

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

КУРСОВИЙ ПРОЕКТ

З курсу:

«Обчислювальні та мікропроцесорні засоби в РЕА»

тема: «Таймер-секундомір на основі STM32F407VG»

Керівник:

доц. Корнєв В.П.

Допущено до захисту

“ ____ ” _____ 2022 г.

Захищено з оцінкою

Виконав:

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 Обчислювальні та мікропроцесорні засоби в РЕА
Спеціальність Інформаційно-обчислювальні засоби електронних систем
Курс IV Група ДК-91 Семестр VII

ЗАВДАННЯ

до курсового проекту студента
Тисяка Євгенія Володимировича
(прізвище, ім'я та по батькові)

1. Тема проекту Таймер-секундомір на основі STM32F407VG
2. Строк здачі студентом закінченого проекту (роботи) 21.12.2022
3. Вихідні дані до проекту (роботи)
Прилад повинен виконувати функції таймера чи секундоміра в залежності від його налаштувань користувачем. Для налаштування приладу користувач повинен мати відповідні засоби введення. Прилад повинен містити візуальні та звукові засоби виведення. Для створення прототипу (макету) пристрою застосовується плата STM32F407VG GlobalLogic Discovery Kit.
4. Зміст розрахунково-пояснювальної записки (перелік питань, що розробляються)
 1. Опис структури пристрою і його складових
 2. Обґрунтування вибору елементної бази
 3. Опис і розрахунок схеми електричної принципової
 4. Алгоритм роботи програми
 5. Інструкція користувача
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
 1. Схема електрична принципова
 2. Перелік елементів
6. Дата видачі завдання 13.10.2022

КАЛЕНДАРНИЙ ПЛАН

[illegible]

Студент

(підпис)

Керівник

(підпис)

« 05 » жовтня 2022р.

ЗМІСТ

Перелік використаних скорочень

Вступ

Розділ 1 - Опис структури пристрою та його складових

1.1 - Структура пристрою

1.2 - Принципи та засоби введення даних

1.3 - Принципи та засоби відображення даних

Розділ 2 - Розробка схеми електричної принципової

2.1 - Вибір та обґрунтування елементної бази

2.1.1 - Вибір мікросхеми

2.1.2 - Вибір LCD екрану

2.1.3 - Вибір котушок індуктивності

2.1.4 - Вибір перемикачів

2.1.5 - Вибір транзисторів

2.1.6 - Вибір кварцових резонаторів

2.1.7 - Вибір конденсаторів

2.1.8 - Вибір резисторів

2.2 - Опис схеми електричної принципової

Розділ 3 Структура програми та її опис

3.1 – Функції ініціалізації

3.2 – Функції для роботи з дисплеєм

3.3 – Функції зовнішніх переривань

3.4 – Функції системного таймера

3.5 – Загальний алгоритм роботи

					ДК91.403272.001 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата	<i>Пристрій цифровий універсальний для лічення часу</i>		
Розробив		Тисяк Є.В.					
Перевірів		Корнєв ВП.					
Реценз.							
Н. Контр.		Корнєв ВП.					
Затвердив		Корнєв ВП.			НТУУ «КПІ»		
					Літ.	Арк.	Аркциів
						1	47

Розділ 4 Керівництво з експлуатації пристрою

4.1 – Запуск приладу

4.2 – Вибір режиму

4.3 – Режим встановлення початкового значення

Висновки

Список використаних джерел

Додаток А. Технічне завдання

Додаток Б. Лістинг програми

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		2

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ

ЕЗ – електричне з'єднання

ПЕЗ – перелік електричних з'єднань

МК – Мікроконтролер

GPIO – General purpose input-output

LCD – Liquid crystal display

CFGR - Clock configuration register

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		3

ВСТУП

Для людей біологічно властиво думати про навколишній світ в контексті часу, вживати слова «тоді», «потім», «згодом», «колись», «пізно», «вчасно» і так далі. Лічити час потрібно не тільки в науці й техніці, а й в повсякденному житті, враховуючи це, буде доцільно створити універсальний пристрій для відстежування часу.

Пристрій, що розробляється не повинен мати дуже високої точності, оскільки покликаний бути не повноцінним годинником, а швидше таймером чи секундоміром. Він може застосовуватись при приготуванні їжі, заняттях спортом та і в принципі для замірів тривалості якихось не надто довгих і не надто коротких процесів.

Отже я поставив перед собою наступну задачу: розробити пристрій, що лічитиме час в трьох режимах:

Перший режим – це секундомір, він лічить вверх в форматі ABCDE, де А – це десятки тисяч секунд, В – тисячі секунд, С – сотні секунд, D – десятки секунд, Е – секунди.

Другий режим – це секундомір-годинник, він лічить вверх в форматі АВ:CD:EF, де А – десятки годин, В – години, С – десятки хвилин, D – хвилини, Е – десятки секунд, F – секунди.

Третій режим – це таймер, він лічитиме вниз в такому ж форматі, як і в другому режимі, тобто АВ:CD:EF, де А – десятки годин, В – години, С – десятки хвилин, D – хвилини, Е – десятки секунд, F – секунди.

Однією з важливих функцій такого пристрою є можливість задати початкове значення часу. Для цього будуть використовуватись кнопки. Детальніше про те як це робити, можна дізнатись в «Розділ 4 Керівництво з експлуатації пристрою».

РОЗДІЛ 1 - ОПИС СТРУКТУРИ ПРИСТРОЮ ТА ЙОГО СКЛАДОВИХ

1.1 - Структура пристрою

Схематична структура пристрою зображена на рис. 1.1.

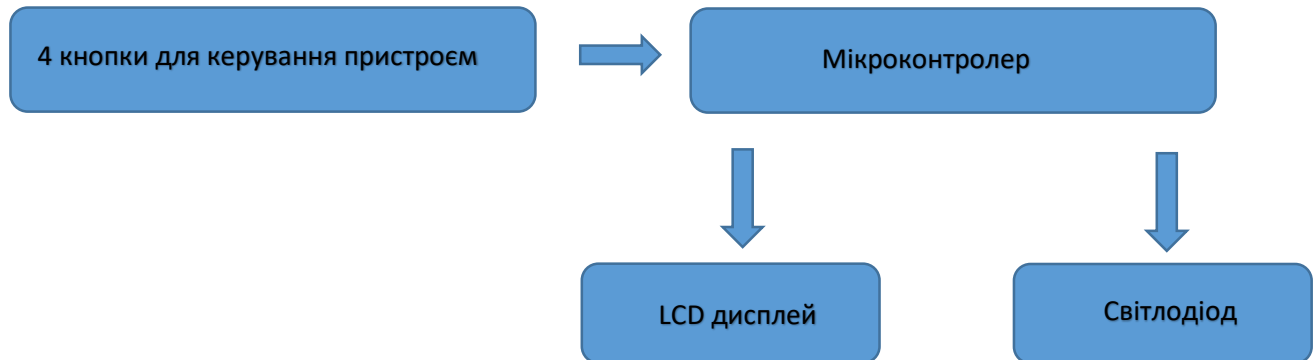


Рис. 1.1. Схематична структура пристрою, що розробляється

Як видно з рисунка, пристрій можна умовно поділити на чотири частини.

