

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКІЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ
ВОЛОДИМИРА ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА програмування ТА МАТЕМАТИКИ

Пояснювальна записка

до дипломної роботи

бакалавр

(Освітньо-кваліфікаційний рівень)

**На тему: «Комп'ютерна система управління побутовим
твердопаливним котлом»**

Виконав: студент 4 курсу, групи КІ-17

Напряму підготовки: 6.050102 «Комп'ютерна
інженерія»

_____Тарабановський Д.Т.
(Підпис)

Керівник,

доцент, к.т.н., _____Марченко Д.М.
(Підпис)

Рецензент,

доцент, к.т.н., _____Іванов В.Г.
(Підпис)

Севєродонецьк 2021

СХІДНОУКРАЇНСЬКІЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет інформаційних технологій та електроніки

Кафедра програмування та математики

Освітньо-кваліфікаційний рівень бакалавр

Напрямок підготовки 6.050102 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри ПМ,

д.т.н., доцент

_____ Лифар В.О.

«___» _____ 2021 р.

завдання

НА дипломних роботах СТУДЕНТА

Тарабановський Дмитро Тимофійович

1. Тема роботи: « Комп'ютерна система управління побутовим твердопаливним котлом »

Керівник роботи Марченко Д.М.

ЗАТВЕРДЖЕНІ наказом ВИЩОГО навчального закладу від «___» _____ 2021 року № ___

Строк подання студентом роботи 18 червня 2021 р.

Вихідні дані (технічні вимоги): створення програмно

апаратного модуля для автоматизації роботи твердопаливного котла та мати можливість керувати ним дистанційно.

Зміст ВКР: вибір апаратної бази для створення пристрою керування

твердопаливним котлом, створення режимів роботи пристрою, організація роботи пристрою через хмарне середовище за допомогою додатку Blynk.

Перелік звітних матеріалів: Пояснювальна записка, ілюстративний матеріал.

Дата видачі завдання: 22 березня 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з / п	Назва етапів дипломної роботи	рядків Виконання етапів роботи	Примітка
1	Огляд літератури за темою роботи	28.03.2021	
2	Вибір апаратної бази для створення автоматизованої системи управління котлом	4.04.2021	
3	Програмування режимів роботи	22.04.2021	
4	Організація єдиної системи управління котлом через хмарне середовище Vlynk	11.05.2021	
5	Оформлення ілюстративного матеріалу	22.05.2021	
6	Оформлення пояснювальної записки	07.06.2021	

студент

(Підпис)

Керівник роботи

(Підпис)

ЛИСТ ПОГОДЖЕННЯ і оцінювання

Дипломної роботи студента гр. КІ-17 Тарабановського Д.Т.

Керівник роботи, доцент, к.т.н. . _____ Марченко Д.М.

Оцінка керівника: _____

рецензент

(ПІБ, місто роботи, посада)

Оцінка рецензента: _____

Кінцева оцінка за результатами захисту:

Голова ЕК _____ Захожай О.І.

РЕФЕРАТ

Пояснювальна записка містить: 61 сторінок., 47 рисунків., 1 таблиця., 14 джерел за переліком посилань.

ARDUINO, ESP8266, BLUNK , АВТОМАТИЗАЦІЯ, ПРИЛАДИ ДЛЯ ВИМІРУВАННЯ ВЕЛИЧИН, NODE MCU LUA, СЕРВОПРИВІД.

Мета роботи: протестувати і вибрати апаратне забезпечення для розробки та програмування пристрою , який зможе автоматизувати роботу твердопаливного котла.

Пристрій який виготовлений у даній кваліфікаційній роботі показує, що можна автоматизувати будь-які процеси у нашому житті.

На сьогодні люди які проживають в селах , мають можливість економити власні гроші за рахунок використання твердопаливних котлів. Але котли с повністю автоматичними системами коштують досить дорого. У людей немає іншої можливості, як купляти більш дешеві котли ,але не такі автоматизовані та безпечні.

При розробки цієї системи було порівняно дві основи для створення цього приладу ,а саме Arduino і Esp8266.

Розглянуто можливості і характеристики використаних приладів ,а саме серво-привід та датчик температури.

Використано додаток Blynk для відображення та керування ,з урахування усіх особливостей платформи на якій будуємо пристрій.

Дана ідея відтворена фізично та буде показана в дії.

СКОРОЧЕННЯ ТЕРМІНІВ

У цьому розділі будуть пояснення скорочених термінів та пояснення деяких термінів.

Arduino - це торгова марка апаратного і програмного забезпечення для створення приладів, для автоматизації процесів у різних сферах.

ESP8266- мікросхема с підтримкою Wi-fi.

ОЗУ-вбудована оперативна пам'ять пристрою.

Бібліотеки-необхідний програмний код для роботи с тим чи іншим датчиком.

Blynk-хмарне середовище для автоматизації процесів у галузі електроніки та робототехніки.

Arduino IDE-програмне середовище для програмування основних та керуючих функцій пристроїв.

Піни- роз'єми для підключення датчиків .

Shields- можливість плат Arduino з'єднуватися методом бутерброду.

PARASIT POWER CIRCUIT-режим «паразитного» живлення .

UART- вузол для організації зв'язку с іншими цифровими простоями.

1-Wire-шина для зв'язку пристроїв та мікроконтролерів які мають свою унікальну адресу .

Зміст

ВСТУП.....	8
1. Аналітичний огляд.....	9
2. Апаратне та програмне забезпечення автоматизованого котла	13
2.1 Автоматизація процесів твердопаливного котла.	13
3. Вибір мікроконтролера.....	16
3.1 Arduino.....	16
3.2 ESP8266.....	18
4. Апаратна частина.....	24
4.1 DS18B20.....	24
4.2 Технічні характеристики датчику DS18B20.....	27
4.3 Бібліотеки.....	29
4.4 Серво привід SG90.....	31
4.5 Резистор 4.7 КОМ.....	33
5. Програмне забезпечення.....	35
5.1 Arduino IDE.....	35
5.2 Blynk.....	41
6. Програмний код.....	45
7. Алгоритм роботи.....	52
Матеріальні витрати для реалізації проекту.....	53
ВИСНОВОК.....	54
Список використаних джерел.....	55
Додаток А.....	57

ВСТУП

На даний момент суспільство все більше переходить до автоматизації в тих чи інших сферах діяльності. Роботи та прилади замінюють людей і роботу людей майже в усіх сферах діяльності. Починаючи від промислових заводів, де зварюються кузовні деталі автомобілів закінчуючи кондитерськими роботами ,які випікають випічку. А саме головне те, що все це робиться дистанційно , а інколи навіть з іншого куточку світу.

Застосування ІТ технологій уже не є чимось дивним та не звичним. І це дуже добре, адже всі процеси які людина намагається автоматизувати забирають у неї той дорогоцінний час, а інколи і здоров'я ,який людина витрачає для виконання тієї чи іншої роботи. Ми живемо у світі технологій, де автомобілі їздять без участі людини, заплатити у кафе можливо телефоном або навіть Bitcoin.

Метою даної роботи була автоматизація процесів твердопаливного котла, яких на даний час дуже багато , але не всі вони працюють в автоматичному режимі.

Використання автоматики значно спрощує роботу котла.

Для цього встановлюється блок для управління такими процесами ,як контроль температури ,положення повітряного засува.

Це над сам перед низька ціна палива с високою тепловіддачею, вони прості та ефективні , але всі вони залежать від людини яка використовує його.

У даній роботі представлено один із варіантів автоматизації низки процесів твердопаливного котла.

1. Аналітичний огляд

1.1 Формулювання актуальності, мети і завдань проекту

Ми стоїмо на порозі цифрової революції. В даній дипломній роботі було розроблено систему контролю температури твердопаливного котла.

Даний пристрій дає можливість керувати, засовом котла для регулювання температури в ньому та в цілому у приміщенні.

Керування відбувається за допомогою мікропроцесорної системи з дистанційним керуванням через мережу інтернет.

Автоматизація твердопаливних котлів зараз дуже актуальна, адже зі зростанням ціни на блакитне пальне попит на котли дуже зріс.

Частіше за все твердопаливні котли встановлюються в селах та селищах міського типу. Головна причина такого розміщення, наявність твердого пального в достатній кількості.

Це дозволяє людям економити свій бюджет. Але вартість котлів теж досить не мала.



Рисунок 1.1 – Ціна котла з автоматикою та без

Як ми бачимо вартість котла з вбудованою автоматикою коштує набагато більше ніж котел без вбудованої автоматики.

Люди шукають інші рішення і встановлюють автоматику від сторонніх виробників. Отже, це питання досить актуальне на даний час.

Давайте подивимось, які рішення автоматики присутні на даний час на ринку.



Рисунок 1.2 – Контролер для твердопаливного котла TECH ST-81 zPID
* (ST-81 zPID *)



Рисунок 1.3 – Автоматика для твердопаливних котлів Kom-ster Atos

Бувають і механічні регулятори температури для котла, але їх функціональність не дуже виправдана ціною.

Так як людині потрібно знаходитися поряд с котлом для регулювання температури.



Рисунок 1.4 – Механічний регулятор температури

Навіщо взагалі встановлюють регулятор температури?

Твердопаливний котел це не звичайна піч яка може витримувати майже будь які температури без наслідків. За температури 100 градусів котел має властивість кипіти та перетворювати воду на пару, яка в свою чергу зруйнує водяну сорочку котла.

Отже регулювання котла в ручну небезпечно. Якщо вчасно не прикрити засов то отримаєте перегрів і вибух.

Крім безпеки регулятор тяги виконує ще дві задачі:

-закриваючи засов та зменшуючи тягу та надходження повітря до місця згорання палива котел зменшує горіння і дає можливість довше опалювати приміщення . Але при зменшенні горіння палива зменшується і КПД теплоносія.

-для опалювання будинку достатньо температури 60-70 градусів температура 80-90 потрібна хіба що в найхолодніші дні.

Звісно регулятор тяги для твердопаливного котла не є досить точним , бо процес тління зупиняється не миттєво та може проскочити налаштовану межу температури на 5-10 градусів після закриття дверцят.

Блок управління автоматики працює у зв'язці з засовом та датчиком температури. Регулятори температури використовуються з метою оптимізації та автоматизації функціонування твердопаливних котлів.

Після вивчення будови та принципу дії автоматики, переходимо до вибору апаратної бази , яка дасть можливість управляти процесом керування засовом відносно температури .

Мета: розробити систему для регулювання температури котла, з використанням мікроконтролера та хмарного середовища для дистанційного управління з телефону

Головним завданням: проекту було створення такого пристрою, який би не поступався за функціональною можливістю вже існуючих на ринку , але ціна його була доступною для кожної людини.

2. Апаратне та програмне забезпечення автоматизованого котла .

З урахуванням можливостей мікроконтролера та датчиків , які будуть використані для реалізації даного проекту, для автоматизації роботи повітряного засова у твердопаливному котлі було обрано програмне та апаратне забезпечення.

2.1 Автоматизація процесів твердопаливного котла.

Використання твердопаливних котлів здобуло широкого розповсюдження через дешеве паливо та сировину, а також високу тепловіддачу. Головна перевага твердопаливних котлів у тому , що можна використовувати різні джерела тепла тобто котел може працювати на різних видах палива.

Використання автоматизації значно зменшує час та сили для обслуговування котла.

Полегшує та дозволяє швидко виконувати ті чи інші процеси. А в сучасних автоматизаціях навіть через інтернет.

