

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

Пояснювальна записка

до бакалаврської дипломної роботи

на тему: «Мікропроцесорна система безконтактного
термометра»

Виконав: студент 2 курсу, групи 1КІ-20мс
напряму підготовки (спеціальності)
123 – «Комп'ютерна інженерія»

Кізіма А.С.

Керівник к.т.н., доц. Кадук О.В.

Рецензент к.т.н., доц. Карпінєць В.В.

м. Вінниця – 2022 року

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

Освітньо-кваліфікаційний рівень — бакалавр

Спеціальність — 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри

обчислювальної техніки

проф., д.т.н. О.Д.Азаров

« » 2022 р.

З А В Д А Н Н Я
НА БАКАЛАВСЬКУ ДИПЛОМНУ РОБОТУ


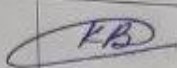
Кізімі Андрію Сергійовичу

1 Тема бакалаврської роботи «Мікропроцесорна система безконтактного термометра», керівник роботи Кадук Олександр Володимирович, к.т.н., доцент, затверджені наказом вищого навчального закладу від 24.03.22 року №66

2 Строк подання студентом роботи: 13.06.2022р.

3 Вихідні дані до роботи: Технічні параметри систем безконтактного вимірювання температури, технічний опис мікроконтролера ESP32, мікропроцесорних платформи TTGO T4, технічна документація на компілятори.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Вступ, огляд і аналіз безконтактних термометрів, вибір мікропроцесорної платформи, проектування мікропроцесорної системи безконтактного термометра, висновки, перелік джерел посилання

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Кадук О.В., к.т.н., доцент кафедри ОТ		

5 Консультанти розділів роботи

6. Дата видачі завдання 10.02.2022р

7 Календарний план наведений в таблиці 1

Таблиця 1 — Календарний план

з/п	Назва етапів виконання бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Постановка задач роботи	14.02.22	
2	Огляд інформаційних джерел	15.02-28.02.22	
3	Огляд і аналіз принципу роботи безконтактних термометрів	01.03-14.03.22	
4	Вибір мікропроцесорної платформи	15.03-28.03.22	
5	Проектування мікропроцесорної системи безконтактного термометра	29.03-11.04.22	
6	Оформлення пояснювальної записки та ілюстративного матеріалу	12.04-25.04.22	
7	Аналіз виконання роботи. Висновки. Додатки	26.04-09.05.22	
8	Перевірка якості виконання бакалаврської роботи та усунення недоліків	17.05.22	

Студент

(підпис)



А.С. Кізіма

(прізвище та ініціали)

Керівник роботи

(підпис)



О.В. Кадук

(прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська дипломна робота відноситься до галузі мікропроцесорних систем безконтактного визначення температури поверхні об'єктів, що знаходяться у полі зору пристрою.

ANNOTATION

The bachelor's degree work is brought to the gallery of microprocessor systems for non-contact monitoring of the temperature of the surface of objects, which are located at the edge of the building.

ЗМІСТ

ВСТУП.....	7
1 ОГЛЯД І АНАЛІЗ ПРИНЦИПУ РОБОТИ БЕЗКОНТАКТНИХ ТЕРМОМЕТРІВ	9
1.1 Призначення та область застосування.....	9
1.2 Аналіз існуючих аналогів.....	12
1.2.1 Датчики температури.....	12
1.2.2 Тепловізори.....	16
1.3 Огляд відлагоджувальних плат для мікроконтролерів.....	19
1.4 Види індикаторів для відображення результатів вимірювань.....	21
1.5 Пам'ять для зберігання вимірювань температури.....	22
2 ВИБІР МІКРОПРОЦЕСОРНОЇ ПЛАТФОРМИ.....	24
2.1 Вибір мікроконтролера	24
2.2 Архітектура платформи TTGO T4 V1.3.....	30
2.3 Програмування.....	33
3 ПРОЕКТУВАННЯ МІКРОПРОЦЕСОРНОЇ СИСТЕМИ БЕЗКОНТАКТНОГО ТЕРМОМЕТРА.....	36
3.1 Структурна схема.....	36
3.2 Принципова схема.....	37
3.3 Програмування.....	39
3.4 Алгоритм роботи.....	46
ВИСНОВКИ.....	50
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	51
ДОДАТОК А Технічне завдання.....	53

					08-23.БДР.034.00.000 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата	Мікропроцесорна система безконтактного термометра Пояснювальна записка			Літ.	Арк.	Аркушів	
Розроб.	Кізіма А.С									6	60
Перевір.	Кадук О.В.										
Реценз.	Карпінєць В.В.										
Н. Контр.	Швець С. І.										
Затверд.	Азаров О.Д.				ВНТУ, гр. ІКІ – 20мс						

ДОДАТОК Б Структурна схема системи.....	56
ДОДАТОК В Схема електрична принципова системи.....	57
ДОДАТОК Г Функціональна схема.....	58
ДОДАТОК Д Код потоку сенсора.....	59
ДОДАТОК Е Код опрацювання потоку сенсора.....	62
ДОДАТОК Є Габаритні розміри та розміщення елементів платформи TTGO T4 V1.3.....	65
ДОДАТОК Ж Протокол перевірки на плагіат	66

ВСТУП

Темою випускної кваліфікаційної роботи є розробка портативного програмно апаратного комплексу T-visor для безконтактного визначення температури поверхні об'єкта.

Актуальність: у багатьох сферах діяльності може виникнути проблема виявлення та вимірювання температури поверхні об'єкта. Застосування тепловізорів обумовлено необхідністю пошуку гарячих (іноді - холодних) місць на температурному полі, наявність яких говорить про порушення нормального режиму експлуатації об'єкта або обладнання, небезпечні дефекти, втрати енергії тощо. Тепловізор не тільки дозволяє локалізувати ці гарячі точки, але й виміряти їх температуру. Унікальність даного приладу полягає в його незамінності у багатьох сферах життєдіяльності людини, від медицини до будівництва.

Тепловізор — це оптико-електронна система, яка призначається для отримання видимого зображення об'єктів, що випромінюють невидиме інфрачервоне випромінювання. Пристрій перетворює інфрачервоне випромінювання на електричні сигнали, які після обробки перетворюються на видиме зображення, яке відображається на екрані пристрою у вигляді кольорового поля, де кожній температурі відповідає свій колір.

Всі навколишні об'єкти випускають власне теплове випромінювання, чим тепліше об'єкт, тим потужніше випромінювання, що випускається від нього. Тепловізор отримуючи дані на свій сенсор перераховує потужність випромінювання в температуру, для цього використовуються калібрувальна крива датчика та фундаментальні залежності тепловізора. Ситуація складна тим, що це об'єкти в інфрачервоній області відбивають інфрачервоне випромінювання навколишніх об'єктів. Зрештою сенсор отримує з кожної точки визначення температури та її власне, і відбите стороннє випромінювання, що у своє чергу носить роль перешкоди для виміру. Щоб врахувати наявність відбитого випромінювання, тепловізору необхідно знати його потужність і правильним чином проводити калібрування сенсора, з урахуванням цих даних.

Мета дослідження: теоретичне та практичне дослідження застосування портативного мікропроцесорного програмно-апаратного комплексу для безконтактного визначення температури поверхні об'єктів.

Об'єктом дослідження є процеси які відбуваються в портативних програмно-апаратних комплексах безконтактного визначення температури поверхні об'єктів.

Предметом дослідження є пристрій, методи та засоби проектування системи, для визначення безконтактним способом температури об'єктів, що знаходяться в радіусі поля зору його сенсора.

Задачі для досягнення поставленої мети є:

1. Проаналізувати архітектуру та пристрій плати TTGO.
2. Проаналізувати особливості мікроконтролера ESP32.
3. Проаналізувати специфіку інфрачервоних датчиків температури.
4. Спроекувати систему.
5. Розробити алгоритм роботи пристрою.
6. Написати програмний код для відображення зображення на екрані.
7. Провести дослідження спроектованого пристрою.

Практичне значення одержаних результатів полягає у тому, що спроектовано мікропроцесорну систему безконтактного термометра з використанням сучасних засобів та методів побудови аналогічних систем.

Апробація. Участь у LI науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії 2022 з доповіддю «Мікропроцесорний безконтактний термометр». Тези опубліковано 30.05.22.

1 ОГЛЯД І АНАЛІЗ ПРИНЦИПУ РОБОТИ БЕЗКОНТАКТНИХ ТЕРМОМЕТРІВ

1.1 Призначення та область застосування

Вибір методу вимірювання температури залежить від діапазону вимірюваних температур, необхідної точності, ступеня автоматизації процесу, його швидкодії і допустимої величини вхідного теплового опору вимірювального пристрою, тобто його вхідної теплоємності.

В діапазон низьких і середніх температур використовуються в основному контактні методи вимірювання, як первинні вимірювальні перетворювачі в даному випадку використовуються термометри опору і термоелектричні перетворювачі. Для вимірювання високих температур використовують безконтактні методи вимірювання, для чого застосовують тепловізори.

Застосування безконтактних засобів вимірювань дозволяє проводити вимірювання температури об'єктів, що рухаються; об'єктів, розташованих у важкодоступних місцях; уникати пошкоджень засобів вимірювань під час контролю високих температур. Висока швидкодія, можливість вимірювання температури без відключення об'єкта від техпроцесу, забезпечення безпеки персоналу, вимірювання температури до 3000 ° С — основні переваги пірометрів та тепловізорів.

Тепловізор — засіб вимірювання температури теплового випромінювання об'єкта, призначене визначення значення температури і перетворення їх у візуальну картину розподілу теплових полів на поверхні об'єкта. Тепловізор дозволяє отримати узагальнену інформацію — теплову картину певної області та конкретне значення температури об'єкта, що цікавиться, розмір якого дорівнює або більше розміру елементарного осередку поля зору тепловізора.

Інфрачервоне випромінювання — електромагнітне випромінювання, що займає спектральну область між червоним кінцем видимого світла (з довжиною

хвилі $\lambda = 0,74$ мкм та частотою 430 ТГц) та мікрохвильовим радіовипромінюванням ($\lambda \sim 1-2$ мм, частота 300 ГГц).



Рисунок 1.1— Інфрачервоне випромінювання

Якщо об'єкт має температуру вище за абсолютну нульову точку 0 К ($-273,15$ °С), він випускає пропорційне своїй температурі електромагнітне випромінювання. Частина посланого випромінювання є інфрачервоним випромінюванням, яке застосовується для безконтактного вимірювання температури. Інфрачервоне випромінювання, що випускається об'єктом, проходить крізь атмосферу і може за допомогою лінзи або входної оптики фокусуватися на елемент детектора. Внаслідок влучення випромінювання елемент детектора створює пропорційний йому електричний сигнал. Перетворення сигналу в пропорційну температурі об'єкта вихідну величину здійснюється за допомогою посилення сигналу та подальшої цифрової обробки. Вимірювана величина може відобразитися на дисплеї або видаватися як сигнал.

Коефіцієнт випромінювання (епсілон) має основне значення при вимірюванні температури з використанням випромінювання. Коефіцієнт випромінювання вказує на співвідношення між реальною величиною випромінювання якогось тіла і величиною випромінювання чорного випромінювача при однаковій температурі. Для чорного випромінювача це співвідношення становить максимум 1. Насправді тіло відповідає ідеалу чорного випромінювача. Насправді для калібрування

датчиків використовуються поверхні випромінювача, які у необхідному діапазоні довжин хвиль досягають коефіцієнти випромінювання до 0,99.

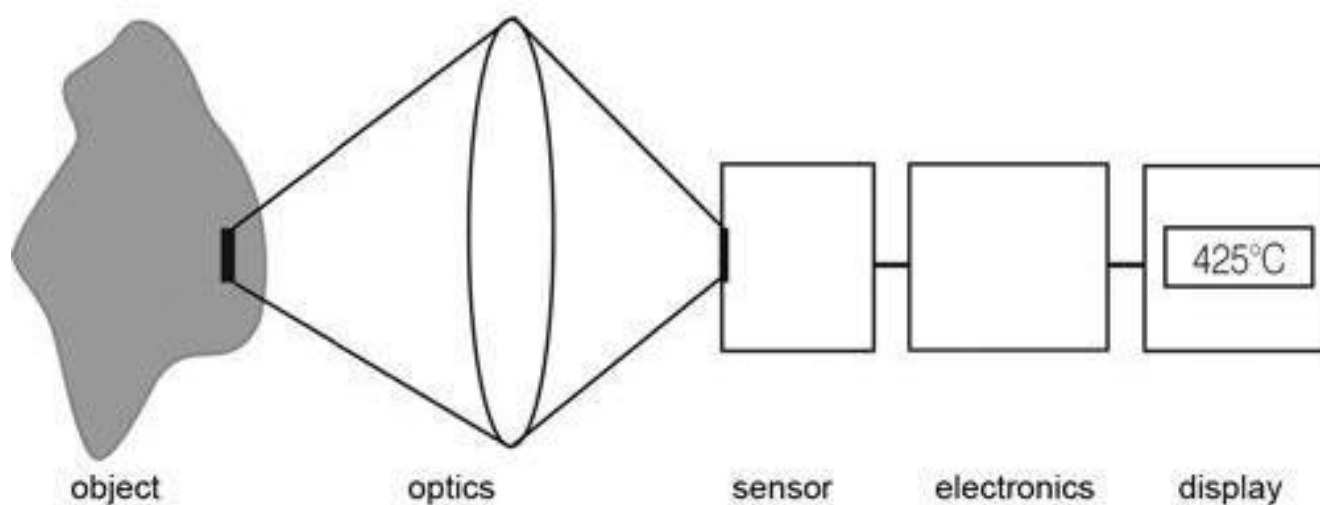


Рисунок 1.2 — Інфрачервоне вимірювання температури об'єкта

У разі коли необхідно виміряти температуру мікроскопічних об'єктів, теплоємність яких мала, безконтактний спосіб вимірювання дозволяє уникнути спотворень температури об'єкта, що вносяться контактним засобом. Безперечними перевагами безконтактних засобів вимірювання температури порівняно з контактними є:

- висока швидкодія (до 1мс) — необхідна у разі вимірювання температури швидкоплинних процесів;
- можливість вимірювання температури об'єкта без його відключення від технологічного процесу — виявлення точок перегріву, що виникають лише під час роботи об'єкт;
- забезпечення безпеки персоналу, що проводить вимірювання температури об'єктів, що знаходяться під напругою, оскільки роботи виробляються на відстані та не вимагають відключення електроенергії.

Тепловізори мають у порівнянні з контактними датчиками температури як ряд переваг, так і деякі недоліки:

- залежність показань від відстані до об'єкта, що вимірюється.
- залежність від відбивних властивостей вимірюваної поверхні.

— залежність від випромінювання областей вимірюваного об'єкта, що прямо не потрапляють у поле зору тепловізора.

