

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

В. о. завідувача кафедри

Михайло НОВОТАРСЬКИЙ

(підпис)

“ ” _____ 2025 р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою “Комп’ютерні системи та мережі”
спеціальності 123 “Комп’ютерна інженерія”**

на тему: Програмно-апаратний комплекс для інтеграції користувацьких пристроїв у систему розумного будинку HomeAssistant.

Виконав : студент 4 курсу, групи ІО-11
(шифр групи)

Куделя Олександр Сергійович

(прізвище, ім’я, по батькові)

(підпис)

Керівник ст. викл. Васильєва М. Д.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант (нормоконтроль) асистент кафедри ОТ Пономаренко А. М.

(назва розділу)

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент доцент каф.ІІІ, к.т.н Вечерковська А.С.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

(підпис)

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

“Комп’ютерні системи та мережі”

спеціальності 123 “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ

В. о. завідувача кафедри

Михайло НОВОТАРСЬКИЙ

_____ (підпис)

“ ___ ” _____ 2025 р.

ЗАВДАННЯ

на бакалаврський дипломний проєкт студента

Куделя Олександр Сергійович

1. Тема проєкту Програмно-апаратний комплекс для інтеграції користувацьких пристроїв у систему розумного будинку HomeAssistant.
керівник проєкту Васильєва Марія Давидівна, ст. викл.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від 23 травня 2025 року №1705-с
2. Термін здачі студентом закінченого проєкту 2 червня 2025 р.
3. Вихідні дані до проєкту технічна документація, теоретичні дані.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які розробляються)
Розділ 1. Аналіз існуючих підходів до інтеграції пристроїв у системи розумного будинку.
Розділ 2. Проєктування програмно-апаратного комплексу.
Розділ 3. Реалізація пристроїв та програмного забезпечення.

Розділ 4. Огляд та тестування розробленого програмного засобу.

5. Перелік графічного матеріалу (з точним позначенням обов'язкових креслень).

6. Консультанта проєкту, з вказівкою розділів проєкту, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Пономаренко А. М.		

7. Дата видачі завдання «б» лютого 2025 р.

Календарний план

№ П/П	Найменування етапів дипломного проєкту	Терміни виконання етапів проєкту	Примітки
1.	<i>Затвердження теми проєкту</i>	<i>10.12.2024-25.12.2024</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>06.01.2025-15.03.2025</i>	
3.	<i>Розробка архітектури та загальної структури системи</i>	<i>15.03.2025-05.04.2025</i>	
4.	<i>Розробка структур окремих підсистем</i>	<i>15.03.2025-05.04.2025</i>	
5.	<i>Програмна реалізація системи</i>	<i>05.04.2025-01.05.2025</i>	
6.	<i>Оформлення пояснювальної записки</i>	<i>01.05.2025-15.05.2025</i>	
7.	<i>Захист програмного продукту</i>	<i>03.06.2025</i>	
8.	<i>Передзахист</i>	<i>10.06.2025</i>	
9.	<i>Захист</i>	<i>21.06.2025</i>	

Студент-дипломник _____ Олександр КУДЕЛЯ
(підпис)

Керівник проєкту _____ Марія ВАСИЛЬЄВА
(підпис)

АНОТАЦІЯ

У даній дипломній роботі розроблено програмно-апаратний комплекс для інтеграції користувацьких пристроїв у систему розумного будинку на базі Home Assistant. Основною метою проєкту стало створення сенсорного пристрою-панелі, який забезпечує двосторонню взаємодію з системою автоматизації будинку: збір та передача даних, а також керування підключеними пристроями. Для реалізації проєкту використано ESPHome, сенсорний дисплей із графічним інтерфейсом LVGL та інфраструктуру Zigbee-пристроїв.

Ключові слова: розумний дім, Home Assistant, ESPHome, автоматизація, інтерфейс користувача, Zigbee.

ANNOTATION

In this thesis, was developed a hardware and software system for integrating user devices into a smart home system based on Home Assistant. The main goal of the project was to create a touch panel device that provides two-way interaction with the home automation system: data collection and transmission, as well as control of connected devices. ESPHome, a touchscreen display with an LVGL graphical interface, and Zigbee device infrastructure were used to implement the project.

Keywords: smart home, Home Assistant, ESPHome, automation, user interface, Zigbee.

Довідки	Формат	Значення	Найменування	Кіл. листів	№ екземпляр а	Додаток
			Документація загальна			
			Знову розроблена			
	A4	ІАЛЦ.467200.002 ТЗ	Програмно-апаратний комплекс комплекс для інтеграції користувачьких пристроїв у систему розумного будинку HomeAssistant	3		
			Технічне завдання			
	A4	ІАЛЦ.467200.003 ПЗ	Програмно-апаратний комплекс комплекс для інтеграції користувачьких пристроїв у систему розумного будинку HomeAssistant	71		
			Пояснювальна записка			
	A4	ІАЛЦ.467200.004 ДА	Програмно-апаратний комплекс комплекс для інтеграції користувачьких пристроїв у систему розумного будинку HomeAssistant	1		
			Структурна схема системи			
	A4	ІАЛЦ.467200.004 ДБ	Програмно-апаратний комплекс комплекс для інтеграції користувачьких пристроїв у систему розумного будинку HomeAssistant	1		
			Функціональна схема			
	A4	ІАЛЦ.467200.004 ДВ	Програмно-апаратний комплекс комплекс для інтеграції користувачьких пристроїв у систему розумного будинку HomeAssistant	1		
			Алгоритм дій програми			
	A4	ІАЛЦ.467200.004 ДГ	Програмно-апаратний комплекс комплекс для інтеграції користувачьких пристроїв у систему розумного будинку HomeAssistant	58		
			Лістинг програмного коду			

ІАЛЦ.467200.001 ОА

№ докум.	Підпис	Дата				
Розробив	Куделя О.С.		Програмно-апаратний комплекс для інтеграції користувачьких пристроїв у систему розумного будинку HomeAssistant	Літ.	Аркуш	Аркушів
Перевірив	Васильєва М. Д.				1	1
Н. Контр.	Пономаренко А. М.			НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІО-11		
Затвердив						
			Опис Альбому			

ТЕХНІЧНЕ ЗАВДАННЯ
ДО ДИПЛОМНОГО ПРОЄКТУ

на тему: *«Програмно-апаратний комплекс для інтеграції користувацьких пристроїв у систему розумного будинку HomeAssistant»»*

Київ – 2025

ЗМІСТ

НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ	2
ПІДСТАВИ ДЛЯ РОЗРОБКИ.....	2
МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ.....	2
ДЖЕРЕЛА РОЗРОБКИ.....	3
ТЕХНІЧНІ ВИМОГИ.....	3
Вимоги до розробленого продукту.....	3
Вимоги до програмного забезпечення	3
Вимоги до апаратної частини	3
ЕТАПИ РОЗРОБКИ	4

					ІАЛЦ.467200.002 ТЗ			
		№ докум.	Підпис	Дата				
Розробив	Куделя О.С.				Програмно-апаратний комплекс для інтеграції користувацьких пристроїв у систему розумного будинку HomeAssistant Технічне завдання	Літ.	Аркуш	Аркушів
Перевірив	Васильєва М. Д.						1	4
Н. Контр.	Пономаренко А. М.					НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, Ю-11		
Затвердив								

1 НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Дане технічне завдання поширюється на розробку програмно-апаратного комплексу для моніторингу, керування та забезпечення безпеки у системі розумного будинку з інтеграцією в платформу Home Assistant. Областю застосування даного пристрою є приватні домогосподарства, що використовують альтернативні джерела живлення (сонячні панелі, інвертори, акумулятори), а також старе газове обладнання, що потребує додаткових заходів безпеки. Комплекс може бути використаний як у побутових, так і в напівпромислових умовах для підвищення рівня автоматизації, безпеки та енергоефективності житлових приміщень, особливо актуально в умовах частих відключень електроенергії.

2 ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання для виконання бакалаврського проєкту по освітньо-професійній програмі “Комп’ютерні системи та мережі” спеціальності 123 “Комп’ютерна інженерія”, затверджене кафедрою Обчислювальної техніки Національного технічного Університету України “Київський Політехнічний інститут імені Ігоря Сікорського”.

3 МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою та призначенням даної роботи є створення програмно-апаратного комплексу, інтегрованого з Home Assistant, що дозволяє здійснювати моніторинг параметрів домашньої енергосистеми, контролювати розумні пристрої, відображати метеодані, а також реалізовувати критично важливі функції безпеки. Розробка спрямована на підвищення безпеки, енергоефективності та зручності використання розумного будинку, особливо в умовах енергетичної нестабільності.

					ІАЛЦ.467200.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		2

4 ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки даного дипломного проекту є офіційні документації, публікації та статті в мережі Інтернет на дану тему, науково-технічна література.

5 ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного коду

Розроблена програма має виконувати такі вимоги:

- Код має бути оформлений відповідно до стандартів програмування.
- Реалізована підтримка безпечної взаємодії пристроїв з HomeAssistant.
- Використання відкритих бібліотек для інтеграції ESP32 із HomeAssistant.
- Забезпечення надійності та стабільності роботи пристроїв у автономному режимі.

5.2. Вимоги до програмного забезпечення

- ОС Windows, Mac чи Linux.
- Lvgl версії 8.0
- EspHome версії 2025.3.

5.3. Вимоги до апаратної частини

- ESP32 або аналоги.
- Wi-Fi модуль.
- Flash пам'ять не менше ніж 8 Мб.

					ІАЛЦ.467200.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

6 ЕТАПИ РОЗРОБКИ

Назва етапів виконання	Термін виконання
Затвердження теми роботи	10.12.2024-25.12.2024
Вивчення та аналіз завдання	06.01.2025-15.03.2025
Розробка архітектури та загальної структури системи	15.03.2025-05.04.2025
Розробка структур окремих частин системи	15.03.2025-05.04.2025
Програмна реалізація системи	05.04.2025-01.05.2025
Виправлення помилок	01.05.2025-15.05.2025
Оформлення пояснювальної записки	01.05.2025

					ІАЛЦ.467200.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОГО ПРОЄКТУ**

на тему: *«Програмно-апаратний комплекс для інтеграції користувацьких пристроїв у систему розумного будинку HomeAssistant»*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО ІНТЕГРАЦІЇ ПРИСТРОЇВ У СИСТЕМИ РОЗУМНОГО БУДИНКУ.....	6
1.1 Огляд існуючих рішень для розумного будинку (HomeAssistant, OpenHAB)	6
1.1.1 HomeAssistant.....	6
1.1.2 OpenHAB.....	10
1.2 Аналіз можливостей інтеграції користувацьких пристроїв у HomeAssistant	13
1.3 Порівняння переваг використання власних пристроїв на базі ESP32	18
ВИСНОВОК ДО РОЗДІЛУ 1	22
РОЗДІЛ 2. ПРОЄКТУВАННЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ	23
2.1 Визначення вимог до апаратної та програмної частини	23
2.2 Архітектура програмно-апаратного комплексу.....	25
2.2.1 Розробка схеми підключення пристроїв.....	25
2.2.2 Структура програмного забезпечення для інтеграції з HomeAssistant	27
2.3 Вибір технологічного стеку для розробки	29
ВИСНОВОК ДО РОЗДІЛУ 2.....	35
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРИСТРОЇВ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	36
3.1 Розробка апаратної частини пристроїв	36
3.1.1 Схеми підключення компонентів	37
3.1.2 Реалізація прототипу пристроїв.....	41

					ІАЛЦ.467200.003 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата	Програмно-апаратний комплекс для інтеграції користувацьких пристроїв у систему розумного будинку HomeAssistant Пояснювальна записка	Літ.	Аркуш	Аркушів
Розробив		Куделя О. С.				1	71	
Перевірив		Васильєва М. Д.				НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІО-11		
Реценз.								
Н. Контр.		Пономаренко А. М.						
Затвердив								

3.2 Інтеграція пристроїв у HomeAssistant	44
3.3 Програмування пристроїв на базі ESP32	46
3.4 Розробка графічного інтерфейсу користувача	49
ВИСНОВОК ДО РОЗДІЛУ 3.....	54
РОЗДІЛ 4 ОГЛЯД ТА ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАСОБУ	55
4.1 Перевірка стабільності роботи пристроїв у різних умовах	55
4.2 Аналіз продуктивності та енергоспоживання	61
4.3 Аналіз економічної доцільності розробки власних пристроїв	63
ВИСНОВОК ДО РОЗДІЛУ 4.....	65
ВИСНОВКИ	66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	68

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК СКОРОЧЕНЬ

IoT	(Internet of Things) Інтернет речей
HA	(Home Assistant) Домашній помічник
OTA	(Over-the-Air) Оновлення «по повітрю»
API	(Application Programming Interface) Прикладний програмний інтерфейс
FPS	(Frames Per Second) Кадри за секунду

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

У сучасному світі зростає потреба в розумних системах автоматизації, які забезпечують комфорт, енергоефективність та безпеку побуту. Однією з найперспективніших платформ для реалізації таких систем є Home Assistant - потужне середовище з відкритим вихідним кодом, яке дозволяє інтегрувати різноманітні пристрої в єдину екосистему. Зі стрімким розвитком інтернету речей (IoT) користувачі отримали можливість самостійно створювати пристрої, що здатні взаємодіяти з системами розумного дому, використовуючи доступні компоненти, зокрема мікроконтролери, сенсори та дисплеї.

Особливої актуальності тема розумних будинків набула в умовах енергетичної нестабільності та частих відключень електроенергії. Автоматизовані системи дозволяють ефективно керувати споживанням електроенергії, контролювати резервне живлення, слідкувати за станом акумуляторів та інверторів, а також оперативно повідомляти користувача про критичні ситуації. Саме в такі моменти на перший план виходить автономність, ефективність і надійність домашніх систем керування.

Особливу увагу у цій роботі приділено програмно-апаратним рішенням, які окрім стандартних функцій керування також враховують аспекти безпеки - зокрема, моніторинг температури газового котла та морозильної камери. Це особливо важливо для старих систем опалення, які можуть становити потенційну небезпеку в разі несправностей. У межах даної дипломної роботи було розроблено пристрій із сенсорним інтерфейсом, який не лише відображає стан енергосистеми будинку, температуру в приміщеннях, погодні умови та керує розетками, а й реалізує критичні функції безпеки, як-от виявлення можливого згасання котла при відкритому поданні газу та контроль збереження температурного режиму в морозильній камері.

					ІАЛЦ.467200.003 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

Метою дипломної роботи є створення функціонального, інтегрованого пристрою з сенсорним екраном, який здатен збирати дані, передавати їх до Home Assistant, надавати зручний графічний інтерфейс користувача та виявляти потенційно небезпечні ситуації. Об'єктом дослідження є розумна система керування та моніторингу домашніх пристроїв, а предметом - розробка програмно-апаратного комплексу на базі ESPHome з використанням сенсорного дисплея та інтерфейсу LVGL.

Практичне значення роботи полягає у створенні пристрою, який може бути адаптований для широкого спектру побутових завдань - від моніторингу енергоспоживання до виявлення аварійних ситуацій. Запропоноване рішення є масштабованим, енергоефективним, підтримує бездротову інтеграцію з іншими пристроями через Zigbee, а також може бути вдосконалене для подальшого використання в системах енергоменеджменту та безпеки. Отже, дана розробка сприяє підвищенню якості життя, рівня автоматизації та безпеки в умовах сучасного побуту.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

РОЗДІЛ 1

АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО ІНТЕГРАЦІЇ ПРИБОРІВ У СИСТЕМИ РОЗУМНОГО БУДИНКУ

1.1 Огляд існуючих рішень для розумного будинку (HomeAssistant, OpenHAB)

1.1.1 HomeAssistant

Home Assistant - це багатофункціональне рішення яке представляє собою операційну систему розумного будинку з відкритим кодом, яке дає змогу керувати різноманітними IoT-пристроями та організовувати їхню взаємодію без використання хмарних сервісів. Система підтримує численні протоколи зв'язку, включаючи Wi-Fi, Zigbee, Z-Wave, Bluetooth, інфрачервоне управління (IR) та інші.

Сфера застосування охоплює:

- Забезпечення безпеки через відеоспостереження, сигналізацію та детектори руху, диму й відчинення;
- Контроль освітлення з можливістю зміни кольору, яскравості, температури й створення атмосфери;
- Керування побутовими приладами, такими як телевізори, кондиціонери чи роботи-пилососи;
- Управління електроживленням за допомогою інтелектуальних розеток, вимикачів і лічильників;
- Збір даних про температуру, вологість, рівень CO₂ та інші параметри.

Для встановлення системи зазвичай обирають одноплатні комп'ютери (Raspberry Pi 3/4, Odroid, Intel NUC, Orange Pi тощо), NAS-пристрої або міні-ПК.

Існує кілька варіантів інсталяції:

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

- Home Assistant OS - оптимізована система з усім необхідним для запуску, рекомендована більшості користувачів.
- Home Assistant Container - контейнеризована версія без доступу до додатків, але з високою гнучкістю.
- Home Assistant Supervised - інсталяція поверх Linux, що зберігає повну функціональність, вимагаючи знань Docker і Linux.
- Home Assistant Core - базове ядро системи, яке можна інтегрувати в різні ОС, але з обмеженими можливостями для резервного копіювання чи доповнень.

Таблиця 1.1 - Порівняння варіантів інсталяції

Х-ка \ Варіант	OS	Container	Core	Supervised
Автоматизації	Так	Так	Так	Так
Панель Lovelace	Так	Так	Так	Так
Інтеграції	Так	Так	Так	Так
Проекти (Blueprints)	Так	Так	Так	Так
Використання контейнерів	Так	Так	Ні	Так
Супервізор	Так	Ні	Ні	Так
Доповнення	Так	Ні	Ні	Так
Резервні копії	Так	Ні	Ні	Так
Керування ОС	Так	Ні	Ні	Ні

Home Assistant забезпечує взаємодію з численними комунікаційними протоколами, однак найчастіше користувачі обирають Zigbee або Z-Wave через їхню стабільність і підтримку широкого спектра пристроїв.

Zigbee Home Assistant (ZHA) є інтегрованим модулем, який можна налаштувати доволі просто. Хоча перелік сумісних пристроїв обмежується приблизно 800 моделями, більшість поширених гаджетів підтримуються. Для

роботи ZHA необхідно використовувати відповідний Zigbee-адаптер - можна купити вже прошитий або прошити самостійно.

Z-Wave JS - це драйвер, створений відкритою спільнотою, що реалізує повноцінну підтримку протоколу Z-Wave. Для активації необхідний сумісний USB-модуль. В Україні зазвичай використовуються пристрої, що працюють на частоті 868,42 МГц - це актуальна вимога для локальних умов експлуатації.

Альтернативою є Zigbee2MQTT - програмне рішення, яке виконує роль мосту між Zigbee-пристроями та системою Home Assistant через протокол MQTT. Цей підхід дозволяє керувати великою кількістю пристроїв різних брендів (понад 2200 моделей), забезпечуючи стабільну роботу та добру сумісність. Для запуску потрібно мати підтримуваний координаційний шлюз. Найдешевше рішення - CC2531, але через застарілість цього адаптера деякі параметри можуть працювати некоректно. Більш сучасні моделі, як-от CC2652P, демонструють стабільнішу роботу та ширшу сумісність (наприклад, пристрої від Sonoff чи Smartlight).

Для зручного доступу з мобільних пристроїв Home Assistant пропонує офіційні застосунки для Android, iOS і macOS. Однією з ключових функцій є система повідомлень, яка дозволяє користувачу отримувати сповіщення про будь-які події вдома. На пристроях Apple можна налаштовувати швидкі дії (Shortcuts), у тому числі з використанням голосових команд. Користувачі розумних годинників теж не залишилися осторонь - управління домом доступне через Apple Watch або Wear OS.

Home Assistant підтримує встановлення доповнень у вигляді попередньо налаштованих Docker-контейнерів. Офіційні додатки публікуються командою проєкту, а також доступні розширення, які створені та підтримуються спільнотою. Їх можна встановлювати з інтерфейсу через меню: Конфігурація → Додатки → Магазин доповнень. Крім того, існує можливість підключення сторонніх репозиторіїв - для цього потрібно скористатись спеціальним меню управління пакетами (значок у правому верхньому куті).

					ІАЛЦ.467200.003 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

Можливості інтеграції ESPHome:

ESPHome - це інструмент для створення власних розумних пристроїв на базі мікроконтролерів ESP8266 та ESP32, без необхідності глибоких знань програмування. Конфігурація виконується через YAML-файли, з яких автоматично генерується прошивка. Після завантаження мікропрограми пристрій автоматично з'являється у середовищі Home Assistant. Це відкриває можливості для реалізації широкого спектра DIY-проектів - від простих датчиків температури до повноцінних систем моніторингу.

Home Assistant має вбудований потужний механізм автоматизації, який дозволяє пристроям взаємодіяти між собою без участі користувача. Один фізичний пристрій може виконувати декілька ролей. Наприклад, сенсор температури може одночасно передавати дані про вологість або атмосферний тиск. У середовищі Home Assistant такі функції представлені окремими об'єктами - сутностями (entities).

Автоматизації використовуються для реакції на зміну стану або настання певних подій - наприклад, автоматичне вмикання світла при низькому рівні освітлення, увімкнення витяжки при підвищеній вологості у ванній, надсилання повідомлень у разі відкриття дверей або активація кондиціонера, якщо температура перевищує встановлений поріг. Всі автоматизації можна знайти в меню Конфігурація → Автоматизації та сцени. Їх можна створювати як через графічний інтерфейс, так і вручну - у YAML-форматі.

Тригер (trigger) - це подія або умова, яка запускає виконання автоматизації. При настанні тригера система перевіряє додаткові умови (якщо вони вказані) і тоді виконує визначені дії. Серед типів тригерів: події, зміна стану, час, місце розташування та інші.

Умова (condition) - необов'язковий компонент автоматизації. Вона дозволяє задавати додаткові обмеження, які визначають, чи буде виконана дія.

Дія (action) - це конкретне завдання або команда, що виконується у відповідь на спрацьовування тригера та дотримання умов.

					ІАЛЦ.467200.003 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

Резервне копіювання є ключовим компонентом надійної роботи системи розумного дому. Завдяки збереженим копіям можна швидко відновити систему у разі помилки або технічного збою. У Home Assistant цю функцію реалізовано через розділ Конфігурація → Система → Резервні копії. Користувач може створювати повні або часткові бекапи вручну. Крім того, система самостійно створює резервні копії перед оновленням прошивки.

Одним із найзручніших способів автоматизованого резервного збереження є використання доповнення Google Drive Backup. Це рішення дозволяє організувати регулярне збереження копій у хмару Google Диск і локально, згідно з заданим графіком.

Оскільки Home Assistant працює локально в межах домашньої мережі, для управління ним ззовні потрібно налаштувати віддалений доступ.

Офіційне рішення - це Home Assistant Cloud, створений компанією Nabu Casa. Послуга надає легке підключення до хмарних помічників, таких як Google Assistant та Amazon Alexa. Вартість становить 6,50 долара США на місяць або 65 доларів на рік.

Ще одне рішення - DuckDNS, яке забезпечує безкоштовний сервіс динамічного DNS. Воно дозволяє створити адресу, яка динамічно оновлюється при зміні IP. Для коректної роботи DuckDNS потрібне налаштування переадресації портів на маршрутизаторі (port forwarding).

Альтернативно можна скористатись Cloudflared - це тунельний проксі-сервер від Cloudflare, який дозволяє налаштувати безпечний доступ до Home Assistant без відкриття портів. Щоб ним користуватись, потрібно мати доменне ім'я. Розширену інструкцію можна знайти на офіційному сайті або в документації доповнення.

1.1.2 OpenHAB

openHAB (Open Home Automation Bus) - це високофункціональна, повністю відкрита програмна платформа, яка дозволяє централізовано керувати

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

вашим розумним домом. Вона не прив'язана до конкретного виробника пристроїв, що робить її гарним вибором для побудови гнучкої системи автоматизації будь-якої складності.

На відміну від багатьох комерційних рішень, які обмежують користувача лише певним набором пристроїв або протоколів, openHAB об'єднує в собі десятки технологій - від Zigbee, Z-Wave до KNX, MQTT, Bluetooth, Modbus, EnOcean, а також інші більш нішеві рішення. В результаті цього користувач отримує єдину екосистему для керування всіма компонентами свого житла - від освітлення і клімату до безпеки та мультимедіа.

Основні переваги платформи openHAB:

1. *Технологічна нейтральність*

Система не залежить від бренду чи виробника. Ви можете поєднувати пристрої з абсолютно різних джерел, і вони будуть працювати разом, немов частини одного механізму.

2. *Кросплатформеність*

Платформа написана мовою Java, що дозволяє запускати її на будь-якій операційній системі, де підтримується Java Virtual Machine: Linux, macOS, Windows, Android (через Termux або OpenHABian), а також на пристроях типу Raspberry Pi, NAS тощо.

3. *Централізоване керування та інтерфейси*

Незалежно від кількості підключених технологій, система пропонує єдиний інтерфейс взаємодії з користувачем. Інтерфейси доступні у веб-версії (Basic UI, HABPanel), а також через мобільні застосунки для Android та iOS.

4. *Потужна система правил (DSL, Blockly, JavaScript)*

Користувач може створювати гнучкі сценарії автоматизації - як прості "якщо-то", так і складні логічні дерева, що реагують на часові події, умови середовища або навіть машинне навчання (через зовнішні інтеграції).

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

5. Відкритий код та підтримка спільноти

Проект підтримується ентузіастами з усього світу. Це означає активне оновлення, обмін плагінами, стабільні релізи та можливість впливати на розвиток проекту. Нові функції або підтримка обладнання можуть з'явитись буквально за ініціативою користувачів.

6. Інтеграційна гнучкість

OpenHAB підтримує понад 400+ офіційних і неофіційних "bindings" - модулів для зв'язку з пристроями. Крім того, система має відкритий REST API, через який її можна інтегрувати з іншими зовнішніми платформами (наприклад, Node-RED, InfluxDB, Grafana, OpenCV, Telegram Bot тощо).

Додаткові можливості та сценарії використання:

- *Голосове керування*

Завдяки інтеграції з Alexa, Google Assistant, Microsoft AI або офлайн-рішеннями (як Rhasspy) openHAB може виконувати голосові команди.

- *Панелі візуалізації для сенсорних панелей*

HABPanel - це візуальний інтерфейс, який можна налаштувати під сенсорний екран або планшет і розмістити в будь-якому місці будинку для керування.

- *Автоматизація за місцезнаходженням*

За допомогою GPS-геолокації з мобільного застосунку можна автоматично вмикати або вимикати світло, опалення або охоронну систему при вході чи виході з дому.

- *Інтеграція з IoT-сервісами*

Платформа підтримує надсилання повідомлень через Telegram, Email, SMS, MQTT, а також дозволяє надсилати дані в хмару або на власний сервер.

- *Сценарії для енергоефективності*

Можна створити правила, які оптимізують споживання електроенергії,

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

наприклад: вимкнення побутової техніки при пікових навантаженнях мережі або увімкнення зарядки вночі.

1.2 Аналіз можливостей інтеграції користувацьких пристроїв у HomeAssistant

Сучасні системи розумного дому потребують гнучких та масштабованих рішень для інтеграції різноманітних пристроїв. Home Assistant підтримує інтеграцію з широким спектром пристроїв, включаючи комерційні продукти та саморобні рішення. Користувачі можуть додавати власні пристрої за допомогою різних протоколів, таких як MQTT, REST API, ті інші. Це дозволяє створювати індивідуальні рішення, які відповідають специфічним потребам користувача. Одним із ключових інструментів для інтеграції користувацьких пристроїв у HA є ESPHome - фреймворк, який спрощує процес створення прошивок для мікроконтролерів ESP8266 та ESP32

MQTT - це легкий мережевий протокол, який широко використовується в IoT. Home Assistant має повну підтримку MQTT-брокера та клієнтів.

Переваги:

- Підтримується практично всіма мовами (C, Python, Arduino, Node.js тощо)
- Працює навіть на дуже слабкому залізі (ESP8266, Raspberry Pi Zero)
- Можна створювати повністю кастомні пристрої
- Підходить для багатонаправленого зв'язку: пристрій ↔ HA

Недоліки:

- Потрібна наявність та налаштування MQTT-брокера (наприклад, Mosquitto)
- Не такий простий, як ESPHome

Node-RED - це візуальне середовище програмування, що дозволяє створювати логіку автоматизацій та підключення пристроїв через блоки.