Першою частиною є пристрій для введення інформації. Цими пристроями будуть 5 кнопок. Чотири з них це ті, за допомогою яких здійснюється керування, а саме вибір режиму, старт/стоп, встановлення початкового значення часу. Також буде одна кнопка для скидання пристрою, тобто перезавантаження.

Другою частиною є мікроконтролер, саме він здійснює всі обчислення, лічить час, обробляє вхідні данні та визначає значення на портах вихідних даних. Як вже відомо, на роль цього компоненту було обрано STM32F407VGT.

Третьою частиною є LCD дисплей формату 1602. Це такий дисплей, що складається з двох рядочків, в кожному з яких може міститись 16 символів. Для відображення кожного символу відводиться площадка 8 на 5 пікселів. На цей екранчик пристрій виводить інформацію про вибраний режим, налаштування часу та сам час.

Четвертою частиною пристрою є світлодіод. Даний світлодіод буде світитись тоді коли йде лічення часу. Тобто, якщо світлодіод горить – пристрій працює, якщо ні – пристрій на паузі.

1.2 - Принципи та засоби введення даних

При створенні керувався наступними принципами щодо засобів введення:

- Невелика кількість кнопок
- Окрема кнопка для перезавантаження пристрою
- Окрема кнопка старт\стоп

Оскільки я хотів добитись невеликої кількості задіяних кнопок, то деякі з них будуть мати різний функціонал в залежності від стану пристрою.

Детальніше про те, як саме виглядає управління пристроєм можна прочитати в «Розділ 4 Керівництво з експлуатації пристрою».

1.3 - Принципи та засоби відображення даних

Основну інформацію про засоби виведення даних я вже виклав в «1.1 - Структура пристрою», тому в цьому розділі я лише добавлю кілька слів про дисплей. Для того, щоб було легше звикнути до пристрою, вибраний режим завжди висвітлюється в верхньому рядку пристрою (Рис 1.2). Якщо ж пристрій НЕ в тому стані, в якому встановлюється початкове значення часу, то сам час, відображається по середині нижнього рядочка (Рис 1.2).



Рис 1.2 Вигляд дисплею при ліченні в режимі секундоміра-годинника

Якщо ж вибраний такий стан, в якому встановлюється початкове значення часу, то час відображається на початку нижнього рядка, а значення, на яке буде змінюватись час, буде відображатись разом з знаком («-» або «+») в кінці нижнього рядка (Рис 1.3).

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		6



Рис 1.3 Вигляд дисплею тому в стані, в якому встановлюється початкове значення часу

Зауважу, що те число, на котре змінюється час, це приріст секунд. Число може бути як додатне, так і від'ємне.

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		7

РОЗДІЛ 2 - РОЗРОБКА СХЕМИ ЕЛЕКТРИЧНОЇ ПРИНЦИПОВОЇ

2.1 - Вибір та обґрунтування елементної бази

2.1.1 – Вибір мікросхеми

Для такого простого пристрою, як мій, зовсім не потрібно вибирати потужний мікроконтролер. Мною було обрано STM32F407VGT (Рис 2.1), оскільки саме до нього в мене був доступ. При перенесенні проекту на інший, слід буде переписати код, втім, змін найбільше зазнають значення регістрів в пам'яті та порти, а сам алгоритм, швидше за все, буде незмінним.



Рис 2.1 Мікроконтролер STM32F407VGT на платі STM32F4DISCOVERY

2.1.2 – Вибір LCD екрану

Я використовував, той, що знаходиться на стенді Global Starter Kit (Рис 2.2).

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		8

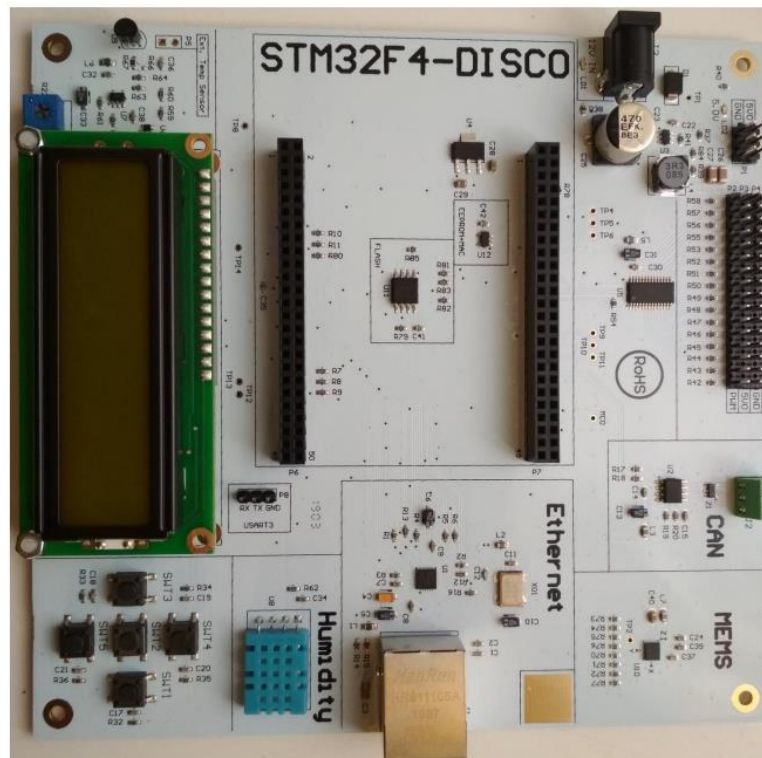


Рис 2.2 LCD дисплей WH1602B-NYG-CT на стенді Global Starter Kit

При зміні дисплея на інший, слід врахувати, що для іншого, швидше за все, буде потрібно переписати бібліотеку lcd.c (Додаток Б).

2.1.3 – Вибір котушок індуктивності

Котушка індуктивності, яку я використовую входить до **схеми стабілізації** напруги для мікроконтролера STM32F407VGT на платі STM32F4DISCOVERY (Рис 2.3), тому я не рекомендую відходити від їх схеми.

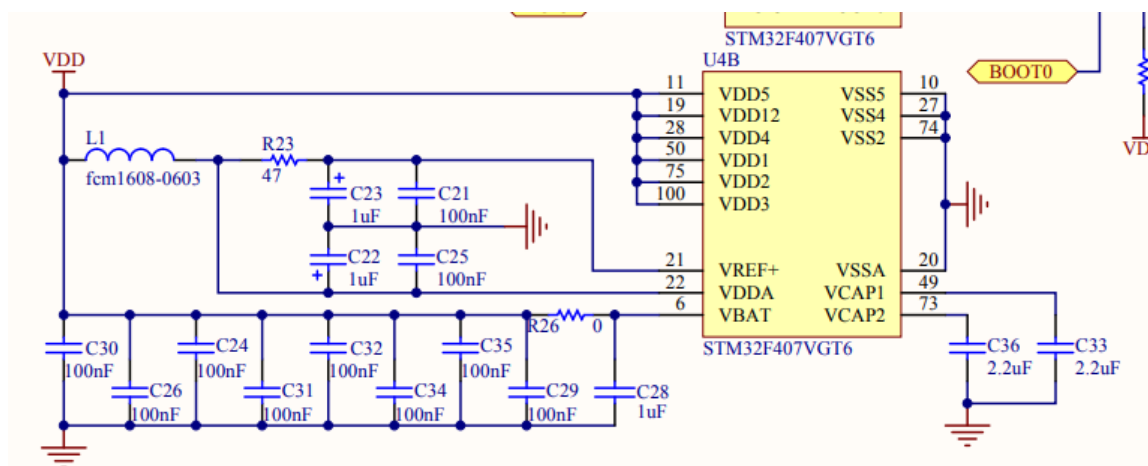


Рис 2.3 – стабілізація напруги для мікроконтролера STM32F407VGT на платі STM32F4DISCOVERY

2.1.4 – Вибір перемикачів

Кнопкою, що перезавантажує пристрій є B2 RESET з плати STM32F4DISCOVERY (Рис 2.4).

