

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
КАФЕДРА КОНСТРУЮВАННЯ ЕЛЕКТРОННО-ОБЧИСЛЮВАЛЬНОЇ  
АПАРАТУРИ

## **КУРСОВИЙ ПРОЕКТ**

**З курсу:**

**«Мікропроцесорні технології та компоненти радіоелектронної  
апаратури»**

**тема: « Пристрій, що керує системою опалення будинку на основі  
STM32F407VG »**

Керівник:

доц. Корнєв В.П.

Допущено до захисту

“ \_\_\_\_ ” \_\_\_\_\_ 2023 р.

Захищено з оцінкою

\_\_\_\_\_

Виконала:

Тисяк Є.В.

студент IV курсу ФЕЛ

групи ДК-91

**Київ – 2022**

Національний Технічний Університет України  
“Київський Політехнічний Інститут  
імені Ігоря Сікорського,,

Кафедра Конструювання електронно-обчислювальної апаратури  
Дисципліна Мікропроцесорні технології та компоненти радіоелектронної апаратури  
Спеціальність Інформаційно-обчислювальні засоби електронних систем  
Курс IV Група ДК-91 Семестр VII

## ЗАВДАННЯ

до курсового проекту студента  
Тисяк Євгеній Володимирович  
(прізвище, ім'я та по батькові)

- Тема проекту Пристрій, що керує системою опалення будинку на основі STM32F407VG
- Строк здачі студентом закінченого проекту (роботи) 28.01.2023
- Вихідні дані до проекту (роботи)  
Прилад повинен керувати системою опалення відштовхуючись від часу доби. Регулювання відбуватиметься за допомогою обертання відповідних регуляторів кроковими двигунами. Для створення прототипу (макету) пристрою застосовується плата STM32F407VG GlobalLogic Discovery Kit.
- Зміст розрахунково-пояснювальної записки (перелік питань, що розробляються)
  - Опис структури пристрою і його складових
  - Обґрунтування вибору елементної бази
  - Опис і розрахунок схеми електричної принципової
  - Алгоритм роботи програми
  - Інструкція користувача
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
  - Схема електрична принципова
  - Перелік елементів
- Дата видачі завдання 13.10.2022

## КАЛЕНДАРНИЙ ПЛАН

[illegible]

Студент \_\_\_\_\_

(підпис)

Керівник \_\_\_\_\_

(підпис)

« 21 » січня 2023р.

## ЗМІСТ

Перелік використаних скорочень.....	3
Вступ.....	4
Розділ 1 - Опис структури пристрою та його складових.....	6
1.1 – Загальний опис пристрою.....	6
1.2 – Опис кнопок.....	6
1.3 – Опис двигуна.....	8
1.4 – Опис LCD 1602.....	9
Розділ 2 - Розробка схеми електричної принципової.....	12
2.1 – Загальний опис схеми електричної принципової.....	12
2.2 – Підключення кнопок .....	13
2.3 – Підключення дисплею.....	15
Розділ 3 - Структура програми та її опис.....	16
3.1 – Загальний опис програми .....	16
3.2 – Опис функції для роботи з LCD1602.....	19
3.3 – Опис функцій зовнішніх переривань.....	20
3.4 – Опис функцій переривань системного таймера.....	20

					ДК91.421415.001 ПЗ		
Змн.	Арк.	№ докum.	Підпис	Дата	<div>Пристрій, що керує системою опалення будинку основі STM32F407VG</div> <div>Літ.      Арк.      Аркцшів</div> <div>1      47</div> <div>НТУУ «КПІ»</div>		
Розробив	Тисяк Є.В.						
Перевірів	Корнєв ВП.						
Реценз.							
Н. Контр.	Корнєв ВП.						
Затвердив	Корнєв ВП.						

3.5 – Опис решти функцій.....	20
Розділ 4 - Керівництво з експлуатації пристрою.....	22
4.1 – Основи користування.....	22
4.2 – Налаштування приладу.....	22
Висновки .....	24
Список використаних джерел.....	25
Додаток А. Технічне завдання.....	26
Додаток Б. Лістинг програми.....	30

## ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ

GPIO – General purpose input-output

LCD – Liquid crystal display

ЕЗ – електричне з'єднання

ПЕЗ – перелік електричних з'єднань

ШИМ – Широтно-імпульсна модуляція

МКС – мікросекунди

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		3

## ВСТУП

Сьогодні, практично в кожному будинку є система опалення з регульованою інтенсивністю. Зазвичай, користувачі власноруч регулюють таку систему, що є не зручно, оскільки доводиться завжди пам'ятати про необхідність це робити.

Зазвичай, регулювання відбувається за часом доби. Наприклад, зранку температуру піднімають, а для кращого сну – опускають.

Враховуючи, що зазвичай регулювання відбувається обертанням відповідного обертового регулятора, то є необхідність створити пристрій, що керуватиме кутом оберту у відповідності до часу доби. Обертання буде здійснюватися за допомогою крокового двигуна.

Програмне забезпечення такого пристрою, слід корегувати протягом року, що є недоліком даного пристрою, проте, це дає великий простір в керуванні для користувача. Наприклад, можна на невеликий проміжок часу, викрутити потужність на максимум для того, щоб будинок швидко нагрівся, і через невеликий проміжок перевести регулятор в таке положення, в якому температура буде зберігатись на сталому рівні.

Зазначу, що для гучності прилад міститиме дисплей LCD 1602. На ньому, в верхньому рядку буде відображатись час доби, а в нижньому умовні одиниці повороту. Час буде відображатись в стандартному для цифрових годинників форматі, наприклад п'ятнадцята година двадцять дві секунди відображатимуться як: «15:22». Умовні одиниці, що позначаються кут обертання, це тривалість високого фронту в сигналі PWM для крокового двигуна. Залежність пряmolінійна.

Зауважу, що пристрій містить кнопки, що дозволять встановити час доби.

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		4

Пристрій, що розробляється, неодмінно повинен містити й засоби для кріплення до системи опалення. Данна задача, як і регулювання приладу, повинна виконуватись під конкретний випадок, тому я не буду розглядати це в раках даного курсового проекту.

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		5



# РОЗДІЛ 1 - ОПИС СТРУКТУРИ ПРИСТРОЮ ТА ЙОГО СКЛАДОВИХ

## 1.1 - Загальний опис пристрою

Пристрій, що розробляється містить такі засоби введення інформації:

- Одну кнопку для перезавантаження
- Дві кнопки для встановлення часу доби

Пристрій, що розробляється містить такі засоби обробки інформації:

- Мікроконтролер STM32F407VG

Пристрій, що розробляється містить такі засоби виведення інформації:

- Дисплей LCD 1602
- Кроковий двигун

Основним компонентом такої схеми можна назвати мікроконтролер STM32F407VG, адже самі від буде лічити час, сприймати інформацію з кнопок, керувати кроковим двигуном.

Кожна кнопка підключена до одного GPIO мікроконтролера. Дисплей підключений до семи GPIO мікроконтролера. Кроковий двигун підключений до одного GPIO мікроконтролера. Детальні підключення описано в «Розділ 2 - Розробка схеми електричної принципової».

## 1.2 – Опис кнопок

Як вже було сказано вище, даний пристрій містить три кнопки. Розглянемо їх більш детально.

SB1 - кнопка, що збільшує значення часу на одну хвилину. Дана кнопка

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		6

задіює порт GPIO C11. При створенні макету, в якості цієї кнопки, була використана кнопка SWT1 на стенді Global Starter Kit (Рис. 1.1).