Переваги:

- Підтримує REST, MQTT, TCP, Modbus, WebSocket тощо

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

- Гнучкість у створенні логіки керування
- Підходить для зв'язку з промисловими або нестандартними пристроями
- Є офіційний додаток до Home Assistant

Недоліки:

- Потребує більше ресурсів (переважно використовується з Raspberry Pi або сервером)
- Не призначений для мікроконтролерів напряму

REST API / Webhook / HTTP. Home Assistant підтримує інтеграцію з будь-яким пристроєм, який може робити HTTP-запити (GET/POST) або приймати їх. Приклад: ESP32 відправляє значення температури через HTTP POST на спеціальну для цього адресу розумного будинку

Переваги:

- Мінімальна залежність
- Працює навіть зі скриптів, браузера, Shell
- Можна використовувати на мікроконтролерах, ПК, Linux-серверах

Недоліки:

- Немає автоматичного виявлення пристроїв
- Потрібна самостійна реалізація автентифікації, структури запитів

Custom Components. Home Assistant дозволяє створювати власні компоненти на Python, що дає повний контроль над інтеграцією. Як приклад це може бути інтеграція нестандартного API від виробника або Bluetooth-пристрою через власну платформу.

Переваги:

- Повна гнучкість та інтеграція «як рідний» пристрій
- Можна створити повноцінний інтерфейс, обробку, локалізацію, налаштування

Недоліки:

- Потрібне знання Python та структури HA
- Потребує більше часу на підтримку

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

Modbus / Serial / I²C / CAN. НА має підтримку інтерфейсів промислової автоматизації, таких як Modbus RTU, Modbus TCP, а також серійного порту (UART).

Переваги:

- Підходить для пристроїв типу PLC, контролерів опалення, сонячних інверторів
- Надійний протокол у промисловості

Недоліки:

- Більш складне налаштування
- Потрібна фізична інтерфейсна плата або перетворювач

ESPHome - це фреймворк з відкритим кодом, який спрощує створення прошивок для мікроконтролерів ESP8266, ESP32, Raspberry Pi RP2040 та навіть Realtek RTL87xx і Beken BK72xx. Замість складного програмування на C++, користувач описує функціональність пристрою у вигляді YAML-конфігурації. Цей файл задає всі параметри - від налаштування сенсорів до логіки роботи реле, дисплеїв чи кнопок.

ESPHome фактично складається з двох основних компонентів, які взаємодіють між собою:

1. Локальна частина (на мікроконтролері)

Це спеціалізована прошивка, яка компілюється на основі YAML - конфігурацій. Вона завантажується безпосередньо на пристрій ESP та забезпечує взаємодію з фізичними сенсорами, кнопками, реле, світлодіодами, дисплеями, ШІМ-сигналами тощо.

2. Конфігураційна частина (на комп'ютері або SBC)

Це середовище розробки - на вашому ПК або, наприклад, Raspberry Pi - де ви створюєте, редагуєте і компілюєте YAML-файли. Саме тут народжується прошивка, яку ви потім «заливаєте» на свій мікроконтролер через USB або по мережі (OTA - over-the-air).

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

Кожен пристрій описується у вигляді окремої YAML-конфігурації, яка може містити:

- Налаштування Wi-Fi, MQTT або API-з'єднання з Home Assistant
- Параметри GPIO-портів, до яких підключено сенсори або виконавчі пристрої
- Автоматизацію на рівні пристрою - наприклад, ввімкнення реле при спрацюванні датчика руху
- Логіку роботи кнопок, ШІМ, дисплеїв, сервоприводів, ІR-датчиків тощо
- Змінні, обробку подій, фільтрацію шумів сенсорів

Ключові переваги:

- Без написання коду: налаштування відбувається за допомогою YAML, що значно спрощує процес.
- Підтримка сотень компонентів: сенсори температури, вологості, тиску, детектори руху, дисплеї, LED-індикатори тощо.
- Оновлення по повітрю (OTA): дозволяє змінювати прошивку без фізичного доступу до пристрою.
- Локальне керування: не потребує підключення до хмари, що підвищує приватність та надійність.
- Автоматичне виявлення в Home Assistant: ESPHome-пристрій буде знайдено автоматично після підключення до Wi-Fi.
- Можливість автономної роботи: підтримка внутрішніх автоматизацій прямо на мікроконтролері.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

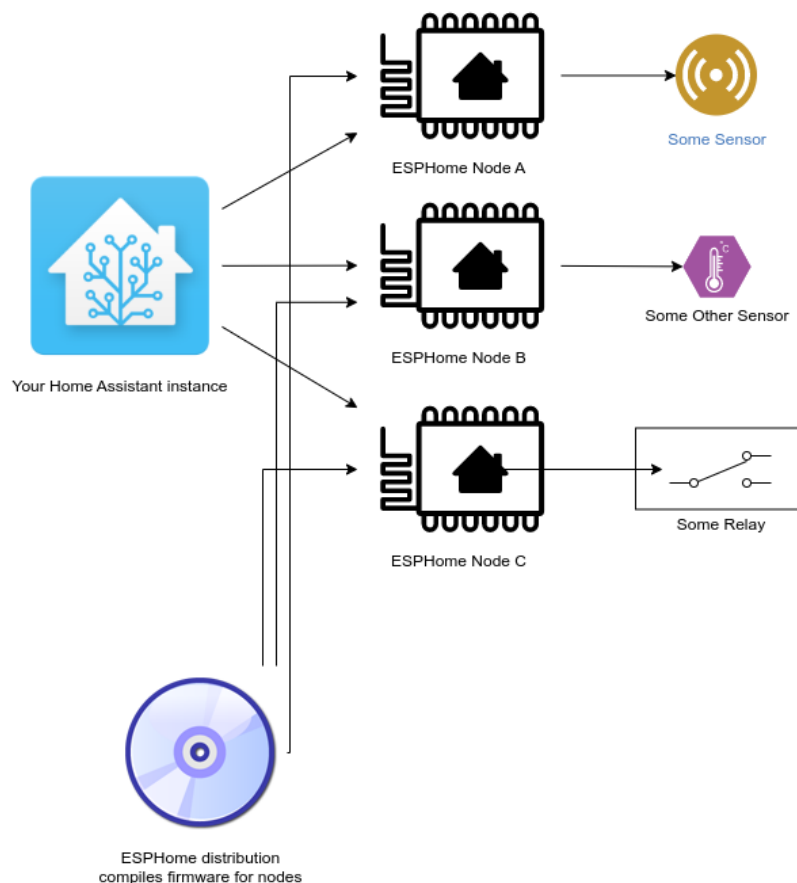


Рис. 1.1 – Спрощена ілюстрація схеми роботи пристроїв

Інтеграція охоплює такі категорії користувачів:

- ДІУ-ентузіасти - створюють власні сенсори, вимикачі, контролери.
- Користувачи розумного дому - інтегрують недорогі користувацькі пристрої у власну екосистему автоматизації.
- Інтегратори - розгортають локальні надійні системи автоматизації для клієнтів.
- Виробники - випускають сертифіковані продукти з підтримкою ESPHome (Made for ESPHome).

ESPHome підтримує:

- ESP8266 та ESP32 - найпоширеніші чипи для IoT-проектів.
- RP2040 - чип від Raspberry Pi Foundation.
- Інші - RTL87xx, BK72xx тощо.
- Навіть десктопні симуляції - для тестування логіки на ПК без пристрою.

Інтеграція створено з урахуванням максимальної сумісності з Home Assistant:

- Автоматичне виявлення: після прошивки пристрою Home Assistant показує повідомлення з пропозицією додати новий пристрій.
- Інтеграція з інтерфейсом: усі ентиті пристрою з'являються в дашборді - температурні сенсори, перемикачі, світлодіоди тощо.
- Підтримка автоматизацій: можна запускати дії в залежності від стану сенсорів, комбінувати кілька умов, додавати до сценаріїв.

Крім того, інтеграція має власний ESPHome Dashboard - вебінтерфейс, який дозволяє:

- створювати, редагувати та оновлювати пристрої;
- зберігати конфігурації;
- бачити лог пристрою в реальному часі;
- інтегрувати резервне копіювання через Home Assistant.

На відміну від класичної схеми, де вся логіка обробляється на сервері, ESPHome дозволяє виконувати логіку безпосередньо на мікроконтролері. Це означає менше затримок, більшу надійність у разі зникнення мережі та автономну роботу.

1.3. Порівняння переваг використання власних пристроїв на базі ESP32

ESP32 - це потужний мікроконтролер від компанії Espressif Systems, який відзначається високою продуктивністю та широкими можливостями для підключення периферійних пристроїв. Основні технічні характеристики ESP32 включають:

- Процесор: Двоядерний Tensilica LX6 з тактовою частотою до 240 МГц, що забезпечує ефективну багатозадачність та підтримку реального часу.
- Пам'ять: 520 КБ SRAM та підтримка зовнішньої Flash-пам'яті, що дозволяє зберігати та обробляти великі обсяги даних.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

- Зв'язок: Вбудовані модулі Wi-Fi 802.11 b/g/n та Bluetooth 4.2 (Classic та Low Energy), що забезпечують гнучкість у бездротовому з'єднанні.
- GPIO: До 39 програмованих виводів для підключення різноманітних сенсорів, моторів та дисплеїв.
- Енергоспоживання: Оптимізований для низького енергоспоживання, з можливістю переходу в режим глибокого сну, споживаючи лише 10 μ A, що ідеально для пристроїв на батарейному живленні.

Використання власних пристроїв на базі мікроконтролера ESP32 у системах розумного дому набуває все більшої популярності завдяки поєднанню високої функціональності, доступної вартості та гнучкості налаштувань. ESP32 є потужним 32-бітним мікроконтролером із вбудованими модулями Wi-Fi та Bluetooth, що дозволяє реалізовувати бездротові рішення без додаткових апаратних витрат. На відміну від класичних платформ, таких як Arduino, ESP32 забезпечує значно вищу продуктивність, більший обсяг оперативної пам'яті та ширші можливості для розширення. Завдяки цьому мікроконтролер ідеально підходить для реалізації локальних сенсорів, виконавчих модулів, інформаційних дисплеїв та інших компонентів систем автоматизації.

Порівняємо ESP32 з найпопулярнішими платами сімейства Arduino - Arduino Uno та Arduino Mega 2560, які часто використовуються початківцями.

Таблиця 1.2 - Порівняння характеристик найпопулярніших плат

Характеристика	ESP32	Arduino Uno	Arduino Mega 2560
Ціна (орієнтовно, \$)	3 - 6	8 - 12	12 - 20
Архітектура	32-біт (Xtensa)	8-біт (AVR)	8-біт (AVR)
Частота процесора	До 240 МГц	16 МГц	16 МГц
RAM	520 КБ	2 КБ	8 КБ

Продовження таблиці 1.2

Flash	До 16 МБ (зовнішня)	32 КБ	256 КБ
Wi-Fi	Так	Ні	Ні
Bluetooth	Так	Ні	Ні
GPIO	До 39	14	54
Енергоспоживання	Дуже низьке (режими сну)	Високе	Високе
OTA оновлення	Так (через ESPHome)	Ні	Ні
Ціна Wi-Fi Shield	Не потрібен	+8–10 \$	+8–10 \$

ESP32 значно перевершує Arduino як за продуктивністю, так і за функціональністю, маючи при цьому нижчу вартість. Для досягнення схожого функціоналу на Arduino Uno або Mega потрібно докуповувати додаткові плати (Wi-Fi, Bluetooth, RTC тощо), що збільшує загальну вартість проекту в 2-3 рази. Arduino використовує просту екосистему, проте ESPHome робить конфігурацію ESP32 ще доступнішою завдяки YAML та автоматичній інтеграції з Home Assistant.

Однією з ключових переваг використання ESP32 у зв'язці з фреймворком ESPHome є простота розробки. Налаштування функціоналу пристрою здійснюється через YAML-конфігурації, без потреби писати код мовою C або C++. Це значно знижує поріг входу для новачків та пришвидшує розробку прототипів. Інтеграція з Home Assistant відбувається автоматично: новий пристрій розпізнається системою одразу після підключення до мережі. Крім того, ESPHome дозволяє налаштовувати пристрої з можливістю OTA-оновлення (оновлення по повітрю), що критично важливо для встановлених у важкодоступних місцях модулів.

Таким чином, використання власних пристроїв на базі ESP32 відкриває широкі можливості для побудови індивідуалізованих систем автоматизації, які точно відповідають вимогам користувача, легко інтегруються з існуючою інфраструктурою Home Assistant і залишаються доступними в фінансовому плані. Це рішення є ідеальним для ентузіастів, розробників, а також малих комерційних проєктів, що прагнуть до гнучкості, безпеки та автономності у сфері розумного дому.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

ВИСНОВОК ДО РОЗДІЛУ 1

У даному розділі було проведено аналіз сучасних рішень для реалізації систем розумного будинку, зокрема Home Assistant та openHAB, які є найпоширенішими платформами з відкритим кодом. У результаті порівняння виявлено, що Home Assistant відзначається кращою інтеграцією з широким спектром пристроїв, активною спільнотою, регулярними оновленнями та високим рівнем зручності для кінцевого користувача. OpenHAB, у свою чергу, забезпечує високу гнучкість та кросплатформність, проте поступається Home Assistant у зручності налаштування та масштабованості для домашніх сценаріїв.

Окрема увага була приділена аналізу можливостей інтеграції користувацьких пристроїв у систему Home Assistant. Зокрема, розглянуто найбільш поширені протоколи зв'язку (MQTT, REST API, Modbus, Zigbee, Z-Wave) та з'ясовано, що однією з найзручніших і відносно найефективніших технологій для швидкої реалізації власних пристроїв є ESPHome. Цей фреймворк дозволяє створювати функціональні модулі без глибоких знань програмування, використовуючи мікроконтролери ESP8266 або ESP32.

У контексті аналізу апаратної платформи, ESP32 було визнано більш продуктивним, економічно доцільним та технічно гнучким рішенням порівняно з популярними платами Arduino. Його використання у поєднанні з ESPHome значно спрощує створення власних автоматизованих пристроїв, дозволяє забезпечити повну інтеграцію з Home Assistant, локальне управління та OTA-оновлення. Власні рішення на ESP32 надають користувачу повний контроль над функціональністю, дизайном та логікою роботи пристроїв, що є важливою перевагою в умовах зростаючого попиту на персоналізовані та автономні системи розумного будинку.

Отже, результати аналізу підтверджують доцільність обраної архітектури для реалізації програмно-апаратного комплексу в межах цієї дипломної роботи.

					ІАЛЦ.467200.003 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2

ПРОЄКТУВАННЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ

2.1 Визначення вимог до апаратної та програмної частини

На основі проведеного аналізу платформ, технологій та вимог до функціонування системи, сформовано повний перелік вимог до апаратної та програмної складових проєктованого програмно-апаратного комплексу. Ці вимоги є базовими критеріями, яким має відповідати система для досягнення заявленої функціональності, надійності та інтеграції з середовищем Home Assistant.

Апаратна частина повинна забезпечувати апаратно-фізичну взаємодію з навколишнім середовищем: збирання інформації від сенсорів, передачу керуючих сигналів до виконавчих пристроїв, обробку даних у режимі реального часу та відображення результатів взаємодії з користувачем. Основні апаратні вимоги:

- Мікроконтролер - ESP32 з двоядерним процесором та вбудованими модулями Wi-Fi і Bluetooth. Обрано через високу продуктивність, велику кількість GPIO-виводів, підтримку енергозберігаючих режимів і повну сумісність з ESPHome.
- Дисплей - кольоровий сенсорний TFT/IPS дисплей (мінімум 3.5"), сумісний з графічною бібліотекою LVGL для виведення інформації у вигляді сторінок, іконок, кнопок та індикаторів стану.
- Температурні сенсори - щонайменше 8 цифрових підключених до ESP32, для контролю температури в різних точках системи: кімната, бойлери, газовий котел, проточна вода, морозильна камера тощо.
- Живлення - стабілізоване джерело живлення з резервуванням або підключенням до безперебійного джерела.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

- Зв'язок з Home Assistant - через Wi-Fi

Програмна частина відповідає за логіку роботи пристрою, збір та передачу даних, обробку подій, взаємодію з користувачем та безперебійну інтеграцію з системою розумного будинку Home Assistant. Програмна архітектура побудована на базі фреймворку ESPHome, що дозволяє реалізувати функціонал пристрою шляхом опису в YAML-файлі.

Основні вимоги до програмного забезпечення:

1. Глибока інтеграція з Home Assistant

- Підтримка Native API для безпосереднього зв'язку без проміжних протоколів (як-от MQTT).
- Автоматичне виявлення пристрою в середовищі Home Assistant після підключення до локальної мережі.
- Передача всіх сенсорних значень та станів сутностей до Home Assistant у реальному часі.
- Можливість керування виконавчими пристроями (розетками) з інтерфейсу Home Assistant або з фізичного інтерфейсу пристрою.

2. Функціональна багатосторінкова графічна оболонка

- Побудова графічного інтерфейсу на базі бібліотеки LVGL.
- Реалізація мінімум 5 сторінок:
 - Головна: поточна генерація електроенергії, споживання з мережі, споживання з акумулятора, загальне споживання.
 - Температури: вивід значень з 8 сенсорів у реальному часі.
 - Керування розетками: кнопки для вмикання/вимикання 11 пристроїв (бойлери, насоси, освітлення, вентилятор, морозильна камера тощо).
 - Погода та час: синхронізація з Home Assistant для відображення часу, дати, температури, вологості, хмарності та стану погоди в вигляді піктограми.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

- Налаштування: керування яскравістю дисплея, активація функцій слайдшоу, антивигорання, автоматичного вимкнення підсвітки.

3. Система безпеки та захисту

- Моніторинг аварійних ситуацій:
 - Температура вихлопу газового котла $<75^{\circ}\text{C}$ за умови відкритого газового клапана понад 90 секунд - виведення критичного повідомлення.
 - Температура морозильної камери $> -5^{\circ}\text{C}$ - відображення попередження.

4. Підтримка користувацьких функцій

- OTA-оновлення (Over The Air) - можливість бездротового оновлення прошивки через Home Assistant або ESPHome Dashboard.
- Функція слайдшоу - автоматична зміна сторінок інтерфейсу в циклі при неактивності.
- Функція антивигорання екрана - тренування пікселів випадковими кольорами 4 рази на ніч по 40 хвилин.
- Кастомізація - передбачено можливість розширення конфігурації YAML для додавання нових сенсорів або зміни логіки автоматизації.

5. Автономність

- Збереження останніх отриманих значень сенсорів та станів керованих пристроїв у разі втрати з'єднання з Home Assistant.

2.2. Архітектура програмно-апаратного комплексу

2.2.1 Розробка схеми підключення пристрої

Побудова схеми підключення програмно-апаратного комплексу є ключовим етапом проєктування системи, оскільки саме через відповідну топологію забезпечується стабільна передача даних, енергетична незалежність та гнучкість у масштабуванні. В межах даного проєкту була створена спрощена

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

логічна схема, яка ілюструє загальні принципи з'єднання основних компонентів системи розумного дому.

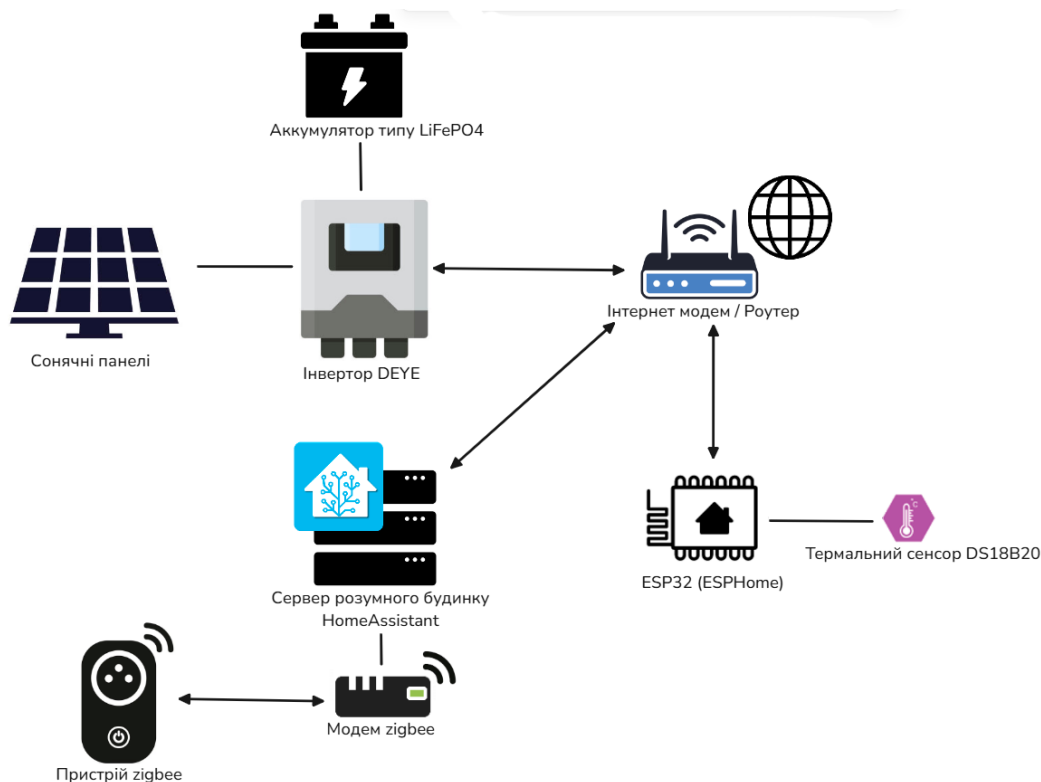


Рис. 2.1 – Загальна схема підключення пристроїв

На схемі показано, що сонячні панелі фізично підключаються до інвертора DEYE, який також з'єднаний з акумулятором типу LiFePO₄. Це дозволяє створити енергетично автономну систему, в якій надлишок енергії зберігається в батареях і може використовуватися в періоди низької генерації.

Інвертор DEYE має можливість передавати інформацію до локальної мережі через інтернет-маршрутизатор, який є центральною точкою зв'язку для всіх пристроїв. Саме через нього здійснюється зв'язок з основним сервером автоматизації - Home Assistant, який працює на локальному комп'ютері. Він отримує дані від інвертора (через мережу), зберігає історію, здійснює аналіз та формує автоматизації.

Для забезпечення керування бездротовими пристроями використовується модем Zigbee, підключений безпосередньо до комп'ютера Home Assistant. Через цей модем передаються сигнали на різні Zigbee-пристрої - зокрема

розумні розетки, освітлення, реле тощо. Передача даних між модемом і пристроями Zigbee відбувається по бездротовому.

Окремим функціональним блоком є пристрій на базі ESP32 з попередньо встановленою прошивкою ESPHome. Він також підключається до Wi-Fi-мережі через маршрутизатор і забезпечує:

- збір та передачу значень температури від підключених термодатчиків DS18B20;
- керування екраном і графічним інтерфейсом;
- локальні обчислення та контроль безпечних сценаріїв (наприклад, температура котла або морозильника);
- виведення попереджень у випадку помилок.

Термодатчик DS18B20 підключається безпосередньо до одного з GPIO-портів ESP32 через цифровий інтерфейс 1-Wire. Передача даних до Home Assistant відбувається через внутрішній API ESPHome.

Ця структура дозволяє масштабувати систему, додавати нові датчики та пристрої без радикальних змін архітектури, а також забезпечує повну автономність та локальне керування без необхідності підключення до зовнішніх хмарних сервісів.

2.2.2 Структура програмного забезпечення для інтеграції з HomeAssistant

Для реалізації інтеграції між апаратною частиною пристрою на базі ESP32 та системою Home Assistant було використано фреймворк ESPHome, що дозволяє створювати прошивки за допомогою YAML-конфігураційних файлів. Завдяки модульній структурі ESPHome, конфігураційний файл можна умовно поділити на кілька логічних блоків, кожен з яких виконує свою окрему функцію та забезпечує гнучке керування як апаратними, так і програмними компонентами системи.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

Загальна структура YAML-файлу:

- Базові налаштування пристрою - визначення імені, типу плати, використання фреймворку ESP-IDF, параметрів PSRAM, а також базові параметри пам'яті та відлагодження. Це забезпечує правильну ініціалізацію апаратної платформи ESP32-S3.
- Налаштування мережі - блок wifi: з даними доступу до локальної мережі, а також API і OTA для активації автоматичного виявлення пристрою системою Home Assistant і оновлення прошивки "по повітрю". Усі чутливі параметри, такі як паролі, винесено у файл secrets.yaml, що забезпечує безпеку конфігурації.
- Часовий модуль - синхронізує внутрішній годинник ESP32 з NTP-сервером та ініціює регулярне виконання задач у певний час доби.
- Інтеграція з Home Assistant - за допомогою компонентів sensor та switch типу platform:homeassistant здійснюється обробка зовнішніх даних, зокрема температури, стану розеток, інвертора тощо. Це дозволяє пристрою не лише обробляти локальні дані, а й приймати рішення на основі станів системи.
- Власні сенсори - наприклад, температурні датчики DS18B20 підключено через компонент dallas, а їх значення обробляються як sensor з вказаною адресою та одиницями виміру.
- Інтерфейс користувача (LVGL) - окрема частина YAML-файлу відповідає за налаштування дисплея та ємнісного сенсорного екрана. Через об'єкт lvgl оголошуються сторінки, шрифти, кольори, поведінка екрана в режимі очікування, а також реалізовано бокову панель і логіку перемикання між сторінками.
- Компоненти керування - блоки output і light використовуються для керування підсвіткою екрана, binary_sensor - для обробки станів кнопок, а script - для реалізації логіки оновлення інтерфейсу, обробки подій, перемикання сторінок, реагування на критичні значення сенсорів тощо.

					ІАЛЦ.467200.003 ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

- Логіка автоматизації - реалізується через `interval` та `lambda` вирази, які дозволяють вбудовувати умовні конструкції прямо в `YAML`.

Уся структура `YAML`-файлу логічно впорядкована, розбита на функціональні секції, що полегшує супровід проєкту, повторне використання конфігурації для інших пристроїв, а також мінімізує ймовірність помилок.

2.3. Вибір технологічного стеку для розробки

При виборі інструментів і платформ для розробки було враховано такі критерії: сумісність з `Home Assistant`, простота розробки та підтримки, наявність документації, масштабованість, енергоефективність, а також можливість використання у вбудованих системах.

У даному проєкті використано перевірені та широко підтримувані інструменти та технології, які дозволяють реалізувати повноцінну систему розумного дому з можливістю глибокої інтеграції користувацьких пристроїв.

`ESP32` - потужний мікроконтролер від компанії `Espressif Systems`, який підтримує одночасну роботу `Wi-Fi` та `Bluetooth`, має двоядерний процесор до 240 МГц, велику кількість `GPIO`-виводів та периферійних інтерфейсів (`I2C`, `SPI`, `UART`, `PWM`, `ADC`, `DAC`). Завдяки поєднанню високої продуктивності, енергоефективності та низької вартості `ESP32` є ідеальним вибором для систем автоматизації та сенсорних пристроїв. Також `ESP32` має широку підтримку в спільноті розробників, велику кількість бібліотек і документації, що значно спрощує розробку.

`ESPHome` - це фреймворк з відкритим вихідним кодом, призначений для створення прошивок для `ESP32/ESP8266` за допомогою конфігураційних файлів на `YAML`. Його головні переваги:

- Спрощений підхід до конфігурації пристроїв без програмування мовою `C++`;
- Автоматична інтеграція з `Home Assistant` через `Native API`;
- Підтримка широкого спектра сенсорів, дисплеїв, виконавчих пристроїв;

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

- Підтримка OTA-оновлень (оновлення прошивки через Wi-Fi);
- Можливість реалізації локальної логіки на мікроконтролері.

ESPHome ідеально підходить для задач, де потрібна швидка, стабільна та зрозуміла інтеграція з Home Assistant.

Одним із ключових елементів обраного технологічного стеку є формат YAML, який використовується у фреймворку ESPHome як основний інструмент опису конфігурації програмної логіки пристрою. YAML - це мова розмітки, що вирізняється простотою синтаксису, високою читабельністю, структурованістю і зручністю для опису даних у вигляді вкладених структур.

Причини вибору:

1. Лаконічність і простота

YAML дозволяє легко описувати навіть складні конфігурації у вигляді деревоподібної структури, не використовуючи зайвих символів чи складного синтаксису, характерного для XML або JSON. Це дає змогу зосередитись на логіці роботи пристрою без написання програмного коду традиційними мовами програмування.

2. Читаємість для людини

YAML орієнтований на людину: він легко читається, не потребує додаткового програмного інтерпретатора для розуміння. Завдяки цьому конфігурації ESPHome можуть супроводжуватися навіть нефахівцями у програмуванні - наприклад, системними адміністраторами або техніками.

3. Повна підтримка в ESPHome

У фреймворку ESPHome конфігураційний файл `.yaml` є єдиним джерелом опису логіки роботи пристрою. У ньому визначаються:

- сенсори (температура, вологість, стан GPIO тощо),
- виконавчі механізми (реле, ШІМ-виходи),
- з'єднання з Home Assistant,
- інтерфейс дисплея (через бібліотеку LVGL),

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

- правила автоматизації (умови, тригери, дії),
- поведінка при помилках,
- оновлення прошивки по повітрю (ОТА).

