

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: « Табло для обмінника валют 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
Тисяка Євгенія Володимировича
(прізвище, ім'я та по батькові)

1. Тема проекту Табло для обмінника валют на основі STM32F407VG
2. Строк здачі студентом закінченого проекту (роботи) 28.01.2023
3. Вихідні дані до проекту (роботи)
Прилад повинен містити табло на котре буде виводитись співвідношення вартості фіксованих валют. Для того, щоб задавати всі необхідні параметри, пристрій повинен містити декілька кнопок. 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)

« 21 » січня 2023р.

ЗМІСТ

Перелік використаних скорочень.....	3
Вступ.....	3
Розділ 1 - Опис структури пристрою та його складових.....	5
1.1 - Структура пристрою.....	5
1.2 - Принципи та засоби введення даних.....	5
1.3 - Принципи та засоби відображення даних.....	8
1.4 - Принципи та засоби обробки даних.....	8
Розділ 2 - Розробка схеми електричної принципової.....	9
2.1 – Загальний опис схеми електричної принципової.....	9
2.2 – Опис та призначення деяких частин схеми	10
2.2.1 – Опис схеми стабілізації напруги мікроконтролера.....	10
2.2.2 - Опис схеми підключення кнопки перезавантаження мікроконтролера.....	11
2.2.3 - Опис схеми підключення кнопок управління.....	13
Розділ 3 - Структура програми та її опис.....	15
3.1 – Функції ініціалізації	15
3.2 – Функції для роботи з дисплеєм.....	17
3.3 – Функції зовнішніх переривань.....	19
3.4 – Загальний алгоритм роботи.....	20

ДК91.694159.001 ПЗ

Змн.	Арк.	№ докум.	Підпис	Дата	<div style="text-align: center;"> <p>Табло для обмінника валют на основі STM32F407VG</p> </div>		
Розробив	Тисяк Є.В.						
Перевірів	Корнєв ВП.						
Реценз.							
Н. Контр.	Корнєв ВП.						
Затвердив	Корнєв ВП.				<div style="text-align: center;"> <p>НТУУ «КПІ»</p> </div>		
					Літ.	Арк.	Аркцшів
						1	49

Розділ 4 - Керівництво з експлуатації пристрою.....	22
4.1 – Запуск приладу.....	22
4.2 – Зміна інформації, що відображається	22
Висновки	24
Список використаних джерел.....	25
Додаток А. Технічне завдання.....	26
Додаток Б. Лістинг програми.....	30

					ДК91.694159.001 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

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		3

ВСТУП

На сьогоднішній день маркетинг став дуже технологічним напрямком людської діяльності, в ньому задіюють багато різноманітних технологій для приваблення клієнтів. Прилад, що розробляється, також покликаний допомогти підприємцям в веденні бізнесу.

Отже, завданням є створення такого приладу, котрий буде відображати співвідношення вартості основних валют. Такий прилад може бути використаний обмінниками чи відділами банків в якості біл-борду, для того, щоб пересічні люди могли дізнатись потрібну інформацію прямо з вулиці.

Даний прилад повинен містити мінімум зайвої інформації, не повинен бути універсальним засобом для відображення інформації, а навпаки, бути заточеним під специфічну роботу. Отже, важливо одразу визначитись, що і в якому форматі буде відображатись. Моя реалізація буде детально описана в «Розділ 1 - Опис структури пристрою та його складових» цього ж документу.

Наступною важливою функцією є можливість для працівника змінити інформацію, що відображається, простим натисканням кнопок. Це важливо, оскільки співвідношення вартості валют є дуже не стабільна річ і в людей, що використовуватимуть пристрій, повинна бути можливість швидко й просто корегувати результати.

Важливо зазначити, що всі компоненти даного пристрою будуть знаходитись в захищеному від вологи й вітру прозорому корпусі, тому якихось особливих кліматичних вимог у нього не буде.

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		4

РОЗДІЛ 1 - ОПИС СТРУКТУРИ ПРИСТРОЮ ТА ЙОГО СКЛАДОВИХ

1.1 - Структура пристрою

Для початку наведу загальну структурну схему пристрою (Рис. 1.1):

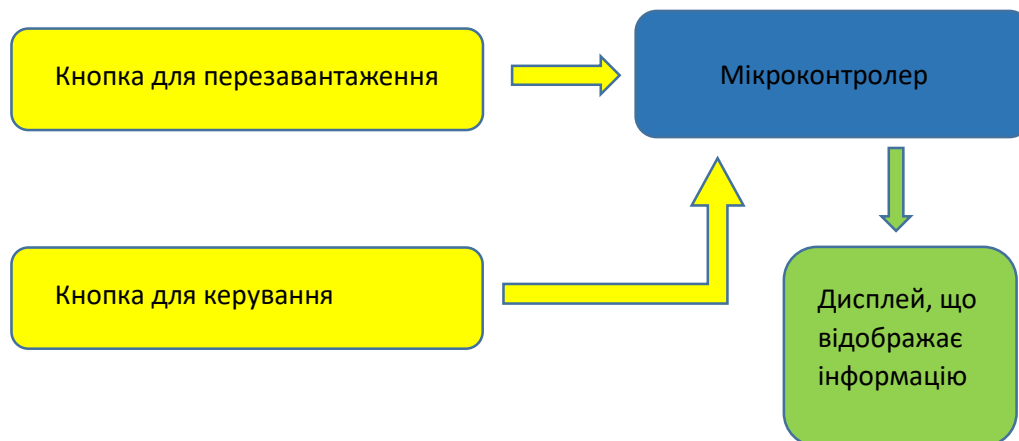


Рис. 1.1 Загальна структурна схема пристрою

Як можна побачити з Рис. 1.1 всі модулі, що відповідають за введення інформації мають жовтий колір прямокутників та стрілочок, синій колір в тих модулів, що обробляються інформацію та зелений в ти, що її виводять. Зазначу, що стрілочки вказують напрямок передачі інформації.

1.2 - Принципи та засоби введення даних

Розглянемо детальніше жовті модулі:

Першим таким модулем, є кнопка, що перезавантажує пристрів. Саме ця кнопка керує напругою на вході NRST мікроконтролера STM32F407VG. Детальніше про те, як вона влаштована буде в «2.2.2 - Опис кнопки перезавантаження мікроконтролера».