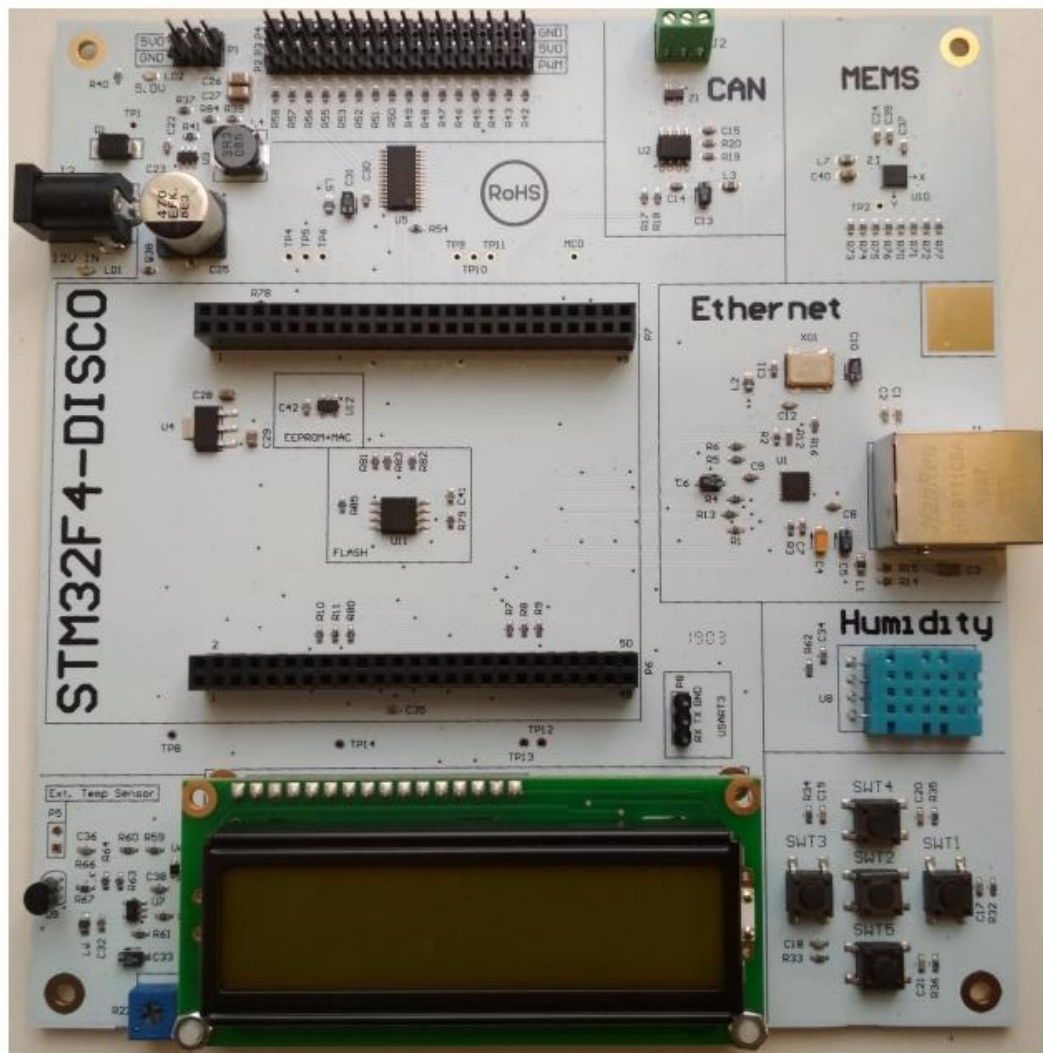


Рис. 1.1 Стенд Global Starter Kit

SB2 - кнопка, що зменшує значення часу на одну хвилину. Дана кнопка задіює порт GPIO C8. При створенні макету, в якості цієї кнопки, була використана кнопка SWT5 на стенді Global Starter Kit (Рис. 1.1).

SB3 – це кнопка перезавантаження мікроконтролера. Натиснувши цю кнопку, можна повернути пристрій в стан за замовчуванням. При створенні макету, в якості цієї кнопки, була використана кнопка B2 RESET на платі STM32F4DISCOVERY (Рис. 1.2).

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		7

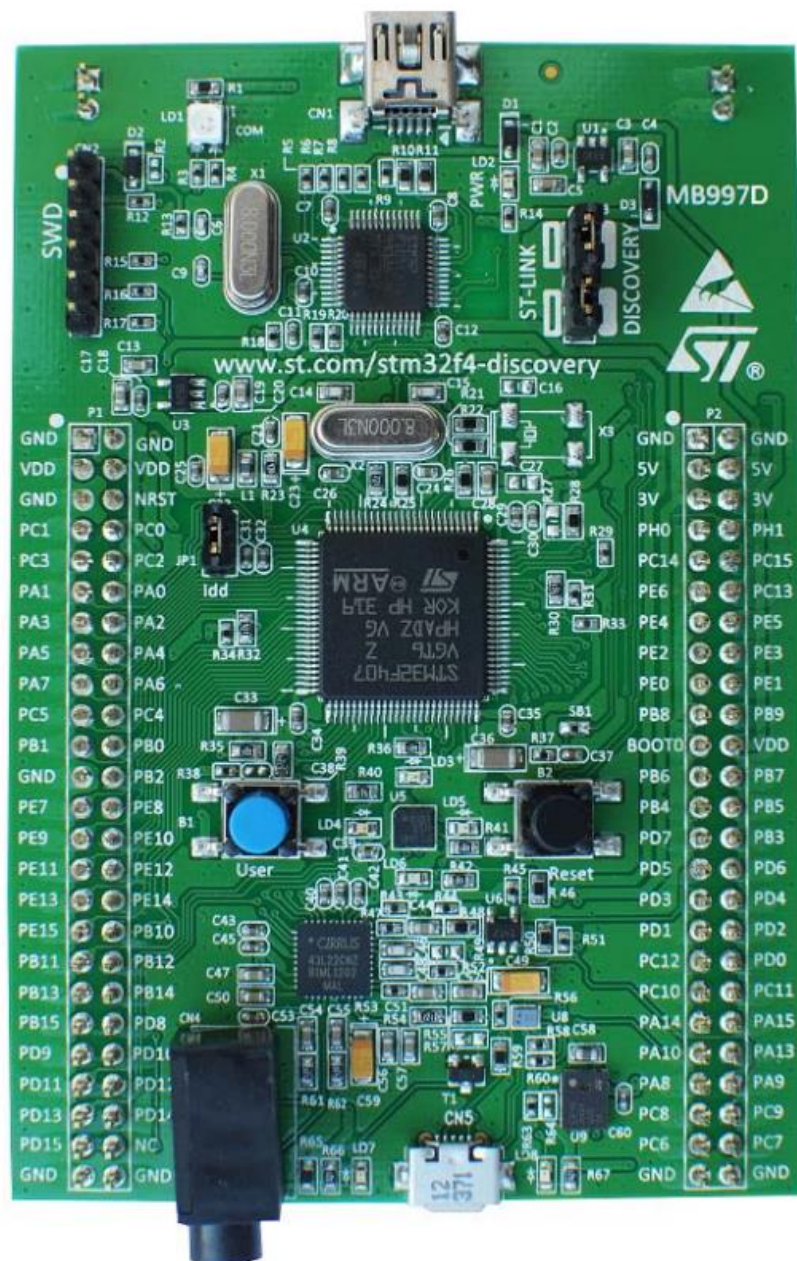


Рис. 1.2 плата STM32F4DISCOVERY

### 1.3 – Опис двигуна

В якості кокового двигуна, при створенні макету, було використано мікросервопривід Tower Pro SG90 micro servo 9g. Зовнішній вигляд даного компоненту наведено на Рис. 1.3:

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		8



Рис. 1.3 Мікросервопривід Tower Pro SG90 micro servo 9g

Даний привід керується ШІМ сигналом на вході PWM, це оранжевий провідник на Рис. 1.3. Цей вхід двигуна підключений до GPIO A4 мікроконтролера STM32F407VG.

#### 1.4 – Опис LCD 1602

LCD 1602 – це рідкокристалічний дисплей, що містить 2 рядки знакомісць, в кожному з яких є по 16 знакомісць. Кожне знакомісце має в собі 8на5 пікселів.

При створенні макету, було використано WH1602B-NYG-CT виробництва WINSTAR DISPLAY CORPORATION. Зовнішній вигляд дисплею на стенді Global Starter Kit наведено на Рис. 1.1.