4. Можливість повторного використання конфігурацій

YAML дозволяє створювати шаблони, компоненти, включати файли з окремими секціями конфігурації. Це значно спрощує підтримку великих проєктів або серійне виробництво однакових пристроїв із незначними відмінностями. У межах цього дипломного проєкту такий підхід дозволяє структурувати налаштування для сенсорів, елементів інтерфейсу, кнопок керування, сторінок GUI окремо

5. Пряма інтеграція з Home Assistant

YAML є основною мовою конфігурацій не тільки в ESPHome, а й у самій платформі Home Assistant. Використання YAML забезпечує концептуальну єдність обох систем, дозволяє легко адаптувати конфігурації, копіювати окремі частини, автоматизувати сценарії та спрощує міграцію конфігурацій між пристроями.

6. Можливість валідації та перевірки помилок

Усі YAML-файли в ESPHome перевіряються на правильність структури перед компіляцією. Це дозволяє на ранніх етапах виявляти помилки, недопустимі параметри, дублікати, що мінімізує ризик некоректної роботи прошивки після OTA-оновлення.

Для побудови графічного інтерфейсу користувача в рамках проєкту було обрано бібліотеку LVGL (Light and Versatile Graphics Library) - одну з найпотужніших, універсальних та оптимізованих бібліотек для створення графіки у вбудованих системах з обмеженими апаратними ресурсами. Це бібліотека з відкритим вихідним кодом, яка активно підтримується розробниками, має широку документацію, підтримує численні платформи та сумісна з багатьма мікроконтролерами, включно з ESP32.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

Основні переваги використання LVGL:

1. Продуктивність на слабкому залізі

LVGL спеціально оптимізована для роботи на пристроях з обмеженим обсягом оперативної пам'яті та невисокою тактовою частотою.

2. Гнучка модульна структура

LVGL побудована за модульною архітектурою, що дозволяє включати лише ті компоненти, які необхідні для проєкту. Це дозволяє зменшити споживання пам'яті та підвищити швидкість рендерингу. Серед доступних елементів інтерфейсу - кнопки, текстові поля, лінії прогресу, перемикачі, вкладки, списки, графіки, зображення, QR-коди та багато іншого.

3. Підтримка сенсорного керування

LVGL дозволяє реалізувати повноцінну сенсорну взаємодію: дотик, свайпи, довге натискання, прокрутку та інші жести. Це особливо важливо для пристроїв, які використовуються у побуті, де інтуїтивне та просте керування є критичною вимогою.

4. Підтримка багатосторінкових інтерфейсів

У контексті даного проєкту реалізована багатосторінкова структура - окремі сторінки для моніторингу енергетичних параметрів, температурних сенсорів, керування розетками, перегляду погоди та часу, а також сторінка з налаштуваннями. Перемикання між сторінками відбувається плавно, без мерехтіння, за рахунок апаратного буферизаційного механізму та інтернального кешування LVGL.

5. Стилзація інтерфейсу

LVGL підтримує створення та застосування стилів до графічних елементів - кольори, прозорість, шрифти, закруглення, тіні, відступи тощо. Це дозволяє реалізувати єдиний графічний стиль, адаптований під інтерфейс розумного дому, зі зручною візуальною ієрархією та чіткими акцентами на важливих елементах.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

6. Події та обробка подій

Кожен елемент інтерфейсу в LVGL може реагувати на події - натискання, зміни стану, втрату фокусу, таймери тощо. Це дозволяє легко прив'язати функціональність пристрою до взаємодії з користувачем, наприклад:

- запуск слайдшоу сторінок після 30 секунд бездіяльності;
- зміна яскравості екрана при натисканні кнопки;
- відображення попереджень при аварійних умовах (низька температура газового котла або висока температура морозильника).

7. Підтримка міжнародних мов та шрифтів

Для відображення різномовного інтерфейсу важливою є підтримка Unicode та можливість підключення власних шрифтів. LVGL дозволяє створити кастомні шрифти, які будуть коректно відображатися на всіх елементах інтерфейсу.

8. Інструменти розробника

LVGL має супровідні інструменти для попереднього проектування інтерфейсу, зокрема SquareLine Studio - GUI-дизайнер, сумісний з LVGL, який дозволяє створювати інтерфейс у візуальному редакторі та експортувати його до проекту. Це значно спрощує макетування й пришвидшує розробку.

9. Масштабованість

У разі потреби інтерфейс, побудований на LVGL, може бути легко масштабований до дисплеїв з вищою роздільною здатністю або до інших мікроконтролерів. Розроблений графічний інтерфейс є перспективним для розширення функціоналу в майбутньому.

Враховуючи всі наведені переваги, LVGL було обрано як базову графічну бібліотеку для реалізації інтерфейсу сенсорного пристрою. Альтернативи, такі як TFT_eSPI або Ucglib, не мають аналогічного функціоналу щодо обробки подій, багатосторінковості, анімацій та стилів. Крім того, LVGL активно

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

використовується в промисловості та вбудованих системах, зокрема в медичних приладах, терміналах самообслуговування, портативних пристроях, що підтверджує її надійність і продуктивність.

Home Assistant - популярна open-source платформа для автоматизації будинку, яка дозволяє об'єднати пристрої різних виробників і стандартів у єдину систему. Її вибір обґрунтований такими перевагами:

- Підтримка понад 2500 інтеграцій, включно з ESPHome, MQTT, Zigbee, Z-Wave тощо;
- Локальна робота без залежності від хмари;
- Візуальний редактор автоматизацій, панель керування, можливість резервного копіювання;
- Широка документація, активна спільнота, регулярні оновлення.

Сервер Home Assistant розгортається на одноплатному комп'ютері (наприклад, Raspberry Pi 4), що забезпечує автономність та локальний контроль.

Wi-Fi: використовується як основний протокол для обміну даними між ESP32-пристроєм та сервером Home Assistant у локальній мережі.

ESPHome Native API: забезпечує швидкий, надійний і безпечний двосторонній зв'язок між пристроєм та системою автоматизації.

Zigbee (через Zigbee2MQTT або ZHA): використовується для керування комерційними розетками та іншими пристроями розумного дому, що не входять безпосередньо в ESP32-систему, але взаємодіють через Home Assistant.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

ВИСНОВОК ДО РОЗДІЛУ 2

У другому розділі було виконано повноцінне проектування програмно-апаратного комплексу для інтеграції у систему розумного дому. Було визначено вимоги до апаратної та програмної частин, обґрунтовано вибір основних апаратних компонентів та побудовано загальну архітектуру системи, яка охоплює взаємодію між інвертором, сервером Home Assistant, Zigbee-пристроями та вузлом на базі ESP32.

Здійснено детальний аналіз структури програмного забезпечення, реалізованого у форматі YAML з використанням фреймворку ESPHome. Обрана технологічна база повністю відповідає вимогам до надійності, масштабованості та енергонезалежності, що дозволяє у подальшому ефективно реалізувати прототип і забезпечити його стабільну інтеграцію в екосистему розумного будинку.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ПРИСТРОЯ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Розробка апаратної частини пристроя

Для реалізації апаратної частини програмно-апаратного комплексу було обрано високофункціональний модуль ESP32-4827S043 - одноплатне рішення на базі ESP32 із кольоровим TFT-дисплеєм (4.3", 800×480) та ємнісним сенсорним інтерфейсом (capacitive touch). Завдяки високому ступеню інтеграції цей модуль значно зменшує кількість зовнішніх з'єднань і спрощує апаратну реалізацію, а також має усі необхідні периферійні інтерфейси для підключення сенсорів, індикаторів, SD-карти, GPIO тощо.

Побудована апаратна система охоплює не лише центральний обчислювальний модуль ESP32, а й окремі функціональні вузли, описані нижче. Схеми були отримані з документації до модуля та адаптовані до умов реалізації конкретного проєкту.

3.1.1. Схеми підключення компонентів

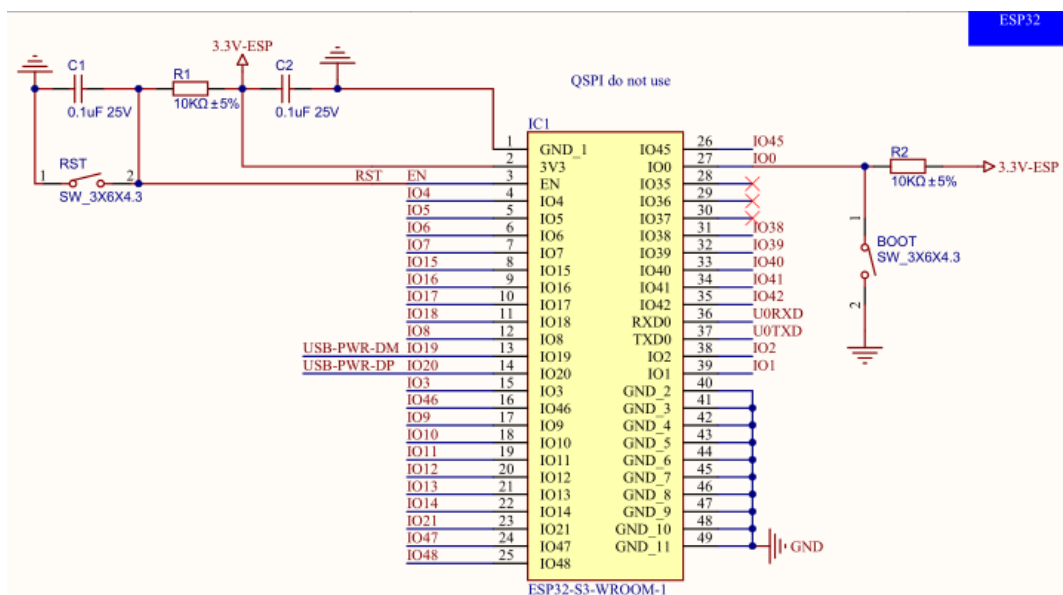


Рис. 3.1 – Загальна схема ESP32 (GND, живлення, основні виводи)

Дисплей підключено до ESP32 через паралельну RGB-шину, що забезпечує високу швидкість оновлення зображення та плавну анімацію інтерфейсу. Передача даних здійснюється через кольорові канали та синхросигнали, що дозволяє виводити графіку в реальному часі без затримок.

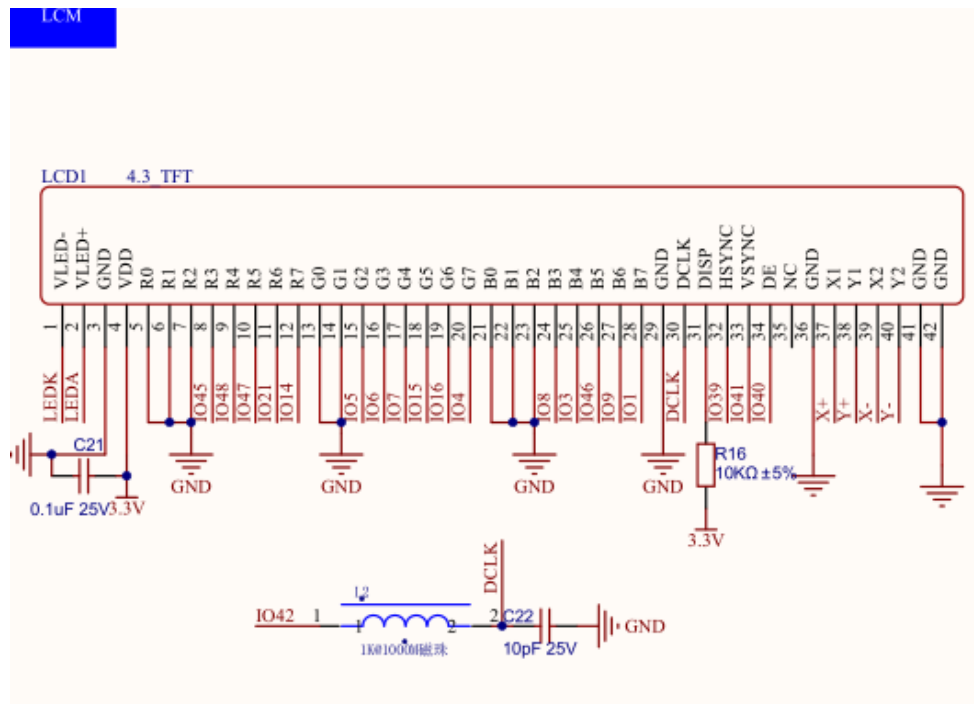


Рис. 3.2 – LCM (Liquid Crystal Module) TFT-дисплей

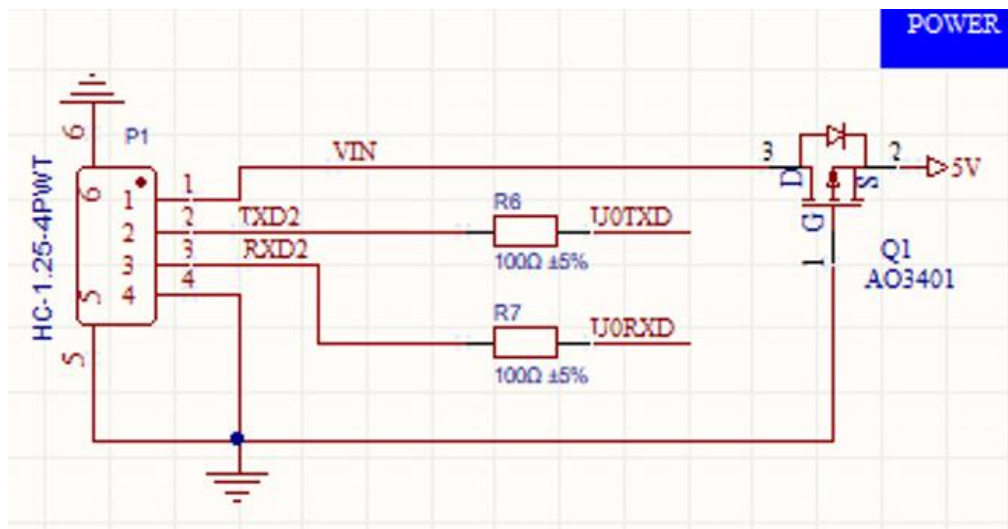


Рис. 3.3 – Схема живлення через роз'єм

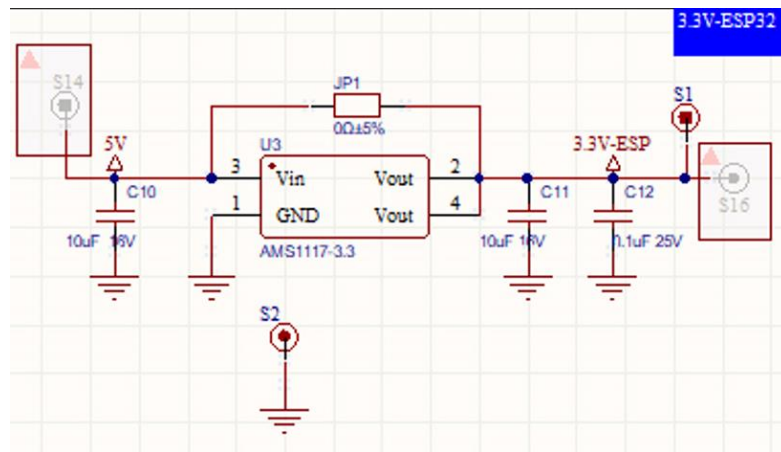


Рис. 3.4 – Схема живлення 3.3В для ESP32

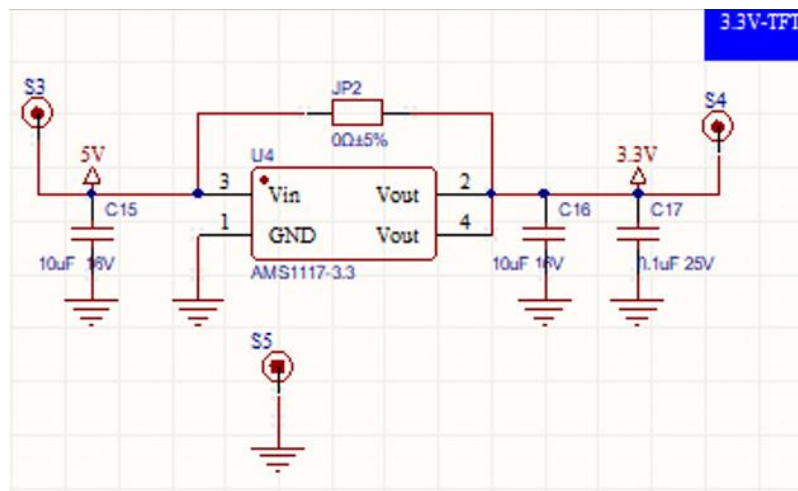


Рис. 3.5 – Схема живлення 3.3В для TFT-дисплею

На платі вбудовано USB to TTL-конвертер (CH340C), який дозволяє здійснювати прошивку ESP32, а також налагодження та серійний моніторинг через інтерфейс UART

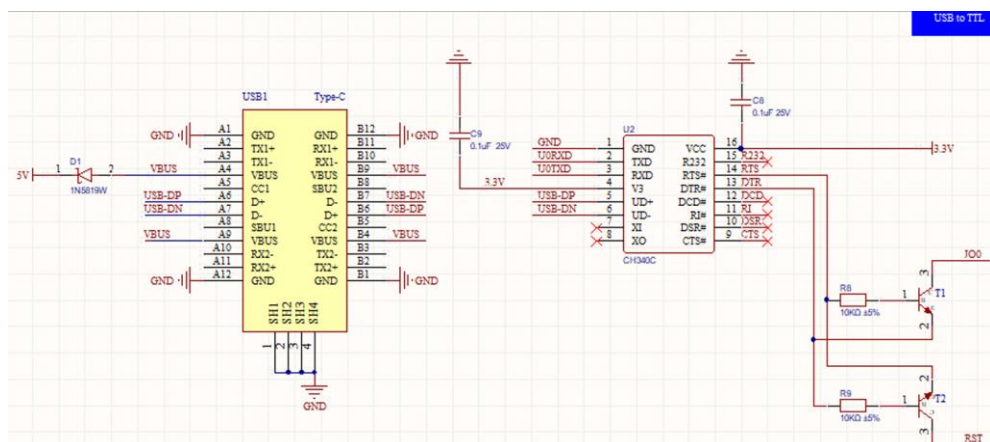


Рис. 3.6 – Схема USB до TTL

На платі ESP32-4827S043 передбачені додаткові роз'єми розширення (Extended IO), реалізовані у вигляді компактних чотириконтактних конекторів типу HC-1.25-4PWT і SH-1.0-4PWB з малим кроком виводів (1.25 мм і 1.0 мм відповідно). Таке інженерне рішення дозволяє щільно розмістити інтерфейси, зберігаючи компактність конструкції плати, та забезпечує надійне підключення зовнішніх модулів, сенсорів і виконавчих пристроїв до шин SPI, UART та інших цифрових ліній ESP32.

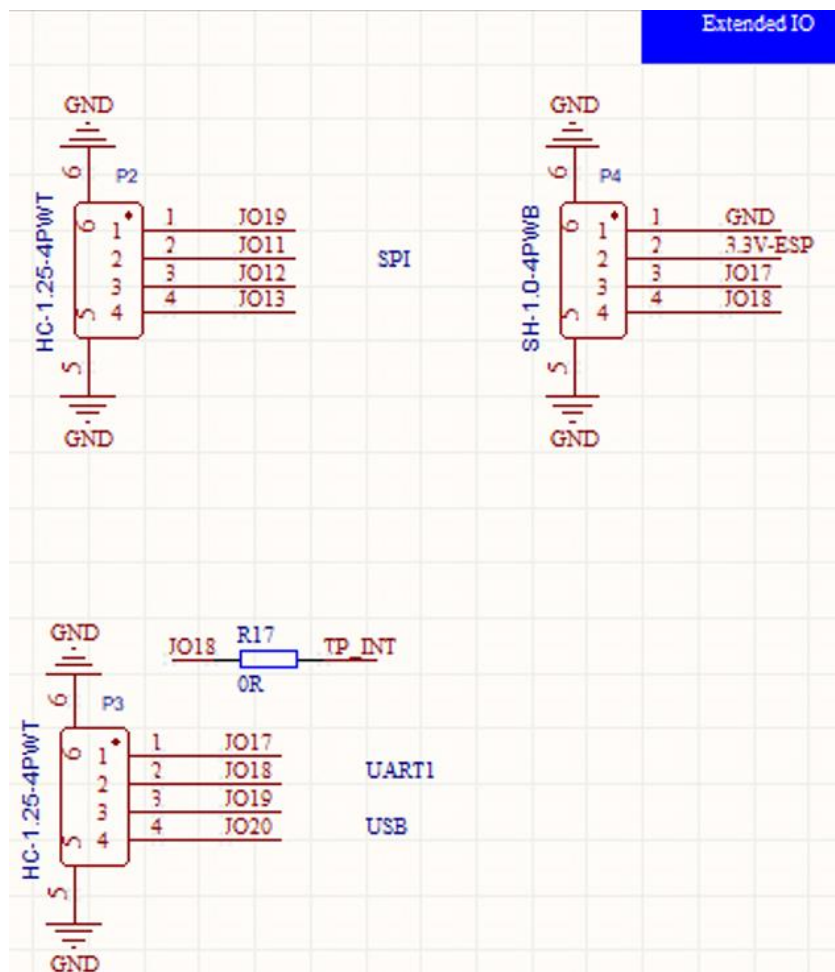


Рис. 3.6 – Схема додаткових роз'ємів вводу-виводу

Через зазначені роз'єми на платі виведені вільні GPIO-виводи, а на SH-1.0-4PWB шини живлення (3.3 В) і земля (GND). Це дозволяє підключати як цифрові, так і аналогові пристрої, використовуюючи стандартний чотириконтактний кабель із конектором відповідного типу. Такий підхід є

надзвичайно зручним для швидкого монтажу, налагодження та подальшого обслуговування системи.

У рамках цього проєкту один із роз'ємів використовується для підключення цифрових температурних сенсорів DS18B20. Сенсори підключаються за допомогою загальної 1-Wire шини з підтягувальним резистором, що дозволяє передавати дані по одному сигнальному проволу навіть при підключенні декількох сенсорів одночасно.

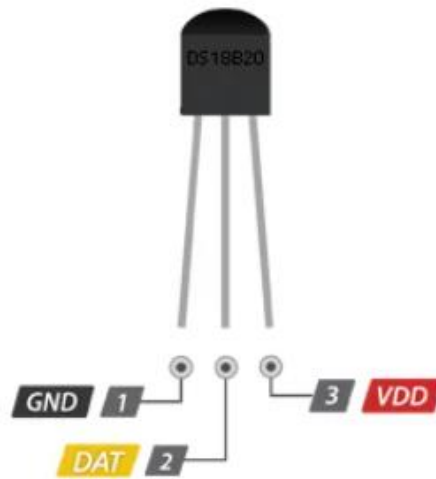


Рис. 3.7 – Схема назначення пінів температурного сенсора DS18B20

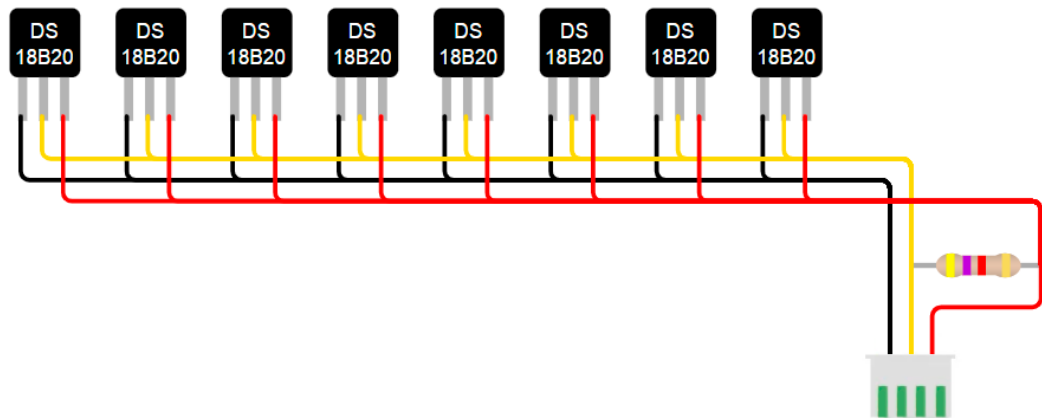


Рис. 3.8 – Схема зв'язки температурних сенсорів для підключення через роз'єм SH-1.0-4PWБ

Також на платі є LED драйвер, який призначений для живлення підсвітки TFT-дисплея. Яскравість підсвітки може регулюватися за допомогою подачі широтно-імпульсного сигналу з мікроконтролера ESP32 на керуючий вхід драйвера. Це дозволяє адаптувати рівень освітлення дисплея до умов використання або вимог інтерфейсу користувача.

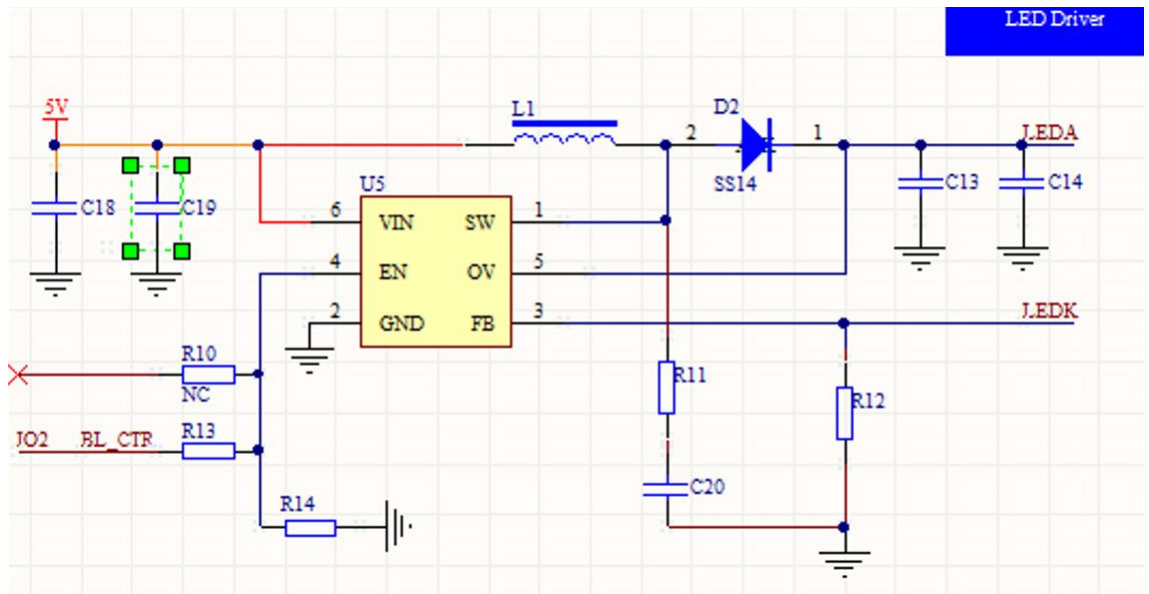


Рис. 3.9 – Схема LED драйвера

3.1.2. Реалізація прототипу пристрою

У процесі реалізації прототипу пристрою було прийнято інженерно обґрунтоване рішення використати вже готовий та комерційно доступний пристрій ESP32-4827S043 як основну апаратну платформу. Даний модуль поєднує в собі мікроконтролер ESP32, кольоровий дисплей із роздільною здатністю 800×480 пікселів, ємнісну сенсорну панель, схему живлення, стабілізатор напруги, інтерфейс підключення SD-карти, інтерфейс USB-to-TTL для прошивки, індикатори стану, та розширені порти вводу/виводу - усе це змонтовано на одній друкованій платі.

З огляду на складність архітектури цього пристрою, щільність монтажу, мініатюрні контактні майданчики та наявність декількох високочастотних інтерфейсів, створення повністю аналогічного прототипу в умовах ручного

виготовлення в навчальному середовищі було б надзвичайно ресурсозатратним як за часом, так і за вартістю. Виробництво друкованої плати із кількома шарами, дотримання радіотехнічних обмежень, правильна розводка шин дисплея та керування підсвіткою - усе це потребує спеціалізованого обладнання, досвіду, а також багатоетапного налагодження. Тому з техніко-економічної точки зору було визнано доцільним використання готового пристрою, який відповідає всім заявленим вимогам до апаратної частини.

Готовий модуль, що використовується в даній розробці, має також низку додаткових переваг. Зокрема, він поставляється з попередньо змонтованими роз'ємами розширення, що дозволяє оперативно підключати зовнішні сенсори, такі як температурні датчики, без необхідності ручного пайкового монтажу. Крім того, пристрій має інтегрований USB-інтерфейс для завантаження прошивки, що значно спрощує етап налагодження та оновлення програмного забезпечення, а також мінімізує потребу в зовнішніх перетворювачах або додаткових елементах живлення. Також USB-інтерфейс може бути використаний для живлення пристрою

Реалізований прототип було зібрано у захищений корпус із доступом до сенсорного дисплея та роз'ємів розширення. Температурні сенсори DS18B20 були підключені до одного з виведених портів GPIO через чотириконтактний роз'єм типу SH-1.0-4PWB.