Другим модулем є три кнопки, за допомогою яких користувач зможе змінити інформацію, що відображається. Для зручності ці кнопки варто

розмістити в наступному порядку (Рис 1.2):

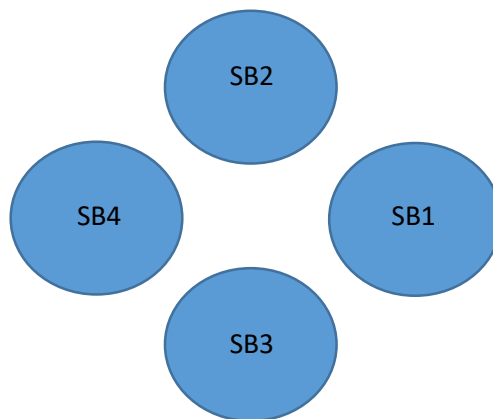


Рис. 1.2 Зручне розташування, для кнопок керування та кнопки перезавантаження

SB4 – це кнопка перезавантаження. SB1, SB2, SB3 – кнопки для керування. Про підключення SB1, SB2, SB3 можна почитати докладніше в «2.2.3 - Опис кнопок управління», про призначення й застосування цих кнопок сказано в «4.2 – Зміна інформації, що відображається».

При створенні прототипу (макету) я буду використовувати:

- кнопку B2 reset button з плати STM32F4DISCOVERY, як SB4.
- SBT1 з стенду Global Starter Kit, як SB1
- SBT4 з стенду Global Starter Kit, як SB2
- SBT5 з стенду Global Starter Kit, як SB3

Наведу зображення плати STM32F4DISCOVERY (Рис. 1.3) та стенду Global Starter Kit (Рис. 1.4) для кращого розуміння того, як виглядатиме макет.

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		6



Рис. 1.3 Плата STM32F4DISCOVERY

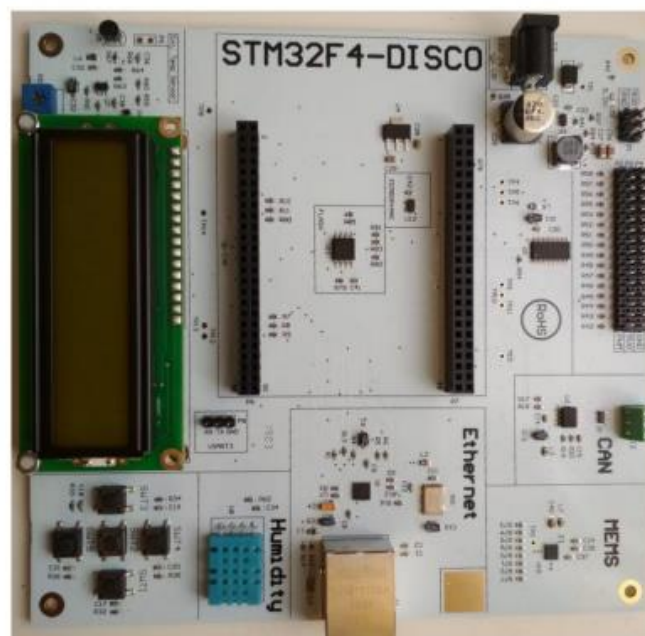


Рис. 1.4 Стенд Global Starter Kit

1.3 - Принципи та засоби відображення даних

Розглянемо зелений модуль (Рис. 1.1)

Для початку слід визначитись з форматом відображення. Оскільки прилад буде використовуватись як табло обмінника, було обрано наступний формат (Рис 1.5):

	КУПІВЛЯ	ПРОДАЖ
USD	XX.XX	XX.XX
USD	XX.XX	XX.XX

Рис. 1.5 Формат відображення інформації

Слова «КУПІВЛЯ» та «ПРОДАЖ» не будуть виведені на екран, а будуть написані на корпусі. Вся інша інформація повинна виводитись на відповідний засіб виведення. Таким чином, буде можливість відображати курс двох валют.

Звичайно варто вибрати найбільш популярні й вживані, наприклад «USD» та «EUR».

При створенні макету я буду виводити всю інформацію на LCD 1602, а саме WH1602B-NYG-CT на стенді Global Starter Kit (Рис 1.4):

1.4 - Принципи та засоби обробки даних

Розглянемо синій модуль (Рис. 1.1)

До модуля обробки даних немає строгих вимог, може підійти практично будь-який мікроконтролер.

При створенні макету я буду використовувати STM32F407VGT.

РОЗДІЛ 2 - РОЗРОБКА СХЕМИ ЕЛЕКТРИЧНОЇ ПРИНЦИПОВОЇ

2.1 – Загальний опис схеми електричної принципової

Наведу загальний вигляд схеми електричної принципової (Рис. 2.1):

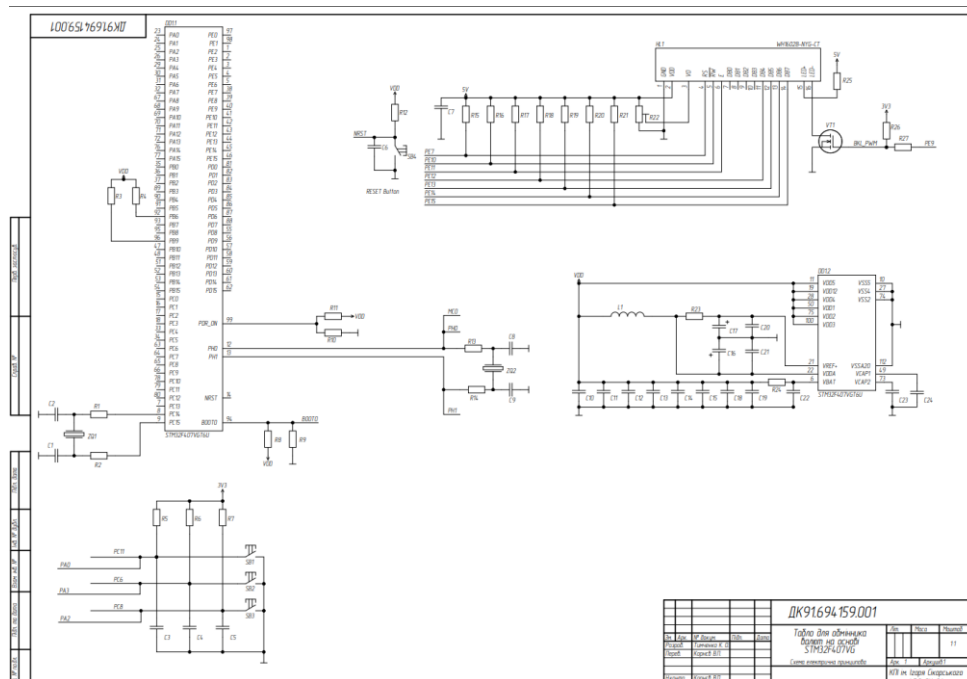


Рис. 2.1 Загальний вигляд схеми електричної принципової

Для загального, аналізу розіб'ю схему на п'ять основних частин (Рис. 2.2)

