата звернення: 24.02.2024).
  14. Tiny, shape-shifting robot can squish itself into tight spaces. URL: <https://www.sciencedaily.com/releases/2023/08/230830131742.htm>. (дата звернення: 25.02.2024).
  15. Cann O. Machines will do more tasks than humans by 2025 but robot revolution will still create 58 million net new jobs in next five years. *The World Economic Forum's 12th Annual Meeting of the New Champions*: materials of international economic conference, Tianjin, 18-20 September. 2018. Tianjin, 2018. URL:

					КВРКІ 101052.21.01.02 ПЗ	Арк. 68
Зм.	Арк.	№ докум.	Підпис	Дата		

<https://www.weforum.org/press/2018/09/machines-will-do-more-tasks-than-humans-by-2025-but-robot-revolution-will-still-create-58-million-net-new-jobs-in-next-five-years/>.

16. How are cobots making a difference in your industry? URL: <https://www.universal-robots.com/industries/>. (дата звернення: 25.02.2024).
17. Powell A. What to know about robotic surgery. URL: <https://www.webmd.com/a-to-z-guides/robotic-surgery-what-know>. (дата звернення: 25.02.2024).
18. Medical robots making a difference in healthcare. URL: <https://online-engineering.case.edu/blog/medical-robots-making-a-difference>. (дата звернення: 26.02.2024).
19. Logistics automation market size, share & trends analysis report by component (hardware, software, services), by function, by logistics type, by organization size, by software application, by vertical, by region, and segment forecasts, 2023 – 2030. URL: <https://www.grandviewresearch.com/industry-analysis/logistics-automation-market-report>. (дата звернення: 27.02.2024).
20. Dan Huang, Qiurong Chen, Jiahui Huang, Shaojun Kong, Zhiyong Li. Customer-robot interactions: understanding customer experience with service robots. *International Journal of Hospitality Management*. 2021. Vol. 99. DOI: <https://doi.org/10.1016/j.ijhm.2021.103078>. (дата звернення: 26.02.2024).
21. Botta A., Cavallone P., Baglieri L., Colucci G., Tagliavini L., Quaglia G. A review of robots, perception, and tasks in precision agriculture. *Applied Mechanics*. 2022. Vol. 3, No. 3. P. 830-854. DOI: <https://doi.org/10.3390/applmech3030049>. (дата звернення: 27.02.2024).
22. Patel A., Karlsson S., Lindqvist B., Haluska J., Kanellakis C., Agha-mohammadi A., Nikolakopoulos G. Towards field deployment of MAVs in adaptive exploration of GPS-denied subterranean environments. *Robotics and Autonomous Systems*. 2023. Vol. 176. DOI: <https://doi.org/10.1016/j.robot.2024.104663> (дата звернення: 01.03.2024).

					КВРКІ 101052.21.01.02 ПЗ	Арк. 69
Зм.	Арк.	№ докум.	Підпис	Дата		

23. Robots in manufacturing. URL: <https://www.britannica.com/technology/automation/Robots-in-manufacturing>. (дата звернення: 03.02.2024).
24. Johnson B. How baxter the robot works. URL: <https://science.howstuffworks.com/baxter-robot.htm>. (дата звернення: 04.03.2024).
25. David M. Rosen, Kevin J. Doherty, Antonio Terán Espinoza, John J. Leonard. Advances in inference and representation for simultaneous localization and mappin. *Annual Review of Control, Robotics, and Autonomous Systems*. 2021. Vol 4. P. 215-242. DOI: <https://doi.org/10.1146/annurev-control-072720-082553> (дата звернення: 04.03.2024).
26. Giuseppe Carbone, Med Amine Laribi. Robot design. Switzerland: Springer Cham, 2023. 269 p. URL: <https://link.springer.com/book/10.1007/978-3-031-11128-0>.
27. Fengbei Liu, Yu Tian, Yuanhong Chen, Yuyuan Liu, Vasileios Belagiannis, Gustavo Carneiro. ACPL: anti-curriculum pseudo-labelling for semi-supervised medical image classification. URL: [https://openaccess.thecvf.com/content/CVPR2022/papers/Liu\\_ACPL\\_Anti-Curriculum\\_Pseudo-Labeling\\_for\\_Semi-Supervised\\_Medical\\_Image\\_Classification\\_CVPR\\_2022\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2022/papers/Liu_ACPL_Anti-Curriculum_Pseudo-Labeling_for_Semi-Supervised_Medical_Image_Classification_CVPR_2022_paper.pdf). (дата звернення: 05.03.2024).
28. Zhang Z, Peng G, Wang W, Chen Y, Jia Y, Liu S. Prediction-based human-robot collaboration in assembly tasks using a learning from demonstration model. *Sensors*. 2022. Vol. 22, No. 11. DOI: <https://doi.org/10.3390/s22114279>. (дата звернення: 06.03.2024).
29. What is visual inspection. URL: <https://www.mathworks.com/discovery/visual-inspection.html>. (дата звернення: 10.03.2024).
30. Data transmission: what is it? Everything you need to know. URL: <https://www.cdnetworks.com/enterprise-applications-blog/everything-you-need-to-know-about-data-transmission/>. (дата звернення: 10.03.2024).