1.2 Аналіз існуючих аналогів

1.2.1 Датчики температури

Інфрачервоний датчик — це електронний пристрій, який дозволяє побачити неозброєним поглядом зміни інтенсивності фонового випромінювання тепла.

Датчик у свою чергу посилає сигнал на електронну схему управління, після чого схема його обробляє та виводить дані на екран, або зберігає.

Розглянемо інфрачервоний датчик на прикладі Melexis MLX90640.

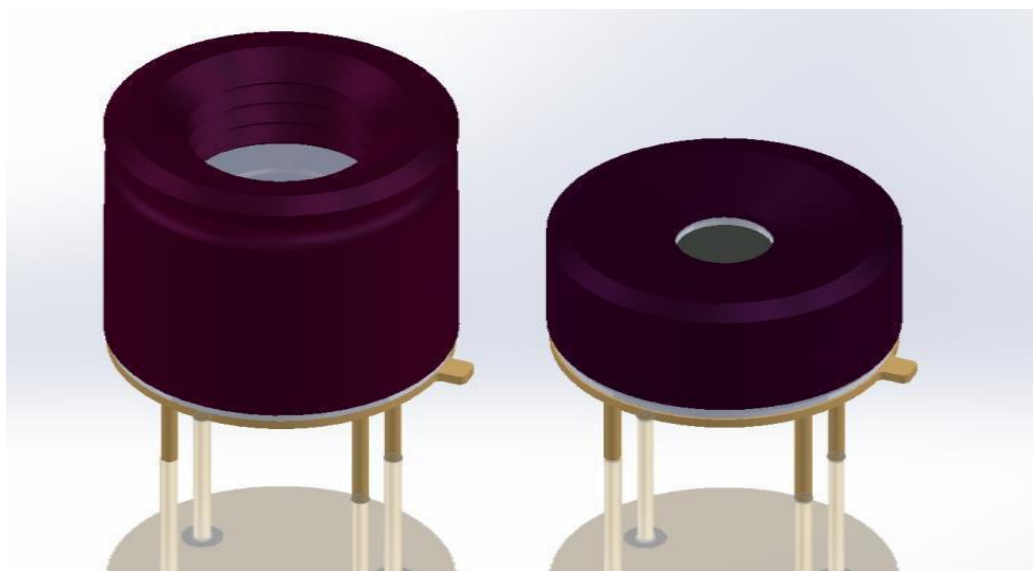


Рисунок 1.3 — Зовнішній вигляд інфрачервоного датчика MLX90640

MLX90640 має повністю відкалібровану теплову ІЧ-матрицю 32x24 пікселів у стандартному 4-вивідному корпусі TO39 з цифровим інтерфейсом.

MLX90640 містить 768 пікселів FIR. Датчик навколишнього середовища вбудований для вимірювання температури навколишнього середовища мікросхеми, а датчик живлення — для вимірювання VDD. Виходи всіх датчиків IR, Ta та VDD зберігаються у внутрішній пам'яті та доступні через I2C.

Характеристики та переваги:

1. Малі розміри та невелика вартість.
2. Легкість інтеграції.
3. Заводське калібрування у широкому температурному діапазоні. I2C-сумісний цифровий інтерфейс.
4. Напруга живлення 3,3 В.
5. Споживаний струм менше 23 мА.
6. Висока точність $0.5\text{ }^{\circ}\text{C}$ у широкому діапазоні температур ($0 \dots 50\text{ }^{\circ}\text{C}$).
7. Вимірювана температура $-40^{\circ}\text{C} \div 300^{\circ}\text{C}$.

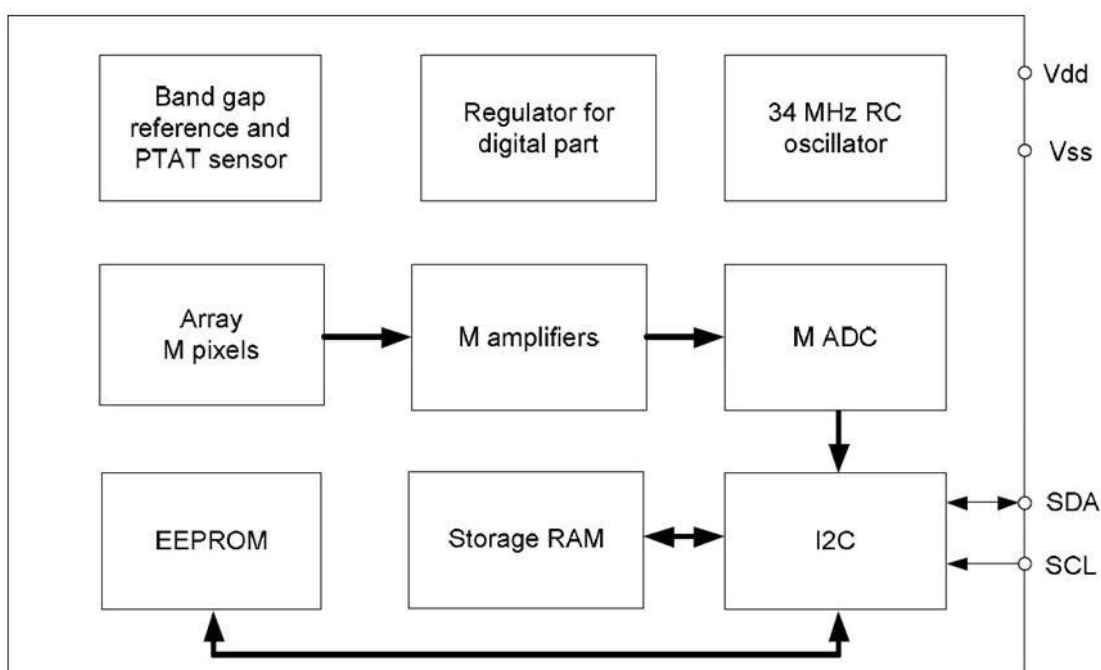


Рисунок 1.4 — Діаграма роботи датчика MLX90640

Інфрачервоний датчик температури складається з послідовно включених термоелементів. Термоелемент складається з двох електропровідних матеріалів, які розташовані у вигляді доріжок, що проводять, які в одній точці контактують один з одним. Якщо за рахунок зовнішнього впливу виникне різниця температур між точкою контакту та обома відкритими кінцями, то на обох кінцях термоелементів з'явиться напруга в кілька мілівольт.

При безконтактному способі вимірювання температури підвищення температури точки викликається за рахунок абсорбування інфрачервоного випромінювання, що потрапляє в цю точку. Кожен об'єкт випромінює

інфрачервоне світло, причому енергія цього світла збільшується з підвищенням температури об'єкта. Базуючись на цьому ефекті ІЧ-датчики вимірюють потужність, що випромінюється, і, таким чином, з високою точністю визначають температуру об'єкта.

В нормальних умовах кімнатної температури може використовуватися простий, неохолодний датчик. Для більш гарячих середовищ пропонуються моделі, що підтримують працездатність за високої температури навколишнього середовища або моделі з повітряним або водяним охолодженням.

Перешкоди, такі як пил, пара та дим, можуть вплинути на зчитування, тому пропонуються спеціальні датчики для отримання точних показань у дуже вологих або брудних умовах. Однак, якщо повітря виглядає ясным, то за допомогою датчика загального призначення можна легко отримати хороші результати.

До аналогів можна віднести однокристальний інфрачервоний цифровий датчик TMP006 виробництва компанії Texas Instruments.

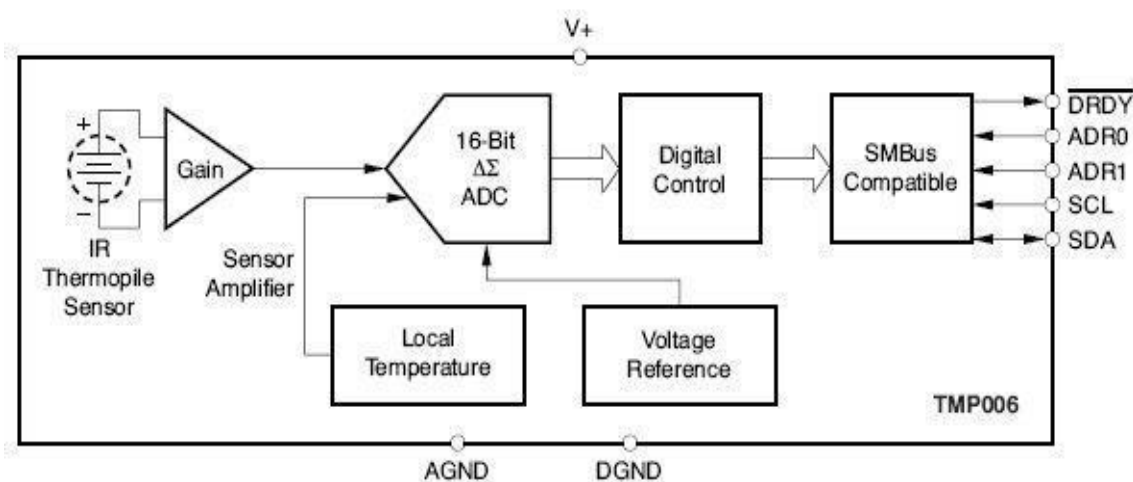


Рисунок 1.5 — Внутрішня архітектура TMP006

Сенсор використовує термопару для поглинання енергії інфрачервоного випромінювання об'єкта та, вимірюючи відповідні зміни вихідної напруги термопар, визначає його температуру. Робочий діапазон напруги інфрачервоного датчика дозволяє вимірювати температуру від -40°C до 125°C , що ідеально підходить для широкого спектру додатків. Мала потужність споживання поруч із

низькою робочою напругою дозволяє застосовувати датчик у портативних пристроях з живленням від акумуляторної батареї.

Характеристики та переваги:

1. Діапазон температур: $-40^{\circ}\text{C} \dots 125^{\circ}\text{C}$.
2. Послідовний інтерфейс, сумісний із SMBus. Програмована висновками адресація.
3. Робочий струм споживання: 240 мкА.
4. Діапазон напруги живлення: 2.2...5.5.

Також до аналогів відноситься матричний датчик температури Omron D6T.



Рисунок 1.6 — Зовнішній вигляд датчика температури D6T

Датчики базуються на термоелектричному ефекті. Термометричний сенсор здатний вимірювати температуру об'єкта шляхом визначення випромінюваної енергії в інфрачервоному спектрі. Крім того, датчик температури D6T здатний вимірювати температуру в декількох областях одночасно (8 та 16 областей). Чутливі осередки моделі D6T-44L-06 організовані як матриці 4×4 , а моделі D6T-8L-06 як матриці 1×8 . Сигнал із осередків надходить на спеціалізовану схему, що виробляє його посилення, обробку та передачу на інтерфейсний контролер. Матриця осередків дозволяє отримувати інформацію про розподіл температури площі, охопленої полем зору датчика.

Характеристики та переваги:

1. Діапазон вимірюваних температур $5\text{-}50^{\circ}\text{C}$.
2. Точність вимірювання: $\pm 1.5^{\circ}\text{C}$.

3. Вихідний інтерфейс PC.
4. Напруга живлення 4.5 ... 5.5 В.
5. Струм споживання 5 мА.

1.2.2 Тепловізори

Розглянемо тепловізор Seek Thermal Compact. Головна особливість цього тепловізора в тому, що його можна безпосередньо підключити до смартфона як під керуванням операційної системи IOS, так і Android. Мобільний тепловізор Seek Thermal Compact є невеликою зовнішньою камерою-приставкою для смартфона, в якій передбачена мікро-болометрична матриця зі спеціальною LWIR-лінзою.



Рисунок 1.7 — Зовнішній вигляд тепловізора Seek Thermal Compact



Рисунок 1.8 — Приклад роботи тепловізора Seek Thermal Compact

Характеристики та особливості:

1. Роздільна здатність теплового сенсора: 206 x 156 пікселів.

2. Кут огляду: 36 градусів.
3. Дальність: до 300 метрів.
4. Температурний діапазон: -40°C до 330°C .
5. Частота розгортки: 9 Гц.
6. Висока ціна: від 8 000 грн.

Розглянемо тепловізор Flir One Pro. Даний тепловізор також є приставкою для смартфона, в якій на відміну від seek thermal встановлена високоякісна камера видимого світла. За допомогою запатентованої технології «Flir MSX» можливе поєднання видимого та теплового зображень, що дає високодеталізовану картинку.



Рисунок 1.9 — Зовнішній вигляд тепловізора Flir One Pro



Рисунок 1.10 — Приклад роботи тепловізора Flir One Pro з технологією Flir MSX

Характеристики та особливості:

1. Теплова роздільна здатність: 160 x 120.
2. Оптична роздільна здатність: 640 x 480.

3. Частота кадрів: 8,7 Гц.
4. Діапазон температур: $-20^{\circ}\text{C} + 120^{\circ}\text{C}$.
5. Чутливість: 0.1°C .
6. Точність виміру: $\pm 2^{\circ}\text{C}$.
7. Кут огляду: 40° .
8. Повертання зображення (0° , 90° , 180° , 270°).
9. Висока вартість від 15000 грн.

1.3 Огляд відлагоджувальних плат для мікроконтролерів

Сигнал з датчиків передається на мікроконтролер, який програмується користувачем на виконання конкретних дій, наприклад, виведення температури на дисплей.

Наш час для мікроконтролерів створено багато різних платформ, які спрощують розробку різних пристроїв, а також роботу з мікроконтролером. Наприклад, за рахунок вбудованого програматора, готову плату можна легко підключити до комп'ютера через

USB-порт, і через спеціальне середовище розробки, у кожній платформі вона своя, написати та завантажити код у мікроконтролер. Також можна легко підключати різні пристрої до мікроконтролера через піни, які розташовані на платі.

Ознайомимося з деякими варіантами платформ для мікроконтролерів:

NUCLEO-L053R8 — налагоджувальна плата на базі 32-розрядного мікроконтролера STM32L053, який має 64 КБ флеш-пам'яті, 8 КБ ОЗП і тактову частоту 32 МГц. У платі є вбудований емулятор ST-LINK/V2, тому підключення зовнішнього програма тора — налагодження не потрібне. Має власне програмне забезпечення STM32CubeMX.

Підтримує функцію віртуального COM-порту через USB. Сумісна з модулями розширення Arduino Uno R3 [13].

Teensy++ 2.0 — відлагоджувальна плата на базі 8-бітного AVR процесора AT90USB1286 з такими характеристиками: 128 КБ флеш-пам'яті, 8 КБ ОЗП, що працює на частоті 16 МГц. Плата має 8 аналогових входів та 38 цифрових

входів/виходів. Має вбудований програматор. Для програмування можна використовувати Arduino IDE [14].