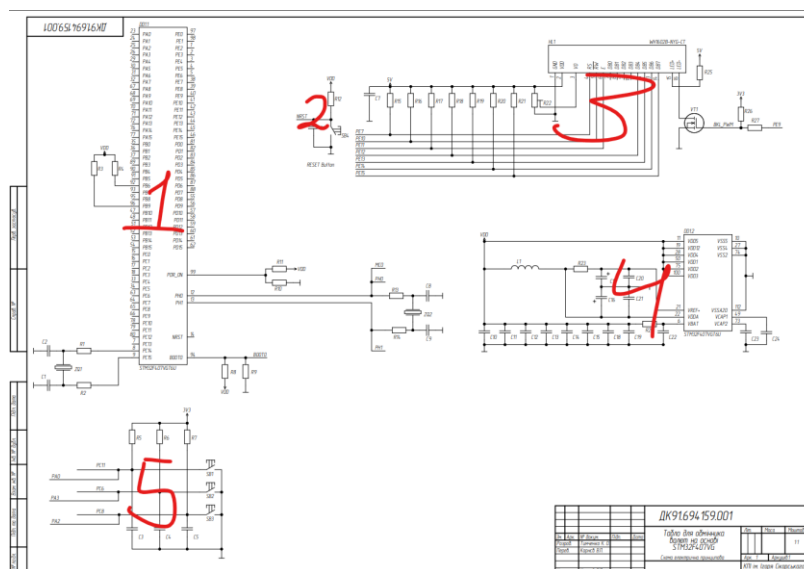


Рис. 2.2 Загальний вигляд схеми електричної принципової з умовною розбивкою на складові

Перша частина – складається з частини мікросхеми DD1 (мікроконтролер STM32F407VGT), кварцових резонаторів (ZQ1, ZQ2), резисторів (R1, R2, R3, R4, R8, R9, R10, R11, R13, R14) та конденсаторів (C1, C2, C8, C9). Це все необхідний мінімум для роботи мікроконтролера.

Друга частина – складається з входу NSRT мікроконтролера STM32F407VGT, резистора R12 та кнопки SB4. Детальніший опис в «2.2.2 - Опис кнопки перезавантаження мікроконтролера».

Третя частина – складається з частини мікросхеми DD1 (мікроконтролер STM32F407VGT), засобу виведення HL1 (WH1602B-NYG-CT), резисторів (R1, R15, R16, R17, R18, R19, R20, R21, R25, R26, R27), потенціометра R22 транзистора VT1 та конденсатора C7. Дана частина схеми, це підключення LCD дисплею.

Четверта частина – складається з частини мікросхеми DD1 (мікроконтролер STM32F407VGT), котушки індуктивності L1, резистора R24 та конденсаторів (C10, C11, C12, C13, C14, C15, C16, C17, C18, C19, C20, C21, C22, C23, C24). Все це слугує для стабілізації напруги живлення. Детальніше в «2.2.1 – Опис схеми стабілізації напруги мікроконтролера»

П'ята частина – складається з частини мікросхеми DD1 (мікроконтролер STM32F407VGT), резисторів (R5, R6, R7), конденсаторів (C3, C4, C5) та кнопок (SB1, SB2, SB3). Це підключення кнопок. Детальніше про цю частину в «2.2.3 - Опис схеми підключення кнопок управління»

2.2 – Опис та призначення деяких частин схеми

2.2.1 – Опис схеми стабілізації напруги мікроконтролера

В цьому розділі зосереджу увагу на конденсаторах C10, C11, C12, C13, C14, C15, C18, C19 (Рис. 2.3).

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		10

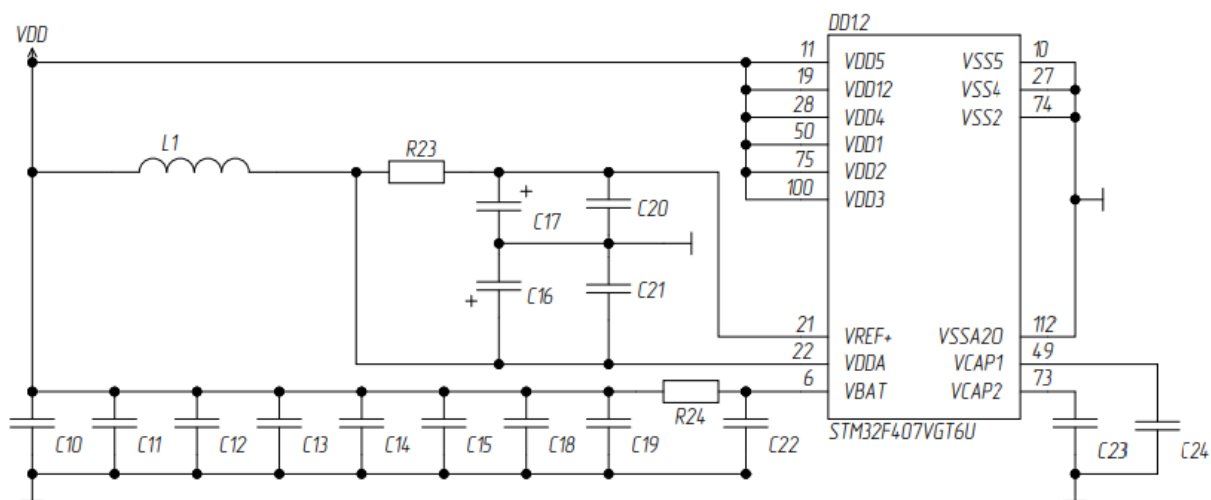


Рис. 2.3 Частина схеми, що відповідає за стабілізацію живлення крупним планом

Особливість цих конденсаторів в тому, що вони під'єднані поралено. Через це виникає питання – чому не використати замість цих багатьох конденсаторів, один, еквівалентної ємності? Відповідь:

Один контакт цих конденсаторів знаходиться на потенціалі землі, а інший на потенціалі живлення VDD, при чому, до цього вузла підключена велика кількість входів живлення (VDD1, VDD2, VDD3, VDD4 ...) і при трасуванні такого підключення до одного конденсатора, ми б отримали дуже велику сумарну довжину провідників цього вузла. Це погано оскільки збільшує паразитний опір, паразитну ємність та робить прилад менш стійким до шумів й завад. Щоб запобігти цим негативним наслідкам, біля кожного порта живлення ставиться окремий конденсатор.