Основна задача цієї роботи автоматизувати підтримку температури без втручання людини в процес.

Даний пристрій буде мати ручний режим роботи:

-Ручний(який в свою чергу поділяється на три режиму)

Ручний режим має в своєму розпорядженні три різних режиму, які будуть регулювати використання палива в котлі:

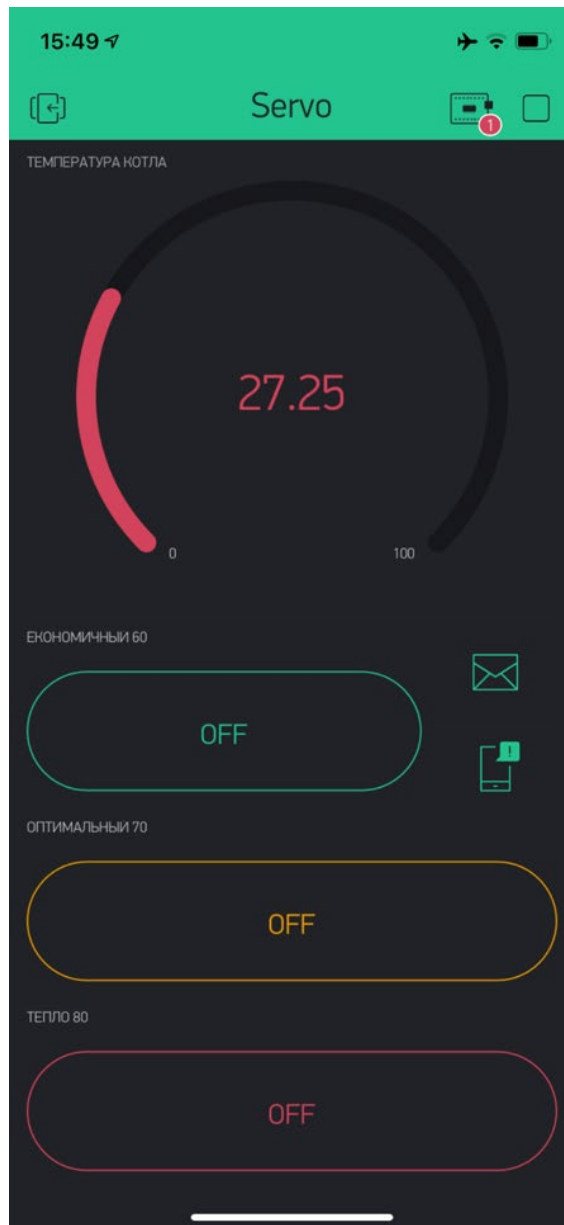


Рисунок 2.1 – Інтерфейс додатку

-Економічний(Еко)- підтримка температурного режиму у котлі 60 С

-Оптимальний(Опт)-підтримка температурного режиму у котлі 70 С

-Тепло- підтримка температурного режиму у котлі 80 С.

Це питання зараз дуже актуальне через високу вартість газу.

Даний пристрій буде керуватися с мобільного телефону по каналу зв'язку WIFI.

Інтерфейс програми був складений та запрограмований у додатку BLYNK.

Вlynk- це апаратно програмна платформа IoT з мобільними додатками, приватним хмарним простором , управління пристроями та аналітикою даних та машинним навчанням.

Основна задача приладу буде регулювання повітряного засову для регулювання температури в котлу.

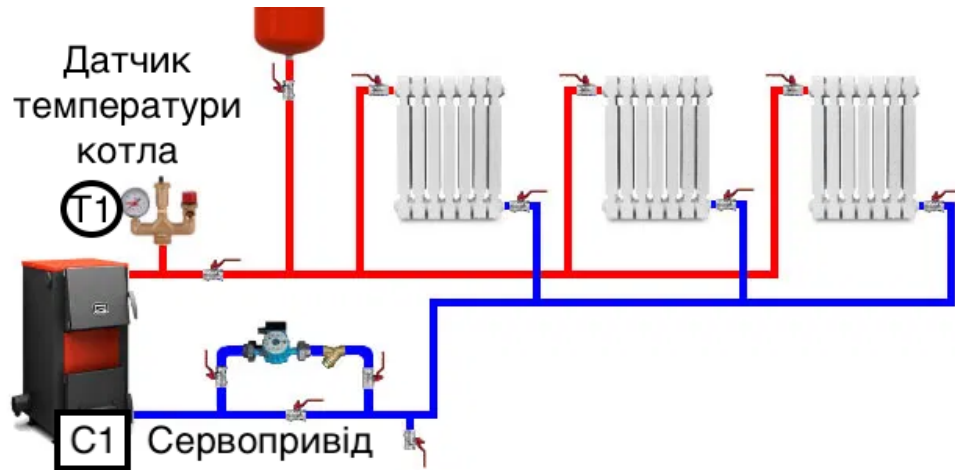


Рисунок 2.2 – Схема опалення

3.Вибір мікроконтролера

Для даного дипломного проекту було обрано мікроконтролер ESP8266
вибір був між Arduino nano та ESP8266.

3.1Arduino

Arduino – це електронний конструктор для створення пристроїв для різних потреб. Arduino- дуже широко розповсюджена в усіх країнах світу через свою простоту, та програмуванні на мові C++.

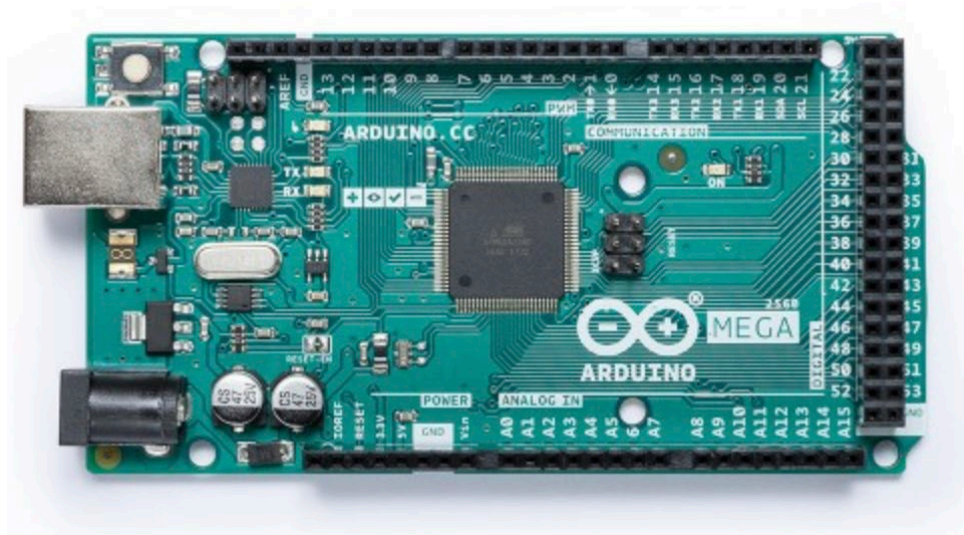


Рисунок 3.1 – Arduino Mega

Пристрій- програмується через порт USB без використання програматорів.
В плату Arduino можна завантажити будь яку програму, яка буде керувати тими чи іншими пристроями за алгоритмом.

За допомогою такого програмного різноманіття та підключення різних датчиків можна робити дуже багато різних гаджетів.

Щоб зрозуміти логіку роботи далі прикладено малюнок.

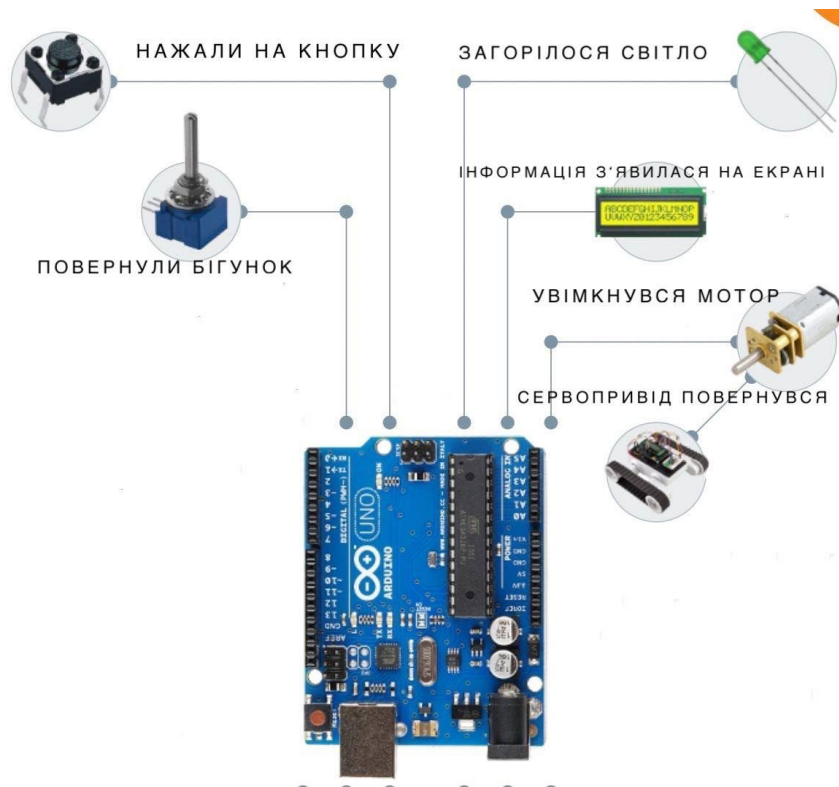


Рисунок 3.2 – Логіка роботи

Логіка роботи зрозуміла, далі нам потрібно програмне середовище Arduino IDE.



Рисунок 3.3 – Інтерфейс Arduino IDE

Ще одна особливість Arduino це наявність так званих shields плат розширення. Це можливість розташовувати пристрої та датчика за принципом бутерброду.

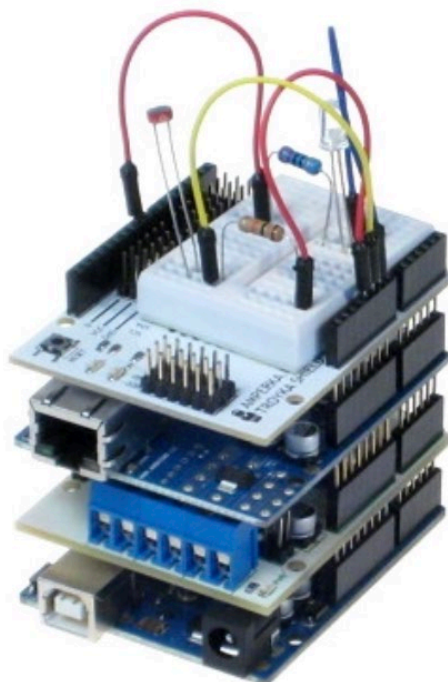


Рисунок 3.4 – З'єднання за допомогою shields

3.2 ESP8266



Рисунок 3.5 – Окремий модуль ESP8266

Мікроконтролер китайського виробника з інтерфейсом Wi-Fi основна відмінність між ними ,що мікроконтролер ESP8266 немає флеш пам'яті.

Програми користувача можуть бути виконані тільки з зовнішньої флеш карти. Особливу увагу мікроконтролер здобув у 2014-му році, коли вийшли перші продукти на його базі. Мікроконтролер виготовляється без кристала.

Виконання програми може бути тільки із зовнішньої ПЗУ

шляхом завантаження деяких кусочків виконуваної програми в кеш.