					КВРКІ 101052.21.01.02 ПЗ	Арк. 70
Зм.	Арк.	№ докум.	Підпис	Дата		

31. Microcontrollers (MCUs). URL: <https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors>. (дата звернення: 10.03.2024)
32. Overview of IBM® watson IoT platform analytics. URL: [https://www.ibm.com/docs/en/mapms/1\\_cloud?topic=analyzing-data](https://www.ibm.com/docs/en/mapms/1_cloud?topic=analyzing-data). (дата звернення: 11.03.2024).
33. Karen F. The basic structure and functionality of robots. URL: <https://essay.biz/blog/topics/the-basic-structure-and-functionality-of-robots/>. (дата звернення: 12.03.2024).
34. Types of robots: how robotics technologies are shaping today's world. URL: <https://www.intel.com/content/www/us/en/robotics/types-and-applications.html>. (дата звернення: 15.03.2024).
35. AgBot II robotic site-specific crop and weed management tool. URL: <https://research.qut.edu.au/qcr/Projects/agbot-ii-robotic-site-specific-crop-and-weed-management-tool/>. (дата звернення: 18.003.2024).
36. Ball D. Robotics for zero-tillage agriculture (AgBot). URL: <https://www.davidmichaelball.com/portfolio-items/robotics-for-zero-tillage-agriculture-agbot/>. (дата звернення: 18.03.2024).
37. ESP32CAM object detection & identification with OpenCV. URL: <https://how2electronics.com/esp32-cam-based-object-detection-identification-with-opencv/>. (дата звернення: 18.03.2024).
38. ESP32CAM object detection with edge impulse. URL: <https://dronebotworkshop.com/esp32-object-detect/> (дата звернення: 18.03.2024).
39. MicroPython\_ESP32\_psRAM\_LoBo. URL: [https://github.com/loboris/MicroPython\\_ESP32\\_psRAM\\_LoBo](https://github.com/loboris/MicroPython_ESP32_psRAM_LoBo) (дата звернення: 20.03.2024).
40. Camera Web Server. URL: [https://docs.sunfounder.com/projects/esp32-starter-kit/en/latest/arduino/iot\\_projects/ar\\_iot\\_camera\\_web.html](https://docs.sunfounder.com/projects/esp32-starter-kit/en/latest/arduino/iot_projects/ar_iot_camera_web.html). (дата звернення: 20.03.2024).