2.2.2 - Опис схеми підключення кнопки перезавантаження мікроконтролера

Особливістю цього входу скидання є те, що для вдалого

перезавантаження мікроконтролера на його вході слід тримати рівень напруги логічного нуля деякий час. Для цього використовується наступна схема (Рис. 2.4):

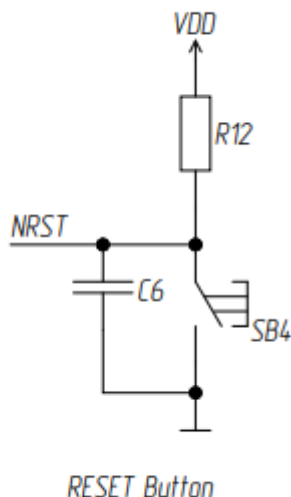


Рис. 2.4 Підключення кнопки перезавантаження пристрою крупним планом

Алгоритм перезавантаження наступний:

- Після довгої не активності кнопки, конденсатор С6 заряджається від живлення VDD через резистор R12. Напруга на вході NRST дорівнює напрузі на конденсаторі, тобто напрузі VDD. Ця напруга відповідає рівню логічної одиниці.
- Після короткого натиснення кнопки SB4, конденсатор С6 встигає розрядитись через цю кнопку на землю. Розрядження конденсатора відбувається настільки швидко, наскільки малі паразитні опори провідників та кнопки.
- Через наявність напруги логічного нуля на вході NRST внутрішній тригер Шмітта мікроконтролера перекидається і починається процес перезавантаження мікроконтролера.

- Оскільки, тепер конденсатор не розряджається через кнопку, а заряджається через резистор, то його зарядження забирає значно більше часу. За цей час мікроконтролер встигає перезавантажитись.
- Потім конденсатор заряджається достатньо, щоб тригер Шмітта знов перекинувся. І все починається заново.

2.2.3 - Опис схеми підключення кнопок управління

В цьому розділі буде пояснене призначення конденсаторів (C3, C4, C5) та резисторів (R5, R6, R7) (Рис. 2.5):

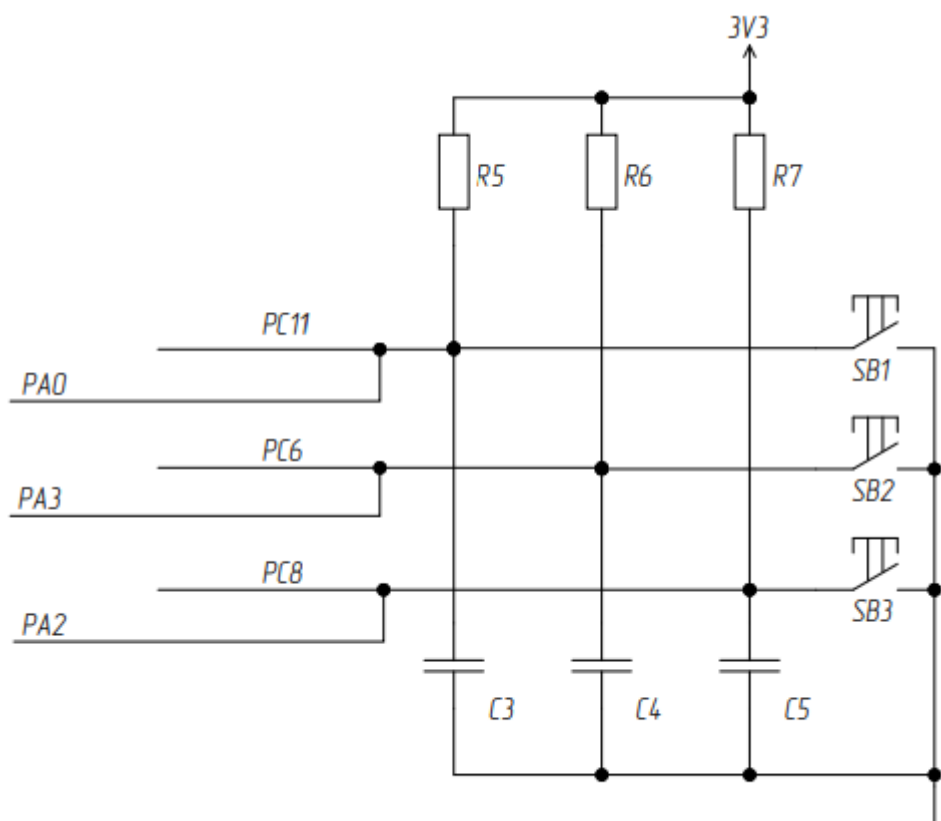


Рис. 2.5 Схема підключення кнопок керування крупним планом

Наведу пояснення для кнопки SB1, для решти кнопок все буде аналогічно.

При використанні кнопки, ми не можемо просто приєднати один її кінець до входу мікроконтролера, а інший до землі і ось чому:

- Після першого ж натискання вивід мікроконтролера набуде потенціалу землі і навіть після розімкнення схеми (ми відпустили кнопку) не втратить його. Через це мікроконтролер просто не буде бачити натискання кнопок, а завжди реєструватиме логічний нуль на вході.
- Через те, що електропровідні елементи кнопки будуть брязкати при натисканні, вхід мікроконтролера буде зчитувати багато імпульсів чи фронтів і це буде призводити до його неправильної роботи.

Для вирішення 2 проблеми є конденсатор, напруга на якому, не може змінитись різко, тому всі це брязкоти будуть згладжені ним.

Для вирішення першої проблеми, все трішки складніше, але вже було описано в «2.2.2 - Опис схеми підключення кнопки перезавантаження мікроконтролера». Ось як це виглядає для SB1:

- При тривалій не активності кнопки SB1, конденсатор C3 заряджається від живлення 3,3В через резистор R5. Напруга на вході PA0 дорівнює напрузі на конденсаторі, тобто напрузі 3,3В. Ця напруга відповідає рівню логічної одиниці.
- Після короткого натиснення кнопки SB1, конденсатор C3 розряджається через цю кнопку на землю.
- Вхід PA0 фіксує відповідний рівень чи фронт (в моєму випадку він фіксує фронт).
- Після відпускання все повторюється по колу.