Але користувач не бачить цього завантаження, це відбувається дуже швидко. Мікроконтролер може мати SD карту найбільшою ємністю 16 МБ.

Головна перевага ESP8266 перед Arduino nano те, що мікроконтролер ESP8266 має вбудований Wi-Fi модуль. Зараз налічується декілька модифікацій ESP8266 які мають назву від ESP-01 до ESP-12. На даний момент з'явилися і другі назви від інших розробників. Головна відмінність в мікроконтролерах в портах введення та виводу інформації, кількості зовнішньої пам'яті.

У процесорі відмінностей немає, тому їх дуже просто програмувати с точки зору програмування.

ESP8266 це пристрій з підтримкою Wi-Fi це означає, що він може підключатися через Wi-Fi до вашого телефону або роутеру.

Але перед тим як підключити ваш телефон або роутер в прошивку даного мікроконтролера потрібно записати ім'я вашої локальної мережі та пароль. Також мікроконтролер може працювати у режимі роутера.

Розробники мікроконтролеру створили спеціальний послідовний порт(UART) для полегшення роботи та програмування мікроконтролеру.

Для підключення до ПК та програмування мікроконтролера використовується спеціальний перехідник USB-UART.

Такий вид підключення надає свої переваги, а саме можна відправляти команди з терміналу та керувати пристроєм, та отримувати відкладку від мікроконтролеру.

Даний мікроконтролер може використовуватися без зовнішнього живлення тобто через порт USB, але тільки якщо пристрій має малу кількість датчиків, коли ж потрібно використовувати велику кількість датчиків потрібно використовувати зовнішня живлення 5 В.

Щоб підключити мікроконтролер через порт підключення UART потрібно в налаштуваннях встановити однакову швидкість портів, але не обов'язково, бо при підключенні мікроконтролер намагається з'ясувати, яка швидкість встановлена та встановити собі таку ж.

Найчастіше для того щоб передати модулю ESP8266 використовують AT-команди.

AT-команди-це спеціалізований набір слів, які розуміє наш мікроконтролер та може виконувати ці команди, а отримані результати видавати в термінал після закінчення виконання.

Також модуля ESP8266 є можливість запускати та записувати власні додатки. Код для програмування використовуються на мові C.

Для чого ж потрібно модуль Wi-Fi? Для того щоб користувачі даного пристрою могли писати програми які будуть автоматизувати якісь процеси та ними можна було керувати дистанційно з іншої точки земної кулі.

Для програмування мікроконтролера зручно використовувати редактор коду ним може бути Visual Studio або ж Arduino IDE.

Нижче наведені характеристики мікроконтролеру.

Специфікація ESP8266:

Напруга живлення: 3,3 В

Енергоспоживання: 10 мка ... 170 мА

Флеш-пам'ять: до 16 мб максимум (зазвичай 512 кб)

Процесор: Tensilica L106, 32 біта

Швидкість процесора: 80 ... 160 МГц

ОЗУ: 32 кб + 80 кб

Порти введення-виведення загального призначення: 17

(мультиплексируемость з іншими функціями)

АЦП: 1 введення з роздільною здатністю 1024

Підтримка 802.11: b / g / n / d / e / i / k / r

Максимальне число підключень TCP: 5

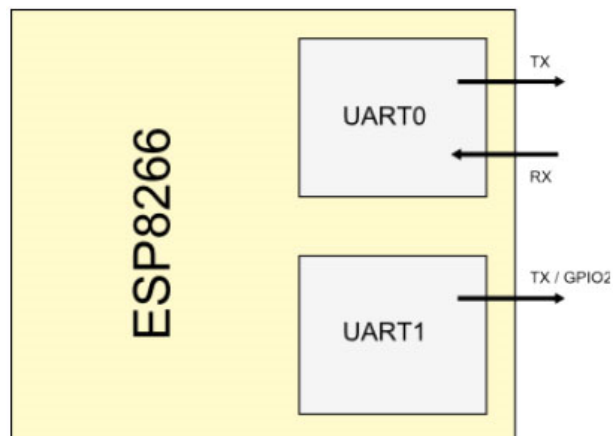


Рисунок 3.6 – Схема підключення

Сам мікроконтролер має вигляд схожий на Arduino Але з модулем Wi-Fi.

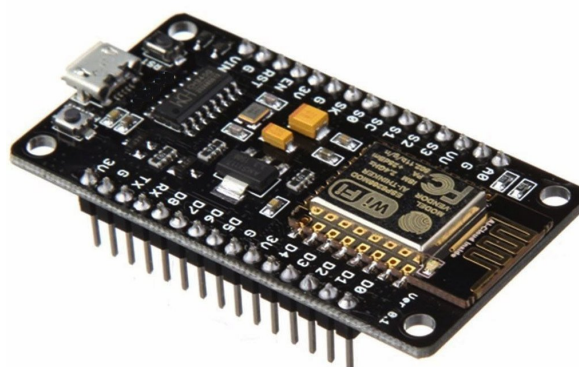


Рисунок 3.7 – ESP8266

Мікроконтролер ESP 8266 має наступну розпіновку.

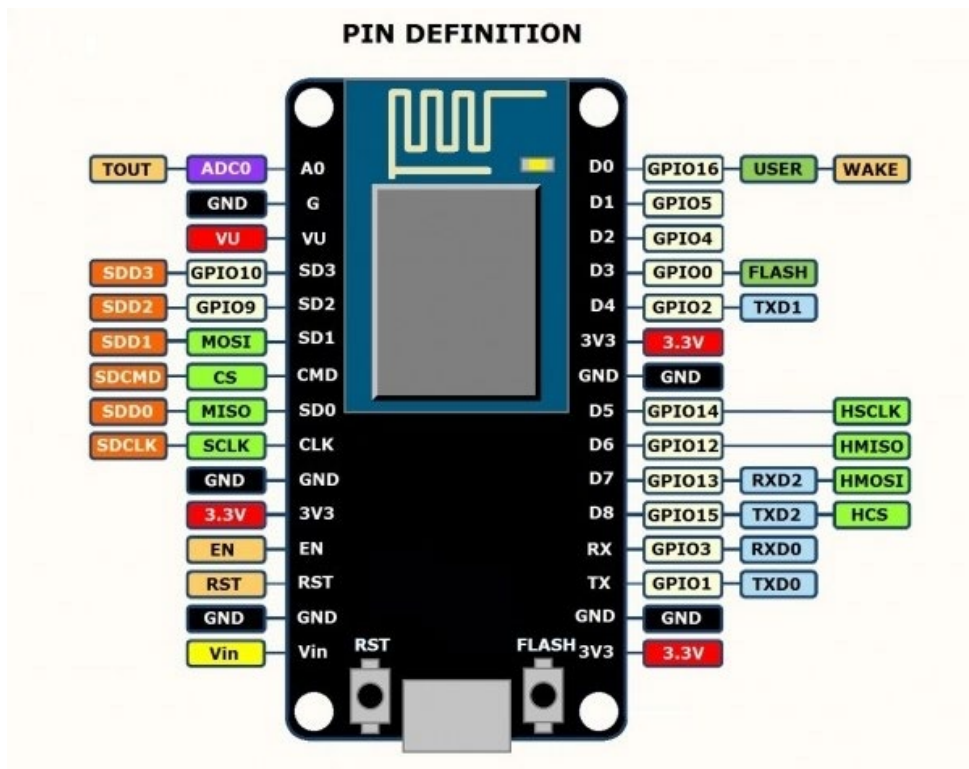


Рисунок 3.8 – Розпіновка даного пристрою ESP8266

3.3V і GND (на зображенні вище Малюнок 1.2.3) – згідно інструкції та документації на ніжку 3.3V можна подавати від 3 до 3.6 В , а GND- це земля тобто мінус живлення цю модель можна підключати до двох АА акумуляторів при цьому модель працював у звичному режимі але розробники рекомендують використовувати живлення у діапазоні від 3 до 3.6 В.

Для того ,щоб остаточно відкинути сумнів ,що використовувати Arduino або ESP8266 далі приведено порівняння цих двох мікроконтролерів.

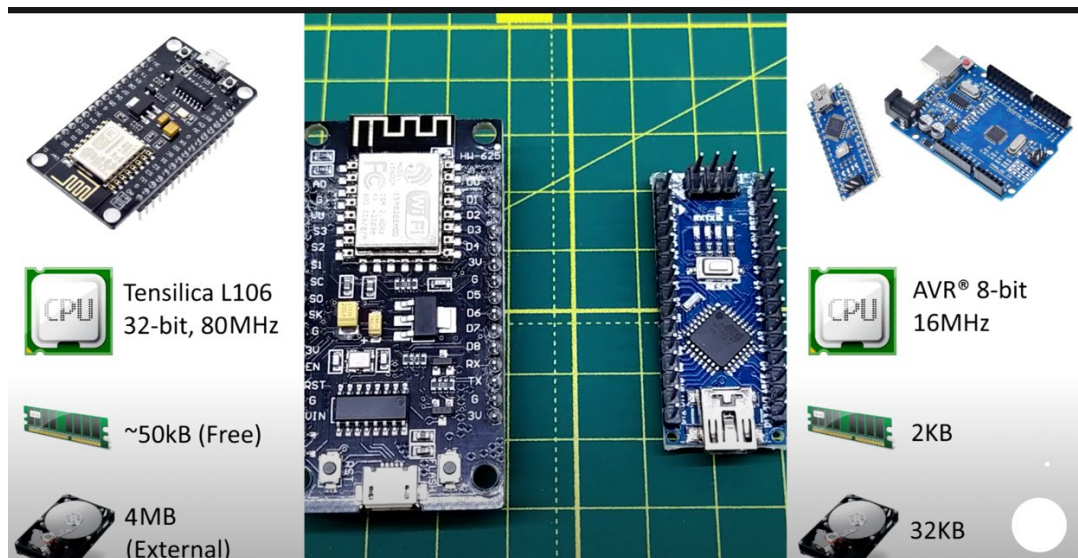


Рисунок 3.9 – Порівняння пристроїв

Та найбільша перевага мікроконтролера ESP це наявність вбудованого WiFi модулю, який буде працювати у зв'язці з додатком Blynk через сервери якого, ми будемо керувати автоматизованою системою опалення.



Рисунок 3.10 – Wi-Fi модуль

4. Апаратна частина

В даному розділі будуть описані датчики, які використовувалися для створення даної автоматизованої системи опалення.

4.1 DS18B20

DS18B20-це датчик який здатен вимірювати температуру простіше кажучи цифровий термометр, що може вимірювати температуру від -55 С до +125 С. Компанія яка виготовляє цей датчик вирішила що він буде мати свою унікальну 64-бітну адресу. Обмін інформацією між температурними датчиком та мікроконтролером відбувається по шині 1-wire.

Це дає змогу підключати до мікроконтролеру цілу низку датчиків.

Датчик DS18B20 може бути в 3-х різних корпусах: TO-92, SO, uSOP.

Нижче показано корпуси та розпіновку датчику DS18B20.