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		9

Розглянемо детальніше використання дисплею:

Для запису команди чи даних слід встановити відповідні рівні на входах керування дисплеєм ( це входи RS, R/W), на входах даних (це входи DB0-DB7) та подати короткотривалий ( біля 40 МКС ) імпульс на вхід E.

Розглянемо вище зазначенні входи:

- E – строб, коли на всіх інших входах встановлене потрібне значення, на цей слід подати короткотривалий імпульс, для того, щоб дисплей зчитав команду чи дані.
- RS – цей вхід сигналізує дисплею, що ми передаємо йому команду (якщо на цьому вході нуль) чи дані (якщо на вході одиниця).
- R/W – Вказує дисплею, чи буде відбуватись запис в нього (якщо на вході нуль) чи буде відбуватись читання з нього (якщо на вході одиниця).

Оскільки в рамках даного курсового проекту, буде відбуватись тільки запис в дисплей, то даний вхід, можна закоротити на землю, так як це зроблено в стенді Global Starter Kit (Рис. 1.4).

Також слід не забувати про підключення інших портів, таких як:

- VCC – живлення 5 вольт
- VSS – земля
- V0 – вхід керування контрастністю. На стенді Global Starter Kit потенціометр R22.
- A – анод світлодіодного підстелення
- K – анод світлодіодного підстелення

Для зручності наведу уривок з документу E3 ДК91.421415.001, де зображено WH1602B-NYG-CT ( Рис. 1.4 ).

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		10

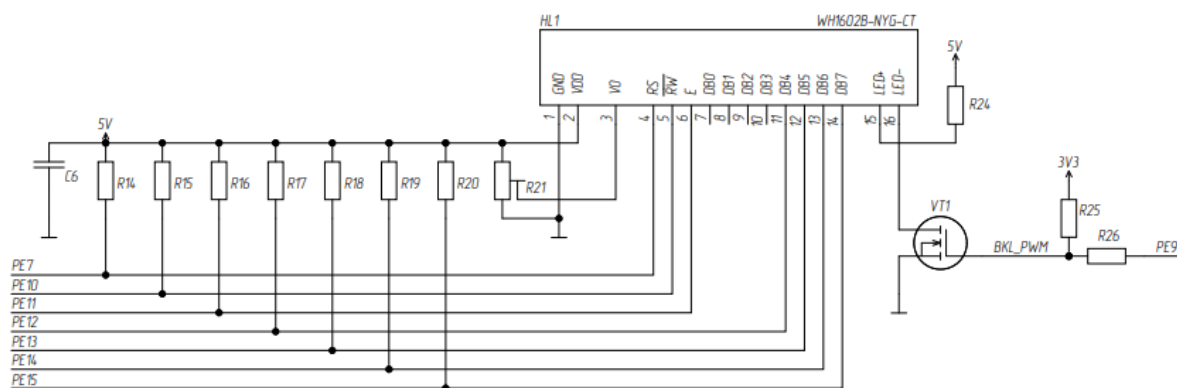


Рис. 1.4 уривок з документу ЕЗ ДК91.421415.001, де зображено WH1602B-NYG-CT

Зауважу, що для вдалої роботи з дисплеєм слід дотримуватись відповідних затримок при передачі інформації. Детальні про це слід читати в документації на пристрій.

Варто також звернути увагу на те, що існує два типи підключення дисплею:

Перший – це восьми бітний режим підключення дисплею. При цьому данні та команди передаються в один такт.

Другий – це чотирьох бітний режим підключення дисплею. При цьому входи DB0-DB3 не використовуються. Для того, щоб передати керуючий байт, спершу подаються старші чотири біти керуючого байту, а потім молодші. Таким чином, передача команд та даних відбувається в два такти. Це економить порти GPIO, але займає більше часу.

При створенні макета було використано саме чотирьох бітний режим.



## РОЗДІЛ 2 - РОЗРОБКА СХЕМИ ЕЛЕКТРИЧНОЇ ПРИНЦИПОВОЇ

### 2.1 – Загальний опис схеми електричної принципової

Наведу Скріншот з документу ЕЗ ДК91.421415.001 на Рис. 2.1:

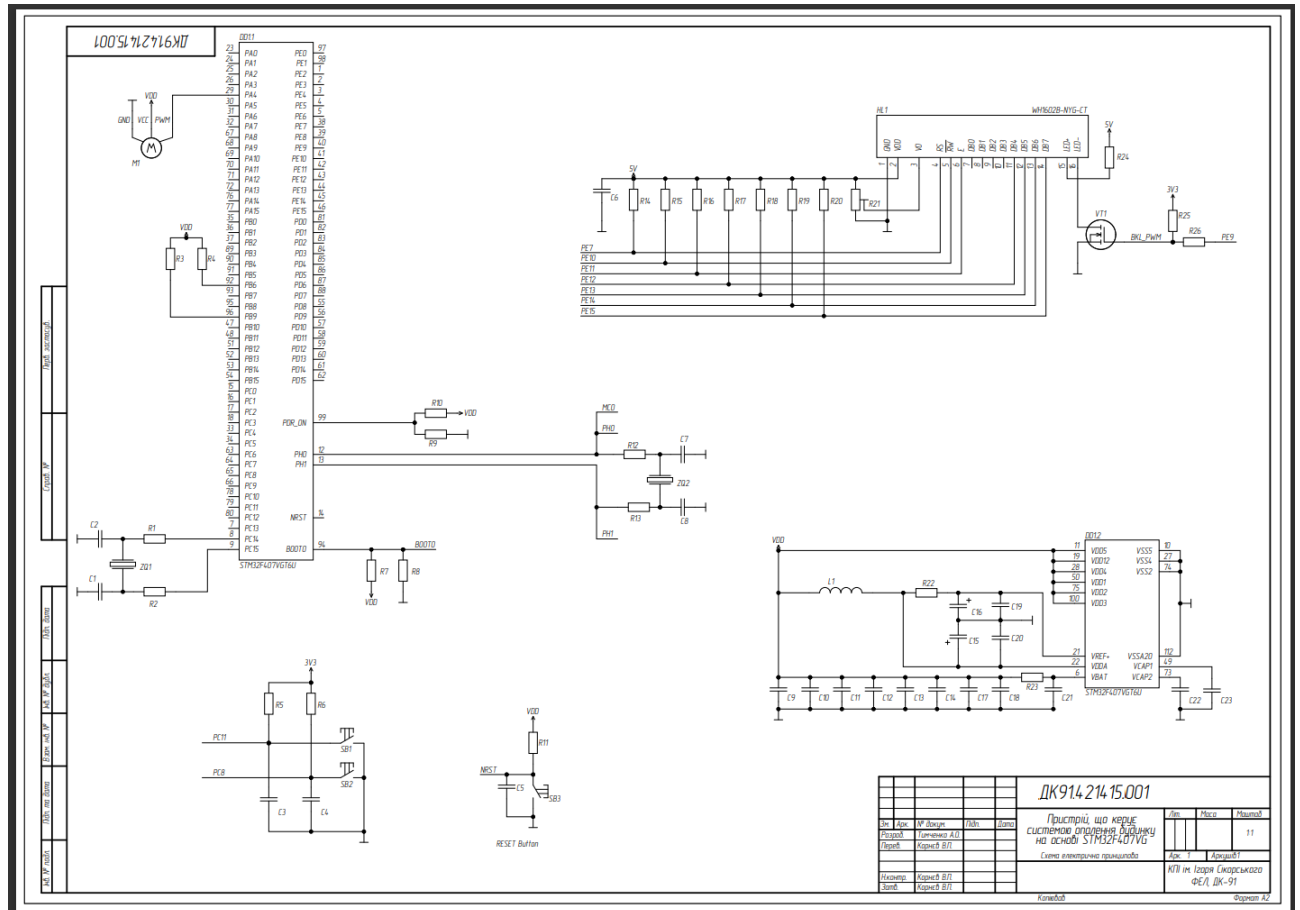


Рис. 2.1 Скріншот з документу ЕЗ ДК91.421415.001

Як можна побачити на Рис. 2.1 головним компонентом є мікроконтролер STM32F407VG, саме до нього підключені всі решту компоненти.