NodeMcu v3 — платформа на основі чіпа ESP8266 з 32-бітним процесором Tensilica Xtensa LX106 з тактовою частотою 80 МГц. Плата має 4 МБ флеш-пам'яті та 20 Кб ОЗП. Напруга живлення 5-12В. Живлення можна подавати через роз'єм Micro USB, який розташований на платі, так само через нього завантажуються програмний код. На платформі є 11 портів вводу/виводу.

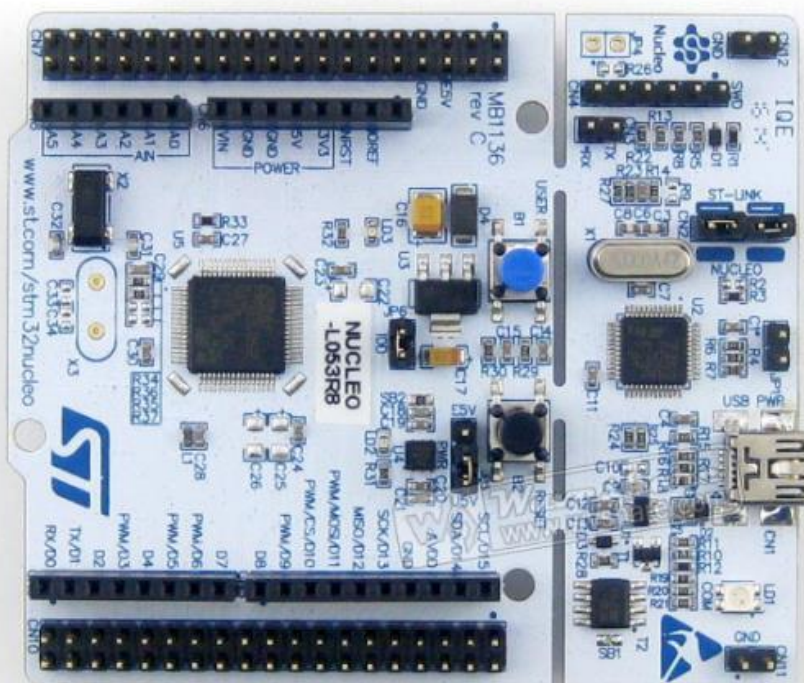


Рисунок 1.11 — Відлагоджувальна плата NUCLEO-L053R8

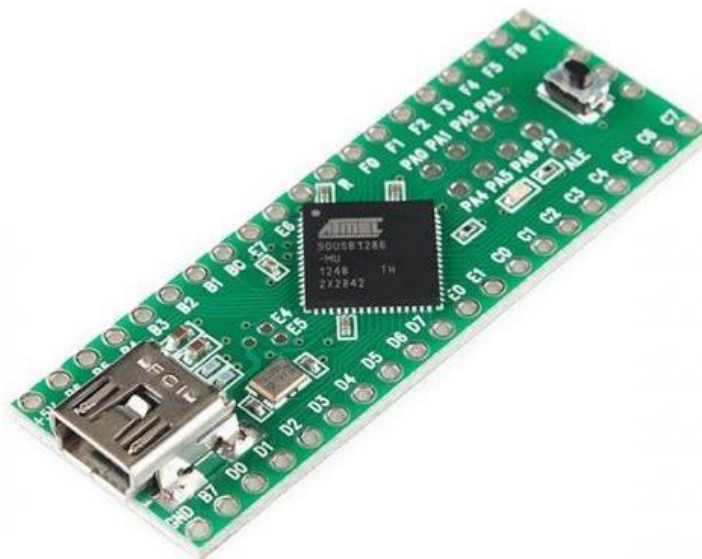


Рисунок 1.12 — Відагоджувальна плата Teensy++ 2.0

На платі розташований модуль Wi-Fi і є вбудована підтримка протоколу TCP/IP. Для програмування NodeMcu можна використовувати програмне забезпечення ESPlorer або Arduino IDE [15].

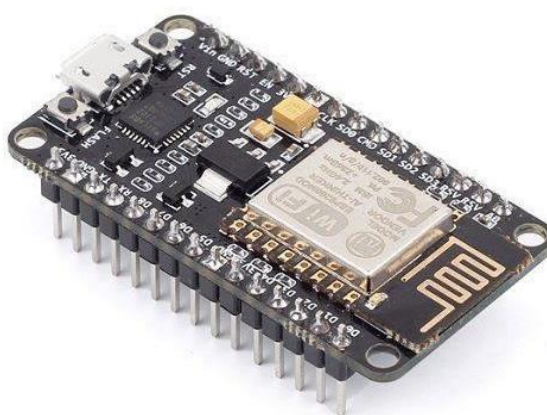


Рисунок 1.13 — Платформа NodeMcu

1.4 Види індикаторів для відображення результатів вимірювань

Для того, щоб була можливість контролювати температуру в даний момент часу, її необхідно відображати на індикаторі. Для виведення рядка виду: $T = 25,75^{\circ} \text{C}$ дисплей повинен мати можливість відображати щонайменше 10 символів у рядку. Розглянемо кілька варіантів дисплеїв для індикації температури:

TFT01-22SP — графічний дисплей з роздільною здатністю 240x320 пікселів. Може відображати графічні елементи, наприклад, зображення та текст. У ньому використовується мікроконтролер ІЛІ9340С. Діагональ 2,2 дюйми. Для підключення використовується інтерфейс SPI. Напруги живлення 5В [16].

3641BS — 4-х розрядний 7-ми сегментний індикатор із загальним анодом. Для відображення інформації необхідно по черзі подавати напругу на аноди, а необхідну інформацію формувати катодами. Робиться це по черзі кожному за розряду [17].



Рисунок 1.14 — Кольоровий графічний екранTFT01-22SP

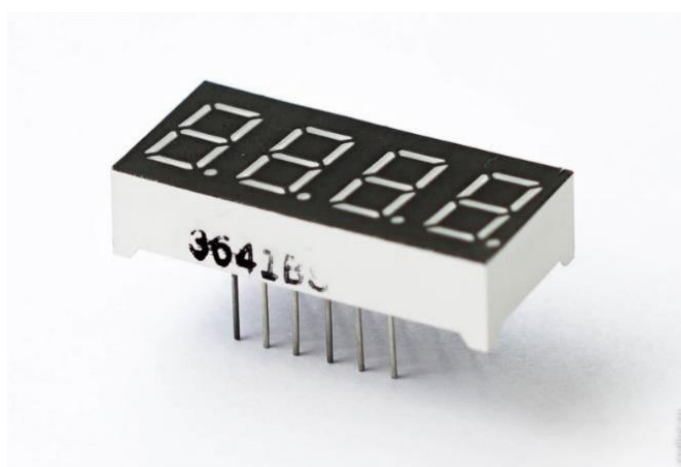


Рисунок 1.15 — Матриця із 4-х 7-ми сегментних індикаторів

1.5 Пам'ять для зберігання вимірювань температури

Простого виміру температури під час проведення дослідів недостатньо, її необхідно фіксувати, тобто зберігати у файл, задля її подальшого аналізу. Зберігатися дані можуть, наприклад, на флешці. Наприклад розглянемо модуль розширення підключення microSD – карт.

SD Shield Arduino — модуль розширення Arduino для підключення до неї SD/microSD – карт. Для роботи використовує інтерфейс SPI [18].



Рисунок 1.16 — Модуль розширення пам'яті Arduino

2 ВИБІР МІКРОПРОЦЕСОРНОЇ ПЛАТФОРМИ

2.1 Вибір мікроконтролера

Мікроконтролер — це інтегральна схема, здатна виконувати програму. Сьогодні на ринку представлено безліч таких моделей від самих різних виробників. Ціни на ці пристрої продовжують падати. Однокристалічні чипи знаходять широке застосування в самих різноманітних сферах: від вимірювальних приладів до виробів розваг і все можливої домашньої техніки. У відмінності від персональних комп'ютерів мікроконтролер поєднує в одному кристалі функцій процесора та периферійних пристроїв, містить оперативну пам'ять і постійний запам'ятовуючий пристрій для зберігання кодів і даних, однак має значні обмеження обчислювальним ресурсам.

Мікроконтролер ESP32 – це одна з найдоступніших і найпотужніших платформ для створення розумних проектів з підтримкою WiFi. Прийшовши на зміну ESP8266, цей чіп дав нові можливості для розробників, хоча, як і раніше, залишилися старі проблеми з підтримкою та документацією.

Відмінності esp32 від esp8266

Платформи ESP8266 та ESP32 вироблені однією компанією Espressif. Мікроконтролер ESP32 відрізняється від свого попередника покращеними характеристиками, збільшеним функціоналом та великим обсягом пам'яті. Вартість нового приладу дорожча приблизно вдвічі.

Важливою перевагою ESP32 є швидший Wi-Fi та Bluetooth. У ESP32 встановлений потужніший процесор, що дозволяє реалізовувати складні проекти. Ця платформа підходить для додатків, які потребують інтернет або нові інтерфейси. Для найдешевших розробок використовується ESP8266.

Об'єм пам'яті нового пристрою ESP32 збільшений – 512 Кб проти 160 Кб ESP8266. ESP32 також відрізняється великою кількістю виводів GPIO. До кількох контактів на ESP32 прикріплені сенсорні ємнісні датчики і датчик температури. На

обох пристроях можна використовувати контакти GPIO по-різному. ESP32 має 18 12-бітових АЦП каналів. У його попередника є лише 1 10-бітовий вивід АЦП.

Потужність процесора значно впливає швидкість роботи. Модуль ESP32 показує рекордну продуктивність у порівнянні із попередником ESP8266. Завантаження сторінки з довгим скетчем та безліччю графіки займає секунди.

З недоліків ESP32 можна виділити відсутність бібліотек для підтримки сенсорів та мала кількість драйверів. Це з тим, що плата з'явилася у продажу недавно. Але з огляду на всі переваги мікроконтролера та його перспективи, цю проблему буде вирішено вже найближчим часом.

Фірма Espressif випустила потужний недорогий мікроконтролер ESP32 улітку 2016 року. Пристрій являє собою систему на кристалі, побудовану за технологією TSMC 40 нм, з Wi-Fi та Bluetooth контролерами. Він оснащений двоядерним 32-бітним процесором, який працює на частотах 80, 160 або 240 МГц. Також у систему інтегровані антенні комутатори, радіочастотні компоненти, фільтри, підсилювачі, модулі керування живленням. Підключається ESP32 до комп'ютера через стандартний USB провід. Плати з ESP32 мають гарну обчислювальну здатність, розвинену периферію і при цьому дуже популярний вид низьких цін у діапазоні 7\$ – 14\$.



Рисунок 2.2 — Мікросхема ESP 32

Технічні характеристики ESP32:

- дво- або одноядерний 32-бітний процесор Tensilica Xtensa LX6;
- тактова частота — 160 чи 240 МГц;
- 520 Кб SRAM;
- максимальний струм споживання 260 мА, в режимі сну — 10 мА;
- Стандарти бездротового зв'язку — Wi-Fi: 802.11 b/g/N, Bluetooth: v4.2 BR/EDR and BLE;
- наявність датчиків температури, Холла, тач-сенсорів;
- інфрачервоне дистанційне керування;
- можна підключати двигуни та світлодіоди через ШІМ роз'єм;
- стандарт IEEE 802.11 з підтримкою WPA, WPA/WPA2 та WAPI;
- можливість безпечного завантаження;
- шифрування флеш-диска.

Також у модулі традиційно присутній вбудоване управління енергоживленням. Для цього використовуються лінійний регулятор, індивідуальне живлення RTC (ядро низького енергоспоживання), пробудження по таймеру або сенсорному датчику.

Мікроконтролер оснащений 48 контактами плюс 1 великий тепловий контакт, які мають різні функції. Виводи мікросхеми:

- 18 каналів 12-розрядного АЦП;
- 10 GPIO портів;
- 3 SPI;
- 3 UART;
- 2 I2C;
- 16 ШІМ висновків;
- 2 8-бітних ЦАП виведення.

Розташування виводів відлагоджувальних плат залежить від виробника. Наприклад, є плата ESP32 DEVKIT V1 DOIT, яка має 36 контактів. Розташування виводів представлено на нижче.

ESP32 DEVKIT V1 – DOIT version with 36 GPIOs

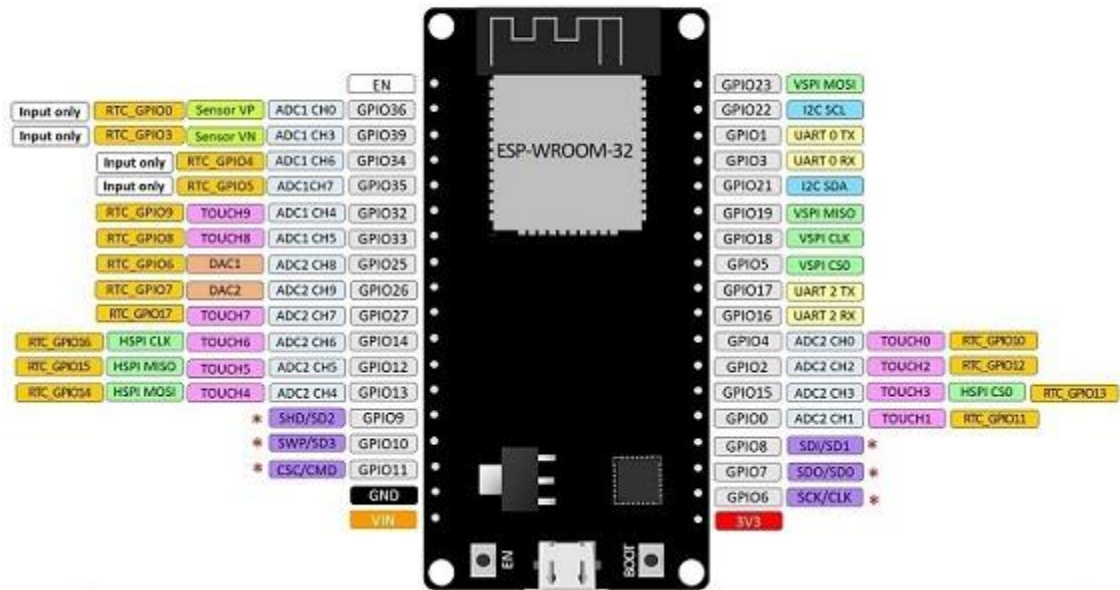
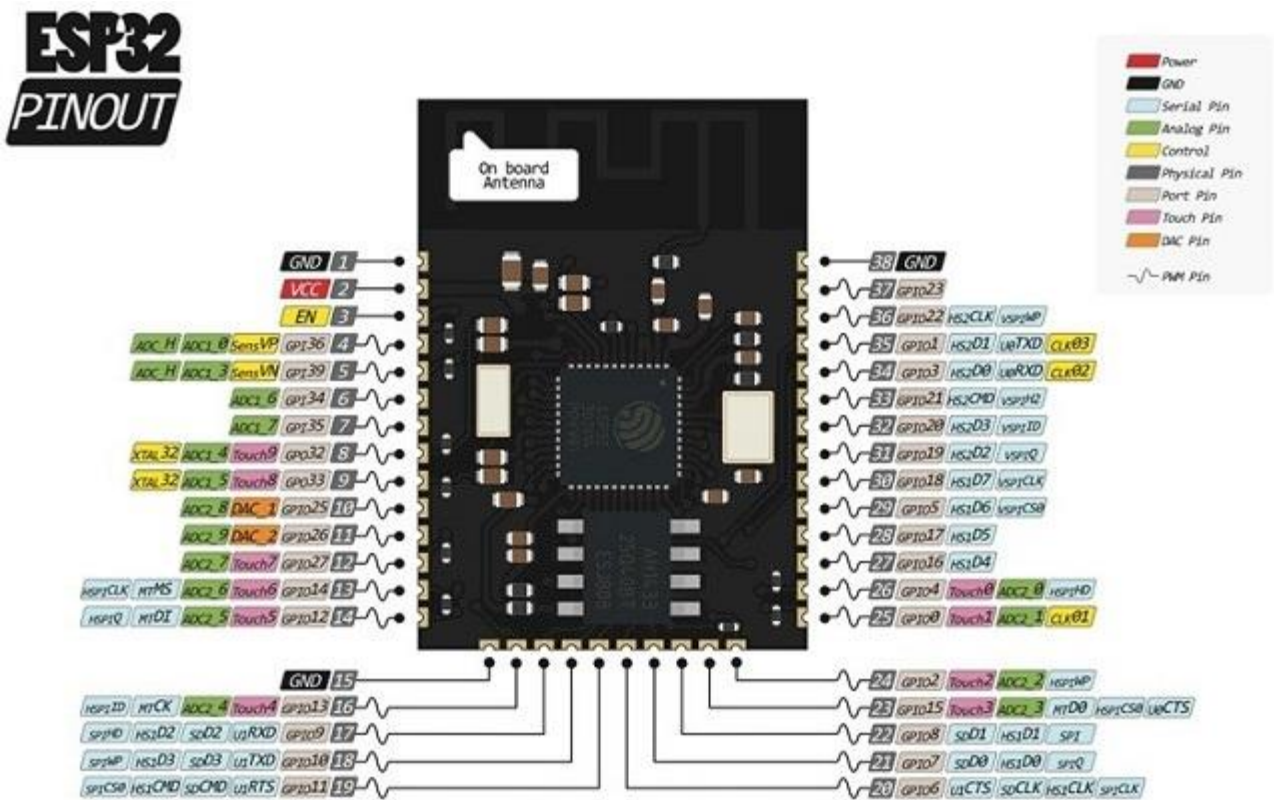