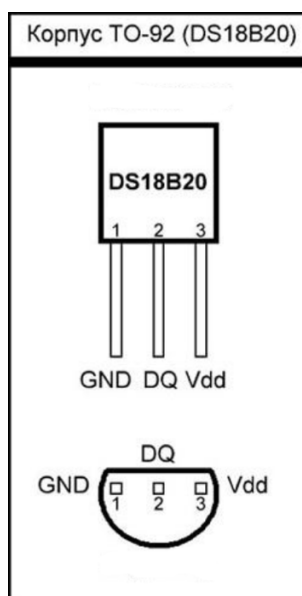


Рисунок 4.1 – Корпус TO-92

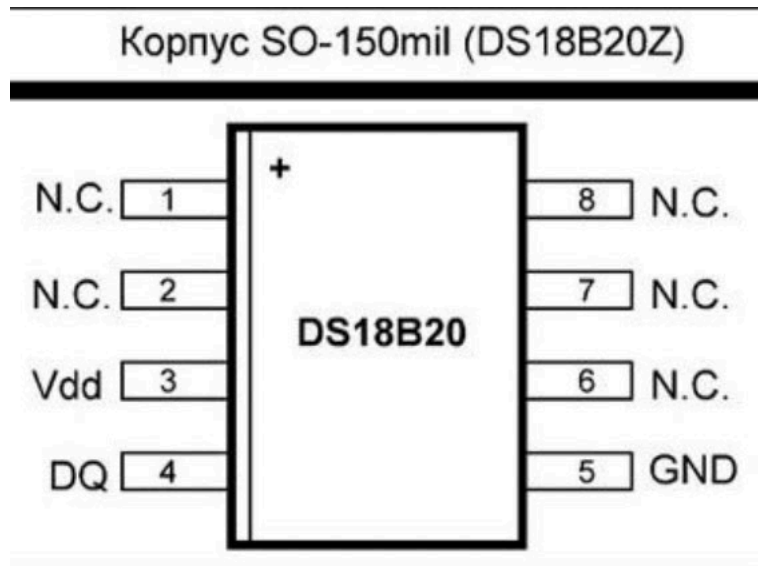


Рисунок 4.2 – Корпус SO

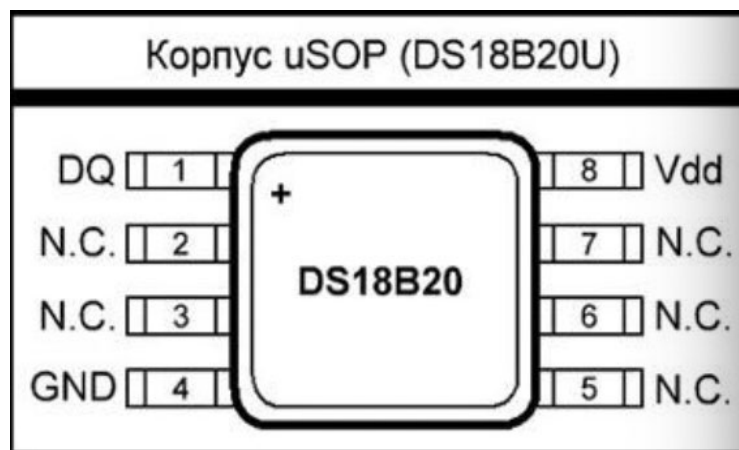


Рисунок 4.3 – Корпус uSOP

Для того, щоб зрозуміти як працювати з датчиком потрібно вивчити його внутрішню будову. Ця маленька на перший погляд мікросхема може мати собі велику кількість електронних модулів та блоків.

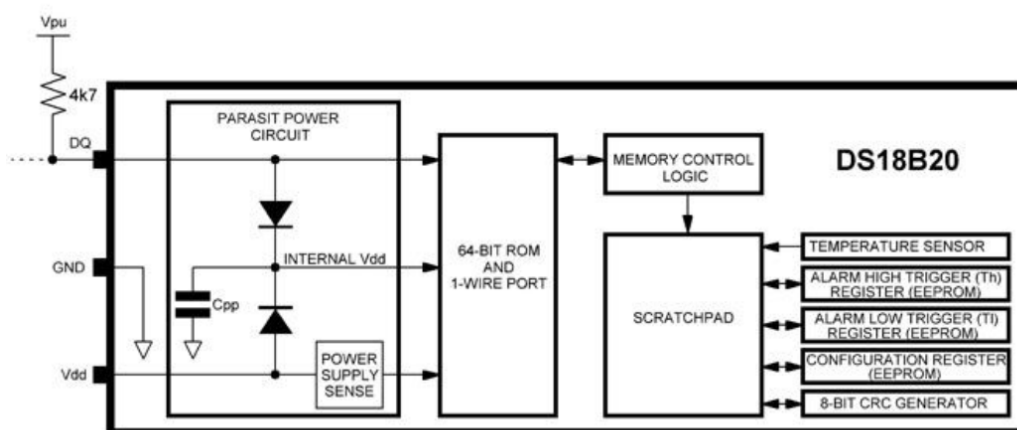


Рисунок 4.4 – Структурна схема

Перед тим як підключати датчик потрібно звернути увагу на схему підключення та спосіб живлення.

Схеми живлення дві:

- Режим прямого живлення (напруга приходить на виводи Vdd і GND)
- Режим «паразитного» живлення(датчик підключається до лінії даних)

Більш детально розглянемо схеми живлення.

Якщо ми використовуємо режим прямого живлення, то за цей процес відповідає модуль (POWER SUPPLY SENSE).

А якщо будемо використовувати режим «паразитного» живлення то відповідно модуль (PARASIT POWER CIRCUIT).

У даного датчика є своя унікальна адреса, яка запам'ятовує інформацію про датчик, це модуль (64-BIT ROM AND 1-WIRE PORT).

В цьому модулі також знаходиться під система для використання 1-wire інтерфейсу.

TEMPERATURE SENSOR. Дозволяє робити перетворення температури.

ALARM HIGH TRIGGER и ALARM LOW TRIGGER. Ця команда дозволяє виставити верхній та нижній аварійні пороги.

CONFIGURATON REGISTER. Ця команда дозволяє настраювати можливості температурного датчика .

8-BIT CRC GENERATOR. Цей реєстр предназначена для захисту даних.

4.2 Технічні характеристики датчику DS18B20

Напруга живлення: 3V-5.5V;

Протокол обміну даними: 1-Wire;

Спосіб підключення: прямий / по одній лінії з паразитних харчуванням;

Дозвіл перетворення температури: 9 біт – 12 біт;

Діапазон вимірювання температури: від -55 до +125 оС;

Період вимірювання температури при максимальній точності 12 біт: 750 мС;

Тип індексації на лінії 1-Wire: унікальний 64-бітову адресу;

Температурний датчик DS18B20 ,як згадувалося вище може бути підключений різними способами та різної кількості датчиків.

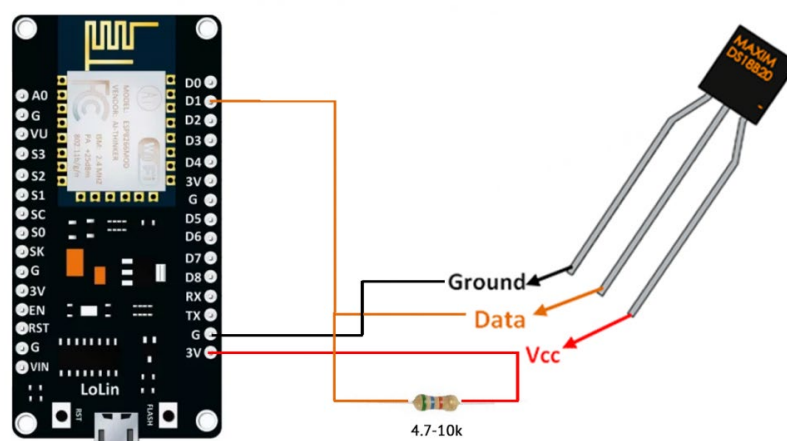


Рисунок 4.5 – Підключення одного датчика.

У даному прикладі все дуже просто ми підключаємо провід живлення подаючи 5 В на вивід Vdd. Точно так же підключаємо землю GND. Вивід для подавання команди підключаємо у вільний вивід на платі.

Єдине що потрібно завжди пам'ятати це обов'язкова наявність резистора номіналом 4,7 к який включаємо між живлення та лінією даних.

Для чого потрібен резистор? Він слугує для підтягування лінії даних, якщо ж даного резистора не буде, то даний датчик не буде ініціалізовано.

Тепер розглянемо підключення групи датчиків.

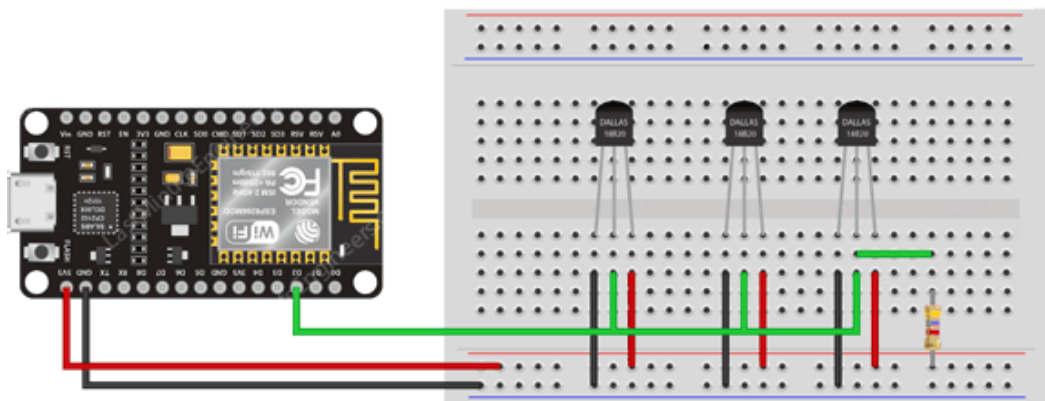


Рисунок 4.6 – Підключення групи датчиків

Зміни по схемі мінімальні , але тепер зміни будуть у програмному кодї, адже датчики підключені до одного виводу на платі. Для чого ж цьому датчику свій унікальний адрес? Саме у цьому підключенню дані адреси потрібні, щоб звертатися до кожного датчику окремо.

4.3 Бібліотеки

Що ж до програмування датчика для його роботи потрібні бібліотеки в середовищі програмування Arduino IDE. Для роботи з датчиком потрібні дві основні бібліотеки `OneWire.h` і `DallasTemperature.h`. Дані бібліотеки можна знайти на просторах інтернету або ж встановити прямо у середовищі Arduino IDE.

Бібліотека `OneWire.h` потрібна для того щоб отримувати інформацію з датчиків та для ініціалізації датчику.

Розглянемо основні команди для роботи датчику:

`OneWire temperatureSensor (uint8_t pinNumber)`-дана команда дозволяє створювати об'єкт `temperatureSensor`

та відкриває канал для зв'язку з датчиком або групою датчиків та вказується номер піна до якого підключено датчик

`uint8_t search(addrArray)`-дана команда дозволяє знайти пристрій який підключений по шині 1-Wire та передати значення у масив `addrArray`. У зв'язку із тим що унікальний адрес датчику складає 64 біти то масив `addrArray` повинен мати розрядність 8 біт.

Якщо до плати підключено декілька датчиків то мікроконтролер буде перебирати усі датчики які підключені у шину 1-Wire. Головна особливість даної функції це здатність запам'ятовувати ті адреси, які уже відпрацювали.

`Void reset_search()`-дана команда припиняє опит пристроїв які підключені по шині 1-Wire.

`Uint8_t reset()`-це функція яка дозволяє скинути процеси на шині 1-Wire.

`Void select(addrArray)`-дана команда дає можливість вибрати конкретний пристрій з яким хоче працювати користувач, вибір відбувається за рахунок запису 64 бітної адреси.

`Void skip()` дана функція дозволяє пропустити вибір датчику але за умови що нас один датчик.