- 41.Засорнов О.С., Засорнова І.О. Програмування мікроконтролерних та робототехнічних систем. Хмельницький: ХНУ, 2023. 280 с.
- 42.Гришук Ю. С. Мікроконтролери: архітектура, програмування та застосування в електромеханіці : навч. посіб. Харків: НТУ ХПІ, 2019. 384 с.
- 43.Круліковський Б. Б., Николайчук Я.М., Шатний С. В. Мікропроцесорні системи. Практикум : навч. посіб. Рівне: НУВГП, 2016. 191 с.
- 44.Зубчук В.І. Делавар-Касмаї. М. Цифрова схемотехніка: навчальний посібник для самостійної роботи студентів. Київ: КПІ ім. Ігоря Сікорського, 2021. 258 с.
- 45.Робототехнічні системи: проектування і моделювання – Київ: КПІ ім. Ігоря Сікорського, 2021. – 112 с.
- 46.Getting Started with the ESP32 Development Board. URL: <https://randomnerdtutorials.com/getting-started-with-esp32/>. (дата звернення:: 12.04.2024)
- 47.Cameron N. ESP32 Microcontroller. Berkeley: Apress, 2022. 54 p.
- 48.Python. URL: <https://www.python.org/downloads/>. (дата звернення:: 12.04.2024).
- 49.Margolis M., Jepson B., Weldin N. Arduino cookbook: recipes to begin, expand, and enhance your projects. California: O'Reilly Media, 2020. 795 p.
- 50.Blum J. Exploring Arduino: Tools and techniques for engineering wizardry. New Jersey: Wiley, 2019. 512 p.
- 51.Banzi M., Shilon M. Make: getting started with arduino. Sebastopol: Maker Media, 2018. 245 p.
- 52.Boxall J. Arduino workshop, 2nd edition: a hands-on introduction with 65 projects. San Francisco: No Starch Press, 2021. 432 p.
- 53.Spahic B. Arduino without prior knowledge: create your own first project within 7 days. Chicago: Independently published, 2020. 111 p.
- 54.Matthes E. Python crash course, 3rd edition: a hands-on, project-based introduction to programming. San Francisco: No Starch Press, 2023. 552 p.

					КВРКІ 101052.21.01.02 ПЗ	Арк. 72
Зм.	Арк.	№ докум.	Підпис	Дата		

55. Jayne B. Python programming language. Boca raton: QuickStudy Reference Guides, 2019. 6 p.
56. Sweigart A. Automate the boring stuff with python, 2nd edition: practical programming for total beginners. San Francisco: No Starch Press, 2019. 592 p.
57. Python programming for beginners: the complete python coding crash course - boost your growth with an innovative ultra-fast learning framework and exclusive hands-on interactive exercises & projects. Chicago: Independently published, 2024. 160 p.
58. Raschka S., Liu Y., Mirjalili V. Machine learning with pytorch and scikit-learn: develop machine learning and deep learning models with pytho. Birmingham: Packt Publishing, 2022. 774 p.
59. Chollet F. Deep learning with python, second edition 2nd edition. New York: Manning, 2021. 504 c.
60. Bruce P., Bruce A., Gedeck P. Practical statistics for data scientists: 50+ essential concepts using r and python. California: O'Reilly Media, 2020. 360 p.

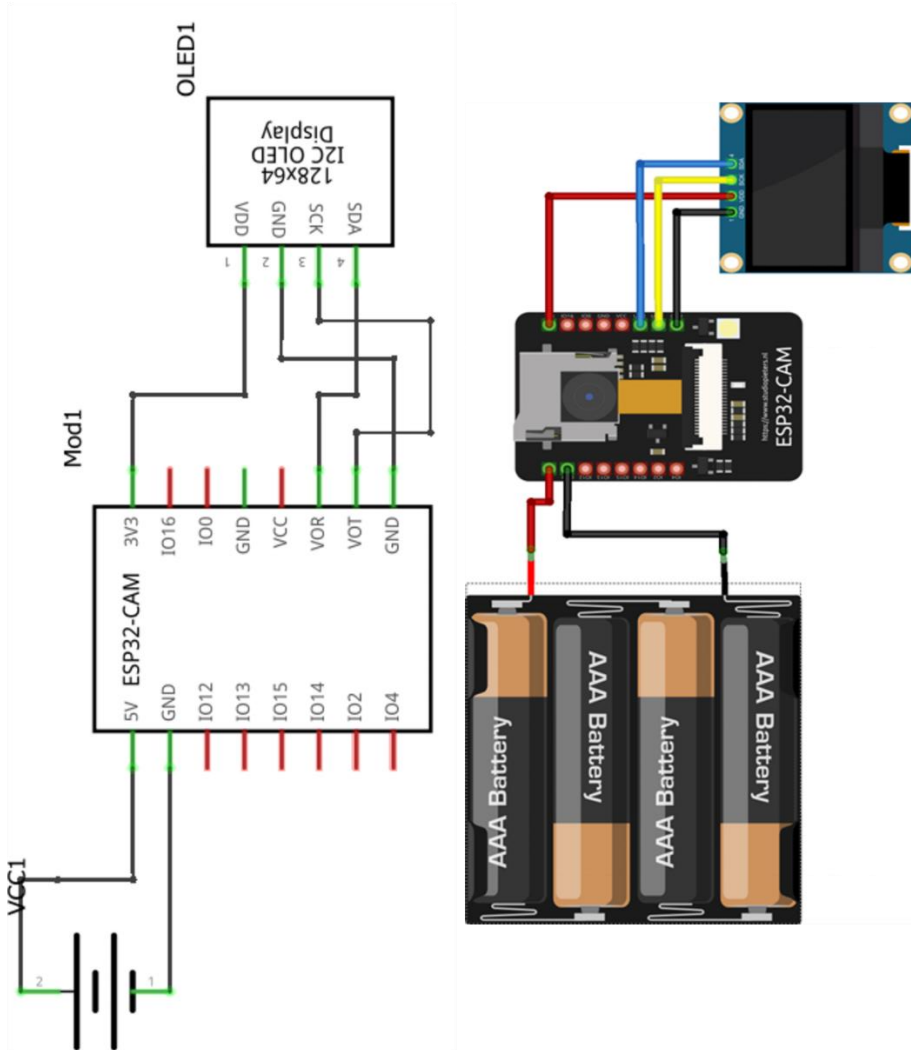
					КВРКІ 101052.21.01.02 ПЗ	Арк. 73
Зм.	Арк.	№ докум.	Підпис	Дата		