Рисунок 2.2 — Розташування виводів ESP32 DEVKIT V1 DOIT

Одним із найпопулярніших модулів є ESP-WROOM-32. Розташування виводів також наведено на зображенні.



Рис

унок 2.2 — Розташування виводів ESP WROOM 32

До портів GRIО 0, 4, 2, 15, 13, 12, 14, 27, 33 та 32 підключені сенсорні виводи. Вони можуть використовуватися для виведення ESP32 із глибокого сну. Вони фіксують будь-яку зміну електричного заряду.

Порти з 34 до 39 використовуються лише для введення інформації. На них відсутні підтягуючі резистори, тому їх не можна використовувати як виходи.

На деяких платах відображаються контакти 6-11. Вони приєднані до інтегрованої SPI flash. Їх не використовують у проектах.

Існують різні версії плати ESP32 розмірами 5x5 мм або 6x6 мм. Також на основі ESP32 є модулі SMT для інтегрування в інші плати.

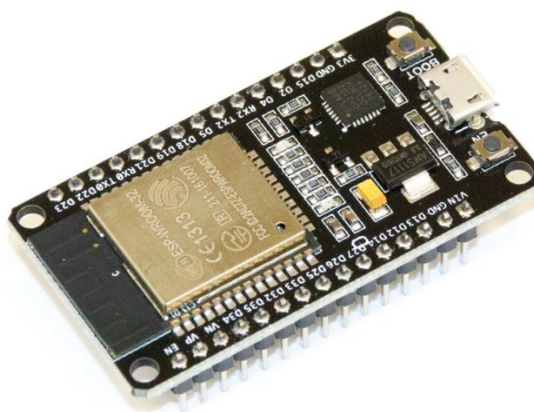


Рисунок 2.3 — Зовнішній вигляд ESP WROOM 32

Відповідно модуль містить:

MCU:

- вбудований ESP32-D0WD-V3, двоядерний 32-розрядний мікропроцесор Xtensa LX6, до 240 МГц;

- 448 КБ ПЗП для завантаження та основних функцій;

- 520 КБ SRAM для даних та інструкцій;

- 16 КБ SRAM в RTC.

Wi-Fi:

- 802.11b/g/n;

- швидкість передачі даних: 802.11n до 150 Мбіт/с;

- агрегація A-MPDU та A-MSDU;

- підтримка захисного інтервалу 0,4 мкс;

— центральний частотний діапазон робочого каналу: 2412 ~ 2484 МГц.

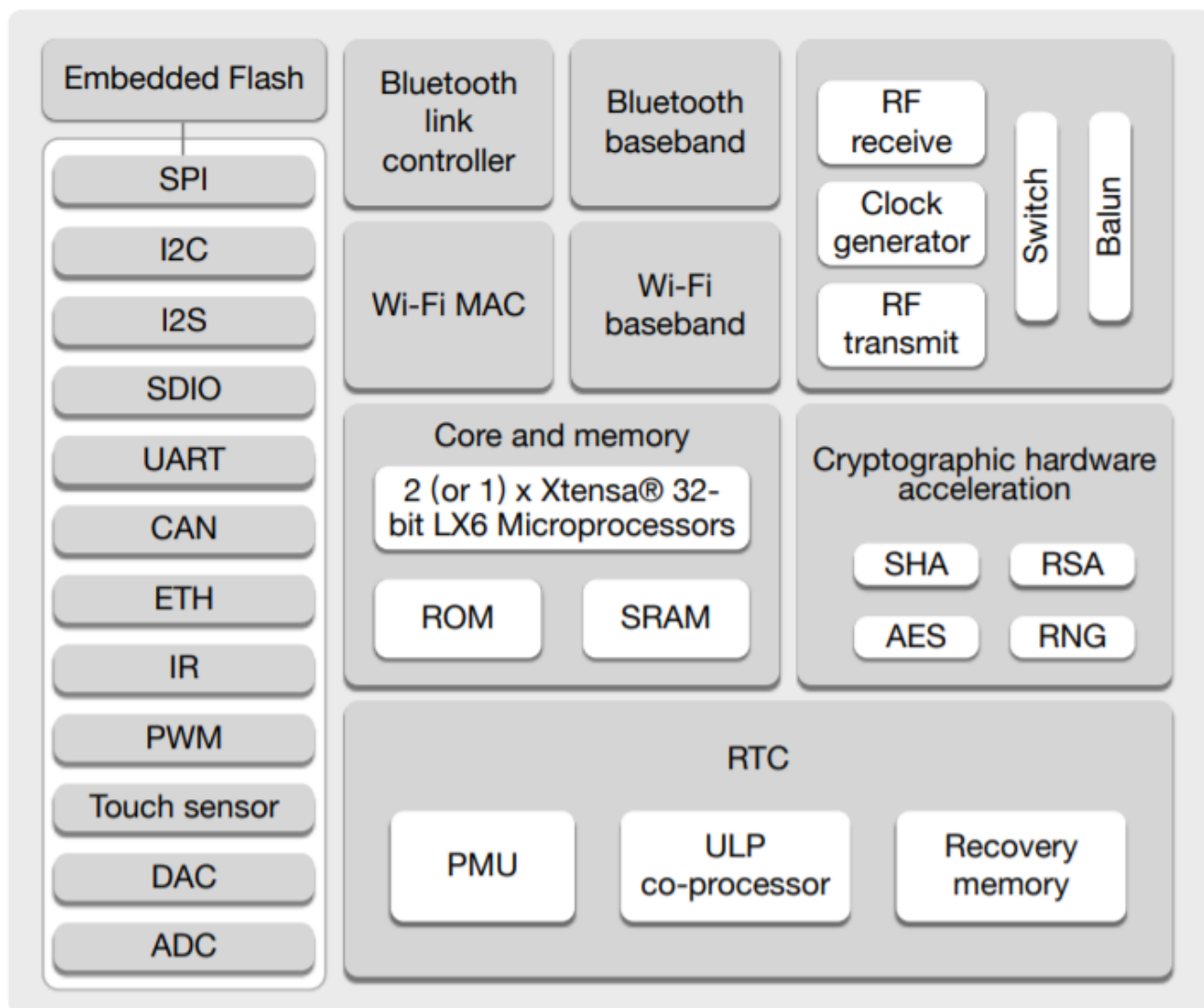


Рисунок 2.4 — Функціональна діаграма ESP WROOM 32

Bluetooth:

- Специфікація Bluetooth V4.2 BR/EDR і Bluetooth LE;
- Передавач класу 1, класу 2 і класу 3;
- AFH;
- CVSD і SBC.

Апаратне забезпечення:

- 40 МГц кварцевий генератор;
- 4 МБ SPI Flash;
- інтерфейси: SD-карта, UART, SPI, SDIO, I²C, LED PWM, двигун PWM, I²S, I²S, ІЧ, лічильник імпульсів, GPIO, ємнісний сенсорний датчик, АЦП, ЦАП

- робоча напруга/живлення: 3,0 ~ 3,6 В;
- діапазон робочих температур: -40 ~ 85 °С.

Сертифікація:

- сертифікація Bluetooth: BQB;
- сертифікація: FCC/CE-RED/SRRC;
- зелена сертифікація: REACH/RoHS.

2.2 Архітектура платформи TTGO T4 V1.3

Вибір пав на мікроконтролер ESP32. При виборі мікроконтролера у розгляд потрапила плата TTGO T4 V1.3 побудована на основі мікроконтролера ESP32 і є відкритою платформою, яка дозволяє збирати всілякі електронні пристрої. Платформа складається з програмної та апаратної частин. TTGO T4 має вбудований контролер заряду та може живитися як від USB підключення, так і від зовнішнього джерела живлення — батарейки. Плата оснащена рідкокристалічним дисплеєм на тонкоплівкових транзисторах (технологія TFT) величиною 2.2 дюйми з роздільною здатністю 240*320 пікселів. Дисплей має контролер ILITEK ILI9341, що підключається до ESP32 за допомогою інтерфейсу SPI. У свою чергу ESP32 підтримує прямий доступ до пам'яті (DMA), що досить зручно при оновленні дисплея. Завдяки цьому на частоті 26МГц час повного оновлення екрана в 16-бітному кольорі дорівнює 46 мс, що досить прийнятно для роботи тепловізора.

На стороні екрана плата має 3 кнопки (IO37, IO38, IO39) для керування цим пристроєм. Збоку плати також знаходяться кнопки (RST) – перезавантаження пристрою та (BOOT) – увімкнення/вимкнення пристрою.

Плата TTGO T4 V1.3 оснащена 32-х розрядним процесором ESP32, який використовує два мікроконтролерні ядра Tensilica Xtensa LX6 з частотою 240MHz на одне ядро від компанії Espressif Systems.

Також платформа містить різновид напівпровідникової технології, що електрично перепрограмується (flash -- пам'ять) з інтерфейсом SPI величина якої становить 4Мб, використовувана мікропроцесором ESP32 для зберігання своєї програми і також частково може використовуватися для зберігання даних. А також

є мікросхема оперативної пам'яті об'ємом 4Мб, що має тип псевдостатичної пам'яті.



Рисунок 2.5 — Зовнішній вигляд плати TTGO T4 V1.3 (вид спереду)

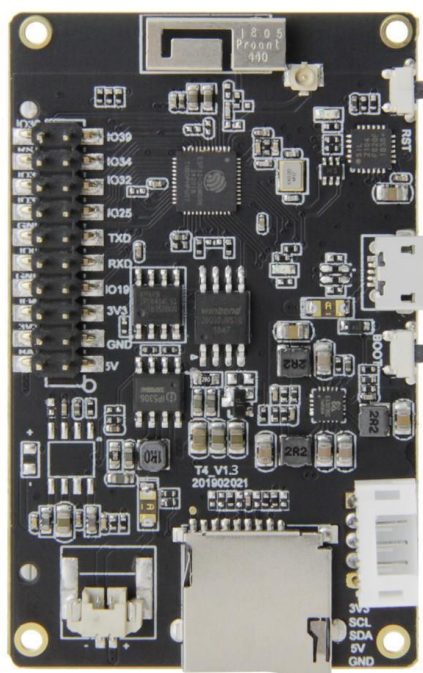


Рисунок 2.6 — Зовнішній вигляд плати TTGO T4 V1.3 (вид ззаду)

PSRAM.

На платі розташовані такі інтерфейси та роз'єми як:

Роз'єм для підключення microSD flash пам'яті, який слугує для збереження зображення та графіків, для подальшої обробки інформації на персональному комп'ютері.

Гніздо для підключення модулів розширення формату GROVE, який використовує інтерфейс I2C.

Антенна Wi-Fi має радіочастотну сертифікацію FCC/CE/TELEC/KCC/SRRC/NCC, що працює за протоколами 802.11 b/g/n/e/i (802.11n, зі швидкістю до 150Mbps) на частоті 2.4ГГц~2.5ГГц із потужністю передачі 22dBm. Також є роз'єм для підключення зовнішньої антени.

Антенна bluetooth v4.2BR/EDR та BLE стандарту з радіочастотою чутливістю — 98дБм.

Роз'єм розширення на який виведено кілька ліній з ESP32 та лінії живлення (див. рис.2.7).

Роз'єм MicroUSB, який використовується для живлення всієї плати, у тому числі для заряджання акумуляторної батареї. Інтерфейс USB працює від мікросхеми USB-UART міст від компанії SiLabs CP2104. ESP32 підключається через USB-UART міст до роз'єму MicroUSB до персонального комп'ютера для прошивки, виводу налагодження або для будь-якого обміну інформацією, що досить зручно.