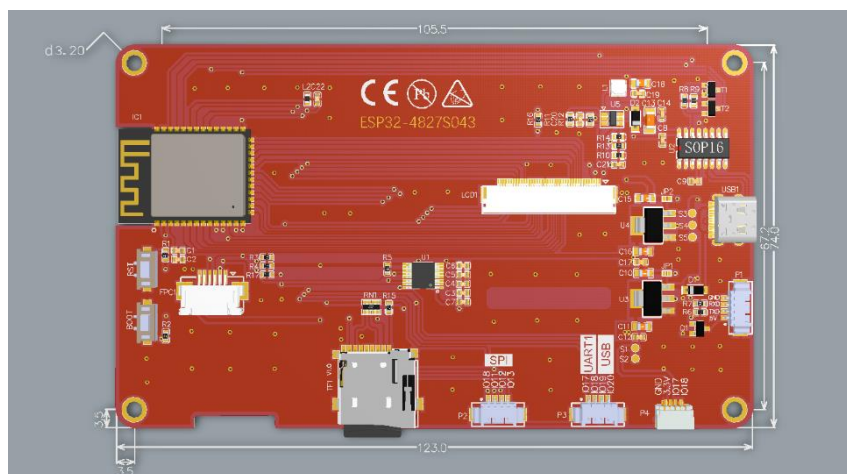


Рис. 3.10 – Схема плати готового пристрою

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42



Рис. 3.11 – Плата готового пристрою



Рис. 3.12 – Передня сторона готового пристрою

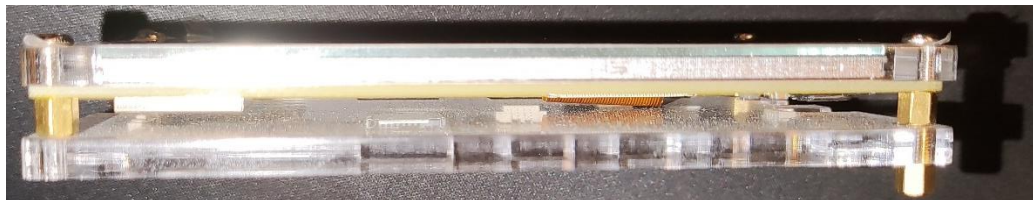


Рис. 3.13 – Бокова сторона готового пристрою

Зм.	Арк.	№ докум.	Підпис	Дата

Реалізований пристрій повністю відповідає технічному завданню, дозволяє виконувати всі необхідні функції, і є прикладом раціонального поєднання готових апаратних рішень з кастомізованим програмним забезпеченням у системах розумного дому.

3.2. Інтеграція пристроя у HomeAssistant

Інтеграція розробленого пристрою на базі ESP32 у систему Home Assistant є ключовим етапом, що забезпечує повноцінну взаємодію апаратного модуля з екосистемою розумного дому. Завдяки тісній інтеграції фреймворку ESPHome з Home Assistant, увесь процес налаштування, додавання пристрою до мережі та управління ним можна здійснити без використання додаткових сторонніх програм, що значно спрощує впровадження навіть для користувачів без технічної підготовки. Через особливості роботи ESPHome інтеграція пристрою в систему виконується першою - і лише після цього відбувається процес програмування.

Першим етапом інтеграції є встановлення додатку ESPHome безпосередньо в середовищі Home Assistant. Це здійснюється через меню «Add-on Store» в розділі налаштувань. Після встановлення компоненту ESPHome, відкривається вебінтерфейс, який дозволяє створити та керувати конфігураціями пристроїв у зручному графічному середовищі.

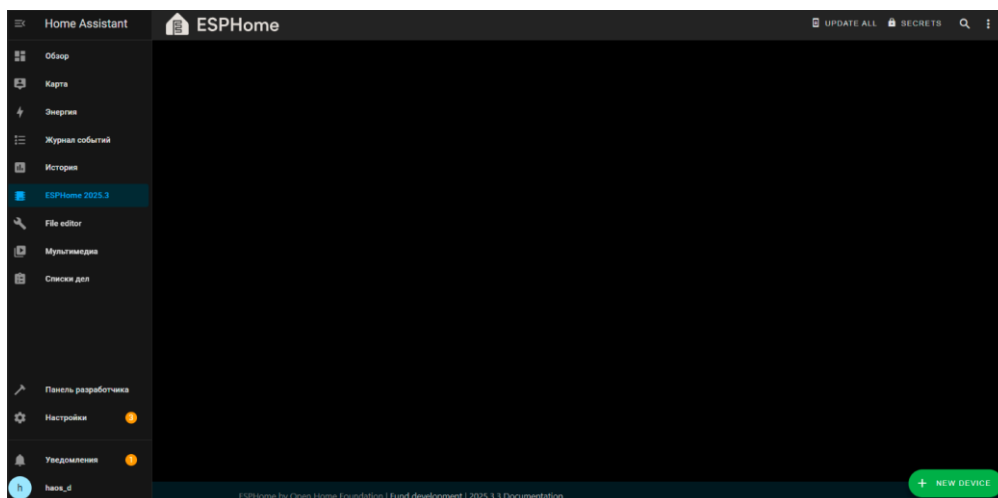


Рис. 3.14 – Панель керування ESPHome

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

Перед початком прошивки пристрою, необхідно фізично підключити модуль ESP32 до комп'ютера за допомогою кабелю USB-C. Це підключення необхідне для початкового запису прошивки, оскільки лише після першого встановлення ESPHome-прошивки пристрій зможе спілкуватися з Home Assistant по Wi-Fi. У вебінтерфейсі ESPHome створюється новий проєкт, де вказується назва пристрою, тип плати, а також мережеві параметри (SSID, пароль до Wi-Fi та ключ API).

Після завершення конфігурації ESPHome автоматично згенерує прошивку у форматі .bin. Цей файл компілюється прямо в браузері або на сервері Home Assistant, після чого користувач отримує можливість завантажити прошивку на комп'ютер. Далі, за допомогою стандартних інструментів, прошивка записується на ESP32 через USB-порт.

Після першого завантаження пристрою та його підключення до Wi-Fi, Home Assistant автоматично виявляє новий пристрій ESPHome у локальній мережі. На панелі керування з'являється повідомлення про нову інтеграцію, і користувач має змогу одним кліком підтвердити додавання модуля до системи. Пристрій одразу стає доступним з усіма визначеними в YAML-конфігурації сутностей, а також з можливістю керування OTA-оновленнями без повторного фізичного доступу до пристрою.

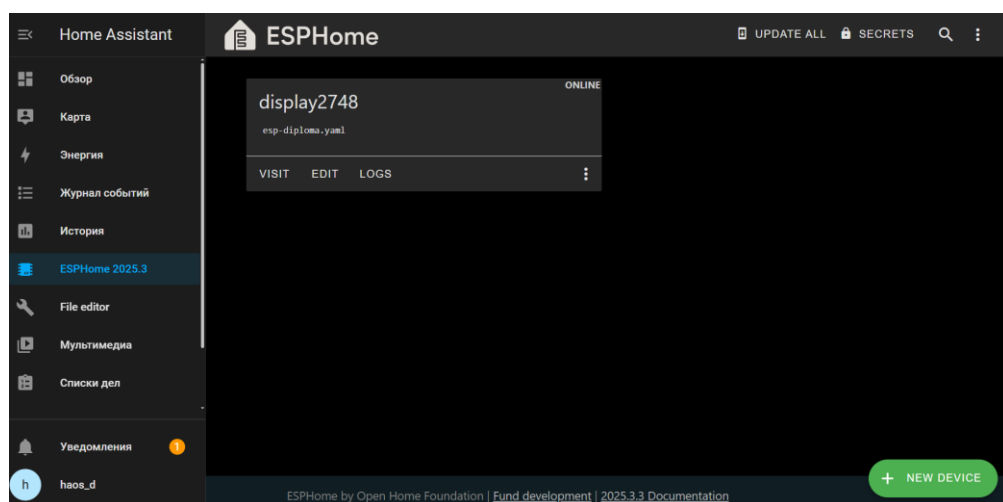


Рис. 3.15 – Інтегрований пристрій в панелі керування ESPHome

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

У межах проєкту розроблений модуль передає в Home Assistant значення температурних сенсорів, а також приймає зворотні команди для керування розетками, освітленням, вентилятором інвертора тощо.

Також в межах інтеграції автоматично синхронізуються такі параметри, як час, дата, поточні погодні умови, які зчитуються Home Assistant із зовнішніх джерел (наприклад, через інтеграцію OpenWeatherMap) і передаються на дисплей пристрою для відображення в інтерфейсі користувача.

У результаті реалізованої інтеграції, пристрій повністю синхронізований із системою автоматизації Home Assistant, що дозволяє створювати сценарії, автоматизації, графіки, історію та здійснювати централізоване керування на одному інтерфейсі. Вся взаємодія відбувається локально без участі зовнішніх серверів, що гарантує надійність, безпеку та швидкодію системи.

3.3. Програмування пристрою на базі ESP32

Програмна логіка пристрою розроблена у середовищі ESPHome за допомогою конфігураційного YAML-файлу, який забезпечує повний контроль над функціональністю апаратного комплексу. Всі програмні вимоги, передбачені технічним завданням, були реалізовані в повному обсязі.

Початкова частина конфігурації містить базові налаштування плати ESP32-S3, яка підтримує розширену оперативну пам'ять (PSRAM), необхідну для роботи графічного інтерфейсу на базі бібліотеки LVGL:

```
esp32:
  board: esp32-s3-devkitc-1
  flash_size: 8MB
  framework:
    type: esp-idf
  psram:
    mode: octal
    speed: 80MHz
```

До важливих складових належать мережеві налаштування (Wi-Fi), шифроване з'єднання з Home Assistant API, OTA-оновлення, а також реалізація модуля часу з підтримкою синхронізації через SNTP-сервери:

```
wifi:
  ssid: !secret wifi_ssid
  password: !secret wifi_password

api:
  encryption:
    key: !secret api_key

time:
  - platform: sntp
    id: sntp_time
    timezone: "Europe/Kyiv"
    servers:
      - 0.pool.ntp.org
      - 1.pool.ntp.org
      - 2.pool.ntp.org
```

У конфігурації реалізовано підключення сенсорів та бінарних сенсорів через платформу homeassistant, що дозволяє працювати як із внутрішніми температурними датчиками (через 1-Wire), так і з Zigbee-пристроями та іншими елементами інфраструктури розумного дому.

Для графічного інтерфейсу критично важливим є правильне налаштування бібліотеки LVGL. В проєкті визначено обробку дисплея та сенсорного керування, а також логіку очікування:

```
lvgl:
  displays:
    - my_display
  touchscreens:
    - my_touch
  on_idle:
    timeout: ${idle_time}
    then:
      - if:
          condition:
            lambda: 'return (id(setting_screen_off_idle));'
          then:
            - lvgl.pause:
```

Програмна логіка значною мірою реалізована через скрипти (script), які відповідають за різні аспекти роботи інтерфейсу. Скрипт `restore_data` використовується для відновлення параметрів після перезапуску пристрою. Окремі скрипти `show_page_1`, `show_page_2` тощо - реалізують логіку зміни сторінок інтерфейсу

Кожна сторінка інтерфейсу має власну логіку оновлення. Наприклад, скрипт `update_page_1` оновлює дані по генерації сонячних панелей, зовнішньому споживанню, заряду акумуляторів, а також загальне споживання будинку

Інші важливі скрипти:

- `update_time` - оновлює час і дату;
- `update_time_dots` - реалізує анімацію миготіння “:” на головному годиннику;
- `update_weather` - оновлює прогноз погоди, піктограму, температуру, вологість, хмарність;
- `update_temp` - опрацьовує температурні дані з усіх сенсорів та реалізує безпекову логіку (газовий котел, морозильна камера);
- `update_slide_page` обробляє логіку автоматичної зміни сторінок у режимі очікування.

Програмна логіка побудована на модульному підході з використанням найкращих практик інтеграції із Home Assistant. Всі функціональні вимоги успішно реалізовано - пристрій підтримує гнучке керування, візуалізацію стану розумного дому, моніторинг енергоспоживання, захист екрана, автоматичну зміну інтерфейсу та віддалене оновлення. Уся логіка відбувається локально без залежності від хмарних сервісів, що значно підвищує надійність і безпеку розгорнутої системи.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

3.4. Розробка графічного інтерфейсу користувача

Графічний інтерфейс розробленого пристрою реалізовано за допомогою бібліотеки LVGL (Light and Versatile Graphics Library), яка є потужним інструментом для створення динамічних та адаптивних GUI на вбудованих платформах. Уся логіка інтерфейсу описана безпосередньо у YAML-файлі ESPHome, що дозволило досягти високого рівня інтеграції з апаратною та програмною частинами пристрою.

Інтерфейс пристрою реалізовано у вигляді п'яти основних сторінок, кожна з яких виконує певну інформаційну або керуючу функцію. Реалізовано фіксовану бокову панель, що розташована праворуч і виконує дві основні функції: дозволяє перемикатися між п'ятьма основними сторінками інтерфейсу, а також відображає статус підключення до Wi-Fi і сервера Home Assistant. Індикатори змінюють колір залежно від наявності або втрати з'єднання, що забезпечує користувача оперативною інформацією про стан мережевої доступності. Перемикання між сторінками здійснюється як вручну (через натискання відповідних елементів), так і автоматично у режимі очікування за допомогою логіки скриптів `update_slide_page` і `show_page`

На першій сторінці відображаються ключові параметри енергосистеми розумного будинку. Тут користувач може бачити:

- поточну потужність генерації сонячних панелей (в Вт) та добову генерацію (в кВт·год);
- споживання з зовнішньої мережі (в Вт) та денне споживання (в кВт·год);
- поточне використання енергії з акумуляторів та рівень їх заряду;
- загальне споживання будинку поточне та денне.

Ці дані надходять у реальному часі з інвертора DEYE через Home Assistant.

Для оновлення інформації використовується скрипт `update_page_1`.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докum.	Підпис	Дата		49

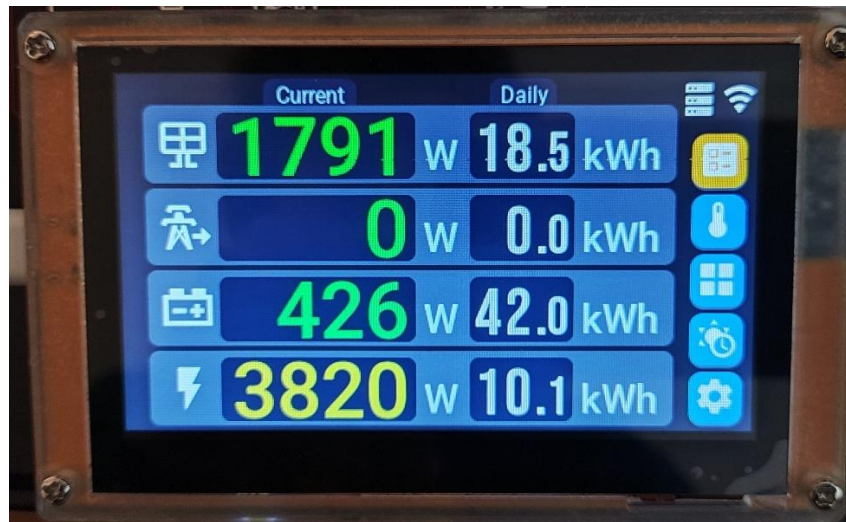


Рис. 3.15 – Вигляд першої сторінки

На другій сторінці реалізовано відображення температур з восьми термодатчиків, підключених безпосередньо до пристрою. Відображаються значення таких зон:

- кімната;
- бойлер №1;
- бойлер №2;
- проточна вода;
- морозильна камера;
- повітровід газового котла;
- вода на вході в газовий котел;
- вода на виході з газового котла.

Кожен сенсор має свій підпис. Дані оновлюються автоматично скриптом `update_temp`, який також включає логіку безпеки: якщо температура газового котла не підвищується при відкритому клапані - видається аварійне попередження. Аналогічно, при підвищенні температури в морозилці вище -5°C .

					ІАЛЦ.467200.003 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		



Рис. 3.16 – Вигляд другої сторінки

Третя сторінка - це панель керування 11 розетками та пристроями, які підключені до системи розумного дому через Zigbee. Користувач має змогу вмикати або вимикати:

- бойлери;
- насоси конвекції системи опалення;
- електрокран подачі газу;
- освітлення у підвалі, погребі, біля сараю, входу на 2 поверх;
- вентилятор охолодження інвертора;
- морозильну камеру.

Кожна кнопка має свій індикатор стану та миттєво оновлюється при натисканні. Керування реалізовано через Home Assistant-перемикачі типу `homeassistant.switch`.

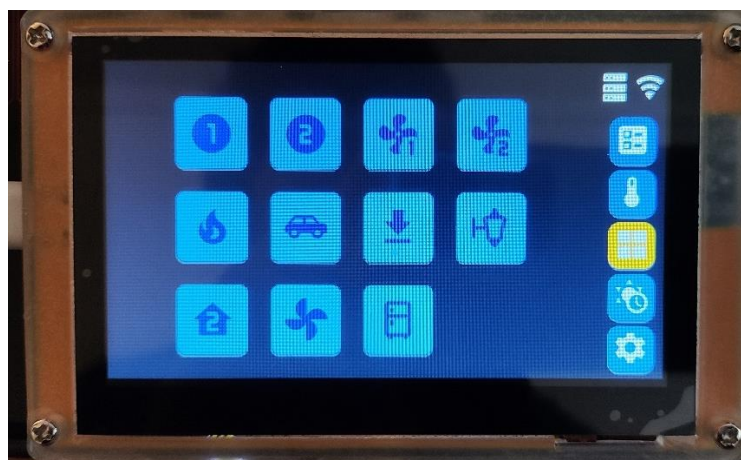


Рис. 3.17 – Вигляд третьої сторінки

Четверта сторінка має загальний інформативний характер. Тут відображається поточний час, день тижня, дата та місяць, а також прогноз погоди, включаючи:

- температуру ззовні;
- піктограму погоди (ясно, хмарно, дощ тощо);
- вологість повітря;
- рівень хмарності у відсотках.

Інформація надходить до пристрою з Home Assistant через інтеграцію з OpenWeatherMap. Для відображення використовується скрипт `update_weather`, а оновлення часу та анімація миготіння двокрапки в годинах - `update_time` та `update_time_dots`.



Рис. 3.18 – Вигляд четвертої сторінки

П'ята сторінка інтерфейсу дозволяє користувачу налаштувати основні параметри пристрою, зокрема:

- змінити яскравість підсвітки;
- увімкнути або вимкнути функцію анти-вигорання (анімоване тренування пікселів у нічний час);
- активувати або деактивувати автоматичне вимкнення екрана під час простою;
- вмикати слайд-шоу сторінок при бездіяльності.

На сторінці також виведено інформацію про автора пристрою. Всі параметри зберігаються локально, а їх значення відновлюються скриптом `restore_data` після перезавантаження.

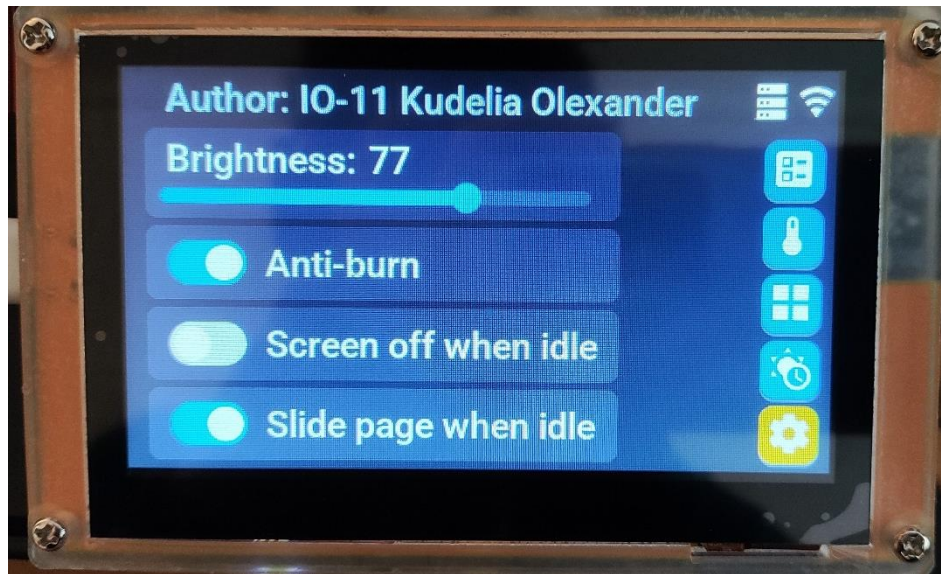


Рис. 3.19 – Вигляд п'ятої сторінки

Одже, графічний інтерфейс було реалізовано відповідно до вимог програмної частини. Завдяки використанню LVGL досягнуто плавну роботу сторінок, адаптивне відображення інформації та інтеграцію з програмною логікою пристрою, що базується на реальних даних системи розумного дому. Всі сценарії використання були охоплені, а інтерфейс протестовано на стабільність, коректність та відгук користувача.

ВИСНОВОК ДО РОЗДІЛУ 3

У цьому розділі було реалізовано повноцінний програмно-апаратний комплекс, що інтегрується в екосистему розумного дому Home Assistant. Було детально розглянуто апаратну частину та розроблено програмне забезпечення пристрою з використанням сучасного високофункціонального модуля ESP32-4827S043, який поєднує в собі відносно високопродуктивний процесор, кольоровий дисплей, сенсорне керування та інтерфейси введення/виведення в одному компактному рішенні.

У межах реалізації було продемонстровано електричні схеми підключення, зібрано функціональний прототип, а також забезпечено підключення температурних сенсорів через роз'єми розширення. Інтеграція пристрою з Home Assistant виконана за допомогою ESPHome, що дозволило реалізувати автоматичне виявлення, обмін даними та віддалене оновлення прошивки (OTA).

Програмна частина охоплює логіку обробки сенсорів, перевірку умов безпеки, оновлення інтерфейсу, синхронізацію часу, а також керування зовнішніми пристроями через API Home Assistant. Усі функціональні вимоги до логіки роботи пристрою були реалізовані відповідно до технічного завдання.

Окрему увагу приділено графічному інтерфейсу на базі LVGL, який складається з п'яти логічно розділених сторінок з відображенням енергетичних, кліматичних і погодних даних, а також налаштувань пристрою. Завдяки вбудованій боковій панелі інтерфейс є інтуїтивним, а повна реалізація у YAML забезпечує легку адаптацію та розширення проєкту в майбутньому.

Таким чином, у результаті реалізації всіх підсистем було отримано завершений, працездатний пристрій, здатний взаємодіяти з іншими компонентами розумного дому, надавати необхідну інформацію в зручному форматі та виконувати керуючі дії, що повністю відповідає поставленій меті дипломного проєкту.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

РОЗДІЛ 4

РЕАЛІЗАЦІЯ ПРИСТРОЯ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1. Перевірка стабільності роботи пристроїв у різних умовах

Для перевірки коректності функціонування розробленого пристрою було проведено тестування в різних типових і граничних експлуатаційних ситуаціях. Особливу увагу було приділено реакції графічного інтерфейсу на зміну вхідних параметрів у режимі реального часу, а також перевірці роботи механізмів безпеки та візуалізації критичних станів.

Сценарії роботи на першій сторінці інтерфейсу (енергетика будинку):

1. Генерація сонячних панелей дорівнює нулю

У цьому випадку текстове значення потужності генерації відображається білим кольором, що сигналізує про відсутність виробництва енергії.



Рис. 4.1 – Вигляд першої сторінки при відсутності виробництва енергії

2. Генерація сонячних панелей більша за нуль

Коли система починає виробляти енергію, текст змінює колір на зелений, що відображає активне постачання живлення.



Рис. 4.2 – Вигляд першої сторінки при виробництві енергії

3. Споживання з зовнішньої мережі (міської) дорівнює нулю

У такому випадку значення також відображається зеленим кольором, що є позитивним індикатором енергонезалежності.



Рис. 4.3 – Вигляд першої сторінки при відсутності споживання

4. Розряд акумулятора (споживання з АКБ)

Колір тексту потужності відображається червоним, що сигналізує про активне споживання запасеної енергії.



Рис. 4.4 – Вигляд першої сторінки при розряді акумулятора

5. Заряд акумулятора

Значення заряджання виводиться зеленим кольором, що вказує на відновлення енергозапасів.



Рис. 4.5 – Вигляд першої сторінки при заряді акумулятора

6. Загальне споживання будинку менше 1900 Вт

Текст відображається білим кольором, що означає помірне, безпечне навантаження.

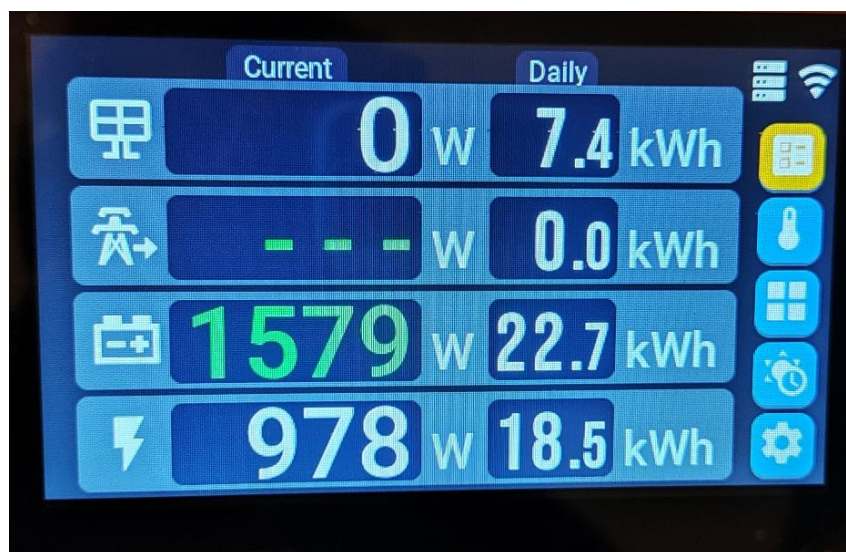


Рис. 4.6 – Вигляд першої сторінки при споживання будинку менше 1900 Вт

7. Споживання від 1900 Вт до 4900 Вт

Вивід здійснюється жовтим кольором, що вказує на середнє, допустиме навантаження.



Рис. 4.7 – Вигляд першої сторінки при споживання будинку від 1900 Вт до 4900 Вт

8. Споживання будинку понад 4900 Вт

У цьому критичному стані колір тексту змінюється на червоний, що попереджає користувача про наближення до граничної потужності інвертора.



Рис. 4.7 – Вигляд першої сторінки при споживання будинку понад 4900 Вт

Сценарії роботи на першій сторінці інтерфейсу (енергетика будинку):

1. Немає з'єднання з Zigbee-розеткою

Рамка відповідної картки стає червоною, що сигналізує про втрату керуваності пристрою.

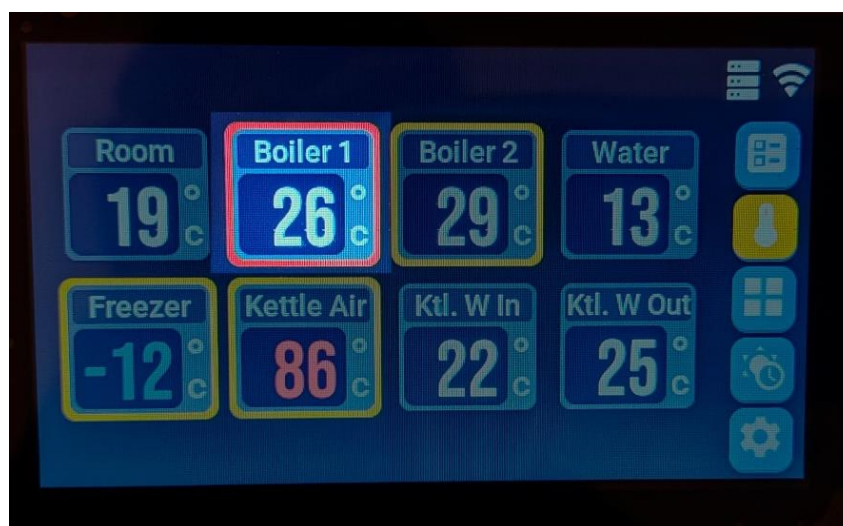


Рис. 4.9 – Приклад вигляду другої сторінки при втраті зв'язку

2. Є з'єднання, але пристрій неактивний (вимкнений)

Рамка змінюється на жовту, що вказує на наявність підключення, але відсутність активності.