# Додаток А

(обов`язковий)

Копія креслення «Спрощена електрично принципова схема»

Спрощена електрично принципова схема



КвРКІ.101052.21.01.02 Е2

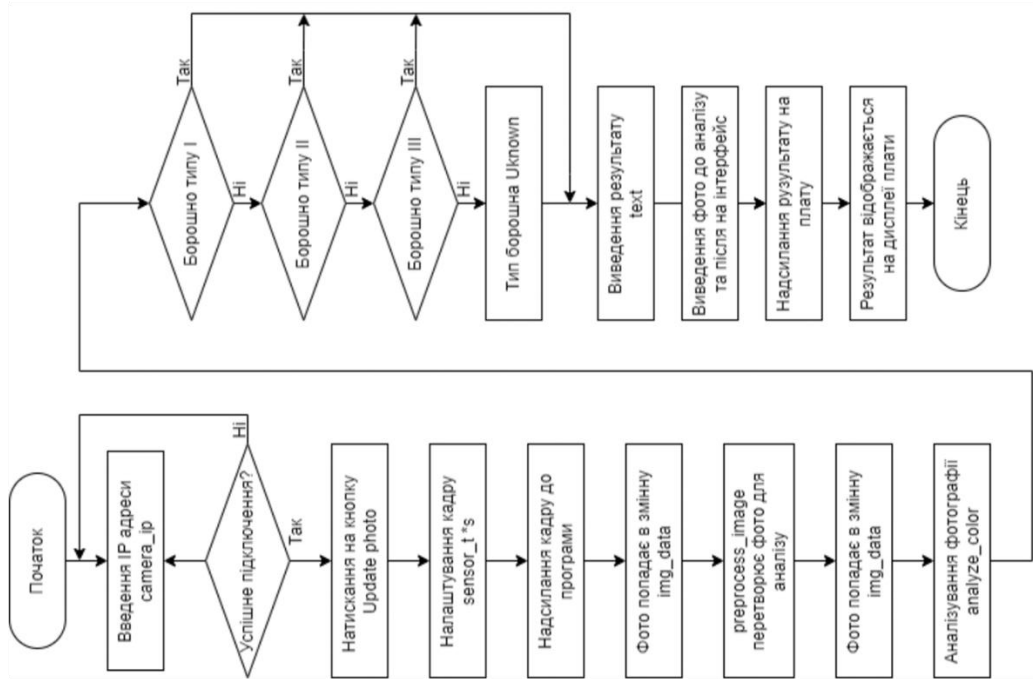
КвРКІ.101052.21.01.02 Е2		Роботодавця система визначення	
Зад. Акт.	№ докум.	Підпис	Дата
Код	Код	Код	Код
Підпис	Заступця ІО	Спрощена електрично принципова	
Н. КОІТР	Н. КОІТР	схема	
Т. КОІТР	Заступця ІО	ХНУ КІ2С-21-1	
З.ІТВ	Розробник ІО		