Дане з'єднання аналогічно тому, що використовується на платі STM32F4DISCOVERY (це підключення резонаторів, схеми стабілізації напруги живлення мікроконтролера та кнопки SB3) та стенді Global Starter Kit (це підключення WH1602B-NYG-CT та кнопок SB1- SB2).

Важливі моменти схеми електричної принципової будуть розглянуті в

наступних підрозділах.

## 2.2 – Підключення кнопок

Розглянемо підключення кнопок SB1 та SB2. Наведу їх під'єднання на Рис. 2.2.

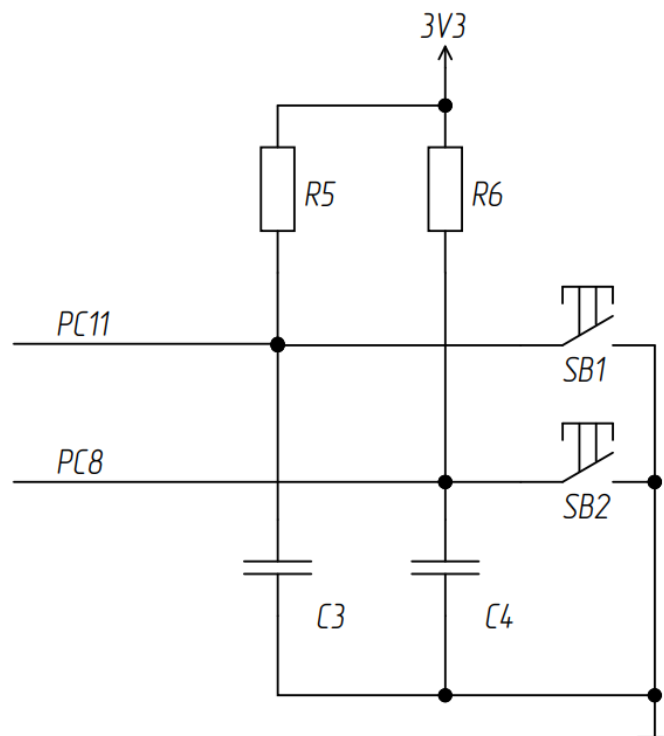


Рис. 2.2 підключення кнопок SB0 та SB1

При підключенні кнопок слід враховувати два моменти:

- Слід забезпечити протидію ефекту брязкоту контактів
- Слід організувати «підтяжку» кнопок на живлення чи землю

Для того, щоб вирішити першу проблему, додані конденсатори C3 та C4. Оскільки напруга на них не може змінюватись різко, то всі брязкоти будуть згладжуватись.

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		13



Для того, щоб вирішити другу проблему, порти GPIO до яких підключаються кнопки, з'єднується з живленням 3,3 Вольти через резистори R5, R6. Таким чином, у випадку не активності кнопок SB1 та SB2, на портах GPIO C11 та GPIO C8 відповідно буде напруга логічного рівня одиниці. При натисканні, відповідні конденсатори будуть розряджатись на землю й на портах буде напруга логічного рівня нуля. Це буде відбуватись, тому що опір між портами й живленнями значно більший ніж між портами й землею.

Зазначу, що програмне забезпечення таке, що програма реагує на саме негативні фронти сигналу на портах GPIO C11 та GPIO C8.

Тепер розглянемо підключення кнопки SB3.

Особливістю цієї кнопки є те, що для вдалого перезавантаження мікроконтролера, слід протримати на вході NRST низький рівень певний час. Це пов'язано з внутрішньою будовою мікроконтролера. Для реалізації такої роботи використовується таке з'єднання (Рис. 2.3):

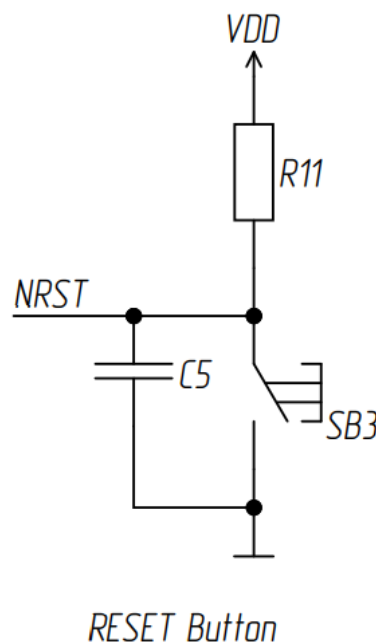


Рис. 2.3 Підключення кнопки SB3

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		14

Алгоритм роботи є наступним:

1. При не активності кнопки SB3 на вході NRST високий рівень, адже конденсатор C5 повністю заряджений.
2. Для перезавантаження мікроконтролера нажимається кнопка SB3. При цьому конденсатор C5 швидко розряджається через кнопку на землю. Ток розрядки великий через низький опір між портом NRST та землею.
3. Коли напруга на NRST стане достатньо мала (як ми вже з'ясували це відбудеться швидко), то внутрішній тригер Шмітта перевернеться і почнеться перезавантаження.
4. При перезавантаженні почнеться зарядження конденсатора C5 вже через резистор R11. Зарядження буде відбуватись повільніше ніж розрядження, через те, що зріс опір. Таким чином, напруга на NRST буде зростати поволі, і тригер Шмітта перевернеться пізніше і мікроконтролер встигне перезавантажитись.

## 2.3 – Підключення дисплею

Схема підключення наведена на Рис. 1.4. Зосереджу увагу на таких моментах:

- Вхід V0 підключено до змінного резистора. Це зроблено для можливості регулювати контрастність дисплею. Зазначу, що таке схема застосовується в макеті, а при створінні окремого пристрою, можна підібрати те значення контрастності, що буде задовольняти, та замінити потенціометр звичайним резистором.
- Входи A та K слід підключати через резистори для обмеження току. Я не рекомендую відходити від схеми представленої в ЕЗ ДК91.421415.001

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		15

## РОЗДІЛ 3 - СТРУКТУРА ПРОГРАМИ ТА ЇЇ ОПИС

### 3.1 – Загальний опис програми

Програмне забезпечення складається з файлів початкової ініціалізації, бібліотеки CMSIS та головного файлу програми main.c (Рис. 3.1)