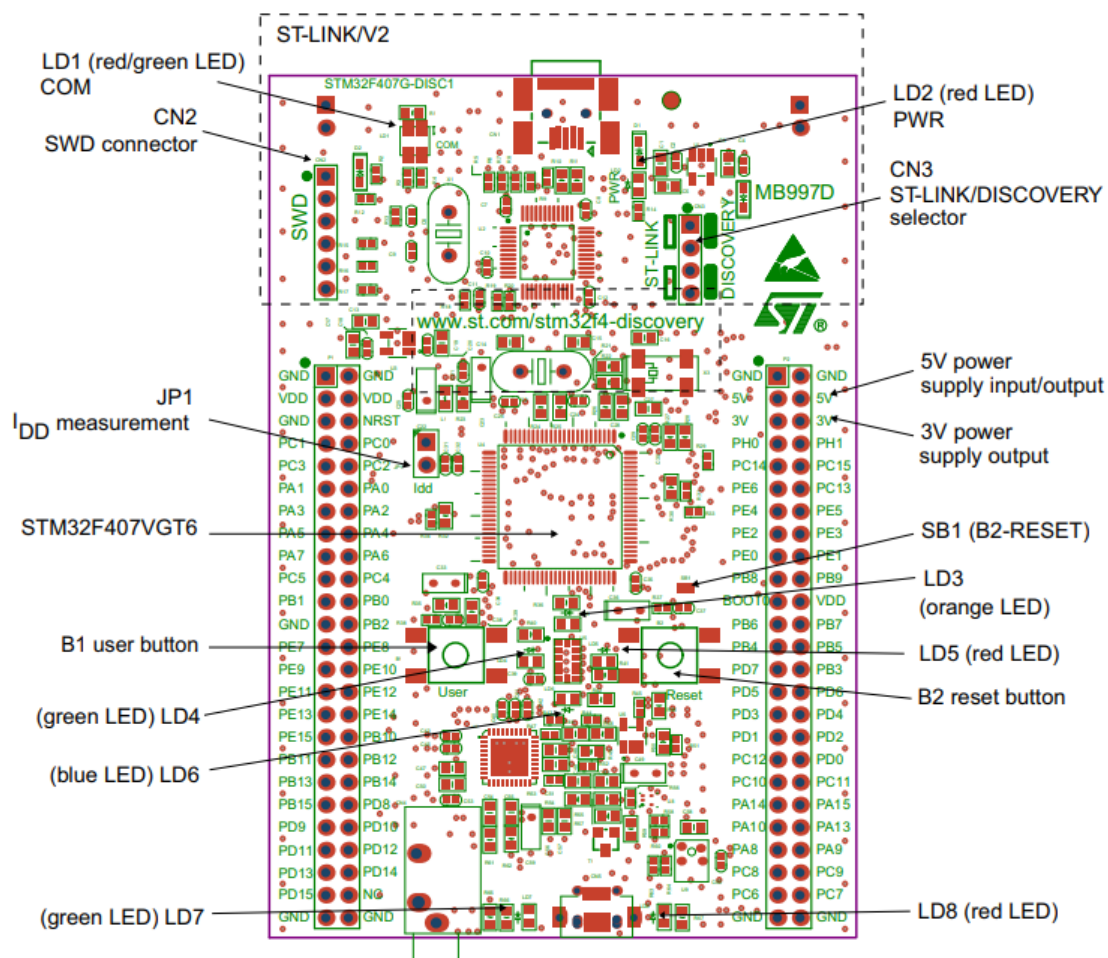


Рис 2.4 Схема плати STM32F4DISCOVERY

З документації на плату STM32F4DISCOVERY можна зрозуміти, що дана кнопка має «підтяжку на живлення» (Рис 2.5).

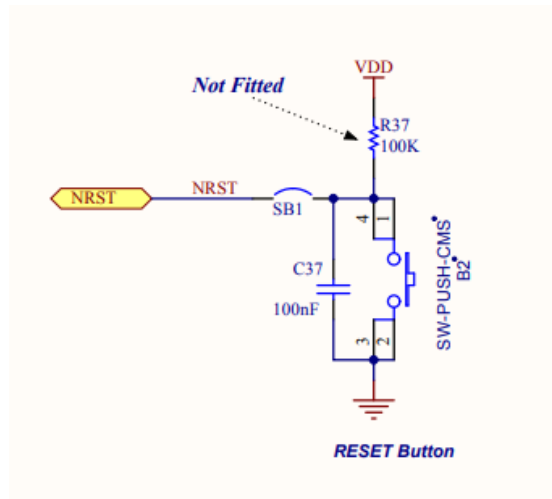


Рис 2.5 Фрагмент схеми плати STM32F4DISCOVERY

Решта ж кнопок це SWT1, SWT2, SWT4, SWT5 з стенду Global Starter Kit (Рис 2.6).

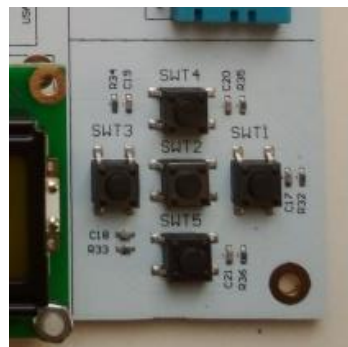


Рис 2.6 Кнопки SWT1, SWT2, SWT4, SWT5 з стенду Global Starter Kit

Зазначу, що в даних кнопок є «підтяжка на живлення» (Рис 5.7).

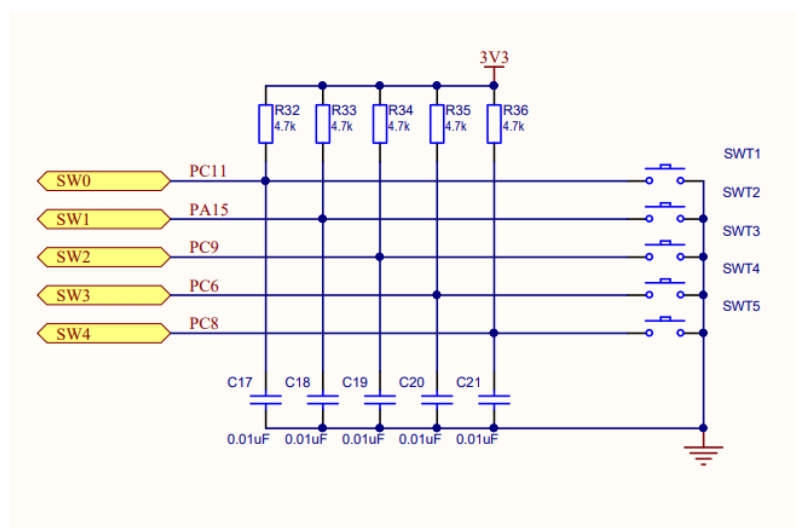


Рис 2.7 Фрагмент схеми стенду Global Starter Kit

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		11

Звичайно, ніяких особливих вимог, що кнопок нема, і їх можна замінити практично будь якими.