Uint8_t read()-дана команда відправляє один байт даних по шині 1-Wire датчику.

Static uint8_t crc8(const uint8_t * addr, uint8_t len)-дана команда використовується для обчислення контрольної суми та правильного обміну даних з датчиком температури.

Бібліотека DallasTemperature.h побудована на основі бібліотеки OneWire.h та значно полегшує програмування за рахунок більш зрозумілих функції та команд.

4.4 Серво привід SG90

Компактний серво привід SG90 можна використовувати для вирішення тих чи з чи інших задач.

Він легкий у підключення та програмування у середовищі розробки Arduino IDE.

Серво привід складається з редуктора який працює від постійного живлення мотора і керуючої електроніки. Шестерні виготовленні із нейлону. Серво привід може обертатися на 180 градусів. А низька швидкість дозволяє більш точно позиціонувати поворотний механізм.

Головні плюси серво приводу це низька ціна, висока продуктивність, здоровий крутний момент та низьке енергоспоживання . Серво привід використовувати це від робототехніки до радіокерованих іграшок.

Для підключення серво приводу потрібно підключити три проводи:

- червоний-живлення
- коричневий-земля
- жовтий-сигнал



Рисунок 4.7 – Сервопривід

Технічні характеристики

Число оборотів холостого ходу – 0.12 секунди / 60 градусів (при напрузі живлення 4.8В);

Крутний момент – 1.6 кг × см (при напрузі живлення 4.8В);

Діапазон обертання – 180°;

Матеріал корпусу – пластик;

Діапазон робочих температур – від -30 до + 60°С;

Час зупинки в процесі позиціонування – 7 мкс;

Робоча напруга – 4.8-6В;

Максимальний споживаний струм – 500 мА;

Довжина кабелю – 180 мм;

Внутрішній інтерфейс – аналоговий;

Габарити – 5x29.5 мм;

Вага: 9 грам.

4.5 Резистор 4.7 КОМ

Вивідні резистори 4.7 кОм маркуються 4-ма кольоровими кільцями на корпусі. Кожен колір формує числове значення опору резистора. Перше кільце (жовтого кольору) і друге кільце (фіолетового кольору) позначає опір в Омах, третє кільце (червоного кольору) означає множник цієї величини. Останнє кільце (золотистого або сріблястого кольору) вказує на установку допуску резистора.

Нижче на малюнку зображена кольорове маркування резистора опором 4.7 кОм і допуском $\pm 5\%$.



Рисунок 4.8 – Резистор 4.7 КОМ

Найбільш використовуваний елемент в електроніці це резистор. З його допомогою можна керувати струмом обмежуватись струм, вимірювати, створювати зворотній зв'язок ланцюга. Жодна схема не може обійтися без опору.

Резистор відноситься до групи пасивних елементів тому ,може знижувати ток, на відмінно від активних елементів пасивні не можуть підсилювати сигнал.

Резистор встановлюється у схемі так щоб обмежувати струм що протікає в схемі напруга розраховується за законом Ома.

$$U = IR$$

Резистори потрібно розрізняти тому є визначення за якими розрізняють резистори різної на потужності.


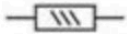
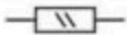
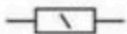
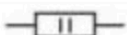
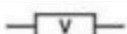
Обозначение по ГОСТ 2.728-74	Описание
	Постоянный резистор без указания номинальной мощности рассеивания
	Постоянный резистор номинальной мощностью рассеивания 0.05 Вт
	Постоянный резистор номинальной мощностью рассеивания 0.125 Вт
	Постоянный резистор номинальной мощностью рассеивания 0.25 Вт
	Постоянный резистор номинальной мощностью рассеивания 0.5 Вт
	Постоянный резистор номинальной мощностью рассеивания 1 Вт
	Постоянный резистор номинальной мощностью рассеивания 2 Вт
	Постоянный резистор номинальной мощностью рассеивания 5 Вт

Рисунок 4.9 – ГОСТ 2.728-74

5. Програмне забезпечення

У даному проєкті ми будемо використовувати програмне забезпечення **Arduino IDE**, яке призначене для програмування мікроконтролерів на мові C, та додаток **Blynk** це додаток який дозволяє керувати власними пристроями через мережу інтернет, за допомогою хмари конструювати та підключати інтерфейс додатку до власного пристрою.

У даному розділі буде розглянуті основні моменти у роботі з ними.

5.1 Arduino IDE

Arduino IDE- Це програмна середа розробки яка використовує мову C++ і Призначена для програмування плат Arduino та плат на базі Arduino.

IDE розшифровується як Integrated Development Environment, в перекладі— інтегрована середа розробки.

За її допомогою програмісти мають можливість писати код програми дуже зручно та набагато швидше ніж у звичайному текстовому редакторі.

Arduino IDE-дає можливість складати код програми в зручному в текстовому редакторі та самостійно компілює її у машинний код, який в наступному можна буде завантажити у плату Arduino.

Інтерфейс даної програми не дуже складний в її основі лежить мова C++.

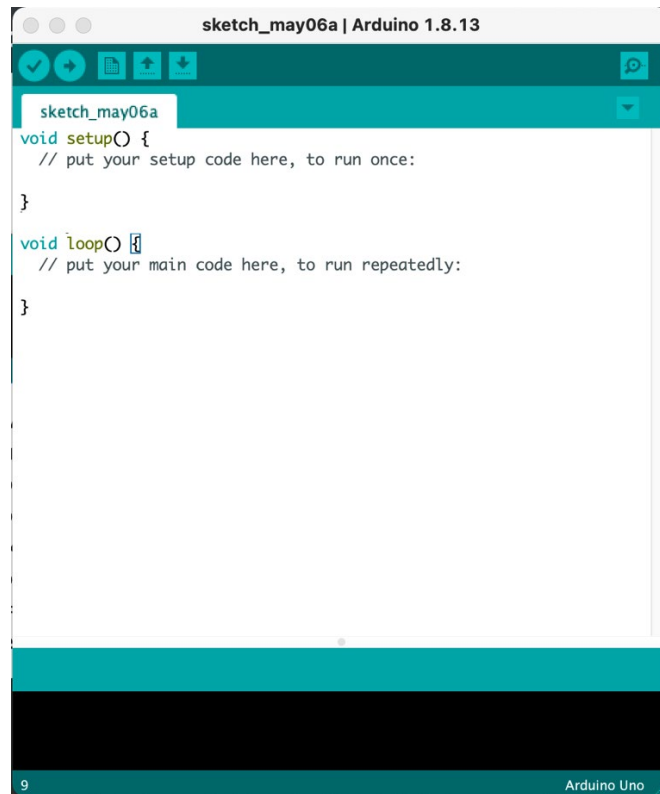


Рисунок 5.1 – Arduino IDE

Void setup()- дана функція потрібно для ініціалізації системи. Для цього потрібно дати команди мікроконтролер, які він виконує тільки один раз при завантаженні.

```
void setup(){  
  
Serial.begin(9600);  
  
pinMode(13, OUTPUT);
```

Рисунок 5.2 – Функція Void setup() приклад

Void loop()- функція loop призначена для того, щоб поміщати туди команди, які будуть виконуватися до тих пір поки включена плата Arduino в мережу. Коли мікроконтролер почне виконувати команду, то дійшовши до кінця, він повернеться на початок програми та почне повторювати до безкінечності.

Void loop()- це головна функція виконання нашої програми.

```
void loop(){  
  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

Рисунок 5.3 – Функція Void loop() приклад

Отже за допомогою цих двох команд ми завантажуюмо мікроконтролеру інструкції нашого скетча.

Arduino IDE має підтримку різних плат Arduino UNO, Arduino Nano, ESP32, ESP8266, сімейство мікроконтролерів Attiny и т.д.

Arduino IDE надає можливість підключити будь-яку підтримку плат через вбудований в менеджер плат Arduino IDE.

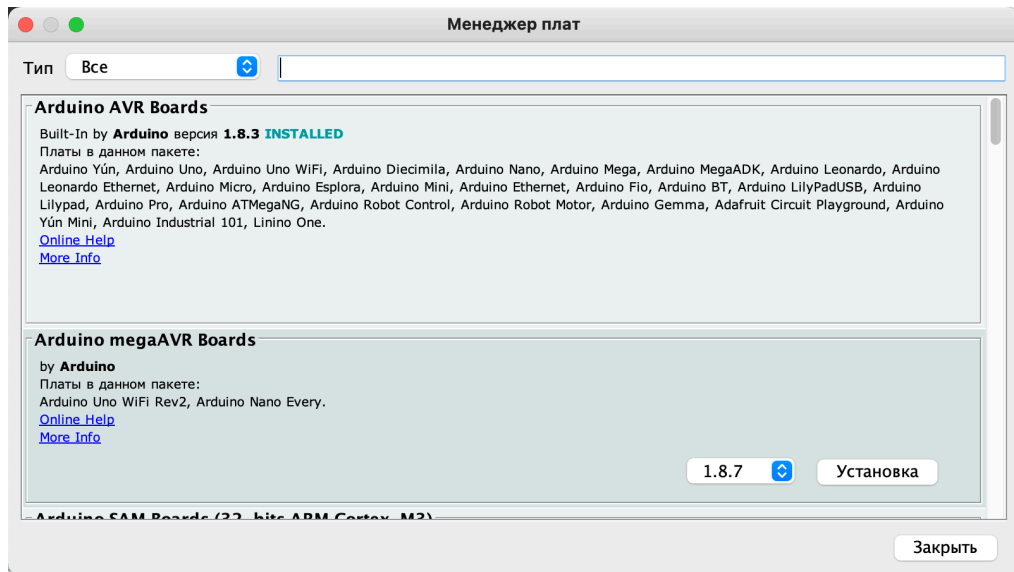


Рисунок 5.4 – Менеджер плат

На сьогоднішній день Arduino зазнало широкого використання по всьому світу

З підтримкою плат все зрозуміло, для того щоб підключити необхідну нам плату потрібно встановити підтримку у менеджері плат.

Що ж до устаткування та приладів які підключені до плати. То тут є теж деякі моменти.

Є вбудовані бібліотеки які дозволяють працювати з тими чи іншими приладами. Але не всі прилади працюють зі стандартними бібліотеками, тому є такий інструмент як менеджер бібліотек.

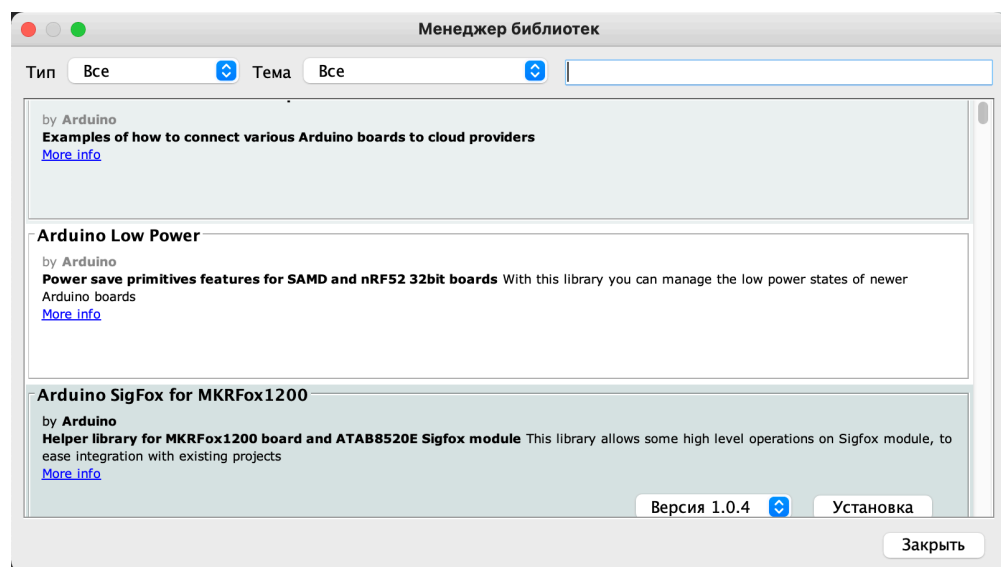


Рисунок 5.5 – Менеджер бібліотек

Але тут теж є деякі моменти. А саме що з 2020р. Arduino IDE припинило підтримку плат ESP8266.

Тому для коректної роботи мікроконтролера та датчиків підключених до нього потрібно встановити бібліотеки зі сторонніх ресурсів у мережі інтернет.

Для того щоб встановити бібліотеку **вручну** потрібно завантажений архів з бібліотекою копіювати в папку де знаходяться всі бібліотеки Arduino.

Після всіх маніпуляції ваша директорія з бібліотеками має мати наступний вигляд.