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		14

РОЗДІЛ 3 - СТРУКТУРА ПРОГРАМИ ТА ЇЇ ОПИС

3.1 – Функції ініціалізації

`void GPIO_init(void);` - функція, що ініціалізує порти вводу\виводу для подальшої роботи з ними. Складається з:

Подання тактування в шині AHB1ENR на GPIO A, E:

```
316 // #####
317 void GPIO_init(void)
318 {
319     // AHB1ENR
320     RCC->AHB1ENR = RCC_AHB1ENR_GPIOAEN;
321     RCC->AHB1ENR = RCC_AHB1ENR_GPIOEEN;
322 }
```

Встановленні режиму General purpose output mode портів A0, A2, A3, E7, E10, E11, E12, E13, E14, E15:

```
323
324 // MODER A
325 GPIOA->MODER &= ~GPIO_MODER_MODE0_0;
326 GPIOA->MODER &= ~GPIO_MODER_MODE0_1;
327 GPIOA->MODER &= ~GPIO_MODER_MODE3_0;
328 GPIOA->MODER &= ~GPIO_MODER_MODE3_1;
329 GPIOA->MODER &= ~GPIO_MODER_MODE2_0;
330 GPIOA->MODER &= ~GPIO_MODER_MODE2_1;
331
332
333 // MODER E
334 GPIOE->MODER |= GPIO_MODER_MODE7_0;
335 GPIOE->MODER &= ~GPIO_MODER_MODE7_1;
336 GPIOE->MODER |= GPIO_MODER_MODE10_0;
337 GPIOE->MODER &= ~GPIO_MODER_MODE10_1;
338 GPIOE->MODER |= GPIO_MODER_MODE11_0;
339 GPIOE->MODER &= ~GPIO_MODER_MODE11_1;
340 GPIOE->MODER |= GPIO_MODER_MODE12_0;
341 GPIOE->MODER &= ~GPIO_MODER_MODE12_1;
342 GPIOE->MODER |= GPIO_MODER_MODE13_0;
343 GPIOE->MODER &= ~GPIO_MODER_MODE13_1;
344 GPIOE->MODER |= GPIO_MODER_MODE14_0;
345 GPIOE->MODER &= ~GPIO_MODER_MODE14_1;
346 GPIOE->MODER |= GPIO_MODER_MODE15_0;
347 GPIOE->MODER &= ~GPIO_MODER_MODE15_1;
348
```

Встановленні режиму Output open-drain портів E7, E10, E11, E12, E13, E14, E15:

```
349
350 // OTYPER E
351 GPIOE->OTYPER &= ~GPIO_OTYPER_OT7;
352 GPIOE->OTYPER &= ~GPIO_OTYPER_OT10;
353 GPIOE->OTYPER &= ~GPIO_OTYPER_OT11;
354 GPIOE->OTYPER &= ~GPIO_OTYPER_OT12;
355 GPIOE->OTYPER &= ~GPIO_OTYPER_OT13;
356 GPIOE->OTYPER &= ~GPIO_OTYPER_OT14;
357 GPIOE->OTYPER &= ~GPIO_OTYPER_OT15;
358
```

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		15

Встановлення логічного рівня нуля на виході E10 (він є входом R\W для LCD):

```

359
360 // ODR
361 GPIOE->ODR &= ~GPIO_ODR_OD10;
362 }
363

```

void lcd_init(void) – функція для ініціалізації LCD 1602:

```

236 void lcd_init(void)
237 {
238     lcd_cmd(0x28); // init
239     lcd_cmd(0x01); // clear
240     lcd_cmd(0x0D); // cursor on
241     my_delay(1000);
242 }

```

void interrupt_init(void) – функція для ініціалізації переривань за заднім фронтом на портах A0, A2, A3:

```

14 // #####
15 // #####
16 void interrupt_init(void)
17 {
18     RCC->APB2ENR |= RCC_APB2ENR_SYSCFGEN;
19
20     SYSCFG->EXTICR[0] |= SYSCFG_EXTICR1_EXTI0_PA;
21     SYSCFG->EXTICR[0] |= SYSCFG_EXTICR1_EXTI3_PA;
22     SYSCFG->EXTICR[0] |= SYSCFG_EXTICR1_EXTI2_PA;
23
24     EXTI->IMR |= EXTI_IMR_IM0;
25     EXTI->IMR |= EXTI_IMR_IM3;
26     EXTI->IMR |= EXTI_IMR_IM2;
27
28     EXTI->FTSR |= EXTI_RTSTR_TR0;
29     EXTI->FTSR |= EXTI_RTSTR_TR3;
30     EXTI->FTSR |= EXTI_RTSTR_TR2;
31
32     NVIC_SetPriority(EXTI0_IRQn, 1);
33     NVIC_SetPriority(EXTI3_IRQn, 2);
34     NVIC_SetPriority(EXTI2_IRQn, 3);
35
36     NVIC_ClearPendingIRQ(EXTI0_IRQn);
37     NVIC_ClearPendingIRQ(EXTI3_IRQn);
38     NVIC_ClearPendingIRQ(EXTI2_IRQn);
39
40     NVIC_EnableIRQ(EXTI0_IRQn);
41     NVIC_EnableIRQ(EXTI3_IRQn);
42     NVIC_EnableIRQ(EXTI2_IRQn);
43
44     __enable_irq();
45 }
46

```

3.2 – Функції для роботи з дисплеєм

`void my_delay(unsigned long delay);` – функція, що створює затримки:

```
363 L
364 void my_delay(unsigned long delay)
365 {
366     delay_counter = 0;
367     for(delay_counter=0;delay_counter<delay;delay_counter++);
368 }
```

`void lcd_cmd(unsigned long cmd)` – функція, що надсилає команду дисплею. Приймає на вхід код команди. Приклад:

```
123
124 lcd_cmd(0x01); // clear
125 lcd_data(0x55);
```

Для реалізації затримок використовується функція `void my_delay(unsigned long delay);`

```
244 void lcd_cmd(unsigned long cmd)
245 {
246     GPIOE->ODR &= ~GPIO_ODR_OD7;
247     GPIOE->ODR &= ~GPIO_ODR_OD11;
248
249     if(cmd&(1<<4)) GPIOE->ODR |= GPIO_ODR_OD12;
250     else GPIOE->ODR &= ~GPIO_ODR_OD12;
251     if(cmd&(1<<5)) GPIOE->ODR |= GPIO_ODR_OD13;
252     else GPIOE->ODR &= ~GPIO_ODR_OD13;
253     if(cmd&(1<<6)) GPIOE->ODR |= GPIO_ODR_OD14;
254     else GPIOE->ODR &= ~GPIO_ODR_OD14;
255     if(cmd&(1<<7)) GPIOE->ODR |= GPIO_ODR_OD15;
256     else GPIOE->ODR &= ~GPIO_ODR_OD15;
257
258     GPIOE->ODR |= GPIO_ODR_OD11;
259     my_delay(400);
260     GPIOE->ODR &= ~GPIO_ODR_OD11;
261     my_delay(400);
262
263     if(cmd&(1<<0)) GPIOE->ODR |= GPIO_ODR_OD12;
264     else GPIOE->ODR &= ~GPIO_ODR_OD12;
265     if(cmd&(1<<1)) GPIOE->ODR |= GPIO_ODR_OD13;
266     else GPIOE->ODR &= ~GPIO_ODR_OD13;
267     if(cmd&(1<<2)) GPIOE->ODR |= GPIO_ODR_OD14;
268     else GPIOE->ODR &= ~GPIO_ODR_OD14;
269     if(cmd&(1<<3)) GPIOE->ODR |= GPIO_ODR_OD15;
270     else GPIOE->ODR &= ~GPIO_ODR_OD15;
271
272     GPIOE->ODR |= GPIO_ODR_OD11;
273     my_delay(400);
274     GPIOE->ODR &= ~GPIO_ODR_OD11;
275     my_delay(40000);
276 }
```