2.1.5 – Вибір транзисторів

При створенні прототипу пристрою, я використовував готову схему на стенді Global Starter Kit, тобто транзистор IRLML2402TRPBF, але загалом може підійти будь який інший такого ж типу.

2.1.6 – Вибір кварцових резонаторів

При створенні прототипу пристрою, я використовував палату STM32F4DISCOVERY, а там використовуються MC306-G-06Q-32.768 JFVNY та J49SMH-P-G-G-K-8M0 Jauch Quartz.

2.1.7 – Вибір конденсаторів

Основні характеристики конденсаторів є в документі ПЕЗ. Загалом, можна підбирати будь які зі схожими характеристиками.

2.1.8 – Вибір резисторів

Основні характеристики резисторів є в документі ПЕЗ. Загалом, можна підбирати будь які зі схожими характеристиками.

2.2 - Опис схеми електричної принципової

Переглянути схему електричну принципову в ПДФ форматі, можна в документі ЕЗ. Наведу «скріншот» на Рис 2.8.

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		12

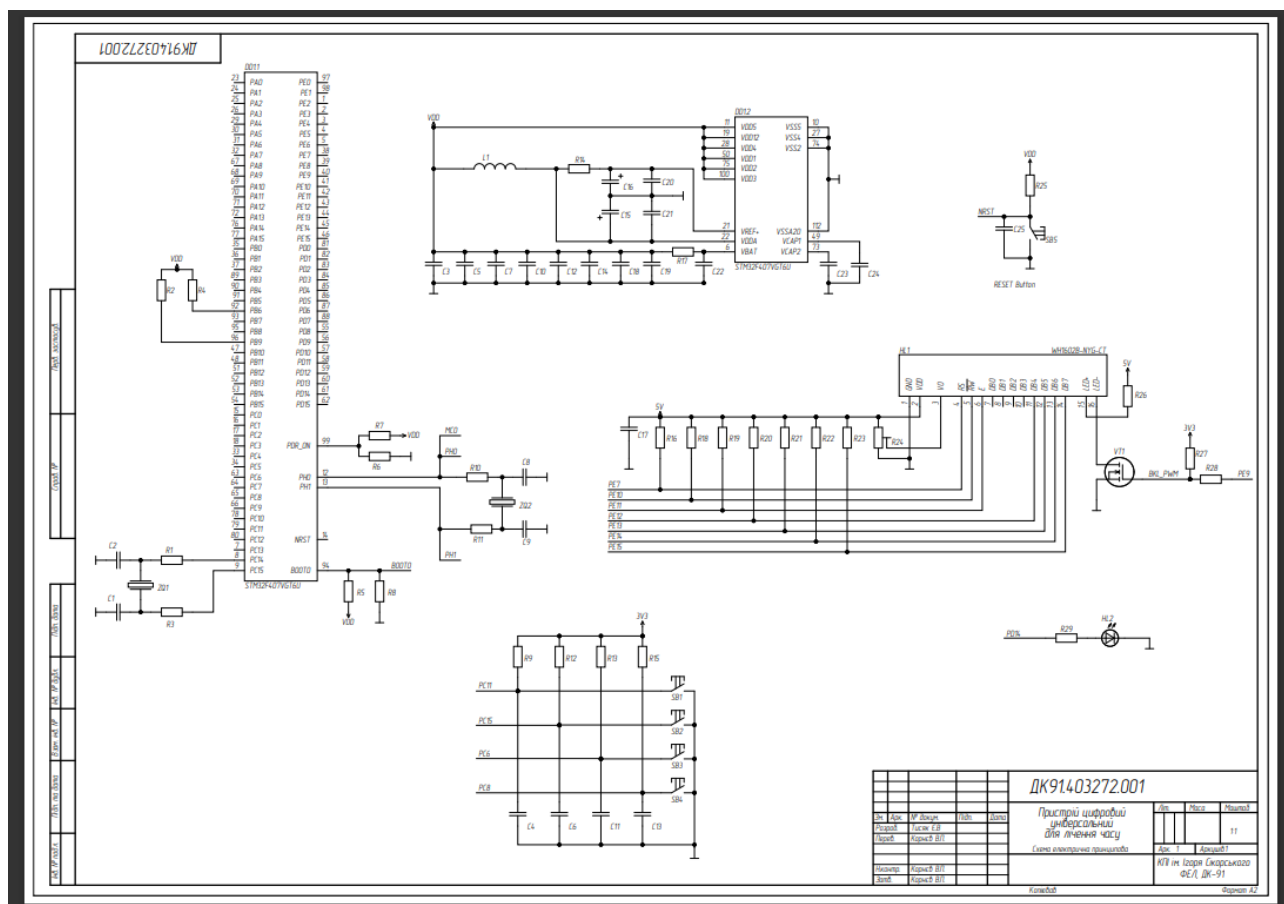
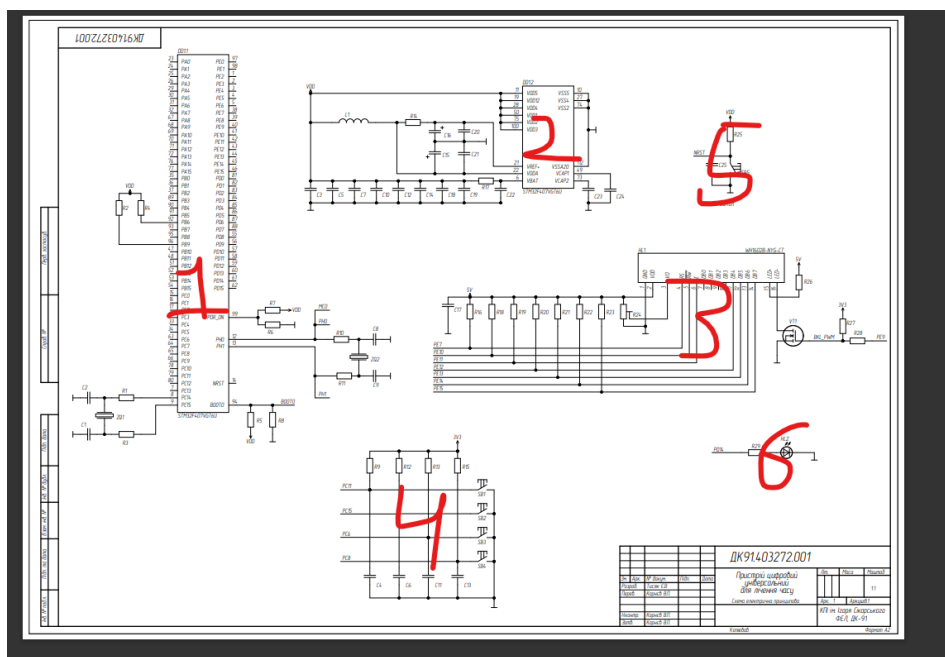


Рис 2.8 Документ ЕЗ

Як можна побачити з Рис 2.8 ЕЗ умовно ділиться на 6 менших частин. Наведу їх нумерування (Рис 2.9).