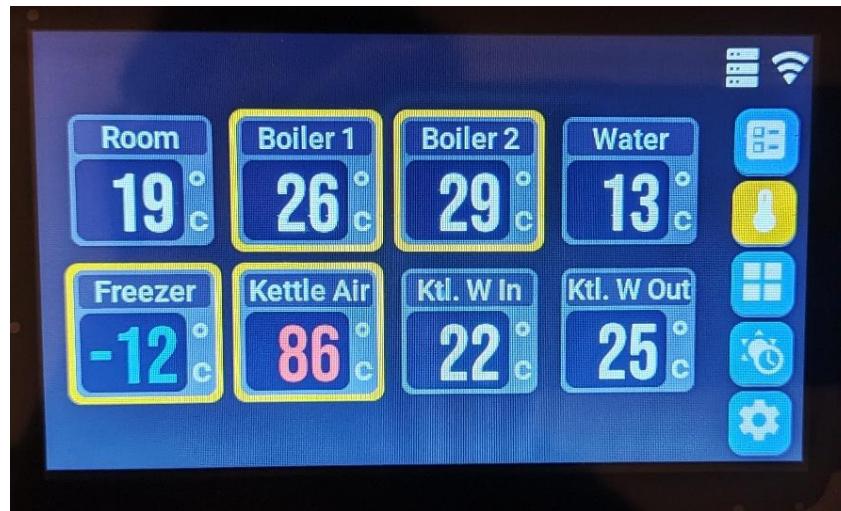


Рис. 4.10 – Приклад вигляду другої сторінки при наявності зв'язку, але вимкненому пристрої

3. Є з'єднання і пристрій увімкнений

У такому випадку рамка стає зеленою, що демонструє активну роботу підконтрольного пристрою.

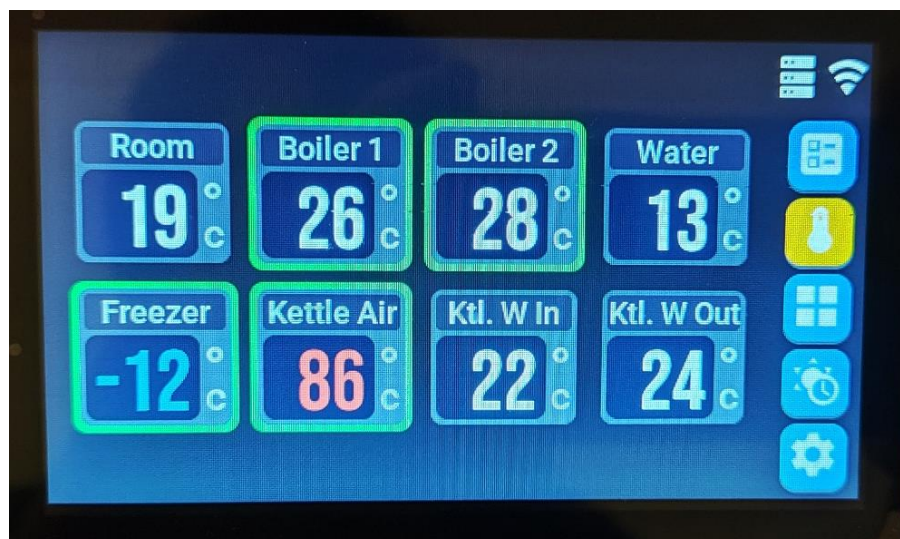


Рис. 4.11 – Приклад вигляду другої сторінки при наявності зв'язку та увімкненому пристрої

4. Температура у морозильній камері перевищує $-7.5\text{ }^{\circ}\text{C}$

Виводиться помилка, що свідчить про загрозу псування продуктів.

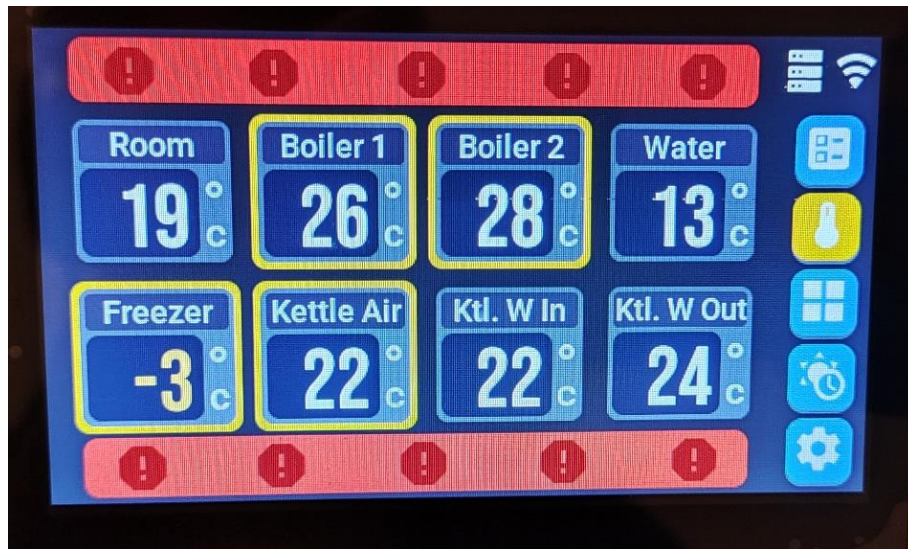


Рис. 4.12 – Приклад вигляду другої сторінки при наявності помилки у зв'язку з температурою морозильної камери

5. Газовий котел увімкнено, минуло 30–60 секунд, а температура $<65\text{ }^{\circ}\text{C}$

Система видає попередження про можливу несправність розпалу.

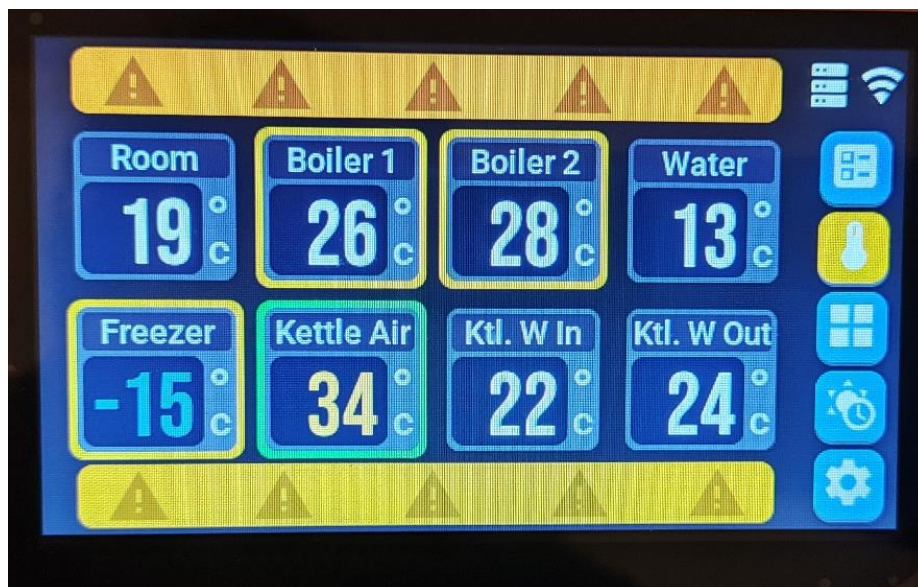


Рис. 4.13 – Приклад вигляду другої сторінки при наявності попередження у зв'язку з котлом

6. Газовий котел працює >60 секунд, температура <75 °C

Виводиться критична помилка, яка потребує втручання користувача.



Рис. 4.14 – Приклад вигляду другої сторінки при наявності критичної помилки у зв'язку з котлом

4.2. Аналіз продуктивності та енергоспоживання

Для оцінки енергоспоживання розробленого пристрою було проведено серію вимірювань споживаного струму та напруги за різних умов роботи, зокрема при зміні рівня яскравості екрана та в режимі автоматичного вимкнення підсвітки під час простою. Джерелом живлення слугував стабілізований блок живлення з напругою 5 В. Також додатково змодельовано сценарій, коли пристрій живиться від мобільного акумулятора (powerbank), що особливо актуально в умовах частих відключень електроенергії.

Таблиця 4.1 – Таблиця енергоспоживання в залежності від ситуації

№	Режим роботи	Струм (А)	Потужність (Вт)
1	Яскравість 100 %	0,20	1,01
2	Яскравість 80%	0,14	0,707
3	Яскравість 50 %	0,08	0,404
4	Яскравість 30 %	0,04	0,202
5	Яскравість 20 %	0,03	0,152

Зм.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 4.1

6	Екран вимкнено	0,02	0,101
---	----------------	------	-------

Як видно з результатів, пристрій демонструє високу енергоефективність, особливо в умовах зниженої яскравості або у стані бездіяльності. При максимальній яскравості дисплей споживає 1,01 Вт, що є порівняно малим показником навіть для вбудованих пристроїв з сенсорними екранами. У режимі вимкненої підсвітки екрана (що активується автоматично після тривалого простою) потужність падає до 0,101 Вт. Таке низьке енергоспоживання дозволяє живлення пристрою від звичайного powerbank'а, що є критично важливо в умовах перебоїв електропостачання. Завдяки цьому розроблений комплекс може функціонувати автономно протягом тривалого часу без зовнішнього живлення, зберігаючи при цьому повну функціональність моніторингу та зв'язку з Home Assistant.

Для оцінки графічної продуктивності було проведено вимірювання середньої частоти оновлення екрана (FPS) під час роботи на кожній із п'яти сторінок інтерфейсу. Також оцінено використання динамічної пам'яті системи під час роботи в штатному режимі.

Таблиця 4.2 – Таблиця продуктивності в залежності від ситуації

№	Сторінка інтерфейсу	Середній FPS
1	Головна сторінка	16
2	Температурні сенсори	15
3	Керування розетками	18
4	Годинник та погода	16
5	Сторінка налаштувань	19

Усі сторінки демонструють стабільну роботу з середнім FPS у межах 16–19 кадрів на секунду, що забезпечує плавне перемикання між елементами інтерфейсу та відсутність візуальних затримок. Завантаження оперативної пам'яті ESP32 знаходиться в діапазоні 11,3 %, а флеш-пам'яті - у діапазоні 39,4

%, що свідчить про оптимізоване використання ресурсів мікроконтролера та залишає резерв для подальшого розширення функціональності без зниження стабільності.

Таким чином, тестування підтвердило, що пристрій є не лише енергоефективним, але й здатним стабільно працювати з графічним інтерфейсом середньої складності без перевантаження апаратних ресурсів.

4.3. Аналіз економічної доцільності розробки власних пристроїв

Розробка індивідуального програмно-апаратного комплексу для системи розумного дому потребує не лише технічного обґрунтування, а й економічного аналізу доцільності такого підходу. Основне питання полягає в тому, чи вигідніше створювати пристрій самостійно, чи використовувати готові комерційні рішення.

В межах цієї роботи був розроблений і зібраний повноцінний пристрій на базі ESP32-4827S043 - це модуль із попередньо встановленим TFT-дисплеєм та емнісним сенсорним екраном. Його загальна вартість разом із усіма необхідними компонентами склала близько 700 грн.

Нижче наведено порівняльну таблицю з іншими готовими пристроями на ринку, що можуть частково виконувати аналогічні функції:

Таблиця 4.3 – Таблиця порівняння вартості пристроїв

Пристрій	Середня вартість (грн)	Особливості / Обмеження
Розроблений пристрій (ESP32 DIY)	~700-1100	Власна логіка, повна інтеграція з HA, OTA
NSPanel Pro (Sonoff)	~2900–3200	Готове рішення, обмежена кастомізація
Tuya Smart Control Panel	~2500–3100	Закритий софт, не підтримує ESPHome

Видно, що розроблений пристрій у 3-4 разів дешевший за комерційні аналоги. І головне - він максимально адаптований під конкретні потреби користувача: його функціональність не обмежується лише базовим виведенням даних чи керуванням розетками. До переваг власного пристрою належать:

- Реалізація алгоритмів безпеки, таких як контроль температури вихлопу газового котла, автоматичне виявлення несправностей чи аварійних режимів.
- Повна кастомізація інтерфейсу, включно з логікою перемикання сторінок, зміною кольорів залежно від критичних значень, а також підтримка локальної автоматизації.
- Локальна автономна робота без необхідності підключення до хмари - система повністю працює у локальній мережі.
- Можливість гнучкого розширення: додавання сенсорів, сторінок, логіки без будь-яких обмежень з боку виробника чи прошивки.

У той час як серійні пристрої проектуються як універсальні рішення “для всіх”, саморобний пристрій - це точно адаптована система “під себе”, яка реалізує саме ті функції, що дійсно необхідні в конкретному будинку. При цьому користувач повністю контролює як апаратну, так і програмну складову, що дозволяє легко доповнювати систему новими функціями, розширювати інтерфейс, підключати додаткові сенсори чи логіку автоматизації без будь-яких ліцензійних чи технічних обмежень. Такий підхід відкриває широкі можливості для майбутньої модернізації та адаптації пристрою до змін у потребах користувача або конфігурації системи розумного дому.

Тому з економічної та практичної точки зору - розробка власного пристрою виявилась не лише дешевшою, але й значно ефективнішою у плані гнучкості, кастомізації, розширюваності та довгострокової підтримки.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

ВИСНОВОК ДО РОЗДІЛУ 4

У процесі тестування було підтверджено стабільну роботу пристрою в різних сценаріях, включаючи критичні ситуації, пов'язані з температурними відхиленнями, порушенням зв'язку та навантаженням на енергосистему. Графічний інтерфейс адекватно реагує на зміни даних у режимі реального часу та інформує користувача через колірні індикатори і повідомлення.

Проведений аналіз енергоспоживання показав, що пристрій є надзвичайно енергоефективним - навіть при максимальній яскравості його потужність не перевищує 1 Вт, що дозволяє жити його від звичайного powerbank. Продуктивність графічної частини перебуває в межах 16–20 FPS, при цьому використання пам'яті в межах норми, що свідчить про ефективність реалізації програмної частини.

Економічний аналіз довів, що власна розробка значно вигідніша порівняно з готовими комерційними рішеннями. Пристрій повністю адаптовано під конкретні потреби, підтримує розширення та не має обмежень у функціональності - на відміну від серійних аналогів.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

ВИСНОВКИ

У ході виконання дипломної роботи було розроблено програмно-апаратний комплекс для інтеграції користувацьких пристроїв у систему розумного дому на базі Home Assistant. Основна мета проєкту - створити власний пристрій, який не лише відображає ключові параметри домашньої автоматизації, а й бере участь у зборі даних, контролі безпеки та керуванні енергоспоживанням - була досягнута повністю.

У першому розділі було проведено огляд існуючих рішень для автоматизації будинку та проаналізовано можливості інтеграції користувацьких пристроїв у систему Home Assistant. Особливу увагу приділено фреймворку ESPHome та мікроконтролеру ESP32 як основі для побудови гнучких, індивідуально адаптованих IoT-рішень.

У другому розділі здійснено проєктування програмно-апаратного комплексу: визначено вимоги, побудовано архітектуру, описано схеми підключення та логіку програмної частини. Було обґрунтовано вибір технологій, зокрема ESPHome, LVGL та YAML як ключових складових для інтеграції, створення інтерфейсу та реалізації логіки пристрою.

У третьому розділі реалізовано прототип пристрою: виконано складання, програмування, підключення до Home Assistant та розроблено повноцінний графічний інтерфейс з п'ятьма функціональними сторінками. Було реалізовано важливі механізми безпеки (контроль газового котла й температури морозильної камери).

У четвертому розділі проведено тестування пристрою в різних ситуаціях, здійснено вимірювання енергоспоживання та частоти оновлення графіки, а також проведено економічний аналіз доцільності розробки власного пристрою. Підтверджено, що самостійно виготовлений пристрій є дешевшим,

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

функціональнішим та краще адаптованим до індивідуальних вимог, ніж наявні комерційні аналоги.

Таким чином, розроблений програмно-апаратний комплекс повністю відповідає поставленим технічним та функціональним вимогам, демонструє стабільну роботу, низьке енергоспоживання, високу гнучкість та можливість подальшого масштабування. Він є практичним прикладом ефективної реалізації сучасного розумного дому з відкритою архітектурою.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ESPHome – Офіційна документація [Електронний ресурс] – Режим доступу: <https://esphome.io>
2. Home Assistant – Документація та інтеграції [Електронний ресурс] – Режим доступу: <https://www.home-assistant.io>
3. OpenHAB – Open Home Automation Bus [Електронний ресурс]. – Режим доступу: <https://www.openhab.org>
4. YAML Ain't Markup Language – Специфікація синтаксису [Електронний ресурс] – Режим доступу: <https://yaml.org>
5. LVGL – Light and Versatile Graphics Library [Електронний ресурс] – Режим доступу: <https://lvgl.io>
6. Espressif Systems – ESP32 Technical Reference Manual [Електронний ресурс]. – Режим доступу: <https://www.espressif.com/en/products/socs/esp32/resources>
7. DEYE Hybrid Inverter User Manual [Електронний ресурс] – Режим доступу: <https://deyinverter.com/product/hybrid-inverter>
8. DS18B20 Programmable Resolution 1-Wire Digital Thermometer [Електронний ресурс] – Режим доступу: <https://www.analog.com/en/products/ds18b20.html>
9. MQTT – OASIS Standard Protocol [Електронний ресурс] – Режим доступу: <https://mqtt.org>
10. Zigbee Alliance – Офіційна інформація про протокол Zigbee [Електронний ресурс] – Режим доступу: <https://csa-iot.org/all-solutions/zigbee>
11. ESP32 vs Arduino [Електронний ресурс] – Режим доступу: <https://www.ntchip.com/de/electronics-news/esp32-vs-arduino>
12. LVGL Examples and Documentation [Електронний ресурс] – Режим доступу: <https://docs.lvgl.io>

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

ДОДАТОК А

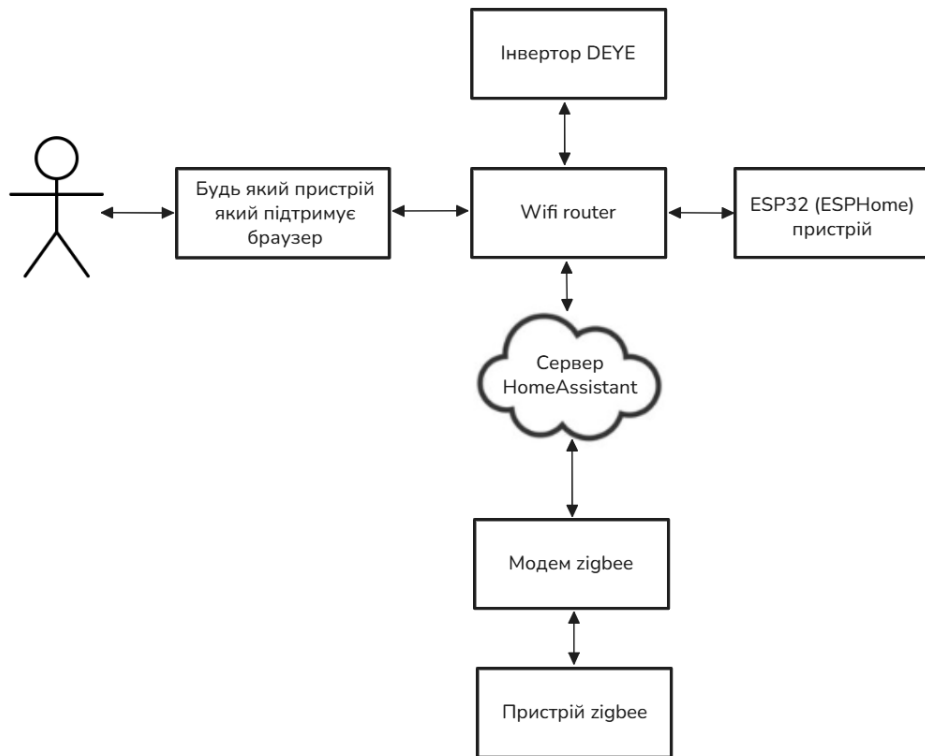
Програмно-апаратний комплекс для інтеграції користувацьких пристроїв у систему розумного будинку HomeAssistant

Структурна схема системи

ІАЛЦ.467200.004 Д1

Аркушів 1

Київ 2025



				ІАЛЦ.467200.004 Д1				
		№ докум.	Підпис	Дата				
Розробив	Куделя О.С.				Програмно-апаратний комплекс для інтеграції користувацьких пристроїв у систему розумного будинку HomeAssistant Структурна схема системи	Літ.	Аркуш	Аркушів
Перевірив	Васильєва М. Д.						1	1
Н. Контр.	Пономаренко А. М.					НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІО-11		
Затвердив								

ДОДАТОК Б

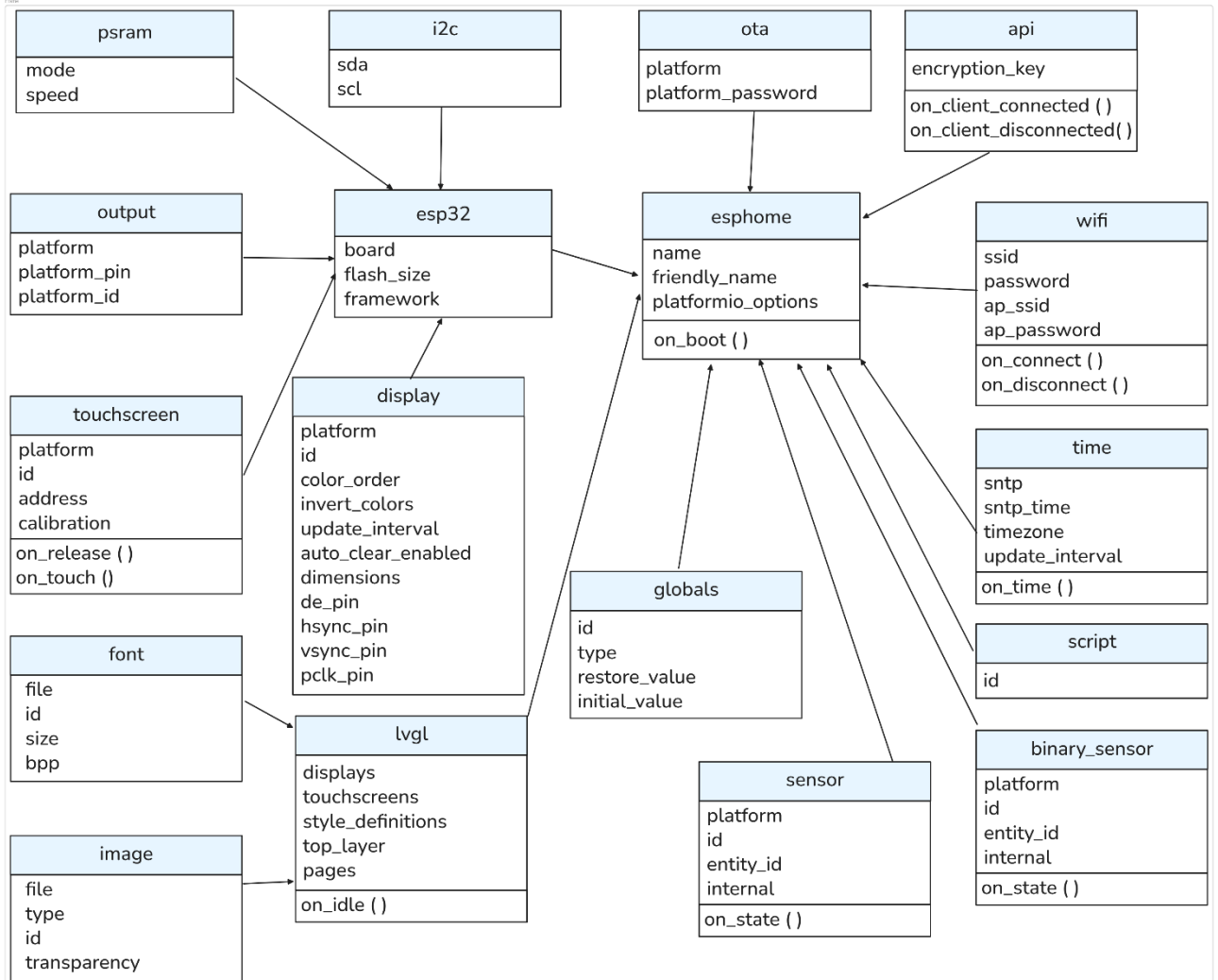
Програмно-апаратний комплекс для інтеграції користувацьких пристроїв у систему розумного будинку HomeAssistant

Функціональна схема (Діаграма класів)

ІАЛЦ.467200.004 Д2

Аркушів 1

Київ 2025



				ІАЛЦ.467200.004 Д2			
	№ докум.	Підпис	Дата				
Розробив	Куделя О.С.			Програмно-апаратний комплекс для інтеграції користувацьких пристроїв у систему розумного будинку HomeAssistant Функціональна схема (Діаграма класів)	Літ.	Аркуш	Аркушів
Перевірив	Васильєва М. Д.					1	1
Н. Контр.	Пономаренко А. М.				НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, Ю-11		
Затвердив							

ДОДАТОК В

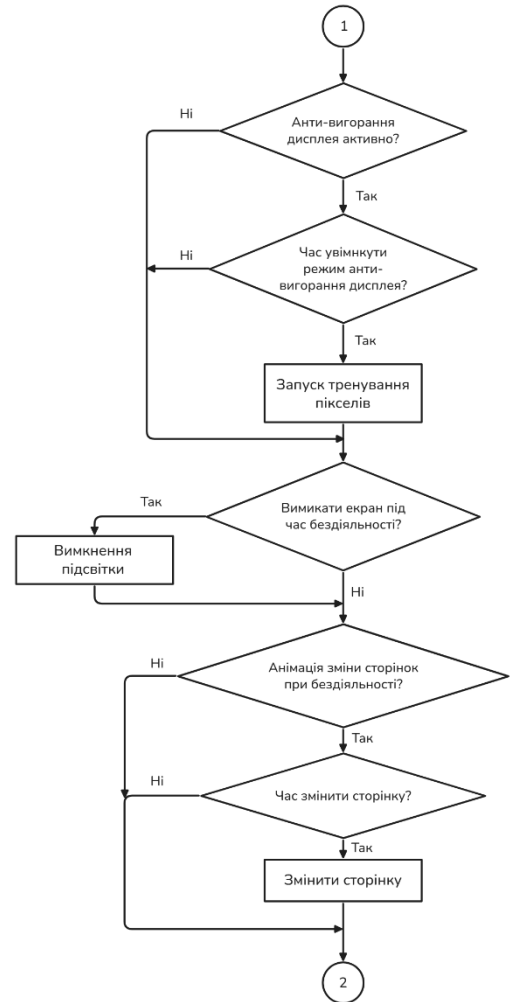
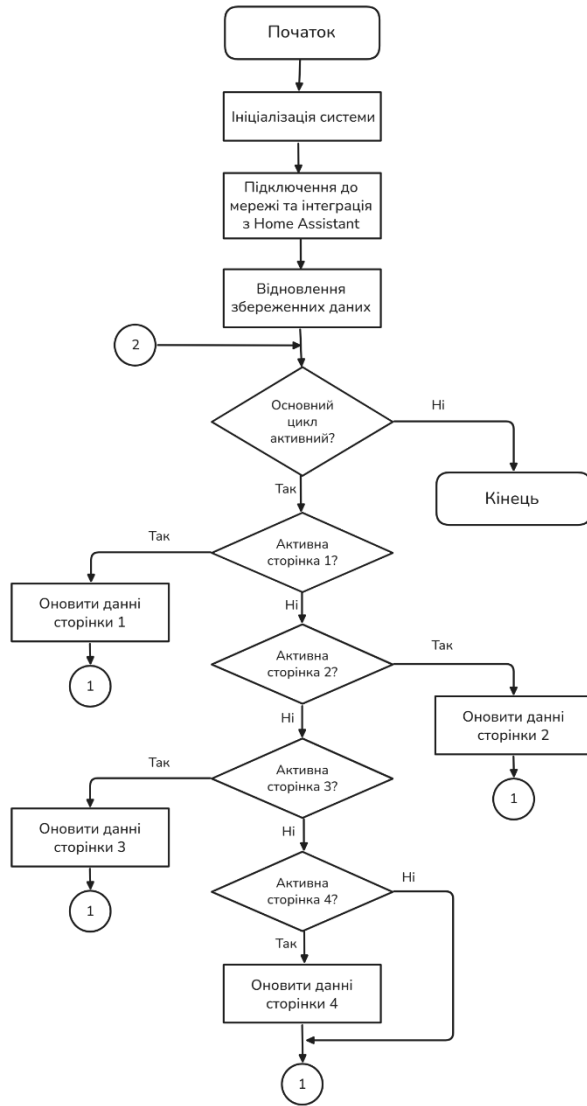
Програмно-апаратний комплекс для інтеграції користувацьких пристроїв у систему розумного будинку HomeAssistant

Алгоритм дій програми (Принципова схема)

ІАЛЦ.467200.004 ДЗ

Аркушів 1

Київ 2025



				ІАЛЦ.467200.004 ДЗ				
		№ докум.	Підпис	Дата				
Розробив	Куделя О.С.				Програмно-апаратний комплекс для інтеграції користувацьких пристроїв у систему розумного будинку HomeAssistant Алгоритм дій програми (Принципова схема)	Літ.	Аркуш	Аркушів
Перевірив	Васильєва М. Д.						1	1
Н. Контр.	Пономаренко А. М.					НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІО-11		
Затвердив								

ДОДАТОК Г

Програмно-апаратний комплекс для інтеграції користувацьких пристроїв у систему розумного будинку HomeAssistant