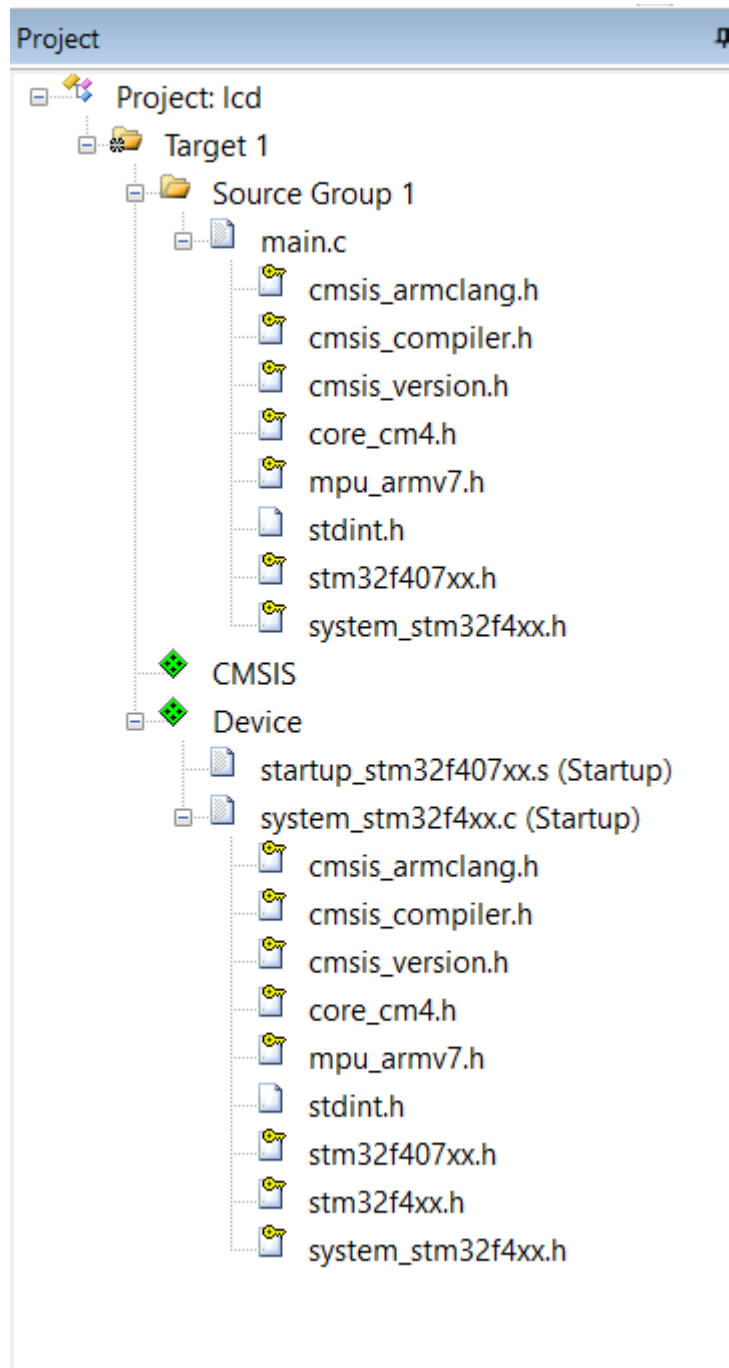


Рис. 3.1 Структура програмного забезпечення.

Зазначу, що підключення всіх потрібних файлі середовище Keil. Ось налаштування в вікні Manage Run - Time Environment (Рис. 3.2):

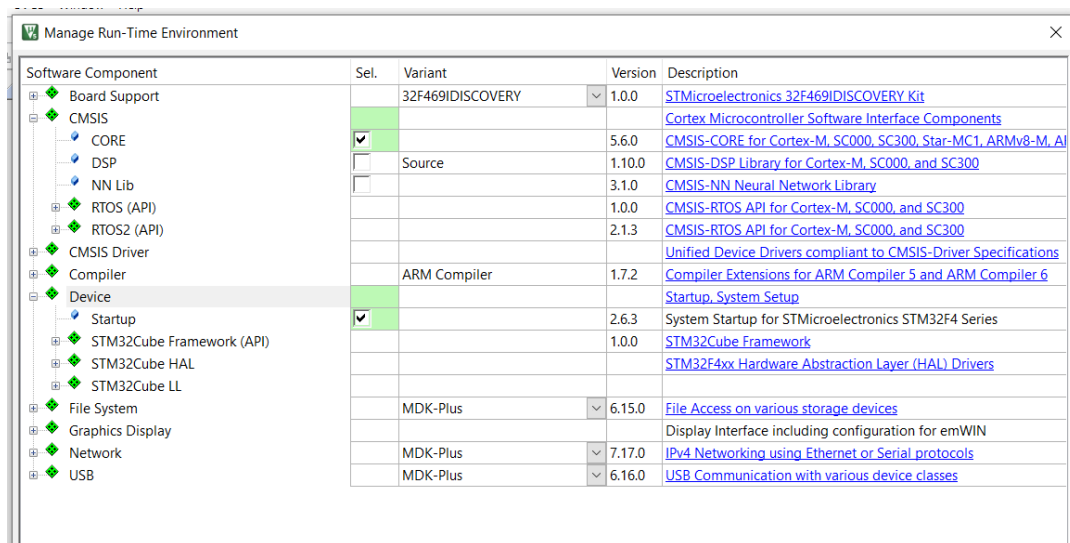


Рис. 3.2 Вікні Manage Run - Time Environment

Зазначу, що всі функції та зміни, проініціалізовані на початку файлу main.c. Ось їх перелік:

```

3  #define ticks_const 0x00F4C985
4  #define Period      5000
5
6  volatile unsigned long delay_counter = 0;
7  volatile unsigned long scaller = 0;
8  volatile unsigned long clock_time = 00*60 + 0;
9  volatile unsigned long angle = 3000;
10 volatile unsigned long angle_buff = 0;
11 volatile unsigned long count_l = 0;
12
13 void lcd_init(void);
14 void lcd_cmd(unsigned long cmd);
15 void lcd_data(unsigned long cmd);
16 void lcd_print_time(void);
17 void lcd_print_angle(void);
18 void lcd_print(void);
19
20 void interrupt_init(void);
21 void EXTI15_10_IRQHandler(void);
22 void EXTI9_5_IRQHandler(void);
23
24 void SysTick_Init (unsigned long ticks);
25 void SysTick_Handler(void);
26
27 void GPIO_init(void);
28 void my_delay(unsigned long delay);
29 void set_angle(void);
30

```

Послідовність роботи виглядає так:

1. Спершу виконується передача управління головній функції main
2. В головній функції main відбуваються всі необхідні функції ініціалізації.

Детальніше про них в інших розділах. Функція main (Рис. 3.3):

```
335 int main()
336 {
337     GPIO_init();
338     interrupt_init();
339     SysTick_Init(ticks_const);
340     lcd_init();
341     lcd_print();
342     while(1)
343     {
344         set_angle();
345     }
346 }
347
```

Рис. 3.3 Функція main

3. При відсутності переривань, циклічно виконується функція set\_angle, детальніше про неї в «3.4 – Опис решти функцій».
4. При перериваннях системного таймера, як стається раз в секунду, відбувається збільшення змінної scaller на одиницю. Кожне 60 переривання збільшує на одиницю вже і змїну clock\_time. При цьому стається виведення нового значення години на екран. Вигляд функція поривання системного таймер (Рис. 3.4):

```
45 void SysTick_Handler(void)
46 {
47     scaller++;
48     if(scaller%60==0)
49     {
50         clock_time++;
51         if(clock_time >= 24*60) clock_time = 0;
52         lcd_init();
53         lcd_print();
54     }
55 }
56
```

Рис. 3.4 Функція переривання системного таймера

Зауважу, що число котре звантажується в лічильник системного таймера підбиралось методом спроб й помилок.

5. При перериванням від кнопок, стається збільшення чи зменшення величини `clock_time` на одиницю, що змінює значення часу на хвилину. Натискання клавіші SB1, викликає переривання `EXTI15_10_IRQHandler`, що збільшує час, а при натисканні SB2, викликає переривання `EXTI9_5_IRQHandler`, що зменшує час. Функції переривання наведені на Рис. 3.5:

```
99 void EXTI9_5_IRQHandler(void) //SWT5
100 {
101     EXTI->IMR &= ~EXTI_IMR_IM8;
102
103     clock_time--;
104     if(clock_time >= 24*60) clock_time = 0;
105     lcd_init();
106     lcd_print();
107
108     EXTI->PR|= EXTI_PR_PR8;
109     EXTI->IMR |= EXTI_IMR_IM8;
110 }

86 void EXTI15_10_IRQHandler(void) //SWT1
87 {
88     EXTI->IMR &= ~EXTI_IMR_IM11;
89
90     clock_time++;
91     if(clock_time >= 24*60) clock_time = 0;
92     lcd_init();
93     lcd_print();
94
95     EXTI->PR|= EXTI_PR_PR11;
96     EXTI->IMR |= EXTI_IMR_IM11;
97 }
```

Рис. 3.5 Функції переривань

### 3.2 – Опис функції для роботи з LCD1602

`void lcd_init(void);` – функція, що ініціалізує дисплей для роботи в чотирьох бітовому режим, очищає дисплей та вимикає курсор.

`void lcd_cmd(unsigned long cmd);` – функція, що надсилає команду в чотирьох бітовому режимі.

`void lcd_data(unsigned long cmd);` – функція, що надсилає данні в чотирьох бітовому режимі.