Розглянемо їх окремо:

Перша частина містить в собі частину МК STM32F407VGT, під'єднані до нього кварцові резонатори з відповідними допоміжними елементами (резисторами та конденсаторами), допоміжні резистори.

Друга частина містить також частину МК STM32F407VGT та інші елементи, що відповідають за стабілізацію напруги живлення для МК STM32F407VGT.

Третя частина містить в собі прилад LCD, а саме WH1602B-NYG-CT, виводи GPIO МК STM32F407VGT до котрих під'єднаний дисплей та їх допоміжні елементи (резистори, конденсатори, транзистор).

Четверта частина показує кнопки (всі окрім тієї, що для перезавантаження), їх підключення, та їх «підтяжку». Зверну вашу увагу, на те, що для переривань я використовую лише GPIO A. Підключення кнопок саме до цих GPIO, це особливість Global Starter Kit.

П'ята частина показує на підключення кнопки, що перезавантажує МК STM32F407VGT, відповідно весь пристрій.

Шоста частина показує підключення світлодіоду. Таке підключення продиктоване використанням готового світлодіоду на STM32F4DISCOVERY (Рис 2.10).

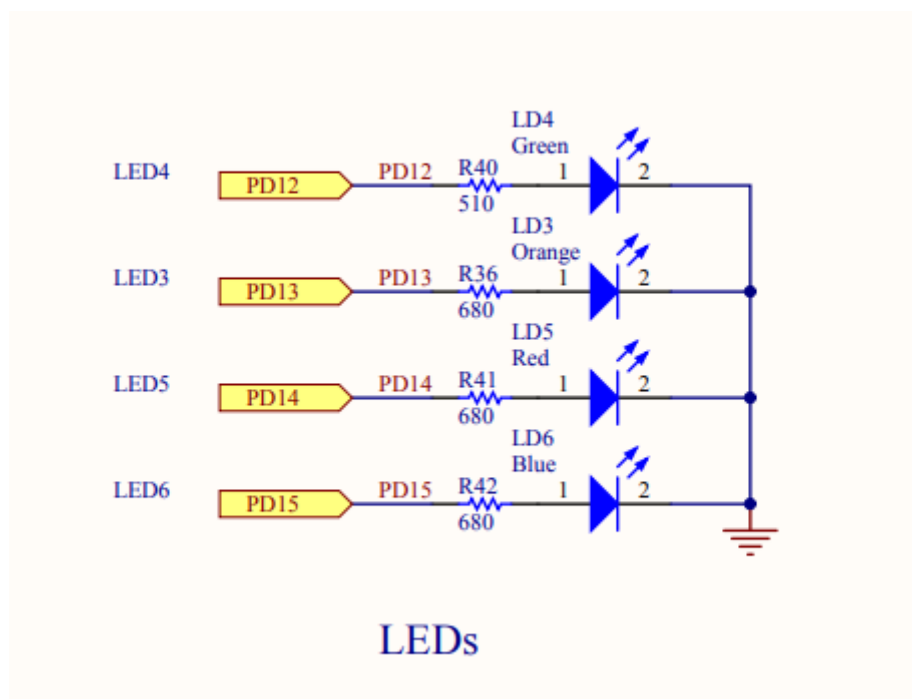


Рис 2.10 Фрагмент схеми STM32F4DISCOVERY

РОЗДІЛ 3 СТРУКТУРА ПРОГРАМИ ТА ЇЇ ОПИС

3.1 – Функції ініціалізації

`void init_PD_14(void)`; – функція, що ініціалізує порт GPIOD 14 як вихідний.

`void init_GPIO_for_interrupt(void)`; – функція, що ініціалізує порти GPIOA 0, 2, 3, 15 для їх використання, як зовнішні переривання.

`void init_interrupt(void)`; – функція, що ініціалізує зовнішні переривання, а саме EXTI0_IRQHandler, EXTI2_IRQHandler, EXTI3_IRQHandler, EXTI15_10_IRQHandler.

`void lcd_init(void)`; – функція, що ініціалізує LCD дисплей, тобто приводить його в готовність сприймати інші команди.

`void SysTick_Init (uint32 ticks)`; – функція для початку роботи з системним таймером.

3.2 – Функції для роботи з дисплеєм

`void lcd_cursor_on(void)`; – функція, що вмикає курсор на LCD дисплеї.

`void lcd_cursor_off(void)`; – функція, що вимикає курсор на LCD дисплеї.

`void lcd_cursor_to_home(void)`; – функція, що повертає курсор в нульову позицію.

`void lcd_clear(void)`; – функція, що очищає пам'ять дисплею та повертає курсор в нульову позицію.

`void lcd_send_cmd(unsigned char sign)`; – функція, що надсилає отриманий аргумент як команду.

`void lcd_send_data(unsigned char sign)`; – функція, що надсилає отриманий аргумент як данні.

`void lcd_print(void)`; – функція, що виводить потрібну інформацію в залежності від стану пристрою.

`void lcd_mod_print(void)`; – функція, що виводить вибраний режим пристрою.

`void lcd_time_print(void)`; – функція, що виводить час в залежності від обраного режиму.

`void lcd_change_print(void)`; – функція, що виводить інтерфейс для встановлення початкового значення лічильників.

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		15

3.3 – Функції зовнішніх переривань

void EXTI0_IRQHandler(void); – функція, що спрацьовує при негативному фронті на GPIOA0. Використовується для зміни режиму роботи пристрою.

void EXTI2_IRQHandler(void); – функція, що спрацьовує при негативному фронті на GPIOA2. Використовується для входу в режим встановлення початкового часу, та для встановлення величини приросту часу.

void EXTI3_IRQHandler(void); – функція, що спрацьовує при негативному фронті на GPIOA3. Використовується для додавання величини приросту часу до значення лічильника.

void EXTI15_10_IRQHandler(void); – функція, що спрацьовує при негативному фронті на GPIOA15. Використовується як кнопка старт/стоп.

3.4 – Функції системного таймера

void ANB_prescale_set_2(void); – функція, що записує в регістр CFGR в біти з 7 по 4 значення 1000, що призводить до поділу частоти системних тіків вдвічі.

void SysTick_Handler(void); – функція переривання системного таймера.

3.5 – Загальний алгоритм роботи

Спрощено алгоритм роботи виглядає так:

Спершу проводимо всі необхідні ініціалізації, ставимо пристрій на паузу та виводимо на екран початкові данні.

З цього моменту і завжди в подальшому, з кожним перериванням по системному таймеру, за умови, що пристрій не в режимі «стоп» та в залежності від обраного режиму лічення часу, відбувається збільшення чи зменшення значення відповідного лічильника на одиницю.

Зауважу, що значення, яке встановлюється в системний таймер після його переривання, є підібраним імперичним шляхом. Отриманої точності вистачає для виконання приладом своїх задач.

Все інше, що може ставатись з приладом, це зовнішні переривання. Саме вони вже визначатимуть подальший шлях його роботи.