Лістинг програмного коду

ІАЛЦ.467200.004 Д4

Аркушів 58

Київ 2025

substitutions:

x1: "35"
 x2: "225"
 x3: "270"
 x4: "1"

```
# Icon
# Status
icon_wifi_on: "\U000F05A9"
icon_wifi_off: "\U000F05AA"
icon_server_on: "\U000F048B"
icon_server_off: "\U000F048F"
# Nav
icon_nav_page_1: "\U000F09C9" # ballot
icon_nav_page_2: "\U000F050F" # thermometer
icon_nav_page_3: "\U000F0570" # view-grid
icon_nav_page_4: "\U000F1A77" # sun-clock
icon_nav_page_5: "\U000F0493" # gear
# Page 1
icon_solar_panel: "\U000F0D9B"
icon_tower: "\U000F192C"
icon_battery: "\U000F010C"
icon_flash: "\U000F0241"
# Weather
icon_weather_cloud: "\U000F0161" # cloud
icon_weather_wet: "\U000F1806" # drop
# Page 2
icon_warning: "\U000F0026" # warning
icon_error: "\U000F0029" # error
# Page 3
icon_kettle: "\U000F0238" # mdi-fire
icon_motor_1: "\U000F1472" # mdi-fan-speed-1
icon_motor_2: "\U000F1473" # mdi-fan-speed-2
icon_boiler_1: "\U000F0CA0" # mdi-numeric-1-circle
icon_boiler_2: "\U000F0CA2" # mdi-numeric-2-circle
icon_door_2: "\U000F0D81" # mdi-home-floor-2
icon_garage: "\U000F07A9" # mdi-car-hatchback
icon_cellar: "\U000F01DA" # mdi-download
icon_shed: "\U000F1020" # mdi-coach-lamp
icon_deye_cooling: "\U000F0AD5" # mdi-pinwheel
icon_fridge: "\U000F028F" # fridge-outline

# Color
main_bg_color: "0x26272c"
status_ok_color: "0xffffffff"
status_err_color: "0xf73131"
nav_but_bg_color: "0x197fe6"
nav_but_pressed_bg_color: "0xf7dd6"
nav_but_actv_bg_color: "0xfc19"
power_ok_color: "0x36ff4d"
power_warning_color: "0xff3d3d"
power_norm_color: "0xffe436"
clock_day_norm_color: "0xd7d8d9" # 0xd7d8d9
clock_day_holi_color: "0xfcc432" # 0xfc19 # 0xffe436
temp_cold_color: "0x3483eb"
temp_warm_color: "0xffbe3d"
temp_hot_color: "0xff4a3d"
```

				ІАЛЦ.467200.004 Д4				
		№ докум.	Підпис	Дата				
Розробив	Куделя О.С.				Програмно-апаратний комплекс для інтеграції користувацьких пристроїв у систему розумного будинку HomeAssistant Лістинг програмного коду	Літ.	Аркуш	Аркушів
Перевірив	Васильєва М. Д.						1	58
Н. Контр.	Пономаренко А. М.					НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, Ю-11		
Затвердив								

```

temp_warning_color: "0xff3d3d"
temp_status_on_color: "0x13d440"
temp_status_off_color: "0xe8b315"
temp_status_err_color: "0xefc3426"

# Config
idle_time: "180s"
slide_page_show_time : "120"

esphome:
  name: display2748
  friendly_name: display2748

  platformio_options:
    board_build.flash_mode: dio

  on_boot:
    priority: 600.0
    then:
      - lambda: |-
          id(led_tft).set_level(1); // id(t1)/100.0);
      - delay: 3s
      - script.execute: restore_data

esp32:
  board: esp32-s3-devkitc-1
  flash_size: 8MB
  framework:
    type: esp-idf
    version: 4.4.8
    platform_version: 5.4.0

psram:
  mode: octal
  speed: 80MHz

# Enable Home Assistant API
api:
  encryption:

    key: !secret api_key
  on_client_connected:
    - if:
        condition:
          lambda: 'return (0 == client_info.find("Home Assistant "));'
        then:
          - lambda: |-
              lv_label_set_text( id(label_api_status), "${icon_server_on}" );
              lv_obj_set_style_text_color(id(label_api_status),
lv_color_hex(${status_ok_color}), LV_PART_MAIN );

  on_client_disconnected:
    - if:
        condition:
          lambda: 'return (0 == client_info.find("Home Assistant "));'
        then:
          - lambda: |-
              lv_label_set_text( id(label_api_status), "${icon_server_off}" );

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		2

```

        lv_obj_set_style_text_color(id(label_api_status),
lv_color_hex(${status_err_color}), LV_PART_MAIN );

ota:
- platform: !secret ota_platform
password: !secret ota_password

wifi:
ssid: !secret wifi_ssid
password: !secret wifi_password

# Enable fallback hotspot (captive portal) in case wifi connection fails
ap:
ssid: !secret api_ssid
password: !secret api_password
on_connect:
- lambda: |-
    lv_label_set_text( id(label_wifi_status), "${icon_wifi_on}" );
    lv_obj_set_style_text_color(id(label_wifi_status), lv_color_hex(${status_ok_color}),
LV_PART_MAIN );
on_disconnect:
- lambda: |-
    lv_label_set_text( id(label_wifi_status), "${icon_wifi_off}" );
    lv_obj_set_style_text_color(id(label_wifi_status),
lv_color_hex(${status_err_color}), LV_PART_MAIN );

captive_portal:

time:
- platform: sntp
id: sntp_time
timezone: "Europe/Kyiv"
servers:
- 0.pool.ntp.org
- 1.pool.ntp.org
- 2.pool.ntp.org
update_interval: 120s
on_time:
- hours: 3, 4, 5, 6
minutes: 5 # 5
seconds: 0
then:
- lambda: |-
    if (id(setting_antiburn)) {
        id(switch_antiburn).turn_on();
    }
- hours: 3, 4, 5, 6
minutes: 40
seconds: 0
then:
- lambda: |-
    if (id(setting_antiburn)) {
        id(switch_antiburn).turn_off();
    }

# Define a PWM output on the ESP32
output:
- platform: ledc

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

```

pin: 2
id: led_tft
i2c:
sda: 19
scl: 20

touchscreen:
platform: gt911
id: my_touch
address: 0x5D
calibration:
x_min: 0 # L:200 R:200
x_max: 480 # L:3830 R:3830
y_min: 0 # L:170 R:170
y_max: 272 # L:3780 R:3780

on_release:
then:
- if:
condition: lvgl.is_paused
then:
- lvgl.resume:
- lvgl.widget.redraw:
- lambda: |-
if (id(antiburn_work)) {
id(antiburn_work) = false;
id(led_tft).set_level(id(setting_brightness)/100.0);
}

if (id(screen_off_idle)) {
id(screen_off_idle) = false;
id(led_tft).set_level(id(setting_brightness)/100.0);
}

if (id(slide_page_mode)) {
id(slide_page_mode) = false;
}

on_touch:
- lambda: |-
ESP_LOGI("cal", "x=%d, y=%d, x_raw=%d, y_raw=%0d",
touch.x,
touch.y,
touch.x_raw,
touch.y_raw
);
#~~~~~

display:
- platform: rpi_dpi_rgb
id: my_display
color_order: RGB
invert_colors: True
update_interval: never
auto_clear_enabled: false # takes 2.8 seconds to clear the display
dimensions:
width: 480
height: 272
de_pin: 40
hsync_pin: 39
vsync_pin: 41
pclk_pin: 42
hsync_back_porch: 39 #39
hsync_front_porch: 50 #50
hsync_pulse_width: 4 #4

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

vsync_back_porch: 9 #9
vsync_front_porch: 8 #8
vsync_pulse_width: 4 #4
pclk_frequency: 9MHz

data_pins:

red:

- 45 #r1
- 48 #r2
- 47 #r3
- 21 #r4
- 14 #r5

green:

- 5 #g0
- 6 #g1
- 7 #g2
- 15 #g3
- 16 #g4
- 4 #g5

blue:

- 8 #b1
- 3 #b2
- 46 #b3
- 9 #b4
- 1 #b5

#~~~~~

font:

- file: "fonts/materialdesignicons-webfont.ttf"
id: icon_status
size: 25
bpp: 4
glyphs:
 - "\${icon_wifi_on}"
 - "\${icon_wifi_off}"
 - "\${icon_server_on}"
 - "\${icon_server_off}"
- file: "fonts/materialdesignicons-webfont.ttf"
id: icon_nav
size: 32
bpp: 4
glyphs:
 - "\${icon_nav_page_1}"
 - "\${icon_nav_page_2}"
 - "\${icon_nav_page_3}"
 - "\${icon_nav_page_4}"
 - "\${icon_nav_page_5}"
- file: "fonts/materialdesignicons-webfont.ttf"
id: icon_page_1
size: 40
bpp: 4
glyphs:
 - "\${icon_solar_panel}"
 - "\${icon_tower}"
 - "\${icon_battery}"
 - "\${icon_flash}"
- file: "fonts/materialdesignicons-webfont.ttf"
id: icon_weather
size: 28

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

```

bpp: 4
glyphs:
  - "${icon_weather_cloud}"
  - "${icon_weather_wet}"
  - "${icon_battery}"

- file: "fonts/materialdesignicons-webfont.ttf"
  id: icon_temp
  size: 35
  bpp: 4
  glyphs:
    - "${icon_warning}"
    - "${icon_error}"

- file: "fonts/materialdesignicons-webfont.ttf"
  id: icon_page_3
  size: 40
  bpp: 4
  glyphs:
    - "${icon_kettle}"
    - "${icon_motor_1}"
    - "${icon_motor_2}"
    - "${icon_boiler_1}"
    - "${icon_boiler_2}"
    - "${icon_door_2}"
    - "${icon_garage}"
    - "${icon_cellar}"
    - "${icon_shed}"
    - "${icon_deye_cooling}"
    - "${icon_fridge}"

# Page 1
- file: "gfonts://Roboto@500"
  id: roboto_energy_measure
  size: 30
  bpp: 4
  glyphs: [k,W,h]

- file: "gfonts://Roboto"
  id: roboto_note
  size: 16
  bpp: 4
  glyphs: [C, u, r, e, n, t, D, a, i, l, y]

- file: "gfonts://Roboto@700"
  id: font_time
  size: 120
  bpp: 4
  glyphs: ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9", ":", " "]

# Clock
- file: "gfonts://Roboto@500"
  id: font_day
  size: 60
  bpp: 4
  glyphs: ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9", ".", " "]

- file: "gfonts://Roboto@500"
  id: font_day_of_week
  size: 60
  bpp: 4
  glyphs: ["П", "Н", "В", "Т", "С", "Р", "Ч", "Б", "Н", "Д"]

# Weather

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

```

- file: "gfonts://Roboto@500"
  id: roboto_weather_t
  size: 80
  bpp: 4
  glyphs: [" ", ".", "-", "°", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9"]

- file: "gfonts://Roboto@500"
  id: roboto_weather_t1
  size: 47
  bpp: 4
  glyphs: ["°"]

- file: "gfonts://Roboto@500"
  id: roboto_weather_t2
  size: 35
  bpp: 4
  glyphs: ["C", "c"]

- file: "gfonts://Roboto"
  id: roboto_weather_2
  size: 35
  bpp: 4
  glyphs: [" ", ".", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "-"]

- file: "gfonts://Roboto"
  id: roboto_weather_3
  size: 28
  bpp: 4
  glyphs: ["%"]

- file: "gfonts://Roboto@500"
  id: roboto_temp_hint
  size: 18
  bpp: 4
  glyphs: [" ", ".", "-", "=", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "a", "b", "c", "d", "e", "f", "g", "i", "j", "k", "l", "m", "n", "o", "p", "r", "s", "t", "u", "v", "w", "x", "y", "z", "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "R", "S", "Q", "T", "U", "V", "W", "X", "Y", "Z"]

- file: "gfonts://Roboto@500"
  id: roboto_temp
  size: 35
  bpp: 4
  glyphs: ["°"]

- file: "gfonts://Roboto@500"
  id: roboto_temp_c
  size: 23
  bpp: 4
  glyphs: ["C", "c"]

- file: "gfonts://Roboto@500"
  id: roboto_setting_author
  size: 25
  bpp: 4
  glyphs: [":", " ", ".", "-", "=", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "r", "s", "t", "u", "v", "w", "x", "y", "z", "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "R", "S", "Q", "T", "U", "V", "W", "X", "Y", "Z"]

# Outhers
- file: "gfonts://Bebas Neue@400" # "gfonts://Rubik@400" # "gfonts://Roboto" #
"gfonts://Oswald@600" #
  id: bebasneue400_40 # my_font2_b
  size: 40

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

```

bpp: 4
glyphs: [ " ", ".", "-", "°", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", a, n, o ]

- file: "gfonts://Bebas Neue@400" # "gfonts://Rubik@400" # "gfonts://Roboto" #
"gfonts://Oswald@600" #
  id: bebasneue400_50          # my_font2_b
  size: 50
  bpp: 4
  glyphs: [ " ", ".", "-", "°", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", a, n, o ]

- file: "gfonts://Roboto@500" # "gfonts://Oswald@500" # "gfonts://Rubik@400" #
"gfonts://Rubik@400" # "gfonts://Oswald@600" #
  id: roboto500_60          # my_font1_b
  size: 60
  bpp: 4
  glyphs: [ " ", ".", "-", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", a, n, o, P, g, e ]

```

image:

```

# Weather
# clear_night
- file: "/config/esphome/img/clear-night.png"
  type: RGB565
  id: img_clear_night
  transparency: alpha_channel
# cloudy
- file: "/config/esphome/img/cloudy.png"
  type: RGB565
  id: img_cloudy
  transparency: alpha_channel
# fog
- file: "/config/esphome/img/fog.png"
  type: RGB565
  id: img_fog
  transparency: alpha_channel
# hail
- file: "/config/esphome/img/hail.png"
  type: RGB565
  id: img_hail
  transparency: alpha_channel
# lightning
- file: "/config/esphome/img/lightning.png"
  type: RGB565
  id: img_lightning
  transparency: alpha_channel
# lightning_rainy
- file: "/config/esphome/img/lightning_rainy.png"
  type: RGB565
  id: img_lightning_rainy
  transparency: alpha_channel
# partly cloudy
- file: "/config/esphome/img/partly cloudy.png"
  type: RGB565
  id: img_partlycloudy
  transparency: alpha_channel
# partly cloudy day
- file: "/config/esphome/img/partly cloudy_day.png"
  type: RGB565
  id: img_partlycloudy_day
  transparency: alpha_channel
# partly cloudy night
- file: "/config/esphome/img/partly cloudy_night.png"

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

```

type: RGB565
id: img_partlycloudy_night
transparency: alpha_channel
# pouring
- file: "/config/esphome/img/pouring.png"
  type: RGB565
  id: img_pouring
  transparency: alpha_channel
# rainy
- file: "/config/esphome/img/rainy.png"
  type: RGB565
  id: img_rainy
  transparency: alpha_channel
# snowy
- file: "/config/esphome/img/snowy.png"
  type: RGB565
  id: img_snowy
  transparency: alpha_channel
# snowy_rainy
- file: "/config/esphome/img/snowy_rainy.png"
  type: RGB565
  id: img_snowy_rainy
  transparency: alpha_channel
# sunny
- file: "/config/esphome/img/sunny.png"
  type: RGB565
  id: img_sunny
  transparency: alpha_channel
# windy
- file: "/config/esphome/img/windy.png"
  type: RGB565
  id: img_windy
  transparency: alpha_channel
# windy-variant
- file: "/config/esphome/img/windy.png"
  type: RGB565
  id: img_windy_variant
  transparency: alpha_channel
# exceptional
- file: "/config/esphome/img/tornado.png"
  type: RGB565
  id: img_exceptional
  transparency: alpha_channel

```

switch:

```

- platform: template
  name: Antibern
  id: switch_antibern
  icon: mdi:television-shimmer
  optimistic: true
  internal: true
  entity_category: "config"
  turn_on_action:
    - if:
        condition: lvgl.is_paused
        then:
          - lvgl.resume:
          - lvgl.widget.redraw:
    - lvgl.pause:
        show_snow: true
    - lambda: |-
        id(antibern_work) = true;
        id(led_tft).set_level(0.0);

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

```

turn_off_action:
  - if:
    condition: lvgl.is_paused
    then:
      - lvgl.resume:
      - lvgl.widget.redraw:
      - lambda: |-
          id(antiburn_work) = false;
          id(led_tft).set_level(id(setting_brightness)/100.0);

```

```

binary_sensor:
  # Fridge / Freezer
  - platform: homeassistant
    id: freezer_work
    entity_id: switch.z2_Rz08
    internal: true
    on_state:
      then:
        lvgl.widget.update:
          id: btn_fridge
          state:
            checked: !lambda return x;
  # Kettle
  - platform: homeassistant
    id: kettle_work
    entity_id: switch.z2_Rz12
    internal: true
    on_state:
      then:
        lvgl.widget.update:
          id: btn_kettle
          state:
            checked: !lambda return x;
  # Boiler 1
  - platform: homeassistant
    id: boiler_1_work
    entity_id: switch.z2_rz01
    internal: true
    on_state:
      then:
        lvgl.widget.update:
          id: btn_boiler_1
          state:
            checked: !lambda return x;
  # Boiler 2
  - platform: homeassistant
    id: boiler_2_work
    entity_id: switch.z2_rz02
    internal: true
    on_state:
      then:
        lvgl.widget.update:
          id: btn_boiler_2
          state:
            checked: !lambda return x;
  # Motor 1
  - platform: homeassistant
    id: rz13_switch
    entity_id: switch.z2_rz13
    internal: true
    on_state:
      then:
        lvgl.widget.update:

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

```

        id: btn_motor_1
        state:
            checked: !lambda return x;
# Motor 2
- platform: homeassistant
  id: rz14_switch
  entity_id: switch.z2_rz14
  internal: true
  on_state:
    then:
      lvgl.widget.update:
        id: btn_motor_2
        state:
            checked: !lambda return x;
# Deye cooling
- platform: homeassistant
  id: rz15_switch
  entity_id: switch.z2_rz15
  internal: true
  on_state:
    then:
      lvgl.widget.update:
        id: btn_deye_cooling
        state:
            checked: !lambda return x;

# Light
# Shed
- platform: homeassistant
  id: sw1_switch
  entity_id: switch.z2_sw1
  internal: true
  on_state:
    then:
      lvgl.widget.update:
        id: btn_shed_light
        state:
            checked: !lambda return x;
# Cellar
- platform: homeassistant
  id: sw2_switch
  entity_id: switch.z2_sw2
  internal: true
  on_state:
    then:
      lvgl.widget.update:
        id: btn_cellar_light
        state:
            checked: !lambda return x;
# Garage
- platform: homeassistant
  id: sw6_switch
  entity_id: switch.z2_sw6
  internal: true
  on_state:
    then:
      lvgl.widget.update:
        id: btn_garage_light
        state:
            checked: !lambda return x;
# Door 2
- platform: homeassistant

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

```

id: sw7_switch
entity_id: switch.z2_sw7
internal: true
on_state:
  then:
    lvgl.widget.update:
      id: btn_door_2_light
      state:
        checked: !lambda return x;
sensor:
# Page 1
# Current
# Solar Panel
- platform: homeassistant
  id: dc_power
  entity_id: sensor.dc_power
  internal: true

# External power supply
- platform: homeassistant
  entity_id: sensor.z2_ph0_power
  id: z2_ph0_power
  internal: true

# External supply voltage value
- platform: homeassistant
  entity_id: sensor.z2_ph0_voltage
  id: z2_ph0_voltage
  internal: true

# Battery power
- platform: homeassistant
  id: solarman_battery_power # мощность батареи
  entity_id: sensor.solarman_battery_power
  internal: true

# Battery JKBMS power total
- platform: homeassistant
  id: bat_power
  entity_id: sensor.bat_power
  internal: true

# House consumption
- platform: homeassistant
  entity_id: sensor.z2_ph1_power
  id: z2_ph1_power
  internal: true

# Daily
# Solar Panel
- platform: homeassistant
  id: solarman_daily_production
  entity_id: sensor.solarman_daily_production
  internal: true

# External supply
- platform: homeassistant
  id: solarman_daily_energy_bought
  entity_id: sensor.solarman_daily_energy_bought
  internal: true

# Ah total Kwh
- platform: homeassistant

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

```

id: bat_energy
entity_id: sensor.but_energy
internal: true

# House consumption
- platform: homeassistant
  id: solarman_daily_load_consumption
  entity_id: sensor.solarman_daily_load_consumption
  internal: true
# Weather
- platform: homeassistant
  id: forecast_temp
  entity_id: weather.forecast_home
  attribute: temperature
  internal: true

- platform: homeassistant
  id: forecast_wet
  entity_id: weather.forecast_home
  attribute: humidity
  internal: true

- platform: homeassistant
  id: forecast_wind_speed
  entity_id: weather.forecast_home
  attribute: wind_speed
  internal: true

- platform: homeassistant
  id: forecast_cloud
  entity_id: weather.forecast_home
  attribute: cloud_coverage
  internal: true

# Temperature
# Freezer Power
- platform: homeassistant
  id: freezer_power
  entity_id: sensor.z2_Rz08_power
# Room
- platform: homeassistant
  id: sensor_temp_room
  entity_id: sensor.display7_temperature6
# Kettle Air
- platform: homeassistant
  id: sensor_temp_kettle_air
  entity_id: sensor.display7_temperature2
# Water
- platform: homeassistant
  id: sensor_temp_water
  entity_id: sensor.display7_temperature3
# Freezer
- platform: homeassistant
  id: sensor_temp_freezer
  entity_id: sensor.display7_temperature4
# Kettle Water Out
- platform: homeassistant
  id: sensor_temp_kettle_water_out
  entity_id: sensor.display7_temperature5
# Kettle Water In
- platform: homeassistant
  id: sensor_temp_kettle_water_in
  entity_id: sensor.display7_temperature1

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

```

# Boiler 1
- platform: homeassistant
  id: sensor_temp_boiler_1
  entity_id: sensor.display7_temperature7
# Boiler 2
- platform: homeassistant
  id: sensor_temp_boiler_2
  entity_id: sensor.display7_temperature8

# Boiler 1 Power
- platform: homeassistant
  id: boiler_1_power
  entity_id: sensor.z2_rz01_power
# Boiler 2 Power
- platform: homeassistant
  id: boiler_2_power
  entity_id: sensor.z2_rz02_power

text_sensor:
- platform: homeassistant
  id: weather_state
  entity_id: weather.forecast_home

lvgl:
  displays:
    - my_display
  touchscreens:
    - my_touch
  on_idle:
    timeout: ${idle_time} #120s
    then:
      - if:
          condition:
            lambda: 'return (id(setting_screen_off_idle));'
          then:
            - lvgl.pause:
            - lambda: |-
                if (id(setting_screen_off_idle)) {
                  id(screen_off_idle) = true;
                  id(led_tft).set_level(0.0);
                }
                if (id(setting_slide_page_idle)) {
                  id(slide_page_mode) = true;
                  id(slide_page_time) = 0;
                }
                if (id(current_page) == 5) {
                  id(show_page_1).execute();
                }
      }

style_definitions:
- id: style_nav_but
  outline_width: 0
  border_width: 1
  bg_color: 0x3399ff
  border_color: 0xffff00
  pad_all: 0
  radius: 10

- id: style_nav_but_active
  outline_width: 0
  border_width: 1
  bg_color: 0xfa5d2d
  border_color: 0xffff00

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

```

    pad_all: 0
    radius: 10

- id: style_top_r0
  pad_right: 0
  align: TOP_RIGHT

top_layer:
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    # Api status
    - label:
      id: label_api_status
      x: -32
      y: 7
      text: ${icon_server_off}
      align: top_right
      text_font: icon_status
      text_align: right
      text_color: ${status_err_color}
    # Wifi status
    - label:
      id: label_wifi_status
      x: -2
      y: 7
      text: ${icon_wifi_off}
      align: top_right
      text_font: icon_status
      text_align: right
      text_color: ${status_err_color}
    # Nav
    - obj:
      x: -8
      y: 45
      height: 225
      width: 40
      bg_color: ${main_bg_color}
      align: top_right
      border_width: 0
      outline_width: 0
      shadow_width: 0
      scrollbar_mode: "off"
      scrollable: 'false'
      pad_all: 0
      widgets:
        - button:
          id: nav_page_1
          styles: style_nav_but_active
          x: 0
          y: 0
          width: 40
          height: 40
          border_width: 0
          bg_color: ${nav_but_actv_bg_color}
          checkable: false
          pressed:
            bg_color: ${nav_but_pressed_bg_color}
          widgets:
            - label:
              x: 4
              y: 3
              pad_all: 0

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

```

        text_font: icon_nav
        text_color: 0xFFFFFF
        text: ${icon_nav_page_1}
    on_press:
        - if:
            condition:
                lambda: 'return (id(current_page) != 1);'
            then:
                - script.execute: show_page_1
- button:
    id: nav_page_2
    styles: style_nav_but
    x: 0
    y: 45
    width: 40
    height: 40
    border_width: 0
    bg_color: ${nav_but_bg_color}
    checkable: false
    pressed:
        bg_color: ${nav_but_pressed_bg_color}
    widgets:
    - label:
        x: 4
        y: 3
        pad_all: 0
        text_font: icon_nav
        text_color: 0xFFFFFF
        text: ${icon_nav_page_2}
    on_press:
        - if:
            condition:
                lambda: 'return (id(current_page) != 2);'
            then:
                - script.execute: show_page_2
- button:
    id: nav_page_3
    styles: style_nav_but
    x: 0
    y: 90
    width: 40
    height: 40
    border_width: 0
    bg_color: ${nav_but_bg_color}
    checkable: false
    pressed:
        bg_color: ${nav_but_pressed_bg_color}
    widgets:
    - label:
        x: 4
        y: 3
        pad_all: 0
        text_font: icon_nav
        text_color: 0xFFFFFF
        text: ${icon_nav_page_3}
    on_press:
        - if:
            condition:
                lambda: 'return (id(current_page) != 3);'
            then:
                - script.execute: show_page_3
- button:
    id: nav_page_4
    styles: style_nav_but

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

```

x: 0
y: 135
width: 40
height: 40
border_width: 0
bg_color: ${nav_but_bg_color}
checkable: false
pressed:
  bg_color: ${nav_but_pressed_bg_color}
widgets:
- label:
  x: 4
  y: 3
  pad_all: 0
  text_font: icon_nav
  text_color: 0xFFFFFFFF
  text: ${icon_nav_page_4}
on_press:
- if:
  condition:
    lambda: 'return (id(current_page) != 4);'
  then:
    - script.execute: show_page_4

- button:
  id: nav_page_5
  styles: style_nav_but
  x: 0
  y: 180
  width: 40
  height: 40
  border_width: 0
  bg_color: ${nav_but_bg_color}
  checkable: false
  pressed:
    bg_color: ${nav_but_pressed_bg_color}
  widgets:
  - label:
    x: 4
    y: 3
    pad_all: 0
    text_font: icon_nav
    text_color: 0xFFFFFFFF
    text: ${icon_nav_page_5}
  on_press:
  - if:
    condition:
      lambda: 'return (id(current_page) != 5);'
    then:
      - script.execute: show_page_5

pages:
- id: page_1
  bg_color: ${main_bg_color}
  widgets:
  # Hint
  - obj:
    x: 0
    y: 3
    height: 30
    width: 330
    bg_color: ${main_bg_color}
    border_width: 0

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

```

outline_width: 0
shadow_width: 0
pad_all: 0
scrollbar_mode: "off"
scrollable: 'false'
widgets:
  # Current
  - obj:
    x: 110
    height: 30
    width: 70
    bg_color: 0x323339
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'
    widgets:
      - label:
        y: 3
        x: 10
        text_font: roboto_note
        text_color: 0xd7d8d9
        text: "Current"

  # Daily
  - obj:
    x: 280
    height: 30
    width: 50
    bg_color: 0x323339
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'
    widgets:
      - label:
        y: 3
        x: 8
        text_font: roboto_note
        text_color: 0xd7d8d9
        text: "Daily"

# Solar Panel
- obj:
  x: 20
  y: 23
  height: 58
  width: 400
  bg_color: 0x53555f
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    # Icon
    - label:
      x: 10
      y: 9
      text_font: icon_page_1

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

```

        text_color: 0xffffffff
        text: ${icon_solar_panel}
# W
- label:
    x: 210
    y: 20
    text_font: roboto_energy_measure
    text_color: 0xd7d8d9
    text: "W"
# kWh
- label:
    x: 330
    y: 20
    text_font: roboto_energy_measure
    text_color: 0xd7d8d9
    text: "kWh"
# Current value
- obj:
    x: 54
    y: 5
    height: 50
    width: 150
    bg_color: 0x182533
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'
    widgets:
        - label:
            id: solar_p
            x: -5
            y: -9
            align: TOP_RIGHT
            text_font: roboto500_60
            text_color: 0xffffffff
            text: "- - -"
# Daily value
- obj:
    x: 245
    y: 5
    height: 50
    width: 79
    bg_color: 0x182533
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'
    text_color: 0xffffffff
    widgets:
        # Int
        - label:
            id: solar_pd_i
            x: -20
            y: -1
            align: TOP_RIGHT
            text_font: bebasneue400_50
            text_letter_space: 3
            text: "--."
        # Float
        - label:

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