# Додаток Б

(обов'язковий)

Копія креслення «Алгоритм роботи системи»

Алгоритм роботи системи



КВРКІ.101052.21.01.02.ES

КВРКІ.101052.21.01.02.ES		Літера	Маса	Масштаб
Зм.Док.	№ докум.	Підпис	Дата	
Розроб.	Рухом. А.О.			
Перевір.	Васильова І.О.			
Ін.Контр.				
Т.Сонтр.	Васильова І.О.			
Затв.	Васильова І.О.			
Роботоздатна система виконання завдань борозна з використанням ESP32CAM Алгоритм роботи системи				
ХНУ КПС-21-1				
Архив 2 Архив 3				



## Додаток Г

Код програми на Python

```
import cv2
import numpy as np
import socket
from PIL import Image, ImageTk
import tkinter as tk
from threading import Thread
from PIL.Image import Resampling # Import Resampling from PIL.Image
# Replace with the IP address of your ESP32-CAM board
camera_ip = "192.168.0.101"
def preprocess_image(img):
    # Resize image
    img = cv2.resize(img, (300, 300))

    # Convert to HSV color space
    hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

    return hsv

def analyze_color(hsv):
    # Define color thresholds for white flour (you may need to adjust these based on
your flour samples)
    lower_white = np.array([0, 0, 200])
    upper_white = np.array([180, 30, 255])

    # Create a mask for white color (flour)
    mask = cv2.inRange(hsv, lower_white, upper_white)
```

```

# Calculate the average HSV values within the masked region
avg_hsv = cv2.mean(hsv, mask=mask)

# Unpack the average HSV values (ignore the fourth element, which is the total
number of non-zero pixels in the mask)
avg_h, avg_s, avg_v, _ = avg_hsv

# Determine the flour type based on the average V (value/brightness) component
flour_type = ""
if avg_v >= 245:
    flour_type = 'Type 1 - Very White'
elif avg_v >= 230:
    flour_type = 'Type 2 - White'
elif avg_v >= 215:
    flour_type = 'Type 3 - Off-White'
else:
    flour_type = 'Unknown'

print(f'Average V (Brightness): {avg_v:.2f}')
print(f'Flour Type: {flour_type}')

return flour_type

def update_image():
    global original_img, analyzed_img

# Get the IP address from the text box
camera_ip = ip_entry.get()

```

```

# Create a TCP client and connect to the ESP32-CAM
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect((camera_ip, 80))

# Receive the raw JPEG image data
img_data = b"
while True:
    data = client_socket.recv(1024)
    if not data:
        break
    img_data += data

# Close the TCP client
client_socket.close()

# Decode the image data
img_np = np.frombuffer(img_data, dtype=np.uint8)
img = cv2.imdecode(img_np, cv2.IMREAD_COLOR)

# Preprocess the image
analyzed_img = preprocess_image(img)

# Analyze color
flour_type = analyze_color(analyzed_img)

# Convert the images to PIL format for displaying in tkinter
original_img = Image.fromarray(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
analyzed_img = Image.fromarray(cv2.cvtColor(analyzed_img,
cv2.COLOR_BGR2RGB))

```

```
# Display the result in the result_label
result_label.config(text=f"Flour Type: {flour_type}")

def main():
    global original_img, analyzed_img, root, original_label, analyzed_label,
ip_entry, result_label

    # Create the main window
    root = tk.Tk()
    root.title("Flour Quality Analysis")
    root.geometry("800x600")

    # Create frames for layout
    top_frame = tk.Frame(root)
    top_frame.pack(side=tk.TOP, fill=tk.X, padx=10, pady=10)

    button_frame = tk.Frame(top_frame)
    button_frame.pack(side=tk.LEFT, padx=(0, 10))

    ip_frame = tk.Frame(top_frame)
    ip_frame.pack(side=tk.RIGHT, padx=(10, 0))

    image_frame = tk.Frame(root)
    image_frame.pack(side=tk.BOTTOM, fill=tk.BOTH, expand=True, padx=10,
pady=10)

    # Create the update button
```

```

        update_button = tk.Button(button_frame, text="Update Photo",
command=update_image)
        update_button.pack(side=tk.TOP)

# Create the IP address entry text box
ip_label = tk.Label(ip_frame, text="IP Address:")
ip_label.pack(side=tk.LEFT)
ip_entry = tk.Entry(ip_frame, width=20)
ip_entry.pack(side=tk.RIGHT)
ip_entry.insert(0, camera_ip)

# Create the result label
        result_label = tk.Label(button_frame, text="", wraplength=200,
justify="center")
        result_label.pack(side=tk.BOTTOM, fill=tk.X, padx=10, pady=(5, 0))

# Create image labels
original_label = tk.Label(image_frame)
original_label.pack(side=tk.TOP, fill=tk.X)
analyzed_label = tk.Label(image_frame)
analyzed_label.pack(side=tk.BOTTOM, fill=tk.X)

# Start the tkinter mainloop
update_image() # Initial image update

# Function to update the displayed images
def update_displayed_images():
    global original_label, analyzed_label

    if original_img is not None and analyzed_img is not None:

```

```
# Resize the images to fit the window using Resampling.LANCZOS
(provides antialiasing effect)
        original_img_resized = original_img.resize((400, 300),
Resampling.LANCZOS)
        analyzed_img_resized = analyzed_img.resize((400, 300),
Resampling.LANCZOS)

# Convert the images to PhotoImage format for displaying in tkinter
original_photo = ImageTk.PhotoImage(original_img_resized)
analyzed_photo = ImageTk.PhotoImage(analyzed_img_resized)

# Update the image labels with the new images
original_label.config(image=original_photo)
original_label.image = original_photo
analyzed_label.config(image=analyzed_photo)
analyzed_label.image = analyzed_photo

# Call this function again after 100 milliseconds
root.after(100, update_displayed_images)

# Start updating the displayed images
update_displayed_images()

root.mainloop()

if __name__ == "__main__":
    main()
```

## Додаток Д

Код програми для ESP32CAM

```
#include "esp_camera.h"
#include <WiFi.h>
#include <Adafruit_SSD1306.h>

// Replace with your WiFi credentials
const char* ssid = "TP-LINK_92E4";
const char* password = "agutsal@74";

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

// Camera model (Adjust according to your ESP32-CAM module)
#define CAMERA_MODEL_AI_THINKER

#if defined(CAMERA_MODEL_AI_THINKER)
#define PWDN_GPIO_NUM 32
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM 0
#define SIOD_GPIO_NUM 26
#define SIOC_GPIO_NUM 27
#define Y9_GPIO_NUM 35
#define Y8_GPIO_NUM 34
#define Y7_GPIO_NUM 39
#define Y6_GPIO_NUM 36
#define Y5_GPIO_NUM 21
```

```
#define Y4_GPIO_NUM    19
#define Y3_GPIO_NUM    18
#define Y2_GPIO_NUM    5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM  23
#define PCLK_GPIO_NUM  22

#endif

// Frame settings
#define FRAME_SIZE FRAMESIZE_QVGA

// WiFi server settings
WiFiServer controlServer(81); // Use a different port for control, e.g., 81
WiFiServer server(80);

void handleControlClient() {
    // Wait for a control client to connect
    WiFiClient controlClient = controlServer.available();
    if (!controlClient) {
        return;
    }

    // Read the command
    String command = controlClient.readStringUntil('\n');
    command.trim();

    // Display the flour type on the OLED display
    display.clearDisplay();
```

```
display.setCursor(0, 0);
display.println(command);
display.display();

// Close the control client
controlClient.stop();
}

void setup() {
  Serial.begin(115200);
  delay(1000);

  // Connect to WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");
  Serial.print(WiFi.localIP());

  // Start the server
  server.begin();

  // Initialize the camera
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
```

```
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;
```

```
if (psramFound()) {
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_QVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
}
```

```
esp_err_t err = esp_camera_init(&config);
```

```
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
}

sensor_t *s = esp_camera_sensor_get();
s->set_framesize(s, FRAME_SIZE);

if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for (;;) // Don't proceed, loop forever
    }
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(WHITE);
display.display();

// Start the control server
controlServer.begin();
}

void loop() {
    // Wait for a client to connect
    WiFiClient client = server.available();
    if (!client) {
        return;
    }

    // Capture and send an image to the client
```

```
camera_fb_t *fb = NULL;
fb = esp_camera_fb_get();
if (!fb) {
    Serial.println("Camera capture failed");
    return;
}

client.write((uint8_t *)fb->buf, fb->len);
esp_camera_fb_return(fb);
handleControlClient();
}
```