РОЗДІЛ 4 КЕРІВНИЦТВО З ЕКСПЛУАТАЦІЇ ПРИСТРОЮ

4.1 – Запуск приладу

Для того, щоб запустити прилад, слід:

- Зібрати відповідну схему
- Завантажити прошивку на МК STM32F407VGT
- Перезавантажити пристрій натисканням кнопки SB5 (може знадобитись два натискання)

Після цього пристрій покаже на дисплеї наступну інформацію:



Рис 4.1 Вигляд пристрою одразу після перезавантаження

4.2 – Вибір режиму

Для того, щоб змінити режим, слід натискати кнопку SB1. Режими будуть перемикатись по колу.

Коли ви оберете потрібний вам режим, натисніть кнопку SB2 для того, щоб почати лік. Для того, щоб зупинити лік, натисніть кнопку SB2 ще раз. Так можна робити по колу.

4.3 – Режим встановлення початкового значення

Якщо ви обрали режим таймера та натиснули «Старт», то могли помітити, що прилад одразу сам поставив паузу. Це тому що, значення лічильника таймера зараз нуль, відповідно він вже дорахував «до кінця».

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		17

Щоб це виправити слід встановити значення лічильника таймера в якесь число відмінне від нуля. Інструкція є наступною:

1. Натисканням кнопки SB3 переходимо в режим встановлення значення лічильника.

Ось результат після натискання клавіші SB3:



2. Для того, щоб збільшити значення приросту натискайте клавішу SB3. З кожним натисканням приріст буде збільшуватись в 10 разів, аж доки не досягнута 10 000 і тоді при наступному натисканні знову стане 1.

Ось результат після, ще трьох натискань клавіші SB3:



Зм.	Лист	№ докум.	Підпис	Дата

ДК91.403272.001 ПЗ

Лист

18

3. Для того, щоб збільшити результат на значення приросту натисніть клавішу SB4. При кожному натисканні цієї клавіші, лічильник буде збільшуватись (або зменшуватись в залежності від знаку) на значення приросту.

Ось результат після, ще трьох натискань клавіші SB4:



Бачимо, що значення часу збільшилось на 30 000 секунд.

$$\begin{array}{r} 20 \times 60 = \\ 1\,200 \\ \\ 8 \times 3600 = \\ 28\,800 \end{array}$$

4. Бувають випадки коли потрібно число не збільшити, а зменшити. Зменшимо дане значення на 100 секунд. Для цього, змінимо знак натисканням клавіші SB2.

Ось результат після натискання клавіші SB2:



5. Тепер встановимо значення приросту 100. Натиснемо для цього клавішу SB3 ще три рази.

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		19

Ось результат після, ще трьох натискань клавіші SB3:



6. Для того, щоб зменшити на 100 натиснемо клавішу SB4.

Ось результат після натискання клавіші SB4:



					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		20

ВИСНОВКИ

Отже, в результаті виконання даного курсового проекту мною був розроблений пристрій цифровий універсальний для лічення часу. Він містить три режими роботи, та як показує практичний досвід, задовільну точність. Відхилення в одну секунду появляється після 15-20 хвилин безперервної роботи. Дану точність можна легко збільшити підбором потрібного значення.

Основою для пристрою слугує МК STM32F407VGT на платі STM32F4DISCOVERY.

При розробці був створений простоти на основі стенду Global Starter Kit.

Разом з цією ПЗ також були створені документи ЕЗ та ПЕЗ.

Для зручності був добавлений Розділ 4, в якому я описав принципи користування приладом.

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		21

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-MQO0qDnYAWWngzQIEuoA3g&authuser=1>

- Дата звернення 18.01.2023.

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		22

ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ

ТЕХНІЧНЕ ЗАВДАННЯ

1. Найменування та галузь використання

- Пристрій цифровий універсальний таймер-секундомір для побутового використання

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 зручного й простого в використанні пристрою
- Створення пристрою, який легко повторити

4. Джерела розробки

- Пристрій розробляється вперше

5. Технічні вимоги

5.1. Функціональні можливості пристрою.

- Лічення часу вгору в форматі секунд
- Лічення часу вгору в форматі годинника
- Лічення часу вниз в форматі годинника

5.2. Технічні характеристики

- Достатня для побутового використання точність
- Зручний 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.

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		23

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.

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.

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		24

9. Економічні показники

В рамках даного проекту не розглядаються.

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

main.c

```
#include "func.h"

int main()
{
    init_PD_14();
    SysTick_Init(systick_init_value);
    init_interrupt();
    lcd_init();
    lcd_print();
    while(1);
}
```

func.h

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

typedef unsigned long unit32;

extern volatile bool work;
extern volatile bool change;
extern volatile bool sign_change_value;
extern volatile unit32 change_value;
extern volatile unit32 systick_init_value; // 0x008F0D18 = 500ms = 9 375 000
extern volatile unit32 counter_mod_0_buff;
extern volatile unit32 counter_mod_0; // Clock
```

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		26

```

extern volatile unit32 counter_mod_1; // Timer
extern volatile unit32 counter_mod_2; // Stopwatch
extern volatile unit32 mod;
extern volatile unit32 delay_counter;
extern volatile unit32 hour;
extern volatile unit32 minute;
extern volatile unit32 second;

//*****

void my_delay(unsigned long delay);
void init_PD_14(void);
void PD_14(bool set_reset);
void PE_7(bool set_reset);
void PE_10(bool set_reset);
void PE_11(bool set_reset);
void PE_12(bool set_reset);
void PE_13(bool set_reset);
void PE_14(bool set_reset);
void PE_15(bool set_reset);

//*****

//*****

void AHB_prescale_set_2(void);
void SysTick_Init (unit32 ticks);
void SysTick_Handler(void);

//*****

```

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		27

```

//*****

void init_GPIO_for_interrupt(void);
void init_interrupt(void);
void init_irq(void);

void EXTI0_IRQHandler(void);

void EXTI2_IRQHandler(void);
void EXTI3_IRQHandler(void);
void EXTI15_10_IRQHandler(void);

//*****

//*****

void lcd_init(void);
void lcd_gpio_init(void);
void lcd_cursor_on(void);
void lcd_cursor_off(void);
void lcd_cursor_to_home(void);
void lcd_clear(void);
void lcd_send_cmd(unsigned char sign);
void lcd_send_data(unsigned char sign);
void lcd_print(void);
void lcd_mod_print(void);
void lcd_time_print(void);
void lcd_change_print(void);

//*****

```

```
#include "func.h"
```

```
void init_PD_14(void)
```

```
{
```

```
    RCC->AHB1ENR |= RCC_AHB1ENR_GPIODEN;
```

```
    GPIOD->MODER |= GPIO_MODER_MODE14_0;
```

```
    GPIOD->MODER &= ~GPIO_MODER_MODE14_1;
```

```
}
```

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

```
{
```

```
    delay_counter = 0;
```

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