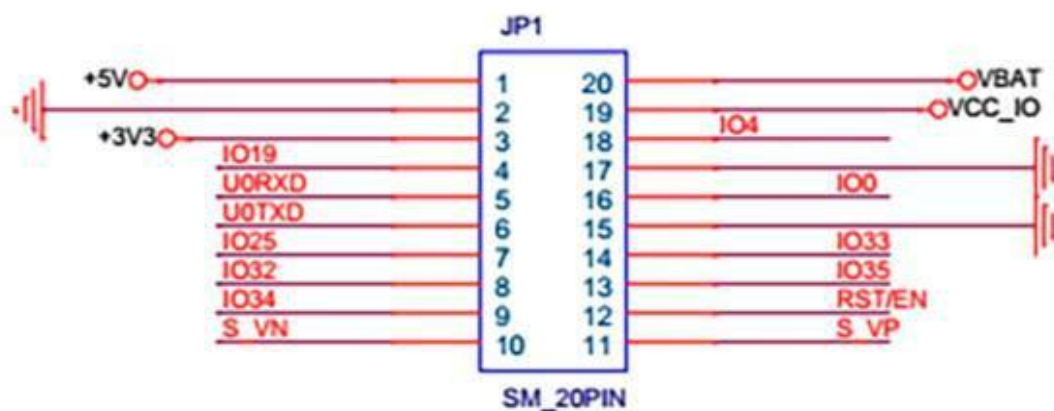


Рисунок 2.7 — Призначення виводів роз'єму розширення

Роз'єм для підключення акумуляторної батареї, що управляється мікросхемою Injoinic IP5306. Дана мікросхема управляє контролем заряду та розряду акумулятора, а також веде управління живлення всієї плати.

Живлення плати може здійснюватися від USB при 5В і 2А, так і від літієвої акумуляторної батареї 3.7В.

Плата TTGO T4 V1.3 на програмному рівні має багатозадачну операційну систему реального часу для систем FreeRTOS, що вбудовуються.

Габарити плати TTGO T4 V1.3:

- висота: 65.86мм;
- ширина: 11мм;
- довжина: 40.77мм.

2.3 Програмування

Програмування модуля ESP32 може проводитися на різних платформах, і середовищах, ось невеликий список найбільш популярних варіантів:

- Arduino IDE;
- Espressif IoT Development Framework;
- Espruino;
- PlatformIO;
- PyMaker IDE.

Більшість проектів реалізується на Arduino IDE та Espruino.

Програмування під плату TTGO T4 V1.3 може здійснюватися як через Arduino IDE, і через SDK від Espressif Systems, яка називається ESP-IDF, а тому розглянемо їх.

Налаштування Arduino IDE для роботи з ESP32

Спочатку вироби від компанії Espressif поставляються з прошивкою, що дозволяє працювати за допомогою команд AT. Це не завжди зручно, тому краще програмувати плату у звичному середовищі розробки – Arduino IDE .



Рисунок 2.8 — Середовище розробки Arduino IDE

Щоб почати створювати проекти на ESP32, спочатку потрібно мати його підтримку серед розробки Arduino IDE. Підтримка зараз перебуває на початковому, але працездатному рівні. Проблеми можуть виникнути з драйверами пристроїв, але через популярність модуля незабаром підтримуватиметься вся периферія.

В першу чергу потрібно завантажити Arduino IDE та офіційну інструкцію з сайту <https://github.com/espressif/arduino-esp32>. Потрібно завантажити файли дистрибутива і помістити в папку `C:\Users\User\Documents\Arduino\hardware\espressif\esp32`.

Так як плата ESP32 нова і підтримка в середовищі розробки Arduino IDE з'явилася нещодавно, можуть виникнути труднощі з установкою. Це з тим, що драйвера лише розробляються і змінено порядок розташування системних директорій. Усі скетчі повинні бути розташовані в папці `C:\Users\User\Documents\Arduino`. Файли дистрибутива повинні бути розміщені всередині цієї папки, як рекомендує виробник. Якщо цього не врахувати, то підтримка в Arduino IDE ESP32 не буде.

Після інсталяції можна відкрити Arduino IDE. Потім потрібно перейти в налаштування та в менеджері плат вибрати потрібну. Тепер можна прошивати модуль.

Налаштування ESP-IDF для роботи з ESP32

Налаштування офіційного середовища розробки ESP-IDF (Espressif IoT Development Framework).

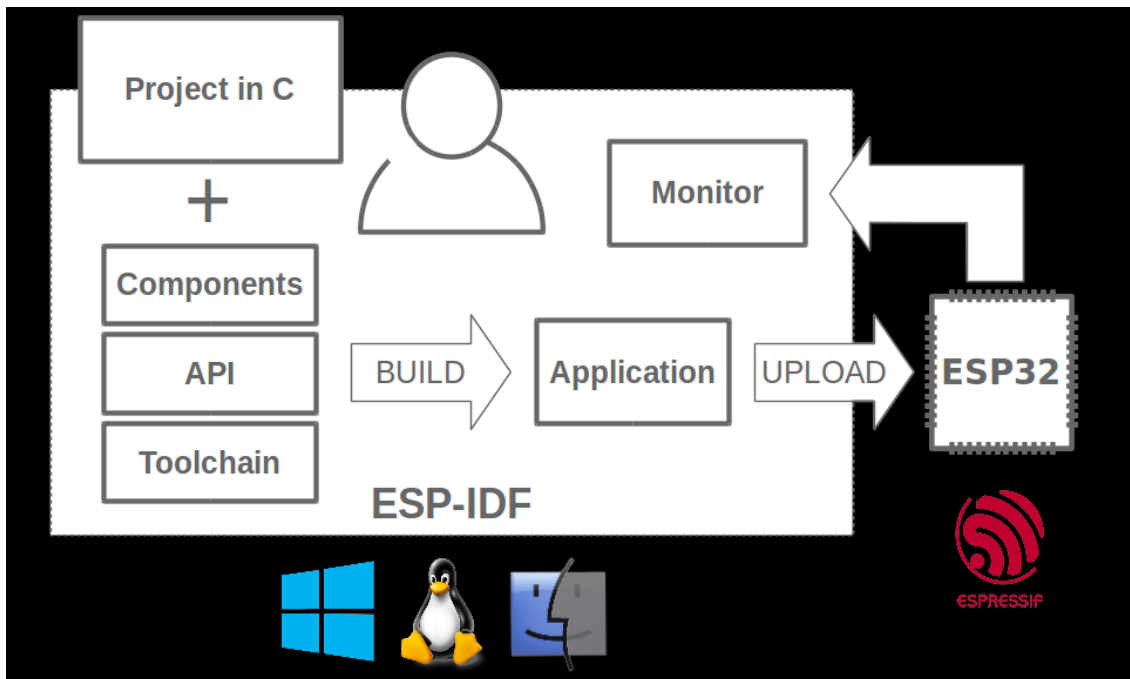


Рисунок 2.9 — Середовище розробки ESP-IDF

Його можна встановити на Windows. Для цього потрібно:

- завантажити з офіційного сайту середовище розробки;
- розпакувати архів, помістити його на диск C;
- потім з'явиться каталог `msys32`, до якого потрібно перейти та запустити програму `mingw32.exe`;
- відобразиться вікно терміналу, де потрібно створити каталог з ім'ям `esp`;
- потрібно перейти до створеного каталогу;
- підключити необхідні бібліотеки за допомогою команди `git clone -recursive https://github.com/espressif/esp-idf.git` ;
- потім встановлюються пакети Python `python -m pip install -user -r $IDF_PATH/requirements.txt`.

Після цього можна розпочинати роботу. У каталозі `\esp-idf\examples` є різні приклади скетчів для роботи з модулем.

3 ПРОЕКТУВАННЯ МІКРОПРОЦЕСОРНОЇ СИСТЕМИ БЕЗКОНТАКТНОГО ТЕРМОМЕТРА

3.1 Структурна схема

Структурна схема розробляється на початкових стадіях проектування та передує розробці схем інших типів. Структурна схема визначає основні функціональні частини виробу, їх призначення та взаємозв'язки між ними. Схема відображає принцип дії виробу у загальному вигляді. Структурна схема системи представлена рисунку 3.1.

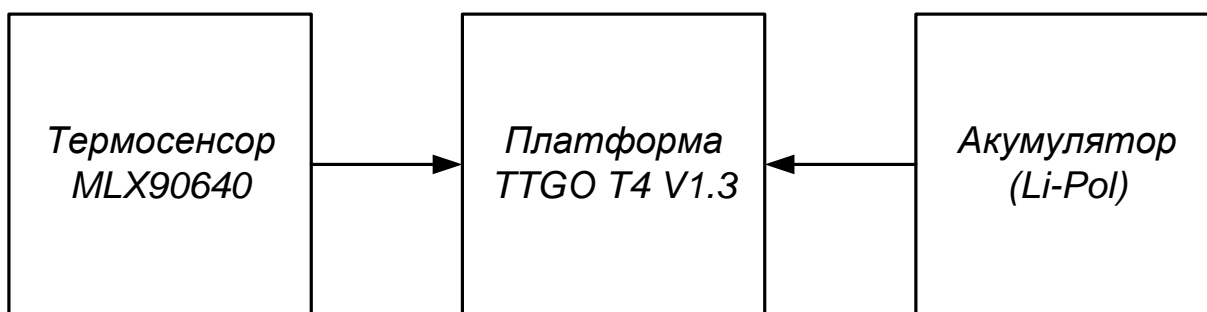


Рисунок 3.1 — Структурна схема системи

Структурна схема складається з наступних блоків:

Власне платформи TTGO T4 V1.3, характеристики якої розглянуто в попередньому розділі. Платформа може реалізувати абсолютно усі функції, крім сприйняття інфрачервоного випромінювання. Для сприйняття випромінювання візьмемо термосенсор MLX90640, який було розглянуто в попередньому розділі і який найбільш оптимально влаштовує вимоги системи, що проектується.

Акумулятор літій-полімерний (Li-Pol) необхідний для живлення системи. Виходячи із значень напруги, струму споживання та сумарної потужності платформи і сенсора оберемо акумулятор з параметрами 1100мА*год 3.7В із захистом.



Рисунок 3.2 — Акумулятор літій-полімерний (Li-Po)

3.2 Принципова схема

Принципова схема — схема, що визначає повний склад елементів та зв'язку між ними та дає детальне уявлення про принципи роботи виробу. Принципова схема є підставою для розробки конструкторської документації, а також використовується при налагодженні, регулюванні, контролі та ремонті виробу.

Опираючись на технічну документацію, окрім основних блоків та елементів, потрібно ще окремі компоненти для правильної роботи системи.

Опис компонентів розміщених на макетній платі:

1. R1 — резистор номіналом рівний 5.1 кОм, 5%.
2. R2 — резистор номіналом рівний 5.6 кОм, 5%.
3. R3 — резистор номіналом рівний 1 кОм, 5%.
4. R4 — резистор номіналом рівний 1 кОм, 5%.
5. C1 — монолітний конденсатор з керамічним чіпом, ємність якого дорівнює 47 мкФ.
6. C2 — SMD конденсатор, ємність якого дорівнює 47 мкФ.
7. C3 — SMD конденсатор, ємність якого дорівнює 1 мкФ.

Електрична принципова схема показана на рисунку 3.3.

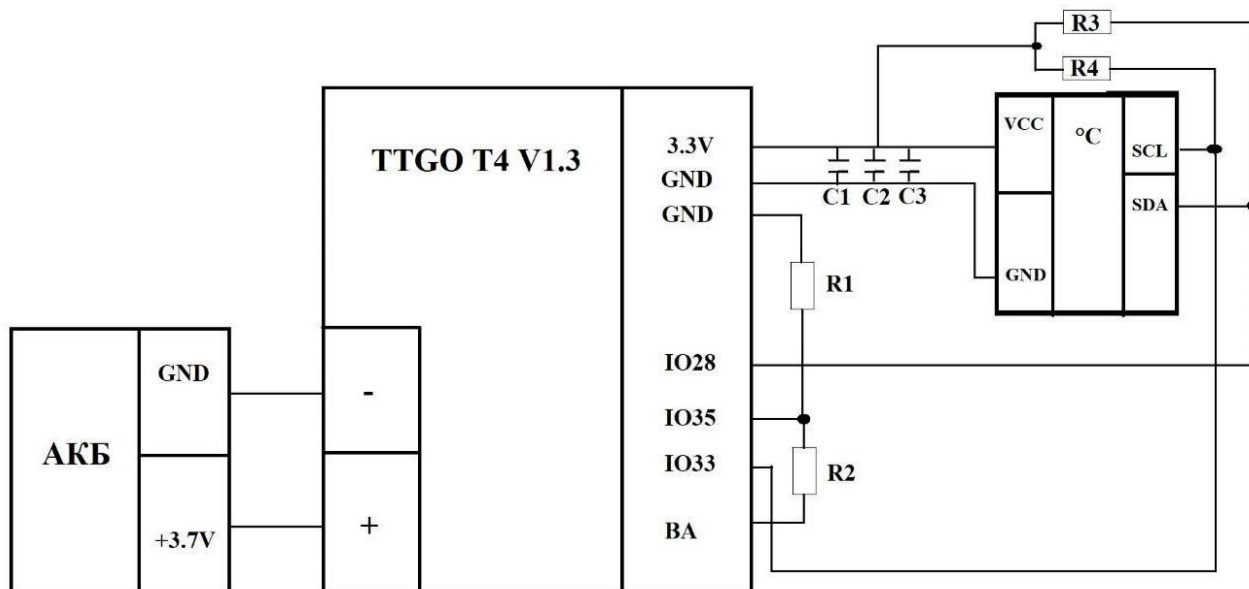


Рисунок 3.3 — Схема електрична принципова системи

Макетна плата з розведенням, резисторами та конденсаторами певних номіналів, а також інфрачервоним датчиком температури MLX90640 показано на рисунках 3.4 і 3.5 відповідно.

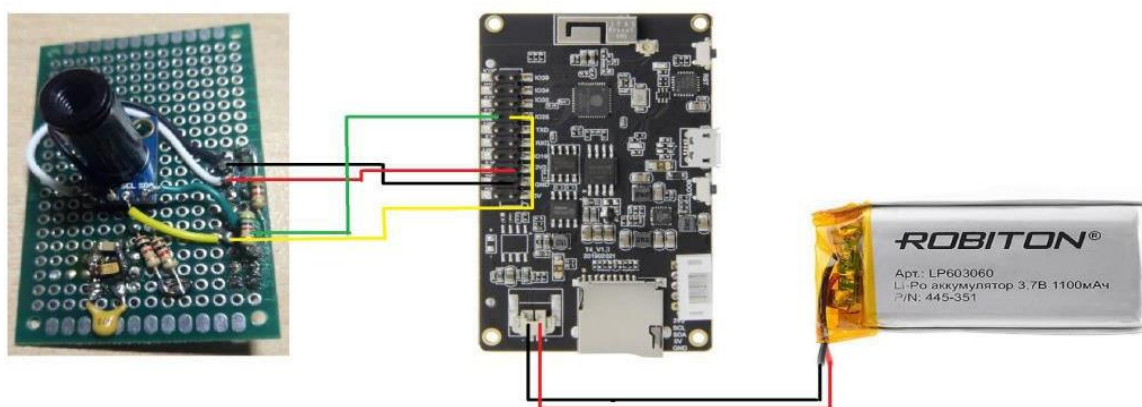


Рисунок 3.4 — Функціональна схема

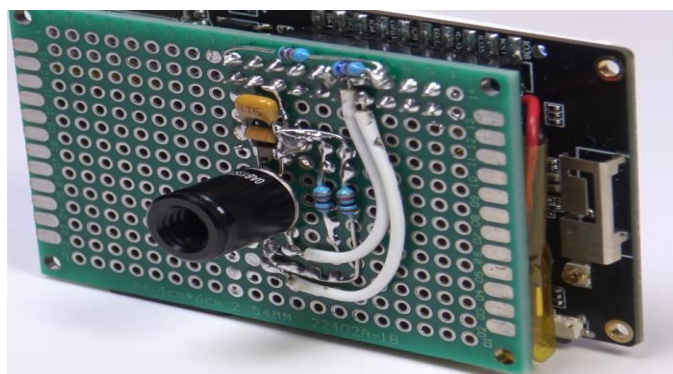


Рисунок 3.5 — Макетна плата

3.3 Програмування

Для ESP32 компанія Espressif розробила фреймворк під назвою IoT Development Framework, відомий як "ESP-IDF". Проект містить дуже хорошу документацію і має приклади, які можна брати за базу. Встановлення та настроювання середовища оточення добре описано в попередньому розділі. Є кілька варіантів встановлення та роботи з фреймворком.