```

        id: solar_pd_f
        x: 59
        y: 8
        text_font: bebasneue400_40
        text_letter_space: 0
        text: "-"

# External power supply
- obj:
    x: 20
    y: 85
    height: 58
    width: 400
    bg_color: 0x53555f
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'
    widgets:
        # Icon
        - label:
            x: 10
            y: 9
            text_font: icon_page_1
            text_color: 0xffffffff
            text: ${icon_tower}
        # W
        - label:
            x: 210
            y: 20
            text_font: roboto_energy_measure
            text_color: 0xd7d8d9
            text: "W"
        # kWh
        - label:
            x: 330
            y: 20
            text_font: roboto_energy_measure
            text_color: 0xd7d8d9
            text: "kWh"
        # Current value
        - obj:
            x: 54
            y: 5
            height: 50
            width: 150
            bg_color: 0x182533
            border_width: 0
            outline_width: 0
            shadow_width: 0
            pad_all: 0
            scrollbar_mode: "off"
            scrollable: 'false'
            widgets:
                - label:
                    id: supply_p
                    x: -5
                    y: -9
                    align: TOP_RIGHT
                    text_font: roboto500_60
                    text_color: 0xffffffff
                    text: "- - -"

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

```

# Daily value
- obj:
  x: 245
  y: 5
  height: 50
  width: 79
  bg_color: 0x182533
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  text_color: 0xffffffff
  widgets:
    # Int
    - label:
      id: supply_pd_i
      x: -20
      y: -1
      align: TOP_RIGHT
      text_font: bebasneue400_50
      text_letter_space: 3
      text: "--."
    # Float
    - label:
      id: supply_pd_f
      x: 59
      y: 8
      text_font: bebasneue400_40
      text_letter_space: 0
      text: "-"

# Battery
- obj:
  x: 20
  y: 147
  height: 58
  width: 400
  bg_color: 0x53555f
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    # Icon
    - label:
      x: 10
      y: 9
      text_font: icon_page_1
      text_color: 0xffffffff
      text: ${icon_battery}
    # W
    - label:
      x: 210
      y: 20
      text_font: roboto_energy_measure
      text_color: 0xd7d8d9
      text: "W"
    # kWh
    - label:
      x: 330

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

```

        y: 20
        text_font: roboto_energy_measure
        text_color: 0xd7d8d9
        text: "kWh"
# Current value
- obj:
    x: 54
    y: 5
    height: 50
    width: 150
    bg_color: 0x182533
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'
    widgets:
        - label:
            id: battery_p
            x: -5
            y: -9
            align: TOP_RIGHT
            text_font: roboto500_60
            text_color: 0xffffffff
            text: "- - -"
# Daily value
- obj:
    x: 245
    y: 5
    height: 50
    width: 79
    bg_color: 0x182533
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'
    text_color: 0xffffffff
    widgets:
        # Int
        - label:
            id: battery_pd_i
            x: -20
            y: -1
            align: TOP_RIGHT
            text_font: bebasneue400_50
            text_letter_space: 3
            text: "--."
        # Float
        - label:
            id: battery_pd_f
            x: 59
            y: 8
            text_font: bebasneue400_40
            text_letter_space: 0
            text: "-"

# House consumption
- obj:
    x: 20
    y: 209
    height: 58

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

```

width: 400
bg_color: 0x53555f
border_width: 0
outline_width: 0
shadow_width: 0
pad_all: 0
scrollbar_mode: "off"
scrollable: 'false'
widgets:
  # Icon
  - label:
    x: 10
    y: 9
    text_font: icon_page_1
    text_color: 0xffffffff
    text: ${icon_flash}

  # W
  - label:
    x: 210
    y: 20
    text_font: roboto_energy_measure
    text_color: 0xd7d8d9
    text: "W"

  # kWh
  - label:
    x: 330
    y: 20
    text_font: roboto_energy_measure
    text_color: 0xd7d8d9
    text: "kWh"

  # Current value
  - obj:
    x: 54
    y: 5
    height: 50
    width: 150
    bg_color: 0x182533
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'
    widgets:
      - label:
        id: consum_p
        x: -5
        y: -9
        align: TOP_RIGHT
        text_font: roboto500_60
        text_color: 0xffffffff
        text: "- - -"

  # Daily value
  - obj:
    x: 245
    y: 5
    height: 50
    width: 79
    bg_color: 0x182533
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

```

scrollable: 'false'
text_color: 0xffffffff
widgets:
  # Int
  - label:
      id: consum_pd_i
      x: -20
      y: -1
      align: TOP_RIGHT
      text_font: bebasneue400_50
      text_letter_space: 3
      text: "--."
  # Float
  - label:
      id: consum_pd_f
      x: 59
      y: 8
      text_font: bebasneue400_40
      text_letter_space: 0
      text: "-"

- id: page_2
  bg_color: ${main_bg_color}
  widgets:
    # Warning
    - obj:
        id: temp_warning_1
        x: 20
        y: 5
        height: 35
        width: 389
        radius: 8
        bg_color: 0xf09f13
        border_width: 0
        outline_width: 0
        shadow_width: 0
        pad_all: 0
        scrollbar_mode: "off"
        scrollable: 'false'
        widgets:
          - label:
              x: 15
              text_font: icon_temp
              text_color: 0xbf6804
              text: ${icon_warning}
          - label:
              x: -15
              align: TOP_RIGHT
              text_font: icon_temp
              text_color: 0xbf6804
              text: ${icon_warning}
          - label:
              align: LV_ALIGN_TOP_MID
              text_font: icon_temp
              text_color: 0xbf6804
              text: ${icon_warning}
          - label:
              x: -83
              align: LV_ALIGN_TOP_MID
              text_font: icon_temp
              text_color: 0xbf6804
              text: ${icon_warning}
          - label:
              x: 83

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

```

        align: LV_ALIGN_TOP_MID
        text_font: icon_temp
        text_color: 0xbf6804
        text: ${icon_warning}
# Warning 2
- obj:
    id: temp_warning_2
    x: 20
    y: -5
    height: 35
    width: 389
    radius: 8
    align: LV_ALIGN_BOTTOM_LEFT
    bg_color: 0xf09f13
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'
    widgets:
        - label:
            x: 15
            text_font: icon_temp
            text_color: 0xbf6804
            text: ${icon_warning}
        - label:
            x: -15
            align: TOP_RIGHT
            text_font: icon_temp
            text_color: 0xbf6804
            text: ${icon_warning}
        - label:
            align: LV_ALIGN_TOP_MID
            text_font: icon_temp
            text_color: 0xbf6804
            text: ${icon_warning}
        - label:
            x: -83
            align: LV_ALIGN_TOP_MID
            text_font: icon_temp
            text_color: 0xbf6804
            text: ${icon_warning}
        - label:
            x: 83
            align: LV_ALIGN_TOP_MID
            text_font: icon_temp
            text_color: 0xbf6804
            text: ${icon_warning}
# Error
- obj:
    id: temp_error_1
    x: 20
    y: 5
    height: 35
    width: 389
    radius: 8
    bg_color: ${temp_status_err_color}
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

```

widgets:
- label:
  x: 15
  text_font: icon_temp
  text_color: 0x910d0d
  text: ${icon_error}
- label:
  x: -15
  align: TOP_RIGHT
  text_font: icon_temp
  text_color: 0x910d0d
  text: ${icon_error}
- label:
  align: LV_ALIGN_TOP_MID
  text_font: icon_temp
  text_color: 0x910d0d
  text: ${icon_error}
- label:
  x: -83
  align: LV_ALIGN_TOP_MID
  text_font: icon_temp
  text_color: 0x910d0d
  text: ${icon_error}
- label:
  x: 83
  align: LV_ALIGN_TOP_MID
  text_font: icon_temp
  text_color: 0x910d0d
  text: ${icon_error}
# Error 2
- obj:
  id: temp_error_2
  x: 20
  y: -5
  height: 35
  width: 389
  radius: 8
  align: LV_ALIGN_BOTTOM_LEFT
  bg_color: ${temp_status_err_color}
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    - label:
      x: 15
      text_font: icon_temp
      text_color: 0x910d0d
      text: ${icon_error}
    - label:
      x: -15
      align: TOP_RIGHT
      text_font: icon_temp
      text_color: 0x910d0d
      text: ${icon_error}
    - label:
      align: LV_ALIGN_TOP_MID
      text_font: icon_temp
      text_color: 0x910d0d
      text: ${icon_error}
    - label:
      x: -83

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

```

        align: LV_ALIGN_TOP_MID
        text_font: icon_temp
        text_color: 0x910d0d
        text: ${icon_error}
    - label:
        x: 83
        align: LV_ALIGN_TOP_MID
        text_font: icon_temp
        text_color: 0x910d0d
        text: ${icon_error}

# Freezer status
- obj:
    id: temp_freezer_status
    x: 16
    y: 141
    height: 88
    width: 94
    radius: 8
    bg_color: ${temp_status_err_color}
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'

# Kettle status
- obj:
    id: temp_kettle_status
    x: 117
    y: 141
    height: 88
    width: 94
    radius: 8
    bg_color: ${temp_status_err_color}
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'

# Boiler 1 status
- obj:
    id: temp_boiler_1_status
    x: 117
    y: 46
    height: 88
    width: 94
    radius: 8
    bg_color: ${temp_status_err_color}
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'

# Boiler 2 status
- obj:
    id: temp_boiler_2_status
    x: 218
    y: 46
    height: 88
    width: 94
    radius: 8

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

```
bg_color: ${temp_status_err_color}
border_width: 0
outline_width: 0
shadow_width: 0
pad_all: 0
scrollbar_mode: "off"
scrollable: 'false'
```

Room

```
- obj:
  x: 20
  y: 50
  height: 80
  width: 86
  bg_color: 0x53555f
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    # Hint
    - obj:
      x: 3
      y: 3
      height: 21
      width: 80
      radius: 3
      bg_color: 0x323339
      border_width: 0
      outline_width: 0
      shadow_width: 0
      pad_all: 0
      scrollbar_mode: "off"
      scrollable: 'false'
      widgets:
        - label:
            y: 0
            align: LV_ALIGN_TOP_MID
            text_font: roboto_temp_hint
            text_color: 0xd7d8d9
            text: "Room"
```

Main Body

```
- obj:
  y: 24
  height: 56
  width: 86
  bg_color: 0x53555f
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    # Value
    - obj:
      x: 3
      y: 3
      height: 50
      width: 65
      bg_color: 0x182533
      border_width: 0
```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

```

        outline_width: 0
        shadow_width: 0
        pad_all: 0
        scrollbar_mode: "off"
        scrollable: 'false'
        widgets:
          - label:
              id: l_temp_room
              x: -5
              y: -1
              align: TOP_RIGHT
              text_font: bebasneue400_50
              text_color: 0xffffffff
              text: "-00"
      # °
      - label:
          x: 70
          y: 2
          text_font: roboto_temp
          text_color: 0xd7d8d9
          text: "°"
      # c
      - label:
          x: -3
          y: -4
          align: LV_ALIGN_BOTTOM_RIGHT
          text_font: roboto_temp_c
          text_color: 0xd7d8d9
          text: "c"
# Boiler 1
- obj:
    x: 121
    y: 50
    height: 80
    width: 86
    bg_color: 0x53555f
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'
    widgets:
      # Hint
      - obj:
          x: 3
          y: 3
          height: 21
          width: 80
          radius: 3
          bg_color: 0x323339
          border_width: 0
          outline_width: 0
          shadow_width: 0
          pad_all: 0
          scrollbar_mode: "off"
          scrollable: 'false'
          widgets:
            - label:
                y: 0
                align: LV_ALIGN_TOP_MID
                text_font: roboto_temp_hint
                text_color: 0xd7d8d9
                text: "Boiler 1"

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

```

# Main Body
- obj:
  y: 24
  height: 56
  width: 86
  bg_color: 0x53555f
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    # Value
    - obj:
      x: 3
      y: 3
      height: 50
      width: 65
      bg_color: 0x182533
      border_width: 0
      outline_width: 0
      shadow_width: 0
      pad_all: 0
      scrollbar_mode: "off"
      scrollable: 'false'
      widgets:
        - label:
          id: l_temp_boiler_1
          x: -5
          y: -1
          align: TOP_RIGHT
          text_font: bebasneue400_50
          text_color: 0xffffffff
          text: "-00"

    # °
    - label:
      x: 70
      y: 2
      text_font: roboto_temp
      text_color: 0xd7d8d9
      text: "°"

    # c
    - label:
      x: -3
      y: -4
      align: LV_ALIGN_BOTTOM_RIGHT
      text_font: roboto_temp_c
      text_color: 0xd7d8d9
      text: "c"

# Boiler 2
- obj:
  x: 222
  y: 50
  height: 80
  width: 86
  bg_color: 0x53555f
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

```

# Hint
- obj:
  x: 3
  y: 3
  height: 21
  width: 80
  radius: 3
  bg_color: 0x323339
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    - label:
      y: 0
      align: LV_ALIGN_TOP_MID
      text_font: roboto_temp_hint
      text_color: 0xd7d8d9
      text: "Boiler 2"
# Main Body
- obj:
  y: 24
  height: 56
  width: 86
  bg_color: 0x53555f
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    # Value
    - obj:
      x: 3
      y: 3
      height: 50
      width: 65
      bg_color: 0x182533
      border_width: 0
      outline_width: 0
      shadow_width: 0
      pad_all: 0
      scrollbar_mode: "off"
      scrollable: 'false'
      widgets:
        - label:
          id: l_temp_boiler_2
          x: -5
          y: -1
          align: TOP_RIGHT
          text_font: bebasneue400_50
          text_color: 0xffffffff
          text: "-00"
    # °
    - label:
      x: 70
      y: 2
      text_font: roboto_temp
      text_color: 0xd7d8d9
      text: "°"
# c

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

```

- label:
  x: -3
  y: -4
  align: LV_ALIGN_BOTTOM_RIGHT
  text_font: roboto_temp_c
  text_color: 0xd7d8d9
  text: "c"

# Water
- obj:
  x: 323
  y: 50
  height: 80
  width: 86
  bg_color: 0x53555f
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    # Hint
    - obj:
      x: 3
      y: 3
      height: 21
      width: 80
      radius: 3
      bg_color: 0x323339
      border_width: 0
      outline_width: 0
      shadow_width: 0
      pad_all: 0
      scrollbar_mode: "off"
      scrollable: 'false'
      widgets:
        - label:
          y: 0
          align: LV_ALIGN_TOP_MID
          text_font: roboto_temp_hint
          text_color: 0xd7d8d9
          text: "Water"

# Main Body
- obj:
  y: 24
  height: 56
  width: 86
  bg_color: 0x53555f
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    # Value
    - obj:
      x: 3
      y: 3
      height: 50
      width: 65
      bg_color: 0x182533
      border_width: 0
      outline_width: 0

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

```

        shadow_width: 0
        pad_all: 0
        scrollbar_mode: "off"
        scrollable: 'false'
        widgets:
          - label:
              id: l_temp_water
              x: -5
              y: -1
              align: TOP_RIGHT
              text_font: bebasneue400_50
              text_color: 0xffffffff
              text: "-00"
      # °
      - label:
          x: 70
          y: 2
          text_font: roboto_temp
          text_color: 0xd7d8d9
          text: "°"
      # c
      - label:
          x: -3
          y: -4
          align: LV_ALIGN_BOTTOM_RIGHT
          text_font: roboto_temp_c
          text_color: 0xd7d8d9
          text: "c"
# Freezer
- obj:
    x: 20
    y: 145
    height: 80
    width: 86
    bg_color: 0x53555f
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'
    widgets:
      # Hint
      - obj:
          x: 3
          y: 3
          height: 21
          width: 80
          radius: 3
          bg_color: 0x323339
          border_width: 0
          outline_width: 0
          shadow_width: 0
          pad_all: 0
          scrollbar_mode: "off"
          scrollable: 'false'
          widgets:
            - label:
                y: 0
                align: LV_ALIGN_TOP_MID
                text_font: roboto_temp_hint
                text_color: 0xd7d8d9
                text: "Freezer"
# Main Body

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

```

- obj:
  y: 24
  height: 56
  width: 86
  bg_color: 0x53555f
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    # Value
    - obj:
      x: 3
      y: 3
      height: 50
      width: 65
      bg_color: 0x182533
      border_width: 0
      outline_width: 0
      shadow_width: 0
      pad_all: 0
      scrollbar_mode: "off"
      scrollable: 'false'
      widgets:
        - label:
          id: l_temp_freezer
          x: -5
          y: -1
          align: TOP_RIGHT
          text_font: bebasneue400_50
          text_color: 0xffffffff
          text: "-00"

    # °
    - label:
      x: 70
      y: 2
      text_font: roboto_temp
      text_color: 0xd7d8d9
      text: "°"

    # c
    - label:
      x: -3
      y: -4
      align: LV_ALIGN_BOTTOM_RIGHT
      text_font: roboto_temp_c
      text_color: 0xd7d8d9
      text: "c"

# Kettle Air
- obj:
  x: 121
  y: 145
  height: 80
  width: 86
  bg_color: 0x53555f
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    # Hint

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

```

- obj:
  x: 3
  y: 3
  height: 21
  width: 80
  radius: 3
  bg_color: 0x323339
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    - label:
      y: 0
      align: LV_ALIGN_TOP_MID
      text_font: roboto_temp_hint
      text_color: 0xd7d8d9
      text: "Kettle Air"
# Main Body
- obj:
  y: 24
  height: 56
  width: 86
  bg_color: 0x53555f
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    # Value
    - obj:
      x: 3
      y: 3
      height: 50
      width: 65
      bg_color: 0x182533
      border_width: 0
      outline_width: 0
      shadow_width: 0
      pad_all: 0
      scrollbar_mode: "off"
      scrollable: 'false'
      widgets:
        - label:
          id: l_temp_kettle_air
          x: -5
          y: -1
          align: TOP_RIGHT
          text_font: bebasneue400_50
          text_color: 0xffffffff
          text: "000"
    # °
    - label:
      x: 70
      y: 2
      text_font: roboto_temp
      text_color: 0xd7d8d9
      text: "°"
    # c
    - label:

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

```

        x: -3
        y: -4
        align: LV_ALIGN_BOTTOM_RIGHT
        text_font: roboto_temp_c
        text_color: 0xd7d8d9
        text: "c"
# Kettle water In
- obj:
  x: 222
  y: 145
  height: 80
  width: 86
  bg_color: 0x53555f
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    # Hint
    - obj:
      x: 3
      y: 3
      height: 21
      width: 80
      radius: 3
      bg_color: 0x323339
      border_width: 0
      outline_width: 0
      shadow_width: 0
      pad_all: 0
      scrollbar_mode: "off"
      scrollable: 'false'
      widgets:
        - label:
          y: 0
          align: LV_ALIGN_TOP_MID
          text_font: roboto_temp_hint
          text_color: 0xd7d8d9
          text: "Ktl. W In"
# Main Body
- obj:
  y: 24
  height: 56
  width: 86
  bg_color: 0x53555f
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    # Value
    - obj:
      x: 3
      y: 3
      height: 50
      width: 65
      bg_color: 0x182533
      border_width: 0
      outline_width: 0
      shadow_width: 0

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

```

        pad_all: 0
        scrollbar_mode: "off"
        scrollable: 'false'
        widgets:
          - label:
              id: l_temp_kettle_water_in
              x: -5
              y: -1
              align: TOP_RIGHT
              text_font: bebasneue400_50
              text_color: 0xffffffff
              text: "-00"
      # °
      - label:
          x: 70
          y: 2
          text_font: roboto_temp
          text_color: 0xd7d8d9
          text: "°"
      # c
      - label:
          x: -3
          y: -4
          align: LV_ALIGN_BOTTOM_RIGHT
          text_font: roboto_temp_c
          text_color: 0xd7d8d9
          text: "c"
# Kettle Water Out
- obj:
    x: 323
    y: 145
    height: 80
    width: 86
    bg_color: 0x53555f
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'
    widgets:
      # Hint
      - obj:
          x: 3
          y: 3
          height: 21
          width: 80
          radius: 3
          bg_color: 0x323339
          border_width: 0
          outline_width: 0
          shadow_width: 0
          pad_all: 0
          scrollbar_mode: "off"
          scrollable: 'false'
          widgets:
            - label:
                y: 0
                align: LV_ALIGN_TOP_MID
                text_font: roboto_temp_hint
                text_color: 0xd7d8d9
                text: "Ktl. W Out"
# Main Body
- obj:

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

```

y: 24
height: 56
width: 86
bg_color: 0x53555f
border_width: 0
outline_width: 0
shadow_width: 0
pad_all: 0
scrollbar_mode: "off"
scrollable: 'false'
widgets:
  # Value
  - obj:
    x: 3
    y: 3
    height: 50
    width: 65
    bg_color: 0x182533
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'
    widgets:
      - label:
        id: l_temp_kettle_water_out
        x: -5
        y: -1
        align: TOP_RIGHT
        text_font: bebasneue400_50
        text_color: 0xffffffff
        text: "000"

  # °
  - label:
    x: 70
    y: 2
    text_font: roboto_temp
    text_color: 0xd7d8d9
    text: "°"

  # c
  - label:
    x: -3
    y: -4
    align: LV_ALIGN_BOTTOM_RIGHT
    text_font: roboto_temp_c
    text_color: 0xd7d8d9
    text: "c"

- id: page_3
  bg_color: ${main_bg_color}
  widgets:
    # Boiler 1
    - button:
      id: btn_boiler_1
      x: 60
      y: 30
      width: 60
      height: 60
      checkable: true
      text_color: 0x0f398c
      on_click:
        - homeassistant.action:
            action: switch.toggle

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

```

        data:
            entity_id: switch.z2_rz01
    widgets:
        - label:
            text_font: icon_page_3
            align: CENTER
            text: ${icon_boiler_1}
# Boiler 2
- button:
    id: btn_boiler_2
    x: 140
    y: 30
    width: 60
    height: 60
    checkable: true
    text_color: 0x0f398c
    on_click:
        - homeassistant.action:
            action: switch.toggle
            data:
                entity_id: switch.z2_rz02
    widgets:
        - label:
            text_font: icon_page_3
            align: CENTER
            text: ${icon_boiler_2}
# Motor 1
- button:
    id: btn_motor_1
    x: 220
    y: 30
    width: 60
    height: 60
    checkable: true
    text_color: 0x0f398c
    on_click:
        - homeassistant.action:
            action: switch.toggle
            data:
                entity_id: switch.z2_rz13
    widgets:
        - label:
            text_font: icon_page_3
            align: CENTER
            text: ${icon_motor_1}
# Motor 2
- button:
    id: btn_motor_2
    x: 300
    y: 30
    width: 60
    height: 60
    checkable: true
    text_color: 0x0f398c
    on_click:
        - homeassistant.action:
            action: switch.toggle
            data:
                entity_id: switch.z2_rz14
    widgets:
        - label:
            text_font: icon_page_3
            align: CENTER
            text: ${icon_motor_2}

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

```

# Kettle
- button:
  id: btn_kettle
  x: 60
  y: 110
  width: 60
  height: 60
  checkable: true
  text_color: 0x0f398c
  on_click:
    - homeassistant.action:
      action: switch.toggle
      data:
        entity_id: switch.z2_Rz12
  widgets:
    - label:
      text_font: icon_page_3
      align: CENTER
      text: ${icon_kettle}
# Garage light
- button:
  id: btn_garage_light
  x: 140
  y: 110
  width: 60
  height: 60
  checkable: true
  text_color: 0x0f398c
  on_click:
    - homeassistant.action:
      action: switch.toggle
      data:
        entity_id: switch.z2_sw6
  widgets:
    - label:
      text_font: icon_page_3
      align: CENTER
      text: ${icon_garage}
# Cellar light
- button:
  id: btn_cellar_light
  x: 220
  y: 110
  width: 60
  height: 60
  checkable: true
  text_color: 0x0f398c
  on_click:
    - homeassistant.action:
      action: switch.toggle
      data:
        entity_id: switch.z2_sw2
  widgets:
    - label:
      text_font: icon_page_3
      align: CENTER
      text: ${icon_cellar}
# Shed light
- button:
  id: btn_shed_light
  x: 300
  y: 110
  width: 60
  height: 60

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

```

checkable: true
text_color: 0x0f398c
on_click:
  - homeassistant.action:
      action: switch.toggle
      data:
          entity_id: switch.z2_sw1
widgets:
  - label:
      text_font: icon_page_3
      align: CENTER
      text: ${icon_shed}
# Door 2 light
- button:
  id: btn_door_2_light
  x: 60
  y: 190
  width: 60
  height: 60
  checkable: true
  text_color: 0x0f398c
  on_click:
    - homeassistant.action:
        action: switch.toggle
        data:
            entity_id: switch.z2_sw7
  widgets:
    - label:
        text_font: icon_page_3
        align: CENTER
        text: ${icon_door_2}
# Deye cooling
- button:
  id: btn_deye_cooling
  x: 140
  y: 190
  width: 60
  height: 60
  checkable: true
  text_color: 0x0f398c
  on_click:
    - homeassistant.action:
        action: switch.toggle
        data:
            entity_id: switch.z2_rz15
  widgets:
    - label:
        text_font: icon_page_3
        align: CENTER
        text: ${icon_deye_cooling}
# Fridge
- button:
  id: btn_fridge
  x: 220
  y: 190
  width: 60
  height: 60
  checkable: true
  text_color: 0x0f398c
  on_click:
    - homeassistant.action:
        action: switch.toggle
        data:
            entity_id: switch.z2_Rz08

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

```

widgets:
  - label:
      text_font: icon_page_3
      align: CENTER
      text: ${icon_fridge}
- id: page_4
  bg_color: ${main_bg_color}
  widgets:
    # Clock
    - obj:
        id: obj_timed
        x: 55
        y: 5
        height: 180
        width: 320
        bg_color: ${main_bg_color}
        border_width: 0
        outline_width: 0
        shadow_width: 0
        pad_all: 0
        radius: 0
        scrollbar_mode: "off"
        scrollable: 'false'
        widgets:
          - label:          # Time hh mm
              id: clock_time
              x: -5
              y: -10
              text_font: font_time
              text_letter_space: 2
              text_color: 0xFFFFFF
              styles: style_top_r0
              text: "00 00"
          - label:          # :
              id: clock_time_dots
              x: -143
              y: -10
              text_font: font_time
              text_letter_space: 2
              text_color: 0xFFFFFF
              styles: style_top_r0
              text: ":"
          - label:          # Date
              id: clock_date
              x: -5
              y: 105
              text_font: font_day
              text_letter_space: 0
              text_color: 0xd7d8d9
              styles: style_top_r0
              text: "00"
          - label:          # Day of week
              id: clock_day_of_week
              x: 25
              y: 105
              text_font: font_day_of_week
              text_letter_space: 0
              text_color: ${clock_day_norm_color}
              text: "Нн"
    # Weather
    - obj:
        x: 0
        y: 180

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

```

height: 85
width: 415
bg_color: 0x323339 #${main_bg_color}
border_width: 0
outline_width: 0
shadow_width: 0
pad_all: 0
radius: 5
scrollbar_mode: "off"
scrollable: 'false'
widgets:
  # Temperature
  - label:
    id: weather_temp
    x: -285
    y: -5
    text_font: roboto_weather_t
    text_letter_space: 2
    text_color: 0xFFFFFFFF
    styles: style_top_r0
    text: "-00"
  - label: # "°"
    x: 133
    y: 1
    text_font: roboto_weather_t1
    text_color: 0xd7d8d9
    text: "°"
  - label: # "C"
    x: 138
    y: 36
    text_font: roboto_weather_t2
    text_color: 0xd7d8d9
    text: "C"
  # Wheather image / icon
  - image:
    id: wheather_image
    src: img_clear_night
    x: 175
    y: 3
  # Wet
  - label: # icon
    x: -101
    y: 12
    text_font: icon_weather
    text_letter_space: 0
    text_color: 0x8ad2ff
    styles: style_top_r0
    text: "${icon_weather_wet}"
  - label: # value
    id: weather_wet
    x: -40
    y: 6
    text_font: roboto_weather_2
    text_letter_space: 0
    text_color: 0x8ad2ff #0xd7d8d9
    styles: style_top_r0
    text: "000"
  - label: # %
    x: 379
    y: 12
    text_font: roboto_weather_3
    text_color: 0xd7d8d9
    text: "%"