`void lcd_print_time(void);` – функція, що виводить поточний час

`void lcd_print_angle(void);` – функція, що виводить поточне число, що

відображає поточний кут повороту.

void lcd\_print(void); – функція, що очищає дисплей та викликає функції lcd\_print\_time та lcd\_print\_angle.

### 3.3 – Опис функцій зовнішніх переривань

void interrupt\_init(void); – функція, що ініціалізує переривання по задньому фронту EXTI15\_10\_IRQHandler та EXTI9\_5\_IRQHandler для GPIOC11 та GPIOC8 відповідно.

void EXTI15\_10\_IRQHandler(void); – переривання, що спрацьовує при натисканні SB1

void EXTI9\_5\_IRQHandler(void); – Переривання, що спрацьовує при натисканні SB2

### 3.4 – Опис функцій переривань системного таймера

void SysTick\_Init (unsigned long ticks); – функція, що ініціалізує системні переривання. Число, що записується в лічильник системного таймера це ticks\_const 0x00F4C985.

void SysTick\_Handler(void); – функція, що виконується при спрацюванні системного таймера.

### 3.5 – Опис решти функцій

void GPIO\_init(void); – функція, що ініціалізує порти GPIO для роботи з

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		20

кнопками, дисплеєм та двигуном.

`void my_delay(unsigned long delay);` – функція, що створює затримки в часі.

`void set_angle(void);` – функція, що генерує сигнал для крокового двигуна.

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		21



## РОЗДІЛ 4 - КЕРІВНИЦТВО З ЕКСПЛУАТАЦІЇ ПРИСТРОЮ

### 4.1 – Основи користування

Для того, щоб повторити прилад слід:

1. Зібрати схему у відповідності до документу ЕЗ ДК91.421415.001
2. Завантажити програмне забезпечення, що наведено в додатку Б
3. Натиснути кнопку SB3

Після цих кроків у вас буде пристрій, що почне лік з 0 годин 0 хвилин і в залежності від часу буде змінювати кут оберту крокового двигуна. При цьому формат відображення інформації буде наступний (Рис. 4.1):



Рис. 4.1 Формат відображення інформації

### 4.2 – Налаштування приладу

При використанні приладу в практичних цілях, слід звернути увагу на 2 моменти:

Перший – це спосіб кріплення приладу до системи опалення. В рамках даного курсового проекту, я не буду надавати якісь поради чи рекомендації. Ця частина залишається повністю за користувачем.

Другий – це налаштування програми під себе. Перше про, що слід

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		22

подбати, то це виявити, яке значення змінної `angle` слід встановити, для того, щоб перевести кроковий двигун в потрібне положення. Найлегше це зробити перебираючи варіанти в ручну. Коли у вас буде інформація про те, які значення змінної `angle` відповідають якому положенню, то можна приступити до планування режимів. Іншими словами слід в функцію `set_angle` додати відповідні оператори `if` з проміжками часу, в які ви хочете отримати відповідне регулювання системи опалення. Наведу приклад на Рис. 4.2:

```

240 void set_angle(void)
241 {
242     angle_buff = angle;
243     if( 07*60 + 30 <= clock_time && clock_time <= 12*60 + 00) angle = 1800;
244     else if( 12*60 + 01 <= clock_time && clock_time <= 16*60 + 00) angle = 1200;
245     else if( 16*60 + 01 <= clock_time && clock_time <= 22*60 + 00) angle = 2500;
246     else if( 22*60 + 01 <= clock_time || clock_time <= 07*60 + 29) angle = 600;

```

Ідея наступна:

- З 07:30 до 12:00 – будинок слід нагріти після ночі ( `angle = 1800` ).
- З 12:01 до 16:00 – температуру в будинку можна зменшити, адже зараз день ( `angle = 1200` ).
- З 16:01 до 22:00 – настає вечір, надворі холодніше і температуру слід трішки підняти ( `angle = 2500` ).
- З 22:01 до 07:29 – температуру варто опустити для кращого сну ( `angle = 2500` ).

В результаті отримуємо великий прості для налаштування, проте, це вимагатиме певних знань в користувача.

## ВИСНОВКИ

Отже, в результаті виконання даного курсового проекту, було розроблено цифровий пристрій для контролю системи опалення будинку. Він може бути фізично встановлений на обертові регулятори такої системи й має великі можливості для налаштування. Високі вимоги, до вмінь користувача є недоліком, проте в рамках даного курсового проекту, це не є критично.

Разом з цим документом, було також розроблено ПЗ та ПЕЗ.

Макет створювався на базі наявних компонентів, відео роботи макету також додано до інших матеріалів на Google Disc.

Курсовий проект вважаю успішно виконаним.

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		24

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. STM32f407VG / [Електронний ресурс] - Режим доступу:

<https://www.st.com/en/microcontrollers-microprocessors/stm32f407vg.html>

- Дата звернення 18.01.2023.

2. STM32F4DISCOVERY / [Електронний ресурс] - Режим доступу:

<https://www.st.com/en/evaluation-tools/stm32f4discovery.html>

- Дата звернення 18.01.2023.

3. Матеріали для розробки надані викладачем/ [Електронний ресурс] -  
Режим доступу:

<https://drive.google.com/drive/folders/0B2C9N3752zhKfkRDSkNiZ09sZ2RiWVl3Q0l3TTJBdVhjOVc1RmpDLS0weC1sM0gxZ1h1VTQ?resourcekey=0-MQO0qDnYAWWngzQlEuoA3g&authuser=1>

- Дата звернення 18.01.2023.

4. Курсові проекти однокласників/ [Електронний ресурс] - Режим доступу:

[https://drive.google.com/drive/u/1/folders/1v0gVm-2L170GAK\\_M5WBIVXUsyFJqr9-e](https://drive.google.com/drive/u/1/folders/1v0gVm-2L170GAK_M5WBIVXUsyFJqr9-e)

- Дата звернення 18.01.2023.

5. Курсові проекти студентів паралельної групи/ [Електронний ресурс] -  
Режим доступу:

[https://drive.google.com/drive/u/1/folders/1Lh\\_SLLGPHiIylmX-Achtx9jsa2FHBk-r](https://drive.google.com/drive/u/1/folders/1Lh_SLLGPHiIylmX-Achtx9jsa2FHBk-r)

- Дата звернення 18.01.2023.

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		25

## ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ

### ТЕХНІЧНЕ ЗАВДАННЯ

#### 1. Найменування та галузь використання

- Пристрій цифровий спеціалізований «Пристрій, що керує системою опалення будинку на основі STM32F407VG»

#### 2. Підстава для розробки

- Підставою для проведення курсового проекту є завдання, що видане викладачем згідно учбового плану на 7 семестр.

#### 3. Мета і призначення розробки

- Створення універсального пристрою
- Великі можливості по налаштуванні приладу

#### 4. Джерела розробки

- Пристрій розробляється вперше

#### 5. Технічні вимоги

##### 5.1. Функціональні можливості пристрою.

- Відображення часу в форматі цифрового годинника
- Можливість керувати кут оберту, обертових регуляторів
- Можливість налаштування приладу під користувача

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		26

## **5.2. Технічні характеристики**

- Надійність й універсальність
- Мінімалістичний інтерфейс

## **5.3. Вимоги до рівня уніфікації та стандартизації.**

Для виготовлення пристрою передбачається максимальне застосування стандартних, уніфікованих деталей та виробів.

## **5.4. Вимоги до безпеки експлуатації та обслуговування**

Керуватися загальними вимогами безпеки до апаратури низької напруги ГОСТ 12.2.007-75.

## **5.5. Вимоги до складу та параметрів технічних засобів.**

Для виробництва пристрою використовують матеріали вітчизняного, а також імпортного виробництва.

## **5.6. Вимоги до умов експлуатації.**

Кліматичне виконання УХЛ 4.1 по ГОСТ 15150-69.

## **5.7. Вимоги до транспортування та зберігання.**

Група умов зберігання Л1 по ГОСТ 15150-69. Зберігати в зачинених, опалювальних та вентильованих приміщеннях, в яких забезпечуються наступні умови: температура повітря +5...+40°C, відносна вологість повітря 60% при 200°C (середньорічне значення), атмосферний тиск 84...106кПа. Транспортувати автомобільним, залізничним або авіаційним видами транспорту в спеціальній транспортній тарі.