void lcd_cmd(unsigned long cmd) – функція, що надсилає дані в дисплей:

```
278 void lcd_data(unsigned long cmd)
279 {
280     GPIOE->ODR |= GPIO_ODR_OD7;
281     GPIOE->ODR &= ~GPIO_ODR_OD11;
282
283     if(cmd&(1<<4)) GPIOE->ODR |= GPIO_ODR_OD12;
284     else GPIOE->ODR &= ~GPIO_ODR_OD12;
285     if(cmd&(1<<5)) GPIOE->ODR |= GPIO_ODR_OD13;
286     else GPIOE->ODR &= ~GPIO_ODR_OD13;
287     if(cmd&(1<<6)) GPIOE->ODR |= GPIO_ODR_OD14;
288     else GPIOE->ODR &= ~GPIO_ODR_OD14;
289     if(cmd&(1<<7)) GPIOE->ODR |= GPIO_ODR_OD15;
290     else GPIOE->ODR &= ~GPIO_ODR_OD15;
291
292     GPIOE->ODR |= GPIO_ODR_OD11;
293     my_delay(400);
294     GPIOE->ODR &= ~GPIO_ODR_OD11;
295     my_delay(400);
296
297     if(cmd&(1<<0)) GPIOE->ODR |= GPIO_ODR_OD12;
298     else GPIOE->ODR &= ~GPIO_ODR_OD12;
299     if(cmd&(1<<1)) GPIOE->ODR |= GPIO_ODR_OD13;
300     else GPIOE->ODR &= ~GPIO_ODR_OD13;
301     if(cmd&(1<<2)) GPIOE->ODR |= GPIO_ODR_OD14;
302     else GPIOE->ODR &= ~GPIO_ODR_OD14;
303     if(cmd&(1<<3)) GPIOE->ODR |= GPIO_ODR_OD15;
304     else GPIOE->ODR &= ~GPIO_ODR_OD15;
305
306     GPIOE->ODR |= GPIO_ODR_OD11;
307     my_delay(400);
308     GPIOE->ODR &= ~GPIO_ODR_OD11;
309     my_delay(40000);
310 }
311
```

Приклад використання:

```
211     lcd_cmd(0x86); // mov 1
212     lcd_data(0x3F);
213     lcd_cmd(0x86); // mov 1
```

void lcd_print(void) – головна функція, що друкує інформацію на дисплеї.

В ній ж і знаходиться алгоритм роботи. Код цієї та інших функцій буде наведений в Додатку Б.

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		18

3.3 – Функції зовнішніх переривань

void EXTI0_IRQHandler(void); – функція, що відповідає за переривання з пору PA0.

```
47 void EXTI0_IRQHandler(void)
48 {
49     EXTI->IMR &= ~EXTI_IMR_IM0;
50
51     change_mod++;
52     if(change_mod > 4) change_mod = 0;
53     lcd_print();
54
55     EXTI->PR|= EXTI_PR_PR0;
56     EXTI->IMR |= EXTI_IMR_IM0;
57 }
```

void EXTI3_IRQHandler(void); – функція, що відповідає за переривання з пору PA3.

```
59 void EXTI3_IRQHandler(void)
60 {
61     EXTI->IMR &= ~EXTI_IMR_IM3;
62
63     if(change_mod!=0)
64     {
65         switch(change_mod)
66         {
67             case 1:
68                 UAH_to_USD++;
69                 break;
70             case 2:
71                 USD_to_UAH++;
72                 break;
73             case 3:
74                 UAH_to_EUR++;
75                 break;
76             case 4:
77                 EUR_to_UAH++;
78                 break;
79         }
80     }
81     lcd_print();
82
83     EXTI->PR|= EXTI_PR_PR3;
84     EXTI->IMR |= EXTI_IMR_IM3;
85 }
```

void EXTI2_IRQHandler(void); – функція, що відповідає за переривання з пору PA2.

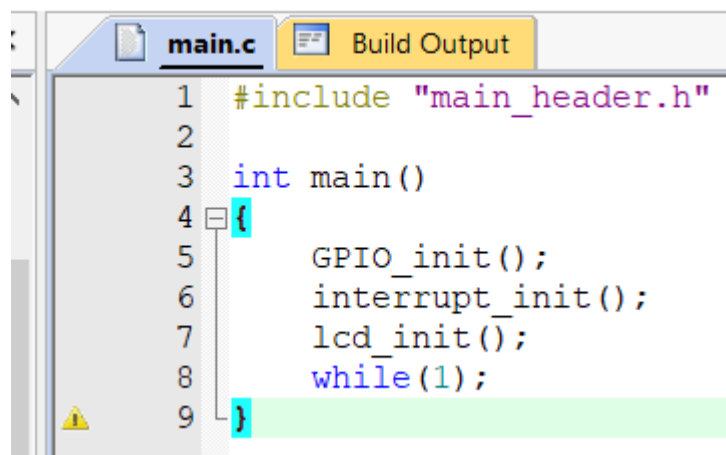
```

87 void EXTI2_IRQHandler(void)
88 {
89     EXTI->IMR &= ~EXTI_IMR_IM2;
90
91     if(change_mod!=0)
92     {
93         switch(change_mod)
94         {
95             case 1:
96                 UAH_to_USD--;
97                 break;
98             case 2:
99                 USD_to_UAH--;
100                break;
101             case 3:
102                 UAH_to_EUR--;
103                 break;
104             case 4:
105                 EUR_to_UAH--;
106                 break;
107         }
108     }
109     lcd_print();
110
111     EXTI->PR |= EXTI_PR_PR2;
112     EXTI->IMR |= EXTI_IMR_IM2;
113 }
114

```

3.4 – Загальний алгоритм роботи

- Відбувається ініціалізація портів вводу виводу:



```

1 #include "main_header.h"
2
3 int main()
4 {
5     GPIO_init();
6     interrupt_init();
7     lcd_init();
8     while(1);
9 }

```

- Далі йде ініціалізація переривань.
- Дисплей отримує відповідні команди, що задаються тип його роботи, та очищають пам'ять, повертаючи курсор на нульову адресу.