```
My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.cpp  
My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.h  
My Documents\Arduino\libraries\ArduinoParty\examples
```

Рисунок 5.6 – Організація директорії з бібліотеками

Для **автоматичного** встановлення все набагато простіше, бо саме встановлення не виходить за рамки Arduino IDE.

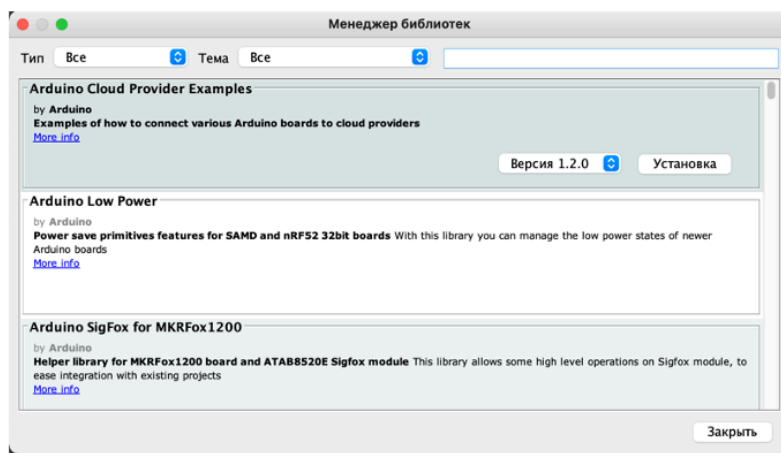
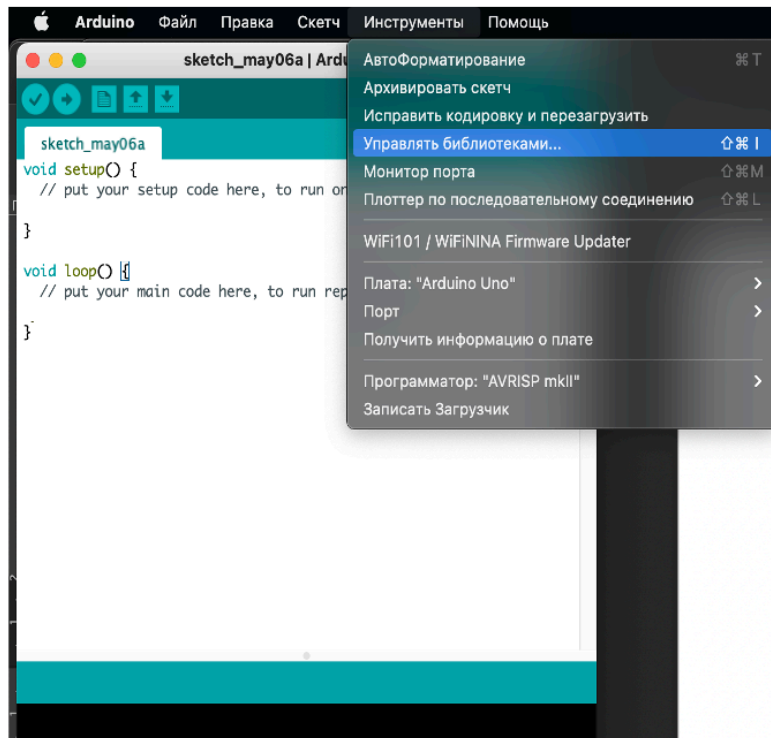


Рисунок 5.7 – Автоматичне встановлення бібліотек.

5.2 Blynk

Blynk-це незалежна апаратна частина з платформою IoT, яка надає можливість підключення ваших пристроїв до хмарного сервісу.

З метою зберігання та контролювання ваших пристроїв дистанційно через мережу інтернет.

Blynk-має зручний та зрозумілий інтерфейс. Підключення може здійснюватися через всі сучасні методи зв'язку WiFi, LTE, 2G-4G или Ethernet.

Blynk Server- це насамперед дуже зручно та просто так як сервер можна відкрити за декілька хвилин і який дає можливість керувати вашими пристроями с мобільних телефонів.

Для використання потрібно пройти легку реєстрація для отримання токену, який буде використовуватися у наступному.

Проведемо ознайомлення з інтерфейсом.

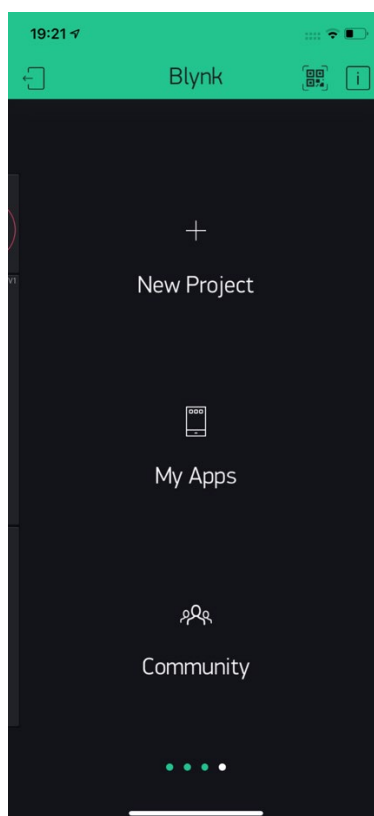


Рисунок 5.8 – Головне меню Blynk

У головному меню ми можемо :

- створити проект
- завантажити вже створений проект
- та зайти до спільноти людей

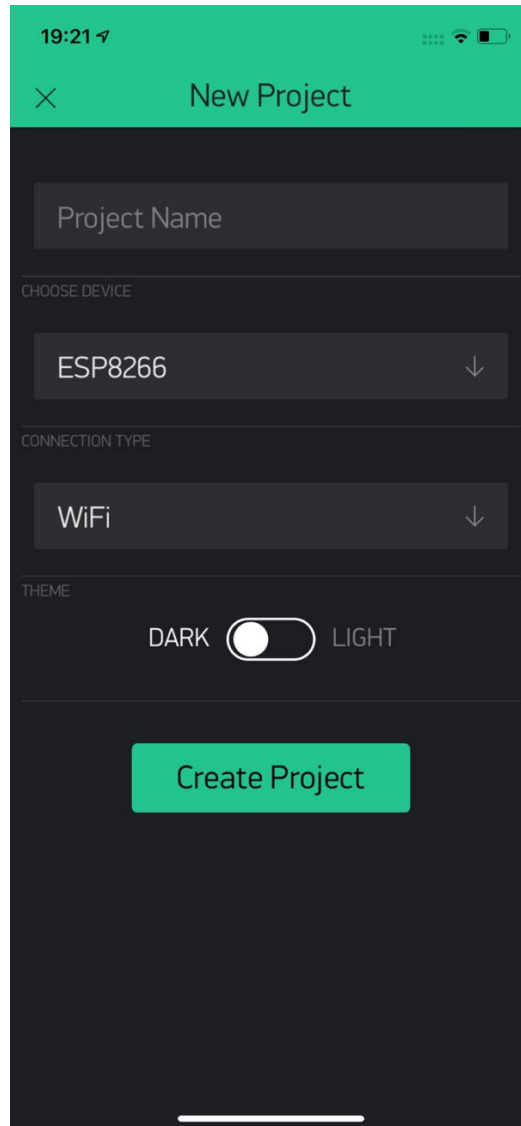


Рисунок 5.9 – Налаштування для проекту

У налаштуваннях проекту ми можемо:

- дати назву проекту
- вибрати мікроконтролер на базі якого будемо будувати додаток
- вибрати канал зв'язку через який будемо керувати пристроєм

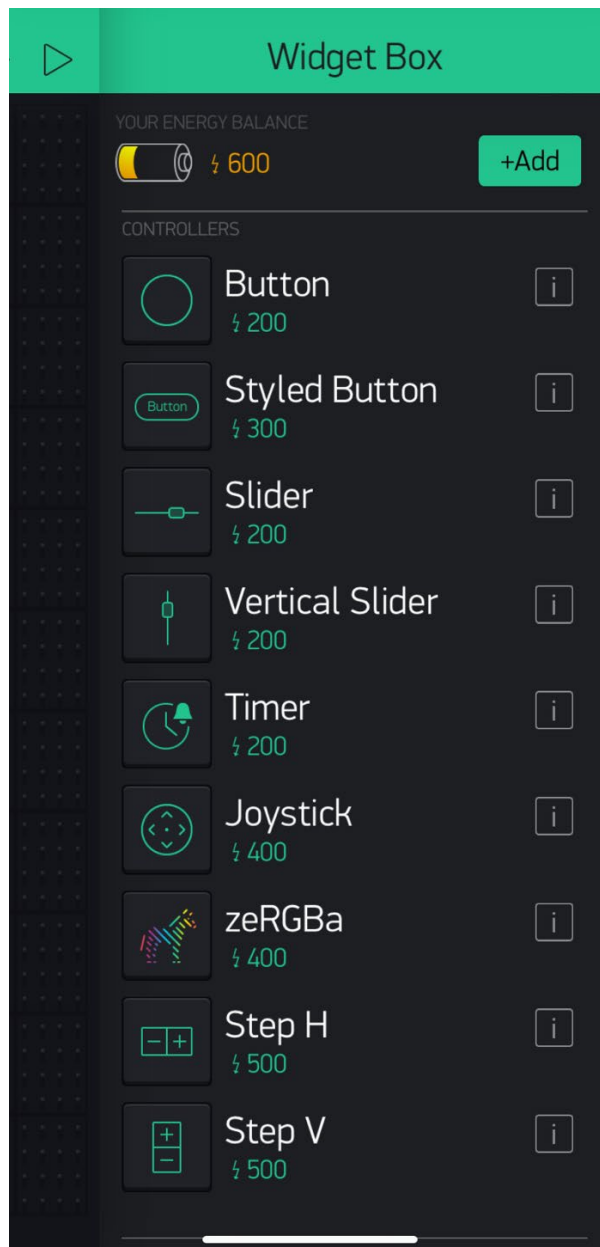


Рисунок – 5.10 Панель віджетів

Кожен віджет має пояснення та інформацію щодо підключення та має фрагменти коду для прикладу.

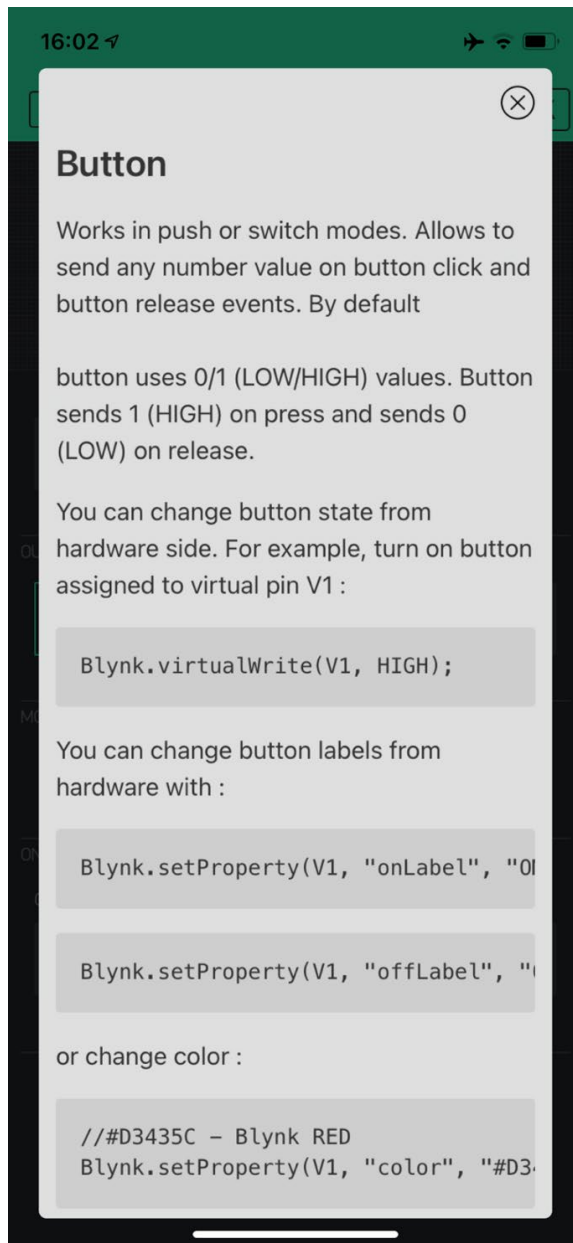


Рисунок 5.11 – Довідка для підключення віджету

6. Програмний код

Розглянемо спочатку бібліотеки які використовуються у проекті автоматизації твердопаливаного котла.

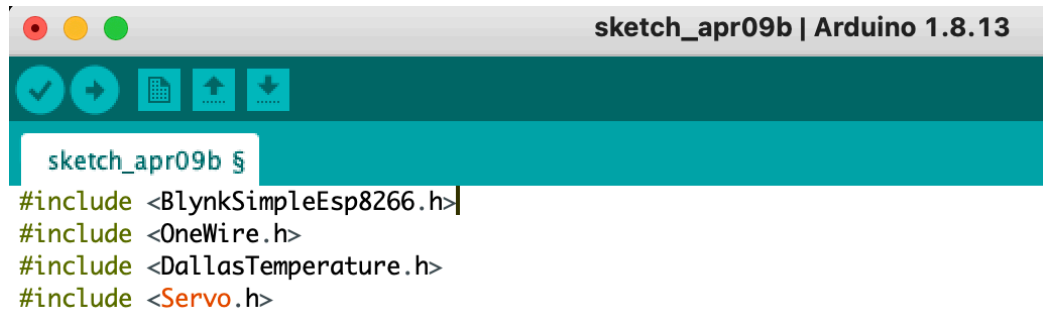


Рисунок 6.1 Використані бібліотеки

`#include<BlynkSimpleEsp8266.h>` дана бібліотека використовується для підтримки плати Esp8266 і підтримки зв'язку через середовище Blynk.

`#include<OneWire.h>`- дана бібліотека потрібна для роботи шини 1-Wire.

`#include<DallasTemperature.h>`-ця бібліотека дає можливість використовувати датчики температури.

`#include<Servo.h>`-бібліотека для роботи сервоприводу.