```
}
```

```
void PD_14(bool set_reset)
```

```
{
```

```
    if (set_reset) GPIOD->ODR |= GPIO_ODR_OD14;
```

```
    else GPIOD->ODR &= ~GPIO_ODR_OD14;
```

```
}
```

```
void PE_7(bool set_reset)
```

```
{
```

```
    if (set_reset) GPIOE->ODR |= GPIO_ODR_OD7;
```

```
    else GPIOE->ODR &= ~GPIO_ODR_OD7;
```

```
}
```



```

void PE_10(bool set_reset)
{
    if (set_reset) GPIOE->ODR |= GPIO_ODR_OD10;
    else GPIOE->ODR &= ~GPIO_ODR_OD10;
}

void PE_11(bool set_reset)
{
    if (set_reset) GPIOE->ODR |= GPIO_ODR_OD11;
    else GPIOE->ODR &= ~GPIO_ODR_OD11;
}

void PE_12(bool set_reset)
{
    if (set_reset) GPIOE->ODR |= GPIO_ODR_OD12;
    else GPIOE->ODR &= ~GPIO_ODR_OD12;
}

void PE_13(bool set_reset)
{
    if (set_reset) GPIOE->ODR |= GPIO_ODR_OD13;
    else GPIOE->ODR &= ~GPIO_ODR_OD13;
}

void PE_14(bool set_reset)
{
    if (set_reset) GPIOE->ODR |= GPIO_ODR_OD14;
    else GPIOE->ODR &= ~GPIO_ODR_OD14;
}

```

```

void PE_15(bool set_reset)
{
    if (set_reset) GPIOE->ODR |= GPIO_ODR_OD15;
    else GPIOE->ODR &= ~GPIO_ODR_OD15;
}

```

systick.c

```

#include "func.h"

```

```

void AHB_prescale_set_2(void)
{
    RCC->CFGR |= 0x00000080;
    RCC->CFGR &= ~0x00000040;
    RCC->CFGR &= ~0x00000020;
    RCC->CFGR &= ~0x00000010;
}

```

```

void SysTick_Init (uint32 ticks)
{
    AHB_prescale_set_2();
    SysTick->CTRL = 0;
    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;
}

```

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		31

```

}

void SysTick_Handler()
{
    if(work)
    {
        switch(mod)
        {
            case 0:
                counter_mod_0++;
                break;

            case 1:
                counter_mod_1--;
                break;

            case 2:
                counter_mod_2++;
                break;
        }
    }
    PD_14(work);
    lcd_time_print();
}

```

interrupt.c

#include "func.h"

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		32

```
void init_GPIO_for_interrupt(void)
```

```
{
```

```
    RCC->AHB1ENR |= RCC_AHB1ENR_GPIOAEN;
```

```
    GPIOA->MODER &= ~GPIO_MODER_MODE0_0;
```

```
    GPIOA->MODER &= ~GPIO_MODER_MODE0_1;
```

```
    GPIOA->MODER &= ~GPIO_MODER_MODE2_0;
```

```
    GPIOA->MODER &= ~GPIO_MODER_MODE2_1;
```

```
    GPIOA->MODER &= ~GPIO_MODER_MODE3_0;
```

```
    GPIOA->MODER &= ~GPIO_MODER_MODE3_1;
```

```
    GPIOA->MODER &= ~GPIO_MODER_MODE15_0;
```

```
    GPIOA->MODER &= ~GPIO_MODER_MODE15_1;
```

```
}
```

```
void init_interrupt(void)
```

```
{
```

```
    init_GPIO_for_interrupt();
```

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

```
    init_irq();
```

```
    __enable_irq();
```

```
}
```

```
void init_irq(void)
```

```
{
```

```
    SYSCFG->EXTICR[0] |= SYSCFG_EXTICR1_EXTI0_PA;
```

```
    SYSCFG->EXTICR[0] |= SYSCFG_EXTICR1_EXTI2_PA;
```

```
    SYSCFG->EXTICR[0] |= SYSCFG_EXTICR1_EXTI3_PA;
```

```
    SYSCFG->EXTICR[3] |= SYSCFG_EXTICR4_EXTI15_PA;
```

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		33

```
EXTI->IMR |= EXTI_IMR_IM0;
EXTI->IMR |= EXTI_IMR_IM2;
EXTI->IMR |= EXTI_IMR_IM3;
EXTI->IMR |= EXTI_IMR_IM15;
```

```
EXTI->FTSR |= EXTI_RTSMR_TR0;
EXTI->FTSR |= EXTI_RTSMR_TR2;
EXTI->FTSR |= EXTI_RTSMR_TR3;
EXTI->FTSR |= EXTI_RTSMR_TR15;
```

```
NVIC_SetPriority(EXTI0_IRQn,1);
NVIC_SetPriority(EXTI2_IRQn,3);
NVIC_SetPriority(EXTI3_IRQn,4);
NVIC_SetPriority(EXTI15_10_IRQn,5);
```

```
NVIC_ClearPendingIRQ(EXTI0_IRQn);
NVIC_ClearPendingIRQ(EXTI2_IRQn);
NVIC_ClearPendingIRQ(EXTI3_IRQn);
NVIC_ClearPendingIRQ(EXTI15_10_IRQn);
```

```
NVIC_EnableIRQ(EXTI0_IRQn);
NVIC_EnableIRQ(EXTI2_IRQn);
NVIC_EnableIRQ(EXTI3_IRQn);
NVIC_EnableIRQ(EXTI15_10_IRQn);
```

```
}
```

```
void EXTI0_IRQHandler()
```

```
{
```

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		34

```
EXTI->IMR &= ~EXTI_IMR_IM0;
```

```
if(!work && !change)
```

```
{
```

```
    mod++;
```

```
    if(mod>2) mod = 0;
```

```
    lcd_print();
```

```
}
```

```
else if(!work && change)
```

```
{
```

```
    sign_change_value = !sign_change_value;
```

```
}
```

```
EXTI->PR|= EXTI_PR_PR0;
```

```
EXTI->IMR |= EXTI_IMR_IM0;
```

```
}
```

```
void EXTI2_IRQHandler()
```

```
{
```

```
    EXTI->IMR &= ~EXTI_IMR_IM2;
```

```
    change = 1;
```

```
    change_value = change_value*10;
```

```
    if(change_value>10000) change_value = 1;
```

```
    lcd_time_print();
```

```
    EXTI->PR|= EXTI_PR_PR2;
```

```
    EXTI->IMR |= EXTI_IMR_IM2;
```

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		35

```

}

void EXTI3_IRQHandler()
{
    EXTI->IMR &= ~EXTI_IMR_IM3;

    if(change)
    {
        switch(mod)
        {
            case 0:
                if(sign_change_value)counter_mod_0 -= change_value;
                else counter_mod_0 += change_value;
                break;

            case 1:
                if(sign_change_value)counter_mod_1 -= change_value;
                else counter_mod_1 += change_value;
                break;

            case 2:
                if(sign_change_value)counter_mod_2 -= change_value;
                else counter_mod_2 += change_value;
                break;
        }
    }

    lcd_time_print();
}

```

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		36

```

EXTI->PR|= EXTI_PR_PR3;
EXTI->IMR |= EXTI_IMR_IM3;
}

void EXTI15_10_IRQHandler()
{
    EXTI->IMR &= ~EXTI_IMR_IM15;

    change = 0;
    change_value = 1;
    work = !work;
    PD_14(work);
    lcd_print();

    EXTI->PR|= EXTI_PR_PR15;
    EXTI->IMR |= EXTI_IMR_IM15;
}

```

lcd.c

```

#include "func.h"

void lcd_init(void)
{
    lcd_gpio_init();
    lcd_send_cmd(0x28);
    lcd_clear();
    lcd_cursor_off();
}

```

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		37