Ім'я користувача:  
Кафедра КІ

ID перевірки:  
1016293123

Дата перевірки:  
29.05.2024 07:15:29 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
29.05.2024 07:24:26 EEST

ID користувача:  
100005591

Назва документа: Гуцалюк\_Робототехнічна система визначення якості борошна з використанням ESP32CAM

Кількість сторінок: 78 Кількість слів: 12456 Кількість символів: 96040 Розмір файлу: 2.51 MB ID файлу: 1016087369

## 3.43% Схожість

Найбільша схожість: 0.87% з Інтернет-джерелом (<https://randomnerdtutorials.com/esp32-cam-video-streaming-web-ser..>)

2.96% Джерела з Інтернету 69 ..... Сторінка 80

1.6% Джерела з Бібліотеки 89 ..... Сторінка 81

## 0.02% Цитат

Цитати 3 ..... Сторінка 82

Посилання 1 ..... Сторінка 82

## 0% Вилучень

Немає вилучених джерел

## Модифікації

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

Замінені символи 24

## Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 0.0%

Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 12%

ID: 127626 Назва: БКР РОБОТОТЕХНІЧНА СИСТЕМА ВИЗНАЧЕННЯ ЯКОСТІ БОРОШНА З ВИКОРИСТАННЯМ ESP32СAM Додано в БД: 2024-05-29 Автор: А.О. Гушак Керівник: І.О. Засорнова Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	82133	662	817 (1%)	13 (2%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Завідувачу кафедри КІС  
д-р.техн.наук, проф. Говорущенко Т. О.

Гуцалюка Артема Олексійовича

ПІБ здобувача вищої освіти

ФІТ, 3 курсу, групи КІ2с-21-1

### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, 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 плагіат оповіщений(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.

3 червня 2024 року



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Гуцалюк Артем Олексійович

Тема: Робототехнічна система визначення якості борошна з використанням ESP32CAM

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень   3   Кількість сторінок записки   66  

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є забезпечення обробки зображення платою ESP32CAM для робототехнічної системи визначення якості борошна.
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: В першому розділі кваліфікаційної роботи проведено дослідження предметної області (проаналізовано теорію робототехнічних систем та наявного програмно-апаратного забезпечення) та виконано постановку задачі дослідження. В другому розділі кваліфікаційної роботи проведено проектування робототехнічної системи визначення якості борошна з використанням ESP32CAM: виконано аналіз різних компонентів системи; на основі аналізу здійснено вибір компонентів; створено алгоритм роботи робототехнічної системи. В третьому розділі кваліфікаційної роботи виконано програмно-апаратну реалізацію системи визначення якості борошна з використанням ESP32CAM, а саме: реалізовано спрощену електрично-принципову схему; здійсненна підготовка бібліотек для Arduino IDE; створено програмне забезпечення для плати ESP32CAM за допомогою Arduino IDE; здійснена підготовка бібліотек для Python; створено програмне забезпечення для аналізу якості борошна на основі Python; визначено шляхи вдосконалення системи.
4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи:

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: *Відмінно*

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

*Мартинюк Валерій Володимирович*  
*зав. каф. АІТІ та Р*

“*28*” *05* 2024 р.

 (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: Робототехнічна система визначення якості борошна з використанням ESP32CAM

Автор:           Гуцалюк Артем Олексійович          

Спеціальність:           123– Комп'ютерна інженерія          

Освітня програма:           освітньо-професійна          

Науковий керівник:           Ірина Олександрівна Засорнова, 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 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:

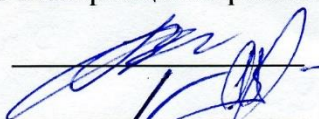


- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
- 3) в якості запозичень в окремих місцях системою зафіксовано код визначення контактів для камери, які є стандартним кодом для подібних задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 3.43% і адресується до 89 першоджерел з бібліотеки та 69 першоджерел з інтернету, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС

І. О. Засорнова

С.М. Лисенко

Т. О. Говорущенко