Клонування проекту з репозиторію та ручне встановлення утиліт.

```
mkdir -p ~/esp
```

```
cd ~/esp
```

```
git clone --recursive https://github.com/espressif/esp-idf.git
```

Для Windows установка утиліт розробки можлива за допомогою інсталятора або за допомогою скриптів для командного рядка:

```
cd %userprofile%\esp\esp-idf
```

```
install.bat
```

Для PowerShell

```
cd ~/esp/esp-idf
```

```
./install.ps1
```

Для Linux та macOS

```
cd ~/esp/esp-idf
```

```
./install.sh
```

Наступним кроком є налаштування змінних оточень середовища. Якщо установка інструментів розробки була виконана на Windows за допомогою інсталятора, то ярлик на командну консоль додається в меню та робочий стіл, після чого можна відкривати командну оболонку та працювати з проектами. Альтернативний попит запуску командної оболонки для Windows:

```
%userprofile%\esp\esp-idf\export.bat
```

або Windows PowerShell:

```
.$HOME/esp/esp-idf/export.ps1
```

Linux та macOS:

```
.$HOME/esp/esp-idf/export.sh
```

Слід звернути увагу на пробіл між точкою і шляхом до скрипту. Далі рекомендується додати аліас на скрипт налаштування змінних оточення в профіль користувача, якщо робота виконується в системі Linux або macOS. Для цього необхідно скопіювати та вставити наступну команду у профіль своєї оболонки (.profile, .bashrc, .zprofile, і т.д.):

```
alias get_idf='. $HOME/esp/esp-idf/export.sh'
```

Викликаючи команду `get_idf` в консоль експортуються необхідні змінні оточення. У моєму випадку також необхідно було прописати аліас на запуск віртуального оточення `python`.

```
alias esp_va='source
$HOME/.espressif/python_env/idf4.2_py2.7_env/bin/activate'
```

А також додати `esp_va` до наступного аліасу

```
alias get_idf='esp_ve && . $HOME/esp/esp-idf/export.sh'
```

Для створення нового проекту з нуля можна схиляти вихідні дані з `github.com` або скопіювати з каталогу з прикладами `esp-idf/examples/get-started/hello_world/`.

Проект є каталогом з наступною структурою:

```
- myProject/
  - CMakeLists.txt
  - sdkconfig
  - components/ - component1/ - CMakeLists.txt
    - Kconfig
    - src1.c
  - component2/ - CMakeLists.txt
    - Kconfig
    - src1.c
    - include/ - component2.h
  - main/ - CMakeLists.txt
    - src1.c
    - src2.c
  - build/
```


Конфігурація проекту міститься у файлі `sdkconfig` у кореневому каталозі. Щоб змінити установки, необхідно викликати команду `idf.py menuconfig` (або `idf.py.exe menuconfig` у Windows).

В одному проекті зазвичай створюються дві програми - "project app" (основний виконуваний файл, тобто ваша кастомна прошивка) і "bootloader app" (програма початкового завантажувача проекту). "components" — це модульні частини автономного коду, які компілюються в статичні бібліотеки (файли `.a`) і пов'язані з програмою. Деякі з них надаються самій ESP-IDF, інші можуть бути отримані з інших джерел.

Утиліта командного рядка `idf.py` надає інтерфейс для простого керування збірками проекту. Її розташування в Windows — `%userprofile%\espressif\tools\idf-exe\1.0.1\idf.py.exe`. Вона керує такими інструментами:

`CMake` — налаштовує проект для збирання

Консольний збирач проекту: Ninja, або GNU Make)

`esptool.py` — для прошивки модулів.

Кожен проект має один файл `CMakeLists.txt` верхнього рівня, який містить параметри збирання для всього проекту. Мінімальна конфігурація файлу включає такі необхідні рядки:

```
cmake_minimum_required(VERSION 3.5)
```

```
include($ENV{IDF_PATH}/tools/cmake/project.cmake)
```

```
project(myProject)
```

Проект ESP-IDF можна як сукупність компонентів, у якому каталог `main` є головним компонентом, що запускає код. Тому в цій директорії також міститься файл `CMakeLists.txt`. Найчастіше його структура подібна:

```
idf_component_register(SRCS "main.c" INCLUDE_DIRS ".")
```

У ній вказується, що вихідний файл `main.c` необхідно зареєструвати для компонента, а файли заголовків містяться у поточному каталозі. При необхідності можна перейменувати каталог `main`, встановивши `EXTRA_COMPONENT_DIRS` у прокті `CMakeLists.txt`.

Крім цього, в каталозі знаходиться вихідний `main.c` (ім'я може бути будь-яке) файл з точкою входу – функцією `void app_main(void)`.

Кастомні компоненти створюються в каталозі `components`.

Підключення модуля ESP32 до комп'ютера в більшості випадків проводиться за допомогою USB-кабеля подібно до плат Arduino за рахунок наявного бутлоадера. Єдине, що необхідно — наявність драйвера конвертора USB to UART в системі. Після встановлення драйвера, необхідно визначити номер COM-порту в системі для завантаження скомпільованої прошивки в модуль.

У більшості випадків підходять стандартні налаштування. Але для виклику консольного інтерфейсу меню необхідно перейти до каталогу проекту та в командному рядку набрати:

```
idf.py menuconfig
```

```
(Top)
Espressif IoT Development Framework Configuration
SDK tool configuration --->
Build type --->
Application manager --->
Bootloader config --->
Security features --->
Serial flasher config --->
Partition Table --->
Compiler options --->
Component config --->
Compatibility options --->

[Space/Enter] Toggle/enter  [ESC] Leave menu          [S] Save
[O] Load                    [?] Symbol info          [/] Jump to symbol
[F] Toggle show-help mode  [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

Рисунок 3.6 — Меню з конфігураційними налаштуваннями

Після виклику цієї команди файл `sdkconfig` буде створений, якщо його раніше не було або він заново налаштований. У ранніх навчальних матеріалах можна зустріти команди `make menuconfig`, які є застарілими.

Додавання кастомних налаштувань у файл `sdkconfig` можливе вручну, наприклад:

```
#
# WiFi Settings кастомне меню
#
```

```
CONFIG_ESP_HOST_NAME="Ім'я точки"
CONFIG_ESP_WIFI_SSID="Назва точки доступу"
CONFIG_ESP_WIFI_PASSWORD="Пароль"
```

Але кращим є спосіб за допомогою додаткового файлу конфігурації `Kconfig.projbuild`, який необхідно розташовувати в каталозі з компонентом. Вміст файлу може бути наступним:

```
# put here your custom config value
menu "Example Configuration"
config ESP_WIFI_SSID
    string "Keenetic"
    default "myssid"
    help
        SSID (network name) for the example to connect to.
config ESP_WIFI_PASSWORD
    string "password"
    default "mypassword"
    help
        WiFi password (WPA or WPA2) for the example to use.
endmenu
```

Після виклику команди `idf.py menuconfig` у файлі `sdkconfig` додати додатковий розділ автоматично. Виклик команди `idf.py menuconfig` можливий і в проєкті PlatformIO, однак, потрібно враховувати факт відмінності структури проєкту PlatformIO від класичного ESP-IDF, через що файл `sdkconfig` може знову згенеруватися і втратити кастомні налаштування. Тут можливі вищезгадані варіанти: редагування файлу руками, тимчасове перейменування каталогу `src` в `main` або налаштування файлу `CMakeLists.txt`

Компіляція та завантаження проєкту.

Для білду проєкту необхідно набрати команду `idf.py build`.

Ця команда скомпілює програму та всі компоненти ESP-IDF, а потім згенерує завантажувач, таблицю розділів та двійкові файли програми.

```
$ idf.py build
```

```
Running cmake in directory /path/to/hello_world/build
```

```
Executing "cmake -G Ninja --warn-uninitialized /path/to/hello_world"...
```

```
Warn about uninitialized values.
```

```
-- Found Git: /usr/bin/git (found version "2.17.0")
```

```
-- Building empty aws_iot component due to configuration
```

```
-- Component names: ...
```

```
-- Component paths: ...
```

```
... (more lines of build system output)
```

```
[527/527] Generating hello-world.bin
```

```
esptool.py v2.3.1
```

```
Project build complete. To flash, run this command:
```

```
../../../../components/esptool_py/esptool/esptool.py -p (PORT) -b 921600 write_flash
--flash_mode dio --flash_size detect --flash_freq 40m 0x10000 build/hello-world.bin
build 0x1000 build/bootloader/bootloader.bin 0x8000 build/partition_table/partition-
table.bin
```

```
or run 'idf.py -p PORT flash'
```

Слід враховувати, що початковий процес компіляції навіть простого проекту займає час, оскільки, на відміну Arduino фреймворка компілюються багато додаткових модулів ESP-IDF. Подальша зміна вихідних джерел призводить лише до компіляції цих файлів. Виняток становить зміну конфігурації. Для завантаження скомпілованих двійкових файлів (bootloader.bin, partition-table.bin та hello-world.bin) на плату ESP32 необхідно запустити команду:

```
idf.py -p PORT [-b BAUD] flash,
```

де PORT замінюємо на те, що нам потрібно (COM1, /dev/ttyUSB1), а також опціонально можемо змінити швидкість завантаження, вказавши необхідне значення для BAUD.

Для відстеження завантаженої програми можна використовувати будь-яку утиліту моніторингу com-порту, такі як HTerm, CoolTerm, або використовувати утиліту моніторингу IDF Monitor, для запуску якої необхідно ввести команду:

```
idf.py -p PORT monitor
```

Розробку програмного забезпечення проведено на офіційному плагіні для Visual Studio Code.

Встановлення та налаштування ESP-IDF фреймворку в цьому розширенні представлено як меню onboarding, про що також йдеться в описі. Завантаження та встановлення всіх компонентів відбувається автоматично у процесі проходження етапів меню. Можна привести всі скріни процесу, але вони інтуїтивно зрозумілі і практично не вимагають пояснення. ESP-IDF плагін керується за допомогою меню команд, яке можна викликати за допомогою клавіші F1 або комбінації клавіш, описаних у мануалі.

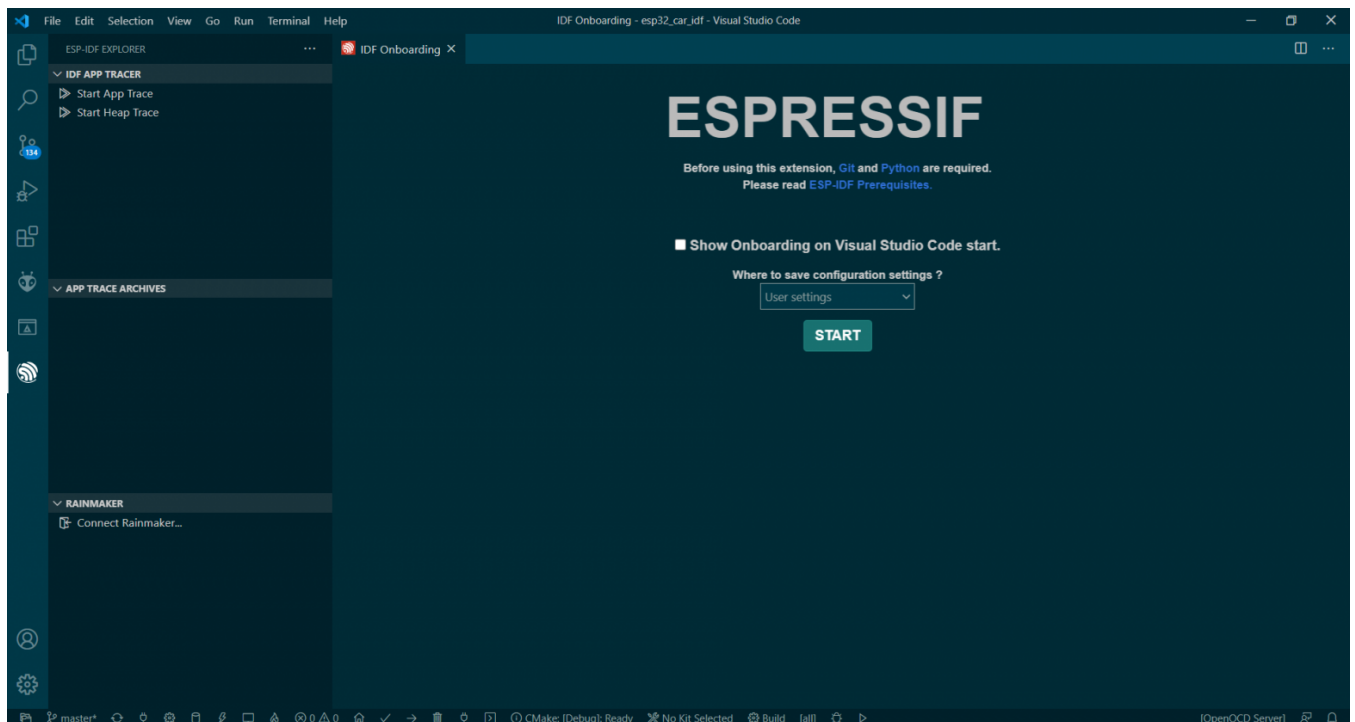


Рисунок 3.7 — Початкове налаштування плагіна

Перевага використання плагіна полягає в тому, що дотримується класична структура проекту, немає необхідності якось «шаманити» з налаштуваннями. Є один нюанс: якщо хочемо відкрити раніше створений проект у Visual studio code з ESP-IDF плагіном, то нам лише необхідно скопіювати в корінь з проектом каталог .vscode, який можна отримати, згенерувавши хоча б один раз шаблонний проект за допомогою ESP- IDF плагіна.

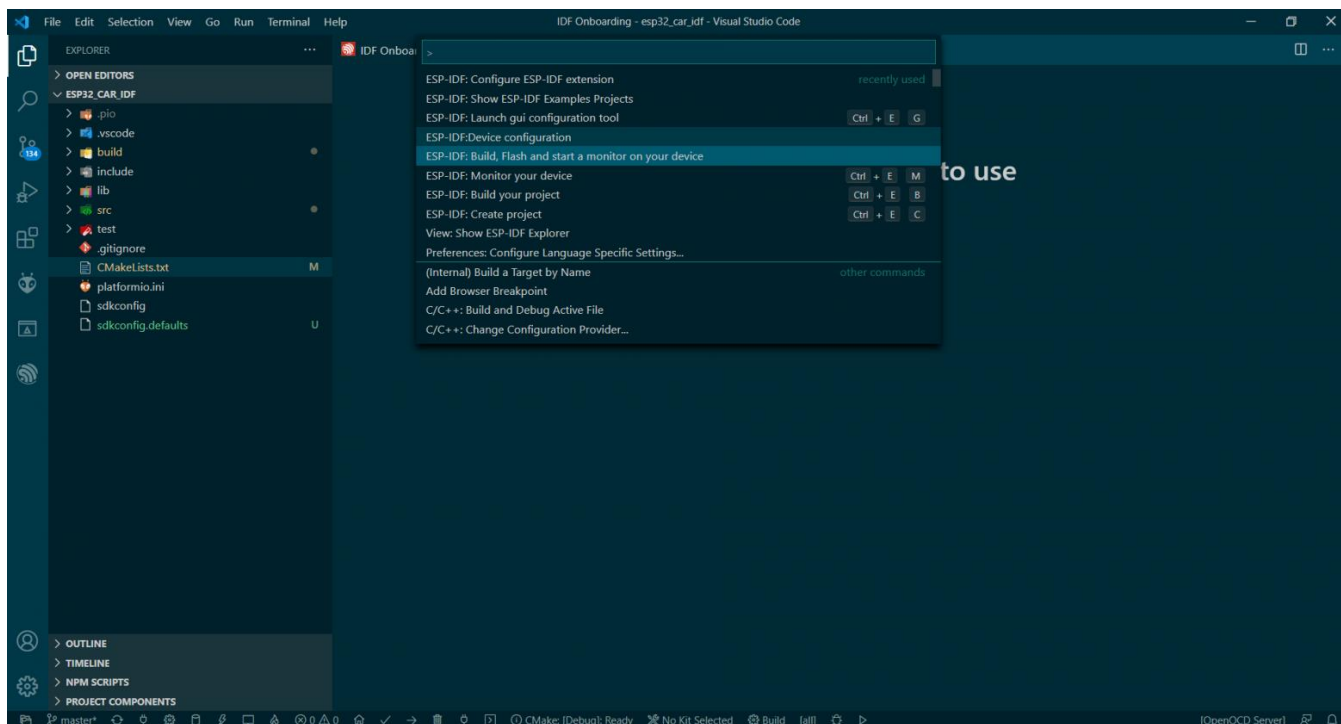


Рисунок 3.8 — Меню команд

3.4 Алгоритм роботи

При підключеному акумуляторі до плати TTGO T4 V1.3 одноразово натискаємо кнопку BOOT, запускається мікросхема управління живленням ініціалізація IP5306 (управління розряду та заряду акумуляторної батареї), на вихід мікросхема видає 5В і вся плата починає працювати. Далі в процесі роботи плати можна робити одноразове натискання на кнопку RST, при цьому відбувається скидання мікроконтролера ESP32, так як ця кнопка заведена не тільки на мікросхему IP5306, але і на ніжку reset мікроконтролера ESP32.

Якщо при увімкненому пристрої натиснути двічі на кнопку RST, мікроконтролер IP5306 повністю знеструмлює плату і вимикає пристрій.

При одноразовому натисканні кнопки RST на екрані пристрою виводиться консоль ініціалізації (див. рис. 3.9). У консолі йде відображення службових повідомлень, а також сервісів, які обслуговують цю консоль. Вихідний код реалізації консолі наведено у додатку Б.



Рисунок 3.9 — Ініціалізація консолі

Далі йде завантаження графічної бібліотеки. Вона дозволяє виводити форматований текст із заданим шрифтом та кольором на дисплей, а також малювати примітивні геометричні фігури (див. рис. 3.10). Для оновлення дисплея використовується буфер кадру.



Рисунок 3.10 — Відображення графічної бібліотеки

Далі відбувається завантаження бібліотеки термосенсора MLX90640. У вихідному коді можна буде використовувати цю бібліотеку, викликаючи функції:

MLX90640_DumpEE — після ініціалізації читаємо вміст EPROM'а сенсора. MLX90640_ExtractParameters — дістаємо з нього параметри та отримані дані далі беруть участь у розрахунку.

MLX90640_GetFrameData — використовується для отримання із сенсора чергового кадру і лічені значення передаються на розрахунок будь-якої функції на вибір.

MLX90640_GetImage — лічені значення передаються в цю функцію для спрощеного розрахунку, якщо потрібна термо-картинка, але не важливі абсолютні значення температури для кожного пікселя, тим самим можна заощадити ресурси платформи.

MLX90640_CalculateTo — функція веде розрахунок масиву з речовими значеннями, де кожне речове значення відповідає температурі в кожному пікселі.

Вихідний код потоку обслуговуючого сенсор наведено у додатку В.

При одноразовому натисканні кнопки IO37 здійснюється вхід у меню налаштування системи (див. рис. 2.11).

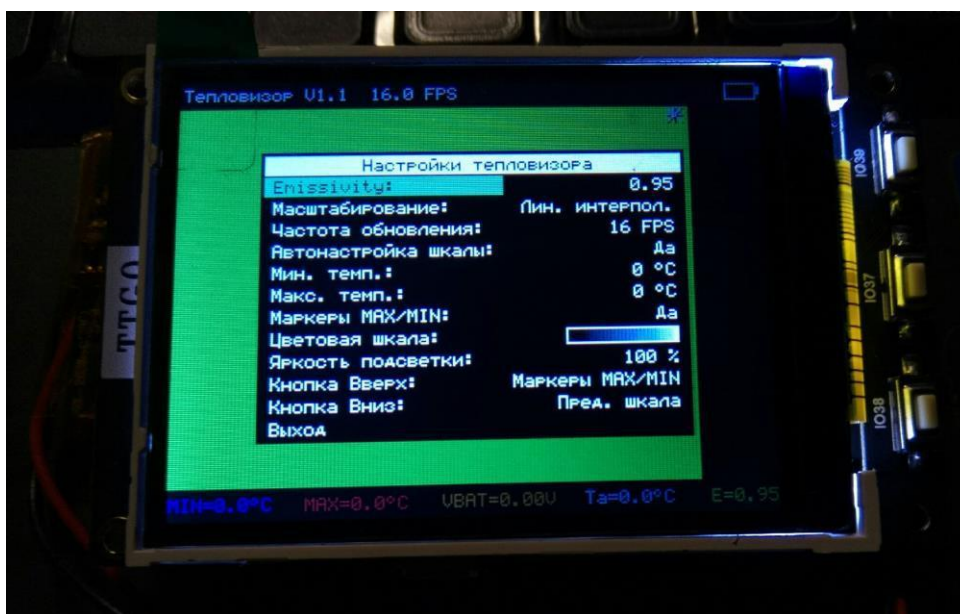


Рисунок 2.11 — Меню налаштування системи

Усі вибрані настройки зберігаються в незалежній пам'яті ESP32. Кнопка IO37 служить для виклику однорівневого меню та підтвердження дій. Кнопка IO39 та

Ю38 з основного екрана керує яскравістю підсвічування дисплея, проте функції цих двох кнопок можна перевизначати у параметрах меню.

Через меню системи можна змінити такі параметри:

Коефіцієнт відображає здатність поверхні (Emissivity). Залежно від вимірюваної поверхні цей коефіцієнт необхідно коригувати. Чим більше поверхня відбиває світло, тим нижче цей коефіцієнт має бути для коректного виміру.

Спосіб масштабування можна вибрати за найближчим — збільшення в дев'ять разів кожного пікселя, що є вихідною термограмою, отриманою з термосенсора, лінійне масштабування, а також алгоритм HQ3X (2X), який проганяється двічі.

Частота оновлення термограми — цей параметр відправляється в сенсор, далі сенсор вже видає зі зміненою частотою термограми, що відображаються на екрані.

Автоналаштування шкали — використовується для увімкнення/вимкнення автоналаштування шкали у правій частині екрана. Діапазон шкали автоматично підлаштовується під поточні об'єкти у кадрі. Якщо вимкнути цей параметр, то з'являється можливість встановити вручну мінімальну і максимальну межу температури для колірної шкали. Останні значення, які відображаються за замовчуванням, — це значення, які використовувалися при автоматичному визначенні шкали.

Увімкнення/вимкнення маркера максимальної та мінімальної температури у кадрі. На екрані пристрою за допомогою піктограми зірки відображається максимальна температура та мінімальна температура за допомогою хрестика.

Вибір колірної шкали. Колірна шкала «Iron» — складається з плавного переходу через кілька кольорів і відповідає кольорам нагрітого металу. Колірна шкала «Райдуга» яка складається з 7 кольорів у більш плавному та контрастних варіантах. Колірна шкала від синього до червоного, а також від чорного до білого.

Яскравість підсвічування, яке можна задати з кроком 5%.

Призначення на кнопки Ю39 і Ю38 таких функцій як: зміна яскравості екрана, збереження на sd-карту даних у форматі CSV або BMP, увімкнення/вимкнення маркерів, перебір колірних шкал, зміна коефіцієнта відображення здатності до поверхні.

ВИСНОВКИ

У процесі виконання бакалаврського проекту вивчили основну концепцію робототехніки, її історію, вплив на сучасний світ та тенденції розвитку на найближче майбутнє.

У бакалаврській роботі розроблено мікропроцесорна система безконтактного термометра для безконтактного визначення температури поверхні об'єкта. Система може визначати температуру будь-якого тіла або матеріалу. За допомогою даного комплексу, без особливих зусиль, можна знайти витік температури або перегріті елементи. Отриманий пристрій має широку сферу застосування і досить універсальний.

В результаті виконаної роботи було вирішено такі:

- проаналізовано архітектуру та влаштування плати TTGO;
- проаналізовано специфіку інфрачервоних датчиків температури;
- проаналізовано специфіки роботи з мікроконтролером ESP32;
- розроблено структурну та принципову схему пристрою;
- спроектовано мікропроцесорну систему безконтактного термометра;.

Написаний програмний код для виведення зображення на екран та подальшої роботи з ним.

Перелік джерел посилання

1. Тепловидение для инженеров: учебное пособие. Вавилов В.П.; Издательство Томского политехнического университета, — г. Томск: 2012г. — 129 с.
2. Инфракрасная термография и тепловой контроль (издание: 2-е). Вавилов В.П.; Издательский дом «Спектр» – г. Москва: 2013г. – 544 с.
3. Современные датчики: справочник. Дж. Фрайден; Книжное издательство «Техносфера» - г. Москва: 2005г. – 592 с.
4. Инфракрасная техника и электронная оптика (издание: 3-е). Пономаренко В.П., Филачев А.М.; Издательство «Физматкнига» - г. Москва: 2013г. – 384 с.
5. Интернет вещей с ESP8266. Марко Шварц; Издательство «BHV-Peterburg» - г. Санкт-Петербург: 2018г. – 192 с.
6. Искусство схемотехники (издание 7-е). Хоровиц Пауль, Хилл Уинфилд; Издательство «Бином» - г. Москва: 2016г. – 704 с.
7. Что нужно знать цифровому разработчику об аналоговой электронике. Бонни Бэйкер; Издательство «Додека-XXI» - г. Москва: 2010г. – 360 с.
8. Справочник по основам инфракрасной техники. Крискунов Л.З.; Издательство «Советское радио» - г. Москва: 1978г. – 400 с.
9. Датчики в современных измерениях. Котюк А.Ф.; Издательство «Горячая линия-Телеком» - г. Москва: 2006г. – 96 с.
10. Интернет вещей. Будущее уже здесь. Грингард Сэмюэл; Издательство «Альпина Паблицер» - г. Москва: 2019г. – 188 с.
11. Тепловизор: кому полезен и где применяется [Электронный ресурс]. – Режим доступа: <https://www.pergam.ru/articles/teplovizor.htm> - Дата доступа: 01.05.2022.
12. Раскрываем тайны бесконтактных датчиков температуры [Электронный ресурс]. – Режим доступа: <https://barnaul.terraelectronica.ru/news/5502> - Дата доступа: 02.05.2022.

13. NUCLEO-L053R8 [Электронный ресурс] Режим доступа:
<https://www.st.com/en/evaluation-tools/nucleo-l053r8.html#overview> – Загл. з экрану.
14. Teensy++ 2.0 [Электронный ресурс]. Режим доступа:
<https://radioprogram.ru/shop/merch/14> - Загл. з экрану.
15. NodeMcu v3 [Электронный ресурс]. Режим доступа:
<https://arduinomaster.ru/platy-arduino/esp8266-nodemcu-v3-lua/> - Загл. з экрану.
16. TFT01-22SP [Электронный ресурс]. Режим доступа:
<http://wiki.amperka.ru/продукти:tft-lcd-240x320> - Загл. з экрану.
17. 3641BS [Электронный ресурс]. Режим доступа:
http://codius.ru/articles/Arduino_UNO_4розрядний_7сегментний_індикатор_12_pin_3641BS_red - Загл. з экрану.
18. SDSHield Arduino [Электронный ресурс]. Режим доступа:
<http://freeduino.ru/arduino/SD-shield.html> - Загл. з экрану.

ДОДАТОК А

Міністерство освіти та науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ ВНТУ

д.т.н., проф.

_____ О. Д. Азаров

“__” _____ 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання бакалаврської дипломної роботи
«Мікропроцесорна система безконтактного термометра»
08-23.БДП.034.00.000 ТЗ

Науковий керівник к.т.н., доц. каф. ОТ

_____ Кадук О.В.

Студент групи ІКІ-186

_____ Кізіма А.С.

Вінниця 2022

1. Найменування та область застосування

Мікропроцесорна система безконтактного термометра.

2. Основи для розробки

Необхідність реалізації мікропроцесорних систем безконтактних термометрів для широкого діапазону температур.

3. Мета та призначення розробки

Теоретичне та практичне дослідження застосування портативного мікропроцесорного програмно-апаратного комплексу для безконтактного визначення температури поверхні об'єктів.

4. Етапи БДП та очікувані результати

Робота виконується в п'ять етапів, що наведені в таблиці 4.1.

Таблиця 4.1 – Етапи виконання роботи

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз завдання. Вступ	14.02.20	14.02.20	Вступ
2	Огляд і аналіз принципу роботи безконтактних термометрів	15.02.20	28.02.20	Розділ 1
3	Вибір мікропроцесорної платформи	01.03.20	14.03.20	Розділ 2
4	Проектування мікропроцесорної системи безконтактного термометра	15.03.20	28.4.20	Розділ 3
5	Оформлення пояснювальної записки	29.04.20	17.05.20	ПЗ, презентація

5. Матеріали, що подаються до захисту БДП

Пояснювальна записка БДП, графічні і ілюстративні матеріали, протокол попереднього захисту БДП на кафедрі, відзив наукового керівника, рецензія опонента, протоколи складання державних екзаменів, анотації до БДП

українською та іноземною мовами, довідка про відповідність оформлення БДП діючим вимогам.

6. Порядок контролю виконання та захисту БДП

Виконання етапів графічної та розрахункової документації БДП контролюється науковим керівником згідно зі встановленими термінами. Захист БДП відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

7. Вимоги до оформлення БДП

Вимоги викладені в МЕТОДИЧНИХ ВКАЗІВКАХ до дипломного проектування, ДСТУ 3008-95, ДСТУ 3974-2000 «Правила виконання дослідно-конструкторських робіт. Загальні положення» та діючого ГОСТ 2.114-95 ЄСКД.

8. Вимоги щодо технічного захисту інформації в БДП з обмеженим доступом

Відсутні.

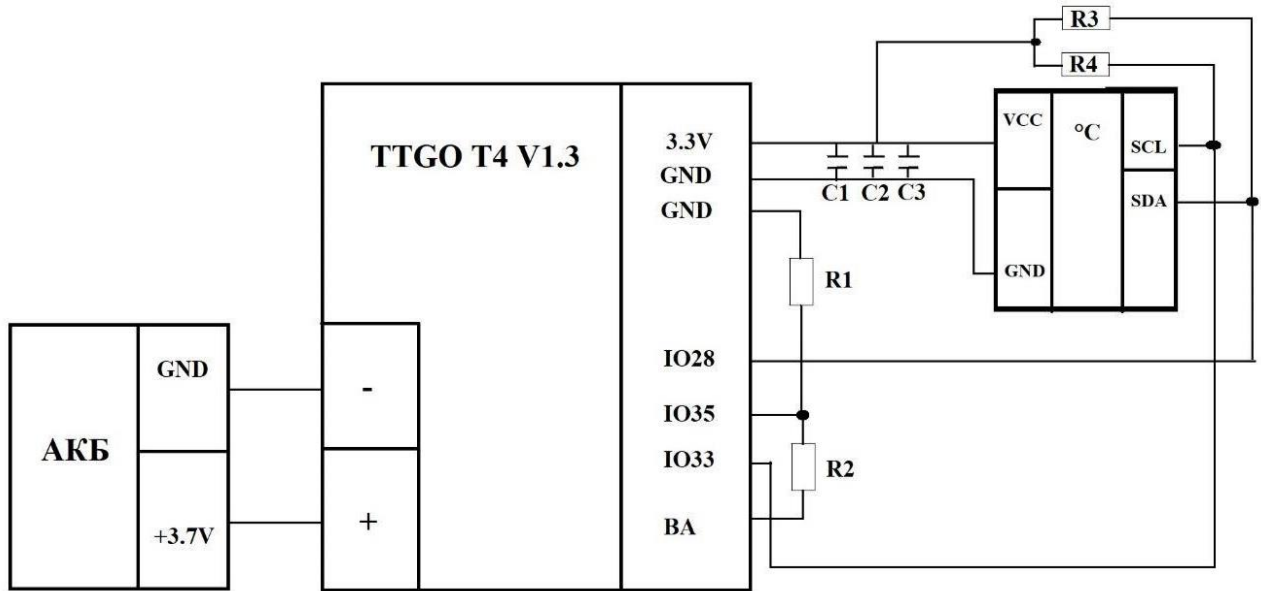
Технічне завдання до виконання отримав _____ Кізіма А.С.

Структурна схема системи



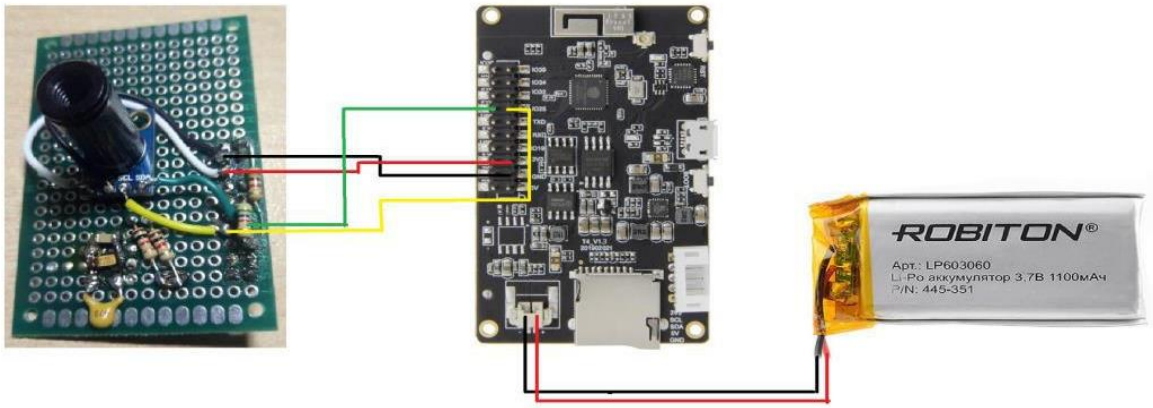
ДОДАТОК В

Схема електрична принципова системи

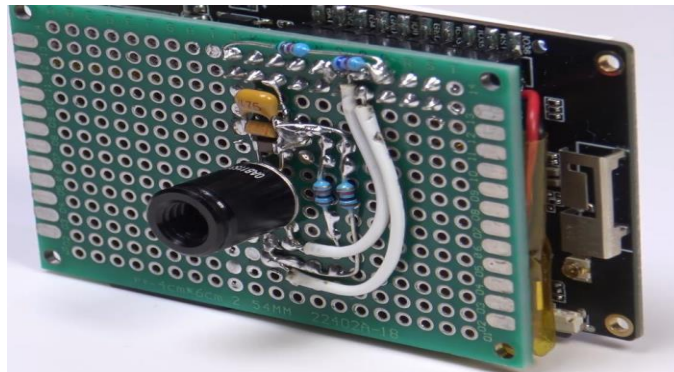


ДОДАТОК Г

Функціональна схема



Макетна плата



ДОДАТОК Д

Код реалізації консолі