```
OneWire oneWireBus(D2);
DallasTemperature sensors(&oneWireBus);
int eco60=0;
int opt70=0;
int teplo80=0;
```

Рисунок 6.2 – Перед програмні налаштування

`OneWire oneWireBus(D2)`-ця команда означає, що датчик температури підключений до цифрового входу D2.

`DallasTemperature sensors(&oneWireBus)`- у цьому рядку ми вказуємо що підключаємо датчик по шині 1-Wire.

```
int eco60=0;
```

```
int opt70=0;
```

```
int teplo80=0;
```

Далі ми створюємо змінні для реалізації кнопок та прирівнюємо їх до нуля.

Servo servo- даємо назву сервоприводу.

```
char auth[] = "1rl-X9dzqA3T2vfH1AWzrofB6kQprnoA";  
char ssid[] = "dimamacan";  
char pass[] = "84433920";
```

Рисунок 6.3 – Організування бездротового зв'язку.

Char auth[] – «1rl-X9dzqA3T2vfH1AwzrofB6kQprnoA»- це унікальна адреса, токен для зв'язку з додатком Blynk.

Char ssid[] = «dimamacan» – ім'я мережі до якої потрібно підключитися.

Char pass[] = «84433920» – пароль від мережі.

```
BLYNK_WRITE(V1);  
BLYNK_WRITE(V7) {  
    eco60 = param.asInt();  
}  
BLYNK_WRITE(V8) {  
    opt70 = param.asInt();  
}  
BLYNK_WRITE(V9){  
    teplo80 = param.asInt();  
}
```

Рисунок 6.4 – Організація кнопок

BLYNK_WRITE(V1)- підключення віджету Gauge до віртуального піна 1.

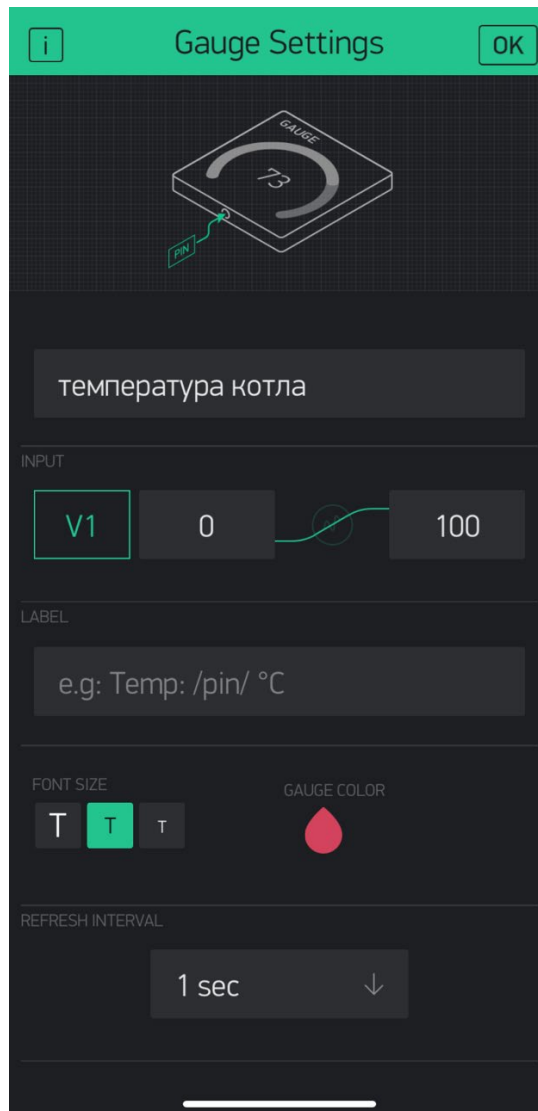


Рисунок 6.5 – Віджет Gauge

```
BLYNK_WRITE(V7) {  
    eco60 = param.asInt();  
}  
BLYNK_WRITE(V8) {  
    opt70 = param.asInt();  
}  
BLYNK_WRITE(V9){  
    teplo80 = param.asInt();  
}
```

Ми повинні прирівняти параметр який ми задавали раніше до методу `param.asInt()`; Для реалізації натискання на кнопку.

```
void setup()
{
    // Debug console

    Serial.begin(9600);
    sensors.begin();

    Blynk.begin(auth, ssid, pass);
    servo.attach(D7);
```

Рисунок 6.6 – Консоль відкладки

`Serial.begin(9600)`- запускаємо порт зі швидкістю 9600мс.

`Sensors.begin()`-даємо команду на зчитування даних з сенсорів та датчиків.

`Blynk.begin(auth, ssid, pass)`- запускаємо канал зв'язку.

`Servo.attach(D7)`- даний фрагмент показує, що сервопривід підключений до цифрового піна (D7).

```
void loop()
{
    Blynk.run();
    Send();
    Work();
    Worked();
    sendTemps();
```

Рисунок – 6.7 Функції

`Blynk.run()`;-запуск сервісів Blynk

`Send()`;-кнопка вибору режиму ЕКО 60

`Work()`;-кнопка вибору режиму ОПТИМАЛЬНИЙ 70

`Worked()`;-кнопка вибору режиму ТЕПЛО 80

`sendTemps()`;- функція зчитування та передачі даних на показчики в додатку Blynk.


```

void sendTemps()
{
    sensors.requestTemperatures();
    float temp = sensors.getTempCByIndex(0);
    Serial.println(temp);
    Blynk.virtualWrite(V1, temp);
}

```

Рисунок 6.8 – Функція зчитування.

Надаємо значення `sensors.getTempCByIndex(0)` для `float temp`, виводимо до порту `Serial.println(temp)`; та на віртуальний пін `Blynk.virtualWrite(V1, temp)`;

```

void Send()
{
    if(eco60 == 1)
    {
        sensors.requestTemperatures();
        Serial.println(sensors.getTempCByIndex(0));
        if (sensors.getTempCByIndex(0) < 60)
        {
            servo.write(0);
            delay(1000);
        }
        else
        {
            servo.write(90);
            delay(1000);
        }
    }
}

```

Рисунок 6.9 – Логіка роботи при натисканні кнопки

```

sensors.requestTemperatures();
Serial.println(sensors.getTempCByIndex(0));

```

Проходить опитування та зчитування датчиків якщо температура менше 60 градусів

`if (sensors.getTempCByIndex(0) < 60)` то сервопривід повертається с початкового положення в положення 0 градусів

```

{
    servo.write(0);
    delay(1000);
}

```

```
}
```

тобто у нашому випадку піднімає повітряний засов. У разі якщо температура не відповідає заданій повітряний засов закривається та повертається на 90 градусів .

```
else
```

```
{
```

```
servo.write(90);
```

```
delay(1000);
```

```
}
```

Далі виконуємо для інших двох кнопок те ж саме , але змінюємо температуру на потрібну нам.