```

    lcd_cursor_to_home();
}

void lcd_gpio_init(void)
{
    RCC->AHB1ENR |= RCC_AHB1ENR_GPIOEEN;

    // MODER
    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;
    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;

```

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		38

```

// OTYPER
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;

// Reset PE10
PE_10(0);
}

void lcd_cursor_on(void)
{
    lcd_send_cmd(0x0D);
}

void lcd_cursor_off(void)
{
    lcd_send_cmd(0x0C);
}

void lcd_cursor_to_home(void)
{
    lcd_send_cmd(0x02);
}

```

```
void lcd_clear(void)
```

```
{
```

```
    lcd_send_cmd(0x01);
```

```
}
```

```
void lcd_send_cmd(unsigned char sign)
```

```
{
```

```
    PE_7(0);
```

```
    PE_11(0);
```

```
    PE_12(sign & (1<<4));
```

```
    PE_13(sign & (1<<5));
```

```
    PE_14(sign & (1<<6));
```

```
    PE_15(sign & (1<<7));
```

```
    PE_11(1);
```

```
    my_delay(40);
```

```
    PE_11(0);
```

```
    my_delay(40);
```

```
    PE_12(sign & (1<<0));
```

```
    PE_13(sign & (1<<1));
```

```
    PE_14(sign & (1<<2));
```

```
    PE_15(sign & (1<<3));
```

```
    PE_11(1);
```

```
    my_delay(40);
```

					ДК91.403272.001 ПЗ	Лист
						40
Зм.	Лист	№ докум.	Підпис	Дата		

```

    PE_11(0);

    my_delay(40000);
}

void lcd_send_data(unsigned char sign)
{
    PE_7(1);

    PE_11(0);

    PE_12(sign & (1<<4));
    PE_13(sign & (1<<5));
    PE_14(sign & (1<<6));
    PE_15(sign & (1<<7));

    PE_11(1);
    my_delay(40);
    PE_11(0);

    my_delay(40);
    PE_12(sign & (1<<0));
    PE_13(sign & (1<<1));
    PE_14(sign & (1<<2));
    PE_15(sign & (1<<3));

    PE_11(1);
    my_delay(40);

```

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		41

```

        PE_11(0);

        my_delay(400);
    }

void lcd_mod_print(void)
{
    lcd_cursor_to_home();
    lcd_send_data('M');
    lcd_send_data('o');
    lcd_send_data('d');
    lcd_send_data(':');
    lcd_send_data(' ');
    switch(mod)
    {
        case 0:
            lcd_send_data('S');
            lcd_send_data('t');
            lcd_send_data('o');
            lcd_send_data('p');
            lcd_send_data('w');
            lcd_send_data('a');
            lcd_send_data('t');
            lcd_send_data('c');
            lcd_send_data('h');

            break;
        case 1:
            lcd_send_data('T');

```

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		42

```

        lcd_send_data('i');
        lcd_send_data('m');
        lcd_send_data('e');
        lcd_send_data('r');
    break;
case 2:
        lcd_send_data('C');
        lcd_send_data('l');
        lcd_send_data('o');
        lcd_send_data('c');
        lcd_send_data('k');
    break;
}
}

```

```
void lcd_time_print(void)
```

```

{
    if(!work && !change)
    {
        lcd_send_cmd(0xC2);
        lcd_send_data('=');
        lcd_send_data('=');
    }
    if(!work && change)
    {
        lcd_send_cmd(0xC0);
    }
}

```

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		43

```

if(work && !change) lcd_send_cmd(0xC4);

switch(mod)
{
    case 0:
        if(counter_mod_0>=100000)
        {
            counter_mod_0 = 0;
            work = 0;
        }
        lcd_send_data(0x30 + (counter_mod_0 -
(counter_mod_0%10000))/10000);
        lcd_send_data(0x30 + ((counter_mod_0 -
(counter_mod_0%1000))/1000)%10);
        lcd_send_data(0x30 + ((counter_mod_0 -
(counter_mod_0%100))/100)%10);
        lcd_send_data(0x30 + ((counter_mod_0 -
(counter_mod_0%10))/10)%10);
        lcd_send_data(0x30 + (counter_mod_0%10));

        break;

    case 1:
        if(counter_mod_1<=0) work = 0;
        second = counter_mod_1 % 60;
        hour = (counter_mod_1 - (counter_mod_1 % 3600)) / 3600;
        minute = ((counter_mod_1 % 3600) - second)/60;
        lcd_send_data(0x30 + ((hour - (hour%10))/10));
        lcd_send_data(0x30 + (hour%10));

```

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		44

```

        lcd_send_data(0x3A);
        lcd_send_data(0x30 + ((minute - (minute%10))/10));
        lcd_send_data(0x30 + (minute%10));
        lcd_send_data(0x3A);
        lcd_send_data(0x30 + ((second - (second%10))/10));
        lcd_send_data(0x30 + (second%10));

break;

case 2:
    if(hour>=24)
    {
        hour = 0;
        counter_mod_2 = 0;
        work = 0;
    }
    second = counter_mod_2 % 60;
    hour = (counter_mod_2 - (counter_mod_2 % 3600)) / 3600;
    minute = ((counter_mod_2 % 3600) - second)/60;
    lcd_send_data(0x30 + ((hour - (hour%10))/10));
    lcd_send_data(0x30 + (hour%10));
    lcd_send_data(0x3A);
    lcd_send_data(0x30 + ((minute - (minute%10))/10));
    lcd_send_data(0x30 + (minute%10));
    lcd_send_data(0x3A);
    lcd_send_data(0x30 + ((second - (second%10))/10));
    lcd_send_data(0x30 + (second%10));

break;
}

```

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		45


```

    if(!work && change)
    {
        lcd_send_data(' ');
        lcd_send_data(' ');
        lcd_change_print();
    }

    if(!work && !change)
    {
        lcd_send_data('=');
        lcd_send_data('=');
    }

}

void lcd_change_print(void)
{
    if(sign_change_value) lcd_send_data('-');
    else lcd_send_data('+');

    lcd_send_data(0x30 + (change_value - (change_value%10000))/10000);
    lcd_send_data(0x30 + ((change_value - (change_value%1000))/1000)%10);
    lcd_send_data(0x30 + ((change_value - (change_value%100))/100)%10);
    lcd_send_data(0x30 + ((change_value - (change_value%10))/10)%10);
    lcd_send_data(0x30 + (change_value%10));
}

void lcd_print()

```

	{				ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		46

```

    lcd_clear();
    lcd_mod_print();
    lcd_time_print();
}

```

variables.c

```
#include "func.h"
```

```

volatile bool work = 0;
volatile bool change = 0;
volatile bool sign_change_value = 0; // if 0 then positive

```

```
volatile unit32 systick_init_value = 0x007AFAE1; // 9 371 648 8 059 617,28 7AFAE1
```

```

volatile unit32 change_value = 1;
volatile unit32 mod = 0;

```

```

volatile unit32 counter_mod_0 = 0; // Clock
volatile unit32 counter_mod_1 = 0; // Timer
volatile unit32 counter_mod_2 = 0; // Stopwatch

```

```
volatile unit32 delay_counter = 0;
```

```

volatile unit32 hour;
volatile unit32 minute;
volatile unit32 second;

```

					ДК91.403272.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		47