- У випадку натискання клавіші SB1 змінюється глобальна змінна change_mod. Якщо ця змінна 0 – то йде просте виведення інформації, якщо 1 – то є можливість змінювати глобальну зміну UAH_to_USD. Цей принцип поширюється й на інші числа, що відображаються на дисплеї.

```

5 extern volatile unsigned long UAH_to_USD; // 1
6 extern volatile unsigned long USD_to_UAH; // 2
7 extern volatile unsigned long UAH_to_EUR; // 3
8 extern volatile unsigned long EUR_to_UAH; // 4

```

Таким чином забезпечується можливість налаштовувати інформацію, що виводиться. Детальніше про алгоритм зміни інформації, що виводиться описано в «4.2 – Зміна інформації, що відображається».

РОЗДІЛ 4 - КЕРІВНИЦТВО З ЕКСПЛУАТАЦІЇ ПРИСТРОЮ

4.1 – Запуск приладу

Для того, щоб запустити прилад слід зробити наступнику послідовність дій:

1. Зібрати прилад чи макет, за схемою наведеною в документі ЕЗ ДК91.694159.001
2. Завантажити програму з «Додаток Б. Лістинг програми» в мікроконтролер
3. Перезавантажити мікроконтролер натиснувши кнопку SB4
4. Натиснути кнопку SB2 або SB3 для того, щоб на екрані появилась початкова інформація

Результат цих дій наведений на Рис. 4.1:



Рис. 4.1 Інформація, що відображається на табло за замовчуванням

4.2 – Зміна інформації, що відображається

Для того, щоб налаштувати інформацію, що відображається слід виконати наступний алгоритм:

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		22

1. Запустити прилад так як це описано в «4.1 – Запуск приладу»
2. Вибрати число, котре потрібно змінити натисканням кнопки SB1.
Після її натискання, в числа котре ми можемо змінювати буде замість крапи мигати знак питання. Це вказує на те, що ми будемо змінювати саме це число. Наприклад, для того, щоб змінити вартість купівлі долара, слід натиснути кнопку SB1 один раз. Ось як це буде виглядати (Рис. 4.2):



Рис. 4.2 Вигляд який приймає табло після двох натискань SB1

3. Для того, щоб збільшити це число на одну копійку, слід натиснути клавішу SB2, для того, щоб зменшити SB3. Наприклад, збільшимо вибране в минулому пункті число на 2 копійки. Для цього натиснем клавішу SB2 два рази. Ось результат (Рис. 4.3):



Рис. 4.3 Вигляд який приймає табло після двох натискань кнопки SB2

4. Після того, як ми встановили, це число, потрібно продовжити натискати SB1 до тих пір, доки табло не вийдемо з режиму редагування інформації.

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		24

ВИСНОВКИ

Отже, в результаті виконання даного курсового проекту мною було розроблено табло для обмінника валют на основі STM32F407VG. Даний пристрій відображає вартість для купівлі та продажу долара та євро. Є можливість легко налаштувати іншу ціну.

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

Разом з цією ПЗ також були створені документи ЕЗ та ПЕЗ. Враховуючи наявність цієї документації та робочого макету приладу, курсовий проект вважаю успішно виконаним.

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		25

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-MQO0qDnYAWWngzQlEu0A3g&authuser=1>

- Дата звернення 18.01.2023.

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

https://drive.google.com/drive/u/1/folders/1v0gVm-2L170GAK_M5WBIVXUyFJqr9-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

- Дата звернення 18.01.2023.

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		26

ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ

ТЕХНІЧНЕ ЗАВДАННЯ

1. Найменування та галузь використання

- Пристрій цифровий спеціалізований «Табло для обмінника валют 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 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.

- Відображення вартості купівлі й продажу
- Відображення курсу одразу 2 валют
- Можливість легко налаштовувати вартість валют

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		27

5.2. Технічні характеристики

- Надійність й простота в експлуатації
- 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.

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		28

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.

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		29

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 "main_header.h"
```

```
int main()
```

```
{
```

```
    GPIO_init();
```

```
    interrupt_init();
```

```
    lcd_init();
```

```
    while(1);
```

```
}
```

Вміст файлу main_header.h:

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

```
extern volatile unsigned long delay_counter;
```

```
extern volatile unsigned long UAH_to_USD;// 1
```

```
extern volatile unsigned long USD_to_UAH;// 2
```

```
extern volatile unsigned long UAH_to_EUR;// 3
```

```
extern volatile unsigned long EUR_to_UAH;// 4
```

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		31

```
extern volatile unsigned char change_mod;
```

```
//-----for lcd
```

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

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

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

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

```
//-----for interrupt
```

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

```
void EXTI0_IRQHandler(void);
```

```
void EXTI3_IRQHandler(void);
```

```
void EXTI2_IRQHandler(void);
```

```
//-----for other
```

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

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

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		32

Вміст файлу src.c:

```
#include "main_header.h"

volatile unsigned long delay_counter = 0;

volatile unsigned long UAH_to_USD = 3523;// 1
volatile unsigned long USD_to_UAH = 3424;// 2
volatile unsigned long UAH_to_EUR = 3515;// 3
volatile unsigned long EUR_to_UAH = 3476;// 4

volatile unsigned char change_mod = 0;

// #####

// #####

void interrupt_init(void)
{
    RCC->APB2ENR |= RCC_APB2ENR_SYSCFGEN;

    SYSCFG->EXTICR[0] |= SYSCFG_EXTICR1_EXTI0_PA;
    SYSCFG->EXTICR[0] |= SYSCFG_EXTICR1_EXTI3_PA;
    SYSCFG->EXTICR[0] |= SYSCFG_EXTICR1_EXTI2_PA;
```

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		33

EXTI->IMR |= EXTI_IMR_IM0;

EXTI->IMR |= EXTI_IMR_IM3;

EXTI->IMR |= EXTI_IMR_IM2;

EXTI->FTSR |= EXTI_RTSR_TR0;

EXTI->FTSR |= EXTI_RTSR_TR3;

EXTI->FTSR |= EXTI_RTSR_TR2;

NVIC_SetPriority(EXTI0_IRQn,1);

NVIC_SetPriority(EXTI3_IRQn,2);

NVIC_SetPriority(EXTI2_IRQn,3);

NVIC_ClearPendingIRQ(EXTI0_IRQn);

NVIC_ClearPendingIRQ(EXTI3_IRQn);

NVIC_ClearPendingIRQ(EXTI2_IRQn);

NVIC_EnableIRQ(EXTI0_IRQn);

NVIC_EnableIRQ(EXTI3_IRQn);

NVIC_EnableIRQ(EXTI2_IRQn);

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		34