```

// Процедура робить паузу виведенні
консолі void console_pause(uint32_t
timeMs)
{
    dispcolor_Update();
    vTaskDelay(timeMs/portTICK_RATE_MS);
}
// Процедура додає новий рядок консоль
void console_printf(eConsoleMsgType msgType, const char*args, ...)
{
    char StrBuff[256];
    uint16_t TextColor = WHITE;
    switch(msgType)
    {
        case MsgInfo:
            TextColor = GREEN;
            break;
        case MsgWarning:
            TextColor = RGB565 (249, 166, 2);
            break;
        case MsgError:
            TextColor = RED;
            break;
    }
    va_list ap;
    va_start(ap, args);
    vsprintf(StrBuff, sizeof(StrBuff), args, ap);
    va_end(ap);
    dispcolor_DrawString(0, ConsoleLine * LineHeight, FONTID_6X8M, (uint8_t*)
StrBuff, TextColor);
    printf("%s", StrBuff);
    if(msgType != MsgInfo)
        console_pause(500);
    if(++ConsoleLine > 240 / LineHeight)
    {
        ConsoleLine = 0;
        console_pause(300);
        dispcolor_ClearScreen();
    }
    dispcolor_Update();
}

```

```
// Процедура виводить повідомлення про перезавантаження в консоль та
перезавантажує esp32
void FatalError()
{
    console_printf(MsgError, "Перезавантаження через 5
    секунд...\r\n"); vTaskDelay(5000/portTICK_RATE_MS);
    fflush(stdout);
    esp_restart();
}
// Процедура виводить повідомлення про помилку і про перезавантаження в консоль
та перезавантажує esp32
void FatalErrorMsg(const char*args, ...)
{
    char StrBuff[256];
    va_list ap;
    va_start(ap, args);
    vsnprintf(StrBuff, sizeof(StrBuff), args, ap);
    va_end(ap);
    console_printf(MsgError, "%s\r\n", StrBuff);
    FatalError(); }
}
```

ДОДАТОК Е

Код опрацювання потоку сенсора

```

// Процедура копіює матрицю температури в буфер
pBuff void GetThermoData(float* pBuff)
{
    sMlxData* pMlxData = &MlxData[(MlxDataIdx + 1) & 1];
    float* ThermoImage = pMlxData->ThermoImage;
    memcpy(pBuff, ThermoImage, (termWidth * termHeight) << 2);
}
void mlx_task(void* arg)
{
    // Читання параметрів mlx90640 з EEPROM
    int result = MLX90640_DumpEE(0x33, Frame);
    if(result < 0)
    {
        MlxError = 1;
        vTaskDelete(NULL);
        return;
    }
    result = MLX90640_ExtractParameters(Frame, ¶ms);
    if(result < 0)
    {
        MlxError = 1;
        vTaskDelete(NULL);
        return;
    }
    // Встановлення частоти оновлення
    // термограми
    result =
    MLX90640_SetRefreshRate(0x33,
    FPS_Idx); if(result < 0)
    {
        MlxError =
        1; vTaskDelete(NU
        LL);
        return;
    }
    FPS_Idx_Old =
    FPS_Idx; while(1)
    {
        sMlxData* pMlxData = &MlxData[MlxDataIdx];
        float* ThermoImage = pMlxData->ThermoImage;

```

```

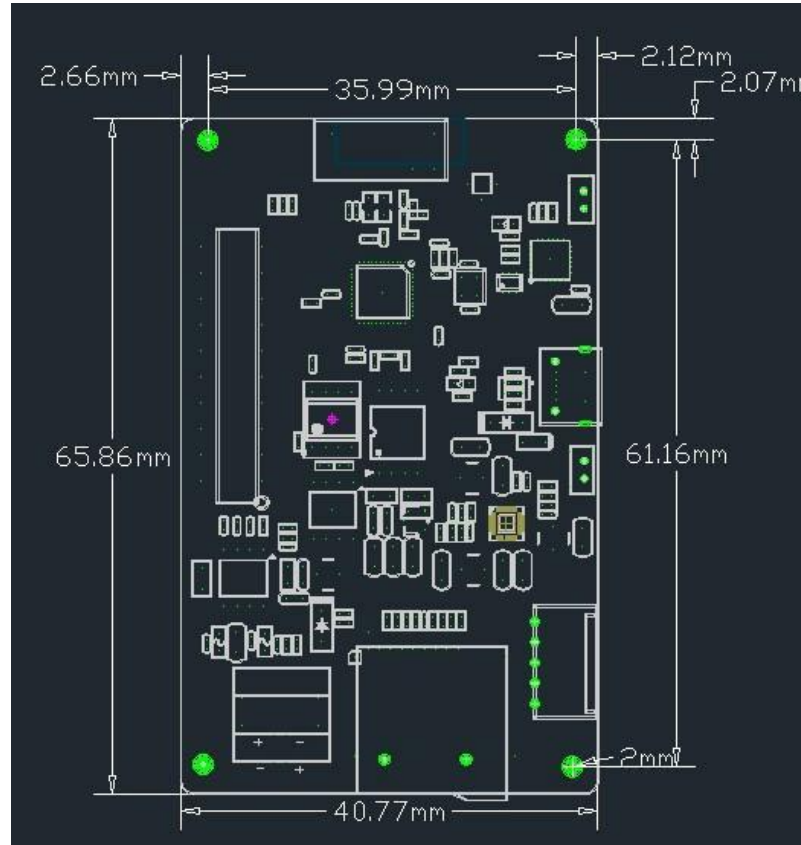
    pMlxData->VBAT= (((float) getBatteryVoltage ()) /
    1000;if(FPS_Idx != FPS_Idx_Old)
    {
// Встановлення частоти оновлення термограми
        result = MLX90640_SetRefreshRate(0x33,
        FPS_Idx);if(!result)
            FPS_Idx_Old = FPS_Idx;
    }
// Читання2 субсторінок даних із сенсора
    for(uint8_t subPage = 0; subPage <2; subPage++)
    {
// Читання кадру із сенсора
        result = MLX90640_GetFrameData(0x33, Frame);
// Вважаємо і виводимо кілька параметрів, зчитаних з mlx90640
        pMlxData->Vdd= MLX90640_GetVdd(Frame, ¶ms);
        pMlxData->Ta= MLX90640_GetTa(Frame, ¶ms);
//Розрахунок матриці
        температурfloattr = pMlxData->Ta -
8;
        MLX90640_CalculateTo(Frame, ¶ms, Emissivity, tr, pMlxData->ThermoImage);
    }
//Вважаємо температуруу центрі
екранурpMlxData->CenterTemp=
        ThermoImage[termWidth * ((termHeight >> 1) - 1) + ((termWidth >> 1) -
1)]
+
        ThermoImage[termWidth * ((termHeight >> 1) - 1) + (termWidth >> 1)]
+ ThermoImage[termWidth * (termHeight >> 1) + ((termWidth >> 1) -
1)] +
        ThermoImage[termWidth*(termHeight >> 1) + (termWidth >>
1)]; pMlxData->CenterTemp/= 4;
// Пошук мінімальноїта максимальної температури у
кадріpMlxData->minT= 300;
pMlxData->maxT= -40;
    for(uint8_t y = 0; y < termHeight; y++)
    {
        for(uint8_t x = 0; x < termWidth; x++)
        {
            floattemp = ThermoImage[y*termWidth + x];
            if(pMlxData->maxT < temp)
            {
                pMlxData->maxT= temp;
                pMlxData->maxT_X= x;
                pMlxData->maxT_Y= y;
            }
            if(pMlxData->minT > temp)
            {
                pMlxData->minT= temp;
            }
        }
    }

```

```
                pMlxData->minT_X= x;
                pMlxData->minT_Y= y;
            }
        }
    }
    if(pMlxData->maxT > MAX_TEMP)
        pMlxData->maxT= MAX_TEMP;
    if(pMlxData->minT <MIN_TEMP)
        pMlxData->minT= MIN_TEMP;
    MlxDataIdx = (MlxDataIdx + 1) & 1;
}
}
```

ДОДАТОК Є

Габаритні розміри та розміщення елементів платформи TTGO T4 V1.3



ДОДАТОК Є

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи Мікропроцесорна система безконтактного термометра

Тип роботи: бакалаврська дипломна робота
(БДР, МКР)

Підрозділ кафедра обчислювальної техніки
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 91,7% Схожість 8,3%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Захарченко С.М.
(підпис) (прізвище, ініціал)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ Кізіма А.С.
(підпис) (прізвище, ініціал)

Керівник роботи _____ Кадук О.В.
(підпис) (прізвище, ініціал)