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

```

# Cloud
- label: # icon
  x: -101
  y: 48
  text_font: icon_weather
  text_letter_space: 0
  text_color: 0xffff0c2
  styles: style_top_r0
  text: "${icon_weather_cloud}"
- label: # value
  id: weather_cloud
  x: -40
  y: 42
  text_font: roboto_weather_2
  text_letter_space: 0
  text_color: 0xffff0c2 #0xd7d8d9
  styles: style_top_r0
  text: "000"
- label: # %
  x: 379
  y: 48
  text_font: roboto_weather_3
  text_color: 0xd7d8d9
  text: "%"

- id: page_5
  bg_color: ${main_bg_color}
  scrollbar_mode: "off"
  scrollable: 'true'
  widgets:
    - label:
      x: 30
      y: 5
      text_font: roboto_setting_author
      text_color: 0xd7d8d9
      text: "Author: IO-11 Kudelia Olexander"

# Brightness
- obj:
  x: 15
  y: 40
  height: 60
  width: 320
  bg_color: 0x323339
  border_width: 0
  outline_width: 0
  shadow_width: 0
  pad_all: 0
  scrollbar_mode: "off"
  scrollable: 'false'
  widgets:
    - label:
      x: 15
      y: 5
      text_font: roboto_setting_author
      text_color: 0xffffffff
      text: "Brightness:"
    - label:
      id: l_brightness_value
      x: 150
      y: 5
      text_font: roboto_setting_author
      text_color: 0xffffffff
      text: "-"
    - slider:

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

```

        x: 10
        y: 40
        width: 290
        height: 10
        id: slider_brightness
        value: 45
        min_value: 20
        max_value: 100
        on_value:
            then:
                - lambda: |-
                    id(setting_brightness) = x;
                    id(led_tft).set_level(id(setting_brightness)/100.0);

                    static char buf[20];
                    sprintf(buf, "%.0f", x);
                    lv_label_set_text( id(l_brightness_value), buf );

# LCD anti burn Protect
- obj:
    x: 15
    y: 105
    height: 50
    width: 320
    bg_color: 0x323339
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'
    widgets:
        - label:
            x: 80
            y: 10
            text_font: roboto_setting_author
            text_color: 0xffffffff
            text: "Anti-burn"
        - switch:
            x: 15
            y: 10
            id: lvgl_switch_antiburn
            on_value:
                then:
                    - lambda: |-

                        id(setting_antiburn) = x;
# Turn off screen when idle
- obj:
    x: 15
    y: 160
    height: 50
    width: 320
    bg_color: 0x323339
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'
    widgets:
        - label:
            x: 80
            y: 10

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

```

        text_font: roboto_setting_author
        text_color: 0xffffffff
        text: "Screen off when idle"
    - switch:
        x: 15
        y: 10
        id: lvgl_switch_screen_off_idle
        on_value:
            then:
                - lambda: |-
                    id(setting_screen_off_idle) = x;
# Slide page when idle
- obj:
    x: 15
    y: 215
    height: 50
    width: 320
    bg_color: 0x323339
    border_width: 0
    outline_width: 0
    shadow_width: 0
    pad_all: 0
    scrollbar_mode: "off"
    scrollable: 'false'
    widgets:
        - label:
            x: 80
            y: 10
            text_font: roboto_setting_author
            text_color: 0xffffffff
            text: "Slide page when idle"
        - switch:
            x: 15
            y: 10
            id: lvgl_switch_slide_page_idle
            on_value:
                then:
                    - lambda: |-
                        id(setting_slide_page_idle) = x;

```

globals:

```

- id: current_page
  type: int
  restore_value: no
  initial_value: '1'

- id: setting_brightness
  type: int
  initial_value: '60'
  restore_value: yes

- id: setting_antiburn
  type: bool
  initial_value: '1'
  restore_value: yes

- id: antiburn_work
  type: bool

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

```
initial_value: '0'  
restore_value: no
```

```
- id: setting_screen_off_idle  
type: bool  
initial_value: '0'  
restore_value: yes
```

```
- id: screen_off_idle  
type: bool  
initial_value: '0'  
restore_value: no
```

```
- id: setting_slide_page_idle  
type: bool  
initial_value: '1'  
restore_value: yes
```

```
- id: slide_page_mode  
type: bool  
initial_value: '0'  
restore_value: no
```

```
- id: slide_page_time  
type: int  
initial_value: '0'  
restore_value: no
```

```
- id: weather_state_old  
type: int  
initial_value: '0'  
restore_value: no
```

```
- id: clock_time_show_dots  
type: bool  
restore_value: no  
initial_value: '0'
```

```
- id: kettle_old_work  
type: bool  
restore_value: no  
initial_value: '0'
```

```
- id: kettle_time_on  
type: unsigned long  
restore_value: no  
initial_value: '0'
```

script:

```
- id: restore_data  
then:  
  - lambda: |-  
    // Brightness  
    lv_slider_set_value(id(slider_brightness), id(setting_brightness), LV_ANIM_OFF);  
    id(led_tft).set_level(id(setting_brightness)/100.0);  
  
    static char buf[20];  
    sprintf(buf, "%i", id(setting_brightness));  
    lv_label_set_text( id(l_brightness_value), buf );  
  
    // Antiburn  
    if (id(setting_antiburn)) {  
      lv_obj_add_state(lvgl_switch_antiburn, LV_STATE_CHECKED);
```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

```

    }
    else {
        lv_obj_clear_state(lvgl_switch_antiburn, LV_STATE_CHECKED);
    }

    // Screen of idle
    if (id(setting_screen_off_idle)) {
        lv_obj_add_state(lvgl_switch_screen_off_idle, LV_STATE_CHECKED);
    }
    else {
        lv_obj_clear_state(lvgl_switch_screen_off_idle, LV_STATE_CHECKED);
    }

    // Slide page idle
    if (id(setting_slide_page_idle)) {
        lv_obj_add_state(lvgl_switch_slide_page_idle, LV_STATE_CHECKED);
    }
    else {
        lv_obj_clear_state(lvgl_switch_slide_page_idle, LV_STATE_CHECKED);
    }

- id: show_page_1
  then:
    - lvgl.page.show: page_1
    - lambda: |-
        id(current_page) = 1;
        id(update_page_1).execute();

        lv_obj_set_style_bg_color(id(nav_page_2), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
        lv_obj_set_style_bg_color(id(nav_page_3), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
        lv_obj_set_style_bg_color(id(nav_page_4), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
        lv_obj_set_style_bg_color(id(nav_page_5), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
        lv_obj_set_style_bg_color(id(nav_page_1), lv_color_hex(${nav_but_actv_bg_color}),
LV_PART_MAIN );
    - id: show_page_2
      then:
        - lvgl.page.show: page_2
        - lambda: |-
            id(current_page) = 2;
            id(update_temp).execute();

            lv_obj_set_style_bg_color(id(nav_page_1), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
            lv_obj_set_style_bg_color(id(nav_page_3), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
            lv_obj_set_style_bg_color(id(nav_page_4), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
            lv_obj_set_style_bg_color(id(nav_page_5), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
            lv_obj_set_style_bg_color(id(nav_page_2), lv_color_hex(${nav_but_actv_bg_color}),
LV_PART_MAIN );
    - id: show_page_3
      then:
        - lvgl.page.show: page_3
        - lambda: |-
            id(current_page) = 3;

            lv_obj_set_style_bg_color(id(nav_page_1), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

```

        lv_obj_set_style_bg_color(id(nav_page_2), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
        lv_obj_set_style_bg_color(id(nav_page_4), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
        lv_obj_set_style_bg_color(id(nav_page_5), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
        lv_obj_set_style_bg_color(id(nav_page_3), lv_color_hex(${nav_but_actv_bg_color}),
LV_PART_MAIN );
- id: show_page_4
  then:
    - lvgl.page.show: page_4
    - lambda: |-
      id(current_page) = 4;
      id(update_time).execute();
      id(update_weather).execute();

        lv_obj_set_style_bg_color(id(nav_page_1), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
        lv_obj_set_style_bg_color(id(nav_page_2), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
        lv_obj_set_style_bg_color(id(nav_page_3), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
        lv_obj_set_style_bg_color(id(nav_page_5), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
        lv_obj_set_style_bg_color(id(nav_page_4), lv_color_hex(${nav_but_actv_bg_color}),
LV_PART_MAIN );
- id: show_page_5
  then:
    - lvgl.page.show: page_5
    - lambda: |-
      id(current_page) = 5;

        lv_obj_set_style_bg_color(id(nav_page_1), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
        lv_obj_set_style_bg_color(id(nav_page_2), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
        lv_obj_set_style_bg_color(id(nav_page_3), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
        lv_obj_set_style_bg_color(id(nav_page_4), lv_color_hex(${nav_but_bg_color}),
LV_PART_MAIN );
        lv_obj_set_style_bg_color(id(nav_page_5), lv_color_hex(${nav_but_actv_bg_color}),
LV_PART_MAIN );

- id: update_page_1
  then:
    - lambda: |-
      static char buf[11];
      float v = 0;
      int v_i,v_f;

      // Solar panel
      // Current
      if (isnan(id(dc_power).state)) {
        lv_label_set_text(id(solar_p), "- - -");
      }
      else {
        if (id(dc_power).state == 0) {
          lv_obj_set_style_text_color(id(solar_p), lv_color_hex(0xffffffff),
LV_PART_MAIN); // Ok
        }
        else {
          lv_obj_set_style_text_color(id(solar_p), lv_color_hex(${power_ok_color}),
LV_PART_MAIN); // Ok

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

```

    }

    snprintf(buf, sizeof(buf), "%i", int(id(dc_power).state));
    lv_label_set_text(id(solar_p), buf);
}

// Daily
if (isnan(id(solarman_daily_production).state)) {
    lv_label_set_text(id(solar_pd_i), "-."); lv_label_set_text(id(solar_pd_f), "-");
}
else {
    v = id(solarman_daily_production).state;
    v_i = int(v);
    v_f = int(10*(v - v_i));

    snprintf(buf, sizeof(buf), "%i.", v_i);
    lv_label_set_text(id(solar_pd_i), buf);

    snprintf(buf, sizeof(buf), "%i", v_f);
    lv_label_set_text(id(solar_pd_f), buf);
}

// External power supply
// Current
snprintf(buf, sizeof(buf), "- - -" );
if ( id(z2_ph0_voltage).state > 170 and !isnan(id(z2_ph0_power).state) ) {
    if(!id(z2_ph0_power).state ) {
        snprintf(buf, sizeof(buf), "0" );
        lv_obj_set_style_text_color(id(supply_p), lv_color_hex(${power_ok_color}),
LV_PART_MAIN); // Ok
    }
    else {
        snprintf(buf, sizeof(buf), "%i", int(id(z2_ph0_power).state)); // get
value
        lv_obj_set_style_text_color(id(supply_p),
lv_color_hex(${power_warning_color}), LV_PART_MAIN); // Warning
    }
}
lv_label_set_text(id(supply_p), buf);

// Daily
if (id(solarman_daily_energy_bought).state !=
id(solarman_daily_energy_bought).state) {
    lv_label_set_text(id(supply_pd_i), "-."); lv_label_set_text(id(supply_pd_f),
"-");
}
else {
    v = id(solarman_daily_energy_bought).state;
    v_i = int(v);
    v_f = int(10*(v - v_i));

    snprintf( buf, sizeof(buf), "%i.", v_i );
    lv_label_set_text( id(supply_pd_i), buf );

    snprintf( buf, sizeof(buf), "%i", v_f );
    lv_label_set_text( id(supply_pd_f), buf );
}

// Battery
// Current
if (id(bat_power).state != id(bat_power).state) {

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

```

        lv_label_set_text(id(battery_p), "- - -");
    }
    else {
        v_i = int(id(bat_power).state);

        if(v_i < 0) {
            lv_obj_set_style_text_color(id(battery_p),
lv_color_hex(${power_warning_color}), LV_PART_MAIN); // Warning
        }
        else {
            lv_obj_set_style_text_color(id(battery_p),
lv_color_hex(${power_ok_color}), LV_PART_MAIN); // Ok
        }

        snprintf(buf, sizeof(buf), "%i", abs(v_i) );
        lv_label_set_text(id(battery_p), buf);
    }

    // Daily
    if (isnan(id(bat_energy).state)) {
        lv_label_set_text(id(battery_pd_i), "-.");
        lv_label_set_text(id(battery_pd_f), "-");
    }
    else {
        v = id(bat_energy).state/20;
        v_i = int(v);
        v_f = int(10*(v-v_i));

        snprintf( buf, sizeof(buf), "%i.", v_i );
        lv_label_set_text(id(battery_pd_i), buf );

        snprintf( buf, sizeof(buf), "%i", v_f );
        lv_label_set_text(id(battery_pd_f), buf );
    }

    // House consumption
    // Current
    if (isnan(id(z2_ph1_power).state)) {
        lv_label_set_text(id(consum_p), "- - -");
    }
    else {
        if (id(z2_ph1_power).state >= 4900) {
            lv_obj_set_style_text_color(id(consum_p),
lv_color_hex(${power_warning_color}), LV_PART_MAIN); // Warning
        }
        else if (id(z2_ph1_power).state >= 1900) {
            lv_obj_set_style_text_color(id(consum_p),
lv_color_hex(${power_norm_color}), LV_PART_MAIN); // Norm
        }
        else {
            lv_obj_set_style_text_color(id(consum_p), lv_color_hex(0xffffffff),
LV_PART_MAIN); // lite
        }

        snprintf( buf, sizeof(buf), "%i", int(id(z2_ph1_power).state));
        lv_label_set_text( id(consum_p), buf );
    }

    // Daily
    if (isnan(id(solarman_daily_load_consumption).state)) {
        lv_label_set_text(id(consum_pd_i), "-.");
        lv_label_set_text(id(consum_pd_f), "-");
    }
}

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

```

else {
    v = id(solarman_daily_load_consumption).state;
    v_i = int(v);
    v_f = int(10*(v - v_i));

    snprintf( buf, sizeof(buf), "%i.", v_i );
    lv_label_set_text( id(consum_pd_i), buf );

    snprintf( buf, sizeof(buf), "%i", v_f );
    lv_label_set_text( id(consum_pd_f), buf );
}
- id: update_time
then:
- lambda: |-
    auto time = id(sntp_time).now();
    static int st_day_of_week = -1;
    static const char* time_d[7]={"Нд","Пн","Вт","Ср","Чт","Пт","Сб"};
    static char buf[20];

    id(clock_time_show_dots) = !id(clock_time_show_dots);

    // Time hh:mm
    snprintf(buf, sizeof(buf), "%02d %02d",time.hour, time.minute);
    lv_label_set_text( id(clock_time), buf );

    // Date
    snprintf(buf, sizeof(buf), "%2d.%02d", time.day_of_month, time.month);
    lv_label_set_text( id(clock_date), buf );

    // Day of week
    if (st_day_of_week != time.day_of_week) {
        st_day_of_week = time.day_of_week;
        lv_label_set_text(id(clock_day_of_week), time_d[time.day_of_week - 1]);

        // Color
        if (time.day_of_week == 1 or time.day_of_week == 7 ) {
            lv_obj_set_style_text_color(id(clock_day_of_week),
lv_color_hex(${clock_day_holi_color}), LV_PART_MAIN); // Holidays
        }
        else {
            lv_obj_set_style_text_color(id(clock_day_of_week),
lv_color_hex(${clock_day_norm_color}), LV_PART_MAIN); // Usual
        }
    }
- id: update_time_dots
then:
- lambda: |-
    id(clock_time_show_dots) = !id(clock_time_show_dots);

    if (id(clock_time_show_dots)) {
        lv_label_set_text( id(clock_time_dots), ":" );
    }
    else {
        lv_label_set_text( id(clock_time_dots), " " );
    }
- id: update_weather
then:
- lambda: |-
    static char buf[11];

    // Temperature
    if (isnan(id(forecast_temp).state)) { sprintf(buf, "-"); }
    else { sprintf(buf, "%.0f", id(forecast_temp).state); }
    lv_label_set_text(id(weather_temp), buf);

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

```

// Wet
if (isnan(id(forecast_wet).state)) { sprintf(buf, "-"); }
else { sprintf(buf, "%.0f", id(forecast_wet).state); }
lv_label_set_text(id(weather_wet), buf);

// Cloud
if (isnan(id(forecast_cloud).state)) { sprintf(buf, "-"); }
else { sprintf(buf, "%.0f", id(forecast_cloud).state); }
lv_label_set_text(id(weather_cloud), buf);

// Weather icon
auto time = id(sntp_time).now();
int weather_state_cnv = -1;

if (id(weather_state).state == "clear-night")           {weather_state_cnv = 0;}
else if (id(weather_state).state == "cloudy")           {weather_state_cnv = 1;}
else if (id(weather_state).state == "fog")              {weather_state_cnv = 2;}
else if (id(weather_state).state == "hail")             {weather_state_cnv = 3;}
else if (id(weather_state).state == "lightning")       {weather_state_cnv = 4;}
else if (id(weather_state).state == "lightning-rainy") {weather_state_cnv = 5;}
else if (id(weather_state).state == "partlycloudy")    {
    if (time.hour >= 6 and time.hour <= 20) { weather_state_cnv = 61; }
    else { weather_state_cnv = 62; }
}
else if (id(weather_state).state == "pouring")          {weather_state_cnv = 7;}
else if (id(weather_state).state == "rainy")           {weather_state_cnv = 8;}
else if (id(weather_state).state == "snowy")           {weather_state_cnv = 9;}
else if (id(weather_state).state == "snowy-rainy")    {weather_state_cnv = 10;}
else if (id(weather_state).state == "sunny")           {weather_state_cnv = 11;}
else if (id(weather_state).state == "windy")           {weather_state_cnv = 12;}
else if (id(weather_state).state == "windy-variant")  {weather_state_cnv = 13;}
else if (id(weather_state).state == "exceptional")    {weather_state_cnv = 14;}

if (weather_state_cnv != id(weather_state_old)) {
    id(weather_state_old) = weather_state_cnv;

    switch(weather_state_cnv) {
        case 0: lv_img_set_src(id(wheather_image), id(img_clear_night)); break;
        case 1: lv_img_set_src(id(wheather_image), id(img_cloudy)); break;
        case 2: lv_img_set_src(id(wheather_image), id(img_fog)); break;
        case 3: lv_img_set_src(id(wheather_image), id(img_hail)); break;
        case 4: lv_img_set_src(id(wheather_image), id(img_lightning)); break;
        case 5: lv_img_set_src(id(wheather_image), id(img_lightning_rainy));
break;
        case 6: lv_img_set_src(id(wheather_image), id(img_partlycloudy)); break;
        case 61: lv_img_set_src(id(wheather_image), id(img_partlycloudy_day));
break;
        case 62: lv_img_set_src(id(wheather_image), id(img_partlycloudy_night));
break;
        case 7: lv_img_set_src(id(wheather_image), id(img_pouring)); break;
        case 8: lv_img_set_src(id(wheather_image), id(img_rainy)); break;
        case 9: lv_img_set_src(id(wheather_image), id(img_snowy)); break;
        case 10: lv_img_set_src(id(wheather_image), id(img_snowy_rainy)); break;
        case 11: lv_img_set_src(id(wheather_image), id(img_sunny)); break;
        case 12: lv_img_set_src(id(wheather_image), id(img_windy)); break;
        case 13: lv_img_set_src(id(wheather_image), id(img_windy_variant)); break;
        // case 14: lv_img_set_src(id(wheather_image), id(img_exceptional));
break;
    }
}

- id: update_temp
then:

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

```

- lambda: |-
    static bool current_warning_show = true;
    static bool current_error_show = true;
    bool warning_show = false;
    bool error_show = false;
    static char buf[11];

    // Room
    if (isnan(id(sensor_temp_room).state)) { sprintf(buf, "-"); }
    else { sprintf(buf, "%.0f", id(sensor_temp_room).state); }
    lv_label_set_text(id(l_temp_room), buf);

    // Boiler 1
    if (isnan(id(sensor_temp_boiler_1).state)) { sprintf(buf, "-"); }
    else { sprintf(buf, "%.0f", id(sensor_temp_boiler_1).state); }
    lv_label_set_text(id(l_temp_boiler_1), buf);

    // Boiler 1 Status
    if (isnan(id(boiler_1_work).state)) {
        lv_obj_set_style_bg_color(id(temp_boiler_1_status),
lv_color_hex(${temp_status_err_color}), LV_PART_MAIN); // Error
    }
    else if (id(boiler_1_work).state and id(boiler_1_power).state >= 10) {
        lv_obj_set_style_bg_color(id(temp_boiler_1_status),
lv_color_hex(${temp_status_on_color}), LV_PART_MAIN); // On
    }
    else {
        lv_obj_set_style_bg_color(id(temp_boiler_1_status),
lv_color_hex(${temp_status_off_color}), LV_PART_MAIN); // Off
    }

    // Boiler 2
    if (isnan(id(sensor_temp_boiler_2).state)) { sprintf(buf, "-"); }
    else { sprintf(buf, "%.0f", id(sensor_temp_boiler_2).state); }
    lv_label_set_text(id(l_temp_boiler_2), buf);

    // Boiler 2 Status
    if (isnan(id(boiler_2_work).state)) {
        lv_obj_set_style_bg_color(id(temp_boiler_2_status),
lv_color_hex(${temp_status_err_color}), LV_PART_MAIN); // Error
    }
    else if (id(boiler_2_work).state and id(boiler_2_power).state >= 10) {
        lv_obj_set_style_bg_color(id(temp_boiler_2_status),
lv_color_hex(${temp_status_on_color}), LV_PART_MAIN); // On
    }
    else {
        lv_obj_set_style_bg_color(id(temp_boiler_2_status),
lv_color_hex(${temp_status_off_color}), LV_PART_MAIN); // Off
    }

    // Water
    if (isnan(id(sensor_temp_water).state)) { sprintf(buf, "-"); }
    else { sprintf(buf, "%.0f", id(sensor_temp_water).state); }
    lv_label_set_text(id(l_temp_water), buf);

    // Freezer
    if (isnan(id(sensor_temp_freezer).state)) { sprintf(buf, "-"); }
    else { sprintf(buf, "%.0f", id(sensor_temp_freezer).state); }
    lv_label_set_text(id(l_temp_freezer), buf);

    // Freezer Font Color
    if (id(sensor_temp_freezer).state <= -10.0) {
        lv_obj_set_style_text_color(id(l_temp_freezer),
lv_color_hex(${temp_cold_color}), LV_PART_MAIN);

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

```

    }
    else if (id(sensor_temp_freezer).state <= -5.0) {
        warning_show = true;
        lv_obj_set_style_text_color(id(l_temp_freezer), lv_color_hex(0xFFFFFFFF),
LV_PART_MAIN);
    }
    else if (id(sensor_temp_freezer).state <= 0.0) {
        error_show = true;
        lv_obj_set_style_text_color(id(l_temp_freezer),
lv_color_hex(${temp_warm_color}), LV_PART_MAIN);
    }
    else {
        lv_obj_set_style_text_color(id(l_temp_freezer),
lv_color_hex(${temp_warning_color}), LV_PART_MAIN);
    }

    // Freezer Status
    if (isnan(id(freezer_work).state)) {
        lv_obj_set_style_bg_color(id(temp_freezer_status),
lv_color_hex(${temp_status_err_color}), LV_PART_MAIN); // Error
    }
    else if (id(freezer_work).state and id(freezer_power).state >= 10) {
        lv_obj_set_style_bg_color(id(temp_freezer_status),
lv_color_hex(${temp_status_on_color}), LV_PART_MAIN); // On
    }
    else {
        lv_obj_set_style_bg_color(id(temp_freezer_status),
lv_color_hex(${temp_status_off_color}), LV_PART_MAIN); // Off

        if (id(sensor_temp_freezer).state >= -7.5) {
            error_show = true;
        }
    }

    // Kettle Air
    if (isnan(id(sensor_temp_kettle_air).state)) { sprintf(buf, "-"); }
    else { sprintf(buf, "%.0f", id(sensor_temp_kettle_air).state); }
    lv_label_set_text(id(l_temp_kettle_air), buf);

    // Kettle Air Font Color
    if (id(sensor_temp_kettle_air).state <= 15.0) {
        lv_obj_set_style_text_color(id(l_temp_kettle_air),
lv_color_hex(${temp_cold_color}), LV_PART_MAIN);
    }
    else if (id(sensor_temp_kettle_air).state <= 30.0) {
        lv_obj_set_style_text_color(id(l_temp_kettle_air), lv_color_hex(0xFFFFFFFF),
LV_PART_MAIN);
    }
    else if (id(sensor_temp_kettle_air).state <= 65.0) {
        lv_obj_set_style_text_color(id(l_temp_kettle_air),
lv_color_hex(${temp_warm_color}), LV_PART_MAIN);
    }
    else {
        lv_obj_set_style_text_color(id(l_temp_kettle_air),
lv_color_hex(${temp_hot_color}), LV_PART_MAIN);
    }

    // Kettle Status
    if (isnan(id(kettle_work).state)) {
        lv_obj_set_style_bg_color(id(temp_kettle_status),
lv_color_hex(${temp_status_err_color}), LV_PART_MAIN); // Error
    }
    else if (id(kettle_work).state) {

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

```

lv_obj_set_style_bg_color(id(temp_kettle_status),
lv_color_hex(${temp_status_on_color}), LV_PART_MAIN); // On

if (!id(kettle_old_work)) {
    id(kettle_old_work) = true;
    id(kettle_time_on) = millis();
}

if (id(sensor_temp_kettle_air).state <= 75.0 and millis() - id(kettle_time_on)
>= 60000) {
    error_show = true;
}
else if (id(sensor_temp_kettle_air).state <= 65.0 and millis() -
id(kettle_time_on) >= 30000) {
    warning_show = true;
}
else {
    lv_obj_set_style_bg_color(id(temp_kettle_status),
lv_color_hex(${temp_status_off_color}), LV_PART_MAIN); // Off
    id(kettle_old_work) = false;
}

// Kettle Water In
if (isnan(id(sensor_temp_kettle_water_in).state)) { sprintf(buf, "-"); }
else { sprintf(buf, "%.0f", id(sensor_temp_kettle_water_in).state); }
lv_label_set_text(id(l_temp_kettle_water_in), buf);

// Kettle Water Out
if (isnan(id(sensor_temp_kettle_water_out).state)) { sprintf(buf, "-"); }
else { sprintf(buf, "%.0f", id(sensor_temp_kettle_water_out).state); }
lv_label_set_text(id(l_temp_kettle_water_out), buf);

if (id(sensor_temp_kettle_water_out).state <= 15.0) {
    lv_obj_set_style_text_color(id(l_temp_kettle_water_out),
lv_color_hex(${temp_cold_color}), LV_PART_MAIN);
}
else if (id(sensor_temp_kettle_water_out).state <= 30.0) {
    lv_obj_set_style_text_color(id(l_temp_kettle_water_out),
lv_color_hex(0xFF0000), LV_PART_MAIN);
}
else if (id(sensor_temp_kettle_water_out).state <= 65.0) {
    lv_obj_set_style_text_color(id(l_temp_kettle_water_out),
lv_color_hex(${temp_warm_color}), LV_PART_MAIN);
}
else {
    lv_obj_set_style_text_color(id(l_temp_kettle_water_out),
lv_color_hex(${temp_hot_color}), LV_PART_MAIN);
}

// Warning
if (!error_show and current_warning_show != warning_show) {
    current_warning_show = warning_show;
    if (warning_show) {
        lv_obj_clear_flag(id(temp_warning_1), LV_OBJ_FLAG_HIDDEN);
        lv_obj_clear_flag(id(temp_warning_2), LV_OBJ_FLAG_HIDDEN);
    }
    else {
        lv_obj_add_flag(id(temp_warning_1), LV_OBJ_FLAG_HIDDEN);
        lv_obj_add_flag(id(temp_warning_2), LV_OBJ_FLAG_HIDDEN);
    }
}
// Error

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

```

if (current_error_show != error_show) {
    current_error_show = error_show;
    if (error_show) {
        lv_obj_clear_flag(id(temp_error_1), LV_OBJ_FLAG_HIDDEN);
        lv_obj_clear_flag(id(temp_error_2), LV_OBJ_FLAG_HIDDEN);
    }
    else {
        lv_obj_add_flag(id(temp_error_1), LV_OBJ_FLAG_HIDDEN);
        lv_obj_add_flag(id(temp_error_2), LV_OBJ_FLAG_HIDDEN);
    }
}

```

```

- id: update_slide_page
then:

```

```

- lambda: |-
    // Slide page
    if (id(slide_page_mode)) {
        id(slide_page_time) += 1;

        if (id(slide_page_time) >= ${slide_page_show_time}) {
            id(slide_page_time) = 0;
            int next_page_id;

            if (id(current_page) == 5) {
                next_page_id = 1;
            }
            else {
                next_page_id = 1 + (id(current_page))%4;
            }

            switch(next_page_id) {
                case 1:
                    id(show_page_1).execute();
                    break;
                case 2:
                    id(show_page_2).execute();
                    break;
                case 3:
                    id(show_page_3).execute();
                    break;
                case 4:
                    id(show_page_4).execute();
                    break;
            }
        }
    }
}

```

```

interval:

```

```

- interval: 1s
then:
- lambda: |-
    switch(id(current_page)) {
        case 1:
            id(update_page_1).execute();
            break;
        case 2:
            id(update_temp).execute();
            break;
        case 4:
            id(update_time).execute();
            id(update_weather).execute();
            break;
    }
}

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

```
id(update_slide_page).execute();  
- interval: 0.5s  
  then:  
    - lambda: |-  
      if (id(current_page) == 4) {  
        id(update_time_dots).execute();  
      }
```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58