```
if(opt70 == 1)
{
  sensors.requestTemperatures();
  Serial.println(sensors.getTempCByIndex(0));
  if (sensors.getTempCByIndex(0) < 70)
  {
    servo.write(0);
    delay(1000);
  }
  else
  {
    servo.write(90);
    delay(1000);
  }
}

void Worked()
{if(teplo80 == 1)
{
  sensors.requestTemperatures();
  Serial.println(sensors.getTempCByIndex(0));
  if (sensors.getTempCByIndex(0) < 80)
  {
    servo.write(0);
    delay(1000);
  }
  else
  {
    servo.write(90);
    delay(1000);
  }
}
```

Рисунок 6.10 – Код для двох інших режимів.

Дана система була індивідуально розроблена саме для нашого сімейного будинку.

Автономне живлення для нашого пристрою забезпечує PowerBank на 10000 mAh , також наш будинок має автономну систему живлення ,яка автоматично заводить бензиновий генератор.

За необхідності даний пристрій може бути модернізований , доукомплектуваний різними датчиками під різні задачі, та для різних житлових проблем для їх автоматизації.

7. Алгоритм роботи

Після подачі живлення на мікроконтролер та ініціалізації датчиків та мікроконтролера. Програмне забезпечення дає можливість вибрати три режими роботи це Економний 60 градусів, Оптимальний 70 градусів та Тепло 80 градусів. Після вибору режиму дані с датчика температури зчитуються ,якщо температура менше 60 градусів повітряний засов відчиняється сервоприводом. Після чого котел активно набирає температуру після перетину межі яка задана кожним режимом окремо повітряний засов зачиняється таким чином регулюючи температуру котла. Головна перевага такої автоматики низька ціна на відміну від існуючих на ринці.

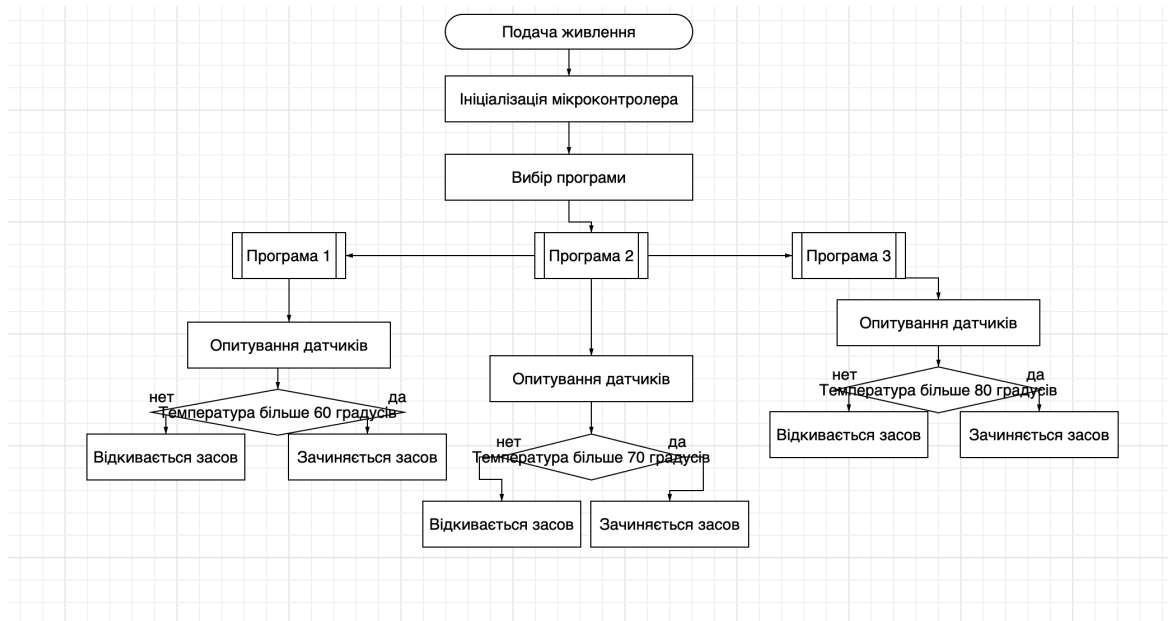


Рисунок 7.11 – Блок схема

Матеріальні витрати для реалізації проекту

Під час побудови даного проекту, головне завдання було зробити даний проект функціональним та автоматизованим та з мінімальними матеріальними витратами. Далі наведена таблиця з назвами комплектуючих та їхніми цінами.

Назва комплектуючого	Ціна грн/штуку
Мікроконтролер ESP8266	110 грн
Сервопривід SG-90	60 грн
Термометр DS18B20	40 грн
З'єднувальний провід	10 грн
Загальна сума	250 грн

Таблиця 1- Матеріальні витрати

ВИСНОВОК

У даній дипломній роботі було показано , як можна автоматизувати роботу твердопаливних котлів з використанням невеликої кількості елементів. А головне зробити це доступним для кожної людини.

Розробка даного пристрою було розроблено за допомогою програмного середовище Arduino та додатку Blynk.

Встановлення даної автоматики значно полегшує використання твердопаливних котлів. Людині не потрібно слідкувати за регуляцією температури тому, що за нас це робить автоматика . Людина може бути впевнена в тому що робота котла автоматизована та може займатися своїми справами. Це полегшує не тільки використання твердопаливних котлів, а й покращую життя людини та економить їй час.

Під час використання автоматики значно економиться і паливо за рахунок підтримки оптимальної температури горіння.

В чому перевага автоматики так це в тому, що вона може бути встановлена на будь-який твердопаливний котел.

Так дана дипломна робота дає наглядний приклад того, що світ іде у майбутнє. Люди намагаються автоматизувати всі процеси які можуть зекономити людям час та здоров'я.

Даний пристрій може бути доповнений різними датчиками які будуть автоматизувати інші процеси.

У висновку ми створили автоматизовану систему дистанційного управління твердопаливного котла на мікроконтролері NodeMCU ESP8266, з використанням термометру та серво приводу. Програмне забезпечення при створенні було використано Arduino IDE , Blynk.

Список використаних джерел

1. Автоматизація твердопаливного котла-[Електронний ресурс]. URL:
<https://xn--e1aamjfh.com.ua/stati-o-komplektuyushih-dlya-otopleniya/avtomatizatsiya-tverdoplivnogo-kotla>
2. Автоматизація особистих процесів-[Електронний ресурс]. URL:
<https://blog.agrokebety.com/avtomatyzatsiya-osobystykh-protsesiv>
3. Що таке мікроконтролери-[Електронний ресурс]. URL:
<http://elektrik.info/main/automation/549-что-такое-mikrokontrollery-naznachenie-ustroystvo-princip-raboty-soft.html>
4. Що таке вбудовані системи та їх використання-[Електронний ресурс]. URL: <http://elektrik.info/main/automation/>
5. Arduino -[Електронний ресурс]. URL: <http://arduino.ru>
6. Довідник мови Arduino-[Електронний ресурс]. URL: <http://arduino.ru/Reference>
7. Blynk-[Електронний ресурс]. URL: <https://blynk.io>
8. ESP8266 -[Електронний ресурс]. URL: <https://esp8266.ru>
9. Використання бібліотек-[Електронний ресурс]. URL:
<https://iarduino.ru/file/>
10. Огляд цифрового датчику температури DS18B20-[Електронний ресурс]. URL:
<https://3d-diy.ru/wiki/arduino-datchiki/tsifrovoy-datchik-temperatury-ds18b20/>
11. Системи автоматизації котлів-[Електронний ресурс]. URL:
<http://engineeringssystem.ru/teplogeneriruyuschiye-ustanovki-sistem-teplosnabzheniya/sistemi-avtomatizacii.php>
12. Робота з ESP8266 в Arduino IDE- <https://voltiq.ru/esp8266-and-arduino/>

13.Можливі варіанти реалізації автоматики-[Електронний ресурс].

URL:

<https://teplovdom.com.ua/news/avtomatizatsija-tverdotoplivnyh-kotlov/>

Додаток А

Вихідний текст програми

`#include <BlynkSimpleEsp8266.h>`//дана бібліотека використовується для підтримки плати Esp8266 і підтримки зв'язку через середовище Blynk.

`#include <OneWire.h>`//дана бібліотека потрібна для роботи шини 1-Wire.

`#include <DallasTemperature.h>` //ця бібліотека дає можливість використовувати датчики температури

`#include <Servo.h>` //бібліотека для роботи сервоприводу.

`OneWire oneWireBus(D2);` //ця команда означає, що датчик температури підключений до цифрового входу D2

`DallasTemperature sensors(&oneWireBus);`//у цьому рядку ми вказуємо що підключаємо датчик по шині 1-Wire.

`int esob0=0;`//Далі ми створюємо змінні для реалізації кнопок та прирівнюємо їх до нуля.

`int opt70=0;`//Далі ми створюємо змінні для реалізації кнопок та прирівнюємо їх до нуля.

`int teplo80=0;`Далі ми створюємо змінні для реалізації кнопок та прирівнюємо їх до нуля.

`Servo servo;`//даємо назву сервоприводу.даємо назву сервоприводу.

`char auth[] = "1rl-X9dzqA3T2vfH1AWzrofB6kQprnoA";`//це унікальна адреса, токен для зв'язку з додатком Blynk.

`char ssid[] = "dimatacan";`//ім'я мережі до якої потрібно підключитися.

`char pass[] = "84433920";`//пароль від мережі.

`BLYNK_WRITE(V1);`//підключення віджету Gauge до віртуального піна 1.

`BLYNK_WRITE(V7) {`

```
    esob0 = param.asInt();//Ми повинні прирівняти параметр який ми задавали  
раніше до методу param.asInt(); Для реалізації натискання на кнопку.
```

```
}
```

```
BLYNK_WRITE(V8) {
```

```
    opt70 = param.asInt();//Ми повинні прирівняти параметр який ми задавали  
раніше до методу param.asInt(); Для реалізації натискання на кнопку.
```

```
}
```

```
BLYNK_WRITE(V9){
```

```
    teplo80 = param.asInt();//Ми повинні прирівняти параметр який ми  
задавали раніше до методу param.asInt(); Для реалізації натискання на кнопку.
```

```
}
```

```
void setup()
```

```
{
```

```
    // Debug console
```

```
    Serial.begin(9600);//запускаємо порт зі швидкістю 9600мс.
```

```
    sensors.begin();//даємо команду на зчитування даних з сенсорів та датчиків.
```

```
    Blynk.begin(auth, ssid, pass);//запускаємо канал зв'язку.
```

```
    servo.attach(D7);//даний фрагмент показує, що сервопривід підключений до  
цифрового піна (D7).
```

```
}
```

```
void loop()
```

```
{ Blynk.run();
```

```
    Send();//кнопка вибору режиму ЕКО 60
```

```
    Work();//кнопка вибору режиму ОПТИМАЛЬНИЙ 70
```

Worked();//кнопка вибору режиму ТЕПЛО 80
sendTemps();//функція зчитування та передачі даних на показчики в
додатку Blynk.

```
}  
void sendTemps()  
{  
  sensors.requestTemperatures();  
  float temp = sensors.getTempCByIndex(0);  
  Serial.println(temp);  
  Blynk.virtualWrite(V1, temp);  
}  
void Send()  
{  
  if(eco60 == 1)  
  {  
    sensors.requestTemperatures();  
    Serial.println(sensors.getTempCByIndex(0));  
    if (sensors.getTempCByIndex(0) < 60)  
    {  
      servo.write(0);  
      delay(1000);  
    }  
  }  
  else  
  {  
    servo.write(90);  
    delay(1000);  
  }  
}
```

```

    }
}
void Work()
{
  if(opt70 == 1)
  {
    sensors.requestTemperatures();
    Serial.println(sensors.getTempCByIndex(0));
    if (sensors.getTempCByIndex(0) < 70)
    {
      servo.write(0);
      delay(1000);
    }
  }
  else
  {
    servo.write(90);
    delay(1000);
  }
}
}

```

```

void Worked()
{if(teplo80 == 1)
  {
    sensors.requestTemperatures();
    Serial.println(sensors.getTempCByIndex(0));
    if (sensors.getTempCByIndex(0) < 80)
    {

```

```
servo.write(0);  
delay(1000);  
}  
else  
{  
servo.write(90);  
delay(1000);  
}  
}
```