```

        __enable_irq();

    }

void EXTI0_IRQHandler(void)

{

    EXTI->IMR &= ~EXTI_IMR_IM0;


    change_mod++;

    if(change_mod > 4) change_mod = 0;

    lcd_print();


    EXTI->PR|= EXTI_PR_PR0;

    EXTI->IMR |= EXTI_IMR_IM0;

}


void EXTI3_IRQHandler(void)

{

    EXTI->IMR &= ~EXTI_IMR_IM3;


    if(change_mod!=0)

    {

        switch(change_mod)

```

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		35

```

        {

            case 1:

                UAH_to_USD++;

            break;

            case 2:

                USD_to_UAH++;

            break;

            case 3:

                UAH_to_EUR++;

            break;

            case 4:

                EUR_to_UAH++;

            break;

        }

    }

    lcd_print();

    EXTI->PR|= EXTI_PR_PR3;

    EXTI->IMR |= EXTI_IMR_IM3;

}

```

```

void EXTI2_IRQHandler(void)

```

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		36

```

{

EXTI->IMR &= ~EXTI_IMR_IM2;


if(change_mod!=0)

{

    switch(change_mod)

    {

        case 1:

            UAH_to_USD--;

            break;

        case 2:

            USD_to_UAH--;

            break;

        case 3:

            UAH_to_EUR--;

            break;

        case 4:

            EUR_to_UAH--;

            break;

    }

}

}

lcd_print();

```

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		37


```

EXTI->PR|= EXTI_PR_PR2;

EXTI->IMR |= EXTI_IMR_IM2;

}

// #####

// #####

void lcd_print(void)

{

    lcd_cmd(0x01); // clear

    lcd_data(0x55);

    lcd_data(0x53);

    lcd_data(0x44);

    lcd_cmd(0x84); // mov 1

    if(UAH_to_USD <= 9999)

    {

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

        lcd_data(0x30 + ((UAH_to_USD -

(UAH_to_USD%100))/100)%10);

        lcd_data(0xD0);

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

```

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		38

```

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

    }

    else

    {

        lcd_data(0x78);

        lcd_data(0x78);

        lcd_data(0xD0);

        lcd_data(0x78);

        lcd_data(0x78);

    }


    lcd_cmd(0x8B); // mov 2

    if( USD_to_UAH <= 9999)

    {

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


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

        lcd_data(0xD0);

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

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

    }

```

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		39

```

else
{
    lcd_data(0x78);

    lcd_data(0x78);

    lcd_data(0xD0);

    lcd_data(0x78);

    lcd_data(0x78);

}

lcd_cmd(0xC0);// mov

lcd_data(0x45);

lcd_data(0x55);

lcd_data(0x52);


lcd_cmd(0xC4);// mov 3

if( UAH_to_EUR <= 9999)
{
    lcd_data(0x30 + (UAH_to_EUR - (UAH_to_EUR%1000))/1000);

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

    lcd_data(0xD0);

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

```

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		40

```

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

    }

    else

    {

        lcd_data(0x78);

        lcd_data(0x78);

        lcd_data(0xD0);

        lcd_data(0x78);

        lcd_data(0x78);

    }


    lcd_cmd(0xCB);// mov 4

    if( UAH_to_EUR <= 9999)

    {

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

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

        lcd_data(0xD0);

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

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

    }

    else

```

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		41

```

{

    lcd_data(0x78);

    lcd_data(0x78);

    lcd_data(0xD0);

    lcd_data(0x78);

    lcd_data(0x78);

}

```

```

if(change_mod!=0)

```

```

{

    switch(change_mod)

    {

        case 1:

            lcd_cmd(0x86);// mov 1

            lcd_data(0x3F);

            lcd_cmd(0x86);// mov 1

            break;

        case 2:

            lcd_cmd(0x8D);// mov 2

            lcd_data(0x3F);

            lcd_cmd(0x8D);// mov 2

            break;

```

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		42

case 3:

lcd_cmd(0xC6);// mov 3

lcd_data(0x3F);

lcd_cmd(0xC6);// mov 3

break;

case 4:

lcd_cmd(0xCD);// mov 4

lcd_data(0x3F);

lcd_cmd(0xCD);// mov 4

break;

}

}

my_delay(1000);

}

void lcd_init(void)

{

lcd_cmd(0x28);// init

lcd_cmd(0x01);// clear

lcd_cmd(0x0D);// cursor on

my_delay(1000);

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		43

```
}
```

```
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(400);
```

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

```
    my_delay(400);
```

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		44

```

        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;


        GPIOE->ODR |= GPIO_ODR_OD11;

        my_delay(400);

        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;

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(400);

GPIOE->ODR &= ~GPIO_ODR_OD11;

my_delay(400);


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;


GPIOE->ODR |= GPIO_ODR_OD11;

```

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		46

```

my_delay(400);

GPIOE->ODR &= ~GPIO_ODR_OD11;

my_delay(4000);

}

// #####

// #####

void GPIO_init(void)

{

// AHB1ENR

RCC->AHB1ENR = RCC_AHB1ENR_GPIOAEN;

RCC->AHB1ENR = RCC_AHB1ENR_GPIOEEN;

// MODER A

GPIOA->MODER &= ~GPIO_MODER_MODE0_0;

GPIOA->MODER &= ~GPIO_MODER_MODE0_1;

GPIOA->MODER &= ~GPIO_MODER_MODE3_0;

GPIOA->MODER &= ~GPIO_MODER_MODE3_1;

GPIOA->MODER &= ~GPIO_MODER_MODE2_0;

GPIOA->MODER &= ~GPIO_MODER_MODE2_1;

```

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		47

// MODER E

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;

// 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;

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		48

```
GPIOE->OTYPER &= ~GPIO_OTYPER_OT14;
```

```
GPIOE->OTYPER &= ~GPIO_OTYPER_OT15;
```

```
// ODR
```

```
GPIOE->ODR &= ~GPIO_ODR_OD10;
```

```
}
```

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

```
{
```

```
    delay_counter = 0;
```

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

```
}
```

					ДК91.694159.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		49