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		27

## 6. Результати роботи

- 6.1. Результати даної роботи можуть бути використані як вихідна документація по створенню прототипу пристрою та подальшого впровадження його в серійне виробництво.
- 6.2. Дана робота (звітна документація) після виконання надається на кафедру КЕОА для подальшого захисту й зберігання в якості навчальної документації.

## 7. Робота повинна містити:

- Пояснювальну записку з Додатками;
- Схему електричну принципову;
- Перелік елементів;

## 8. Порядок розгляду й приймання роботи

Порядок розгляду й приймання роботи на загальних умовах, прийнятих на кафедрі КЕОА. Рецензування й прийняття роботи комісією на загальних умовах.

## 9. Економічні показники

В рамках даного проекту не розглядаються.

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		28

## 10. Етапи розробки:

№	Назва етапу курсового проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Створення схеми електричної принципової	14.10 – 2.11	
2	Опис структури пристрою і його окремих складових	4.11 – 12.11	
3	Обґрунтування вибору елементної бази	13.11 – 17.11	
4	Опис і розрахунок схеми електричної принципової	19.11 – 21.11	
5	Розробка та затвердження графічної частини проекту	22.11 – 03.12	
6	Алгоритм роботи програми	22.11 – 03.12	
7	Інструкція користувача	04.12 – 19.12	
8	Подача КП до захисту	20.12	



## ДОДАТОК Б. ЛІСТИНГ ПРОГРАМИ

### Вміст файлу main.c:

```
#include <stm32f407xx.h>
```

```
#define ticks_const 0x00F4C985
```

```
#define Period      5000
```

```
volatile unsigned long delay_counter = 0;
```

```
volatile unsigned long scaller = 0;
```

```
volatile unsigned long clock_time = 00*60 + 0;
```

```
volatile unsigned long angle = 3000;
```

```
volatile unsigned long angle_buff = 0;
```

```
volatile unsigned long count_1 = 0;
```

```
void lcd_init(void);
```

```
void lcd_cmd(unsigned long cmd);
```

```
void lcd_data(unsigned long cmd);
```

```
void lcd_print_time(void);
```

```
void lcd_print_angle(void);
```

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		30

```
void lcd_print(void);
```

```
void interrupt_init(void);
```

```
void EXTI15_10_IRQHandler(void);
```

```
void EXTI9_5_IRQHandler(void);
```

```
void SysTick_Init (unsigned long ticks);
```

```
void SysTick_Handler(void);
```

```
void GPIO_init(void);
```

```
void my_delay(unsigned long delay);
```

```
void set_angle(void);
```

```
// #####
```

```
// #####
```

```
void SysTick_Init (unsigned long ticks)
```

```
{
```

```
    SysTick->CTRL = 0;
```

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		31

```

SysTick->LOAD = ticks - 1;

NVIC_SetPriority(SysTick_IRQn, (1 << __NVIC_PRIO_BITS) - 1);

SysTick->VAL = 0;

SysTick->CTRL |= SysTick_CTRL_CLKSOURCE_Msk;

SysTick->CTRL |= SysTick_CTRL_TICKINT_Msk;

SysTick->CTRL |= SysTick_CTRL_ENABLE_Msk;

}

void SysTick_Handler(void)

{

    scaller++;

    if(scaller%60==0)

    {

        clock_time++;

        if(clock_time >= 24*60) clock_time = 0;

        lcd_init();

        lcd_print();

    }

}

```

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		32

```
// #####
```

```
// #####
```

```
void interrupt_init(void)
```

```
{
```

```
    RCC->APB2ENR |= RCC_APB2ENR_SYSCFGEN;
```

```
    SYSCFG->EXTICR[2] |= SYSCFG_EXTICR3_EXTI11_PC;
```

```
    SYSCFG->EXTICR[2] |= SYSCFG_EXTICR3_EXTI8_PC;
```

```
    EXTI->IMR |= EXTI_IMR_IM11;
```

```
    EXTI->IMR |= EXTI_IMR_IM8;
```

```
    EXTI->FTSR |= EXTI_RTSR_TR11;
```

```
    EXTI->FTSR |= EXTI_RTSR_TR8;
```

```
    NVIC_SetPriority(EXTI15_10_IRQn,2);
```

```
    NVIC_SetPriority(EXTI9_5_IRQn,3);
```

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		33

```

    NVIC_ClearPendingIRQ(EXTI15_10_IRQn);

    NVIC_ClearPendingIRQ(EXTI9_5_IRQn);


    NVIC_EnableIRQ(EXTI15_10_IRQn);

    NVIC_EnableIRQ(EXTI9_5_IRQn);

    __enable_irq();

}

void EXTI15_10_IRQHandler(void) //SWT1
{

    EXTI->IMR &= ~EXTI_IMR_IM11;


    clock_time++;

    if(clock_time >= 24*60) clock_time = 0;

    lcd_init();

    lcd_print();


    EXTI->PR|= EXTI_PR_PR11;

```

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		34

```

EXTI->IMR |= EXTI_IMR_IM11;

}

void EXTI9_5_IRQHandler(void) //SWT5

{

    EXTI->IMR &= ~EXTI_IMR_IM8;


    clock_time--;

    if(clock_time >= 24*60) clock_time = 0;

    lcd_init();

    lcd_print();


    EXTI->PR|= EXTI_PR_PR8;

    EXTI->IMR |= EXTI_IMR_IM8;

}

```

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		35

```
// #####
```

```
// #####
```

```
void lcd_print(void)
```

```
{
```

```
    lcd_cmd(0x01); // clear
```

```
    lcd_print_time();
```

```
    lcd_print_angle();
```

```
}
```

```
void lcd_print_time(void)
```

```
{
```

```
    lcd_cmd(0x80);
```

```
    lcd_data('T');
```

```
    lcd_data('i');
```

```
    lcd_data('m');
```

```
    lcd_data('e');
```

```
    lcd_data(0x20);
```

```
    lcd_data(0x2D);
```

```
    lcd_data(0x20);
```

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		36

```

        lcd_data(0x30 + (((clock_time-clock_time%600)/600)%6));

        lcd_data(0x30 + (((clock_time-clock_time%60)/60)%10));

        lcd_data(0x3A);

        lcd_data(0x30 + ((clock_time%60-clock_time%10)/10));

        lcd_data(0x30 + (clock_time%10));

    }

```

```

void lcd_print_angle(void)

```

```

{

    lcd_cmd(0xC0);

    lcd_data('A');

    lcd_data('n');

    lcd_data('g');

    lcd_data('l');

    lcd_data('e');

    lcd_data(0x20);

    lcd_data(0x2D);

    lcd_data(0x20);

    lcd_data(0x30 + (angle - (angle%1000))/1000);

```

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		37



```

        lcd_data(0x30 + ((angle - (angle%100))/100)%10);

        lcd_data(0x30 + ((angle - (angle%10))/10)%10);

        lcd_data(0x30 + (angle%10));

    }

```

```

void lcd_init(void)

{

    lcd_cmd(0x28);// init

    lcd_cmd(0x01);// clear

    //lcd_cmd(0x0D);// cursor on

    lcd_cmd(0x0C);// cursor off

    my_delay(1000);

}

```

```

void lcd_cmd(unsigned long cmd)

{

    GPIOE->ODR &= ~GPIO_ODR_OD7;

    GPIOE->ODR &= ~GPIO_ODR_OD11;

```

```

if(cmd&(1<<4)) GPIOE->ODR |= GPIO_ODR_OD12;

    else GPIOE->ODR &= ~GPIO_ODR_OD12;

if(cmd&(1<<5)) GPIOE->ODR |= GPIO_ODR_OD13;

    else GPIOE->ODR &= ~GPIO_ODR_OD13;

if(cmd&(1<<6)) GPIOE->ODR |= GPIO_ODR_OD14;

    else GPIOE->ODR &= ~GPIO_ODR_OD14;

if(cmd&(1<<7)) GPIOE->ODR |= GPIO_ODR_OD15;

    else GPIOE->ODR &= ~GPIO_ODR_OD15;

```

```
GPIOE->ODR |= GPIO_ODR_OD11;
```

```
my_delay(40);
```

```
GPIOE->ODR &= ~GPIO_ODR_OD11;
```

```
my_delay(4000);
```

```
if(cmd&(1<<0)) GPIOE->ODR |= GPIO_ODR_OD12;
```

```
    else GPIOE->ODR &= ~GPIO_ODR_OD12;
```

```
if(cmd&(1<<1)) GPIOE->ODR |= GPIO_ODR_OD13;
```

```
    else GPIOE->ODR &= ~GPIO_ODR_OD13;
```

```
if(cmd&(1<<2)) GPIOE->ODR |= GPIO_ODR_OD14;
```

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		39

```

        else GPIOE->ODR &= ~GPIO_ODR_OD14;

if(cmd&(1<<3)) GPIOE->ODR |= GPIO_ODR_OD15;

        else GPIOE->ODR &= ~GPIO_ODR_OD15;


GPIOE->ODR |= GPIO_ODR_OD11;

my_delay(40);

GPIOE->ODR &= ~GPIO_ODR_OD11;

my_delay(4000);

}

```

```

void lcd_data(unsigned long cmd)

```

```

{

    GPIOE->ODR |= GPIO_ODR_OD7;

    GPIOE->ODR &= ~GPIO_ODR_OD11;


if(cmd&(1<<4)) GPIOE->ODR |= GPIO_ODR_OD12;

        else GPIOE->ODR &= ~GPIO_ODR_OD12;

if(cmd&(1<<5)) GPIOE->ODR |= GPIO_ODR_OD13;

        else GPIOE->ODR &= ~GPIO_ODR_OD13;

```

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		40

```

if(cmd&(1<<6)) GPIOE->ODR |= GPIO_ODR_OD14;

    else GPIOE->ODR &= ~GPIO_ODR_OD14;

if(cmd&(1<<7)) GPIOE->ODR |= GPIO_ODR_OD15;

    else GPIOE->ODR &= ~GPIO_ODR_OD15;


GPIOE->ODR |= GPIO_ODR_OD11;

my_delay(40);

GPIOE->ODR &= ~GPIO_ODR_OD11;

my_delay(4000);


if(cmd&(1<<0)) GPIOE->ODR |= GPIO_ODR_OD12;

    else GPIOE->ODR &= ~GPIO_ODR_OD12;

if(cmd&(1<<1)) GPIOE->ODR |= GPIO_ODR_OD13;

    else GPIOE->ODR &= ~GPIO_ODR_OD13;

if(cmd&(1<<2)) GPIOE->ODR |= GPIO_ODR_OD14;

    else GPIOE->ODR &= ~GPIO_ODR_OD14;

if(cmd&(1<<3)) GPIOE->ODR |= GPIO_ODR_OD15;

    else GPIOE->ODR &= ~GPIO_ODR_OD15;

```

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		41

```

    GPIOE->ODR |= GPIO_ODR_OD11;

    my_delay(40);

    GPIOE->ODR &= ~GPIO_ODR_OD11;

    my_delay(4000);

}

// #####

// #####

void set_angle(void)

{

    angle_buff = angle;

    /// if( 07*60 + 30 <= clock_time && clock_time <= 12*60 + 00) angle =
    1800;

    /// else if( 12*60 + 01 <= clock_time && clock_time <= 16*60 + 00) angle
    = 1200;

    /// else if( 16*60 + 01 <= clock_time && clock_time <= 22*60 + 00) angle
    = 2500;

```

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		42

```
//// else if( 22*60 + 01 <= clock_time || clock_time <= 07*60 + 29) angle =  
600;
```

```
if( 0 <= clock_time && clock_time <= 2 ) angle = 4200;
```

```
else if( 3 <= clock_time && clock_time <= 5) angle = 1200;
```

```
else if( 6 <= clock_time && clock_time <= 8) angle = 2500;
```

```
else if( 9 <= clock_time && clock_time <= 12) angle = 600;
```

```
if(angle != angle_buff)
```

```
{
```

```
    lcd_init();
```

```
    lcd_print();
```

```
    count_1 = 0;
```

```
    while(count_1 != 100)
```

```
    {
```

```
        GPIOA->ODR |= GPIO_ODR_OD4;
```

```
        my_delay(angle);
```

```
        GPIOA->ODR &= ~GPIO_ODR_OD4;
```

```
        my_delay(Period - angle);
```

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		43

```

        count_1++;

    }

}

}

void GPIO_init(void)

{

    // AHB1ENR

    RCC->AHB1ENR |= RCC_AHB1ENR_GPIOAEN;

    RCC->AHB1ENR |= RCC_AHB1ENR_GPIOCEN;

    RCC->AHB1ENR |= RCC_AHB1ENR_GPIOEEN;

    // MODER A

    GPIOA->MODER |= ~GPIO_MODER_MODE2_0;

    GPIOA->MODER &= ~GPIO_MODER_MODE2_1;

    GPIOA->MODER |= ~GPIO_MODER_MODE4_0;

    GPIOA->MODER &= ~GPIO_MODER_MODE4_1;

```

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		44

// MODER C

GPIOC->MODER &= ~GPIO\_MODER\_MODE8\_0;

GPIOC->MODER &= ~GPIO\_MODER\_MODE8\_1;

GPIOC->MODER &= ~GPIO\_MODER\_MODE11\_0;

GPIOC->MODER &= ~GPIO\_MODER\_MODE11\_1;

// MODER E

GPIOE->MODER |= ~GPIO\_MODER\_MODE5\_0;

GPIOE->MODER &= ~GPIO\_MODER\_MODE5\_1;

GPIOE->MODER |= GPIO\_MODER\_MODE7\_0;

GPIOE->MODER &= ~GPIO\_MODER\_MODE7\_1;

GPIOE->MODER |= GPIO\_MODER\_MODE10\_0;

GPIOE->MODER &= ~GPIO\_MODER\_MODE10\_1;

GPIOE->MODER |= GPIO\_MODER\_MODE11\_0;

GPIOE->MODER &= ~GPIO\_MODER\_MODE11\_1;

GPIOE->MODER |= GPIO\_MODER\_MODE12\_0;

GPIOE->MODER &= ~GPIO\_MODER\_MODE12\_1;

GPIOE->MODER |= GPIO\_MODER\_MODE13\_0;

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		45



GPIOE->MODER &= ~GPIO\_MODER\_MODE13\_1;

GPIOE->MODER |= GPIO\_MODER\_MODE14\_0;

GPIOE->MODER &= ~GPIO\_MODER\_MODE14\_1;

GPIOE->MODER |= GPIO\_MODER\_MODE15\_0;

GPIOE->MODER &= ~GPIO\_MODER\_MODE15\_1;

// OTYPER E

GPIOE->OTYPER &= ~GPIO\_OTYPER\_OT7;

GPIOE->OTYPER &= ~GPIO\_OTYPER\_OT10;

GPIOE->OTYPER &= ~GPIO\_OTYPER\_OT11;

GPIOE->OTYPER &= ~GPIO\_OTYPER\_OT12;

GPIOE->OTYPER &= ~GPIO\_OTYPER\_OT13;

GPIOE->OTYPER &= ~GPIO\_OTYPER\_OT14;

GPIOE->OTYPER &= ~GPIO\_OTYPER\_OT15;

// ODR

GPIOE->ODR &= ~GPIO\_ODR\_OD10;

}

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		46

```

void my_delay(unsigned long delay)

{

    for(delay_counter=0;delay_counter<delay;delay_counter++);

}


// #####

// #####

int main()

{

    GPIO_init();

    interrupt_init();

    SysTick_Init(ticks_const);

    lcd_init();

    lcd_print();

    while(1)

    {

        set_angle();

    }

}

```

					ДК91.421415.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		47